CORESA 2006 CAEN

9 et 10 Novembre

COmpression et REprésentation des Signaux Audiovisuels

Université de Caen Basse-Normandie

Laboratoire Universitaire des Sciences Appliquées de Cherbourg
LUSAC
EA 2607

france telecom R&D

GRYE

ENSICAEN

Actes / Résumés

La RWHT+P pour un codage multirésolution sans perte avancé

Olivier Déforges[1] Marie Babel[1] Jean Motsch[2]

IETR UMR CNRS 6164 - Groupe Image et Télédétection

[1] INSA de Rennes

[2] LESTP/CREC Saint-Cyr, Écoles de Coëtquidan

{odeforge,mbabel}@insa-rennes.fr,jean.motsch@st-cyr.terre.defense.gouv.fr

Résumé

Cet article présente une méthode complète de codage sans perte multirésolution, à forte scalabilité sémantique. En particulier, une forme réversible de la classique Transformée de Walsh Hadamard (RWHT) est tout d'abord introduite comme alternative aux transformées sans perte standard. Une représentation pyramidale et des schémas de décomposition reposant sur cette transformée sont ensuite proposés pour un codage multirésolution réversible. Des améliorations significatives y sont apportées en ajoutant deux concepts supplémentaires : la "résolution localement adaptative" à travers une représentation quadtree, et une étape de prédiction. Les résultats expérimentaux montrent en final que la méthode RWHT+P proposée aboutit à d'excellentes performances comparées à celles de l'état de l'art.

Mots clefs

Scalabilité, codage réversible, représentation pyramidale.

1 Introduction

Les nouvelles générations de codecs d'images doivent bien sûr être efficaces en termes de performances de compression, mais aussi fournir des fonctionnalités avancées telles que la scalabilité, le contrôle du débit, l'encodage de régions d'intérêt, ou encore la description de scènes. De plus, un schéma de codage unique, capable de compresser depuis les bas débits jusqu'au sans perte, constituerait une solution souhaitable pour des usages variés. Dans [1], nous avons introduit la méthode LAR (Locally Adaptive Resolution) comme codage irréversible efficace à bas et moyens débits. Cette technique repose sur un schéma global d'encodage à deux couches : une couche spatiale et une couche spectrale. Le premier codec spatial fournit une image principale à bas débit, alors que le codec spectral encode la texture locale. La qualité des images compressées par le LAR a été évaluée par un autre laboratoire, et reconnue meilleure que celle des images obtenues par Jpeg-2000 [1]. La méthode LAR repose en partie sur une décomposition en taille de blocs variable de type quadtree, estimée à partir de l'activité locale. Cette structure particulière a permis une extension du schéma simple à une représentation en régions construite à partir des images fortement compressées, et autorisant ensuite un codage par région d'intérêt [2]. Récemment, nous avons également proposé une version modifiée du codec permettant du codage sans perte tout en augmentant son caractère scalable [3] : les couches spatiales et spectrales initiales y ont été substituées par deux décompositions multirésolution de type quadtree, à partir d'une solution modifiée de la transformée en S. Ce papier présente une alternative à cette méthode quant au noyau de décomposition. Cette nouvelle génération de codec LAR, appelée "RWHT+P", surpasse les performances précédentes à la fois pour du codage à bas débit et du sans perte. Nous nous limiterons toutefois ici au dernier cas. Plus précisément, la première partie de l'article s'attachera à introduire une technique permettant de rendre la transformée de Walsh-Hadamard (WHT), avec un noyau 2×2, réversible. Le reste du papier proposera un schéma de codage sans perte complet avec une scalabilité avancée en résolution.

En codage sans perte, la compression et décompression de données source doivent résulter dans l'exacte récupération de tout le signal d'origine. Le codage des images sans perte est nécessaire dans les applications où la dégradation des données n'est pas tolérée. C'est le cas par exemple pour des applications dans le domaine médical, en télédétection, ou encore pour l'archivage d'images et de vidéos à qualité studio. L'état de l'art en codage sans perte peut être grossièrement divisé en deux approches : les méthodes prédictives dans le domaine spatial, avec des codeurs populaires tels que CALIC [4], et les méthodes basées transformation généralement reposant sur la théorie des ondelettes. Le principal intérêt des codecs fondés ondelettes reside dans le fait qu'ils proposent un codage scalable, avec la possibilité d'une représentation multirésolution de l'image. Les transformées sans perte ont la particularité d'une correspondance non équivoque d'entiers à entiers, au contraire de l'essentiel des transformées dites avec pertes. La plupart des méthodes utilisent pour cela le concept "d'arrondi" [5]. La WHT est une technique très connue pour la compression des images et du signal. Beaucoup de méthodes multirésolution, utilisant cette transformation sur des blocs

2×2, ont été proposées dans la littérature. A des fins de compression réversible, une version modifiée de la WHT 1D a été introduite par P. Lux [6], puis popularisée par Said [5] et connue sous le nom de transformée en 'S", ou "transformée en ondelettes entière de Haar" [7]. La transformée en S est actuellement reconnue comme une des meilleures bases d'ondelettes entières parmi celles existantes pour le codage réversible [8]. Afin d'améliorer la compression, une étape de prédiction a été ajoutée à la transformée elle-même, conduisant à la technique très connue de méthode "S+P", qui fut plus tard généralisée à travers le concept de "lifting scheme" [9].

Le paragraphe 2 introduit l'adaptation réalisée sur la transformée non réversible $WHT_{2\times2}$ pour une forme sans perte, appelée la $RWHT$. Le paragraphe 3 présente un schéma pyramidal de compression sans perte fondé sur la $RWHT$. Ce schéma est ensuite amélioré à travers l'ajout de deux fonctionnalités supplémentaires : une décomposition quadtree et une phase de prédiction/interpolation. Finalement, nous concluons dans le paragraphe 4.

2 La transformée RWHT

De manière à retrouver les données en entrée à partir du vecteur transformé, la transformée $WHT_{2\times2}$ initiale a été transformée dans sa normalisation pour aboutir à la transformée en S.

$$WHT_{2\times2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & -1 \end{bmatrix}. \quad (1)$$

La correspondance non équivoque d'entiers à entiers est ensuite possible grâce aux opérations duales d'arrondi pendant les transformations directes et inverses. Une transformation 2D peut être réalisée en appliquant successivement la transformée suivant les deux directions. Cependant, ce noyau est moins efficace à des fins de codage avec pertes, dès lors qu'il augmente la dynamique des coefficients de hautes fréquences.

Nous introduisons ainsi une technique de transformation 2D réversible utilisant directement la matrice de transformation formelle de la $WHT_{2\times2}$.

Soit $U_{2\times2}$ le bloc d'entrée avec :

$$U_{2\times2} = \begin{bmatrix} u_0 & u_1 \\ u_2 & u_3 \end{bmatrix}. \quad (2)$$

Le bloc transformé $Z_{2\times2}$ est alors défini par :

$$\begin{aligned} Z_{2\times2} &= WHT_{2\times2}(U_{2\times2}) \\ &= W_{2\times2}\, U_{2\times2}\, W_{2\times2} \\ &= \begin{bmatrix} z_0 & z_1 \\ z_2 & z_3 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} u_0+u_1+u_2+u_3 & u_0+u_1-u_2-u_3 \\ u_0-u_1+u_2-u_3 & u_0-u_1-u_2+u_3 \end{bmatrix}. \end{aligned} \quad (3)$$

Soit $\hat{Z}_{2\times2}$ le bloc arrondi de $Z_{2\times2}$ tel que :

$$\begin{aligned} \hat{Z}_{2\times2} = Round(Z_{2\times2}) &= \begin{bmatrix} \hat{z}_0 & \hat{z}_1 \\ \hat{z}_2 & \hat{z}_3 \end{bmatrix} \\ &= \begin{bmatrix} Round_{z_0}[z_0] & Round_{z_1}[z_1] \\ Round_{z_2}[z_2] & Round_{z_3}[z_3] \end{bmatrix}. \end{aligned} \quad (4)$$

$Round_{z_i}[.]$ correspond à l'opération d'arrondi appliquée sur z_i : arrondi inférieur ($\lfloor . \rfloor$) ou supérieur ($\lceil . \rceil$).

La transformée inverse est identique à la transformée directe. Notons $\tilde{U}_{2\times2}$ le bloc inverse transformé de $\hat{Z}_{2\times2}$, et $\hat{U}_{2\times2}$ le bloc arrondi de $\tilde{U}_{2\times2}$. Définir une transformée réversible implique que $\hat{U}_{2\times2} = U_{2\times2}$ malgré les opérations d'arrondi. Pour y parvenir directement dans l'espace 2D, nous proposons une méthode de contrôle des valeurs arrondies fondée sur la fonction de parité $P(.)$ telle que :

$$P(x) = \begin{cases} o \text{ si } x \text{ impair} \\ e \text{ si } x \text{ pair} \end{cases}, \qquad x \in \mathbb{N}. \quad (5)$$

En posant $z_0 = \lfloor z_0 \rfloor + \frac{\epsilon}{2}, \epsilon \in \{0,1\}$, et lorsqu'il est substitué dans l'équation (3), $Z_{2\times2}$ peut être exprimé selon :

$$Z_{2\times2} = \frac{1}{2} \begin{bmatrix} 2\lfloor z_0 \rfloor + \epsilon & 2(\lfloor z_0 \rfloor - u_2 - u_3) + \epsilon \\ 2(\lfloor z_0 \rfloor - u_1 - u_3) + \epsilon & 2(\lfloor z_0 \rfloor - u_1 - u_2) + \epsilon \end{bmatrix} \quad (6)$$

A cette étape, deux cas demeurent possibles.

Somme paire : Si $P(\sum_{i=0}^{3} u_i) = e$, alors $\epsilon = 0$ et $\hat{Z}_{2\times2} = Z_{2\times2}$. Ceci implique des valeurs entières reconstruites :

$$\tilde{u}_0 = \frac{1}{2}(4\lfloor z_0 \rfloor - 2(u_1+u_2+u_3)) = \frac{1}{2}(2u_0) = u_0 \quad (7)$$

et $\hat{u}_0 = u_0$.

Somme impaire : Si $P(\sum_{i=0}^{3} u_i) = o$, le problème de l'arrondi de $Z_{2\times2}$ est déplacé à celui d'arrondir $\epsilon/2$ pour chaque coefficient. Soit $\Delta_i \in \{0,1\}$ l'arrondi de $\epsilon/2$ pour z_i ($\Delta_i = Round_{z_i}\left[\frac{\epsilon}{2}\right] = \frac{\epsilon}{2} + \frac{\epsilon_i}{2}, \epsilon_i \in \{-1,+1\}$).

$$\begin{aligned} \hat{Z}_{2\times2} &= \begin{bmatrix} \lfloor z_0 \rfloor + \Delta_0 & \lfloor z_0 \rfloor - u_2 - u_3 + \Delta_1 \\ \lfloor z_0 \rfloor - u_1 - u_3 + \Delta_2 & \lfloor z_0 \rfloor - u_1 - u_2 + \Delta_3 \end{bmatrix} \\ &= Z_{2\times2} + \frac{1}{2} \begin{bmatrix} \epsilon_0 & \epsilon_1 \\ \epsilon_2 & \epsilon_3 \end{bmatrix}. \end{aligned} \quad (8)$$

Ainsi les coefficients reconstruits s'écrivent :

$$\tilde{U}_{2\times2} = \frac{1}{2} \begin{bmatrix} 2(u_0-\epsilon) + (\Delta_0+\Delta_1+\Delta_2+\Delta_3) & 2u_1 + (\Delta_0+\Delta_1-\Delta_2-\Delta_3) \\ 2u_2 + (\Delta_0-\Delta_1+\Delta_2-\Delta_3) & 2u_3 + (\Delta_0-\Delta_1-\Delta_2+\Delta_3) \end{bmatrix}. \quad (9)$$

Dès lors, la reconstruction exacte implique :

$$\begin{cases} \Delta_0+\Delta_1+\Delta_2+\Delta_3 = 2\epsilon = 2 \\ \Delta_0+\Delta_2 = \Delta_1+\Delta_3 \\ \Delta_0+\Delta_1 = \Delta_2+\Delta_3 \\ \Delta_0+\Delta_3 = \Delta_1+\Delta_2 \end{cases} \quad (10)$$

Clairement, le système d'équations sur les valeurs de Δ_i ne peut être résolu. De ce fait, aucun arrondi systématique, comme pour la transformée en S, ne permet une transformation réversible.
L'alternative réside dans un contrôle des opérations d'arrondi, de sorte que le procédé de décodage soit à même de distinguer des valeurs reconstruites entières ou non entières. Fixer $\{\Delta_i\}$ tel que $P(\sum_{i=0}^{3}\Delta_i) = o$ aboutit uniquement à des valeurs réelles pour les coefficients $\tilde{u}_i$. Si nous imposons que $\sum_{i=0}^{3}\Delta_i = 1$, alors

$$\begin{aligned} \Delta_0+\Delta_1+\Delta_2+\Delta_3=1 &\Rightarrow 4\frac{\epsilon}{2}+\frac{\epsilon_1}{2}+\frac{\epsilon_2}{2}+\frac{\epsilon_3}{2}+\frac{\epsilon_3}{2}=1 \\ &\Rightarrow \epsilon_0+\epsilon_1+\epsilon_2+\epsilon_3=-2. \end{aligned} \quad (11)$$

Par exemple, l'ensemble $\{\epsilon_0 = 1,\ \epsilon_1 = \epsilon_2 = \epsilon_3 = -1\}$ est une solution pour la condition dans (11). Avec un tel choix, la transformation inverse est finalement réalisée en deux phases :

1. calcul de $\tilde{U}_{2\times2} = WHT(\hat{Z}_{2\times2})$.
2. si $\tilde{u}_i$ réel, alors calcul d'un nouveau $\tilde{U}_{2\times2}$ tel que :

$$\tilde{U}_{2\times2} = WHT\left(\hat{Z}_{2\times2} - \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}\right). \quad (12)$$

On vérifie aisément que $\tilde{U}_{2\times2} = U_{2\times2}$ dans tous les cas.

3 Codage sans perte par la pyramide RWHT+P

Notations :
$I(i,j)$ dénote le pixel dans une image I avec les coordonnées (i,j), $I(\mathbf{b}^N(i,j))$ le bloc $\mathbf{b}^N(i,j)$ dans I incluant l'ensemble des pixels $\{I(N.i, N.j), \dots I(N.i+N-1, N.j+N-1)\}$.

3.1 La pyramide RWHT

Nous introduisons la pyramide $\{Y_l\}_{l=0}^{L_{max}}$ comme la représentation multirésolution d'une image I de taille $N_x \times N_y$, où L_{max} est le niveau haut de la pyramide et $l = 0$ le niveau pleine résolution. Comme pour le cas de la $WHT_{2\times2}$ classique, la pyramide est construite itérativement en regroupant quatre blocs pour former un bloc moyen au niveau supérieur :

$$\begin{cases} l=0, & Y_0(i,j) = I(i,j); \\ l>0, & Y_l(i,j) = \left\lfloor \frac{1}{4}\sum_{k=0}^{1}\sum_{m=0}^{1} Y_{l-1}(2x+k, 2y+m) \right\rfloor. \end{cases} \quad (13)$$

avec $0 \le i \le N_x^l$, $0 \le j \le N_y^l$, où $N_x^l = N_x/2^l$ et $N_y^l = N_y/2^l$.

La décomposition *top-down* de la pyramide consiste à encoder le bloc transformé $Z_l(\mathbf{b}^2(i,j))$ par la $RWHT$ de chaque bloc d'entrée $Y_l(\mathbf{b}^2(i,j))$. De (3) et (13), nous obtenons :

$$\begin{aligned} &Y_{l+1}(i,j) = \left\lfloor \frac{z_{0_l}(2i,2j)}{2} \right\rfloor \\ &\Rightarrow z_{0_l}(2i,2j) = 2 \times Y_{l+1}(i,j) + \epsilon_{z_{0_l}(2i,2j)}, \end{aligned} \quad (14)$$

avec $\epsilon_{z_{0_l}(2i,2j)} \in \{0,1\}$.
Ainsi, la composante DC de chaque bloc est reconstruite sans ambiguïté depuis le niveau supérieur plus un bit additionnel. Ce bit est ici codé séparément des autres coefficients. Si nous notons $\dot{Z}_l(\mathbf{b}^2(i,j))$ le bloc WHT transformé de $Y_l(\mathbf{b}^2(i,j))$ avec ce seul bit comme composante DC ($\dot{z}_{0_l} = \epsilon_{z_{0_l}}$), alors la reconstruction à partir du niveau supérieur du bloc WHT courant est donnée par :

$$\begin{aligned} &\tilde{Y}_l(\mathbf{b}^2(i,j)) = EXP(Y_{l+1}(i,j)) + \tilde{\dot{Y}}_l(\mathbf{b}^2(i,j)) \\ &\text{avec } \tilde{\dot{Y}}_l(\mathbf{b}^2(i,j)) = WHT_{2\times2}^{-1}\left(\dot{Z}_l\left(\mathbf{b}^2(i,j)\right)\right). \end{aligned} \quad (15)$$

La fonction *EXP* duplique simplement une valeur d'un nœud de l'arbre à ses quatre fils.
A cette étape, nous possédons une représentation pyramidale et un encodage classiques fondés sur une transformation par $WHT_{2\times2}$, mais avec l'exception d'une possible décomposition sans perte. Le tableau 1 donne les valeurs entropiques d'ordre zéro pour une compression réversible avec à la fois la transformée de S et la RWHT proposée. Le niveau supérieur a été codé pour les deux cas par un simple MICD. Les résultats démontrent que la méthode proposée améliore la compression tout en généralisant le noyau 2D non réversible de la WHT à une version réversible.

3.2 Décomposition quadtree

Nos précédents travaux ont été consacrés à l'élaboration d'un schéma de codage fondé sur une représentation à taille de blocs variable, efficace en termes de compression à la fois pour les hauts et les très bas débits [1]. L'idée est ici de montrer que ce concept, appliqué dans un contexte de codage sans perte, apporte des améliorations significatives au schéma pyramidal original.
Une partition quadtree suppose la décomposition de l'image entière en blocs de taille $N \times N$, avec $N = 2^k$, et $k \in \mathbb{N}^+$. La représentation pyramidale précédente induit une décomposition dyadique, ordinairement associée à une partition multiniveaux quadtree $QP^{[2^{L_{max}} \dots 2^l]}$, où le niveau l de la pyramide spécifie également la résolution la plus fine. Plus généralement, nous considérons une partition quadtree globale de l'image. Ainsi, $QP^{[N_{max} \dots N_{min}]}$ définit les tailles de blocs autorisées, et le paramètre $N_l \in [N_{max} \dots N_{min}]$ donne la limite supérieure des tailles de blocs à décomposer au niveau l de la pyramide. A titre

	Entropie (bpp)						
Image	Raw	S	RWHT	CALIC	S+P	RWHT+P	RWHT+P Qd
Barbara2	7.51	5.45	5.47	4.93	5.04	5.06	**4.89**
Hotel	7.57	5.11	5.09	**4.57**	4.97	4.83	4.60
Lena	7.44	4.77	4.75	4.33	4.33	4.30	**4.19**
Gold	7.60	5.08	5.06	4.65	4.73	4.73	**4.63**
Peppers	7.57	4.89	4.87	4.58	4.67	4.54	**4.43**
us	4.84	3.65	3.64	3.60	3.78	3.78	**3.26**
tools	7.52	5.95	5.95	5.53	5.73	5.71	**5.50**
Average	**7.15**	4.99	4.97	4.60	4.75	4.71	**4.49**

TAB. 1 – *Comparaison des approches proposées avec l'état de l'art. Entropie du premier ordre (bit/pixels).*

d'exemple, une partition globale $QP^{[32\ldots2]}$ conduit à l'encodage de la seule représentation utilisant les blocs de taille 32 à 2, tandis que la notation $N_0 = 4$ implique la décomposition au niveau 0 des blocs de taille 4 et 2.
Enfin, le paramètre L_{min} indique le dernier niveau à encoder. Ainsi, pour tous les niveaux inférieurs à L_{min}, la valeur de l'ensemble des nœuds de la pyramide résulte simplement d'une phase de duplication.
La partition de l'image est réalisée en fonction de l'activité locale, estimée par un gradient morphologique (différence entre les valeurs minimales et maximales) calculé sur chaque bloc. Ainsi, une première phase de décomposition de la pyramide s'attache à raffiner uniquement les petits blocs situés sur les contours, selon l'expression :

$$\tilde{Y}_l(\mathbf{b}^2(i,j)) = \begin{cases} EXP(Y_{l+1}(i,j)) + \tilde{\tilde{Y}}_l(\mathbf{b}^2(i,j)), \\ \quad \text{si } \mathbf{b}^2(i,j) \notin QP^{[N_{max}\ldots N_l[} \\ \quad \text{et } l \geq L_{min} \\ EXP(Y_{l+1}(i,j)) \text{ sinon} \end{cases} \quad (16)$$

avec $l < L_{max}$. $\tilde{Y}_l(\mathbf{b}^2(i,j))$ représente le bloc reconstruit du bloc original $Y_l(\mathbf{b}^2(i,j))$.
La figure 1 illustre les étapes de codage relatives à ce modèle (codeur C_1).

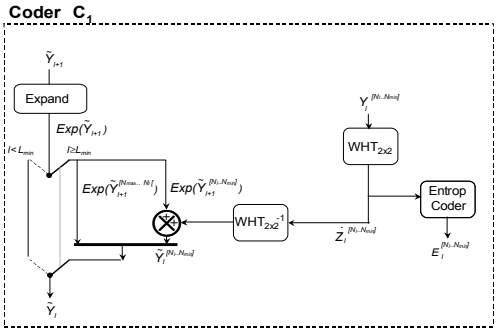


FIG. 1 – *Simple pyramidal coder*

La deuxième descente de la pyramide consiste en la décomposition de tous les blocs du niveau courant qui n'ont pas été traités lors de la première phase : l'information de texture locale est ainsi encodée.
La décomposition quadtree utilisée ici possède plusieurs avantages :

1. le nombre de niveau de décomposition est doublé ($2 \times L_{max}$), augmentant la scalabilité du schéma,
2. des images de bonne qualité sont disponibles à bas débit,
3. l'approche agit comme une **modélisation de contexte** objective, décorrélant naturellement les lois entropiques des erreurs de prédiction : entropie haute pour la première descente, entropie basse pour la seconde.

3.3 Pyramide RWHT et prédiction

Si la prédiction et l'interpolation constituent deux fonctions relativement proches dans le domaine spatial, elles poursuivent cependant deux objectifs différents. La première vise à optimiser la compression en limitant l'erreur de prédiction, alors que la seconde augmente la qualité et la résolution de l'image. Un bon prédicteur n'est pas nécessairement un bon interpolateur, et vice-versa. Les deux fonctions s'avèrent cependant utiles à notre schéma de codage. En particulier, à un niveau décomposition donné, la première descente requiert à la fois une phase de prédiction, pour l'encodage des blocs décomposés, et une phase d'interpolation pour le lissage des zones homogènes (blocs non décomposés). De ce fait, nous proposons une méthode unifiée pour les deux fonctions via la définition d'un procédé d'estimation unique.
Dans ce qui suit, nous notons $\breve{Y}_l(\mathbf{b}^2(i,j))$ le bloc reconstruit du bloc orignal $Y_l(\mathbf{b}^2(i,j))$. L'estimation consiste en la reconstruction linéaire des valeurs inconnues à partir de leur valeur moyenne de bloc. L'information inter et intra niveau est alors exploitée dans un contexte 2D selon :

$$\begin{aligned} &\text{Initialisation :} \\ &\quad \breve{Y}_l(\mathbf{b}^2(i,j)) = \tilde{Y}_{l+1}(i,j), \forall (i,j) \in \tilde{Y}_{l+1} \\ &\text{Estimation :} \\ &\quad \breve{Y}_l(2i+k, 2j+m) = \tilde{Y}_{l+1}(i,j) \\ &\qquad + \beta_m \left(\breve{Y}_l(2i+k, 2j-1+3m) - \tilde{Y}_{l+1}(i,j) \right) \\ &\qquad + \beta_k \left(\breve{Y}_l(2i-1+3k, 2j+m) - \tilde{Y}_{l+1}(i,j) \right), \\ &\quad (k,m) \in \{0,1\}^2, \end{aligned} \quad (17)$$

avec β_m et β_k les poids appliqués au gradient local.
Sans quantification, les valeurs de voisinage diffèrent selon la configuration, et correspondent

– soit à une valeur exactement reconstruite (position déjà traitée au niveau courant par un codage exact),
– soit à une valeur moyenne de bloc (position non traitée au niveau courant),
– soit à une valeur interpolée (position déjà traitée mais non encodée).

Dans ce dernier cas, il existe une inter-dépendance des données, dans la mesure où la valeur du voisinage a été partiellement calculée à partir de la valeur moyenne du bloc courant. Ceci implique, pour deux positions adjacentes de blocs, une relation entre les coefficients β. Si l'on considère, pour deux positions $(2i, 2j)$ et $(2i-1, 2j)$, uniquement les relations horizontales, l'expression 17 devient

$$\left|\begin{array}{l} \breve{Y}_l(2i,2j) = \\ \qquad \tilde{Y}_{l+1}(i,j) + \beta_0 \left(\breve{Y}_l(2i-1,2j) - \tilde{Y}_{l+1}(i,j)\right) \\ \breve{Y}_l(2i-1,2j) = \\ \qquad \tilde{Y}_{l+1}(i-1,j) + \beta_1 \left(\breve{Y}_l(2i,2j) - \tilde{Y}_{l+1}(i-1,j)\right) \\ \Rightarrow \breve{Y}_l(2i,2j) = \\ \qquad \tilde{Y}_{l+1}(i,j) + \beta_0 \Big(\tilde{Y}_{l+1}(i,j) - \tilde{Y}_{l+1}(i-1,j)\Big)(\beta_1 - 1)\,. \end{array}\right. \tag{18}$$

Si un gradient symétrique est de plus imposé de telle sorte que

$$\breve{Y}_l(2i-1,2j) - \tilde{Y}_{l+1}(i-1,j) = -\left(\breve{Y}_l(2i,2j) - \tilde{Y}_{l+1}(i,j)\right), \tag{19}$$

alors nous obtenons la relation suivante :

$$\beta_0 = \frac{\beta_1}{1-\beta_1}, \qquad \beta_1 \in [0, 0.5]. \tag{20}$$

L'effet de l'estimation se calibre via la valeur de β_1 :
– pour $\beta_1 = 0$, $\breve{Y}_l(2i,2j) = \tilde{Y}_{l+1}(i,j)$: l'estimation s'avère sans effet (le bloc est reconstruit par sa valeur moyenne),
– pour $\beta_1 = 0.25$, $\breve{Y}_l(2i,2j) - \tilde{Y}_{l+1}(i,j) = \tilde{Y}_{l+1}(i-1,j) - \breve{Y}_l(2i-1,2j)$: la pente est régulière entre les deux points interpolés (lissage de l'image),
– pour $\beta_1 = 0.5$, $\breve{Y}_l(2i,2j) = \breve{Y}_l(2i-1,2j)$: les points reconstruits adjacents sont identiques (accentuation des contours).

En fait, nos expérimentations ont montré que le mode lissage ($\beta_1 = 0.25$) conduit à la meilleure prédiction.

La figure 2 donne le nouveau schéma de codage incluant la phase d'estimation. Le codeur C_2 s'appuie uniquement sur des relations intra-niveaux, et est adapté à une reconstruction progressive de l'image (le rehaussement de la résolution d'obtient directement à partir de l'interpolation de l'image issue du niveau précédent). Le codeur C_3 tire aussi parti des valeurs reconstruites au niveau courant, conduisant naturellement à des performances de compression supérieures.

Les résultats de la compression sans perte par la méthode proposée sont regroupés dans le tableau 1, et comparés à ceux obtenus par les codeurs de l'état de l'art CALIC (non scalable) et S+P (scalable). Le choix de S+P, plutôt que tout

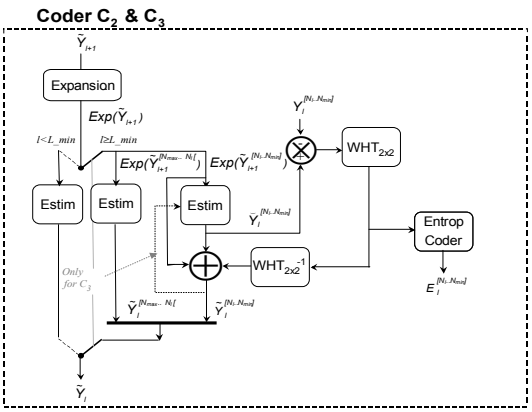


FIG. 2 – *Pyramidal coder with prediction step*

autre noyau d'ondelettes entières, a été motivé par deux états de fait : d'une part, cette solution demeure l'une des meilleures, et d'autre part, un codeur est disponible et permet de réaliser des expérimentations sans mettre en œuvre la couche de codage entropique.

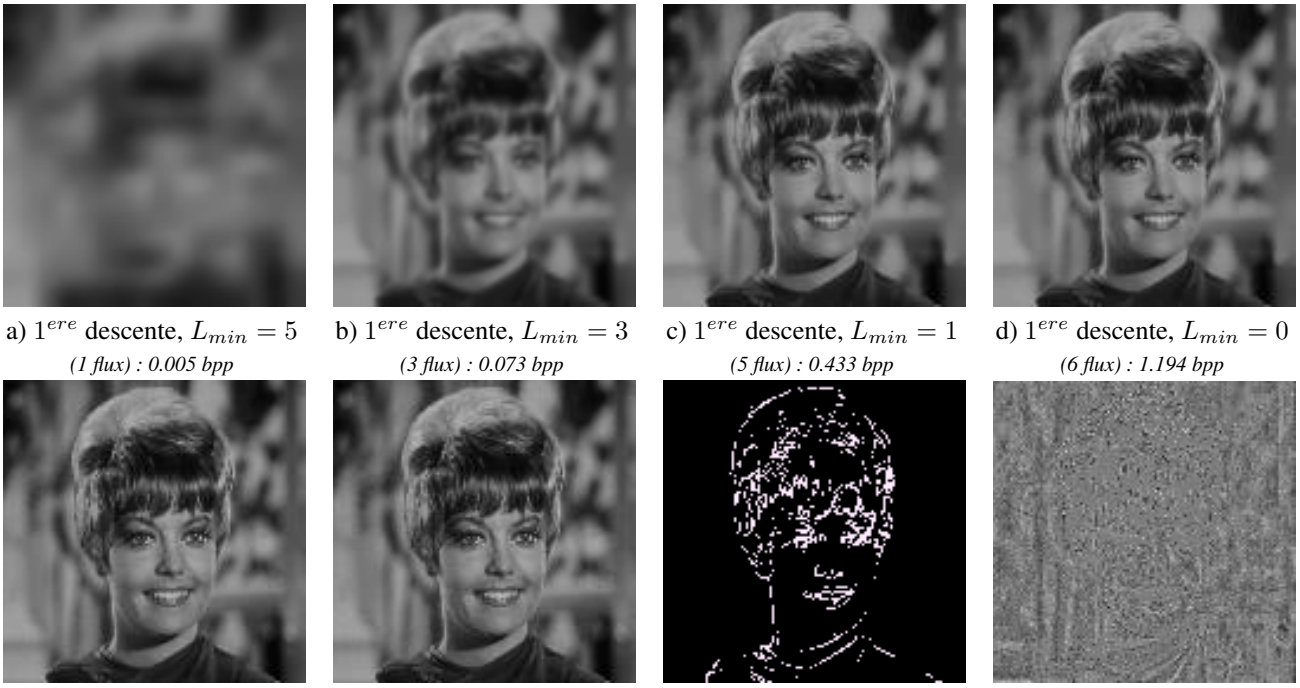
La configuration "RWHT+P" correspond au mode C_3 du codeur sans partition (décomposition en une seule descente). "RWHT+P & Qd" implique une partition $QP^{[64\ldots2]}$, avec $N_0 = 2$ et $N_l = 2^l$. Nous remarquons immédiatement que la séparation des lois entropiques pour les symboles à encoder, suivant la décomposition quadtree, compense largement le coût de sa structure : les résultats de codage de cette configuration dépassent largement ceux de S+P et de CALIC.

Afin d'illustrer la **scalabilité sémantique** de notre approche, la figure 3 montre des images intermédiaires obtenues lors du processus de décomposition pyramidale. Pour six niveaux de décomposition, l'encodage sans perte de l'image nécessite onze flux successifs ($1 + 2 \times 5$). Il est à noter que la distorsion visuelle est essentiellement due à un effet de flou, moins perturbant pour l'observateur que les effets de blocs ou de rebonds. Les images reconstruites à la fin de la première passe sont caractérisées par des contours globalement conservés et des zones homogènes lissées.

4 Conclusion

La transformée en S a été initialement conçue afin d'introduire la notion de réversibilité dans la transformée Walsh-Hadamard classique $WHT_{2\times2}$. La première partie de l'article s'est attachée à démontrer que, moyennant des opérations d'arrondi suivant un critère de parité, le noyau de la $WHT_{2\times2}$ possède à lui seul cette propriété. La décomposition pyramidale RWHT qui en résulte présente des performances meilleures que celles de la pyramide en S.

Deux innovations majeures ont de plus été exposées, contribuant à une décorrélation supplémentaire de l'information, ainsi qu'à des améliorations significatives du schéma original, à savoir : une décomposition pyramidale conditionnelle au contenu de l'image, et une phase de



a) 1^{ere} descente, $L_{min} = 5$ *(1 flux) : 0.005 bpp*
b) 1^{ere} descente, $L_{min} = 3$ *(3 flux) : 0.073 bpp*
c) 1^{ere} descente, $L_{min} = 1$ *(5 flux) : 0.433 bpp*
d) 1^{ere} descente, $L_{min} = 0$ *(6 flux) : 1.194 bpp*
e) 2^{eme} descente, $L_{min} = 1$ *(10 flux) : 2.014 bpp*
f) 2^{eme} descente, $L_{min} = 0$ *(11 flux) : 3.87 bpp*
g) Partition quadtree
h) Image d'erreurs x10 Image (f) - image (d)

FIG. 3 – *Codage scalable sans perte sur "Zelda", partition quadtree associée* $QP^{[64...2]}$

prédiction. Le schéma scalable global surpasse, en termes de compression sans perte, à la fois S+P et CALIC. En outre, il réalise une représentation multirésolution localement adaptée de l'image, permettant la reconstruction à très bas débit d'images de bonne qualité visuelle.

Cette méthode de codage a aussi prouvé son efficacité dans le contexte de la compression avec pertes à bas débit. Ainsi, il est possible d'introduire une étape de quantification de l'erreur ajustée au contenu : quantification fine sur les gros blocs (l' œil humain s'avère plus sensible aux variations de luminance sur les zones uniformes), quantification grossière sur les petits blocs (l' œil humain est moins sensible aux variations sur les contours).

Une application directe de cette méthode consiste en la définition d'un système d'archivage des images à haute résolution issues du musée du Louvre. La bibliothèque numérique permettra l'accès à différentes qualités d'images. Ces travaux sont supportés par le ministère français de la recherche dans le cadre du projet ANR "TSAR".

Références

[1] O. Deforges et J. Ronsin. Region of interest coding for low bit-rate image transmission. Dans *ICME*, volume 1, pages 107–110, july 2000.

[2] O. Deforges et J. Ronsin. Supervised segmentation at low bit rates for region representation and color image compression. Dans *ICME*, volume 1, pages 665–668, 2002.

[3] M. Babel, O. Deforges, et J. Ronsin. Interleaved s+p pyramidal decomposition with refined prediction model. Dans *ICIP*, volume 2, pages 750–753, 2005.

[4] X. Wu et N. Memon and. A Context-based, Adaptive, Lossless/Nearly-Lossless Coding Scheme for Continuous-Tone Images (CALIC). *International Standards Organization working document, ISO/IEC SC29/WG 1/N256*, 1995.

[5] A. Said et W. Pearlman. Reversible image compression via multiresolution representation and predictive coding. Dans *Visual Communication and Image Processing*, volume 209, pages 664–674. SPIE, Novembre 1993.

[6] P. Lux. A novel set of closed orthogonal functions for picture coding. *Archiv für Electronik und übertragungs Technik*, 31(7) :267–274, 1977.

[7] K. Komatsu et K. Sezaki. Lossless 2d discrete walsh-hadamard transform. *Proc. IEEE ICASSP*, May 2001.

[8] G. C. K. Abhayaratne. *Lossless and Nearly Lossless Digital Video Coding*. Thèse de doctorat, Univ. of Bath, 2002.

[9] Wim Sweldens. The Lifting Scheme : A Construction of Second Generation Wavelets. *SIAM Journal on Mathematical Analysis*, 29(2) :511–546, 1998.

Codage audio scalable basé sur le codeur MPEG-4 SSC

David VIRETTE[1], Jean-Bernard RAULT[2], Pierrick PHILIPPE[2]

France Telecom, Division R&D/TECH
prénom.nom@orange-ft.com
[1] : Laboratoire SSTP ; 2, Av. Pierre Marzin. 22307 Lannion Cedex France
[2] : Laboratoire IRIS ; 4, rue du Clos Courtel – BP 59 – 35512 Cesson Sévigné Cedex France

Résumé

Dans cet article, nous présentons un codeur audio scalable basé sur le codeur paramétrique MPEG-4 SSC (SinuSoidal Coder). Ce nouveau codeur combine deux stratégies de codage, la première étant le codage audio sinusoïdal (MPEG-4 SSC) et la deuxième étant le codage de type ACELP (Algebraic Code-Excited Linear Prediction), habituellement utilisé pour les signaux de parole. Nous montrons que cette approche permet d'une part, d'améliorer la qualité audio des codeurs paramétriques (sinusoïdaux) à bas-débits et d'autre part, d'offrir une flexibilité en terme de compromis qualité/débit comparé aux codeurs audio traditionnels.

Mots clefs

Codage audio paramétrique, codage sinusoïdal, MPEG4-SSC, scalabilité, ACELP.

1 Introduction

Depuis l'introduction du CD dans les années 80, et plus récemment avec l'explosion de l'Internet, les besoins en compression des signaux audio se sont rapidement développés. Les codeurs audio développés et standardisés par ISO/MPEG, comme le MP3, l'AAC ou l'HE-AAC présentés dans [1] et [2], sont largement utilisés de nos jours pour des applications de diffusion et de téléchargement de signaux audio. Ces algorithmes de codage audio, qui appartiennent à la famille des codeurs par transformée, exploitent les caractéristiques du système auditif humain, et notamment les effets de masquage fréquentiel, afin de réduire au maximum la distorsion perçue par l'auditeur sous contrainte de débit.

De nouvelles techniques de codage audio, généralement appelées codage audio paramétrique, ont été proposées plus récemment. Ces techniques s'appuient sur une décomposition du signal audio selon un modèle de codage simulant la façon dont le son est produit. Le signal audio est découpé en trames (quelques ms) pour être analysé relativement au modèle choisi. Les paramètres du modèle sont alors extraits, quantifiés et codés pour être transmis ou stockés. Au décodeur, le signal est re-synthétisé à l'aide des paramètres reçus. Citons en particulier le modèle sinusoïdal qui permet de modéliser les signaux audio à l'aide de simples oscillateurs, dont les paramètres (amplitudes, fréquences et phases) varient lentement dans le temps. Ces modèles ont été initialement développés dans les années 80 pour coder la parole en bande téléphonique [3].

Ces schémas d'analyse/synthèse ont ensuite été généralisés à tout type de signaux audio notamment avec le modèle Sinusoïdes + Bruit [4]. Dans ce modèle, le signal résiduel, obtenu une fois les composantes sinusoïdales retirées, est modélisé par un processus stochastique (bruit blanc) mis en forme temporellement et fréquentiellement.

Plus récemment, des modèles Sinusoïdes + Transitoires + Bruit ont été proposés afin d'améliorer la représentation des signaux percussifs [5].

Des algorithmes de codage audio ont été développés sur chacun de ces modèles. Nous pouvons citer par exemple le codeur MPEG-4 HILN (Harmonic and Individual Lines and Noise) [6] ou encore le codeur MPEG-4 SSC (SinuSoidal Coder) [7]. La qualité de ces codeurs paramétriques souffre d'un manque de naturel de par la limitation du nombre de composantes sinusoïdales sélectionnées et surtout par l'utilisation d'un simple modèle stochastique du résiduel.

Dans cet article, nous commencerons par présenter brièvement le codeur audio paramétrique MPEG-4 SSC en nous intéressant plus particulièrement à la modélisation du signal résiduel. Ensuite, nous proposerons un nouveau codeur audio basé sur l'association du codeur SSC et du codage ACELP. Nous verrons comment cette nouvelle structure de codage offre une plus grande flexibilité en termes de débit. Finalement, nous comparerons les performances de la solution proposée en comparaison avec le codeur MPEG-4 SSC. Cette comparaison sera effectuée à l'aide d'une mesure objective de qualité et par la réalisation d'un test subjectif formel.

2 Le codage audio paramétrique

Cette section présentera le standard MPEG-4 SSC qui est l'état de l'art en matière de codage audio paramétrique,

puis nous donnerons les différents points faibles du modèle utilisé.

2.1 Le codeur MPEG-4 SSC (SinuSoidal Coder)

Le codeur MPEG-4 SSC s'appuie sur un modèle Sinusoïdes + Transitoires + Bruit. Ce codeur fonctionne en bande HiFi à 44.1 kHz de fréquence d'échantillonnage. Les différentes composantes sonores de ce modèle sont représentées de la façon suivante :

- transitoires : sinusoïdes contraintes par une enveloppe temporelle;
- sinusoïdes : sinusoïdes contrôlées en amplitude, phase et fréquence;
- bruit : bruit aléatoire large bande mis en forme temporellement et spectralement.

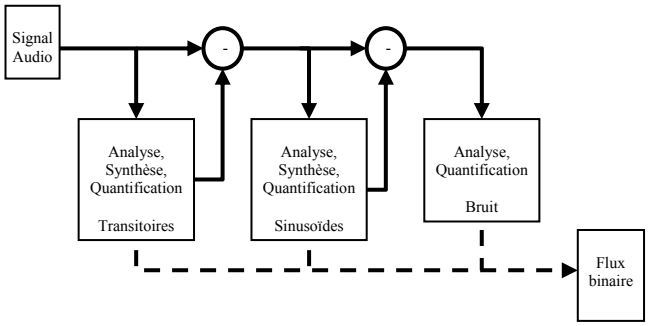


Figure 1 – *Codeur MPEG-4 SSC*

La figure 1 donne le schéma fonctionnel du codeur SSC. Les paramètres du modèle sont extraits en trois étapes successives. Tout d'abord, les transitoires sont détectées en mesurant les variations rapides et importantes de l'énergie du signal, puis modélisées et soustraites du signal original. Ensuite sur le signal restant, les composantes tonales qui sont perceptivement les plus importantes sont détectées et modélisées par des sinusoïdes, puis soustraites. Enfin, le signal déduit des deux étapes précédentes est considéré comme une composante de bruit. Il est modélisé par son enveloppe temporelle et fréquentielle. Les paramètres issus de ces trois étapes de codage sont ensuite quantifiés et multiplexés dans un flux binaire pour la transmission.

Le décodeur réalise les opérations de décodage et de synthèse des trois composantes du modèle afin de générer un signal perceptivement proche du signal original.

Nous allons nous intéresser plus particulièrement au module de synthèse de bruit décrit à la Figure 2. Comme le montre cette figure, la composante « Bruit » est synthétisée par un bruit large bande. Ce bruit est tout d'abord mis en forme temporellement à partir d'une enveloppe temporelle transmise sous forme de LSFs (Line Spectral Frequencies) [8] et convertie dans le domaine temporel. Le bruit est ensuite ajusté en énergie par des gains, également transmis. Les paramètres étant transmis trame par trame, un module de fenêtrage et d'Overlap-Add est ensuite utilisé pour la reconstruction du signal. Enfin, ce bruit est mis en forme spectralement par un filtre de Laguerre, qui offre une bonne résolution fréquentielle dans les basses fréquences [9].

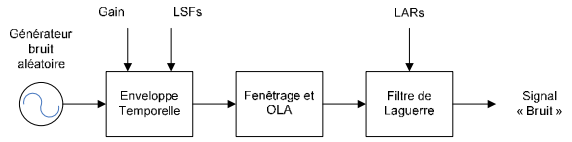


Figure 2 – *Synthétiseur du « Bruit » du décodeur MPEG-4 SSC*

Lors des tests de vérification réalisés par MPEG, le codeur MPEG-4 SSC a obtenu des notes MUSHRA entre *Convenable* et *Bonne* pour un débit de 24 kbps en stéréo [10].

2.2 Limites du codage audio paramétrique

Un problème bien connu concernant les modèles Transitoires + Sinusoïdes + Bruit est que, en général, le « bruit » résiduel n'est pas réellement un bruit et ceci pour les raisons suivantes :

- Le nombre limité de sinusoïdes transmises implique que le signal résiduel peut encore contenir des composantes tonales;
- Les paramètres des sinusoïdes (amplitude, phase et fréquence) peuvent avoir été mal estimés. La soustraction de ces composantes, quantifiées, peut entrainer la présence de caractéristiques tonales dans le résiduel;

De plus certains signaux audio ne sont pas adaptés au modèle (nombre fini de sinusoïdes), ce qui implique que le résiduel est fortement « coloré » et donc mal modélisé par un processus stochastique. La qualité des codeurs paramétriques souffre en général d'un manque de réalisme. Des informations de localisation ou d'ambiances sont souvent éliminées, ce qui entraine, en général, un manque de « présence » et de « naturel ».

Une conséquence importante de ces deux dernières limitations est que, même en augmentant le débit associé à ce codeur, la transparence ne peut être atteinte. En se basant sur ces limites du codage audio paramétrique, nous proposons donc une nouvelle architecture afin d'en améliorer la qualité.

3 SSC-ACELP scalable

3.1 Codeur

Ayant présenté le codeur SSC et les limitations associées au modèle Sinusoïdes + Transitoires + Bruit, nous allons considérer une nouvelle architecture de codage associant le codeur SSC avec un codage ACELP en sous-bande comme le montre la Figure 3. Dans cette nouvelle architecture de codage, le codeur SSC, tel que décrit dans la section précédente, est utilisé comme codeur principal. Ensuite, un codage du résiduel SSC est réalisé en sous-bandes, suivant ainsi les principes psychoacoustiques basé sur la sensibilité de l'oreille humaine en fréquence. Cette découpe en sous-bandes permet de mieux répartir le débit des sous-codeurs sur chaque sous-bande. Il est ainsi possible d'associer un débit plus élevé aux sous-bandes basses, qui seront confiées à un codage ACELP. Pour les hautes fréquences, le module de synthèse de bruit sera souvent suffisant pour assurer un codage de bonne qualité. On le voit donc, le débit peut être consacré à représenter efficacement les premières bandes qui sont les plus significatives perceptivement.

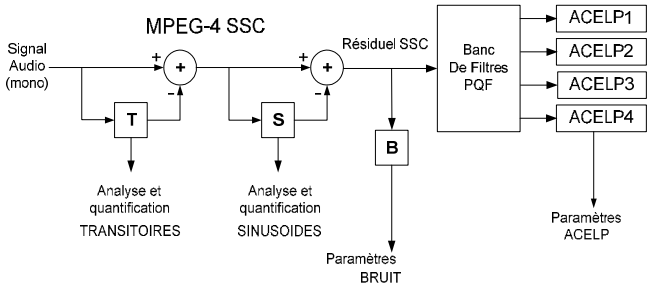


Figure 3 – *Codeur MPEG-4 SSC + ACELP*

La séparation en quatre sous-bandes est réalisée par un banc de filtre de type PQF (Polyphase Quadrature Filter) utilisé dans le processus de contrôle de gain du MPEG-2 AAC [11]. Les coefficients du banc de filtres d'analyse sont donnés par la formule suivante :

$$h_i(n) = \frac{1}{4}\cos\left(\frac{(2i+1)(2n+5)\pi}{16}\right)Q(n)$$

$$0 \le n \le 95, 0 \le i \le 3$$

Avec $Q(n) = Q(95-n), 48 \le n \le 95$

Q(n) représente le filtre à réponse impulsionnelle finie (FIR) prototype passe-bas de longueur 96.

Le schéma de codage décrit à la Figure 3 permet de définir directement un format de flux binaire scalable associant des couches additionnelles de codage ACELP (bande 1 à 4) au cœur SSC.

Les modules de codage ACELP ont été adaptés à partir du codeur AMR-WB (Adaptive Multi-Rate – WideBand) normalisé au 3GPP comme codeur conversationnel en bande élargie [12]. La trame de l'AMR-WB est composée de 4 sous-trames de 64 échantillons. Pour chaque sous-trame, un filtre de prédiction linéaire, une excitation adaptative (pitch et gain) et une excitation algébrique (impulsions et gain) sont sélectionnées pour modéliser le signal. L'AMR-WB possède plusieurs débits de fonctionnement définis principalement par les tailles des dictionnaires algébriques utilisés. Ces dictionnaires sont imbriqués de par leur construction. Le débit d'un mode de l'AMR-WB est donc défini par le nombre d'impulsions +/-1 sélectionnées pour construire l'excitation algébrique.
Dans le codeur SSC-ACELP, les modules ACELP travaillent sur des trames composées de 6 sous-trames de 64 échantillons. Les modifications apportées à l'AMR-WB portent principalement sur la résolution de la recherche du pitch pour l'excitation adaptative (pitch entier uniquement) et sur la quantification des gains (2 gains quantifiés en absolu et 4 en relatif). Les différents débits du codeur SSC-ACELP sont définis par les dictionnaires algébriques sélectionnés dans chaque sous-bande.

3.2 Décodeur

Le décodeur associé est décrit à la Figure 4. Nous pouvons noter que dans un premier temps un décodage conforme au MPEG-4 SSC est réalisé. Le signal alors généré offre la qualité du codeur paramétrique. Ensuite, en fonction des couches ACELP reçues, les différentes sous-bandes préalablement décodées sont remplacées.

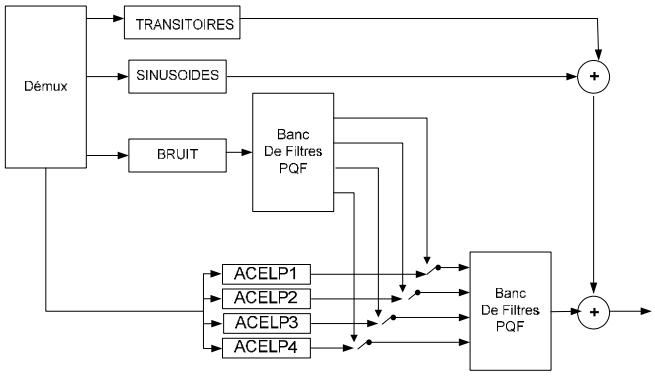


Figure 4 – *Décodeur scalable MPEG-4 SSC + ACELP*

La Figure 5 présente le train binaire associé au codeur SSC-ACELP. Dans le cas le plus simple, il comporte 5 couches permettant d'améliorer la qualité en remplaçant la synthèse de bruit du SSC dans une sous-bande par le décodage ACELP associé. Toutefois, cette structure de train binaire peut être enrichie de couches supplémentaires de raffinement des excitations algébriques des modules

ACELP. Ainsi, les différentes sous-bandes seront encodées par un ACELP multi-étage. Dans ce cas particulier, pour réduire la complexité, des méthodes de transcodage ACELP peuvent être exploitées à l'encodage lors de la recherche des codes algébriques [13]. On pourra par exemple effectuer la recherche dans le dictionnaire ACELP le plus riche afin de favoriser la qualité du débit le plus élevé, puis « dégrader » le code algébrique choisi en supprimant certaines impulsions afin qu'il reste compatible avec les débits plus faibles.

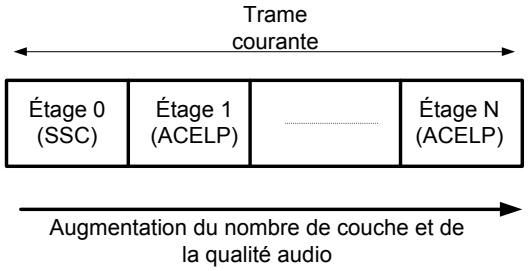


Figure 5 – *Flux binaire MPEG-4 SSC + ACELP pour la scalabilité*

Dans cette section, nous avons présenté la nouvelle architecture de codage proposée. En associant le codage audio paramétrique (SSC) au débit nominal de 24 kbps avec des modules de codage ACELP multi-étage, il est possible d'obtenir une granularité d'environ 4 kbps par couche entre 24 kbps et 128 kbps pour offrir une amélioration continue de la qualité perçue.

4 Performances

4.1 Test subjectif

Dans cette section, nous allons présenter les résultats d'un test subjectif qui a été réalisé dans le but d'évaluer la qualité audio de l'architecture de codage SSC-ACELP. Ce test visait à montrer que l'association du MPEG-4 SSC avec une seule sous-bande de codage ACELP améliore la qualité audio. Dans ce mode, le débit associé à la partie Sinusoïdes + Transitoires + Bruit est d'environ 18 kbps, alors que le débit associé à la première sous-bande ACELP est d'environ 6 kbps. Le module de codage ACELP de la première sous-bande de fréquence utilisé dans ce cas est le dictionnaire algébrique de plus faible débit (2 impulsions sur les 64 positions). Ce mode de codage est donc comparé au SSC à un débit de 24 kbps.
Le test d'écoute a été réalisé en suivant la méthodologie de test CMOS avec l'échelle de notation définie dans la Recommandation ITU-R BS.562-3. Selon cette méthodologie de test, pour chaque signal audio, l'ordre d'écoute est Ref/A/B, avec Ref correspondant au signal de référence (signal original dans notre cas), A et B sont les signaux à évaluer présentés dans un ordre aléatoire non connu du sujet (« test en aveugle »). Dans notre cas, A et B représentaient soit le signal audio encodé avec le MPEG-4 SSC, soit avec le SSC-ACELP, tous deux à un débit de 24 kbps. Les signaux audio de test étaient composés des 12 signaux critiques habituellement utilisés par MPEG pour l'évaluation des codecs audio. Cette liste est donnée à la Figure 6.

Item	Description
es01	vocal (Suzanne Vega)
es02	German speech
es03	English speech
si01	Harpsichord
si02	Castanets
si03	Pitch pipe
sm01	Bagpipes
sm02	Glockenspiel
sm03	Plucked strings
sc01	Trumpet solo and orchestra
sc02	Orchestral piece
sc03	Contemporary pop music

Figure 6 – *Liste des signaux de test*

Huit sujets ont participé à ce test. La Figure 7 montre les résultats de ce test pour chaque signal et en moyenne sur l'ensemble des 12 signaux. Cette figure montre le score moyen pour chaque signal, ainsi que l'intervalle de confiance à 95%. Il apparaît que le codage SSC-ACELP améliore la qualité sur les échantillons de parole (es01, es02 et es03) de manière significative. Par contre, sur les échantillons de musique, il n'y a pas de différence significative à débit équivalent. Ces résultats peuvent s'expliquer par le fait que la parole encodée par un codeur sinusoïdal manque de naturel. L'utilisation d'un schéma de codage ACELP permet donc d'améliorer la qualité sur ces signaux critiques.

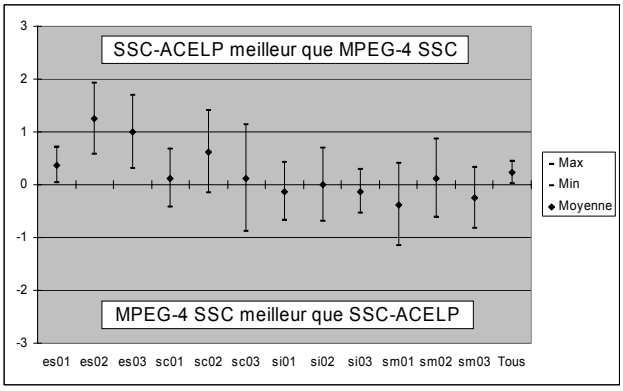


Figure 7 – *Résultats du test subjectif*

4.2 Mesure objective

Nous avons également utilisé l'outil PEAQ (Perceptual Evaluation of Audio Quality) pour mesurer les performances de la structure de codage proposée. Cet outil développé par l'ITU-R dans la recommandation BS-1387 [14] permet de fournir une note appelée ODG (Objective Difference Grade), représentative de la qualité audio du signal testé. Le résultat de l'ODG donne une note comprise entre 0 et -4, où 0 correspond à une dégradation imperceptible et -4 à une dégradation très gênante. La note est négative car le signal testé est considéré comme moins bon que le signal de référence.
Les notes ODG des échantillons encodés avec le codeur SSC-ACELP sont meilleures en moyenne que la référence MPEG. Nous pouvons aussi noter que pour la majorité des échantillons, le SSC-ACELP est meilleur que le codeur MPEG4-SSC. La figure 8 montre les résultats détaillés pour les deux codeurs.

Item	SSC-ACELP	MPEG4-SSC
es01	-2.359	-3.491
es02	-2.637	-3.418
es03	-2.636	-3.514
si01	-2.344	-2.827
si02	-3.609	-3.818
si03	-1.080	-1.960
sm01	-1.819	-2.362
sm02	-3.098	-2.444
sm03	-1.904	-3.362
sc01	-3.297	-3.261
sc02	-2.848	-3.549
sc03	-1.557	-3.388
Moyenne	-2.432	-3.116

Figure 8 – *Résultats ODG*

5 Conclusion

Dans cet article, nous avons introduit le codeur audio scalable MPEG-4 SSC-ACELP. Nous avons présenté l'intérêt de combiner le codage audio paramétrique avec des modules de codage ACELP. Cette nouvelle architecture de codage permet de mieux représenter le signal résiduel et offre une grande flexibilité (scalabilité) en termes de débit. Des tests subjectifs à 24 kbps ont montré que ce nouveau codeur permet d'offrir une meilleure qualité audio qu'un codeur audio paramétrique « état de l'art ». De nouvelles évaluations seront menées dans le but de caractériser les performances du codeur scalable à différents points de fonctionnement. Il sera ainsi intéressant de confirmer de manière formelle l'amélioration continue de la qualité constatée de manière informelle.

Références

[1] Karlheinz Brandenburg. "MP3 and AAC Explained", Présenté à la 17ème Conférence International AES, Florence, Italie, Septembre 1999.

[2] Martin Wolters, Kristofer Kjörling, Daniel Homm, Heiko Purnhagen, "A closer look into MPEG-4 High Efficiency AAC", 115ème Convention AES, New York, USA, Octobre 2003.

[3] R.J. McAulay et T.F. Quatieri, "Speech analysis & synthesis based on a sinusoidal representation", IEEE Trans. on ASSP, Vol. 34, No. 4, Août 1986.

[4] B. Edler, H. Purnhagen, et C. Ferekidis, "ASAC– Analysis/synthesis codec for very low bit rates", Preprint 4179 (F-6) 100th AES Convention, Copenhagen, 11-14 Mai 1996.

[5] S. Levine, Audio Representations for Data Compression and Compressed Domain Processing, PhD thesis, Stanford University, Août 1998.

[6] H. Purnhagen, N. Meine, "HILN - The MPEG-4 Parametric Audio Coding Tools", IEEE International Symposium on Circuits and Systems (ISCAS 2000), Genève, Suisse, Mai 2000.

[7] E. Schuijers, W. Oomen, B. den Brinker and J. Breebart, "Advances in Parametric Coding for High-Quality Audio", 114th AES Convention, Amsterdam, Mars 2003.

[8] F. Itakura, "Line spectral representation of linear predictive coefficients of speech signals", J. Acoust. Soc. Amer., vol. 57, Supplément no. 1, S35, 1975.

[9] B.den Brinker et F. Riera-Palou, "Pure Linear Prediction", 115th AES Convention, New York, Octobre 2003.

[10] http://www.chiariglione.org/mpeg/working_documents/mpeg-04/audio/param-audio-VT.zip

[11] ISO/IEC JTC1/SC29/WG11/N6428,"ISO/IEC13818-7:2004 (AAC 3rd edition)", Mars 2004, Munich,

[12] 3GPP TS 26.190, "Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; Transcoding functions", 2004.

[13] M. Ghenania, C.Lamblin, "Low-cost Smart Transcoding Algorithm between ITU-T G.729 (8 kbit/s) and 3GPP NarrowBand AMR", Eusipco 2004.

[14] ITU Radiocommunication Study Group 6, "Recommandation ITU-R BS.1387 – Method for objective measurements of perceived audio quality"

Compression embarquée d'images satellites : Vers l'exploitation de la géométrie

X. Delaunay[1] C. Thiebaut[2] V. Charvillat[3]

[1] TéSA/CNES/NOVELTIS*, 14-16 port St Etienne, 31000 Toulouse
[2] CNES, 18 avenue Edouard Belin, 31401 Toulouse Cedex 9
[3] IRIT/ENSEEIHT, 2 rue Camichel, 31071 Toulouse cedex 7

xavier.delaunay@tesa.prd.fr, carole.thiebaut@cnes.fr, Vincent.Charvillat@enseeiht.fr

Concours Jeune Chercheur : Oui

Résumé

La résolution des images acquises à bord des satellites d'observation de la terre est de plus en plus grande et la compression à bord doit donc être de plus en plus performante pour transmettre les données au sol. L'augmentation des capacités de calcul et de mémoire permet la mise en place d'algorithmes de plus en plus complexes. Actuellement, on envisage des compresseurs capables de compenser les faiblesses de la transformée en ondelettes séparable, et qui utiliseraient une autre transformée et/ou des codeurs plus performants. Dans cette communication, nous mettons en évidence deux sortes de corrélations résiduelles entre coefficients d'ondelettes qu'il serait souhaitable d'éliminer pour améliorer la compression. Les premières corrélations sont situées dans un voisinage local. Les secondes sont liées aux structures géométriques de l'image et sont observées sur de plus grandes distances.

Mots clefs

Satellites, ondelettes, corrélations, structures, contours.

1 Introduction

Les applications issues de l'imagerie satellite sont de plus en plus diverses. Certaines nécessitent des images muti-, voire hyper-spectrales [1, 2, 3], d'autres deux prises de vue (stéréovision) [4, 5]. Ces nouveaux besoins contribuent fortement à l'augmentation du volume des données à transmettre au sol. Dans ces situations, des techniques de compression spécifiques ont été, et sont en cours de développement.

Dans le cas plus classique des images panchromatiques, l'évolution de la résolution des capteurs constitue aussi un facteur d'augmentation du volume des données. L'utilisation à bord de systèmes de compression de plus en plus efficaces est nécessaire pour transmettre les données collectées vers le sol.

La transformée en ondelettes a permis une nette amélioration des performances des compresseurs embarqués par rapport à la DCT (Discrete Cosine Transform) [6]. Cependant, les performances des compresseurs à base de transformée en ondelettes ne suffiront pas pour les futures missions THR (Très Haute Résolution). Aussi, l'exploitation de la géométrie des images lors de la compression est envisagée.

La compression des images à bord des satellites d'observation de la terre pose des problèmes que l'on ne rencontre pas au sol. Les spécificités de la compression satellite embarquée seront présentées dans une première partie. Ensuite, nous montrerons que la transformée en ondelettes séparable ne décorrèle pas complètement les images en mettant en évidence des redondances résiduelles dans les images transformées. Nous étudierons, d'abord ces redondances dans un voisinage restreint, puis nous considèrerons des régions plus étendues qui suivent les géométries locales des images.

2 La compression embarquée

2.1 Besoins et contraintes

Les satellites d'observation de la terre sont utilisés pour diverses applications : agriculture, urbanisme, cartographie. Les images acquises sont donc diverses et la qualité minimale requise est un PSNR (Peak Signal-to-Noise Ratio) de 40dB. L'augmentation de la résolution des images entraîne une augmentation des débits en entrée des compresseurs. Ceci est illustré dans le Tableau 1 dans les cas de trois missions récentes d'observation de la terre : SPOT4, SPOT5 et PLEIADES. Les besoins en compression sont donc de plus en plus importants avant la transmission des données au sol.

	SPOT4 (1998)	SPOT5 (2002)	PLEIADES (2008)
Fauchée	60km	60km	20km
Résolution	10m	2,5m	0,7m
Débit	32Mb/s	128Mb/s	4,5Gb/s

Tableau 1 – *Résolutions et débits des satellites d'observation de la terre en imagerie panchromatique*

*Ces travaux ont été menés grâce au soutient financier de NOVELTIS et du CNES.

Il existe trois contraintes majeures en compression embarquée. La première est liée au mode d'acquisition des images d'observation de la terre. Les capteurs balayant la surface de la planète produisent une image de largeur fixe mais de longueur virtuellement infinie. Cette image doit donc être compressée et transmise au fur et à mesure de son acquisition. La technique employée consiste à traiter l'image au fil de l'eau, c'est-à-dire, bloc de lignes par bloc de lignes. Sur PLEIADES, les blocs de lignes sont de 16 lignes.
La seconde contrainte lors de la compression à bord est liée à la capacité du canal de transmission des données vers le sol. Cette capacité est limitée et le débit des données transmises est fixe. En conséquence, le débit en sortie du compresseur doit lui aussi être fixe. Cependant, le taux de compression dépend des statistiques de l'image. Il peu être élevé sur des zones uniformes et plus faible ailleurs. C'est pourquoi, sur SPOT5, un algorithme de régulation de débit couplé à une mémoire tampon est utilisé [6]. L'utilisation de la transformée en ondelettes et d'un codeur par plan de bits produisant un train binaire emboîté permet aussi d'obtenir un débit constant. En effet, dans ce cas, la transmission est progressive et la régulation du débit est effectuée au bit près par simple troncature du train binaire emboîté. L'utilisation d'un tel codeur est recommandée par le CCSDS (Consultative Committee for Space Data Systems) [7].
La troisième contrainte est liée aux circuits électroniques. En effet, pour assurer un traitement en temps réel, les algorithmes sont implantés matériellement. Les capacités de calcul dépendent donc du nombre de portes disponibles sur les circuits. C'est pourquoi, l'algorithme de compression doit être peu complexe. Néanmoins, avec l'évolution des technologies (figure 1), cette contrainte est relaxée. En 2004, le nombre de portes par centimètre carré de circuit était de 2 millions, ce qui permettait une complexité calculatoire de plus de 200 opérations par pixels.

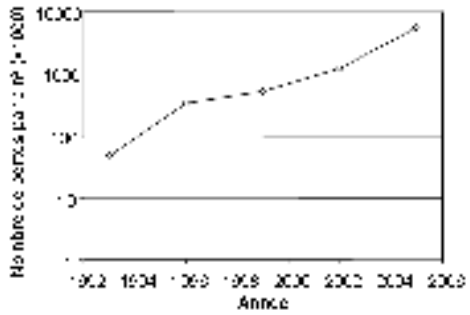


Figure 1 – *Évolution du nombre de portes des circuits électroniques*

2.2 Les méthodes actuelles

Sur SPOT5, lancé en mai 2002, le compresseur utilise une transformée en cosinus discrets (DCT) [6]. Grâce à un algorithme de régulation de débit, le taux de compression est maintenu constant et égal à 2,8.

Sur PLEIADES, qui sera lancé fin 2008, le compresseur utilise une transformée en ondelettes. Ceci permet d'augmenter le taux de compression à 6 pour une qualité d'image identique. De plus, les images décompressées ne présentent plus d'effet de bloc mais un léger flou. L'algorithme de transformée en ondelettes utilise le « lifting scheme » (schéma de lissage) [8] qui présente une complexité calculatoire plus faible que le schéma de convolution et qui ne nécessite pas de mémoire auxiliaire. Ce schéma « lifting » permet donc d'économiser deux ressources critiques à bord. Comme dans la recommandation du CCSDS [7], le codage est effectué par plan de bits. La régulation de débit est donc implicitement effectuée par troncature du train binaire emboîté généré par le codeur.

2.3 Avenir et enjeux

Figure 2 – *Image d'un capteur aéroporté à la résolution 20 cm*

Les diverses applications liées à l'observation de la terre, telles que la défense, la sécurité civile, et la prévention des catastrophes naturelles, demandent des résolutions toujours plus fines. De plus, dans les futures missions d'observation de la terre, des systèmes de détection à bord et de régions d'intérêt (ROI) pourraient être mis en place. La résolution des images sera supérieure à celle de PLEIADES (70 cm). La figure 2 illustre une image à grande résolution. Enfin, pour un PSNR de 50dB, le taux de compression devra être supérieur à 6. Les débits d'acquisition seront donc plus importants et l'utilisation de la transformée en ondelettes ne suffira plus pour compresser les données au débit imposé par le canal de transmission.
Pour remédier à ce problème, deux pistes sont envisagées : dans le paragraphe 3, on identifie des résidus de corrélations dans les images transformées en ondelettes qui pourraient être exploités par des codeurs contextuels afin de mieux compresser les images. Dans le paragraphe 4, l'exploitation de la géométrie des images est envisagée, soit en utilisant une autre transformée, soit en travaillant directement dans les images transformées en ondelettes.

3 Redondances résiduelles entre coefficients voisins dans la transformée en ondelettes

3.1 Redondances résiduelles dans la transformée en ondelettes

Bien que la transformée en ondelettes soit utilisée pour décorréler les images, il est encore possible d'observer des structures dans les images transformées. La figure 3, représente la transformée en ondelettes de l'image de la figure 2 obtenue avec les filtres 9/7 CDF (Cohen-Daubechies-Feauveau [9]). Ce sont les filtres les plus utilisés en compression. Ils sont notamment présents dans le standard JPEG2000 [10] et la recommandation du CCSDS [7]. Sur la figure 3, l'échelle de représentation des coefficients d'ondelettes a été ajustée de sorte que toute la gamme de gris soit utilisée dans chaque sous-bande. Ceci permet de faire apparaître les structures de l'image.

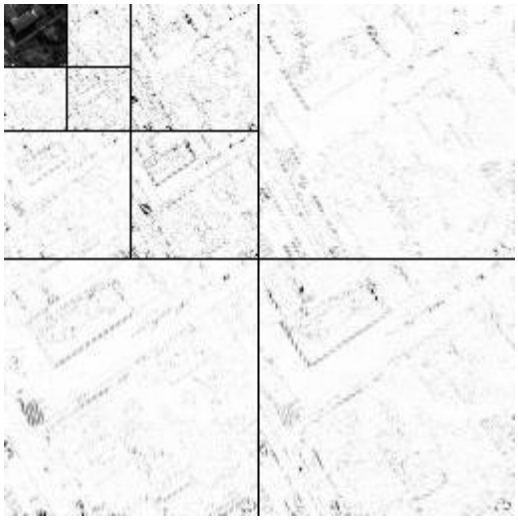


Figure 3 – *Transformée en ondelettes de la figure 2*

Des dépendances statistiques ont aussi été observées entre coefficients d'ondelettes dans un voisinage très local. Liu et Moulin [11] ont étudié les dépendances inter et intra échelles en mesurant l'information mutuelle entre deux ou plusieurs coefficients d'ondelettes. Simoncelli a modélisé les probabilités d'apparition conjointes entre coefficients d'ondelettes voisins [12], et avec Buccigrossi, développé un codeur basé sur cette modélisation [13].

3.2 Identification quantitative de redondances entre coefficients d'ondelettes voisins

Dans ce paragraphe, on s'intéresse aussi aux dépendances statistiques entre coefficients d'ondelettes dans un voisinage très local mais, plutôt que d'étudier l'information mutuelle ou les probabilités d'apparitions conjointes, nous étudions les coefficients de corrélation entre coefficients d'ondelettes non quantifiés. Une mesure d'information mutuelle sera aussi effectuée afin de valider les résultats obtenus. Pour un coefficient d'ondelettes C considéré, on définit un ensemble de sept voisins dans la transformée en ondelettes [13]. Ce voisinage est explicité à la figure 4.

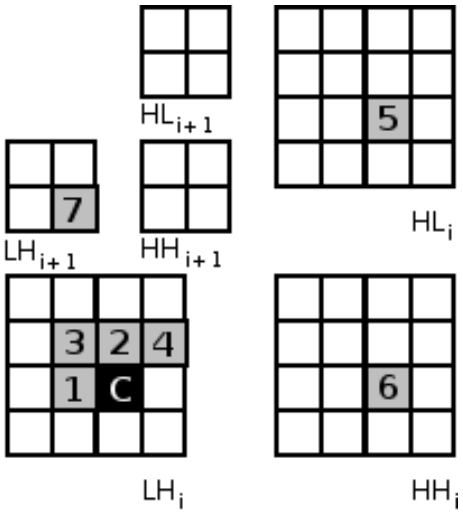


Figure 4 – *Définition d'un voisinage dans le domaine ondelettes. Les voisins numérotés de 1 à 4 sont situés dans la même sous-bande que le coefficient C. Les voisins numérotés 5 et 6 sont les cousins du coefficient C, et le voisin numéroté 7 est son parent. Ce denier n'est pas défini pour un coefficient dans une des sous-bandes à la résolution 3.*

La figure 5 présente les coefficients de corrélation entre le coefficient C et chacun de ses sept voisins sur la transformée en ondelettes illustrée à la figure 3. Ces coefficients de corrélation ont été obtenus en considérant successivement l'ensemble des coefficients de chaque sous-bande comme les réalisations d'une variable aléatoire. On a donc fait abusivement l'hypothèse d'ergodicité et de stationnarité des coefficients d'ondelettes [12]. Néanmoins, on constate qu'il existe des coefficients de corrélation de plus de 0,4 en valeur absolue entre coefficients voisins de la même sous-bande (voisins numérotés 1 à 4). Les coefficients de corrélation inter sous-bandes et inter échelles sont eux toujours inférieurs à 0,1 en valeur absolue. De plus, la corrélation entre les coefficients d'ondelettes diminue lorsque l'échelle de la transformée augmente. Enfin, le signe des coefficients de corrélation est à mettre en relation avec les filtres d'ondelettes appliqués à l'image.
Ces corrélations résiduelles dans la transformée en ondelettes ont été observées sur d'autres images satellites avec les mêmes ordres de grandeurs mais avec des variations selon la géométrie prédominante. Par contre, sur les transformées en ondelettes d'images de bruit gaussien et impulsionnel, les coefficients de corrélation observés sont inférieurs à 0,01 en valeur absolue, ce qui suggère que ces corrélations entre coefficients d'ondelettes voisins sont liées à la structure des images. Sur des images synthétiques contenant des contours mais sans textures, on observe le même

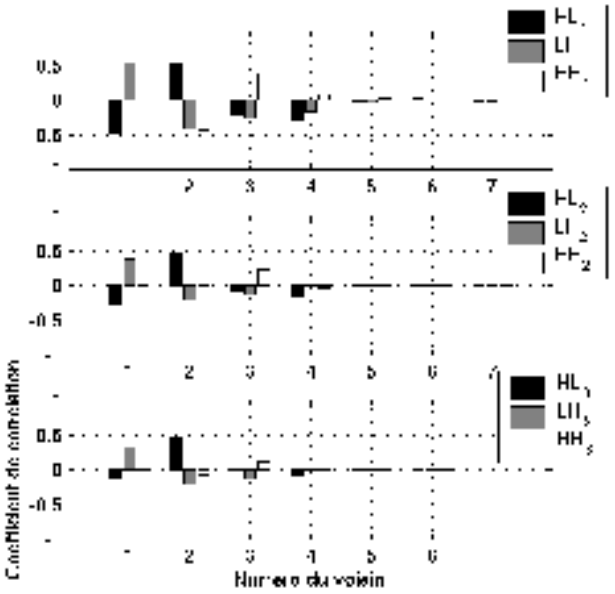


Figure 5 – *Coefficients de corrélation entre coefficients d'ondelettes voisins dans les diverses sous-bandes*

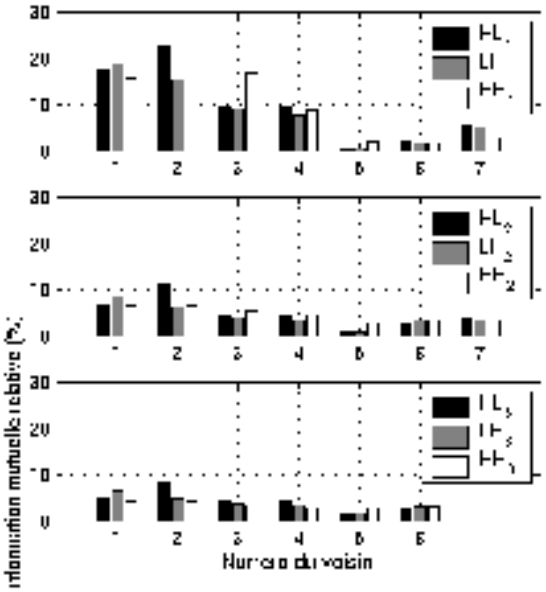


Figure 6 – *Information mutuelle relative entre coefficients d'ondelettes voisins dans les diverses sous-bandes*

type de corrélation que sur les images naturelles mais dans ce cas, les corrélations augmentent avec l'échelle de la transformée. Sur des images de textures sans aucun contour telles que des images satellites de mer ou de champ, on a pu observer, dans les sous-bandes à la résolution 1, des coefficients de corrélation atteignant 0,5 pour les voisins numérotés 1 et 2. Cependant, dans ce cas, les corrélations diminuent très rapidement avec l'échelle de la transformée. Elles sont toutes inférieures à 0,2 à la résolution 2. On peut donc faire l'hypothèse que ces corrélations sont issues des très petites structures qui constituent la texture et qui n'existent qu'à haute résolution. Seules les corrélations issues des grandes structures (les contours) peuvent être observées à faible résolution.

La figure 6 présente l'information mutuelle relative [14] entre coefficients d'ondelettes voisins. On préfère utiliser une mesure relative plutôt qu'une simple mesure d'information mutuelle pour pouvoir comparer des sous-bandes qui n'ont pas les mêmes entropies. L'information mutuelle relative $I_r(C,N)$ peut être interprétée comme la proportion d'information qu'il est possible d'économiser pour le codage de la variable aléatoire C si on connaît parfaitement la variable aléatoire N. Elle s'exprime en fonction des entropies d'ordre 0 de la manière suivante :

$$I_r(C,N) = \frac{2\left(H_0(C) - H_0(C|N)\right)}{H_0(C) + H_0(N)}$$

La figure 6 confirme qu'il est possible d'exploiter les relations entre coefficients d'ondelettes dans un voisinage très local. En effet, dans les sous-bandes à la résolution la plus fine, l'information mutuelle relative approche les 20% pour les voisins 1 et 2. Néanmoins, elle est inférieure à 10% dans les autres échelles et autour de 5% pour les voisinages inter sous-bandes (voisins 5, 6 et 7).

La plupart des dépendances statistiques sont donc observées à l'intérieur des sous-bandes. Afin de savoir si des corrélations existent à plus grande distance, la même étude a été effectuée sur un voisinage plus étendu à l'intérieur de chaque sous-bande. On a constaté qu'à toutes les échelles, et dans toutes les sous-bandes, les corrélations entre coefficients d'ondelettes diminuent avec la distance qui les sépare. A une distance de 3 pixels, toutes les corrélations sont inférieures à 0,1.

Cette étude statistique montre que des dépendances très locales liées à la structure des images sont observables à l'intérieur d'une même sous-bande. Ces dépendances existent sur la totalité des coefficients de chaque sous-bande de la transformée en ondelettes. Les coefficients de corrélation entre coefficients voisins spatialement sont de l'ordre de 0,2. Cependant, dès que la taille du voisinage augmente, les corrélations observées deviennent très faibles. Pourtant, des grandes structures sont observées dans les images transformées (figure 3). Elles proviennent de l'inefficacité de la transformée en ondelettes 2D séparable à représenter correctement les contours. En effet, cette transformée n'est capable d'exploiter les régularités géométriques que dans les directions des deux axes alors que des contours interviennent avec toutes les orientations. Ces régularités géométriques résiduelles doivent donc être éliminées afin d'améliorer la compression. Les transformées géométriques présentées dans le paragraphe suivant peuvent être une piste d'amélioration.

4 Redondances résiduelles liées à la géométrie

4.1 Les transformées géométriques

De nouvelles transformées, dérivées des ondelettes, ont récemment été développées dans le but d'améliorer la description des contours et des structures géométriques des images. Do et Vetterli [15] ont formulé une liste de cinq caractéristiques que devrait posséder une nouvelle transformée géométrique. Ces caractéristiques sont la multirésolution, la localisation spatiale et fréquentielle des éléments de base, l'absence de redondance, la multi-directionnalité, et l'anisotropie des éléments de base. La transformée en ondelettes 2D séparable possède les trois premières caractéristiques. Les deux dernières permettent une représentation efficace des structures géométriques. Les contourlets [15] et les curvelets [16] sont deux exemples de transformées géométriques qui sont proches conceptuellement. Elles constituent des familles de « frames » d'ondelettes géométriques conçues pour représenter les contours de façon parcimonieuse. Le problème majeur pour utiliser ces transformées en compression est qu'elles sont redondantes. La seconde génération de bandelettes développée par Peyré [17] ne suit pas le même raisonnement. La transformée en bandelettes est construite au-dessus d'une transformée en ondelettes 2D séparable et garde ainsi les trois bonnes caractéristiques de cette dernière. L'information résiduelle dans des blocs de 4×4 coefficients d'ondelettes est ensuite compactée en utilisant des bases de polynômes directionnelles adaptées à la géométrie. Cette transformée permet d'atteindre des taux de compression plus élevés (en terme de débit/PSNR) que la transformée en ondelettes seule. La comparaison des performances d'un compresseur basé sur les bandelettes avec l'algorithme de compression proposé par le CCSDS [7] est en cours. Le compresseur en bandelettes présente un avantage non négligeable en compression satellite : l'erreur maximale commise à la reconstruction est toujours très faible. Sur l'image de la figure 2 dont la dynamique est 4096, à 1 bpp l'erreur maximale n'est que de 164 lorsque les bandelettes sont utilisées. Elle est de 245 avec l'algorithme du CCSDS. En effet, les bandelettes permettent de mieux décrire les contours et les effets de flou introduits par les ondelettes sont diminués. Cependant, dans l'état, les bandelettes ne diminuent pas l'entropie d'ordre 0 et certaines corrélations on été observées entre les coefficients de bandelettes. Un codeur contextuel est donc nécessaire pour atteindre des taux de compression élevés.

4.2 Identification quantitative de redondances géométriques

Les bandelettes sont appliquées successivement sur des blocs de coefficients d'ondelettes de petite taille et ne peuvent donc pas exploiter l'ensemble des grandes structures visibles sur la figure 3. Dans ce paragraphe, nous allons mettre en évidence les corrélations qui peuvent exister

Figure 7 – *Détection d'un contour (en blanc)*

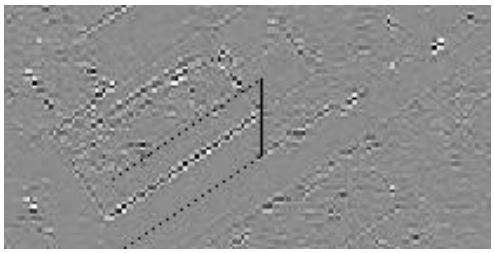


Figure 8 – *Sous-bande HL_1 de la transformée en ondelettes de la figure 7 et zone du contour étudié*

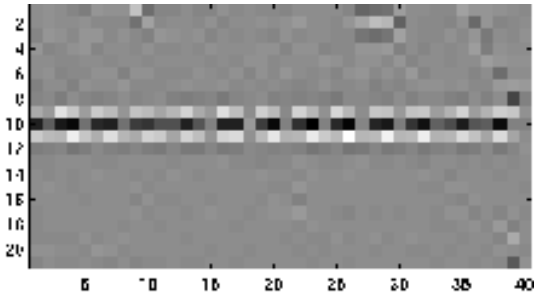


Figure 9 – *Lecture colonne par colonne de la zone du contour étudié, les dépendances géométriques sont mises en évidence*

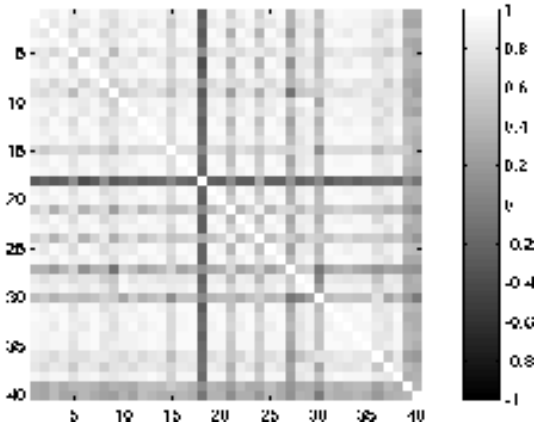


Figure 10 – *Matrice des coefficients de corrélation entre les colonnes de coefficients d'ondelettes de la zone étudiée*

le long de l'une de ces structures.
Sur la figure 7, un contour a été détecté en utilisant un filtrage de Sobel vertical. Les coefficients d'ondelettes de la figure 8 ont été réordonnés le long de ce contour et sont présentés à la figure 9. En considérant chaque colonne de cette figure comme une variable aléatoire et chaque ligne comme une réalisation, on peut tracer la matrice des coefficients de corrélation de toutes ces variables aléatoires 2 à 2 (figure 10). On constate des corrélations supérieures à 0,7 même à longue distance, par exemple entre la colonne 1 et la colonne 38.
Comme la résolution des capteurs embarqués à bord des satellites augmente, les structures observées sont de plus en plus longues. Les grandes régularités sont donc de plus en plus fréquentes et leur exploitation suscite beaucoup d'intérêt. L'objectif pour des codeurs en ondelettes orientés vers l'exploitation de la géométrie serait donc de pouvoir tenir compte des structures de l'image afin de décorréler plus efficacement l'information.

5 Conclusions et perspectives

Nous avons montré que la transformée en ondelettes ne décorrèle pas totalement le contenu des images. Dans cette communication, deux types de corrélations résiduelles ont été mis en évidence : les corrélations entre coefficients voisins spatialement, et les corrélations le long des structures de l'image. Pour augmenter les taux de compression, des codeurs tels que EBCOT [18] exploitent déjà, en partie, les corrélations entre coefficients voisins. Les futurs compresseurs embarqués devront aussi savoir exploiter les grandes régularités géométriques qui existent dans les zones urbaines ou agricoles pour atteindre des débits inférieurs à ceux des compresseurs actuels. L'exploitation de la géométrie peut être menée sur deux fronts : en utilisant de nouvelles transformées dérivées des ondelettes et/ou des codeurs s'adaptant aux dépendances statistiques et aux régularités géométriques. L'augmentation de la longueur des structures avec la résolution et les premières études réalisées montrent que ce sont des pistes tangibles. De plus, cette voie pourrait s'ouvrir sur la détection à bord de zones d'intérêt et sur l'analyse sémantique des images satellites.

Références

[1] X. Tang, W. A. Pearlman, et J. W. Modestino. Hyperspectral image compression using three-dimensional wavelet coding. *SPIE Image and Video Communications and Processing*, 5022 :1037–1047, Janvier 2003.

[2] E. Christophe, D. Léger, et C. Mailhes. Quality criteria benchmark for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9) :2103–2114, Septembre 2005.

[3] C. Thiebaut, D. Lebedeff, C. Latry, et Y. Bobichon. On-board compression algorithm for satellite multispectral images. Dans *Data Compression Conference (DCC'06)*, page 467, Snowbird, Mars 2006.

[4] William L. Barnes, éditeur. *SPOT5 THR mode*, San Diego, Octobre 1998. Earth Observing Systems III, SPIE.

[5] A. Baudoin, M. Schroeder, C. Valorge, M. Bernard, et V. Rudowski. The HRS-SAP initiative : A scientific assessment of the high resolution stereoscopic instrument on board of SPOT 5 by ISPRS investigators. *IAPRS*, 35(B1) :372–378, Juillet 2004.

[6] William L. Barnes, éditeur. *Selection of the SPOT-5 Image Compression algorithm*, volume 3439-70, San Diego, Octobre 1998. Earth Observing Systems III, SPIE.

[7] CCSDS. Image data compression. recommended standard. *CCSDS 122.0-B-1, Blue Book*, Novembre 2005.

[8] W. Sweldens. The lifting scheme : a construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2) :511–546, Mars 1998.

[9] A. Cohen, I. Daubechies, et J. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45 :485–560, 1992.

[10] JPEG committee. JPEG 2000 Part I Final committee draft version 1.0. (ISO/IEC FCD15444-1 : 2000), Mars 2000.

[11] J. Liu et P. Moulin. Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients. *IEEE Transactions on Image Processing*, 10(11) :1647 – 1658, Novembre 2001.

[12] E.P. Simoncelli. Modeling the joint statistics of images in the wavelet domain. Dans *Proc SPIE, 44th Annual Meeting*, volume 3813, pages 188–195, Denver, CO, Juillet 1999.

[13] R.W. Buccigrossi et E.P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing*, 8(12) :1688–1701, Décembre 1999.

[14] J. Malo, I. Epifanio, R. Navarro, et E.P. Simoncelli. Non-linear image representation for efficient perceptual coding. *IEEE Transactions on Image Processing*, 15(1) :68–80, Janvier 2006.

[15] M.N. Do et M. Vetterli. The contourlet transform : an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12) :2091–2106, Décembre 2005.

[16] E. Candès, L. Demanet, D. Donoho, et L. Ying. Fast discrete curvelet transforms. *Tech. Report, California Institute of Technology*, 2005.

[17] G. Peyré. *Géométrie multi-échelles pour les images et les textures*. Thèse de doctorat, Ecole Polytechnique, Décembre 2005.

[18] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7) :1158–1170, Juillet 2000.

Amélioration de codeurs DCT par orientation des blocs de la transformée

ROBERT Antoine[1] AMONOU Isabelle[1] PESQUET-POPESCU Béatrice[2]

[1] France Telecom R&D
4, rue du Clos Courtel
35512 Cesson-Sévigné Cedex
{a.robert, isabelle.amonou}@francetelecom.com

[2] TSI - ENST Paris
46, rue Barrault
75634 Paris Cedex 13
pesquet@tsi.enst.fr

Résumé

Cet article décrit un pré-traitement pour des codeurs de type DCT, et plus généralement pour des codeurs d'images ou de vidéo basés blocs utilisant avantageusement l'orientation de ces blocs. Contrairement à la plupart des solutions proposées jusqu'alors, ce n'est pas ici la transformée qui s'adapte au signal, mais le signal qui est traité pour s'adapter à la transformée. Les blocs sont orientés grâce à des permutations circulaires appliquées au niveau pixel. Avant de réaliser ces permutations, l'orientation de chaque bloc est évaluée à l'aide d'une sélection basée sur un critère débit-distorsion. Ce pré-traitement introduit dans un codeur AVC [1] et appliqué aux images résiduelles intra permet d'en améliorer les performances.

Mots clefs

Transformation directionnelle, orientation, permutations circulaires, codeurs DCT, H.264-MPEG4/AVC.

1 Introduction

La compression d'images et de séquences vidéo est motivée par la recherche de représentations compactes pour les données en utilisant des transformées. Les premières transformées introduites étaient séparables et simples comme la DCT et les ondelettes de première génération. Par simple, on entend que ces transformations ne sont pas optimales pour représenter les données images de manière compacte et qu'elles sont souvent redondantes, mais elles sont toutefois généralement rapides et peu complexes. Cette sous-optimalité est partiellement due au fait que ces transformées ne sont pas adaptées aux données possédant des discontinuités situées le long de courbes régulières. Afin de profiter des structures géométriques des images ou des séquences vidéo, plusieurs auteurs ont proposé de nouvelles transformations telles que les curvelets, les contourlets, les bandelettes ou les directionlets, et d'autres ont cherché à améliorer les transformations existantes en tenant compte de ces structures géométriques.

Les curvelets, introduites par Candès et Donoho [2], permettent grâce à leur grand degré de directionnalité d'obtenir une approximation optimale des images lisses possédant des contours C^2. Cette transformée nécessite une rotation et correspond à un partitionnement 2D des fréquences basé sur les coordonnées polaires, ce qui est équivalent à un banc de filtres directionnels. Elles ont été définies initialement pour le cas continu et ne sont pas trivialement transférables au cas discret. Pour outrepasser ce problème, Do et Vetterli ont proposé les contourlets [3] qui ont les mêmes caractéristiques géométriques que les curvelets, mais définies directement dans le cas discret. Cette transformation effectue une analyse directionnelle d'un signal 2D en utilisant un banc pyramidal de filtres directionnels. Pour cela, le signal subit d'abord une décomposition Laplacienne redondante et pyramidale avant que chacune des sous-bandes obtenues ne soit traitée par le banc de filtres directionnels. Ces méthodes d'analyse impliquent que les curvelets et les contourlets sont redondantes, de plus ces transformées ne sont pas efficaces à haut débit.

Mallat a proposé les bandelettes [4] et plus récemment les bandelettes de seconde génération [5] qui sont toutes deux des transformées adaptatives. Dans le cas de seconde génération, une transformée géométrique orthogonale est appliquée aux coefficients en ondelettes, c'est à dire qu'après avoir décomposé les données grâce à un banc de filtres d'ondelettes orthogonales, les coefficients sont traités à l'aide de filtres directionnels orthogonaux. Chacune des directions géométriques correspond à une transformée différente (un filtre directionnel différent). Ceci nécessite alors une détection des contours et une déformation des données afin de pouvoir appliquer la bonne transformée sur le bon treillis.

Plus récemment, Velisavljević et Vetterli ont introduit les directionlets [6]. Elles travaillent à échantillonnage critique en appliquant un filtrage séparable non seulement à l'horizontale et à la verticale mais aussi suivant des sous-ensembles de "co-lignes" numériques. Ces co-lignes numériques représentent toutes les directions définies sur un treillis entier. Le filtrage le long de ces co-lignes est réalisé en effectuant une rotation définie par la pente de ces co-lignes avant d'appliquer un filtrage horizontal. Cependant, ces directionlets ne sont pas des transformées basées blocs.

D'autres méthodes utilisent des rotations complètes de l'image ou d'une partie de l'image, mais elles nécessitent généralement une interpolation. Unser et al. [7] ont conçu des algorithmes rapides de rotation d'images permettant de conserver la qualité initiale de cette image. Ces rotations

sont décomposées en trois translations s'appuyant sur une interpolation basée sur une convolution. Le principal désavantage de ces méthodes rotationnelles est que les informations contenues dans les coins des blocs ou de l'image sont généralement perdues.

Peu de méthodes présentées précédemment utilisent des transformées basées blocs qui sont pourtant les plus courantes dans les standards actuels (e.g. MPEGx, H.26x, JPEG, ...). Ceci vient du fait que les blocs introduisent de nouvelles discontinuités à cause de leurs bords. Notre but est donc de construire une rotation basée bloc qui conserve la forme (rectangulaire) de ce bloc afin de pouvoir y appliquer une transformée basée bloc.

Cet article est organisé comme suit : dans la Section 2 nous introduisons notre pré-traitement et décrivons comment il s'applique aux blocs. Dans la Section 3, nous présentons la méthode de sélection d'orientation avant de décrire le codage de ces informations dans la Section 4. Enfin, quelques résultats numériques de notre pré-traitement appliqué aux images résiduelles intra de AVC sont présentés en Section 5 avant de conclure et de donner quelques perspectives dans la Section 6.

2 Orientation des blocs de la transformée

Toutes les transformées présentées précédemment essaient de s'adapter au signal, mais chacune d'entre elles nécessite pour cela des opérations non entières en contradiction avec la reconstruction parfaite. Les contourlets et les curvelets sont redondantes, les bandelettes nécessitent une détection de contours et une déformation des données, et les rotations une interpolation. La méthode que nous proposons ici est un pré-traitement des images ou des séquences vidéo qui tient compte des structures géométriques des données sans déformation ni interpolation du signal.

La plus connue des transformées basées bloc est la DCT (Discrete Cosine Transform) qui est utilisée dans la plupart des standards images et vidéo tels que JPEG [8], MPEGx, H.26x comme H.264-MPEG4/AVC [1]. Cette transformée s'applique à des blocs de coefficients qui peuvent avoir une taille variable : soit 8x8 dans le cas de la DCT flottante de JPEG, soit 4x4 et 8x8 pour les DCT entières de AVC. De plus, les coefficients de ces blocs peuvent être de différentes natures : soit directement issus des images (données brutes) comme dans le cas JPEG, soit issus de prédiction comme dans le cas AVC. Dans tous ces cas, les blocs de coefficients à traiter présentent des motifs réguliers orientés comme représenté sur la Figure 1.

Figure 1 – *Extrait de l'image Flower (CIF) et son résidu de prédiction intra AVC*

Les images intra de AVC subissent une prédiction spatiale : chaque bloc 4x4 ou 8x8 est prédit à partir de ces voisins et dans une des 9 directions possibles [1], le bloc résiduel à coder est alors la différence entre le bloc original et sa prédiction. Pour les macroblocs 16x16, le même procédé est appliqué, mais il n'existe que 4 directions [1].

Notre pré-traitement cherche à exploiter l'orientation de ces blocs sans utiliser de rotation qui nécessite une interpolation, mais en effectuant des permutations circulaires au niveau pixel. Il effectue un cisaillement ("shear") simulant ainsi une pseudo-rotation des blocs. Ceci permet alors de redresser les blocs vers l'horizontale ou la verticale.

2.1 Blocs 4x4

Dans le cas des blocs 4x4, nous avons défini cinq états différents qui correspondent à sept orientations prédéfinies du bloc et qui sont associés à des permutations.

A l'état 0, aucune opération n'est à effectuer parce que soit les blocs sont non-orientés (si leur direction est horizontale ou verticale), soit ils n'ont pas de directions acceptables : ils sont orientés suivant un angle trop éloigné (plus de $\pm 3^o$) des angles prédéfinis pour pouvoir correspondre à une des permutations.

Les autres états spécifient les blocs dont la direction est proche (moins de $\pm 3^o$) des angles définis par le Tableau 1.

État	Angle	État	Angle
1	$+27^o$	2	-27^o
3	$+45^o$	4	-45^o

Tableau 1 – *Les états du cas 4x4*

Pour chacun de ces états, une permutation circulaire est appliquée au niveau pixel afin de simuler une rotation par cisaillement. Ces permutations circulaires nous permettent de nous affranchir d'une étape d'interpolation inhérente à tout processus réel de rotation (matricielle). De plus, par ces simples réarrangements de pixels nous simulons une rotation sans créer de trous dans les coins des blocs.

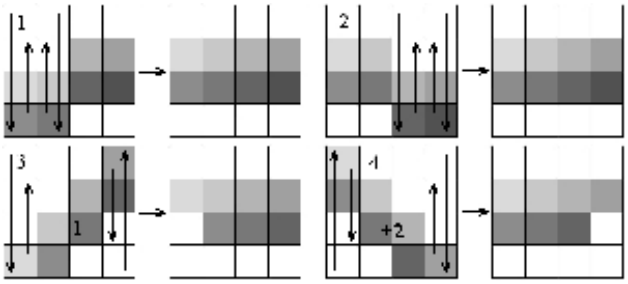


Figure 2 – *Permutations circulaires du cas 4x4*

Dans l'état 1 (cf Figure 2, en haut à gauche), une permutation circulaire est réalisée sur les deux premières colonnes, et dans l'état 2, son opposé, sur les deux dernières colonnes. Les états 3 et 4 utilisent des réarrangements de pixels plus complexes : l'état 3 correspond à des permutations circulaires appliquées sur la première et la dernière colonne, puis à la même permutation circulaire que celle de l'état 1. L'état 4 est similaire à l'état 3 mais les opérations sont réalisées sur les lignes : la première et la dernière ligne sont réarrangées avant de subir la permutation de l'état 2 (cf Figure 2, au milieu).

Cette figure montre bien que les blocs sont redressés vers l'horizontale ou la verticale après notre pré-traitement. Ces permutations circulaires permettent donc de simuler une rotation réelle sans ses désavantages.

2.2 Blocs 8x8

Comme dans le cas 4x4, nous définissons ici 9 états d'orientation pour les blocs 8x8 [9].

Comme précédemment, l'état 0 reflète les blocs non-orientés et les blocs dont l'orientation est trop éloignée (plus de $\pm 3^o$) des angles de rotation prédéfinis.

Les autres états correspondent aux blocs dont la direction est proche des angles : $\pm 14^o$, $\pm 27^o$, $\pm 37^o$ et $\pm 45^o$.

Chacune de ces orientations est associée à un réarrangement des pixels appliqué aux blocs par des permutations circulaires comme dans le cas 4x4 (cf Figure 3).

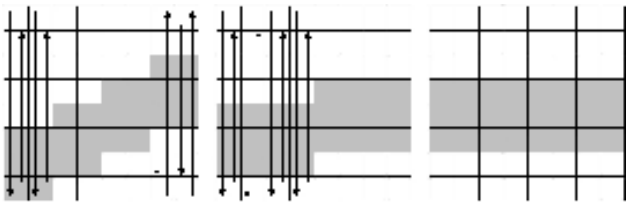


Figure 3 – *Exemple de permutations circulaires du cas 8x8 : l'état 3 (angle de $+27^o$)*

2.3 Blocs 16x16

Comme dans les cas 4x4 et 8x8, on définit ici 17 états d'orientation pour les blocs 16x16 ou macroblocs.

L'état 0 correspond toujours aux blocs non-orientés et aux blocs dont la direction est trop éloignée (plus de $\pm 3^o$) des angles prédéfinies.

Les autres états correspondent aux blocs dont la direction est proche des angles : $\pm 7^o$, $\pm 14^o$, $\pm 20^o$, $\pm 27^o$, $\pm 32^o$, $\pm 37^o$, $\pm 41^o$ et $\pm 45^o$.

Chacune de ces directions est associée à un réarrangement des pixels appliqué aux blocs 16x16 grâce à des permutations circulaires comparables à celles des cas 4x4 et 8x8 (cf Figure 4).

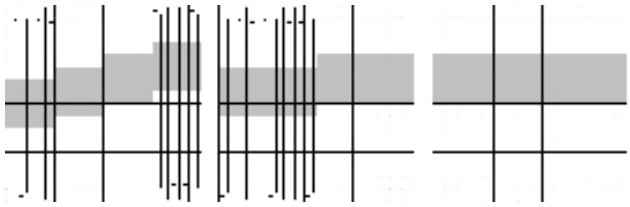


Figure 4 – *Exemple de permutations circulaires du cas 16x16 : l'état 3 (angle de $+14^o$)*

3 Sélection de l'orientation

Avant d'appliquer notre pseudo-rotation, il faut sélectionner la bonne orientation pour chacun des blocs de l'image ou de la séquence vidéo.

Nous nous basons sur l'optimisation débit-distorsion (RDO) de AVC [10]. Cette optimisation consiste à utiliser toutes les combinaisons de modes disponibles et à coder les macroblocs avec celle qui donne les meilleures performances : la plus faible distorsion pour un débit donné ou le meilleur débit pour une distorsion donnée.

Le coût de codage d'un macrobloc dépend donc de deux variables : le débit et la distorsion. Le débit est, dans tous les cas, la somme des débits des blocs le composant (16×(4x4), 4×(8x8), ou 1×(16x16)). Et la distorsion est toujours la distorsion globale du macrobloc quels que soient les blocs le composant, elle est donnée par l'erreur quadratique du macrobloc reconstruit :

$$D = \sum_{m=0}^{15} \sum_{n=0}^{15} \left(i_{MB}(m,n) - \hat{i}_{MB}(m,n) \right)^2 \qquad (1)$$

où $i_{MB}(m,n)$ est le pixel (m,n) du macrobloc original et $\hat{i}_{MB}(m,n)$ celui du macrobloc reconstruit.

Pour chaque macrobloc, toutes les tailles de blocs disponibles sont testées avec tous les modes de prédiction disponibles. Par rapport à un codeur classique, notre prétraitement ne fait qu'ajouter des combinaisons à tester. Par exemple, dans un codeur vidéo AVC et pour une image intra, cela revient à :

– dans le cas 16x16, au lieu de tester une seule fois chacun des quatres modes de prédiction [1], nous les testons 17 fois avec nos orientations 16x16 (cf 2.3).

– dans le cas 8x8 et pour chaque bloc 8x8, nous testons 9 fois chacun des 9 modes de prédiction [1] avec nos orientations 8x8 (cf 2.2).
– dans le cas 4x4, pour chaque bloc 4x4 du macrobloc, nous testons les 9 modes de prédiction [1] 5 fois avec nos cinq orientations candidates (cf 2.1).

Donc pour le codage d'un macrobloc, tous les cas sont testés et la meilleure combinaison est conservée dans chacun des cas. Ces trois solutions sont ensuite comparées à l'aide de la même méthode d'optimisation débit-distorsion afin d'en isoler la meilleure combinaison, combinaison utilisée par la suite pour le codage réel du macrobloc.

Cette méthode de sélection des orientations est efficace, mais complexes en termes d'évaluations de débit-distorsion : par exemple, dans un codeur vidéo AVC et pour une image intra, nous testons $17\times4+4\times(9\times9)+16\times(5\times9) = 1112$ modes par macrobloc (contre 184 pour AVC). Cette complexité, quasiment multipliée par 7, ne touche cependant que la sélection débit-distorsion des modes de prédiction intra qui ne représente qu'une partie du codage AVC. La complexité globale de l'algorithme AVC n'en est que très peu affectée.

Pour réduire la complexité de cette première méthode, nous nous attacherons à développer une autre méthode calculant la direction de chaque bloc afin d'appliquer directement la meilleure permutation.

4 Codage et décodage

4.1 Ensemble du macrobloc

Après avoir sélectionné le meilleur mode (le meilleur mode étant une combinaison de taille de blocs, de permutations et le cas échéant de modes de prédiction), le codage du macrobloc est effectué par le codeur image ou vidéo utilisé. Chaque macrobloc est transformé par DCT, quantifié et codé entropiquement, et ce quelque soit le codeur DCT utilisé. Par exemple, dans le cas de AVC, après la prédiction intra, les résidus (orientés ou non) sont transformés par la DCT entière 4x4 ou 8x8 de AVC, quantifiés et codés avec CABAC [11] : le Context-based Adaptive Binary Arithmetic Coding de AVC parvient à de bonne performance en compression grâce à (a) une sélection de modèles de probabilité pour chacun des éléments de syntaxe en fonction du contexte de cet élément, (b) une adaptation de l'estimation des probabilités basée sur des statistiques locales et (c) l'utilisation d'un codage arithmétique (il n'est utilisable qu'en profils Main et High (défini dans FRExt [9]) sinon CAVLC [1] [12] le remplace dans les autres profils). Les macroblocs orientés sont traités de la même manière que les non-orientés.

Avec notre pré-traitement, les blocs sont redressés vers l'horizontale ou la verticale avant la transformation, ceci implique que la transformée DCT est plus efficace sur ces données et donc améliore les performances générales en débit-distorsion.

Le décodage est aussi effectué par le codeur image ou vidéo utilisé. Les macroblocs sont décodés entropiquement (avec CABAC pour AVC), déquantifiés, et subissent une transformation DCT inverse (DCT entière 4x4 ou 8x8 pour AVC). Les macroblocs qui ont été orientés avant codage doivent être réorientés après le décodage en utilisant les permutations inverses de celles définies précédemment en section 2. Pour cela, nous devons transmettre les informations de permutations nécessaire à cette réorientation.

4.2 Information de permutation

L'information de permutation correspondant à l'état d'orientation du meilleur mode de codage est écrite, quelque soit la taille de bloc utilisée et même si elle correspond à l'état 0, dans l'entête du bloc. Dans le cas 16x16, cette information est écrite dans l'entête du macrobloc (pour le cas AVC, après le mode de prédiction intra). Dans les cas 8x8 et 4x4, elle est écrite dans l'entête de chacun des blocs (pour le cas AVC, entre le mode de prédiction intra pour les luminances et le mode de prédiction intra pour les chrominances).

Pour le 16x16, l'information de permutation est codée en utilisant CABAC et le même contexte que celui défini pour les modes de prédiction intra 16x16 dans AVC. Pour les autres cas, cette information est prédite en fonction de son voisinage (comme pour les modes de prédiction intra 4x4 et 8x8 de AVC) avant d'être codée avec CABAC et les contextes des modes de prédiction intra 4x4 et 8x8 respectivement. Les états d'orientation des blocs adjacents au dessus (A) et à gauche (B) (cf Figure 5) sont comparés s'ils sont disponibles sinon ils sont considérés nuls. Celui possédant la plus grande valeur définit l'état d'orientation le plus probable pour le bloc à coder (C). Si l'orientation du bloc (C) est égale à l'état d'orientation le plus probable, on ne transmet qu'un drapeau sinon cette orientation est complètement codée.

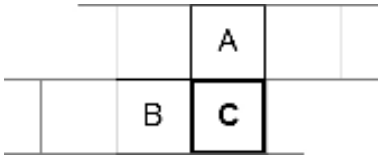


Figure 5 – *Blocs adjacents 4x4 ou 8x8 du bloc à coder*

Le décodage est effectué classiquement, le décodeur image ou vidéo utilisé lit toutes les informations y compris celle de permutations. Les macroblocs sont ensuite décodés, et si l'orientation des blocs n'est pas nulle ils sont réorientés. Dans le cas de AVC, les macroblocs résiduels sont décodés, puis réorientés (si besoin), et enfin ajoutés à leurs prédictions pour obtenir les macroblocs reconstruits.

5 Résultats expérimentaux

Le pré-traitement proposé a été implémenté dans le JM10 [13] fourni par le JVT, et uniquement pour les luminances des images résiduelles intra. Tous les tests ont été réalisés en profil High et à niveau 4.0 permettant ainsi l'utilisation

de CABAC et de FRExt [9] avec la transformée DCT 8x8. Les séquences sont générées en faisant varier les valeurs de QP des slices intra sur toute la plage disponible (0-51 et fixée à 28 pour les inter), elles sont alors composées de vingt images I.
Nous présentons ici deux séries de résultats : la première sans le codage de l'information de permutation, et la seconde avec cette information encodée.

5.1 Sans information de permutation

Les résultats sans le codage de l'information de permutation pour la séquence Container en CIF à 15Hz sont présentés sur la Figure 6 et ceux pour Mobile&Calendar en CIF à 15Hz sur la Figure 7.

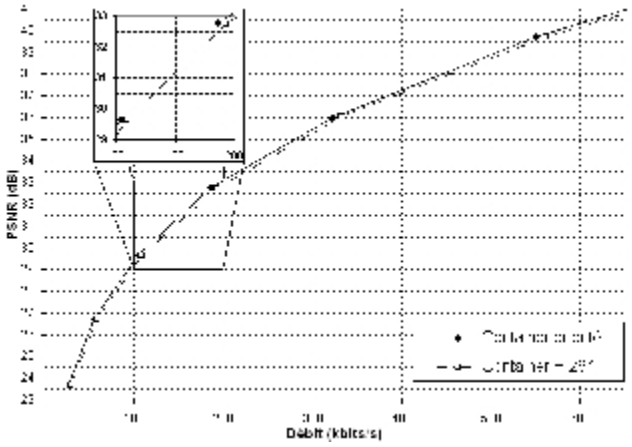


Figure 6 – *Résultats pour Container (CIF)*

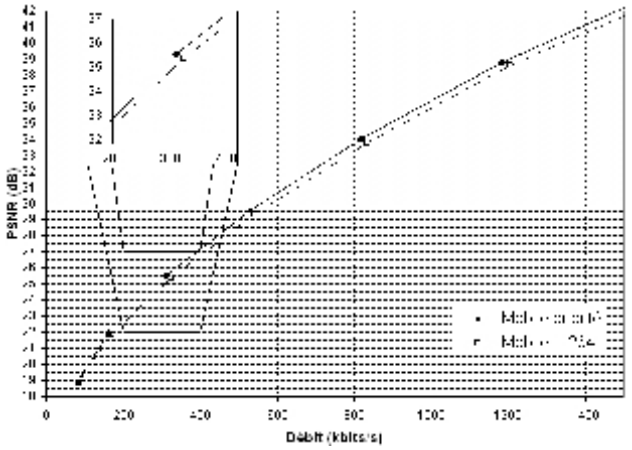


Figure 7 – *Résultats pour Mobile&Calendar (CIF)*

On peut voir sur ces figures que notre pré-traitement améliore le codage AVC à tous les débits. De plus, ces tests ont montré que toutes les possibilités de notre pré-traitement ont été exploitées : les images de la séquence Container sont composées de 396 macroblocs. A 550kbits/s, 179 de ces macroblocs ont été codés en intra 4x4, 188 en intra 8x8 et 29 en intra 16x16. Et dans tous les cas, les orientations ont pu être utilisées. L'amélioration de PSNR est à partir de 150kbits/s de près de 0.25dB pour Container et de près de 0.3dB pour Mobile&Calendar, et jusqu'à plus de 1dB dans les deux cas à très haut débit (à partir de 2500-3000kbits/s). Des résultats similaires ont été obtenus avec un grand nombre de séquences telles que Akiyo, Foreman, Bus, Tempete comme indiqué dans le Tableau 2. Par exemple, la séquence Tempete génère un gain de 0.32dB par rapport à AVC à 200kbits/s, et un gain supérieur à 0.5dB au delà de 3100kbits/s soit pour un QP inférieur à 9.

Séquence CIF	$d = 200kbits/s$ $\Delta_{PSNR} =$	$\Delta_{PSNR} > +0.5dB$ $d >$	$QP <$
Akiyo	+0.12dB	1200kbits/s	11
Bus	+0.18dB	2700kbits/s	9
Flower	+0.29dB	3000kbits/s	10
Football	+0.16dB	2600kbits/s	6
Foreman	+0.18dB	1900kbits/s	8
Tempete	+0.32dB	3100kbits/s	9
Séquence QCIF	$d = 80kbits/s$ $\Delta_{PSNR} =$	$\Delta_{PSNR} > +0.5dB$ $d >$	$QP <$
Carphone	+0.21dB	500kbits/s	9
Foreman	+0.19dB	600kbits/s	7

Tableau 2 – *Les résultats sur d'autres séquences*

5.2 Avec information de permutation

Dans un second temps, nous codons l'information de permutation selon la méthode présentée précédemment dans la section 4.2. Les courbes de débit-distorsion correspondantes pour les séquences Container et Mobile&Calendar sont présentées sur les Figures 8 et 9 respectivement.

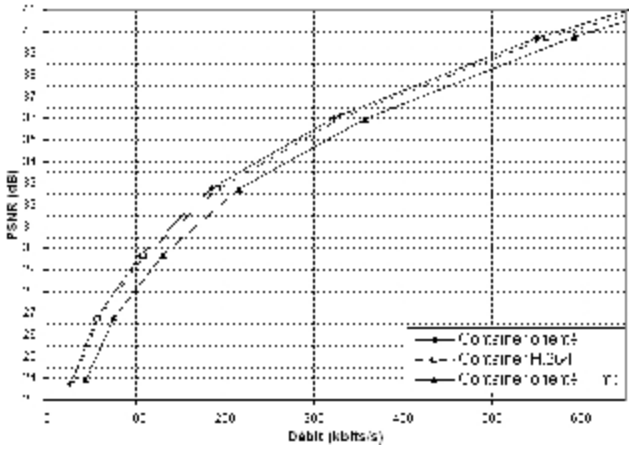


Figure 8 – *Résultats pour Container avec l'information de permutation*

Ces figures montrent que le débit nécessaire pour encoder l'information de permutation est supérieur au gain apporté

par notre pré-traitement jusqu'à un certain (haut) débit (ici près de 2000kbits/s). Pour la séquence Mobile&Calendar à 200kbits/s, l'information de permutation nous fait perdre 0.7dB contre un gain de 0.3dB sans cette information, soit une perte de 0.4dB par rapport à AVC. L'encodage simple de l'information que nous réalisons ici n'est pas efficace et nécessitera une amélioration future : en améliorant la prédiction du mode d'orientation le plus probable et en définissant de nouveaux contextes et éléments de syntaxe pour ces orientations et pour chaque taille de blocs (4x4, 8x8 et 16x16).

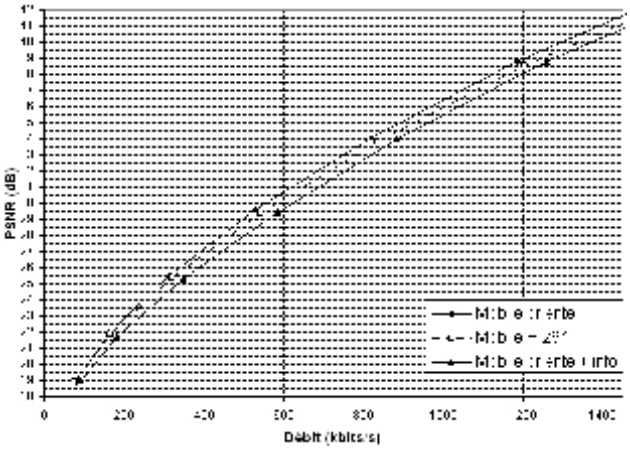


Figure 9 – *Résultats pour Mobile&Calendar avec l'information de permutation*

6 Conclusions et perspectives

Nous avons introduit dans cet article un pré-traitement pour des codeurs basés blocs qui tient compte de l'orientation de ces blocs pour en améliorer les performances. Prendre en considération l'orientation avant l'encodage nous permet d'adapter les blocs à la transformée sans la modifier. Ces blocs ou macroblocs sont orientés en appliquant de simples permutations circulaires au niveau pixel évitant ainsi les problèmes inhérents aux méthodes classiques de rotation avec interpolation. Le codeur doit ensuite encoder et transmettre l'information de permutation en utilisant CABAC. Le codage que nous adoptons ici pour cette information est très proche de celui utilisé pour les modes de prédiction intra de AVC. Ce pré-traitement semble prometteur puisqu'il n'est appliqué ici qu'aux images résiduelles intra d'un codeur AVC, images qui ne représentent qu'une faible proportion des images et des blocs d'une séquence vidéo.
Nos futurs travaux s'attacheront à améliorer ce pré-traitement. En particulier, nous travaillerons à réduire l'impact de l'information additionnelle d'orientation qui doit être transmise au décodeur, et de réduire la complexité de l'algorithme. Nous avons aussi l'intention d'étendre notre méthode au codage des chrominances et, dans le cas de la vidéo, aux images inter avec des blocs hybrides (16x8, 8x16, 8x4 et 4x8).

Références

[1] JVT - ISO/IEC 14496-10 AVC - ITU-T Recommendation H.264. *Advanced video coding for generic audio-visual services*. Draft ITU-T Recommendation and Final Draft International Standard, JVT-G050r1, 2003.

[2] E. Candès et D. Donoho. Curvelets, multiresolution representation, and scaling laws. Dans *Wavelet Applications in Signal and Image Processing VIII*. Proc. SPIE 4119, 2000.

[3] M.N. Do et M. Vetterli. The contourlet transform : An efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12) :2091–2106, décembre 2005.

[4] E. Le Pennec et S. Mallat. Sparse geometric image representations with bandelets. *IEEE Transactions on Image Processing*, 14(4) :423–438, avril 2005.

[5] G. Peyré et S. Mallat. Discrete bandelets with geometric orthogonal filters. *IEEE International Conference on Image Processing (ICIP'05)*, 1 :65–68, septembre 2005.

[6] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, et P.L. Dragotti. Directionlets : Anisotropic multidirectional representation with separable filtering. *IEEE Transactions on Image Processing*, septembre 2005.

[7] M. Unser, P. Thévenaz, et L.P. Yaroslavsky. Convolution-based interpolation for fast, high-quality rotation of images. *IEEE Transactions on Image Processing*, 4(10) :1371–1381, octobre 1995.

[8] ISO/IEC 10918. *Digital Compression and Coding of Continuous-Tone Still Images*. JPEG, 1994.

[9] JVT - ISO/IEC 14496-10 AVC - ITU-T Recommendation H.264 Amendment 1. *Advanced Video Coding Amendment 1 : Fidelity Range Extensions*. Draft Text of H.264/AVC Fidelity Range Extensions Amendment, juillet 2004.

[10] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, et G.J. Sullivan. Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7) :688–703, juillet 2003.

[11] D. Marpe, H. Schwarz, et T. Wiegand. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7) :620–636, juillet 2003.

[12] T. Wiegand, G.J. Sullivan, G. Bjontegaard, et A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7) :560–576, juillet 2003.

[13] Joint model 10, 2006. http :\\ iphome.hhi.de\ suehring\ tml\ index.htm.

Compression Vidéo Distribuée utilisant la TCQ et un Turbo Code

Khaled Lajnef[1] Christine Guillemot[1] Pierre Siohan[2]

[1] IRISA/INRIA, Campus de Beaulieu, 35042 RENNES

[2] France Télécom R&D, 4 rue du Clos Courtel, B.P 59, 35512 RENNES

[1] {prenom.nom}@irisa.fr[2] {pierre.siohan}@orange-ft.com

Résumé

L'idée principale de cette étude est d'utiliser la quantification codée par treillis (TCQ : Trellis Coded Quantizer) dans les systèmes de codage de deux sources distribuées basés sur la technique de Turbo Code. L'objectif est de diminuer la distorsion du système et de se rapprocher des bornes théoriques de Wyner-Ziv. Une application aux systèmes de compression vidéo utilisant la DCT (Discrete Cosine Transform) suivie d'un quantificateur TCQ est proposée. Les résultats de simulation indiquent des gains de performances par rapport aux techniques basées sur des quantifications scalaires.

Mots clefs

Compression, Codage de source distribuée, Codage vidéo, Turbo Code, TCQ.

1 Introduction

En codage vidéo, afin de réduire le débit, les systèmes actuels utilisent la corrélation temporelle en mettant en œuvre un codage prédictif compensé en mouvement. Ce choix se traduit par des encodeurs dont la complexité est en général grandement plus élevée (5 à 10 fois) que celle du décodeur. Cette différence est donc liée au fait que l'estimation et la compensation de mouvement sont calculées et effectuées au codeur, tandis que le décodeur va simplement utiliser les vecteurs de mouvement pour reconstruire l'image décodée. Ce schéma de conception asymétrique est tout à fait adapté pour les applications actuelles du codage vidéo, que ce soit la télévision numérique ou encore le téléchargement sur des mobiles à partir de serveurs. Le développement considérable des mobiles va inéluctablement accentuer le besoin inverse, c'est-à-dire celui de transmettre un flux vidéo vers une station de base. Pour facilité cette possibilité dans le cas de la vidéo il est plus judicieux de rechercher un schéma de codage dual du précédent avec un codeur de complexité relativement limitée et un décodeur disposant d'une puissance de traitement nettement plus importante. Dans ce contexte le codage de sources distribuées (DSC : Distributed Source Coding) peut être vu comme une façon différente d'exploiter cette corrélation temporelle en considérant, par exemple, que dans 2 images successives, la deuxième constitue une version bruitée de la première et en exploitant cette corrélation au décodeur.

Le codage de sources distribuées concerne le cas de signaux fortement corrélés que l'on code séparément et décode conjointement. Ce genre de techniques peut s'appliquer à des réseaux de capteurs mais également au codage vidéo. En particulier, le DSC a été récemment étudié comme solution potentielle pour la compression de l'information dans des applications exigeant des encodeurs simples. En pratique, les systèmes de compression vidéo appliquant le principe du DSC dans le domaine Pixel [1] ou transformé [2], [3] ont été décrits. Un aperçu complet sur la compression de vidéo distribuée peut être trouvé dans [4].

Le codage de sources distribuées est introduit par Slepian-Wolf [5] en 1973. Dans ce cas, il est possible de réaliser, sans perte en efficacité de compression, un codage séparé et un décodage conjoint de 2 sources corrélées X et Y à valeurs discrètes. Plus tard ce principe a été étendu par Wyner et Ziv [6] au cas de sources Gaussiennes à valeurs continues. Plusieurs techniques basées sur les codes de canal (codes convolutifs [7], turbo codes [8] ou encore des codes Low Density Parity Check (LDPC) [9]) ont été proposées pour approcher les limites théoriques. Dans ce cas, la compression de X est réalisée par la transmission de seulement des bits de parité. Au décodeur, la connaissance de Y (désigné par information de bord) et des bits de parité permettent de reconstruire X.

Dans le cas de codage Wyner-Ziv de deux sources Gaussiennes, le modèle de corrélation entre X et Y s'exprime par $X = Y + N$. N est une variable aléatoire Gaussienne indépendante de Y, de moyenne nulle et de variance σ^2 qui représente le degré de corrélation entre les deux sources. Le codeur Wyner-Ziv peut être vu comme un quantificateur concaténé en série avec un codeur Slepian-Wolf. Par conséquent, beaucoup de travaux ont été réalisés pour trouver le meilleur quantificateur adapté au codage des sources distribuées. La TCQ a été utilisée dans un codeur Wyner-Ziv dans [10], [11] et [12]. Introduites dans [13] en 1990, la TCQ a apporté une amélioration considérable à haut débit par rapport à la quantification scalaire. Nous proposons

dans notre étude l'utilisation de la TCQ combinée avec un codeur turbo poinçonné dans un schéma de codage Wyner-Ziv.

Cet article est organisé comme suit. Dans une première partie (section 2), nous présentons le principe de la TCQ. Dans la section 3, nous considérons le cas du codeur de sources distribuées utilisant la TCQ et le Turbo Code. La section 4 propose une solution au codage vidéo basé sur la DCT et utilisant le principe de DSC et la quantification TCQ. La section 5 présente les résultats de simulation obtenus avec la quantification TCQ. Enfin, dans la section 6 nous concluons.

2 Principe de la TCQ

L'approche de la TCQ consiste à partitionner un dictionnaire de quantification initial en sous-dictionnaires complémentaires associés aux transitions entre les états d'un code convolutif.

Soit une source X *i.i.d* à quantifier en utilisant la TCQ. Le taux de compression désiré est de R bits/symbole (le quantificateur TCQ est appelé dans ce cas R-bits TCQ). Considérons un quantificateur scalaire uniforme dont le dictionnaire de quantification D est de cardinal 2^{R+1} et un code convolutif de rendement 1/2.

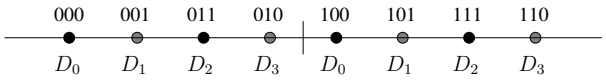


Figure 1 – *Exemple de partition pour une TCQ à 2 bits/symbole.*

Le dictionnaire du quantificateur scalaire D est partitionné en 4 sous-dictionnaires D_0, D_1, D_2 et D_3 comprenant chacun 2^{R-1} mots de code. Chaque sous-dictionnaire représente une transition dans le treillis du code convolutif. Comme exemple, considérons la partition de la figure 1 correspondant au débit $R = 2$ bits/symbole (2-bits TCQ) et le treillis d'un codeur convolutif de rendement 1/2 à la figure 2 (les branches en traits discontinus signifient que le bit inséré est 0 alors que les branches en traits continus correspondent à 1). Etant donné la source X, le décodeur Viterbi est utilisé pour chercher la version quantifiée X_Q la plus proche de la source X avec une distance euclidienne minimale, c'est-à-dire une erreur quadratique la plus faible. A la sortie du quantificateur TCQ, la séquence de X_Q est représentée par deux séquences. La première séquence, désignée par Ct_X, est constituée des bits spécifiés par le chemin du treillis à la fin du décodage Viterbi. La deuxième séquence Mc_X est composée des mots de code de longueur $R-1$ bits appartenant aux sous-dictionnaires indexés par le bit de transition.

La reconstruction de la source X s'effectue comme l'indique la figure 3 en deux étapes. D'abord, la séquence Ct_X est codée en utilisant le codeur convolutif de rendement 1/2 pour récupérer la séquence de sous-dictionnaires Cc_X. Ensuite, les mots de code de la séquence Mc_X indexent dans

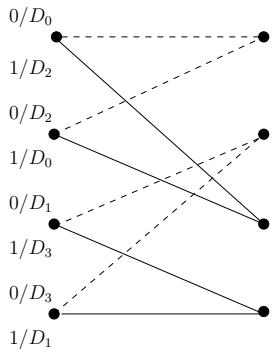


Figure 2 – *Treillis d'un code convolutif de rendement 1/2 associé à la partition de la figure 1.*

le sous-dictionnaire la valeur de reconstruction $\hat{X} = X_Q$ (figure 3).

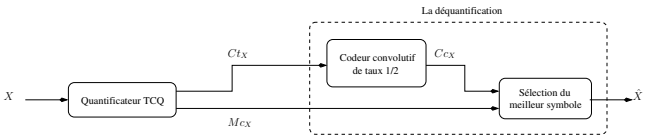


Figure 3 – *Quantification et déquantification d'une source X en utilisant la TCQ.*

3 Codeur Wyner-Ziv basé sur la TCQ et le Turbo Code

Soient X et Y deux sources *i.i.d* Gaussiennes corrélées. Le modèle de corrélation entre Y et X est défini par : $X = Y + N$ avec N un bruit Gaussien *i.i.d* de moyenne nulle et de variance σ^2. N est indépendant de $Y \sim \mathcal{N}(0, \sigma_Y^2 = 1)$. On définit par Corrélation-SNR (CSNR=$\frac{\sigma_Y^2}{\sigma^2}$) le terme qui représente le rapport des variances de Y et N. On suppose que les valeurs de la source Y sont connues au décodeur [6]. Pour diminuer la distorsion de la source X, nous proposons d'utiliser un quantificateur TCQ (R-bits TCQ) comme l'indique la figure 4.

Les symboles de la séquence de mots de code Mc_X sont codés avec un codeur turbo constitué de deux codeurs convolutionnels récursifs systématiques (RSC : Recursive Systematic Code) de taux $n/n+1$ concaténés en parallèle et séparés par un entrelaceur. Les sorties de chaque codeur sont poinçonnées. Pour obtenir un taux de compression élevé, les bits systématiques des deux codeurs élémentaires sont éliminés et seulement quelques bits de parité sont transmis au décodeur.

L'entropie conditionnelle $H(Ct_X|Y)$ des bits de la séquence Ct_X s'approche de un quand le débit R augmente. Par conséquent et à l'inverse des méthodes proposées dans [10], [11] et [12], les bits de la séquence Ct_X sont transmis sans compression au décodeur.

La région des taux de compression admissible dans ce cas est définie par :

$$R_X \quad \geq \quad R^*_{X|Y}(D_X) = I(X; X_Q|Y) \qquad (1)$$

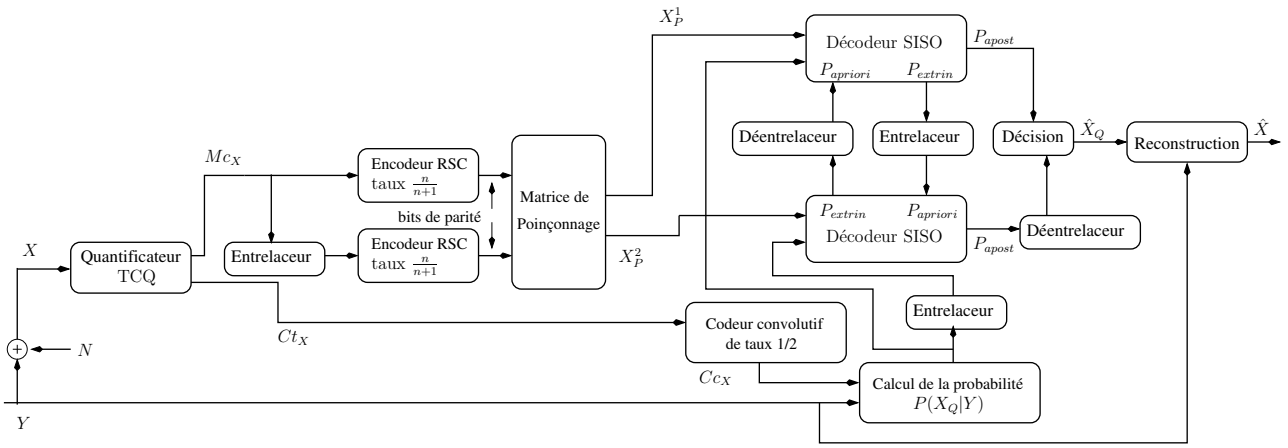


Figure 4 – *Schéma d'un codeur Wyner-Ziv utilisant la TCQ et le Turbo Code.*

$$D_X \quad \geq \quad E[d(X, \hat{X})] \tag{2}$$

avec X_Q la version quantifiée de X et D_X la valeur moyenne de la distorsion théorique entre X et $\hat{X}$ (la valeur reconstruite de X) qui s'exprime par :

$$D_X = \sigma^2 / 2^{2R^*_{X|Y}(D_X)} \tag{3}$$

Au décodage, les bits de la séquence Ct_X sont codés en utilisant le codeur convolutif de rendement 1/2 pour récupérer la séquence de sous-dictionnaire (Cc_X). Pour estimer $\hat{X}_Q$, un décodeur turbo composé de deux décodeurs (SISO : Soft-Input Soft-Output) concaténés en série via un entrelaceur/désentrelaceur est utilisé. Chaque décodeur prend en entrée les bits de parité générés par l'encodeur correspondant et la probabilité $P(X_Q|Y)$ qui dépend de l'information de bord Y et de la séquence Cc_X. Pour déterminer la probabilité $P(X_Q|Y)$ et à la différence de [12], les régions d'intégration dans notre cas ne sont que celles qui correspondent aux sous-dictionnaires Cc_X. En effet, pour améliorer les performances du décodage turbo, on profite de la disponibilité de la séquence Cc_X qui a été compressée sans perte.
La reconstruction optimale de X peut être déterminée à partir de $\hat{X}_Q \in [a, b]$ et Y comme suit :

$$\begin{aligned} \hat{X} &= E(X = x|\hat{X}_Q, Y = y) \\ &= \int_a^b x\ P(X = x \in [a, b]|\hat{X}_Q, Y = y)dx \\ &= \int_a^b x\ \frac{P(\hat{X}_Q|x)P(x|y)}{P(\hat{X}_Q|y)}dx \end{aligned} \tag{4}$$

4 Codeur vidéo distribuée dans le domaine transformé utilisant la TCQ et le Turbo Code

Un codeur vidéo distribuée (DVC : Distributed Video Coding) est un système basé sur un codeur Wyner-Ziv, où les images d'une séquence sont codées indépendamment (Intra codées) mais décodées conjointement (Inter décodées).

Les images d'une séquence vidéo sont divisées en deux sous-ensembles comme l'indique la figure 5. Un premier sous-ensemble d'images, régulièrement espacées dans la séquence, va être codé en mode Intra et constitué des images clés ("Keyframe") désignées par I. Ces images sont codées avec un codeur H.264. Le deuxième sous-ensemble d'images est codé en utilisant le codeur Wyner-Ziv. Ces images sont désignées par WZ (images Wyner-Ziv).

Une séquence vidéo est divisée en GOP (Group Of Pictures) contenant un certain nombre d'images. La première image de chaque GOP est codée en mode Intra et les restantes sont codées avec un codeur Wyner-Ziv.

Pour les images Wyner-Ziv WZ, chaque bloc de 4×4 pixels est transformé en un bloc de 4×4 coefficients grâce à la transformée DCT. Les coefficients de la transformée de chaque bloc sont groupés en 16 sous-bandes. La bande 0 contient les coefficients les plus significatifs alors que la bande 15 est composée en général de coefficients à valeur très faible.

Après la DCT, un quantificateur TCQ est utilisé seulement pour les coefficients de bandes 0 pour préserver une complexité faible à l'encodage. La séquence Ct_{b0} est envoyée directement au décodeur. Alors que la séquence de mots de code Mc_{b0} est codée par plans de bits. Quant aux coefficients des autres sous-bandes (autres que zéro), le quantificateur scalaire uniforme (UQ) est utilisé. Les coefficients quantifiés sont codés par plans de bits. Chaque plan de bits est ensuite compressé en utilisant un codeur turbo poinçonné et seulement les bits de parité retenus sont transmis au décodeur.

Les images I sont décodées avec un décodeur H.264. Ainsi, pour chaque image Wyner-Ziv, une information de bord SI est générée par interpolation, compensée en mouvement à partir des images I. Comme pour les images WZ au codage, une transformée DCT est appliquée aux images SI et des bandes de coefficients sont formées. Le modèle de corrélation considéré entre les coefficients de DCT des images WZ et leurs correspondants des images SI (l'information de bord) est une distribution Laplacienne.

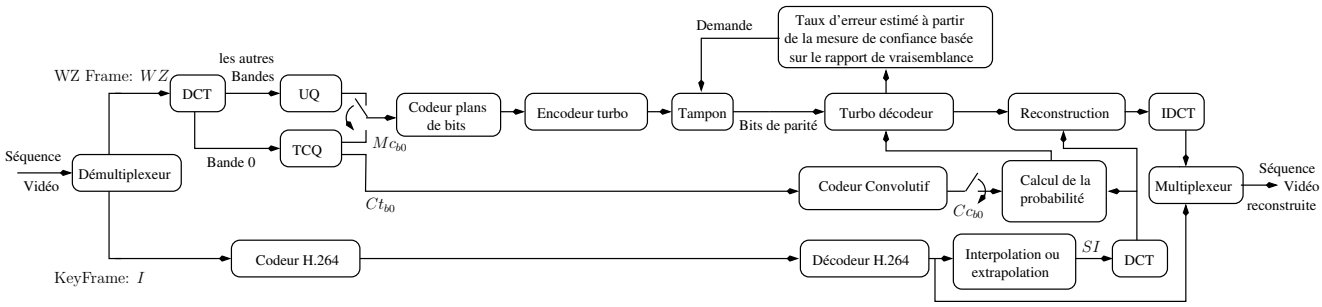


Figure 5 – *Schéma d'un codeur vidéo distribuée utilisant la TCQ et le Turbo Code.*

Pour chaque bande non nulle des images WZ, le décodeur turbo utilise les coefficients de DCT de l'information de bord SI et les bits de parité reçus pour estimer les plans de bits à l'entrée du codeur turbo. Si le nombre d'erreurs estimé est supérieur à 10^{-3}, alors une demande de bits de parité additionnels est envoyée vers le tampon du codeur turbo par "feed-back". Le taux d'erreur est estimé à partir de la mesure de confiance basée sur le rapport de vraisemblance obtenue à la sortie du décodeur turbo. Le processus est répété jusqu'à ce qu'une probabilité d'erreur acceptable (10^{-3}) soit atteinte.

Pour les bandes 0, le même processus sera répété pour les plans de bits sauf qu'avant le décodage turbo, la séquence Cc_{b0} doit être récupérée par codage convolutif de la séquence Ct_{b0}. Puis, les probabilités se basant sur des distributions Laplaciennes à l'entrée du turbo décodeur sont calculées seulement dans les régions de quantification où leurs indexes contiennent les symboles de Cc_{b0}.

Après le décodage turbo, les indices des coefficients DCT quantifiés sont estimés. Chaque bande de l'image WZ est reconstruite. Si le coefficient DCT de l'information de bord et l'indice décodé sont dans la même région de quantification (fixée par l'indice décodé) alors le coefficient DCT reconstruit prendra une valeur égale à celle de l'information de bord. Cependant, si le coefficient DCT de l'information de bord est inférieur à l'extrémité inférieure de la région de quantification, alors le coefficient DCT reconstruit prendra la valeur de cette extrémité. Par contre, si le coefficient DCT de l'information de bord est supérieur à l'extrémité supérieure de la région de quantification, alors la valeur reconstruite prendra la valeur de cette extrémité supérieure. Enfin, une transformée inverse de la DCT (IDCT) sera appliquée sur les coefficients reconstruits.

5 Résultats de simulation

Tous les résultats de simulation présentés sont réalisés avec un algorithme de décodage MAP (Maximum A Posteriori). L'entrelaceur utilisé est aléatoire et la longueur de contrainte du codeur convolutif de la quantification TCQ est $K_{TCQ} = 9$.

5.1 Cas de deux sources Gaussiennes

Le codeur RSC utilisé est de taux de codage 2/3, de longueur de contrainte $K = 5$. Le vecteur générateur du code élémentaire est $G = (1, 23, 35/27)$, la taille de chaque bloc est 10^5, et enfin le nombre de bits simulés est 10^7. Le quantificateur TCQ utilisé est 3-bits TCQ.

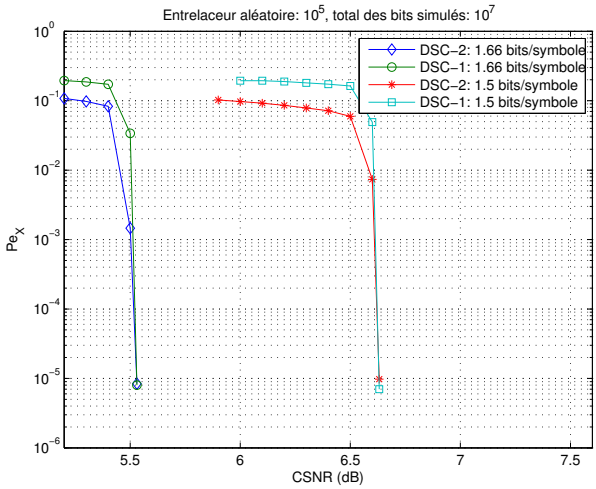


Figure 6 – *Résultats de simulation pour DSC avec deux sources Gaussiennes : Probabilités des symboles erronés.*

Dans la figure 6, la probabilité des symboles erronés $Pe_X = P(X_Q \neq \hat{X}_Q$ est tracée en fonction de la Corrélation-SNR (CSNR) pour deux codeurs DSC utilisant un quantificateur scalaire uniforme (UQ) (le codeur est désigné par DSC-1) et un 3-bits TCQ (désigné par DSC-2), respectivement. Les performances obtenues avec des débits de 1.5 et 1.6 bits/symbole peuvent être observées dans la figure 6.

Pour des débits de 1.5 et 1.66 bits/symbole, la figure 7 montre les valeurs des distorsions moyennes mesurées de la source X obtenues avec les deux codeurs DSC-1 et DSC-2. Pour un débit de 1.66 bits/symbole, la distorsion moyenne de X dans un DSC-2 basé sur la TCQ se situe à une distance de 0.916 dB de la borne théorique de Wyner-Ziv. Cependant, pour le DSC-1 la distance entre les distorsions théorique et mesurée est de l'ordre de 3.46 dB. Pour des débits plus faibles (1.5 bits/symbole), la différence entre la distorsion mesurée et la borne théorique diminue avec le DSC-2. Alors que pour le DSC-1, une petite amélioration de la distorsion peut être ob-

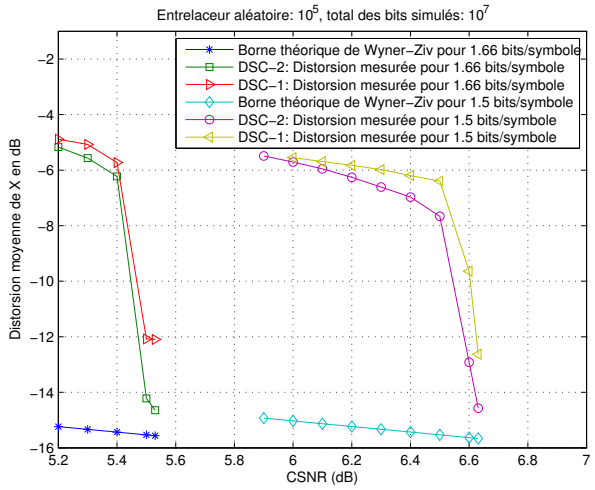


Figure 7 – *Résultats de simulation pour DSC avec deux sources Gaussiennes : Distorsion moyenne mesurée*

servée. Néanmoins, le DSC-2 s'approche mieux de la borne théorique que le DSC-1 (pour 1.5 bits/symbole, la différence avec la limite de Wyner-Ziv est de l'ordre de 1.092 dB pour DSC-2 alors qu'elle est égale à 3.04 db avec le DSC-1).

5.2 Application aux séquences vidéos

Le codeur RSC utilisé ici est de taux de codage 1/2, de longueur de contrainte $K = 5$ et de vecteur générateur $G = (1, 33/23)$. La séquence vidéo à coder est Foreman en format QCIF et de fréquence égale à 15 Hz. Ici, nous ne nous intéressons qu'aux blocs de luminance (Y) des images WZ.

Nous comparons deux codeurs vidéos distribuées en terme de performance débit-distorsion. Le premier codeur (désigné par DVC-1) utilise différents quantificateurs scalaires uniformes pour chaque bande de la DCT. Quant au deuxième codeur (désigné par DVC-2), les bandes 0 sont quantifiées avec la TCQ et pour les autres bandes on utilise différents quantificateurs scalaires uniformes. Les résultats de simulation obtenus avec la TCQ sont réalisés avec différents débits : 3-bits TCQ, 4-bits TCQ, 5-bits TCQ et 6-bits TCQ.

Pour un GOP=2 (c'est-à-dire entre deux images intra, il y a une seule image WZ), la figure 8 illustre le gain de performance obtenu avec la quantification TCQ au niveau de la bande 0 par rapport à la quantification uniforme. A un débit (débit global des images WZ et I) de 500 kbps, on peut voir que le $PSNR$ de Y (des images WZ et I) obtenu avec le DVC-2 utilisant la TCQ est meilleur de 0.29 dB que celui obtenu avec un DVC-1 à base d'un quantificateur scalaire uniforme. Ce gain de performance augmente en fonction de la taille du GOP et du débit. Dans la figure 9 pour un GOP=4, la différence de $PSNR$ entre les deux approches passe de 0.39 dB à 0.435 dB pour des débits de 500 à 600 kbps, respectivement. La figure 10 illustre un gain obtenu avec le DVC-2 de l'ordre de 0.568 dB par rapport à DVC-1 pour un GOP=8 et un débit de 700 kbps. Le DVC-2 est pénalisé par les performances de la TCQ à faible débit. On observe que plus le débit est petit, moins le gain de performances du DVC-2 par rapport à DVC-1 est important.

Pour mieux illustrer l'effet de la TCQ sur les bandes 0, la figure 11 montre les résultats de simulation des deux codeurs DVC codant et décodant seulement les bandes 0 (les autres bandes ne sont pas ni codées ni décodées). Pour un GOP=4, le gain de performance du DVC-2 par rapport à DVC-1 est de 0.717 dB pour un débit égal à 100 kbps.

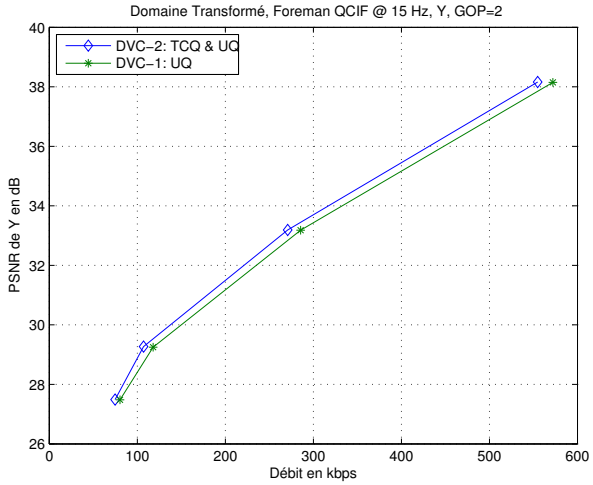


Figure 8 – *Résultats de simulation pour DVC d'une séquence Foreman QCIF, 15 Hz avec un $GOP = 2$.*

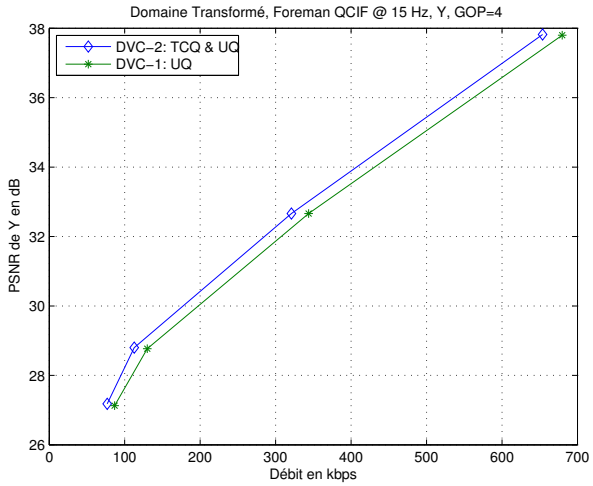


Figure 9 – *Résultats de simulation pour DVC d'une séquence Foreman QCIF, 15 Hz avec un $GOP = 4$.*

6 Conclusion

Ce papier décrit un codeur de sources distribuées combinant une quantification TCQ et un Turbo Code. Du point de vue théorique, nous observons que la distorsion mesurée du système peut s'approcher de 0.916 dB de la borne de

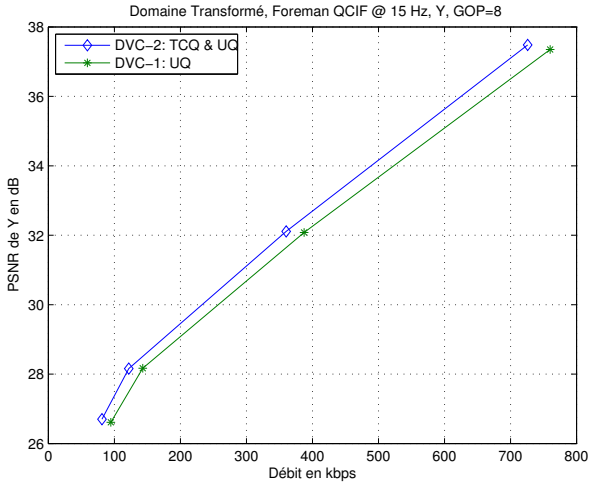


Figure 10 – *Résultats de simulation pour DVC d'une séquence Foreman QCIF, 15 Hz avec un* $GOP = 8$.

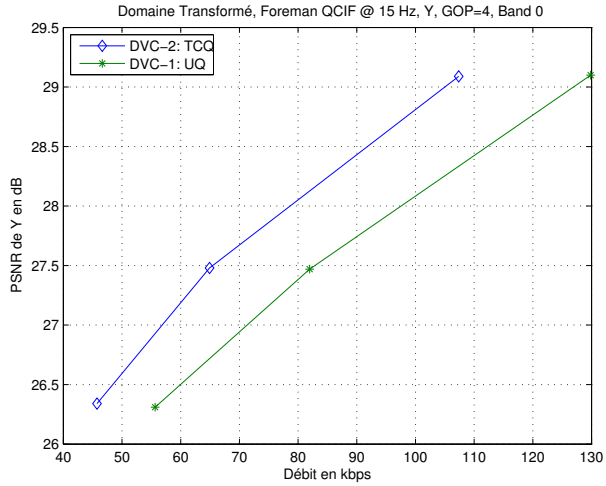


Figure 11 – *Résultats de simulation pour DVC d'une séquence Foreman QCIF, 15 Hz avec un* $GOP = 4$ *: seule la bande 0 est codée et décodée.*

Wyner-Ziv. Pour une application au codage vidéo, l'utilisation de la TCQ dans un système DVC permet d'améliorer les performances en terme de débit-distorsion. Nous orientons actuellement nos travaux vers une utilisation complète de la TCQ pour toutes les bandes de la DCT.

Références

[1] A. Aaron, R. Zhang and B. Girod, "Wyner-Ziv Coding of Motion Video" , in *Proc. 36th Asilomar Conference on Signals, Systems and Computer, Pacific Grove, USA*, Nov. 2002.

[2] R. Puri and K. Ramchandran, "PRISM : A new robust video coding architecture based on distributed compression principles" in *Proc. Allerton Conference on Communication, Control and Computing*, Oct. 2002.

[3] A. Aaron, S. Rane, E. Setton and B. Girod, "Transform-domain Wyner-Ziv Codec for Video", in *Proc. SPIE Conference on Visual Communication and Image Processing*, Jan. 2004.

[4] B. Girod, A. Aaron, S. Rane, D. Rebollo-Monedero, "Distributed Video Coding", in *Proc. IEEE, Special issue on advances in video coding and delivery*, vol.93, No. 1, pp. 71-83, Invited paper, Jan. 2005.

[5] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources", *IEEE Trans. Inform. Theory*, IT-19, pp. 471-480, Mar. 1973.

[6] A. D. Wyner and J. Ziv, "The Rate-Distortion Function for Source Coding with Side Information at the Decoder", *IEEE Trans. Inform. Theory*, vol-22, pp. 1-10, Jan. 1976.

[7] S.S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS) : Design and Construction", *Proc. IEEE DCC*, pp. 158-167, Mar. 1999.

[8] A. Aaron and B. Girod, "Compression with Side Information using Turbo Codes", *Proc. IEEE DCC*, pp. 252-261, Apr. 2002.

[9] A. D. Liveris, Z. Xiong and C. N. Georghiades, "Compression of Binary Sources with Side Information at the Decoder using LDPC Codes", *IEEE Comm. Letters*, Vol-6, pp. 440-442, Oct. 2002.

[10] S. S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS) : Design and construction," *IEEE Trans. Inf. Theory*, VOL. 49, NO. 3, pp. 626–643, Mar. 2003.

[11] J. Chou, S. Pradhan, and K. Ramchandran, "Turbo and trellis-based constructions for source coding with side information," *Proc. IEEE DCC*, Snowbird, UT, March 2003.

[12] Y. Yang, S. Cheng, Z. Xiong and W. Zhao, "Wyner-Ziv coding based on TCQ and LDPC codes," *Proc. Asilomar Conference on Signals, Systems and Computer*, Pacific Grove, CA, Nov. 2003.

[13] M. W. Marcellin and T. R. Fischer, "Trellis Coded Quantization of Memoryless and Gauss-Markov Sources", *IEEE Trans. Comm.*, Vol. 38, No. 1,pp. 82-93, Jan. 1990.

Modèle énergétique pour la représentation d'images par ondelettes déformées

Benjamin Le Guen[1,2] Stéphane Pateux[1] Jacques Weiss[2]

[1] France Télécom R&D, 4, rue du Clos Courtel, 35512 Cesson-Sévigné
{benjamin.leguen, stephane.pateux}@orange-ft.com
[2] Supélec-SCEE/IETR-AC, avenue de la Boulaie, 35511 Cesson-Sévigné
jacques.weiss@supelec.fr

Résumé

Une image comporte une structure géométrique que l'ondelette séparable classique ne peut exploiter. Ce papier décrit un nouveau schéma d'analyse-synthèse permettant d'apporter une dose d'adaptivité au noyau classique. Une modélisation globale de la structure géométrique par un maillage quadrangulaire régulier est proposée. Les positions des noeuds du maillage sont adaptées au signal via la minimisation d'une fonctionnelle que nous dérivons. A l'issue de l'analyse, l'image d'origine est représentée par une information de texture adaptée au noyau classique et par une information de géométrie. Cette représentation permet d'injecter à l'ondelette une dose de directionnalité et d'anisotropie tout en conservant sa propriété de multi-résolution. Elle est donc adaptée à une application de compression scalable. Dans ce cadre, des gains perceptuels sensibles sont observés dans les régions singulières, telles que les contours.

Mots clefs

Ondelettes déformées, Maillage, Estimation de géométrie, Compression scalable.

1 Introduction

1.1 Limite des Ondelettes Classiques

L'ondelette est un noyau de représentation puissant au coeur de nombreuses recherches. Cependant, son extension séparable classique en 2D a fourni des résultats d'approximation reconnus comme sous-optimaux [1], bien que prometteurs. En effet, les discontinuités d'une fonction bi-dimensionnelle présentent souvent une structure géométrique que le noyau classique n'exploite pas. Dans le cas de l'image, ces singularités, ou contours, possèdent pourtant un fort poids sémantique et perceptuel. La Figure 1 schématise sur la gauche un contour et le support d'une ondelette 2D séparable classique. Le support de l'ondelette ne capte pas la régularité existant le long du contour, ce qui provoque une perte en compacité de représentation. Lors d'une approximation, cette perte se répercute visuellement par des rebonds tout autour de la singularité. Une adaptation du noyau au contour est également illustrée sur la droite et met en avant les deux propriétés qui font défaut au noyau classique : la directionnalité et l'anisotropie.
L'objectif est donc de rechercher de nouvelles bases de représentation exhibant ces deux propriétés tout en conservant celles ayant fait le succès des ondelettes classiques : échantillonnage critique, localisation (spatiale et fréquentielle), multi-résolution (voir [2]).

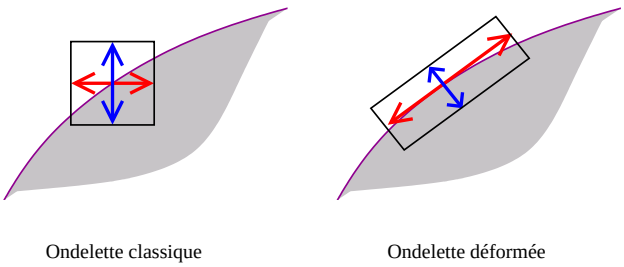


Figure 1 – *A gauche, le support rectangulaire des ondelettes classiques est inadapté au contour. A droite, l'ondelette a été déformée pour s'adapter au contour.*

1.2 Ondelettes Déformées : Art Antérieur

De nouvelles transformées s'appuyant sur une déformation de l'ondelette ont émergé au cours de la dernière décennie. Certaines reposent sur des bancs de filtres fixes analysant l'image à des échelles, positions et orientations données. Les Ridgelets [3], Curvelets [4], ou Contourlets [2] en sont les exemples les plus connus. D'autres méthodes, dites adaptatives, proposent d'estimer la structure d'une image par un modèle géométrique explicite. Les déformations à appliquer à l'ondelette sont alors dictées par l'instanciation de ce modèle à l'image étudiée, et une information supplémentaire de géométrie est nécessaire pour représenter l'image. Nos travaux se situent dans cette seconde catégorie de méthodes, dont les Bandelettes [1, 5] et les Ondelettes Directionnelles [6, 7] sont des exemples.
En particulier, le modèle utilisé par les Bandelettes de $1^{ère}$ génération est un quad-tree, dont chaque feuille correspond à un bloc de l'image. Une minimisation énergétique basée sur un gradient local associe à chaque bloc une fonction paramétrique captant le flux de régularité. La Figure 2 illustre

un flux de régularité type calculé par les Bandelettes sur une zone image. Une fois le modèle géométrique instancié, l'approche suivie par les Bandelettes de $1^{ère}$ génération possède deux spécificités intéressantes :

– *Extraction de la géométrie* par déformation du bloc. Cette déformation est dictée par la fonction régulière calculée et réaligne les contours sur l'horizontale ou la verticale. Lorsque l'alignement des contours est optimal, la lacune en directionnalité de l'ondelette 2D classique s'annule.
– *Bandelettisation*. Ce procédé permet d'injecter au noyau une dose d'anisotropie en augmentant son échelle dans la direction de régularité. Cet allongement de l'ondelette le long du flux géométrique permet ainsi de capter sa régularité et aboutit à une meilleure décorrélation spatiale.

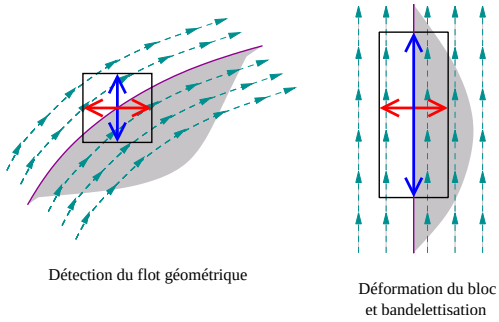


Figure 2 – *Procédé de bandelettisation.*

La qualité de ces deux étapes dépend directement de la qualité du modèle géométrique choisi et du mode de calcul des paramètres. Bien que performant, le quad-tree souffre d'une limitation : chaque région étant analysée de manière indépendante, les déformations des noyaux ne sont pas continues à la frontière des blocs, provoquant une perte d'orthogonalité. Par ailleurs, l'étape de bandelettisation ne permet de modifier le support de l'ondelette que dans la direction du flux, la direction orthogonale étant négligée. Enfin, utiliser un critère local pour le calcul des paramètres pose un problème de robustesse, car les données d'une image sont très bruitées.

L'étude de l'art antérieur a motivé la recherche d'un modèle géométrique souple permettant une déformation continue du noyau d'ondelette 2D. La seconde section de cet article pose les grandes lignes d'un schéma d'analyse-synthèse basé sur une modélisation de la géométrie par un maillage quadrangulaire déformable. L'étape d'analyse permet de déterminer les positions des sommets du maillage. Elle est basée sur une modélisation énergétique du problème décrite dans la troisième partie. Cette modélisation permet de ne pas devoir recourir à un critère local. Enfin, la dernière partie montre des résultats d'analyse et teste les performances en compression du schéma proposé.

2 Schéma d'Analyse-Synthèse

Le principe général du schéma décrit dans cette section est basé sur l'hypothèse suivante : adapter l'ondelette à l'image ou adapter l'image à l'ondelette sont deux processus similaires. Le problème peut ainsi être inversé pour se concentrer sur la définition et le calcul d'une déformation de l'image qui soit adaptée au noyau classique. En se référant à l'a priori que les contours sont les régions mal représentées par ce noyau, la déformation calculée aura idéalement la propriété d'aligner les contours sur l'horizontale ou la verticale, et de les contracter en fonction de leur régularité tangentielle et orthogonale. De façon équivalente, l'alignement apportera donc au noyau une dose de directionnalité et les contractions (ou étirements) une dose d'anisotropie (voir Figure 3).

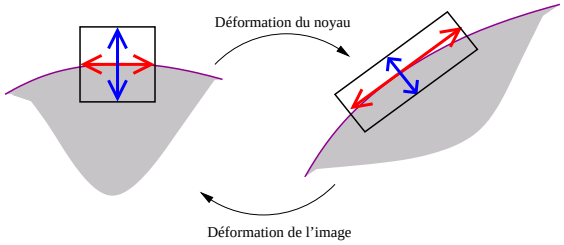


Figure 3 – *Adaptation d'un contour au noyau classique.*

2.1 Maillage comme Modèle de Déformation

Soit $\mathcal{M}$ un maillage quadrangulaire régulier déformable comportant M sommets. Nous utilisons $\mathcal{M}$ comme modèle géométrique de déformation. Les positions $\{P_i = (x_i, y_i)\}_{1 \leq i \leq M}$ des sommets dans le domaine image sont les paramètres du modèle. Nous définissons alors un mapping de coordonnées global noté τ_{TI} (Figure 4) :

$$\begin{array}{rcl} \tau_{TI} : \mathcal{D}_T & \longrightarrow & \mathcal{D}_I \\ (u,v) & \longmapsto & (x,y) = \sum_i P_i \cdot \phi_i(u,v) \end{array}$$

où $\mathcal{D}_I$ représente le domaine image, $\mathcal{D}_T$ le domaine déformé aussi appelé domaine texture, et $\phi_i(u,v)$ une fonction de forme définie dans $\mathcal{D}_T$ (la fonction d'interpolation bilinéaire par exemple).

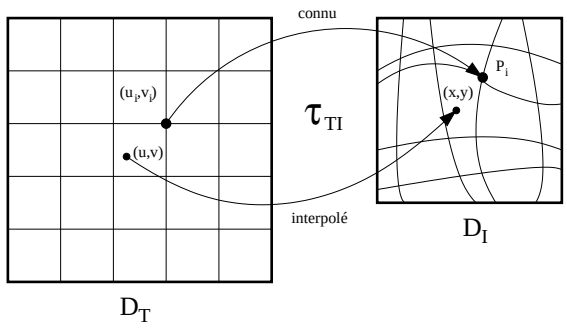


Figure 4 – *Mapping de coordonnées global entre $\mathcal{D}_T$ et $\mathcal{D}_I$.*

La déformation dictée par τ_{TI} revient donc à projeter $\mathcal{M}$ sur une grille régulière uniforme fixée dans $\mathcal{D}_T$. Cette déformation est globale et évolue de façon continue à la frontière des mailles. Nous notons I l'image originale et T l'image obtenue par déformation dans $\mathcal{D}_T$. T est aussi nommée texture :

$$T(u,v) = I[\tau_{TI}(u,v)] = I[\sum_i P_i \cdot \phi_i(u,v)]$$

2.2 Analyse

La recherche des positions $\{P_i\}_i$ est la problématique principale du schéma. La qualité de la décorrélation spatiale en dépend directement. Plusieurs méthodes ont été proposées pour rééchantillonner une image à l'aide d'une grille régulière adaptée [8, 9]. D'autres ont été proposées pour le remaillage de surface [10]. Cependant, l'application de ces méthodes, basées sur l'utilisation de critères locaux, montre des limites en terme de robustesse car les données d'une image se révèlent très bruitées.
Dans [11], les auteurs proposent une formulation énergétique basée sur une erreur totale d'interpolation dans le domaine texture [1]. Dans la lignée de ces travaux, nous introduisons en section 3 une modélisation énergétique découlant d'une formulation inverse du problème. Une fois les paramètres calculés, la texture est obtenue en extrayant de l'image originale sa structure géométrique. Cette texture est ensuite décomposée dans la base d'ondelettes séparables classiques pour laquelle les paramètres ont été optimisés. $\mathcal{M}$ pouvant se représenter sur différents niveaux de résolution, la décomposition $\{texture, géométrie\}$ permet en particulier de conserver la propriété de multi-résolution de l'ondelette classique.

2.3 Synthèse

Connaissant le mapping τ_{TI}, il est possible de calculer le mapping inverse $(\tau_{TI})^{-1}$. Une méthode de calcul pour un maillage quadrangulaire est fournie dans [12] dans le cas où $\phi_i(u,v)$ est la fonction bilinéaire. Connaissant $(\tau_{TI})^{-1}$ et la texture T, on reconstruit l'image I :

$$I = T((\tau_{TI})^{-1}(x,y))$$

[1]Dans [11], le domaine texture est appelé domaine *maître*.

3 Modèle Energétique d'Analyse

3.1 Formulation du Problème

Pour calculer les paramètres du modèle géométrique, les techniques adaptatives antérieures raisonnent dans le domaine image en utilisant des a priori, tel que l'alignement dans la direction orthogonale au gradient [1]. Nous raisonnons dans le domaine texture en proposant la formulation inverse suivante : *Quelle est la texture la mieux adaptée à une représentation par ondelettes classiques ?* Ce qui peut être reformulé de la manière suivante : *Quels sont les paramètres $\{P_i\}_i$ qui minimisent un coût de description de la texture dans $\mathcal{D}_T$?* Cette formulation permet de n'émettre aucune hypothèse sur les régions sensibles de l'image, et donc de ne pas recourir à des critères locaux. Elle impose néanmoins de définir la notion de *coût de description texture*, objet du paragraphe suivant.

3.2 Coût de Description Texture E_T

Considérons la décomposition en ondelettes de la texture sur J niveaux de résolution :

$$T(u,v) = \sum_{\mathbf{k}} c^J_{\mathbf{k}} \varphi^J_{\mathbf{k}}(u,v) + \sum_{j=1}^{J} \sum_{\mathbf{k}} d^j_{\mathbf{k}} \psi^j_{\mathbf{k}}(u,v), \quad (1)$$

où $\varphi^j_{\mathbf{k}}$ et $\psi^j_{\mathbf{k}}$ sont les versions translatées en $\mathbf{k} = 2^j(k_1,k_2)$ ($(k_1,k_2) \in \mathbb{Z}^2$) et dilatées du facteur 2^j d'une fonction d'échelle φ et d'une fonction d'ondelette ψ.
Nous définissons alors le *coût de description texture* E_T comme la somme pondérée des énergies hautes fréquences :

$$E_T = \sum_{j=1}^{J} \upsilon_j^2 \cdot E_j, \quad (2)$$

où $E_j = \sum_{\mathbf{k}} (d^j_{\mathbf{k}})^2$ représente l'énergie de la $j^{ème}$ sous-bande haute fréquence et υ_j un poids associé à cette sous-bande. Les poids υ_j peuvent être définis en fonction d'un modèle statistique choisi pour décrire les sous-bandes, permettant ainsi une meilleure adéquation aux données. Supposons par exemple que chaque sous-bande haute fréquence j suit un modèle de répartition gaussien $\mathcal{N}(0,\sigma_j)$ fixé. Le coût de description des J plus hautes fréquences peut alors s'exprimer de la façon suivante :

$$E_T = -\sum_{j=1}^{J} \sum_{\mathbf{k}} \log_2 \left\{ \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(d^j_{\mathbf{k}})^2}{2\sigma_j^2} \right) \right\},$$

Soit :

$$E_T = \sum_{j=1}^{J} \sum_{\mathbf{k}} \frac{(d^j_{\mathbf{k}})^2}{2\sigma_j^2} + constante,$$

qui est, à une constante près, le modèle défini par l'équation (2) en posant $\upsilon_j^2 = 1/(2\sigma_j^2)$.

Nous proposons ici de reformuler le modèle E_T. Pour un facteur d'échelle 2^J donné, nous notons T_J l'approximation de T obtenue en soustrayant au signal ses J plus hautes fréquences :

$$T_J(u,v) = \sum_{\mathbf{k}} c^J_{\mathbf{k}} \varphi^J_{\mathbf{k}}(u,v) \tag{3}$$

En remplaçant dans l'équation (1), nous obtenons :

$$T(u,v) - T_J(u,v) = \sum_{j=1}^{J} \sum_{\mathbf{k}} d^J_{\mathbf{k}} \psi^J_{\mathbf{k}}(u,v).$$

Le théorème de Parseval fournit l'égalité suivante :

$$\sum_{(u,v)} (T(u,v) - T_J(u,v))^2 = \sum_{j=1}^{J} \sum_{\mathbf{k}} (d^j_{\mathbf{k}})^2.$$

L'énergie texture (2) peut alors se réécrire comme suit :

$$E_T = \sum_{j=1}^{J} \omega_j^2 \cdot \sum_{(u,v)} (T(u,v) - T_j(u,v))^2. \tag{4}$$

On peut montrer que les poids ω_j^2 sont liés aux poids υ_j^2 par la relation suivante :

$$\upsilon_j^2 = \sum_{i=j}^{J} \omega_i^2 \Rightarrow \omega_j^2 = \upsilon_j^2 - \upsilon_{j+1}^2$$

Ainsi, le modèle n'est valable que si une contrainte de décroissance des poids à travers les sous-bandes est imposée : $\forall j, \upsilon_j^2 > \upsilon_{j+1}^2$. Cette contrainte est respectée par le modèle gaussien dans l'hypothèse, vérifiée en pratique, que $\forall j, \sigma_{j+1} > \sigma_j$.

3.3 Minimisation Energétique

Nous avons introduit dans le paragraphe précédent le coût de description texture E_T. Nous proposons maintenant une approche permettant de rechercher les paramètres $\{P_i\}_i$ qui minimisent ce coût.
E_T peut être réécrite en incluant à la formule le modèle géométrique :

$$E_T = \sum_{j=1}^{J} \omega_j^2 \cdot \sum_{(u,v)} (I(\tau_{TI}(u,v)) - T_j(u,v))^2 \tag{5}$$

La minimisation de l'énergie écrite sous cette forme est un problème sous déterminé, car à la fois la déformation τ_{TI} et les textures T_j sont inconnues. Pour la mettre en oeuvre, nous formulons donc cette minimisation comme un problème d'optimisation sous-contrainte où le meilleur couple (τ_{TI}, T) est recherché. Nous proposons alors un algorithme itératif : à chaque itération n, τ_{TI} est mis à jour en fixant T (donc chaque T_j) ; puis T est mise à jour en fixant τ_{TI}.

Après avoir fixé les textures $T_j^{(n)}$, la déformation est mise à jour en minimisant l'énergie suivante :

$$E_T^{(n+1)} = \sum_{j=1}^{J} \omega_j^2 \cdot \sum_{(u,v)} (I(\tau_{TI}^{(n+1)}(u,v)) - T_j^{(n)}(u,v))^2, \tag{6}$$

où $\tau_{TI}^{(n+1)}$ est la nouvelle déformation.
On montre que minimiser $E_T^{(n+1)}$ revient à minimiser l'énergie suivante :

$$\tilde{E}_T^{(n+1)} = \sum_{(u,v)} (I(\tau_{TI}^{(n+1)}(u,v)) - T_{cible}^{(n)}(u,v))^2,$$

avec

$$T_{cible}^{(n)}(u,v) = \frac{\sum_j \omega_j^2 \cdot T_j^{(n)}(u,v)}{\sum_j \omega_j^2}.$$

Cette modification apporte un gain en complexité : chaque étape de la minimisation revient désormais à minimiser une *Différence d'Image Déplacée* (DID) entre l'image originale et une texture cible courante. Par comparaison, l'équation précédente (6) exigeait de minimiser une DID entre l'image et *chaque* approximation $T_j^{(n)}$ à chaque itération.
Connaissant $\tau_{TI}^{(n+1)}$, les textures $T_j^{(n+1)}$ sont déterminées en calculant la texture $T^{(n+1)} = I(\tau_{TI}^{(n+1)})$.
Notons enfin que l'algorithme est initialisé en posant $T^{(0)} = I$ et $\tau_{TI}^{(0)} = Id$.
La dernière section montre les résultats obtenus à l'issue de l'analyse puis les résultats obtenus par le schéma global dans une application de compression.

4 Résultats

4.1 Résultats d'Analyse

Les résultats présentés ici ont été obtenus en appliquant l'algorithme d'analyse sur l'image *Lena* de dimension 256x256. L'ondelette de Daubechies 9/7 a été choisie pour représenter la texture et quatre niveaux de décomposition ont été considérés pour le calcul du coût de description ($L = 4$). Une taille de maille 16x16 a été choisie pour le maillage $\mathcal{M}$ initial. Par ailleurs, le déplacement des noeuds sur le bord du maillage a été contraint à la frontière du domaine $\mathcal{D}_I$.
La Figure 5 montre l'évolution des énergies E_j dans les sous-bandes hautes fréquences j de la texture au cours des itérations, ainsi que l'énergie totale. Les facteurs de multiplication $E_j^{(n)}/E_j^{(0)}$ sont représentés. On observe un déplacement de l'énergie vers les basses fréquences : l'énergie des deux sous-bandes de plus hautes fréquences diminuent sensiblement, tandis que l'énergie de la sous-bande $j = 4$ augmente. L'énergie totale, incluant la basse fréquence, reste constante. Une représentation de l'évolution des variances donne un graphique semblable.

Ceci témoigne de l'efficacité de l'analyse et des hypothèses avancées. On peut observer, enfin, qu'une vingtaine d'itérations suffisent pour faire converger l'énergie de la sous-bande de plus haute fréquence.

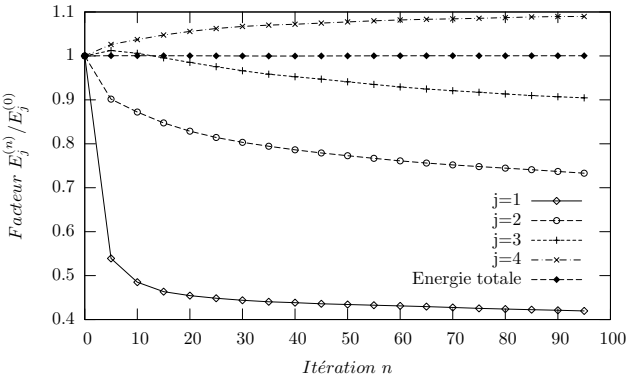


Figure 5 – *Evolution des énergies dans les sous-bandes hautes fréquences au cours des itérations. Image Lena.*

La Figure 6 montre le résultat visuel de l'analyse sur *Lena*. Sur la droite, l'image originale est représentée dans son domaine $\mathcal{D}_I$. Le maillage obtenu à l'issue de l'analyse lui est superposé. Nous observons que les mailles sont restées quadrangulaires mais se sont déplacées vers les contours. Dans les régions texturées comme les plumes, les déplacements sont négligeables, signifiant qu'aucun contour n'a été détecté.
Sur la gauche, la texture obtenue est représentée. Nous observons que dans le nouveau domaine $\mathcal{D}_T$, certains contours ont été alignés sur l'axe horizontal ou vertical, comme les cheveux au-dessus de l'épaule, d'autres ont été alignés et étirés comme le bas de l'épaule.

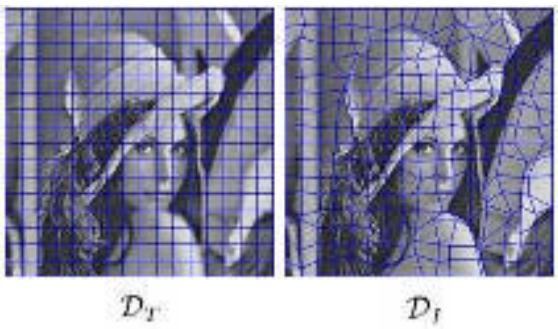


Figure 6 – *Texture et maillage obtenus à l'issue de l'analyse.*

4.2 Résultats de Compression

Dans un cadre de compression, les méthodes dites adaptatives ont en commun de devoir transmettre une information supplémentaire pour reconstruire au décodage les noyaux utilisés au codage. Cette information est ici le maillage. L'utilisation d'un maillage régulier permet de limiter le poids de cette information dans le flux car seules les positions des sommets doivent être transmises. Une information de connectivité s'avère en effet coûteuse. Les positions sont ainsi décomposées dans la base d'ondelettes de Daubechies 9/7, puis quantifiées et codées en utilisant un codage en plans de bits et un codeur arithmétique contextuel. L'information de texture est quant à elle codée en utilisant le logiciel JPEG2000 VM8.0 avec ses options par défaut (soit une décomposition sur 5 niveaux d'ondelettes de Daubechies 9/7). Notons que les deux flux encodés sont entièrement scalables et peuvent être décodés séparément. Décoder la géométrie avec une légère perte n'a qu'un faible impact visuel. Ceci permet en outre d'octroyer plus de débit au décodage de la texture, aboutissant à une meilleure qualité visuelle subjective à bas débits.

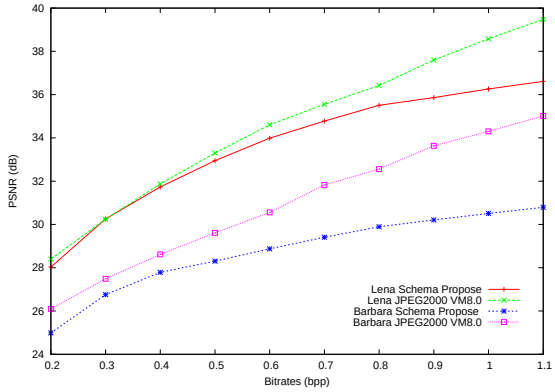


Figure 7 – *Résultats de compression en termes de PSNR sur Lena et Barbara.*

La Figure 7 montre le résultat du schéma de codage en terme de PSNR, pour les images *Lena* et *Barbara* 256x256. Les paramètres de la phase d'analyse sont les mêmes que précédemment, excepté que toutes les sous-bandes de haute fréquence sont considérées, soit $L = 8$. La géométrie est ici décodée sans perte et son coût est inclus dans les courbes. Il est proche de 0.05 bpp pour les deux images. Les courbes de PSNR sont comparées à celles obtenues avec JPEG2000 VM8.0. Tous les points des courbes ont été obtenus à partir des mêmes flux tronqués à différents débits.
Nous observons que les courbes de JPEG2000 sont au-dessus à tous les débits. A bas débits, les courbes restent proches, mais à hauts débits (autour de 1 bpp), l'écart atteint 2 dB pour *Lena* et 3 dB pour *Barbara*. La raison principale expliquant cette dégradation dans les hauts débits est le phénomène d'aliasing et/ou la perte de hautes fréquences intervenant lors des déformations directes et inverses. Ces distorsions, bien que n'ayant qu'un faible impact visuel, font chuter le PSNR. Nous pouvons cependant noter que le choix du PSNR comme mesure de qualité objective est inadapté à la méthode proposée. Nous obervons en effet

qu'une légère perte lors de la transmission des positions $\{P_i\}_i$ a un effet catastrophique sur le PSNR par rapport à une transmission sans perte. Visuellement, cette perte n'est pourtant pas détectable.
La Figure 8 présente le comparatif visuel avec JPEG2000 pour l'image *Lena* à 0.3bpp. Le premier zoom sur l'épaule, les lèvres et les cheveux montre le gain visuel apporté par le nouveau schéma. L'effet rebond autour des contours est sensiblement atténué par rapport à JPEG2000, menant à un meilleure qualité subjective générale. Le second zoom montre à la fois le gain et la limite de l'approche : les rebonds ont été réduits le long du contour formé par le chapeau, mais d'autre part, des hautes fréquences ont été perdues au niveau du ruban.

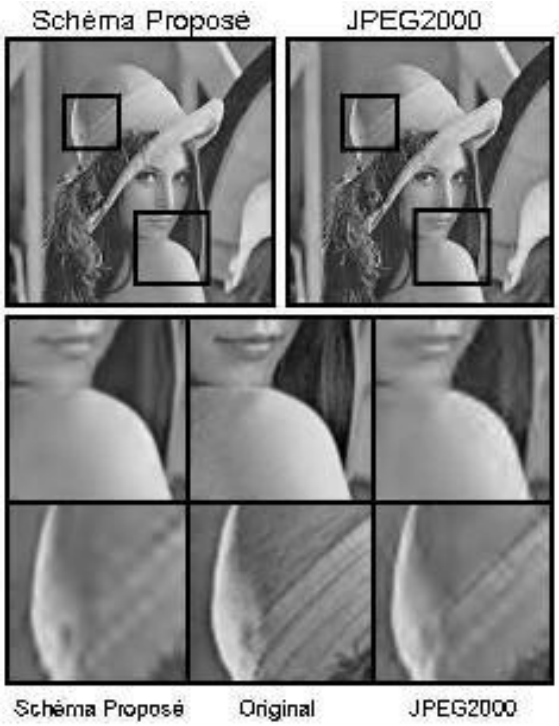


Figure 8 – *Résultat de compression visuel sur Lena à 0.3 bpp.*

5 Conclusion

Nous avons proposé une nouvelle méthode adaptative de déformation d'ondelettes pour la représentation d'images. Le choix d'une modélisation globale de la structure géométrique par un maillage déformable permet d'une part d'assurer la continuité des déformations et d'autre part de conserver la propriété de multi-résolution des ondelettes. Une formulation énergétique de la recherche des paramètres du modèle a été proposée et des résultats d'analyse et de compression ont été présentés.
Plusieurs orientations se dessinent pour des recherches futures. Tout d'abord, nous remarquons que la contrainte de régularité imposée au modèle géométrique limite le degré d'adaptation dans certaines régions. Par exemple, deux contours proches peuvent être mis en concurrence lors de l'analyse, empêchant les mailles de converger librement vers l'un ou l'autre de ces contours. Il serait donc intéressant de réfléchir à un modèle plus souple permettant de mieux représenter les structures complexes. Ensuite, les distorsions provoquées par l'aller-retour entre $\mathcal{D}_T$ et $\mathcal{D}_I$ restent un problème ouvert. Augmenter la résolution de la texture pourrait offrir une solution simple dans les hauts débits. Enfin, l'adaptation à la vidéo des concepts introduits est un axe de recherche important motivé par les résultats obtenus pour l'image fixe.

Références

[1] E. Le Pennec et S. Mallat. Sparse Geometric Image Representations with Bandelets. *IEEE Transaction on Image Processing*, 14(4) :423–438, Avril 2005.

[2] M. N. Do et M. Vetterli. The Contourlet Transform : An Efficient Directional Multiresolution Image Representation. *IEEE Transactions on Image Processing*, 14(12) :2091–2106, Décembre 2005.

[3] E. J. Candès et D. L. Donoho. Ridgelets : a key to higher-dimensional intermittency ? *Roy Soc of London Phil Tr A*, 357(1760) :2495–2509, Septembre 1999.

[4] E. J. Candès et D. L. Donoho. *Curvelets - A Surprisingly Effective Nonadaptive Representation for Objects with Edges*. Vanderbilt University Press, Nashville, TN, 1999.

[5] G. Peyré et S. Mallat. Discrete bandelets with geometric orthogonal filters. Dans *IEEE International Conference on Image Processing*, pages 65–68, Atlanta, GA, USA, Septembre 2006.

[6] W. Ding, F. Wu, et S. Li. Lifting-based Wavelet Transform with Directionally Spatial Prediction. Dans *Picture Coding Symposium*, San Francisco, USA, Décembre 2004.

[7] V. Chappelier, C. Guillemot, et S. Marinkovic. Codage d'images par ondelettes unidimensionnelles orientées. Dans *Actes de la conférence CORESA*, pages 117–120, Lille, Mai 2004.

[8] D. Terzopoulos et M. Vasilescu. Sampling and Reconstruction with Adaptive Meshes. Dans *IEEE Computer Vision and Pattern Recognition Conference*, pages 70–75, Lahaina, HI, 1991.

[9] M. Jansen, H. Choi, S. Lavu, et R. Baraniuk. Multiscale Image Processing Using Normal Triangulated Meshes. Dans *IEEE International Conference on Image Processing*, Thessaloniki, Greece, Octobre 2001.

[10] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, et M. Desbrun. Anisotropic polygonal remeshing. *ACM Transactions on Graphics*, 22(3) :485–493, 2003.

[11] O. Lee et Y. Wang. Non-uniform image sampling and interpolation over deformed meshes and its hierarchical extension. Dans *SPIE Visual Communications and Image Processing*, pages 389–400, Taipei, Taiwan, Mai 1995.

[12] Y. Wang et O. Lee. Use of 2D deformable mesh structures for video compression. Part I — The synthesis problem : Mesh based function approximation and mapping. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(6) :636–646, Décembre 1996.

Une image couleur cachée dans une image en niveaux de gris

M. Chaumont[1,2] W. Puech[1,2]

[1] Laboratoire LIRMM, UMR CNRS 5506,
Université Montpellier II - France

[2] Centre Universitaire de Formation et de Recherche de Nîmes - France

william.puech@lirmm.fr, marc.chaumont@lirmm.fr

Résumé

Dans cet article, nous proposons une méthode originale d'insertion des informations couleur d'une image dans l'image en niveaux de gris correspondante. L'objectif de ce travail est de mettre en place une base de données images dont les images en niveaux de gris comprimées sont accessibles librement et dont la reconstruction de l'image couleur n'est possible qu'avec l'utilisation d'une clé secrète. Cette méthode est composée de trois étapes importantes : la quantification couleur, l'ordonnancement des couleurs et l'insertion des données cachées basée DCT. La nouveauté de cet article concerne la construction d'une image d'index *associée à une palette couleur qui est également une image en niveaux de gris sémantiquement intelligible. Pour obtenir cette* image d'index *particulière, qui doit être robuste à l'insertion de données cachées, nous proposons un algorithme original d'ordonnancement des K couleurs :* l'algorithme de parcours en couche. *Enfin la méthode d'insertion repose sur une approche d'aqua-compression qui combine l'utilisation d'un codeur JPEG hybride permettant de comprimer les images dans un format standard du World Wide Web avec une fonctionnalité d'insertion de données cachées.*

Mots clefs

Insertion de données cachées, aqua-compression, quantification couleur.

1 Introduction

Actuellement peu de solutions sécurisées sont proposées pour donner à la fois un accès gratuit à des images de basse qualité et à la fois un accès sécurisé aux mêmes images de qualités supérieures. Nous proposons ici une solution à ce problème de sécurisation des bases de données images par l'intermédiaire d'une méthode d'insertion de données cachées. L'image peut être obtenue librement, mais sa visualisation en haute qualité exige l'utilisation d'une clé secrète. Plus précisément, dans notre solution l'image en niveaux gris comprimée en JPEG est librement accessible, mais seulement les possesseurs d'une clé secrète peuvent reconstruire l'image en couleur. Notre objectif est donc de protéger les informations couleur en incorporant ces informations dans l'image en niveaux de gris[1].

La méthode proposée est composée de trois grandes étapes : la quantification couleur (section 2.1), l'ordonnancement des couleurs (section 2.2) et l'insertion de données basée DCT (section 3). Dans l'étape de quantification couleur, le but est de trouver K couleurs et d'attribuer à chacun des pixels une de ces K couleurs. Dans l'étape d'ordonnancement, l'objectif est d'organiser ces K couleurs pour construire une palette de couleurs régulière chromatiquement et une *image d'index* sémantiquement intelligible. Dans l'étape d'insertion de données cachées basée DCT, le but est d'incorporer la palette de couleurs dans l'*image d'index* et être robuste à la compression JPEG.

Aucun travail similaire n'a été porté à la connaissance des auteurs. On peut citer par exemple Wu *et al.* qui proposent de construire une nouvelle palette pour incorporer un bit de message dans chaque couleur de la palette [1], mais ils n'incorporent pas la palette dans l'*image d'index*.

2 Quantification couleur et algorithme de parcours en couche

Dans cette section nous présentons la quantification couleur et les étapes d'ordonnancement des couleurs.

2.1 Quantification couleur

La réduction du nombre de couleurs d'une image couleur est un problème de quantification classique. La solution optimale, pour extraire les K couleurs, est obtenue en résolvant :

$$\{P_{\imath,k}, C(k)\} = \arg \min_{P_{\imath,k}, C(k)} \sum_{\imath=1}^{N} \sum_{k=1}^{K} P_{\imath,k}.dist^2(I(\imath), C(k)), \quad (1)$$

où I est une image couleur de dimension N pixels, $C(k)$ est la k-ième couleur parmi les K couleurs recherchées,

[1]Ce travail s'inscrit dans le cadre du projet TSAR 2005-2008 (Transfert Sécurisé d'images d'Art haute Résolution) retenu par l'ANR (Agence Nationale de la Recherche) dont l'objectif est de donner un accès sécurisé aux peintures numériques de la base de données EROS (European Research Open System) du C2RMF (Centre de Recherche et de Restauration des Musées de France, UMR CNRS), Paris.

$dist()$ est une fonction de distance dans l'espace couleur (L2 dans l'espace couleur RGB) et $P_{\imath,k} \subset \{0,1\}$ est la valeur d'appartenance du pixel $\imath$ à la couleur k.
Une solution pour minimiser l'équation (1) et ensuite obtenir les K couleurs, est d'utiliser l'algorithme ISODATA des k-moyens [2]. $P_{i,k}$ est définie telle que :

$$\forall i, \forall k, P_{i,k} = \begin{cases} 1 \text{ si } k = \underset{\{k'\}}{\arg}\{\min_{k' \in [1,K]} dist(I(i), C(k'))\}, \\ 0 \text{ sinon,} \end{cases} \tag{2}$$

avec $C(k) = \frac{\sum_{i=1}^{N} P_{i,k} \times I(i)}{\sum_{i=1}^{N} P_{i,k}}$.
Dans notre approche le nombre K est significatif en comparaison du nombre original de couleurs. Si nous utilisons l'algorithme classique des k-moyens, le nombre de couleurs extrait sera souvent en dessous de K. C'est le problème bien connu "de classes mortes". Pour surmonter ce problème, on initialise les valeurs $P_{\imath,k}$ en résolvant l'équation floue des k-moyens ci-dessous :

$$\{P_{\imath,k}, C(k)\} = \arg \min_{P_{\imath,k}, C(k)} \sum_{\imath=1}^{N} \sum_{k=1}^{K} P_{\imath,k}^{m} . dist^2(I(\imath), C(k)), \tag{3}$$

où m est le coefficient flou (m est positionné à 1.6 comme proposé dans [3]) et les $P_{\imath,k} \in [0,1]$ sont les coefficients d'appartenance flous. Cette équation est résolue par un algorithme k-moyens flou [4].

2.2 Algorithme de parcours en couche

Une fois que la quantification couleur a été traitée, l'image à K couleurs obtenue peut être représentée par une *image d'index* (grâce au valeurs des $P_{\imath,k}$) et une palette de couleurs (grâce aux valeurs $C(k)$). Notre but est alors de résoudre deux contraintes ; la première contrainte est d'obtenir une *image d'index* où chaque niveaux de gris est proche de la luminance de l'image couleur originale ; la deuxième contrainte consiste à obtenir une palette de couleurs dont les couleurs consécutives sont peu éloignées. Grâce à la quantification couleur, nous possédons déjà une *image d'index* et une palette de couleurs. Notre problème est alors de trouver une fonction de permutation qui permutent dans le même temps les valeurs de l'*image d'index* et les valeurs de la palette de couleurs. La fonction de permutation Φ est trouvée en résolvant l'équation :

$$\Phi = \arg\min_{\Phi} \left[\sum_{\imath=1}^{N} E_i^{ind} + \lambda \sum_{k=1}^{K-1} E_k^{palette} \right], \tag{4}$$

$$E_i^{ind} = dist^2(Y(\imath), \Phi(Index(\imath))), \tag{5}$$

$$E_k^{pal} = dist^2(Palette(\Phi^{-1}(k)), Palette(\Phi^{-1}(k+1)))], \tag{6}$$

où Y est la luminance de l'image couleur originale et λ est la valeur du Lagrangien. La fonction de permutation Φ est une fonction bijective dans $\mathbb{N}$ et défini tel que $\Phi : [1..K] \rightarrow [1..K]$.

Nous approchons l'optimum de l'équation (4) en utilisant un algorithme heuristique. Le but de cet algorithme est de trouver un ordonnancement pour les K couleurs tel que les couleurs consécutives soient peu éloignées et tel que la luminance des couleurs soient ordonnées des plus sombres aux plus claires. Cette ordonnancement définit pour chaque k-ième couleur une position k' qui nous donne la fonction Φ tel que $\Phi(k) = k'$.

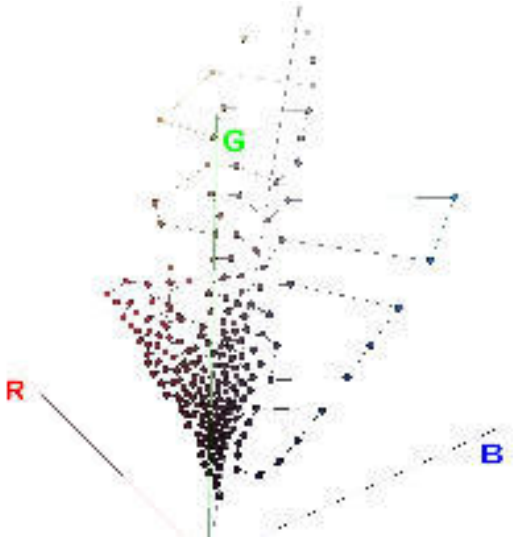


Figure 1 – *Vue du parcours en couche dans le cube RGB.*

Pour trouver un ordonnancement des K couleurs, l'algorithme parcours l'espace des couleurs pour construire la suite ordonnée de couleurs. La figure 1 illustre le chemin obtenu après un parcours dans le cube RGB. Ce parcours est effectué en "sautant" de couleur en couleur, dans l'espace couleur, en choisissant la couleur plus proche de la couleur courante. La première couleur de cette suite est choisie comme étant la couleur la plus sombre parmi les K couleurs. Une contrainte supplémentaire à ce parcours consiste à limiter la recherche de couleur aux couleurs peu éloignées en luminance. Cela signifie que le parcours dans l'espace des couleurs est limitée à une fenêtre définie sur les informations de luminance. Cette *algorithme de parcours en couche* peut être vu comme un "parcours 3D en spirale" dans l'espace des couleurs.

3 Insertion de la palette de couleurs

3.1 Choix du nombre de couleurs

Dans la section précédente nous avons présenté la méthode utilisée pour construire une *image d'index* (peu éloigné de la luminance de l'image couleur originale) et une palette de couleurs (dont les couleurs consécutives sont proches). Le nombre de couleurs K était supposé connu. Dans cette section nous présentons une solution empirique pour choisir le nombre K de couleurs. On pourrait choisir un nombre de couleurs égal à 256 mais ce nombre n'est pas adapté pour

construire une *image d'index* semblable à l'image de luminance. En effet, les 256 valeurs d'index sont couvertes dans l'*image d'index* alors qu'il y a souvent beaucoup moins de niveaux de gris dans l'image de luminance. Une solution plus intelligente est de choisir un nombre de couleurs égal à l'intervalle de niveaux de gris *significatif* de l'image de luminance. L'énergie de l'équation (4) doit être modifiée afin d'exprimer la réduction du nombre de couleurs. Seul le premier terme est changé :

$$E_i^{ind} = dist^2(Y(\imath), t + \Phi(Index(\imath))), \qquad (7)$$

où t est une valeur de translation.
Pour choisir le nombre de couleurs à partir de l'histogramme de luminance nous définissons un *seuil significatif* correspondant à 1% de la valeur maximale de l'histogramme. Les valeurs des niveaux de gris inférieures à ce seuil sont considérées comme *non significatives*. Un *intervalle significatif* de niveaux de gris est alors défini tel que : la borne inférieure de cet intervalle est le premier niveau de gris *significatif*, et la borne supérieure de cet intervalle est le dernier niveau gris *significatif*. La largeur de l'*intervalle significatif* correspond au nombre K de couleurs et la borne inférieure est la valeur t de translation.
Remarquons que choisir un nombre de couleurs égal à la largeur de l'*intervalle significatif* réduit l'intervalle des index. Il en résulte une *image d'index* moins contrastée comparée à celle obtenue avec $K = 256$. L'*image d'index* est alors visuellement meilleure ; son PSNR est également amélioré par rapport à l'image de luminance. Remarquons également qu'attribuer à t la valeur de la borne inférieure de l'intervalle n'est pas nécessairement la meilleure solution. Cependant, en considérant que l'histogramme de l'*image d'index* est pratiquement plat, la valeur de t qui minimise au mieux le premier terme d'énergie de l'équation (7), est nécessairement proche de la valeur de la borne inférieure de l'*intervalle significatif*.

3.2 Méthode utilisée pour l'insertion de données cachées basée DCT

Les méthodes d'insertion de données cachées peuvent être utilisées pour faire de la transmission d'image sécurisée. Pour les applications traitant des images, l'objectif est d'insérer de manière invisible un message ou une marque à l'intérieur de l'image. L'insertion de données cachées est alors effectuée de manière différente selon la longueur du message et la robustesse désirée [5, 6, 7]. On définit généralement deux groupes de méthodes d'insertion de données cachées relativement au domaine d'insertion : les méthodes d'insertion dans le domaine spatial [8, 9, 10] et les méthodes d'insertion dans le domaine fréquentiel [11, 12, 13]. Nous nous intéressons dans cet article à une méthode d'insertion similaire à celle de [14] et qui combine les deux domaines spatial et fréquentiel pour effectuer l'insertion. Nous proposons de plus une solution d'**aqua-compression** c'est-à-dire une solution permettant de faire de manière conjointe une insertion de données cachées et une compression.
Dans cette section, nous décrivons dans un premier temps un codeur JPEG hybride avec une méthode d'insertion de données cachées dans le domaine fréquentiel. La méthode d'insertion de données cachées est effectuée après une transformation DCT. Chaque bit $b_\imath$, d'un message $M = b_1 b_2 ... b_m$ composé de m bits, est inséré dans le coefficient **DC** d'un bloc DCT [15]. Le processus d'insertion s'effectue en substituant le bit de poids faible (*Least Significant Bit*) du coefficient de DC par le bit $b_\imath$ à insérer.
Avant insertion du message, nous calculons un facteur d'insertion (fonction de la longueur du message et de la taille de l'image) indiquant le nombre de bits à insérer par pixels de l'image. Le facteur d'insertion, en $bits/pixel$ est :

$$E_f = m/N. \qquad (8)$$

L'*image d'index* est alors divisée en régions de taille $\lfloor 1/E_f \rfloor$ pixels. Comme nous utilisons la composante DC pour insérer un bit du message, notons que la taille minimale de l'image (en pixel) doit être au minimum égale à 64 fois le nombre de bits du message M à insérer ($N > 64m$). Chaque région est alors utilisée pour cacher **un seul** bit $b_\imath$ du message. Ce bit est caché dans la composante DC de l'**un** des blocs appartenant à la région. Cette procédure de partitionnement garantie une répartition homogène du message sur toute l'image.
L'objectif est donc d'insérer le message M représentant la palette de couleurs. Pour cacher la palette de couleurs dans l'image nous devons donc insérer dans l'*image d'index* $m = 3 \times K \times 8$ bits plus un en-tête précisant les valeurs de K et de t. Par conséquent, le facteur d'insertion E_f, équation (8) est égal à :

$$E_f = (3 \times 8 \times K + 2 \times 8)/N. \qquad (9)$$

Soit un bloc carré composé de n^2 pixels d'une image I, à partir de la DCT, le coefficient continu $F(0,0)$ du bloc est :

$$F(0,0) = \frac{1}{n} \sum_{\imath=0}^{n^2-1} I(\imath). \qquad (10)$$

Dans la compression JPEG avec perte, le coefficient DC est quantifié et donne un coefficient quantifié $F'(0,0)$:

$$F'(0,0) = [F(0,0)/Q(0,0)], \qquad (11)$$

où le $[.]$ est la fonction retournant le nombre entier le plus proche et $Q(0,0)$ est le coefficient de quantification.
Une solution classique pour insérer le message est de remplacer $F(0,0)$ par $F_w(0,0)$. Cette substitution prend donc en compte l'étape de quantification du codeur JPEG tel que :

$$F_w(0,0) = \begin{cases} \lfloor \frac{F(0,0)}{Q(0,0)} \rfloor \times Q(0,0) & \text{si } \lfloor \frac{F(0,0)}{Q(0,0)} \rfloor mod\ 2 = b_\imath, \\ \lceil \frac{F(0,0)}{Q(0,0)} \rceil \times Q(0,0) & \text{si } \lceil \frac{F(0,0)}{Q(0,0)} \rceil mod\ 2 = b_\imath, \end{cases} \qquad (12)$$

où $\lfloor . \rfloor$ est la partie entière d'un nombre et $\lceil . \rceil$ est la fonction retournant le nombre entier supérieur ou égal le plus proche. Notons que la substitution de $F(0,0)$ par $F_w(0,0)$ est effectué avant l'étape de quantification. Remarquons également que $F_w(0,0)$ est un nombre entier.
Nous proposons maintenant d'améliorer la méthode d'insertion précédente. En effet, la modification du coefficient de DC ne prend pas en compte les informations spatiales du bloc correspondant. La modification du coefficient DC peut entraîner une gêne visuelle. Pour améliorer le résultat visuel, la modification du coefficient DC est obtenue par modification des niveaux de gris d'un certain nombre de pixels appartenant au bloc correspondant. Les pixels modifiés sont les pixels possédant la plus forte variance. Ainsi, lors de l'insertion, au lieu de modifier le coefficient $F(0,0)$ en effectuant une insertion dans le domaine fréquentiel, nous modifions n_w pixels du bloc correspondant de sorte qu'après la DCT, on obtienne la valeur adéquate pour $F_w(0,0)$. L'insertion est alors faite dans le domaine spatial avec prise en compte de l'impact fréquentiel et de quantification JPEG. Les n_w pixels $I(\imath)$ sont modifiés pour obtenir de nouveaux pixels $I_w(\imath)$ tel que :

$$I_w(\imath) = I(\imath) - sign(F(0,0) - F_w(0,0)), \qquad (13)$$

où $sign(x) = -1$ si $x < 0$ et $sign(x) = 1$ si $x \geq 0$. Notons que le nombre de pixels à modifier n_w vaut :

$$n_w = [|F(0,0) - F_w(0,0)| \times n]. \qquad (14)$$

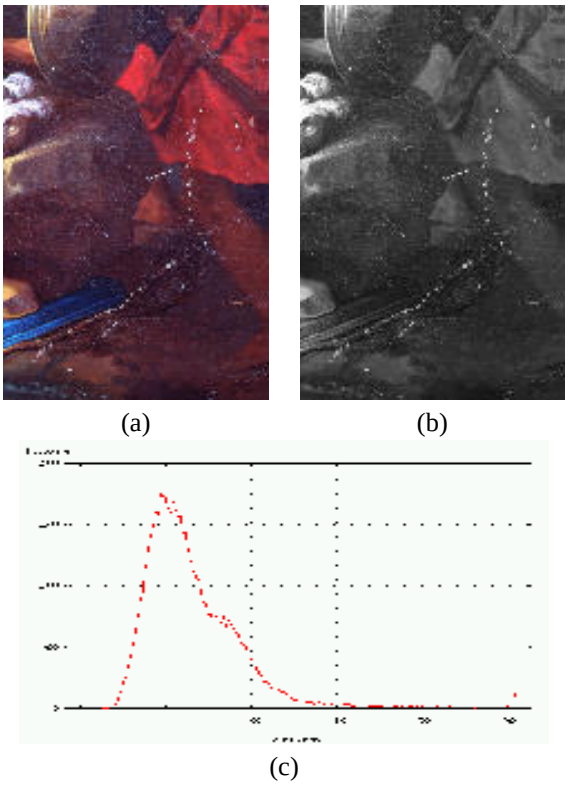
Remarquons également que lorsque $n_w > n^2$, nous appliquons une première fois l'équation (13) sur tous les pixels du bloc et nous répétons de nouveau cette opération sur $n_w \ mod \ n^2$ pixels.
Pour résumer notre méthode d'insertion (qui ajoute la fonctionnalité d'insertion de données cachées au codeur JPEG), l'équation d'un coefficient DC quantifié marqué est :

$$F'_w(0,0) = \frac{1}{n \times Q(0,0)} \Big(\sum_{i \in \Omega_w} I_w(\imath) + \sum_{i \in \overline{\Omega_w}} I(\imath) \Big), \qquad (15)$$

où Ω_w est l'ensemble des n_w pixels modifiés d'un bloc.

4 Résultats

Nous avons appliqué notre méthode sur le détail d'une peinture numérique illustré figure 2.a issu de la base de données EROS. Ce détail, de taille 523×778 pixels, du C2RMF provient d'une peinture représentant Saint Jean-Philippe baptisant l'eunuque de la Reine Candace (conservé au musée du Louvre, inventaire INV 2536). L'image de luminance est illustrée figure 2.b et son histogramme est représenté figure 2.c.
Nous pouvons observer à partir de l'histogramme de luminance qu'un grand nombre de niveaux de gris sont non *significatifs*. Pour obtenir le nombre de couleurs K et la valeur de translation t, un seuillage automatique est réalisé et donne un intervalle de niveaux de gris *significatif* de $[20, 222]$ comme présenté section 3.1. Le nombre de couleurs et la translation automatiquement déduits sont : $K = 203$ et $t = 20$. Choisir un nombre de couleurs égal à la taille de l'intervalle de niveaux de gris garantit une forte réduction du premier terme de l'énergie de l'équation (7) sans forte augmentation du deuxième terme d'énergie de l'équation (6) et donc sans forte augmentation de la distorsion sur l'image couleur après insertion de données cachées.
En appliquant l'algorithme de quantification couleur présenté section 2.1 nous obtenons les images quantifiées figure 3.c pour $K = 254$ et figure 3.d pour $K = 203$. Pour ces deux images quantifiées, aucune différence visuelle ne peut être remarquées entre elles. En comparant les images d'index correspondantes (figure 3.a pour $K = 254$ et figure 3.b pour $K = 203$) nous pouvons constater visuellement que la réduction du nombre d'index permet d'obtenir une image en niveaux de gris plus plaisante visuellement (moins contrastée) pour l'image avec $K = 203$ couleurs.



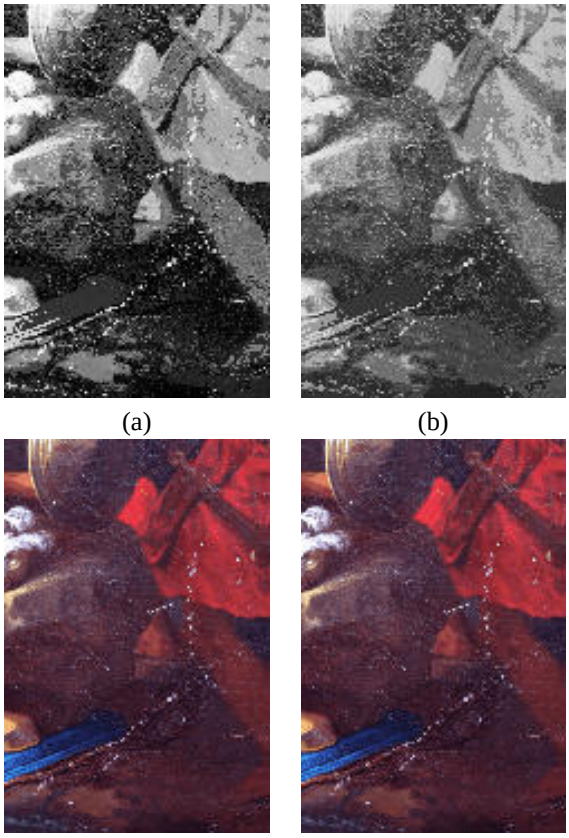
(a) (b)

(c)

Figure 2 – *a) Image couleur originale, b) Luminance de l'image couleur originale, c) Histogramme de la luminance.*

Nous détaillons maintenant l'algorithme proposé en utilisant $K = 203$. Une fois que la quantification avec $K = 203$ couleurs a été réalisée, une palette de couleurs et son *image d'index* sont obtenus. Les figures 4.a et 4.c. illustrent le résultat classique obtenu pour la phase de quantification couleur. On peut remarquer que l'*image d'index* ne permet pas de comprendre ni même d'identifier son contenu. Avec l'application de l'*algorithme de par-*

cours en couche, nous obtenons une palette de couleurs ordonnée présentée figure 4.b) et une *image d'index* (figure 4.d) sémantiquement intelligible. Notons que *l'algorithme de parcours en couche* ne change pas le contenu informationnel. En effet, la palette de couleurs et l'*image d'index* permettent de reconstruire la même image couleur avant et après déroulement de *l'algorithme de parcours en couche*.

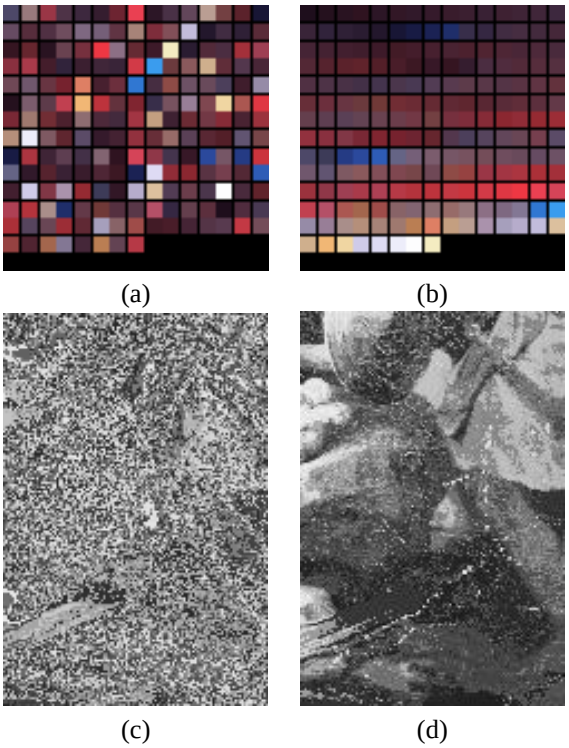


(a) (b)

(c) (d)

Figure 3 – *Comparaison entre* $K = 254$ *couleurs et* $K = 203$ *couleurs. a)* L'image d'index *après l'ordonnancement des couleurs avec* $K = 254$*, b)* L'image d'index *après l'ordonnancement des couleurs avec* $K = 203$*, c) Image quantifiée avec* $K = 254$ *couleurs, d) Image quantifiée avec* $K = 203$ *couleurs.*

Nous détaillons maintenant la phase d'insertion des données. Avant l'insertion, le message à insérer (la palette de couleurs) est codé prédictivement puis arithmétiquement. La longueur du message diminue de 4888 bits à 3183 bits. Le facteur d'insertion est alors $E_f = 0.0078$ bits/pixel. L'*image d'index* est donc partitionnée en régions de $\lfloor 1/E_f \rfloor = 128$ pixels et un bit du message est inséré dans un bloc appartenant à une région. Pour répartir le message sur l'image avec une distribution dépendant de la clé, nous utilisons une clé secrète de 128 bits comme "germe" pour le générateur de nombres pseudo aléatoires. Cette clé secrète est également utilisée pour crypter la palette de couleurs avant l'insertion. L'image est alors comprimée avec notre codeur hybride JPEG (compression + insertion de données cachées) avec un facteur de qualité de 100%.



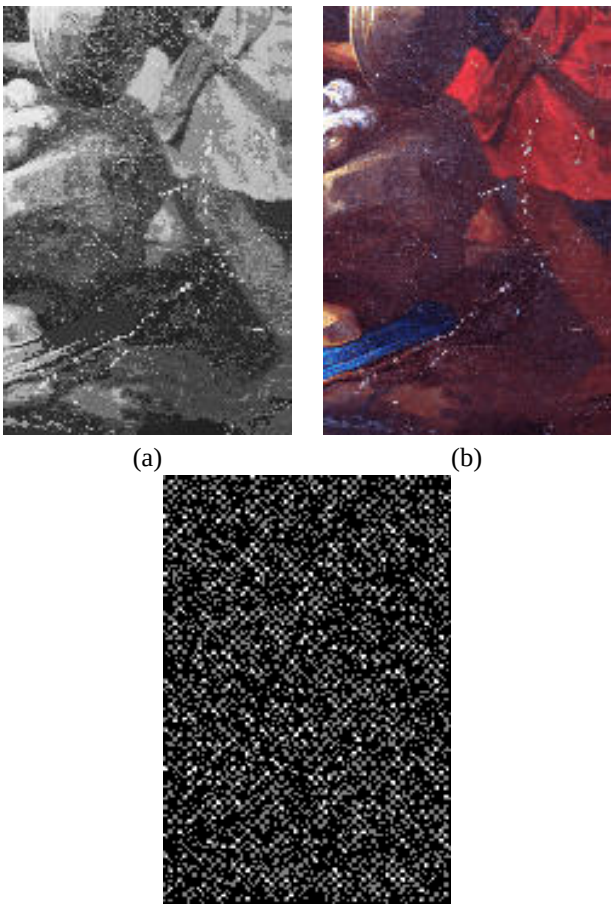
(a) (b)

(c) (d)

Figure 4 – *Application de* l'algorithme de parcours en couche. *a) La palette de couleurs de l'image originale quantifiée (*$K = 203$ *couleurs), b) La palette de couleurs après l'ordonnancement des couleurs, c)* L'image d'index *de l'image quantifiée, d)* L'image d'index *après l'ordonnancement des couleurs.*

Une fois que notre codeur JPEG hybride (compression + insertion de la palette couleur) a encodé l'*image d'index*, il est possible avec un décodeur JPEG classique et sans clé d'accéder librement à l'image JPEG en niveaux de gris représentée figure 5.a. Avec la clé secrète, la palette de couleurs est extraite et l'image couleur est reconstruite. La figure 5.b montre l'image couleur reconstruite à partir de l'*image d'index* marquée. On peut observer que la qualité de l'image est très bonne. La valeur du PSNR entre l'image quantifiée en $K = 203$ couleurs et l'image couleur reconstruite à partir de l'*image d'index*-marquée de $41.2\ dB$, confirme cette évaluation subjective. La méthode d'insertion de données cachées utilisée et la proximité des couleurs consécutives dans la palette de couleurs sont à l'origine de ce bon résultat. La figure 5.c montre l'image de différence calculée entre l'*image d'index* et l'image reconstruite à partir de l'*image d'index-marquée-comprimée*. On peut remarquer que la modification de pixels se produit sur l'ensemble de l'image. On peut également remarquer que même si la modification de l'*image d'index* est dense,

l'image reconstruite est toujours satisfaisante visuellement.



(a) (b)

(c)

Figure 5 – *Insertion de données cachées basée DCT. a)* Image d'index *marquée avec* $K = 203$ *et* $m = 3283$ *bits, b) Image couleur reconstruite à partir de l'*image d'index *marquée, c) Image de différence entre l'*image d'index *et l'*image d'index *marquée.*

5 Conclusion

Dans cet article, nous avons proposé une méthode pour cacher les informations couleur dans une image en niveaux de gris comprimée. Cette méthode est composée de trois étapes importantes qui sont la quantification couleur, l'ordonnancement des couleurs et l'insertion de données cachées. L'originalité de cet article est de construire une *image d'index* qui est une image en niveaux de gris intelligible sémantiquement. Pour obtenir cette *image d'index* particulière, un algorithme original d'ordonnancement en K couleurs est proposé : *l'algorithme de parcours en couche*. Un codeur JPEG hybride permet d'insérer la palette de couleurs au sein de l'*image d'index*. Ce processus d'insertion de données cachées permet de comprimer les images avec un format standard du Word Wide Web et donne une solution prête-à-l'emploi pour publier de manière sécurisée les peintures numériques de la base de données EROS du C2RMF.

Remerciements

Nous remercions M. Lahanier Christian, chef du département Documentation du C2RMF (Centre de Recherche et de Restauration des Musées de France) pour nous avoir donné un accès aux peintures numériques de la base de données EROS et également pour sa participation aux discussions du groupe de travail.

Références

[1] M.-Y. Wu, Y.-K. Ho, et J.-H. Lee. An Iterative Method of Palette-Based Image Steganography. *Pattern Recognition Letters*, 25 :301–309, 2003.

[2] G. H. Ball et D. J. Hall. ISODATA, a novel method of data analysis and pattern classification. Dans *In Proceedings of the International Communication Conference*, Juin 1966.

[3] R. Castagno et A. Sodomaco. Estimation of image feature reliability for an interactive video segmentation scheme. Dans *International Conference on Image Processing, ICIP'1998*, volume 1, pages 938–942, Chicago, USA, Octobre 1998.

[4] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3 :32–57, 1974.

[5] J.F. Delaigle, C. De Vleeschouwer, et B. Macq. Watermarking algorithm based on a human visual model. *Special Issue on Watermarking, Signal Processing*, 66(3) :319–336, 1998.

[6] Z. Duric. *Information Hiding, Steganography and Watermarking - Attacks and Countermeasures*. Kluwer Academic Publishers, Boston, 2001.

[7] F.A.P. Petitcolas, R. J. Anderson, et M.G. Kuhn. Information Hiding-A Survey. *IEEE, special issue on protection of multimedia content*, 87(7) :1062–1078, July 1999.

[8] W. Bender, D. Gruhl, N. Morimoto, et A. Lu. Techniques for Data Hiding. *I.B.M. Systems Journal*, 35(3-4) :313–336, 1996.

[9] N. Nikolaidis et I. Pitas. Robust Image Watermarking in the Spatial Domain. *Signal Processing*, 66(3) :385–403, 1998.

[10] G. Jagpal. Steganography in Digital Images. Dans *Dissertation, University of Cambridge, Selwyn College*.

[11] A.G. Bors et I. Pitas. Image watermarking using block site selection and DCT domain constraints. *Optics Express*, 3(12) :512–522, 1998.

[12] C.-C. Chang, T.-S. Chen, et L.-Z. Chung. A Steganographic Method Based Upon JPEG and Quantization Table Modification. *Information Sciences, Elsevier*, 141 :123–138, 2002.

[13] H.-W. Tseng et C.-C. Chang. High Capacity Data Hiding in JPEG-Compressed Images. *Informatic, Institute of Mathematics and Informatic, Vilnius*, 151(1) :127–142, 2004.

[14] F. Y. Shih et S. Y.T. Wu. Combinational image watermarking in the spatial and frequency domains. *Pattern Recognition*, 36 :969–975, 2003.

[15] D. Upham. Jpeg-jsteg, Modification of the Independent Jpeg Group's Jpeg Software for 1-bit Steganography in Jfif Output Files. Dans *ftp ://ftp.funet.fi/pub/crypt/steganography*, 1997.

Gradients morphologiques de texture. Application à la segmentation couleur+texture par LPE

Jesús Angulo
Centre de Morphologie Mathématique - Ecole des Mines de Paris
35, rue Saint-Honoré, 77305 Fontainebleau cedex - France
jesus.angulo@ensmp.fr

Résumé

Cet article présente une approche morphologique pour le calcul de gradients de texture et illustre comment les utiliser pour la segmentation d'image selon la texture ; et plus généralement, pour la segmentation conjointe texture + couleur (e.g., segmentation structurelle). Le point de départ est une décomposition de l'image couleur en deux composantes : la couche des objets et la couche de texture. La couche des objets est l'image couleur obtenue par simplification de l'image originale, sur laquelle le gradient couleur est calculé. La couche de texture est obtenue comme le résidu des composantes de luminance des images originales et simplifiées. Une analyse multi-échelle locale de la couche de texture est construite avec des opérateurs morphologiques : ouvertures/fermetures et nivellements sur des FAS. Des gradients de texture sont définis sur cette analyse, qui sont combinés avec le gradient couleur pour construire des segmentations mixtes par LPE. Les partitions obtenues avec des gradients structurels sont, dans la plupart des cas, plus pertinentes que celles obtenues seulement avec des gradients couleur : les régions de texture sont mieux déterminées et la sur-segmentation des régions grandes et homogènes est réduite.

Mots clefs

morphologie mathématique, granulométrie, nivellement, segmentation couleur+texture, gradient couleur, gradient texture, ligne de partage des eaux, espace LSH.

1 Introduction

Le paradigme de segmentation morphologique est la Ligne de Partage des Eaux (LPE) avec des marqueurs imposés [12], qui a démontré être l'une des techniques les plus puissantes pour la segmentation. Des approches hiérarchiques basées sur la LPE ont permis d'aborder des domaines pour lesquels le choix de marqueurs n'est pas facile, comme c'est le cas des images naturelles, images de vidéo-surveillance, etc... Parmi ces approches nous pouvons en souligner deux : (1) l'algorithme de cascades [5], qui, d'un niveau de la hiérarchie au suivant, élimine les contours complètement entourés par des contours plus forts ; (2) les hiérarchies basées sur les valeurs d'extinction [19, 13], en particulier les critères volumiques, qui

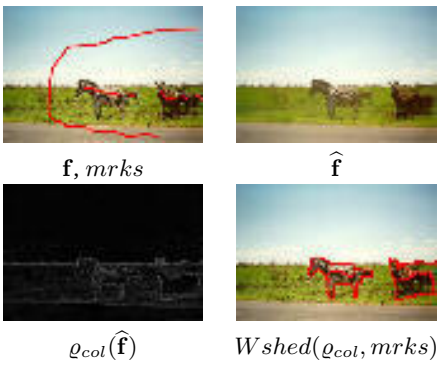


Figure 1 – *Exemple de segmentation d'une image couleur par LPE et marqueurs.*

combinent la taille et le contraste des régions, créant un bon critère pour évaluer la pertinence visuelle des régions. Ces algorithmes se bâtissent sur un gradient scalaire. Un gradient couleur doit être calculé pour appliquer la LPE à une image couleur. Au cours de nos travaux précédents [2], nous avons étudié en détail différentes définitions de gradient couleur et nous avons montré que le gradient complet dans une représentation luminance/saturation/teinte (LSH) [3] est assez performant et fournit des segmentations robustes et stables face aux changements d'éclairage. Plus précisément, si $\mathbf{f} = (f_L, f_S, f_H)$ correspond à une image couleur dans sa représentation LSH, son gradient couleur est donné par :

$$\varrho_{col}(\mathbf{f}) = (1 - f_S) \times \varrho(f_L) + f_S \times \varrho^{\circ}(f_H) + \varrho(f_S) \quad (1)$$

où $\varrho(g)$ est le gradient morphologique d'une image scalaire g et $\varrho^{\circ}(a)$ est le gradient circulaire centré d'une image sur le cercle unité (dans ce cas, la teinte).
Dans la manière classique de procéder pour segmenter une image par l'algorithme de la LPE, l'image couleur est préalablement filtrée grâce à un filtre connexe, type nivellement [14], $\lambda(m(f), f)^i = \left[f \wedge \delta^i(m)\right] \vee \varepsilon^i(m)$ jusqu'à l'idempotence $\lambda(m, f)^i = \lambda(m, f)^{i+1}$ (f est l'image référence et $m(f)$ est l'image marqueur), qui permet de simplifier les textures et d'éliminer les régions les moins significatives tout en préservant les contours des objets qui restent sur l'image. Pour les images couleur, nous pouvons appli-

quer un nivellement marginal à chaque composante R,V,B ou bien calculer un nivellement couleur [1]. En tout cas, le nivellement a besoin d'une image marqueur qui détermine les structures à préserver, i.e.,

$$\widehat{\mathbf{f}} = \lambda(ASF_{nB}(\mathbf{f}), \mathbf{f}),$$

où ASF_{nB} est un filtre alterné séquentiel de taille n et B est un élément structurant isotrope (d'autres filtres comme des Gaussiennes peuvent être utilisés pour construire le marqueur). Ensuite, la LPE est calculée sur le gradient couleur de l'image $\widehat{\mathbf{f}}$. L'exemple de la Fig. 1 illustre la segmentation avec un marqueur pour chaque objet d'intérêt, ici chacune des zébras (plus un marqueur pour l'extérieur). Cependant, dans ce cas, la couleur ne permet pas d'extraire correctement les contours des objets.

En effet, dans certaines images la texture est une information très discriminante pour la séparation des objets, même si l'introduire dans la segmentation n'est pas si simple que pour la couleur : la texture est une notion régionale qui n'est pas facile à quantifier. Dans [7], Hill *et al.* ont proposé une méthode pour construire un gradient de texture à partir d'une transformation par ondelettes, qui est ensuite utilisé avec la LPE pour segmenter les images à niveaux de gris. L'usage combiné de la couleur et de la texture est le sujet d'un certain nombre de travaux récents. Vanhamel *et al.* introduisent dans [21] une approche marginale pour appliquer des filtres de Gabor à chaque composante d'une image couleur et ainsi construire un espace de caractéristiques couleur/texture utilisé dans la segmentation. De manière similaire, Hoang *et al.* [8] utilisent des filtres de Gabor pour mesurer la texture-couleur et la segmentation est obtenue avec une classification par k-means. Les travaux de l'équipe Malik *et al.* [10] s'appuient aussi sur des banques de filtres gaussiens pour calculer un gradient de texture qui est ensuite combiné avec des gradients de brillance et de couleur dans un schéma d'apprentissage supervisé.

Plus proche de ce que nous faisons, Sofou *et al.* [18] introduisent une segmentation conjointe intensité/texture par LPE implémentée en forme d'EDP, où la texture est mesurée par démodulation d'une banque de filtres. Ce dernier travail part d'une décomposition de l'image selon le modèle $f = u + v$ de Y. Meyer [15], où u est la "cartoon component" (plateaux homogènes des objets) et v est la "texture oscillation". Ce modèle a été initialement étudié dans le cadre d'une approche variationelle par Vese et Osher [22]. Des travaux plus récents, par exemple Patwardhan et Sapiro [16] et Aujol *e*t al. [4], explorent des algorithmes variationels rapides pour le calcul des images u et v. Sofou *et al.* proposent d'obtenir la composante de texture comme le résidu du nivellement, i.e., $v = f - u = f - \lambda(m, f)$ (m le marqueur, étant une Gaussienne).

Dans cet article, nous nous plaçons dans un cadre similaire à celui de Sofou *et al.* [18], moins coûteux en termes computationnels et qui fournit aussi des décompositions valables pour l'objectif de segmentation. Notre point de départ est aussi une décomposition de l'image couleur $\mathbf{f}$ en deux composantes :

$$\mathbf{f} \approx \widehat{\mathbf{f}} \uplus f_{tex}, \tag{2}$$

où $\widehat{\mathbf{f}}$ est la *couche des objets* et f_{tex} est la *couche de texture*. Cette dernière est obtenue comme le résidu des composantes de luminance, i.e., $f_{tex} = f_L - \widehat{f_L}$, car les variations liées à la texture sont principalement associées à la luminance. Nous allons d'abord montrer comment nous pouvons construire une analyse multi-échelle locale de l'image f_{tex} avec des opérateurs morphologiques (ouvertures et nivellements), et à partir de cette analyse, comment définir des gradients morphologiques de texture. Ces gradients seront combinés avec le gradient couleur pour construire des segmentations mixtes par LPE.

2 Granulométries et analyse multi-échelle morphologique

En utilisant un parallèle avec le tamisage de grains de gravier, Matheron [11] présentait la notion de granulométrie ou distribution de taille, appliquée aux images binaires. L'extension aux fonctions à niveaux de gris est faite par Serra [17]. Une granulométrie est l'étude de la distribution de taille des objets d'une image. Formellement, pour le cas discret, dans lequel nous travaillons, une granulométrie est une famille d'ouvertures $\Gamma = (\gamma_n)_{n\geq 0}$ indexée par rapport à un paramètre entier n telle que : 1) $\gamma_0(f) = f$; 2) $f \leq g \Rightarrow \gamma_n(f) \leq \gamma_n(g), \forall n \geq 0, \forall f, g$; 3) $\gamma_n(f) \leq f, \forall n \geq 0, \forall f$; 4) γ_n vérifie la loi d'absorption ; i.e., $\forall n, m \geq 0$, $\gamma_n\gamma_m = \gamma_m\gamma_n = \gamma_{\max(n,m)}$. D'ailleurs, une granulométrie par fermetures (ou *anti-granulométrie*) peut être définie comme une famille de *fermetures* croissantes $\Phi = (\varphi_n)_{n\geq 0}$.

Dans la pratique, les granulométries et les anti-granulométries les plus utiles sont celles données par des ouvertures/fermetures morphologiques : $\gamma_n(f) = \delta_{nB}(\varepsilon_{nB}(f))$ et $\varphi_n(f) = \varepsilon_{nB}(\delta_{nB}(f))$, respectivement, où B est un élément structurant fini de taille unitaire (typiquement un disque ou un segment de ligne droite) et $n = 1, 2, \cdots$. Les algorithmes traditionnels de mise en œuvre des granulométries impliquent par conséquent des ouvertures (fermetures) de taille croissante, et donc sont relativement lents. Toutefois, des algorithmes rapides spécifiques pour le calcul des granulométries ont été développés par Vincent [23].

L'analyse granulométrique d'une image f avec Γ consiste à évaluer chaque ouverture de taille n avec une mesure $\mathcal{M}(\gamma_n(f))$ (étant l'intégrale ou volume -somme de valeurs des pixels- de la fonction numérique). La *courbe granulométrique*, ou *spectre de forme* [9], de f par rapport à Γ et Φ, $PS_{\Gamma,\Phi}(f, n)$ ou $PS(f, n)$, est définie par la transformation normalisée suivante :

$$PS(f,n) = \frac{1}{\mathcal{M}(f)} \begin{cases} \mathcal{M}(\gamma_n(f)) - \mathcal{M}(\gamma_{n+1}(f)), & for\ n \geq 0 \\ \mathcal{M}(\varphi_{|n|}(f)) - \mathcal{M}(\varphi_{|n|-1}(f)), & for\ n \leq -1 \end{cases} \tag{3}$$

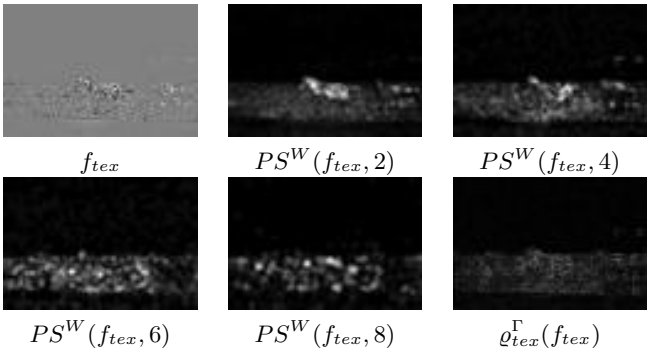


Figure 2 – *Couche de texture, granulométrie locale (fenêtre $W_x = 10 \times 10$) par des ouvertures isotropes, gradient morphologique de texture.*

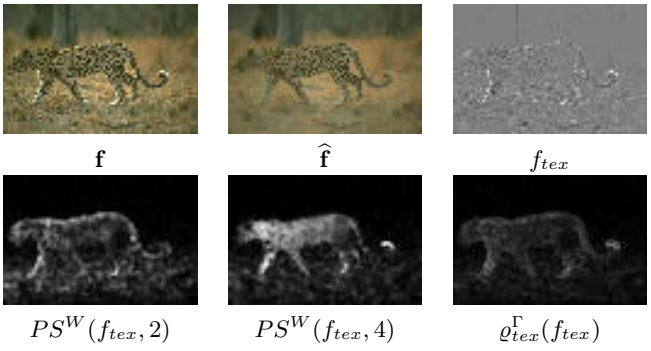


Figure 3 – *Décomposition colour/texture, deux images d'énergie locale et gradient morphologique de texture.*

Le spectre de forme $PS(f, n)$ fait correspondre à chaque taille n une certaine mesure des structures lumineuses (et sombres) de l'image qui ont cette taille. Le spectre de forme $PS(f, n)$ est une fonction de densité de probabilité (i.e. un histogramme) : un impulse ou mode dans le spectre de forme à une échelle donnée indique la présence de beaucoup de structures dans l'image associées à cette échelle.

Les distributions granulométriques de taille peuvent être utilisées comme descripteurs dans des schémas de classification de textures. Cependant, le descripteur de texture $PS(f, n)$ est global à toute l'image f, et si f contient plus d'une texture, la classification doit pouvoir se faire au niveau du pixel. Ceci est la notion derrière l'analyse granulométrique locale [6], qui consiste à calculer un spectre local de forme, ou plutôt dans une fenêtre $W_x = size_h \times size_v$ (de taille $size_h$ pixels en horizontale et $size_v$ en verticale) centré dans le pixel x. Pour les images à niveaux de gris, le spectre local de forme, $PS^W_x(f, n)$ ou simplement $PS^W(f, n)$, est obtenu en calculant le fonctionnel $PS(f_{W_x}, n)$ pour chaque pixel x, où f_{W_x} est la restriction de l'image f à l'intérieur de la fenêtre W_x. Cette méthode est très lourde du point de vue du calcul. Une approche plus rapide pour obtenir $PS^W(f, n)$ est fondée sur le calcul d'une seule série d'ouvertures/fermetures puis, pour chaque pixel x consiste à calculer l'intégrale de manière locale à W_x, i.e., $\mathcal{M}^{W_x}(g) = \sum_{y \in W_x} g(y)$. Comme résultat de ces calculs, nous obtenons une courbe granulométrique dans chaque pixel. Ce descripteur local de texture peut être utilisé pour classifier les différentes régions de texture dans l'image [6].

En fait, dans notre cas, cette analyse granulométrique locale se fera sur la couche de texture f_{tex} et on notera par $\{t^{\Gamma\Phi}_k(x)\}_{k \in K}$ la série d'images qui codent cette analyse, i.e.,

$$t^{\Gamma\Phi}_k(x) = PS^W(f_{tex}(x), k). \quad (4)$$

Nous dirons que $t^{\Gamma\Phi}_k(x)$ est l'*image d'énergie locale de taille* k ($k \geq 0$ pour les structures claires et $k \leq -1$ pour les structures sombres). Dans la Fig. 2 est montrée la couche de texture pour l'image des zébras, ainsi que les images d'énergie associées à la granulométrie locale (fenêtre $W_x = 10 \times 10$ et $K = \{-16, -14, \cdots, -2, 2, 4, \cdots, 16\}$) par des ouvertures isotropes. Comme nous pouvons l'observer, les structures de f_{tex} ont associé des valeurs élevées d'énergie locale à la taille qui leur correspond. Dans l'exemple sont montrés seulement des $t^{\Gamma}_k(x)$ (structures claires) ; l'analyse duale $t^{\Phi}_k(x)$ fournit l'énergie aux différentes échelles des structures sombres. Il est clair que le choix de la taille de la fenêtre dépend de la "taille" de la texture. Toutefois, l'influence est limitée : pour tous les exemples d'images naturelles que nous avons traitées, le choix $W_x = 10 \times 10$ a montré son adéquation. Un autre example de décomposition colour/texture, qui montre aussi deux images d'énergie locale, est donné dans la Fig. 3. Bien évidemment nous pouvons utiliser d'autres éléments structurants B non isotropes pour décrire, par exemple, des textures orientées.

En morphologie mathématique nous pouvons construire d'autres analyses multi-échelle avec d'autres opérateurs que les ouvertures/fermetures. Soit $ASF_n(f) = \varphi_n\gamma_n \cdots \varphi_2\gamma_2\varphi_1\gamma_1(f)$ le filtre alterné séquentiel de taille n (nous pouvons définir une autre famille de filtres en inversant l'ordre de l'ouverture et de la fermeture). La famille $\Xi = (ASF_n)_{n \geq 0}$ ne vérifie pas toutes les propriétés d'une granulométrie : elle n'est ni extensive ni anti-extensive, mais le plus important est qu'elle vérifie la loi d'absorption granulométrique et que par conséquent, elle permette de construire un tri multi-échelle des structures de f_{tex}. Par ailleurs, si chaque échelle est associée à un nivellement, la nouvelle famille de transformations, $\Lambda = (\lambda_n)_{n \geq 0}$ tel que $\lambda_n(f) = \lambda(ASF_n(f), f)$, fournit une décomposition des objets reconstruits dans chaque échelle n, i.e., perte d'objets consécutive aux simplifications successives. Il faut noter que maintenant les objets clairs/sombres de taille n apparaissent sur la même image.

Une analyse quantitative des objets associés à chaque taille n permet de définir une *pseudo-courbe granulométrique* que nous dénoterons Λ-*spectre de forme* et qui est définie comme suit :

$$\Lambda - PS(f, n) = \mathcal{M}(\lambda_n(f)) - \mathcal{M}(\lambda_{n+1}(f)), \ for\ n \geq 0. \quad (5)$$

De manière parallèle aux granulométries, nous pouvons construire une version locale de $\Lambda - PS(f,n)$, en calculant la mesure dans une fenêtre W centrée dans chaque pixel. La nouvelle série d'images d'énergie locale, i.e.,

$$t_k^{\Lambda}(x) = \Lambda - PS^W(f_{tex}(x), k), \quad (6)$$

donne une représentation multi-échelle alternative à celle associée aux ouvertures/fermetures. Ici, pour les exemples, $k \in K = \{2, 4, \cdots, 16\}$, car il faut noter que sur une image de 256 niveaux de gris, la valeur $t_k^{\Lambda}(x) = 128$, correspond à une énergie nulle, si $t_k^{\Lambda}(x) > 128$ l'énergie est associée à des structures claires et si < 128 à des structures sombres. Enfin, notons que la taille maximale aussi bien pour $t_k^{\Gamma\Phi}$ que pour t_k^{Λ} est limitée par la taille du nivellement utilisé pour construire $\widehat{\mathbf{f}}$ et par conséquent, f_{tex}.
D'autres décompositions morphologiques multi-échelle pourraient être définies pour bâtir d'autres descripteurs de "texture" : transformations associées à la dynamique, surface, volume, etc. [19, 20].

3 Gradients morphologiques de texture

Nous considérons maintenant les alternatives pour construire un gradient, associé à cette analyse multi-échelle, qui puisse permettre la détermination des contours des régions de textures différentes.
Dans chaque point x, le gradient morphologique $\varrho(x)$ de taille "boule unité" centré, $B(x)$, d'une image g peut être écrit en termes d'accroissements, i.e., $\varrho(g(x)) = \delta_B(g(x)) - \varepsilon_B(g(x)) = \vee[g(x) - g(y), y \in B(x)]$. Ceci permet d'utiliser une distance euclidienne pour définir un gradient de type morphologique pour la série d'images d'énergie locale, i.e.,

$$\varrho_{tex}(f_{tex}(x)) = \vee_y[d_E(t_k(x), t_k(y)), y \in B(x)], \quad (7)$$

où $d_E(t_k(x), t_k(y)) = \sqrt{\sum_{k\in K}(t_k(x) - t_k(y))^2}$ est la distance euclidienne entre deux pixels x et y pour toutes les images d'énergie locale.
En plus d'un gradient vectoriel de ce type, il est aussi possible de construire une autre sorte de gradient, en combinant les gradients de chaque image scalaire d'énergie. Différents tests ont montré que le gradient par supremum, i.e.,

$$\varrho_{tex}(f_{tex}(x)) = \bigvee_{k\in K}[\varrho(t_k(x))], \quad (8)$$

est aussi performant pour la segmentation que le gradient vectoriel par distance euclidienne. La Fig. 2 donne justement le gradient morphologique $\varrho_{tex}^{\Gamma}(f_{tex})$ calculé selon la Rel. 8. C'est bien ce dernier que nous avons appliqué à tous les exemples de cette étude.
Ces gradients de texture, dérivés des images d'énergie locale $\{t_k^{\Gamma\Phi}(x)\}$ et $\{t_k^{\Lambda}(x)\}$, respectivement $\varrho_{tex}^{\Gamma\Phi}(x)$ et $\varrho_{tex}^{\Lambda}(x)$, peuvent être utilisés avec la LPE pour segmenter l'image en régions selon la texture, voir les deux résultats correspondants dans la Fig. 4, pour la segmentation

Figure 4 – *Exemples de segmentation avec des gradients de texture et de structure (texture+couleur) par LPE et marqueurs, i.e.* $Wshed(\varrho, mrks)$.

des zébras par rapport aux marqueurs de la Fig. 1. Comme nous pouvons l'observer, ces deux gradients de texture segmentent correctement la région de chaque zébra (qui est bien définie par la texture), tel que nous le voulions. Cependant, comme nous pouvons le constater aussi, les contours des régions obtenues ne sont pas très précis. En effet, la segmentation selon un gradient de texture donne des régions approximatives.

4 Gradient structurel pour la segmentation texture+couleur par LPE

L'approche que nous avons choisie pour réaliser ce que nous appellerons une segmentation structurelle consiste à bâtir un gradient conjoint de couleur et de texture. Une fois qu'on a à notre disposition un gradient couleur et un gradient de texture, il semble évident que nous pouvons les combiner pour obtenir un gradient structurel. Parmi la multitude d'alternatives pour la combinaison de gradients, nous en avons retenu deux qui nous paraissent particulièrement simples à mettre en œuvre et suffisamment flexibles pour évaluer l'influence d'un gradient par rapport à l'autre. En fait, il s'agit d'une part, de la somme du gradient couleur et d'une pondération du gradient de texture (pour contrôler l'influence du deuxième) ; et d'autre part, d'une combinaison linéaire barycentrique des deux gradients. En termes mathématiques, nous avons :

$$\begin{aligned} \varrho_{str}^{I-\alpha}(\mathbf{f}) &= \varrho_{col}(\widehat{\mathbf{f}}) + \alpha\varrho_{tex}(f_{tex}), \\ \varrho_{str}^{II-\alpha}(\mathbf{f}) &= (1-\alpha)\varrho_{col}(\widehat{\mathbf{f}}) + \alpha\varrho_{tex}(f_{tex}), \end{aligned} \quad (9)$$

où, pour les deux cas, $0 \leq \alpha \leq 1$. Pour les deux cas, ϱ_{col} et ϱ_{tex} correspondent aux définitions Rel.(1) et (8) respectivement.
Sur la Fig. 4 nous montrons une comparaison de segmentation par LPE des zébras selon différents gradients structurels. Nous observons que, aussi bien pour l'analyse de

texture liée aux ouvertures/fermetures que pour celle liée aux nivellements des filtres alternés séquentiels, le gradient structurel équilibré, i.e., $\varrho_{str-\Gamma\Phi}^{I-\alpha=1}(\mathbf{f})$ et $\varrho_{str-\Lambda}^{I-\alpha=1}(\mathbf{f})$ respectivement, améliore les segmentations par rapport au gradient exclusivement de couleur $\varrho_{col}(\widehat{\mathbf{f}})$. Par ailleurs, pour cet exemple, nous observons aussi que le meilleur résultat correspond à $\varrho_{str-\Gamma\Phi}^{II-\alpha=0.8}(\mathbf{f})$; ce qu'on pourrait attendre, car comme nous l'avons signalé précédemment, la texture ici est plus importante que la couleur. De plus, le fait de combiner avec le gradient couleur permet d'obtenir des contours plus précis.

Pour compléter les résultats de notre étude, nous avons testé les gradients structurels sur une large série d'exemples d'images couleur naturelles et nous avons évalué la segmentation couleur vs. la segmentation structurelle par LPE. La Fig. 5 montre sept images représentatives : les exemples 1-3 correspondent à la segmentation avec un marqueur pour l'objet et un marqueur pour l'extérieur et les exemples 4-7 correspondent à la segmentation en sélectionnant 50 régions volumiques. Nous donnons pour chaque image la segmentation selon la couleur et la segmentation structurelle équilibrée couleur+texture pour les deux familles de descripteurs de texture que nous avons étudiés ($\varrho_{str-\Gamma\Phi}^{I-\alpha=1}$ et $\varrho_{str-\Lambda}^{I-\alpha=1}$). Etant donné qu'à priori on ne peut pas savoir pour une image si c'est la couleur ou bien la texture qui constitue l'information la plus pertinente pour la segmentation, il nous paraît que le choix le plus judicieux est une combinaison équilibrée de ce type.

Nous constatons que pour l'exemple du papillon, la segmentation structurelle est toujours plus cohérente que celle de la couleur. Avec $\varrho_{str-\Gamma\Phi}^{I-\alpha=1}$, on n'obtient qu'une partie des ailes (qui ont la même couleur-texture) et avec $\varrho_{str-\Lambda}^{I-\alpha=1}$ les deux couleurs-textures des ailes sont prises en compte, en produisant une segmentation parfaite. Une analyse similaire est valable pour l'image suivante des cellules. L'image du tigre est un bon contre-exemple qui montre que si la texture entre l'objet d'intérêt et le reste est très similaire, le fait d'utiliser un gradient structurel ne ferait qu'introduire un biais de texture. Pour les segmentations en 50 régions sur des images qui contiennent des objets colorés et bien contrastés ainsi que des régions grandes avec ou sans texture, nous observons que le gradient structurel permet d'améliorer en partie le problème bien connu de la LPE qui sur-segmente les régions grandes et homogènes. Par ailleurs, certains objets de "petite taille" sont mieux segmentés avec les gradients structurels. Ceci peut être observé sur les images 4 et 5. Les images 6 (avec le léopard) et 7 (avec la famille d'ours) montrent l'apport de la texture pour permettre l'algorithme de segmentation de trouver les contours de certaines régions qui ne sont pas déterminées par la couleur. Finalement, il est difficile d'affirmer d'une manière générale si les partitions pour $\varrho_{str-\Gamma\Phi}^{I-\alpha=1}$ sont plus pertinentes que celles pour $\varrho_{str-\Lambda}^{I-\alpha=1}$.

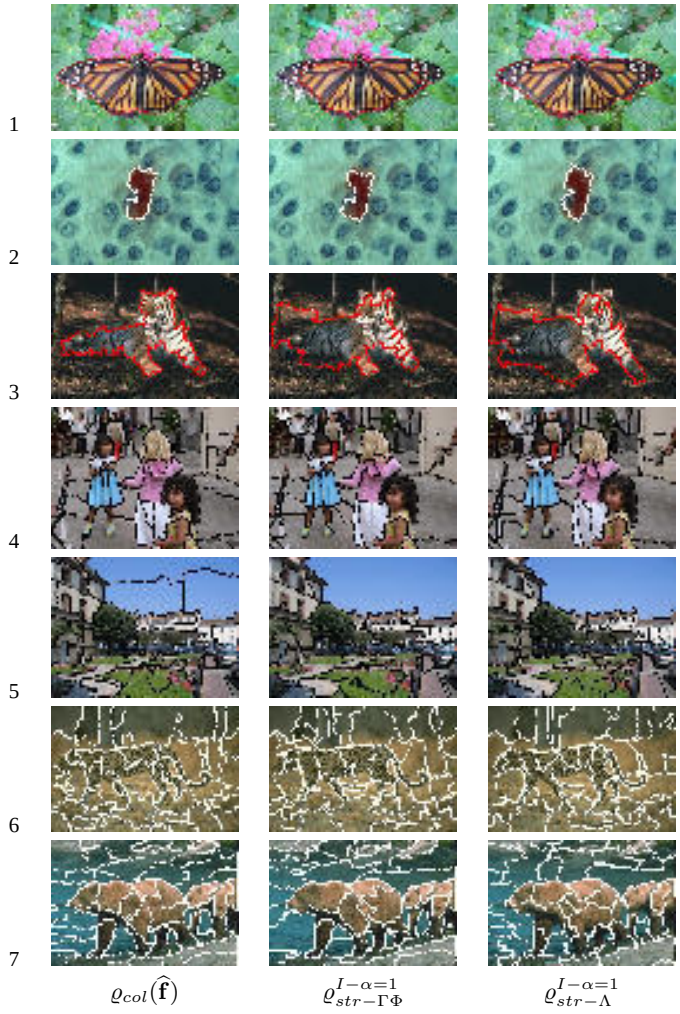


Figure 5 – *Segmentation couleur vs. segmentation structurelle par LPE. Exemples 1-3, marqueur pour l'objet ; exemples 4-7 sélection des 50 volumiques.*

5 Conclusions et perspectives

Cet article présente une approche morphologique pour le calcul de gradients de texture et illustre comment les utiliser pour la segmentation d'image selon la texture ; et plus généralement, pour la segmentation conjointe texture + couleur (e.g., segmentation structurelle). Nous avons montré que ces différents gradients sont directement utilisables pour la segmentation morphologique par ligne de partage des eaux et que les partitions obtenues avec des gradients structurels sont, dans la plupart des cas, plus pertinentes que celles obtenues seulement avec des gradients couleur. En particulier, nous avons montré que les régions de texture sont mieux déterminées et qu'on arrive à réduire la sur-segmentation des régions grandes et homogènes.

A présent, nous nous intéressons à l'étude de la manière de construire des décompositions texture-couleur, en ne limitant pas la couche de texture à l'information de luminance. Pour ce faire, nous allons évaluer l'intérêt d'uti-

liser des résidus d'ouvertures/nivellements couleur (typiquement construits avec des ordres totaux dans la représentation luminance/saturation/teinte). D'autre part, nous travaillons sur une combinaison automatique du gradient couleur et du gradient texture pour que ce couplage d'information s'adapte aux caractéristiques des images.

Références

[1] J. Angulo, J. Serra. Morphological coding of color images by vector connected filters. In *Proc. of IEEE 7th International Symposium on Signal Processing and Its Applications (ISSPA'03)*, Paris, France July 2003, Vol. I, pp. 69–72.

[2] J. Angulo, B. Marcotegui. Sur l'influence des conditions d'éclairage dans la segmentation morphologique couleur par LPE. In *Actes de CORESA 2005 (Compression et Représentation des Signaux Audiovisuels)*, pp. 313–318, Rennes, France, November 2005.

[3] J. Angulo, J. Serra. Modelling and Segmentation of Colour Images in Polar Representations. To appear in *Image and Vision Computing*, 2006.

[4] J.-F. Aujol, G. Gilboa, T. Chan, S. Osher. Structure-Texture Image Decomposition - Modeling, Algorithms, and Parameter Selection. *International Journal of Computer Vision*, 67(1) :111–136, 2006.

[5] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In *Mathematical Morphology and its Applications to Image and Signal Processing, Proc. ISMM'94* Kluwer, 69–76, 1994.

[6] E.R. Dougherty, J.T. Newell, J.B. Peltz. Morphological texture-based maximum likelihood pixel classification based on local granulometric moments. *Pattern Recognition*, 25 : 1181-1198, 1992.

[7] P.R. Hill, C.N. Canagarajah, D.R. Bull. Image segmentation using a texture gradient based watershed transform. *IEEE Transactions on Image Processing*, 12(12) :1618–1633, 2003.

[8] M.A. Hoang, J.-M. Geusebroek, A.W.M. Smeulders. Color texture measurement and segmentation. *Signal Processing*, 85 : 265-275, 2005.

[9] P. Maragos. Pattern Spectrum and Multiscale Shape Representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11 : 701-716.

[10] D.R. Martin, C.C. Fowlkes, J. Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26 : 1-20, 2004.

[11] G. Matheron. *Ensembles aléatoires et géométrie intégrale. Tome II*. Les Cahiers du Centre de Morphologie Mathématique, Fascicule 6, Ecole des Mines de Paris, 1972.

[12] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1) : 21–45, 1990.

[13] F. Meyer. An Overview of Morphological Segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(7) : 1089–1118, 2001.

[14] F. Meyer. Levelings, Image Simplification Filters for Segmentation. *Journal of Mathematical Imaging and Vision*, 20 : 59–72, 2004.

[15] Y. Meyer. Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. In *University Lecture Series Vol. 22*, AMS 2002.

[16] K.A. Patwardhan, G. Sapiro. Automatic Image Decomposition. In *Proc. of IEEE International Conference on Image Processing (ICIP'04)*, Singapore, October 2004, Vol. I, pp. 645-648.

[17] J. Serra. *Image Analysis and Mathematical Morphology*, Vol. I. Academic Press, London, 1982.

[18] A. Sofou, G. Evangelopoulos, P. Maragos, Coupled geometric and texture PDE-based segmentation. In *Proc. of IEEE International Conference on Image Processing (ICIP'05)*, Genova, Italy, September 2005, Vol. II, pp. 650–653.

[19] C. Vachier and F. Meyer. Extinction value : a new measurement of persistence. In *1995 IEEE Workshop on Nonlinear Signal and Image Processing*, Neos Marmaras, Greece, 254–257, 1995.

[20] C. Vachier. Morphological Scale-Space Analysis and Feature Extraction. In *Proc. of IEEE International Conference on Image Processing (ICIP'01)*, Thessaloniki, Greece, September 2005, Vol. III, pp. 676–679.

[21] I. Vanhamel, A. Katartzis, H. Sahli. Hierarchical segmentation via a diffusion scheme in color/texture feature space. In *Proc. of IEEE International Conference on Image Processing (ICIP'03)*, Barcelona, Spain, September 2003, Vol. I, pp. 969–972.

[22] L. Vese, S. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comp.*, 19 : 553-572, 2003.

[23] L. Vincent. Local grayscale granulometries based on opening trees. In *Mathematical Morphology and its Applications to Image and Signal Processing, Proc. ISMM'96*, 273–280, Kluwer, 1996.

Extension de l'espace d'acquisition pour les méthodes de Shape-from-silhouette

Concours Jeune Chercheur : Oui

Résumé

L'acquisition de la forme tridimensionnelle d'un personnage est une étape indispensable pour un grand nombre d'applications de réalité virtuelle, augmentée et dans la conception de jeux vidéos. Celle-ci soit être complète et précise pour offrir le meilleur réalisme possible. Les méthodes dites "Shape From Silhouette" (SFS) permettent d'obtenir cette estimation en temps réel à partir de plusieurs caméras. L'une des limitations de ces méthodes est que le personnage doit être entièrement visible dans toutes les caméras pour être reconstruit entièrement. Dans cet article nous proposons une extension à SFS qui permet de reconstruire une estimation 3d de la forme d'un objet même s'il sort du champ de vision d'une ou plusieurs caméras.

Mots clefs

Réalité virtuelle, réalité augmentée, reconstruction géométrique

1 Introduction

Ce travail s'insère dans un projet de réalité augmentée dont l'un des objectifs est l'insertion en temps réel, d'un personne réelle filmée par plusieurs caméras, dans un décor virtuel. Pour assurer une insertion la plus réaliste possible, il est important de modéliser précisément les interactions géométriques et photométriques entre la personne et son environnement. Pour cela il est nécessaire de disposer d'une représentation tridimensionnelle de la personne.

La littérature propose un grand nombre de méthodes permettant l'acquisition de la forme d'une personne. L'une des approches les plus populaire est celle connue sous le nom de "Shape-from-silhouette" que nous noterons SFS dans la suite de cet article. Parmi les travaux consacrés à SFS, certains proposent l'acquisition d'une forme humaine en temps réel. On peut citer ceux de l'équipe de Kong Man Cheung [1] qui a été dans les premières à proposer en 2000 un algorithme temps réel. Depuis d'autres méthodes [2, 3, 4, 5] ont permis d'obtenir du temps réel en utilisant principalement les outils proposés par les cartes graphiques programmables [6, 7].

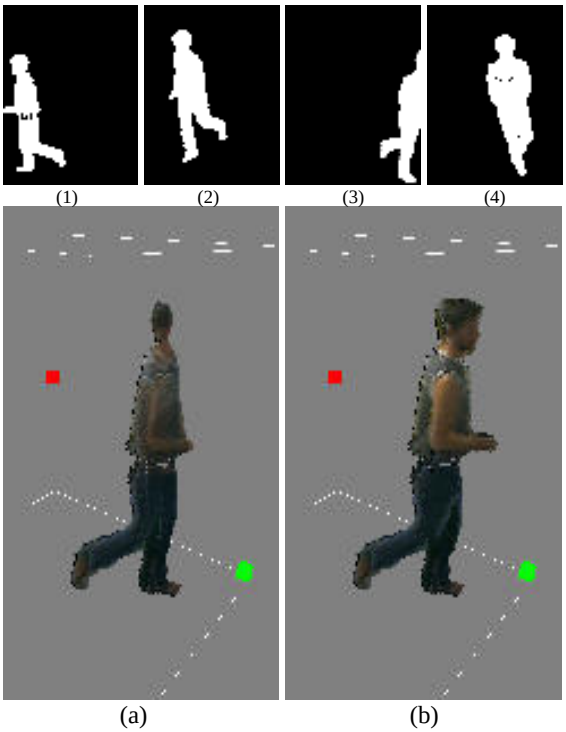


Figure 1 – Différences de reconstruction d'un objet lorsque celui-ci sort du champs de vision d'une caméra : (a) en utilisant les méthodes basées SFS actuelles ; (b) en utilisant notre algorithme. Le coloriage voxélique (réalisé par un lancer de rayon) n'est donné que pour faciliter la compréhension des images.

A partir des silhouettes[1] d'un l'objet, les algorithmes SFS permettent d'estimer la forme 3d de cet objet. Si ces méthodes permettent de retrouver rapidement une forme globale du sujet filmé, l'une de ses limites réside dans la contrainte que seules les parties visibles depuis toutes les caméras à chaque instant, peuvent être reconstruites. Dans le cas où le sujet à reconstruire est une personne en mouvement, il est difficile, à moins d'avoir des caméras haute définition placées loin de l'objet, d'assurer une visibilité totale dans chaque caméra. La figure 1 illustre cette contrainte, une grande partie de la personne n'est pas vue dans la silhouette 3 et n'est donc pas reconstruite (Figure 1.a).
Dans cet article, nous proposons une extension de l'algorithme SFS qui permet de pallier ce problème, en supposant

[1] images binaires associées aux images acquises d'un objet, où 1 représente l'objet et 0 représente le reste.

que l'objet acquis soit partiellement visible depuis toutes les caméras. Après un court résumé des principes de SFS, le chapitre 3 présente notre extension de SFS qui permet de reconstruire une estimation 3d de la forme d'un objet O même s'il sort du champ de vision d'une ou plusieurs caméras. Nous discutons ensuite des résultats obtenus. Enfin dans le chapitre 5 nous concluons sur le travail effectué et proposons certaines perspectives de ce travail.

2 Méthodologie des algorithmes basés SFS

Les méthodes basées SFS sont fréquemment utilisées pour calculer une estimation 3d de la forme d'un objet. Le formalisme de construction de la VH d'un objet a été introduit par A. Laurentini [8].

Il peut être décrit comme suit :
Soit un objet 3d O filmé par n caméras cam_i. M_i est la matrice de projection associée à la caméra cam_i et I_i l'image acquise depuis cette caméra. Enfin, S_i est l'image de silhouette associée à I_i.
Soit un point 3d P. Si celui-ci est contenu dans le volume de O alors il se projette dans toutes les silhouettes :

$$\forall i = 1, ..., n \ , \ \exists p_i \in S_i \ , \ p_i = M_i.P.$$

où p_i est la projection de P sur la silhouette S_i.
La VH de O est alors définie comme le volume contenant l'ensemble des points 3d se projetant sur toutes les silhouettes S_i. Il y a principalement deux méthodes pour calculer la VH d'un objet O, que nous allons maintenant détailler.

L'approche basée surface

La VH d'un objet déduite d'un ensemble de n images de silhouette, est construite à partir de l'intersection des n cônes de silhouette. Ces cônes sont définis par la projection, dans l'espace 3d, des contours des silhouettes à travers le centre de projection de la caméra associée. Ainsi, la VH d'un objet O sera décrite par un ensemble de surfaces 2d, ces surfaces sont définies par l'intersection des surfaces des cônes de silhouette.
D'un coté, cette approche permet des calculs en temps réel [2, 7, 9]. De l'autre, les résultats obtenus ne sont pas utilisables pour calculer les informations volumiques nécessaires à la mise en correspondance avec des modèles génériques d'humanoïdes (utilisés pour l'estimation de la posture et l'interprétation de mouvement de la personne acquise) [10].

L'approche basée volume

Une approche équivalente définie la VH d'un objet O comme étant le volume maximum qui se projette exactement sur toutes les silhouettes de O [8]. Basé sur cette définition, l'approche le plus utilisée [1, 3, 6, 4, 5] calcule une estimation de la VH de O par un ensemble de voxels.

La zone d'acquisition est partitionnée en m voxels V_j où $j = 1, ..., m$. Soit v_{ij} l'ensemble des pixels de I_i sur lesquels se projetteV_j :

$$v_{ij} = (M_i.V_j) \cap I_i.$$

Le nombre nb_j de silhouettes sur lesquelles se projette V_j est défini par :

$$nb_j = Card\{v_{ij} \ , \ v_{ij} \cap S_i \neq \emptyset\}.$$

Nous définissons $CS(O)$ la carte de silhouette de l'objet O comme étant l'ensemble de tous les couples (V_j, nb_j).
Si un voxel V_j se projette sur toutes les silhouettes de O alors il appartient à sa VH. Ainsi, $CS(O)$ peut être divisée en n sous-ensembles SFS_i :

$$SFS_i = \bigcup_{j=1}^{m} (V_j \ , \ nb_j = i).$$

L'estimation voxélique de la VH de O est alors définie par SFS_n où n est le nombre de caméras utilisées (voir Figure 2).

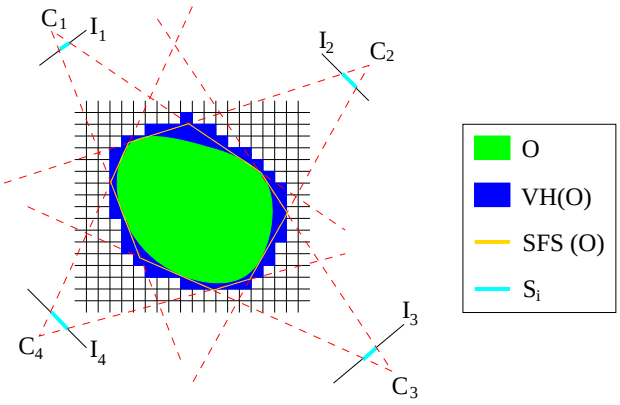


Figure 2 – Représentation 2d d'un objet O vu par 4 caméras, la VH correspondante et son estimation voxélique.

Si certaines parties du corps d'une personne en mouvement ne sont plus visibles depuis l'une des caméras, alors les informations correspondantes ne seront pas estimées dans sa VH. Pourtant, ces informations pourraient être obtenues à partir des autres caméras. Dans la suite, nous utiliserons ce postulat pour ajouter de nouvelles informations à la VH d'un objet.

3 Contributions

Dans la plupart des méthodes basées SFS, la zone d'acquisition, où un objet est reconstruit, correspond à l'intersection des cônes de vision des caméras. Les limitations de cette approche proviennent du fait que :
– la zone d'acquisition ne peut être étendue au delà de l'intersection des cônes de vision, notamment lorsqu'un grand nombre de caméras sont utilisées ;

– Il est difficile, pour une personne en mouvement, de rester visible à tout moment depuis toutes les caméras.

Pour éviter cela, nous allons prendre en compte le nombre potentiel de caméras qui voient un voxel. Si un voxel est visible depuis seulement $n-k$ caméras (où $k \in [1, ..., n_{min}]$ avec $n_{min} < n$) sur les n disponibles, alors il se projettera sur un maximum de $n-k$ silhouettes. Mais dans les méthodes actuelles basées SFS, il est nécessaire qu'un voxel soit vu par toutes les caméras ($nb_j = n$) pour qu'il soit contenu dans l'estimation voxélique de la VH d'un objet.

Dans un premier temps, nous utiliserons ce concept pour calculer une estimation de la forme d'un objet dans l'espace non visible depuis k caméras. Puis nous utiliserons les propriétés de connexité de l'objet acquis, afin de choisir les informations pertinentes pour étendre sa VH.

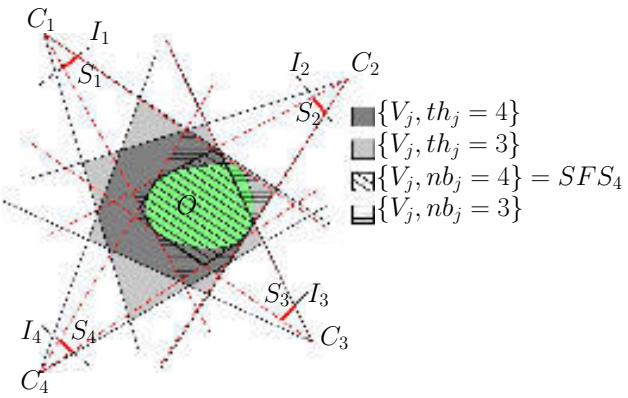


Figure 3 – Représentation 2d d'un objet O et son estimation en utilisant SFS_4. Les voxels qui étendent la forme de O sont ceux pour lesquels $th_j = nb_j = 3$.

Soit th_j le nombre d'images sur lesquelles se projette le voxel V_j :

$$th_j = Card\{v_{ij} \ , \ v_{ij} \cap I_i \neq \emptyset\}.$$

La carte de projection $CP(O)$ d'un objet O est alors définie comme étant l'ensemble de tous les couples (V_j, th_j).

Si un voxel V_j est contenu dans le volume de O, il se projette alors sur th_j images et nb_j silhouettes ; ainsi $th_j = nb_j$.

Nous cherchons l'ensemble des voxels V_j contenus dans le volume d'un objet O. Pour cela nous comparons $CS(O)$ et $CP(O)$:

– si $th_j \neq nb_j$ alors V_j n'est pas contenu dans le volume de O ;
– sinon V_j est potentiellement contenu dans le volume de O (Figure 3).

L'ensemble $R = \{V_j, nb_j = th_j\}$ de tous les voxels potentiels peut être séparé en n sous-ensembles R_i :

$$R_i = \{V_j \ , \ nb_j = th_j = i\}.$$

Notons que

$$SFS_n = \bigcup_{V_j \in R_n} \{V_j\}.$$

Ainsi, pour étendre SFS_n, nous choisirons des voxels contenus dans les sous-ensembles R_{n-k} avec $k \in [1, ..., n_{min}]$ et $n_{min} < n$. Soit $\Re_{n_{min}}$ l'union de tous les R_{n-k} :

$$\Re_{n_{min}} = \bigcup_{k=1}^{n_{min}} R_{n-k}.$$

Un objet 3d est connexe ainsi l'estimation 3d de sa forme doit aussi être connexe. $\Re_{n_{min}}$ est l'union de L composantes connexes notées c_l où $l = 1, ..., L$. Afin de satisfaire la contrainte de connexité du volume reconstruit, nous choisissons les composantes connexes de $\Re_{n_{min}}$ qui sont connectées à SFS_n (comme décrit dans la figure 4).

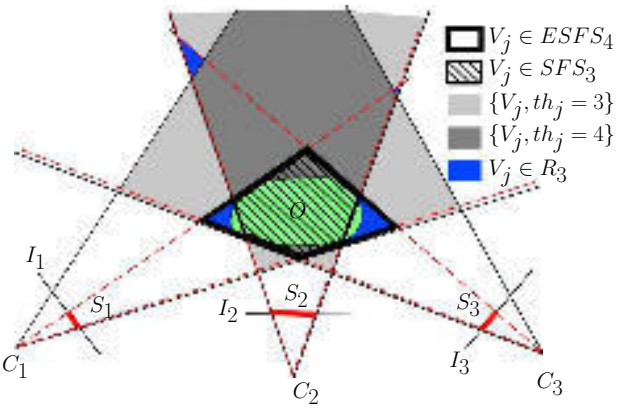


Figure 4 – Représentation 2d de O, son estimation volumique à partir de SFS_3 et de $ESFS_3$. Nous notons que $ESFS_3$ est plus précis que SFS_3.

Soit $C_{n_{min}}$ l'ensemble des composantes connexes de $\Re_{n_{min}}$ connectées à R_n :

$$C_{n_{min}} = \bigcup_{l=1}^{L} (c_l \ , \ connexe(c_l \cup R_n)).$$

$ESFS_n$ défini le volume estimé à partir de notre algorithme, et étendant SFS_n :

$$ESFS_n = SFS_n \cup C_{n_{min}}.$$

L'estimation de forme définie par $ESFS_n$ dépend de la valeur de n_{min}. A. Laurentini [8] a montré que plus le nombre de caméras utilisé est grand, meilleure est l'estimation de la forme d'un objet.

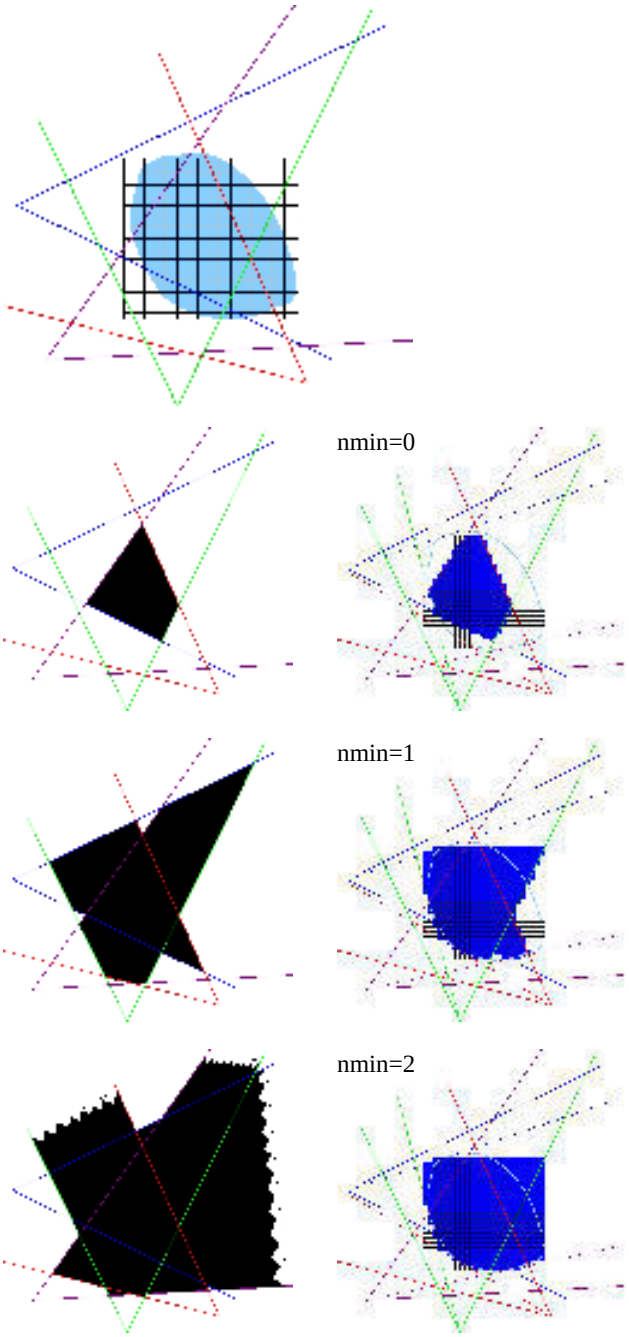


Figure 5 – Variation de l'estimation de la forme d'un objet en fonction de la valeur de $nmin$. L'image du haut représente la configuration caméras/objet/espace voxélique, à gauche l'espace d'acquisition disponible et à droite la forme obtenue.

Si $n_{min} = 1$, alors $ESFS_n$ complète l'estimation SFS_n, avec les informations visibles depuis $n - 1$ caméras. Ceci est la meilleure estimation possible (en terme de précision) avec strictement moins de n caméras. De plus, la zone d'acquisition utilisée pour construire $ESFS_n$ est légèrement plus grande que celle utilisée pour SFS_n. Si n_{min} est proche de $n-1$, alors la zone d'acquisition de $ESFS_n$ est nettement plus grande que celle de SFS_n. Mais la forme estimée sera moins précise pour les parties des O visibles depuis seulement $n - n_{min}$ caméras. Ainsi, la valeur de n_{min} doit être choisie en fonction de l'application visée (ie, l'utilisation faite de l'estimation de la forme de O) comme montré dans la figure 5.

4 Résultats

Notre algorithme a été testé sur différents jeux de données réelles, acquis depuis 4 caméras avec une résolution de 320x240 pixels. La meilleure précision de reconstruction, lorsque certaines parties de l'objets n'étaient pas visibles depuis une caméra, a été obtenue en utilisant $n_{min} = 3$.

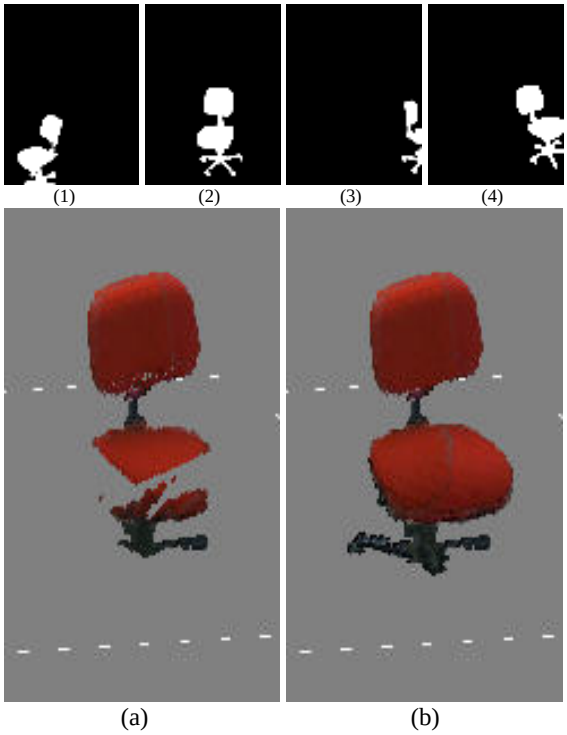


Figure 6 – Estimation voxélique de la forme d'un objet complexe : (a) algorithme de base de SFS ; (b) notre algorithme.

La figure 6 montre les résultats obtenus sur un objet complexe. Il y a visibilité partielle dans les silhouettes 1, 3 et 4. La méthode de base de SFS fourni une reconstruction partielle de l'objet (voir Figure 6.a). Notre algorithme permet une reconstruction plus complète de la chaise du fait que les portions non visibles depuis une caméra sont visibles depuis les autres caméras. Le pied de la chaise n'est, quant à lui, pas reconstruit complètement, du fait qu'il n'est pas visible depuis la plupart des caméras.
La figure 7 montre les résultats obtenus lors de l'acquisition d'une personne en mouvement. Dans les deux cas (figures 6 et 7), notre méthode ajoute de nouvelles informations valides (du point de vue des images de silhouette) à l'estimation de la forme de l'objet.

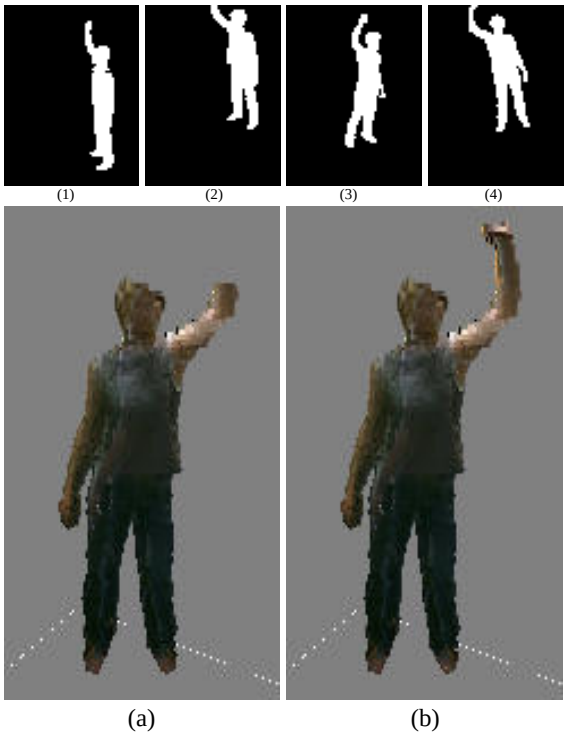


Figure 7 – Estimation voxélique de la forme d'une personne en mouvement : (a) algorithme de base de SFS ; (b) notre algorithme.

Les performances de notre algorithme sont très proches de celles de l'algorithme de base, seules deux étapes ont été ajoutées :

1. Le calcul de la carte de projection $CP(O)$, qui est réalisé une fois en pré-traitement de la phase d'acquisition. En effet, $CP(O)$ dépend uniquement des paramètres intrinsèques et extrinsèques des caméras qui sont constants durant l'acquisition ;
2. Le calcul de la connexité 3d : à chaque pas de temps, nous parcourons les ensembles R_n et R_{n-k}. Le temps de parcours étant négligeable par rapport au temps de calcul de la projection des voxels sur chaque silhouette.

L'implémentation expérimentale de notre algorithme permet 60 estimations de forme par secondes (pour une résolution voxélique de 128^3 et $n_{min} = 3$, alors que notre implémentation de la méthode de base de SFS atteint les 65 estimations de forme par seconde. Ces résultats nous montrent que notre algorithme est utilisable dans le cadre d'applications visant le temps réel.

5 Conclusions et perspectives

Dans cet article, nous avons proposé une extension de l'algorithme de "Shape-from-silhouette". Cette méthode permet d'avoir une zone d'acquisition étendue par rapport à celle disponible avec les algorithme habituels de SFS. Hormis le problème du calibrage des caméras, notre seule hypothèse porte sur le fait que l'objet doit être majoritairement visible dans toutes les caméras.
L'estimation de forme obtenue à partir de notre méthode contient celle qui peut être obtenue par SFS. Notre extension permet, de plus, d'estimer la forme des parties de l'objet qui ne sont pas visibles depuis une ou plusieurs caméras, du moment qu'elles le soient depuis les autres. De plus cette méthode est applicable même avec une contrainte temps réel.
Nous travaillons actuellement sur la formalisation de l'erreur de reconstruction en fonction du nombre de caméras utilisées. Cette méthode est utilisée dans le cadre d'une application de suivi de mouvement en temps réel et devrait être implémentée sur GPU afin de pouvoir travailler avec des caméras haute fréquence.

Références

[1] Kong Man Cheung, Takeo Kanade, J.-Y. Bouguet, et M. Holler. A real time system for robust 3d voxel reconstruction of human motions. Dans *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, volume 2, pages 714 – 720, Juin 2000.

[2] Wojciech Matusik, Chris Buehler, et Leonard McMillan. Polyhedral visual hulls for Real-Time rendering. Dans *12th Eurographics Workshop on Rendering Techniques*, pages 115–126, 2001.

[3] Jean-Marc Hasenfratz, Marc Lapierre, et François Sillion. A real-time system for full body interaction with virtual worlds. *Eurographics Symposium on Virtual Environments*, pages 147–156, 2004.

[4] Fabrice Caillette, Aphrodite Galata, et Toby Howard. Real-Time 3-D Human Body Tracking using Variable Length Markov Models. Dans *Proceedings of British Machine Vision Conference (BMVC)*, volume 1, pages 469–478, 2005.

[5] Bastian Goldluecke et Marcus Magnor. Real-time free-viewpoint video rendering from volumetric geometry. *Visual Communications and Image Processing 2003*, 5150(1) :1152–1158, 2003.

[6] Jean-Marc Hasenfratz, Marc Lapierre, Jean-Dominique Gascuel, et Edmond Boyer. Real-time capture, reconstruction and insertion into virtual world of human actors. Dans *Vision, Video and Graphics*, pages 49–56. Eurographics, Elsevier, 2003.

[7] M. Li, M. Magnor, et H. Seidel. Hardware accelerated visual hull reconstruction and rendering. Dans *Graphics Interface*, 2003.

[8] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2) :150–162, 1994.

[9] Jean-Sébastien Franco et Edmond Boyer. Exact polyhedral visual hulls. Dans *Fourteenth British Ma-*

chine Vision Conference (BMVC), pages 329–338, Septembre 2003. Norwich, UK.

[10] Ivana Mikic, Mohan Trivedi, Edward Hunter, et Pamela Cosman. Human body model acquisition and tracking using voxel data. *Int. J. Comput. Vision*, 53(3) :199–223, 2003.

Comparaison de schémas de décomposition en ondelettes pour un traitement local des maillages surfaciques triangulaires

C. Roudet[1] F. Dupont[1] A. Baskurt[2]

[1]Laboratoire LIRIS, UMR 5205 CNRS
Université Claude Bernard Lyon 1, Villeurbanne Cedex

[2]Laboratoire LIRIS, UMR 5205 CNRS
INSA Lyon, Villeurbanne Cedex

{croudet, fdupont, abaskurt}@liris.cnrs.fr

Résumé

Depuis quelques années, les objets tridimensionnels concurrencent le multimédia traditionnel (images, sons et vidéos) et sont exploités par de plus en plus d'applications. Les résultats récents en compression d'objets lisses par morceaux, représentés sous forme de maillages surfaciques, ont motivé notre recherche d'une adaptation de ces techniques aux surfaces naturelles par le biais de l'analyse multirésolution. Nous présentons une analyse des détails haute-fréquence obtenus à partir de plusieurs schémas de décomposition en ondelettes, afin d'envisager une segmentation en patchs surfaciques suivie d'une décomposition adaptative de ces maillages.

Mots clefs

Modèles 3D, surfaces de subdivision, ondelettes géométriques, analyse multirésolution, compression.

1 Introduction

Grâce aux dernières avancées des techniques d'échantillonnage, les images, sons et vidéos numériques font maintenant largement partie de notre quotidien. Plus récemment, le développement de l'infographie et de la vision tridimensionnelle a ouvert la voie à la modélisation d'objets ou de scènes complexes en trois dimensions. Ceux-ci sont le plus souvent représentés sous forme de maillages surfaciques triangulaires où l'on code la position des sommets dans l'espace 3D euclidien (information géométrique) ainsi que la manière dont ils sont connectés entre eux (information topologique).

Pour répondre aux attentes de réalisme actuelles, il est nécessaire de sélectionner un grand nombre d'échantillons sur la surface de ces objets afin d'obtenir une représentation précise et détaillée. C'est pourquoi, même si les espaces de stockage des ordinateurs et la vitesse de transmission des réseaux ne cessent d'augmenter, il paraît indispensable de disposer de techniques de compression efficaces pour stocker, échanger et même visualiser de tels objets.

Pour les applications manipulant des données sensibles, des méthodes de compression sans perte sont généralement utilisées. Celles-ci se caractérisent par une réorganisation de l'information et sont les premières à avoir été proposées pour la compression d'objets 3D. Mais dans la majorité des cas, il est possible d'obtenir des taux de compression bien meilleurs en s'autorisant quelques pertes que l'on cherche à dissimuler.

L'obtention d'une séquence de bits de taille minimale n'est pas le seul objectif visé par les applications manipulant les maillages. En effet, un des enjeux actuels est de proposer une adaptation du transfert de ces données aux ressources à disposition (type de réseau et nature des terminaux utilisés pour la visualisation) ainsi qu'aux diverses demandes de l'utilisateur. Les techniques d'analyse multirésolution permettent de répondre à ces besoins car elles s'appuient sur une représentation scalable des données. Celles-ci utilisent généralement une transformée en ondelettes, outil performant qui a fait ses preuves en terme de compression d'images et de vidéos puisqu'il a notamment été intégré dans la norme JPEG2000 [1].

Actuellement ces méthodes réalisent une projection globale des maillages dans l'espace transformé, sans chercher à adapter la décomposition en ondelettes et la quantification des coefficients à la courbure, la rugosité ou la direction des textures caractérisant la surface des objets. Afin d'exploiter ces remarques, nous présentons une analyse des coefficients d'ondelettes obtenus lors l'utilisation de différents schémas de décomposition sur des surfaces plus ou moins bruitées. L'étude de leur répartition permet d'envisager une segmentation basée sur les caractéristiques surfaciques des maillages qui constituerait la première étape d'une chaîne de compression adaptative. Cette étude pourrait également être exploitée pour envisager des méthodes adaptatives de débruitage, filtrage, lissage ou tatouage de maillages.

Dans le paragraphe suivant, nous présentons brièvement plusieurs travaux récents d'analyse de maillages surfaciques. Ensuite nous détaillons notre méthode d'analyse multirésolution utilisant le potentiel des ondelettes de sec-

onde génération. Enfin nous commentons les résultats obtenus avant de présenter les perspectives de ce travail.

2 Etat de l'art

Contrairement aux images non vectorielles, aux sons et aux vidéos, les maillages se caractérisent le plus souvent par un échantillonnage irrégulier. C'est la raison pour laquelle les techniques de traitement d'images sont difficiles à étendre aux maillages. Pour l'analyse de ces données tridimensionnelles, il existe plusieurs méthodes de projection du signal afin d'obtenir une information moins redondante. Certaines proposent des concepts spécialement adaptés à cette irrégularité et se basent sur une extension de l'analyse de Fourier classique aux maillages surfaciques. Cependant ces méthodes ne proposent qu'une progressivité géométrique. C'est pourquoi beaucoup de travaux se sont tournés vers l'analyse multirésolution, qui utilise une réorganisation de la topologie ce qui permet d'adapter les techniques utilisées en traitement d'images ou de vidéos, comme la transformée en ondelettes.

2.1 Principes de l'analyse multirésolution

Le principe de ces méthodes est de réaliser une décomposition réversible d'un maillage, à l'aide de deux filtres appliqués en cascade. Durant la phase d'analyse, un filtre passe-bas (représenté par la lettre A sur la figure 1) et un filtre passe-haut (lettre B sur la figure 1) sont appliqués sur le maillage initial produisant respectivement une approximation plus grossière et un ensemble de détails haute-fréquence.

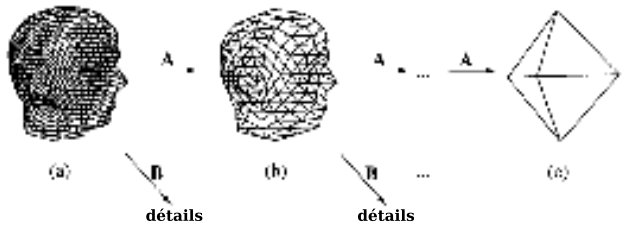


Figure 1 – *Décomposition en ondelettes d'un maillage surfacique triangulaire. Image tirée de [2].*

L'analyse multirésolution pour les maillages de topologie arbitraire a été introduite par Lounsbery et al. [2] qui ont choisi d'appliquer une technique de raffinement en partant d'un maillage très simple. Pour cela, ils ont utilisé une subdivision canonique des facettes ainsi qu'une transformée en ondelettes de seconde génération. L'atout principal des ondelettes est d'éliminer une grande partie de la redondance présente dans les signaux.

2.2 Les ondelettes de seconde génération

L'analyse multirésolution produit une décomposition de l'espace en une somme de sous-espaces imbriqués. Ainsi pour chaque niveau de résolution le maillage grossier et les détails sont obtenus respectivement par projection sur une base de fonctions d'échelles et d'ondelettes. Le maillage reconstruit par synthèse est considéré comme la meilleure approximation du modèle original au sens des moindres carrés si ces fonctions sont toutes orthogonales entre elles. Or cette orthogonalité est souvent difficile à obtenir avec des outils d'analyse par bancs de filtres basés sur la transformée de Fourier. C'est pourquoi les ondelettes de seconde génération, basées sur des arguments purement spatiaux, sont très utilisées pour les maillages. Le procédé de construction de ces ondelettes appelé lifting (introduit par Sweldens [3]) permet d'élever l'ordre de l'ondelette utilisée et consiste à intervertir les phases de filtrage et de sous-échantillonnage utilisées lors de l'analyse par bancs de filtres. On limite ainsi le nombre d'opérations à effectuer et de plus l'étape de synthèse est simplement obtenue par inversion des signes et de l'ordre des filtres d'analyse.
Pour pouvoir traiter les maillages surfaciques par ondelettes, ils sont considérés alors non plus comme des objets géométriques mais comme des fonctions via une paramétrisation de celles-ci. Cette paramétrisation doit alors tenir compte du fait que l'extension de l'analyse multirésolution introduite par Lounsbery et al. [2] ne fonctionne que sur des maillages possédant une topologie particulière. Une fois cette paramétrisation déterminée, elle est alors utilisée par la phase de remaillage afin de construire un maillage semi-régulier approchant l'objet initial et possédant une topologie propice à l'application d'une décomposition en ondelettes. Les travaux en analyse multirésolution utilisant cette phase de remaillage se différencient par la façon de construire le maillage semi-régulier.

2.3 Remaillage construit par raffinement

Lounsbery et al. [2] ont d'abord proposé une technique de remaillage par raffinement d'un modèle très simple (un octaèdre par exemple). Mais il faut alors plusieurs itérations avant d'aboutir à une forme ressemblant à l'objet de départ. Afin de réduire le nombre d'itérations nécessaires lors de la reconstruction, Turk [4] a proposé de partir d'un maillage ressemblant plus à l'objet initial, qu'il construit en répartissant un nombre limité de points sur la surface de départ. Le nuage obtenu est ensuite retriangulé en préservant la topologie de l'objet. Une démarche similaire est utilisée par Eck et al. [5] qui partitionnent le modèle initial en cellules de Voronoï. La triangulation de Delaunay permet alors de construire le maillage grossier. Enfin, une amélioration de ces techniques, qui repose sur une paramétrisation respectant les caractéristiques géométriques et les propriétés visuelles du maillage, a été mise au point par Gioia et al. [6]. Celle-ci leur permet d'obtenir en moyenne 2 fois moins de coefficients d'ondelettes sur des objets naturels.

2.4 Remaillage construit par décimation

Il existe un autre type de remaillage qui consiste premièrement à appliquer une simplification séquentielle

du maillage original et ensuite à raffiner régulièrement le résultat par subdivision. Plusieurs types de simplifications séquentielles ont été introduites, dont les plus célèbres sont décrites dans [7, 8, 9]. Les techniques d'analyse multirésolution utilisant ce procédé de remaillage, se distinguent également par le schéma de subdivision utilisé qui peut être de nature approximante ou interpolante. La plupart de ces méthodes utilisent le schéma de "butterfly" lifté [2, 10] qui permet de mieux contrôler la surface résultante qu'en utilisant un schéma approximant. Mais le schéma approximant de Loop utilisé par Khodakovski et al. [11], prenant en compte un plus large voisinage que le schéma "butterfly", donne des courbes de reconstruction équivalentes en terme de taux de distorsion avec un aspect visuel globalement meilleur.
Enfin l'un des codeurs les plus efficaces actuellement [12] utilise un schéma de "butterfly" non lifté et concentre la quasi-totalité de l'information géométrique dans les composantes normales des coefficients d'ondelettes. Le fait que le schéma lifting ne donne pas de meilleurs résultats en terme de fidélité géométrique et de codage, est dû à la technique de remaillage utilisée.
Les méthodes citées précédemment proposent toutes, en plus de l'analyse, une chaîne complète de compression qui est illustrée par le schéma de la figure 2. Les résultats en terme de compression sont fortement conditionnés par le remaillage et la transformée en ondelettes choisis. Mais l'étape de quantification des coefficients est également importante pour obtenir des taux de compression intéressants. Pour preuve, Payan et al. [13] obtiennent une meilleure qualité visuelle de reconstruction que Khodakovsky et al. [12] à débit similaire grâce à une optimisation de l'étape de quantification. Cela leur permet en effet de minimiser l'erreur de reconstruction sous la contrainte d'un débit fixé.

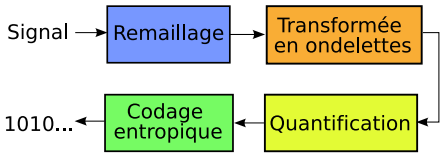


Figure 2 – *Principales étapes intervenant dans toute chaîne de compression avec pertes de maillages utilisant une transformée en ondelettes.*

Nous avons vu que, lors de la phase d'analyse, toutes ces méthodes appliquent une décomposition globale de l'objet. Or il peut être intéressant de chercher à traiter localement certains modèles, aussi bien en terme de décomposition en ondelettes que de quantification des coefficients surtout si leur surface est très peu uniforme.

3 Méthode d'analyse proposée

Notre approche d'analyse multirésolution repose sur la mise en place de plusieurs schémas de subdivision de nature interpolante qu'il est possible d'associer avec différents types de transformées en ondelettes afin d'obtenir une décomposition adaptée localement aux différents aspects surfaciques des maillages.

3.1 Schémas de subdivision envisagés

Les schémas de subdivision retenus consistent d'une part à ajouter un sommet au milieu de chaque arête (transformation topologique) et d'autre part à appliquer à ces nouveaux sommets un masque de lissage tenant compte du voisinage (transformation géométrique).
Les surfaces de subdivision ont été ici retenues car elles permettent de définir facilement un schéma multirésolution et de bénéficier d'une représentation hiérarchique utile pour le codage et la transmission de modèles 3D. Elles sont très utilisées conjointement avec une transformée en ondelettes pour le codage de surfaces naturelles. Les coefficients d'ondelettes renferment alors les détails qui n'ont pas pu être pris en compte par la subdivision seule.
Les schémas interpolants sont les plus utilisés pour les maillages surfaciques triangulaires, car ils génèrent des matrices d'analyse creuses, applicables en temps linéaire. Nous avons ainsi cherché à comparer la décomposition produite pour les schémas suivants :

- le schéma "midpoint" où chaque nouveau sommet est ajouté au milieu de chaque arête ;
- le schéma "butterfly" de Dyn et al. [14] produisant une surface limite C^1 pour les maillages de topologie régulière. Ces pondérations sont indiquées en noir sur la partie droite de la figure 4 ;
- deux améliorations du schéma précedent, la première que nous avons mise au point et la seconde proposée par Zorin et al. [15]. Ces deux méthodes sont décrites dans la suite de ce paragraphe.

Le schéma de subdivision de "butterfly", introduit par Dyn et al. [14] est le schéma interpolant possédant le plus petit support, mais il produit des artéfacts indésirables sur des topologies irrégulières (visibles sur la figure 3).

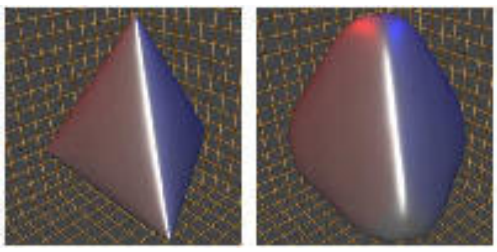


Figure 3 – *Subdivision d'un tétraèdre avec le schéma de "butterfly" de Dyn et al. [14] (à gauche) et l'amélioration de Zorin et al. [15] (à droite). Image tirée de [15].*

Pour éviter ces artéfacts, nous avons tout d'abord proposé un nouveau schéma qui conserve le masque introduit par

Dyn et al. [14] pour le déplacement des nouveaux sommets ayant un voisinage régulier et qui utilise le masque illustré par la partie gauche de la figure 4 dans tous les autres cas. Ce masque est tiré de la subdivision interpolante de Loop [16] et permet de s'adapter à tout type d'irrégularité de part son support restreint, tout en produisant une surface résultante globalement plus lisse.
L'amélioration proposée par Zorin et al. [15] introduit également de nouveaux masques pour les sommets irréguliers, tout en conservant la simplicité et le comportement du schéma originel. Les pondérations de ces masques, qui prennent en compte l'ensemble des voisins de chaque sommet irrégulier, ont été calculées à l'aide d'une transformée de Fourier discrète ainsi qu'une analyse en composantes principales. Ce schéma de subdivision, contrairement aux schémas interpolants classiques, donne des résultats comparables aux surfaces obtenues par des techniques approximantes en très peu d'itérations.
A la différence de l'extension proposée par Zorin et al. [15], qui utilise des pondérations adaptées à la valence, nous utilisons un unique masque pour l'ensemble des sommets irréguliers, afin d'obtenir un gain en temps d'exécution ainsi qu'une d'homogénéisation du traitement des irrégularités.

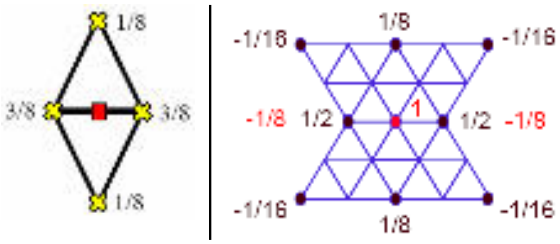


Figure 4 – *A gauche : masque utilisé dans le schéma interpolant de Loop pour le déplacement des nouveaux sommets. A droite : masques du schéma de "butterfly" lifté tirés de [10]. En noir le masque de prédiction, en rouge (clair) le masque de mise à jour.*

3.2 Transformées en ondelettes utilisées

Les différentes transformées en ondelettes de seconde génération utilisées conjointement aux schémas de subdivision permettent de coder les détails sous forme de vecteurs géométriques 3D. Elles consistent à appliquer les trois grandes étapes suivantes ou seulement les deux premières selon que l'on parle de schéma lifté ou non lifté :

- une opération de séparation du signal en composantes paires et impaires par l'utilisation d'ondelettes paresseuses (lazy wavelets), qualifiée de transformée polyphase ;
- une opération de prédiction (appelée aussi lifting dual et représentée par la lettre P sur les figures 5 et 6) qui prédit les échantillons de rang pair à partir des échantillons de rang impair ;
- une opération de mise à jour (appelée aussi lifting primal et représentée par la lettre U sur les figures 5 et 6) qui permet de conserver sur une partie du signal la valeur moyenne de l'ensemble des informations.

Notre but étant d'obtenir les plus petits coefficients d'ondelettes possibles, nous proposons de comparer les transformées en ondelettes suivantes, qui peuvent être associées à chaque schéma décrit précédemment :

- transformée sans étape de mise à jour ;
- transformée utilisée par Lounsbery et al. [2] et plus récemment par Valette et al. [10], dont le principe est illustré sur 2 canaux par la figure 5 ;
- transformée introduite par Sweldens [3] et utilisée par Payan et al. [13], dont le principe est illustré sur 2 canaux par la figure 6.

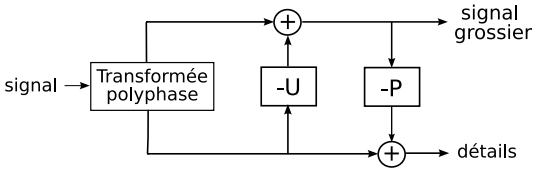


Figure 5 – *Principe de l'analyse du schéma lifting à 2 canaux tiré des travaux de Lounsbery et al. [2].*

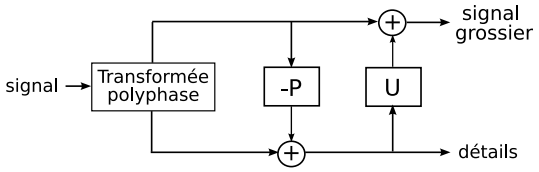


Figure 6 – *Principe de l'analyse du schéma lifting à 2 canaux introduit par Sweldens [3].*

4 Résultats

Pour effectuer une comparaison de ces différents masques et mener une analyse détaillée des décompositions en ondelettes, nous avons développé un outil d'analyse en C++ qui utilise la librairie géométrique CGAL (the Computational Geometry Algorithm Library). Il permet de visualiser, pour chaque niveau de résolution, l'approximation obtenue ainsi que l'amplitude et la direction des coefficients d'ondelettes sous forme de champs de vecteurs, comme le montre la figure 7.
Cette figure nous illustre également que les coefficients d'ondelettes sont principalement dirigés suivant la normale à la surface, ce qui montre bien l'intérêt de quantifier plus finement la composante normale lors d'une compression comme l'ont constaté Khodakovsky et al. [11] ainsi que Payan et al. [13].

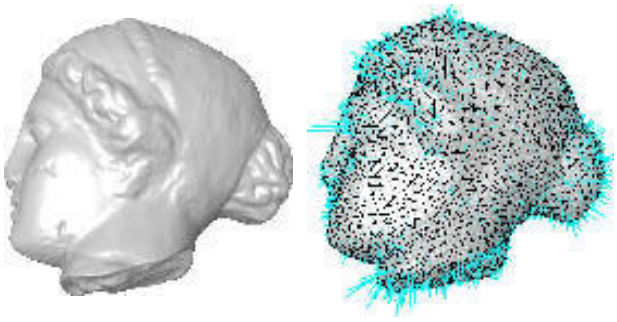


Figure 7 – *Aperçu des coefficients d'ondelettes sous forme de champs de vecteurs (en bleu sur l'image de droite) après l'analyse du modèle Venus (à gauche) sur 3 niveaux de décomposition. Les coefficients ont été multipliés par un facteur 10 pour une meilleure visualisation.*

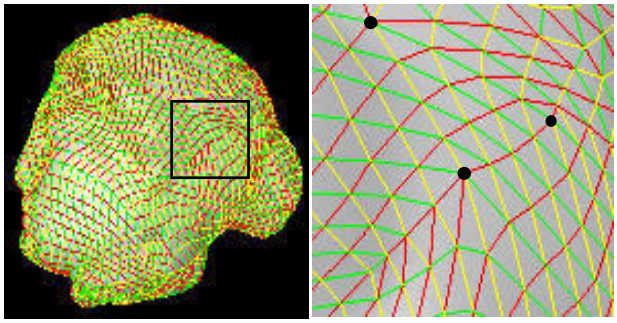


Figure 8 – *Illustration de la séparation en sous-bandes haute-fréquence sur l'approximation de droite de la figure 7. On remarque l'influence des sommets irréguliers (marqué en noir sur l'image de droite) qui produisent une déviation des directions caractérisées.*

On remarque enfin sur la figure 7 que les coefficients d'amplitude maximale sont situés au niveau des arêtes vives (sur le cou) et des détails haute-fréquence (yeux, chignon), puis que leur amplitude diminue au fur et à mesure que la surface devient lisse. Il est ainsi possible d'utiliser ces informations afin de procéder à une segmentation de l'objet en patchs surfaciques d'aspect plus ou moins rugueux. Cette segmentation permettrait alors de procéder à une décomposition en ondelettes adaptative, suivie d'une quantification spécifique pour chaque patch, afin d'obtenir une chaîne de compression locale.
Le schéma lifting étant basé sur une grille d'échantillonnage triangulaire "par arête", il est possible de séparer les détails haute-fréquence en 3 sous-bandes. Cette différenciation est symbolisée par les 3 couleurs distinctes de la figure 8. Ce traitement permet alors de se rapprocher des techniques utilisées en traitement d'images où les composantes horizontale, verticale et diagonale des coefficients d'ondelettes sont séparées afin de les quantifier différemment et d'exploiter leurs corrélations dans ces directions lors du codage entropique.

Dans le cas des maillages, il peut ainsi être intéressant de différencier le traitement des détails en suivant des directions privilégiées propres au maillage. L'algorithme proposé dans cette optique, traite tout d'abord les arêtes incidentes à un sommet régulier choisi aléatoirement sur le maillage, puis propage les informations obtenues vers les sommets réguliers voisins. Tant que le maillage est régulier, les directions des sous-bandes restent bien distinctes comme nous pouvons le voir sur la figure 8 à gauche des 3 points noirs, mais elles sont évidemment dépendantes du remaillage choisi. On remarque également qu'une déviation se produit généralement sous l'influence des sommets irréguliers, rendant difficile la caractérisation de directions privilégiées sur l'ensemble du maillage. Pour remédier à cela et ainsi obtenir des directions reliées à la courbure, la rugosité et la texture des objets, un remaillage adapté à ces caractéristiques surfaciques sera nécessaire.
Enfin, nous avons généré plusieurs objets plus ou moins bruités, regroupé sur la figure 9. Le modèle (b) a par exemple été construit par subdivision sur 4 niveaux d'un modèle de base et par ajout d'un bruit blanc additif uniforme d'amplitude $\pm$ 0.01 sur la version la plus fine (modèle (a) de la figure 10), $\pm$ 0.1 sur l'approximation intermédiaire (b) et $\pm$ 0.05 sur le modèle le plus grossier (c). L'analyse multirésolution du maillage (b) de la figure 9 est illustrée sur 3 niveaux par la figure 10.

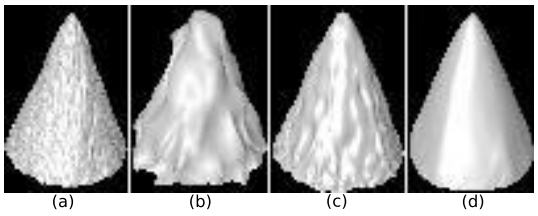


Figure 9 – *Aperçu des objets générés par notre application, par subdivision et ajout d'un bruit blanc additif uniforme.*

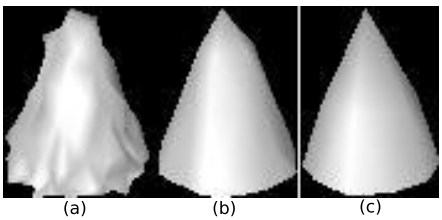


Figure 10 – *Illustration de 3 niveaux de décomposition en ondelettes sur l'objet (b) de la figure 9.*

Nous avons enfin analysé les coefficients d'ondelettes dans les 2 premiers niveaux de résolution décrits précédemment. Les histogrammes de la figure 11 montrent la répartition des coordonnées (x, y et z) de ces coefficients dans un repère local lié à la surface du modèle, la coordonnée z représentant leur composante normale. Nous pouvons re-

marquer que la différence d'amplitude du bruit généré lors de l'étape d'analyse en ondelettes se retrouve bien sur les histogrammes de la figure 11.

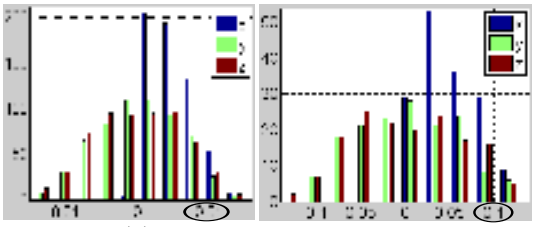


Figure 11 – *Histogrammes montrant la répartition des coordonnées locales x, y et z des coefficients d'ondelettes provenant de l'analyse de la figure 10 au premier (a) et au second (b) niveau de décomposition.*

5 Conclusion et perspectives

Nous avons proposé une méthode d'analyse permettant la comparaison de plusieurs schémas de décomposition et utilisant le pouvoir de décorrélation des ondelettes de subdivision. Nous avons également introduit un nouveau schéma de subdivision qui est une amélioration du schéma "butterfly" de Dyn et al. [14]. Les résultats obtenus sont encourageants car sa version liftée produit globalement des coefficients de plus faible amplitude qu'avec ce dernier, sur des objets naturels.

Nous avons enfin développé une application permettant la visualisation des coefficients d'ondelettes obtenus lors de l'analyse à différents niveaux de résolution ainsi que leur répartition sur 3 sous-bandes haute-fréquence. La séparation en sous-bandes proposée pourrait servir à éliminer une partie de la corrélation du signal non prise en compte par la plupart des méthodes actuelles.

Nous avons vu que l'analyse des coefficients d'ondelettes permet de caractériser les différents aspects de la surface des modèles 3D qui peut être plus ou moins lisse sur des objets naturels. Cette observation peut être exploitée pour réaliser une segmentation de ces objets en patchs.

Une fois cette étape de segmentation réalisée, il serait alors possible de proposer une décomposition et une quantification différente pour chaque type de patch afin d'aboutir à une méthode de compression efficace.

6 Remerciements

Ce travail est soutenu par France Télécom R&D Rennes dans le cadre du projet CoSurf (Compression de surface).

References

[1] JPEG 2000 Part 1 Final Draft International Standard. ISO/IEC FDIS15444-1. December 2000.

[2] M. Lounsbery, T. D. DeRose and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, 1997.

[3] W.Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998.

[4] G. Turk. Re-tiling polygonal surfaces. *Computer Graphics*, 26(2):55–64, 1992.

[5] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182, New York, NY, USA, 1995.

[6] P. Gioia. Reducing the number of wavelet coefficients by geometric partitioning. *Comput. Geom. Theory Appl.*, 14(1-3):25–48, 1999.

[7] H. Hoppe. Progressive meshes. *Computer Graphics*, 30(Annual Conference Series):99–108, 1996.

[8] P. Schroder L. Cowsar A. W. F. Lee, W. Sweldens and D. Dobkin. Maps: Multiresolution adaptive parameterization of surfaces. *Computer Graphics*, 32(Annual Conference Series):95–104, 1998.

[9] W. Sweldens I. Guskov, K. Vidimce and P. Schroder. Normal meshes. In *Siggraph 2000, Computer Graphics Proceedings*, pages 95–102, 2000.

[10] S. Valette. *Modèles de maillages déformables 2D et multirésolution surfaciques 3D sur une base d'ondelettes*. Thèse de doctorat, INSA Lyon, 2002.

[11] A. Khodakovsky, P. Schroder and W. Sweldens. Progressive geometry compression. In *Siggraph 2000, Computer Graphics Proceedings*, pages 271–278, 2000.

[12] A. Khodakovsky and I. Guskov. Compression of normal meshes. In *Geometric Modeling for Scientific Visualization. Springer-Verlag*, 2003.

[13] F. Payan. *Optimisation du compromis débit-distorsion pour la compression géométrique de maillages surfaciques triangulaires*. Thèse de doctorat, Université de Nice-Sophia Antipolis, décembre 2004.

[14] N. Dyn, D. Levin and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph.*, 9(2):160–169, 1990.

[15] D. Zorin, P. Schroder and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. *Computer Graphics*, 30(Annual Conference Series):189–192, 1996.

[16] C. T. Loop. Smooth subdivision surfaces based on triangles. Mémoire de D.E.A., Department of Mathematics, University of Utah, Salt Lake City, 1987.

Reconstructed Image by hypergeometric function Of Legendre

S. Benzzoubeir, H. Qjidaa, and A. Hmamade.
LESSI. Département de physique Faculté des sciences
Dhar El Mehraz B.P. 1796, Fès Atlas, Morocco
Sanaeben55@yahoo.fr
Qjidah@yahoo.fr

Abstract— **This paper introduces a new set of orthogonal moments function hypergeometric based on the discrete Legendre polynomials. The Legendre moments can be effectively used as pattern features in the analysis of two-dimensional images. The implementation of moments proposed in this paper does not involve any numerical approximation, since the basis set is orthogonal in the discrete domain of the image coordinate space. The paper presents the experimental results of Legendre moments with hypergeometric function and demonstrates their feature representation capability using the method of image reconstruction.**

Index Terms: **Discrete orthogonal systems, Image feature representation. Orthogonal moments Legendre.**

I. INTRODUCTION

THE function moments have been used as shape descriptors in a variety of applications in image analysis, like visual pattern recognition ,[1], [4], object classification [7], template matching [6], edge detection [5], pose estimation [13], robot vision[12], data compression [9] . In all these applications, geometric moments and their extensions in the form of radial and complex moments have played important roles in characterizing the image shape, and in extracting features that are invariant with respect to image plane transformations. Teague [18] introduced moments with orthogonal basis functions, with the additional property of minimal information redundancy in a moment set. In this class, Zernike moments have been extensively researched in the recent past, and several new techniques have emerged involving orthogonal moment based feature detectors [10], [14][19].In the following, we consider some of the major problems that are commonly encountered while implementing moment functions.

A. Two-dimensional Numerical Approximation of Continuous Integrals

The general two-dimensional (2-D) moment definition using a moment weighting kernel (also known as the basis function) $\psi_{pq}(x,y)$, and an image intensity function $f(x,y)$ is given as

$$\Psi_{pq} = \int_x \int_y \psi_{pq}(x,y) f(x,y) dxdy.$$

p, q=0,1,2… (1)

The integrals in the above equation are usually approximated by discrete summations, and this process not only leads to numerical errors in the computed moments, but also severely affects the analytical properties which they were intended to satisfy, such as invariance, orthogonal etc.

B. Coordinate Space Transformation

Orthogonal basis functions do not have the aforesaid problem of large dynamic range variation, but they generally have a domain which is completely different from the image coordinate space. For example, the Legendre and Tchebichef polynomials are valid only in the range [-1,1],while the Zernike radial polynomials are defined inside the unit circle. The Laguerre polynomials are defined in the range $[0,\infty[$, [2], [10], [11], [18].

The above problems motivate us to consider using discrete orthogonal polynomials as the basis set, and to define the corresponding moments directly on the image coordinate space. Since the implementation of discrete orthogonal moments does not involve any numerical approximations, the basis functions will exactly satisfy the orthogonal property, and thus yield a superior image reconstruction. Consider a discrete orthogonal system $\{f_n(i)\}$, where $a \le i \le b$.

The orthogonal property in the above domain can then be written as

$$\sum_{i=a}^{i=b} \omega(i) f_m(i) f_n(i) = \rho(n,a,b)\delta_{mn}. \quad (2)$$

Where $\omega(i)$ is the weighting function (also called the jump function), and $\rho(.)$ is the squared norm.

II. DISCRETE ORTHOGONAL MOMENTS

The following well-known theorem on orthogonal functions provides the mathematical basis for arriving at a definition for discrete orthogonal moments of an image intensity distribution $f(x,y)$: If $\{P_n(x)\}$ is a set of discrete

orthogonal polynomials with unit weight, satisfying the condition

$$\sum_{x=0}^{N-1} P_n(x)P_m(x) = \rho(n,N)\delta_{mn},$$

$$0 \le m,n \le N-1 \qquad (3)$$

Then any bounded function f(x, y), $0 \le \{x,y\} \le N-1$,has the following polynomial representation in terms of the functions $P_n(x)$

$$f(x,y) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \lambda_{mn} P_m(x)P_n(y), \qquad (4)$$

Where the coefficients moments λ_{pq} are given by

$$\lambda_{pq} = \frac{1}{\rho(p,N)\rho(q,N)} \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} P\left(\frac{2i-N+1}{N-1}\right) \times P\left(\frac{2j-N+1}{N-1}\right) f(i,j),$$

p, q=0, 1,2,…N-1 (5)

The above theorem can be generalized for orthogonal polynomials with weight $\omega(x)$,by replacing each orthogonal function $P_n(x)$ by the function $P_n(x)\sqrt{\omega(x)}$, in (3)–(5).

Equation (15) is easily obtained by substituting for $f(x,y)$ using (4) in the expression:

$\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} P_p(x)P_q(y)f(x,y)$, and noting that

$$\rho(p,N) = \sum_{p=0}^{N-1}\left\{P_p\left(\frac{2p-N+1}{N-1}\right)\right\}^2, \qquad (6)$$

Conversely, (4) follows from (5). In the context of image moments, it means that if we define a discrete orthogonal moment function as in (5) with $\{P_n(x)\}$ as the basis set, then the image may be reconstructed from the moments, using (4) as the inverse moment transform. The moment definition as given in (5) completely eliminates the need for any approximation of continuous integrals, and does not require coordinate space transformations.

We propose a modified version of Legendre polynomials as a convenient set of discrete orthogonal basis functions with unit weight, for defining moments of the above type.

The discrete generalized Legendre polynomial [1], [3],[8] can be defined as

$$P_n(x) = {}_2F_1\left(-n,n+1;1;\left(\frac{1-x}{2}\right)\right)$$

$$x \in [-1,1], \qquad \text{n,} = 0,1,2,\ldots \text{N-1}, \qquad (7)$$

With ${}_2F_1(.)$ is the generalized hyper geometric function

$${}_2F_1(a,b;c;z) = \sum_{k}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!} \qquad (8)$$

From the relation (8), I give a new definition of the polynomials of Lgendre in the discrete base of the hypegeometric functions.

$$P_n(x) = \sum_{k=0}^{N-1} (-1)^k \frac{(k+n)!}{(k!)^2(n-k)!}\left(\frac{1-x}{2}\right)^k, \qquad (9)$$

The Legendre polynomials satisfy the property of orthogonal (3), with

$$\rho(n,N) = \sum_{l=0}^{2n}\sum_{j=0}^{l} \frac{(-1)^l}{(N-1)^l} \times C^{n}{}_{n+(l-j)} \times C_n{}^{(l-j)} \sum_{i=0}^{N-1} (i)^l, \qquad (10)$$

And the following recurrence formula holds:

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x),$$

n=2, 3, 4…., (11)

The equation (5) also leads to the following inverse moments transform:

$$f(x,y) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \lambda_{mn} P_n(x)P_m(y)$$

x, y=0,1,…N-1. (12)

EXPERIMENTAL RESULTS

This section presents the test data and results used to validate the theoretical framework presented above, and also to establish the feature representation capability of Legendre moments with hypergeometric function through image reconstruction. A multi- level real image of "LENA" (see Fig 2) on a 100x100 pixel.

The sequence of reconstructed images, as the maximum order of moments used in the reconstruction is varied from one to 40, is shown in (Fig.1).We used the following formula to characterize the MSE between an input A multi level real image $f(x,y)$, and the reconstructed image $\hat{f}(x,y)$.

$$MSE = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left|f(x,y)-\hat{f}(x,y)\right|^2 . \quad (13)$$

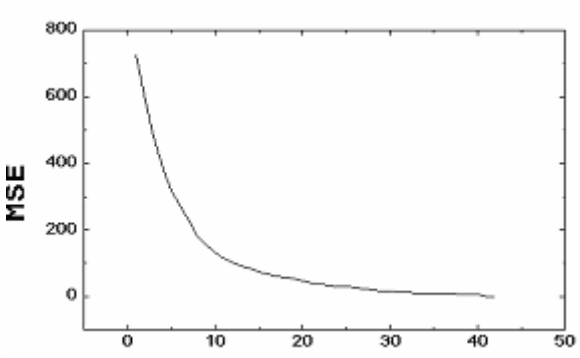


Fig.1. Order of rebuilding

III. CONCLUSION

A new set of discrete orthogonal moment features based on Legendre polynomial with the hypergeometric function has been proposed in this paper. The basis functions are orthogonal in the domain of the image coordinate space, and this feature completely eliminates the need for any discrete approximation in their numerical implementation.

Experimental results conclusively prove the effectiveness of Legendre moments with the hypregemetric function as the feature descriptors.

REFERENCES

[1] G. Szego, "Orthogonal polynomials "Pattern *Recognit,* vol. 23, pp. 100– 110, 1975.

[2] (Iyanaga and Kawada 1980, p. 1481; Zwillinge.

[3] Sonine 1880, p. 41; Whittaker and Watson 1990

[4] J. Flusser, "Pattern recognition by affine mome

[4] J. Flusser, "Pattern recognition by affine mome invariants," *Pattern Recognit.*, vol. 26, no. 1, pp. 167–174, 1993.

[5] S. Ghosal and R. Mehrotra, "Orthogonal moment operators for sub pixel edge detection," *Pattern Recognit.*, vol. 26, no. 2, pp. 295–306,1993.

[6] A. Goshtasby, "Template matching in rotated images," *IEEE Trans Pattern Anal. Machine Intell.*vol. PAMI-7, pp. 338–344, Mar. 1985.

[7] M. I. Heywood, "Fractional central moment method for moment- invariant object classification," *Proc.Inst. Elect. Eng.*, vol. 142, no.4, pp. 213–219, 1995. 174, 1993.

[8] F. Hilderbrand, *Introduction to Numerica l Analysis*. New York: Mc-Graw-Hill, 1956.

[9] H. S. Hsu, "Moment preserving edge detection and its application to image data compression," *Opt. Eng.*, vol. 32, no. 7, pp. 1596–1608, 1993.

[10] A. Khotanzad, "Invariant image recognition by Zernike moments," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. 12, pp. 489–497, May 1990.

[11] S. X. Liao and M. Pawlak, "On image analysis by moments," *IEEE Trans. Pattern Anal. Machine Intell.* vol. 18, pp. 254–266, Mar. 1996.

[12] V. Markandey and R. J. P. Figueiredo, "Robot sensing techniques based on high dimensional moment invariants and tensors," *IEEE Trans. Robot. Automat.*, vol. 8, pp 186–195, Feb. 1992.

[13] R. Mukundan, "Estimation of quaternion parameters from two dimensional image moments," *Graph. Models Image Process,* vol. 54, no. 4, pp. 345–350, 1992.

[14] R. Mukundan and K. R. Ramakrishnan, "Fast computation of Legendre and Zernike moments," *Pattern Recognit.*, vol. 28, no. 9, pp. 1433–1442, Sept. 1995.

[15] *Moment Functions in Image Analysis-Theory and Applications*. Singapore: World Scientific, 1998.

[16] A. V. Nikiforov, S. K. Suslov, and V. B. Uvarov, *Classical Orthogonal Polynomials of a Discrete Variable*. Berlin, Germany: Springer-Verlag, 1991.

[17] G. Szego, *Orthogonal Polynomials*, 4th ed. New York: Amer. Math. Soc., 1975, vol. 23.

[18] M. R. Teague, "Image analyis via the general theory of moments," *J.Opt. Soc. Amer.*, vol. 70, no. 8, pp. 920–930, 1980.1959.

[19] A. Wallin, "Complete sets of complex Zernike moment invariants and the role of pseudo invariants," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 1106–1110, Nov. 1995.

Various orders of rebuilding of image “LENA" by Legendre.

Fig.2

Ordre de couleurs : une approche par graphe

O. Lezoray C. Meurie A. Elmoataz
LUSAC EA 2607, IUT SRC, 120 Rue de l'exode, 50000 SAINT-LÔ, FRANCE
olivier.lezoray@unicaen.fr,cmeurie@info.unicaen.fr,abder.elmoataz@greyc.ensicaen.fr

Résumé

Cet article présente une nouvelle approche d'ordre de données vectorielles. Nous nous intéressons ici plus particulièrement au cas des images couleur. L'ordre que nous proposons permet de pallier les défauts des ordres vectoriels classiques. Celui-ci est construit sur un voisinage de pixels et non défini a priori. L'approche que nous proposons est basée d'une part sur l'extraction des infimum et suprememum d'un ensemble de couleurs puis d'autre part sur la construction de l'ordre à partir de l'infimum. L'ordre ainsi construit défini un chemin hamiltonien sur le graphe non orienté totalement connecté représentant l'élément structurant.

Mots clefs

Ordre vectoriel, graphe, couleur, filtrage.

1 Introduction

En traitement d'images, dès lors que l'on s'intéresse à des images multivariées, se pose le problème de l'extension des algorithmes usuels pour le traitement des images en niveaux de gris. Ceci n'est pas un problème récent et de nombreux auteurs se sont penchés sur le cas des images couleur qui représente le cas le plus fréquent d'images multivariées [1]. La majorité des premiers travaux a concerné l'extension du filtre médian aux images couleur [2]. Dans le cas des images en niveaux de gris, à chaque pixel est associé une valeur dans $\mathbb{R}$ et la relation $<$ étant un ordre total sur $\mathbb{R}$ il est facile d'ordonner les valeurs d'un ensemble de pixels suivant cette relation d'ordre. Puisque chaque composante d'une image couleur peut être considérée comme une image en niveaux de gris, un filtre médian peut être appliqué sur chaque composante séparément. Ce type d'approche est appellé traitement marginal. Le traitement marginal n'exploite pas la corrélation existante entre les différentes composantes (particulièrement importante pour le cas des images couleur) et peut provoquer l'apparition de fausses couleurs. Un traitement vectoriel est donc plus approprié afin de prendre en compte la nature vectorielle des données. Cependant, cela nécéssite la définition de nouveaux modèles pour l'extension des algorithmes scalaires au cas vectoriel. Le principal problème de ce genre d'extension repose sur la définition d'un ordre vectoriel approprié car il n'existe pas d'ordre naturel sur les vecteurs. Dans cet article nous proposons une approche de construction d'un ordre vectoriel. Nous utilisons ici le terme de contruction car l'ordre que nous proposons n'est pas défini *a priori* mais dépend des données à ordonner. Dans la suite de cet article nous rappelons la notion d'ordre et les différents ordres vectoriels de la littérature. Nous présentons ensuite le lien qu'il existe entre un ordre vectoriel et la notion de filtrage. A partir des constatations effectuées, nous proposons une approche par graphe qui permet d'ordonner un ensemble de couleurs selon une relation d'ordre découverte à partir des données. Cette relation d'ordre est obtenue par décimation de l'arbre de recouvrement minimum selon le degré des noeuds du graphe. Nous présentons quelques résultats de comparaison de notre ordre vectoriel avec les ordres usuels.

2 Ordre vectoriel

Les trois ordres vectoriels importants sont le pré-ordre, l'ordre partiel et l'ordre total. Rappelons tout d'abord les définitions utiles à la caractérisation d'une relation d'ordre. Soit R une relation binaire sur un ensemble quelconque A. R est réflexive ssi $\forall x \in A, xRx$. R est transitive ssi $\forall x, y, z \in A, xRy$ et $yRz \Rightarrow xRz$. R est anti-symétrique ssi $\forall x, y \in A, xRy$ et $yRx \Rightarrow x = y$. Une relation binaire R sur un ensemble A est un pré-ordre ssi R est réflexive et transitive. Par exemple, pour les relations d'ordre sur $\mathbb{R}$, la relation binaire R est en général la relation $<$ ou $\leq$. Une relation binaire R sur un ensemble A est un ordre partiel ssi R est réflexive, transitive et anti-symétrique. Un ordre partiel est totalement ordonné ssi $\forall x, y \in A, xRy$ ou yRx. Cette dernière définition caractérise un ordre pour lequel il n'existe pas de paire de membres de l'ensemble A non ordonnés. On appelle un ordre partiel qui est totalement ordonné un ordre total. Selon Barnett [3], on peut distinguer plusieurs méthodes pour ordonner des données vectorielles : l'ordre marginal, l'ordre réduit, l'ordre partiel et l'ordre conditionnel ou lexicographique. Nous rappelons leurs principes. Soit $x_1, x_2, \cdots, x_n$ un ensemble de n vecteurs de dimension p, avec $x_i = \{x_i^1, x_i^2, \cdots, x_i^p\}, x_i \in \mathbb{R}^p$.

- L'ordre marginal : les composantes vectorielles sont ordonnées selon chaque dimension p indépendamment. Pour deux vecteurs x_i et x_j, on a $x_i \leq x_j \Leftrightarrow x_i^k \leq x_j^k, \forall k \in \{1, 2, \cdots, p\}$. Cet ordre est un ordre partiel.
- L'ordre réduit : chaque vecteur est réduit à un scalaire et les données vectorielles sont triées selon l'ordre des valeurs scalaires obtenues. Avec une fonction scalaire

$d : \mathbb{R}^p \rightarrow \mathbb{R}$ utilisée pour trier les vecteurs, on obtient pour deux vecteurs x_i et x_j : $x_i \leq x_j \Leftrightarrow d(x_i) \leq d(x_j)$. Cette approche est très utilisée notamment dans les filtres médians vectoriels couleur où la fonction scalaire est une fonction de distance entre les pixels couleurs.

– L'ordre partiel : les vecteurs sont regroupés en sous-ensembles qui sont ensuite ordonnés, cet ordre est fondé sur la structure géométrique des vecteurs et notamment sur leur enveloppe convexe.
– L'ordre conditionnel ou lexicographique est basé sur la relation d'ordre suivante. Pour deux vecteurs x_i et x_j :

$$x_i \leq x_j \begin{cases} x_i^1 < x_j^1 \text{ , ou} \\ x_i^1 = x_j^1 \text{ , et, } x_i^2 < x_j^2 \text{ ou} \cdots \\ x_i^1 = x_j^1 \text{ , et, } x_i^2 = x_j^2 \cdots \; x_i^p < x_j^p \end{cases}$$

Cet ordre est un ordre vectoriel total mais il introduit une forte dissymétrie entre les composantes.

On trouve également d'autres ordres dans la littérature, nous ne les détaillons pas ici et nous ne retenons que l'ordre par entrelacement de bits pour ses propriétés intéressantes. Celui-ci, proposé par Chanussot [4], est un ordre total et constitue un apport important dans la littérature concernant les ordres vectoriels. Il s'appuie sur une représentation binaire des composantes codées sur 8 bits en RVB et construit un scalaire codé sur 24 bits en entrelacant de façon symétrique chacun des bits des composantes.

3 Ordres et filtres

Un des champs d'application très important du traitement d'images multivariées et plus précisémment d'images couleur est le filtrage. Il est bien établi à présent que le traitement d'images couleur doit se faire en tenant compte de la nature vectorielle des données étant donné la corrélation qui existe entre des composantes couleur. Nous insistons, dans cette section, sur le lien entre ordre vectoriel de couleur et deux familles de filtres que sont les filtres morphologiques et les filtres médians. Chacune de ces familles est basée sur un ordre de couleurs spécifique qui présente certains avantages ou inconvénients que nous présentons en détail.

3.1 Filtres morphologiques vectoriels

La morphologie mathématique est une approche du traitement d'images qui est basée sur une structure fondamentale, le treillis complet, $\mathcal{L}$ [5] tel qu'une relation d'ordre $\leq$ est définie sur $\mathcal{L}$ et tel que pour chaque sous ensemble fini $\mathcal{K}$ de $\mathcal{L}$, existent un supremum $\vee\mathcal{K}$ et un infimum $\wedge\mathcal{K}$. Pour pouvoir appliquer la morphologie mathématique aux images couleur et construire un treillis complet, il est nécessaire de pouvoir ordonner les couleurs et de vérifier l'existence des supremum et infimum. Pour certains ordres vectoriels, le supremum et l'infimum d'un ensemble de vecteurs ne font pas toujours partie de cet ensemble. Dans le cas spécifique des images couleur, ce problème se manifeste par l'introduction de fausses couleurs par les opérateurs morphologiques. Une contrainte supplémentaire à la définition de filtres morphologiques est donc que le supremum ($\vee$) et l'infimum ($\wedge$) d'un ensemble fassent partie de celui-ci. Ceci n'est pas vrai pour tous les ordres et l'ordre marginal introduit par exemple de fausses couleurs. C'est pourquoi les ordres habituellement utilisés en morphologie mathématique sont des ordres totaux. Dans ce cadre, les principaux ordres de couleur sont l'ordre lexicographique [6, 7] et l'ordre par entrelacement de bits [4] qui permettent d'imposer que le supremum et l'infimum appartiennent au treillis $\mathcal{L}$. L'ordre lexicographique est fortement dissymétrique et quand un ordre lexicographique est utilisé avec des opérateurs morphologiques dans un espace couleur quelconque, on trouve que la plupart des décisions sur l'ordre des vecteurs dans un élément structurant sont prises au premier niveau de la relation d'ordre. Ceci donne des opérateurs qui ne sont pas homogènes dans leur traitement de l'espace. Ceci est un inconvénient majeur pallié par l'utilisation d'espaces de type luminance/teinte/saturation [7, 8]. L'ordre par entrelacement de bits permet quand à lui de limiter la dissymétrie entre les composantes, mais étant donné qu'il est basé sur un entrelacements de bits, il n'est utilisable que pour des espaces où les composantes sont codées en entiers. Ceci n'est pas le cas de beaucoup d'espaces couleur. Cet ordre est donc principalement conçu pour l'espace RVB. Une fois l'ordre de couleurs choisi, on peut appliquer les deux principales opérations de morphologie mathématique à savoir l'érosion ϵ et la dilatation δ. On peut ensuite obtenir un certain nombre d'opérations morphologiques par composition de ces deux opérations élémentaires.

3.2 Filtres médians vectoriels

Le filtre vectoriel le plus populaire est le filtre médian vectoriel (VMF) [2]. Le VMF est un opérateur vectoriel qui a été introduit comme une extension du filtre médian scalaire. Ce filtre est basé sur un ordre de vecteurs sur un élément structurant donné. La sortie de ce type de filtre est définie comme le vecteur de plus faible rang selon un ordre réduit [9] basé sur les distances entre couleurs. Nous en détaillons le principe. Sur un élément structurant donné, on obtient un ensemble W de n vecteurs $x_1, x_2, \cdots, x_n$ de dimension p, avec $x_i = \{x_i^1, x_i^2, \cdots, x_i^p\}, x_i \in \mathbb{R}^p$. En général, la différence entre deux vecteurs x_i et x_j peut être mesurée par une distance de Minkowski $\|x_i - x_j\|_\gamma = \left(\sum_{k=1}^{p} \left\| x_{k(i)} - x_{k(j)} \right\|^\gamma \right)$ où γ désigne la norme employée qui est habituellement une norme euclidienne ($\gamma = 2$). A partir d'un ensemble de vecteurs, le filtre VMF définit un ordre réduit où chaque vecteur est réduit à une valeur scalaire définie par $d_i = \sum_{k=1}^{n} \|x_i - x_k\|$. Ceci revient à calculer pour chaque vecteur de l'ensemble la somme des distances à tous les autres vecteurs de l'ensemble. Les valeurs d_i des vecteurs x_i sont ensuite triées en ordre ascendant et cet ordre des distances ($d_{(1)} \leq d_{(2)} \leq \cdots \leq$

$d_{(n)}$) est utilisé pour trier les vecteurs de l'ensemble initial ($x^{(1)} \leq x^{(2)} \leq \cdots \leq x^{(n)}$). La sortie d'un filtre VMF est donc le vecteur qui minimise la distance aux vecteurs de l'ensemble constituant l'élément structurant. La différence d'orientation entre deux vecteurs peut également être utilisée comme mesure de distance. Ce type de distance est à la base des filtres médians vectoriels directionels (VDF) [10]. Le VDF est un filtre dont la sortie est le vecteur qui minimise la somme des orientations avec les autres vecteurs de l'élément structurant. Ce type de filtre ne prenant en compte que l'information chromatique, un autre filtre d'ordre, appellé filtre directionnel de distance (DDF) combinant les deux critères [11], a été proposé. On trouve énormément d'autres améliorations de ce type de filtre médian vectoriel, ces améliorations portant essentiellement sur une pondération des distances (voir dans [1, 2] pour une revue complète). Bien que très utilisé, ce type d'ordre est un ordre réduit et ne fournit donc pas d'ordre total entre les vecteurs, ce qui ne permet pas son utilisation pour des filtres morphologiques, même si ils sont très utiles pour supprimer du bruit [1].

3.3 Constatations

La première constatation est que les ordres réduits utilisés pour les filtres médians ne sont pas adaptés au filtrage mophologique puisqu'ils ne définissent pas un treillis complet. Ils sont cependant adaptés à n'importe quel type d'image multivariée et à n'importe quel espace couleur. La seconde constatation est que les ordres utilisés pour le filtrage morphologique sont majoritairement des ordres totaux mais qui présentent le principal défaut de n'être adaptés qu'au traitement dans un espace couleur donné : RVB pour l'ordre par entrelacement de bits et HSI pour l'ordre lexicographique. De plus, ces ordres ne peuvent pas être utilisés sur des images multivariées : pour l'ordre par entrelacement, nous sommes limités au nombre de bits maximum pouvant coder un entier et pour l'ordre lexicographique, contrairement à ce qui est dit dans [12], il n'est absolument pas envisageable de l'utiliser pour des données de grande dimensions car déterminer l'ordre optimal parmi toutes les permutations possibles d'un ordre lexicographique est un problème NP-complet [13]. Tous les ordres usuels ayant des désavantages, nous cherchons à en concevoir un qui fonctionne sur des vecteurs de dimensions quelconques.

4 Ordre par graphe

Dans cette section, nous proposons donc une alternative aux approches classiques pour ordonner des couleurs. Trouver un ordre de vecteurs, sur un voisinage de pixels constitué par un élément structurant, revient à trouver un chemin hamiltonien dans un graphe non-orienté. Ce graphe a pour noeuds chacun des pixels et pour arêtes toutes les associations deux à deux des noeuds car on cherche à ordonner des pixels non forcément connexes (le graphe est complet). Trouver un chemin hamiltonien (un chemin passant par tous les sommets du graphe en ne passant qu'une fois par chaque noeud) est un problème NP-complet, nous proposons donc ici une méthode heuristique pour trouver un tel chemin. Nous rappelons tout d'abord ce que l'on veut obtenir. A partir d'un ensemble de pixels décrits par un vecteur de dimension p, on défini un ensemble non ordonné de vecteurs $x_1, x_2, \cdots, x_n$. On désire définir un ordre sur ces vecteurs, c'est-à-dire obtenir un ensemble ordonné $x^{(1)} \leq x^{(2)} \leq \cdots \leq x^{(n)}$. $x^{(1)}$ et $x^{(n)}$ doivent représenter respectivement l'infimum et le supremum de cet ensemble et $x^{(n+1)/2}$ le médian de cet ensemble. Ceci correspond exactement à la définition d'un ordre scalaire. Nous nous restreignons au cas de la couleur ($p = 3$) mais le principe reste identique pour des dimensions plus élevées.

4.1 Extraction de l'$\wedge$ et du $\vee$

L'idée principale de cette nouvelle approche est la suivante : un ordre total définit un chemin hamiltonien le long de tous les pixels de l'élément structurant. Considérons un élément structurant B et un graphe complet G_0 où chaque pixel de B est un noeud du graphe et où les noeuds sont tous liés deux à deux (connexité totale). Pour obtenir un ordre total de tous les vecteurs couleurs, nous avons besoin de simplifier le graphe G_0 afin d'obtenir un chemin partant de la borne inférieure $\wedge$ et arrivant à la borne supérieure $\vee$ de l'ensemble couleur. Ce chemin doit posséder deux propriétés. Il doit passer par tous les noeuds mais en ne les traversant qu'une seule et unique fois. Il doit aussi définir un ordre des vecteurs couleurs comme étant un treillis complet (un ordre total et la définition du $\vee$ et de l'$\wedge$). Il est difficile (problème NP-complet) de trouver un tel chemin parmi toutes les possibilités présentes dans le graphe G_0. C'est pourquoi, nous proposons de faire une approximation de ce chemin [14] en calculant l'arbre de recouvrement minimal [15] (« Minimum Spanning Tree : MST ») du graphe complet G_0 et où toutes les arêtes reliant deux noeuds sont valuées par une norme L_2 entre les vecteurs décrivant chaque noeud. L'arbre de recouvrement minimal n'est pas très loin de la solution que nous attendons [16]. La figure 1 illustre ce point, le MST du graphe est calculé et les noeuds peuvent être classés comme noeuds internes et externes (feuilles) selon leur degré. On rappele que le degré d'un noeud est son nombre de voisins. Un chemin est hamiltonien si deux noeuds sont de degré 1 et que tous les autres sont de degré 2. Donc, si le MST est un ordre des vecteurs couleurs, il définit un tel chemin parmi les noeuds : ce chemin possède seulement deux noeuds externes et chaque noeud interne est connecté à exactement deux noeuds. C'est rarement le cas, pourtant nous pouvons faire une supposition sur les propriétés des noeuds du MST puisque c'est une généralisation à des dimensions plus élevées d'une liste à une dimension triée [16]. Si un noeud du MST est un noeud interne, il ne peut pas être le $\vee$ ou l'$\wedge$ de l'ensemble des couleurs, car il aurait été considéré dans ce cas comme un noeud externe du chemin d'ordre des vecteurs couleurs. Pour trouver le $\vee$ et l'$\wedge$ de l'ensemble des vecteurs couleurs, nous pouvons utiliser

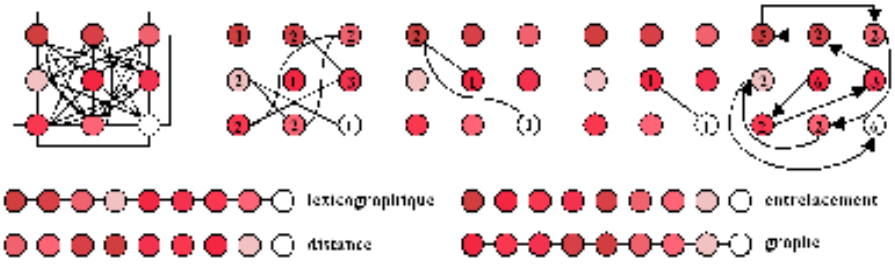


Figure 1 – *Ligne du haut : processus de décimation du MST du graphe (de gauche à droite, le graphe G_0, les MST successifs $MST(G_\lambda)$ avec $\lambda \in [0-2]$) et l'ordre de vecteurs construit. Lignes du bas : comparaison entre différents ordres.*

cette dernière propriété en construisant un nouveau graphe complet G_1 constitué uniquement des couleurs possibles : les noeuds externes du MST de G_0. Chaque noeud de G_1 est lié à tous ses autres noeuds candidats (les externes de MST(G_0)) et le MST de G_1 est calculé. Ce procédé peut être réitéré jusqu'à ce que le MST obtenu à l'itération i soit réduit à deux noeuds définissant respectivement le $\vee$ et l'$\wedge$. Cependant une dernière étape est à effectuer à savoir, définir quel noeud est le $\vee$ et par élimination l'$\wedge$. L'$\wedge$ est identifié comme étant le noeud se rapprochant le plus d'une couleur de référence C_{ref} [6] (généralement le noir). L'algorithme complet [14] est résumé dans l'algorithme 1 présenté ci-après.

end $\leftarrow$ **faux**
$\lambda \leftarrow 0$
Construire G_λ sur l'élément structurant B
Répéter
  Calculer MST(G_λ)
  $n = NombreNoeudsExternes(\text{MST}(G_\lambda))$
  Si (n=2) **Alors**
    end $\leftarrow$ **vrai** ;
  Sinon
    $\lambda \leftarrow \lambda + 1$
    $G_\lambda = NoeudsExternes(\text{MST}(G_{\lambda-1}))$
    Lier tous les noeuds de G_λ
  Fin Si
jusqu'à ce que (end= **vrai**)
$G_\lambda = (\{V_1, V_2\}, \{E_1\})$
$\vee$ =argmax $d(V_i, C_{ref})$ et $\wedge$ =argmin $d(V_i, C_{ref})$

Algorithme 1 – *Détermination du $\vee$ et de l'$\wedge$ d'un ensemble de vecteurs couleurs.*

La figure 1 illustre la détermination du $\vee$ et de l'$\wedge$: la première étape est la construction du graphe G_0, les autres étapes quant à elles présentent les $MST(G_\lambda)$ pour $\lambda \in [0-2]$ jusqu'à ce que le critère d'arrêt soit atteint. Sur chaque noeud candidat, à chaque niveau λ, est précisé son degré. Dans cet exemple, la couleur de référence est le noir, le $\vee$ est donc représenté par le pixel inférieur droit et l'$\wedge$ par le pixel central. Une fois ces deux limites déterminées, nous pouvons effectuer les opérations élémentaires de morphologie mathématique : l'érosion et la dilatation sur un voisinage de pixels considéré.

4.2 Construction de l'ordre

La méthode que nous venons de présenter ne construit pas un ordre des vecteurs couleurs d'un élément structurant mais permet d'en extraire ses deux vecteurs extrêmes. A partir de ces deux vecteurs extrêmes, nous disposons des deux bornes de l'ordre $\wedge = x^{(1)}$ et $\vee = x^{(n)}$. Il nous reste à présent à construire un ordre. Pour cela nous utilisons la cascade de MST générée lors de l'algorithme 1. Pour chaque niveau de décimation λ, nous pouvons associer à chaque noeud x_i son degré $deg_\lambda(x_i)$. Nous pouvons quantifier l'importance globale de chaque noeud en effectuant une somme de ces degrés : le degré global de chaque noeud est défini par $deg(x_i) = \sum_\lambda \lambda deg_\lambda(x_i)$. Pour effectuer un parcours du graphe initial G_0 en partant de l'$\wedge$ jusqu'au $\vee$, il suffit de sélectionner comme noeud suivant du chemin celui qui a le degré global minimum en prenant soin de ne passer qu'une fois par chaque noeud. Comme certains noeuds peuvent avoir le même degré global, nous pondérons celui-ci par la valuation de l'arête reliant les noeuds afin de sélectionner le noeud le plus proche en tenant également compte de la distance. En procédant ainsi on attendra naturellement le $\vee$ en dernier car son degré global est égal à celui de l'$\wedge$ et est supérieur à tous les autres. L'algorithme 2 présente la construction du chemin hamiltonien constituant l'ordre. $u \sim v$ signifie que le noeud u est un voisin de v et que u n'appartient pas déjà au chemin construit (on ne passe qu'une fois par chaque noeud). $w_{uv} = \|u - v\|$ est la valuation de l'arête entre deux noeuds u et v. Le chemin hamiltonien définissant l'ordre des pixels des couleurs de la figure 1 est précisé à droite de celle ci, les degrés globaux des noeuds étant superposés sur ceux-ci. Afin d'illustrer les différences entres les principaux ordres de la littérature, la figure 1 présente une comparaison des ordres obtenus respectivement par un ordre lexicographique, un ordre par entrelacement de bits, un ordre réduit basé sur les distances et notre approche par graphe. La comparaison est effectuée

sur les couleurs de l'élément structurant de la figure 1. Une analyse visuelle nous montre que deux ordres sont moins efficaces (lexicographique et par mesure de distance) et que les ordres par entrelacement de bits et par notre approche par graphe sont meilleurs et très proches (une seule différence ici).

$i \leftarrow 1$
$x^{(i)} \leftarrow \wedge$
Répéter
  $x^{(i+1)} \leftarrow arg \min_{x_k \sim x^{(i)}} \left(w_{x_k x^{(i)}} deg(x_k)\right)$
  $i \leftarrow i + 1$
jusqu'à ce que $(i = n)$

Algorithme 2 – *Construction du chemin reliant l'*$\wedge$ *et le* $\vee$ *dans un ensemble de vecteurs couleurs.*

5 Résultats et conclusion

La figure 2 présente une comparaison entres les ordres lexicographiques, par entrelacement de bits et avec notre approche par graphe (figures 2(b)-(g)) sur des opérations d'érosion et de dilatation en RVB. Les figures 2(h)-(k) présentent les différences entre les résultats obtenus. On peut constater tout d'abord qu'il est très difficile visuellement de quantifier la différence entre ces ordres, même si les images de différence nous montrent que notre ordre et l'ordre lexicographique sont proches. La figure 3 présente le même genre de résultat mais en comparant, sur une image bruitée, des filtres VMF et VDF utilisant un ordre réduit basé sur les distances avec un ordre basé sur notre approche par graphe. Dans le cas de notre approche, il suffit de changer la pondération des arêtes (distance ou angle) pour obtenir un filtre VMF ou VDF. De même que précédemment, les filtres VMF ou VDF par ordre de distance et notre ordre produisent des résultats très similaires. Cependant, ce qu'il faut remarquer, c'est que notre approche fait aussi bien que les ordres reconnus de la littérature tout en dépassant leurs principales limitations. En effet, notre ordre peut s'appliquer à des images codées dans n'importe quel espace couleur, à des images multispectrales contenant un nombre arbitraire même élevé de composantes et il permet d'effectuer, sous un même formalisme de base, des filtres morphologiques ou médians. En conséquence cet ordre peut s'utiliser facilement pour des filtres morphologiques plus complexes tels que les nivellements, des filtres médians pondérés, mais d'autres applications telles que le tri de couleurs dans un histogramme avant leur mise en correspondance. Ceci sera l'objet de nos futurs travaux ainsi que des comparaisons plus complètes entre les différents ordres.

Références

[1] K. Plataniotis et A.N. Venetsanopoulos. *Color Image Processing and Applications*. Springer Verlag, 2000.

[2] R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, et A.N. Venetsanopoulos. Vector filtering for color imaging. *IEEE Signal Processing Magazine, Special Issue on Color Image Processing*, 22(1) :74–86, 2005.

[3] V. Barnett. The ordering of multivariate data. *Journal of the royal society of statistics*, A 139(3) :318–355, 1976.

[4] J. Chanussot et P. Lambert. Total ordering based on space filling curves for multivalued morphology. Dans *Proc. of ISMM*, pages 51–58, 1998.

[5] C. Ronse. Why mathematical mophology needs complete lattices. *Signal Processing*, 21(2) :129–154, 1990.

[6] A. Hanbury et J. Serra. Mathematical morphology in the HLS colour space. Dans *Proc. of BMVC'2001*, volume 2, pages 451–460, 2001.

[7] J. Angulo. Unified morphological color processing framework in a lum/sat/hue representation. Dans *Proc. of ISMM)*, pages 387–396, 2005.

[8] G. Louverdis, M.I. Vardavoulia, I. Andreadis, et Ph. Tsalides. A new approach to morphological color image processing. *Pattern recognition*, 35(8) :1733–1741, 2002.

[9] J. Astola, P. Haavisto, et Y. Neuvo. Vector median filters. *Proceedings of the IEEE*, 74(4) :678–689, April 1990.

[10] P.E. Trahanias, D. Karakos, et A.N. Venetsanopoulos. Directional processing of color images : Theory and experimental results. *IEEE Transactions on Image Processing*, 5 :868–880, 1996.

[11] D.G Karakos et P.E. Trahanias. Combining vector median and vector directional filters : the directional-distance filters. Dans *Proceedings of the International Conference on Image Processing*, pages 171–174, 1995.

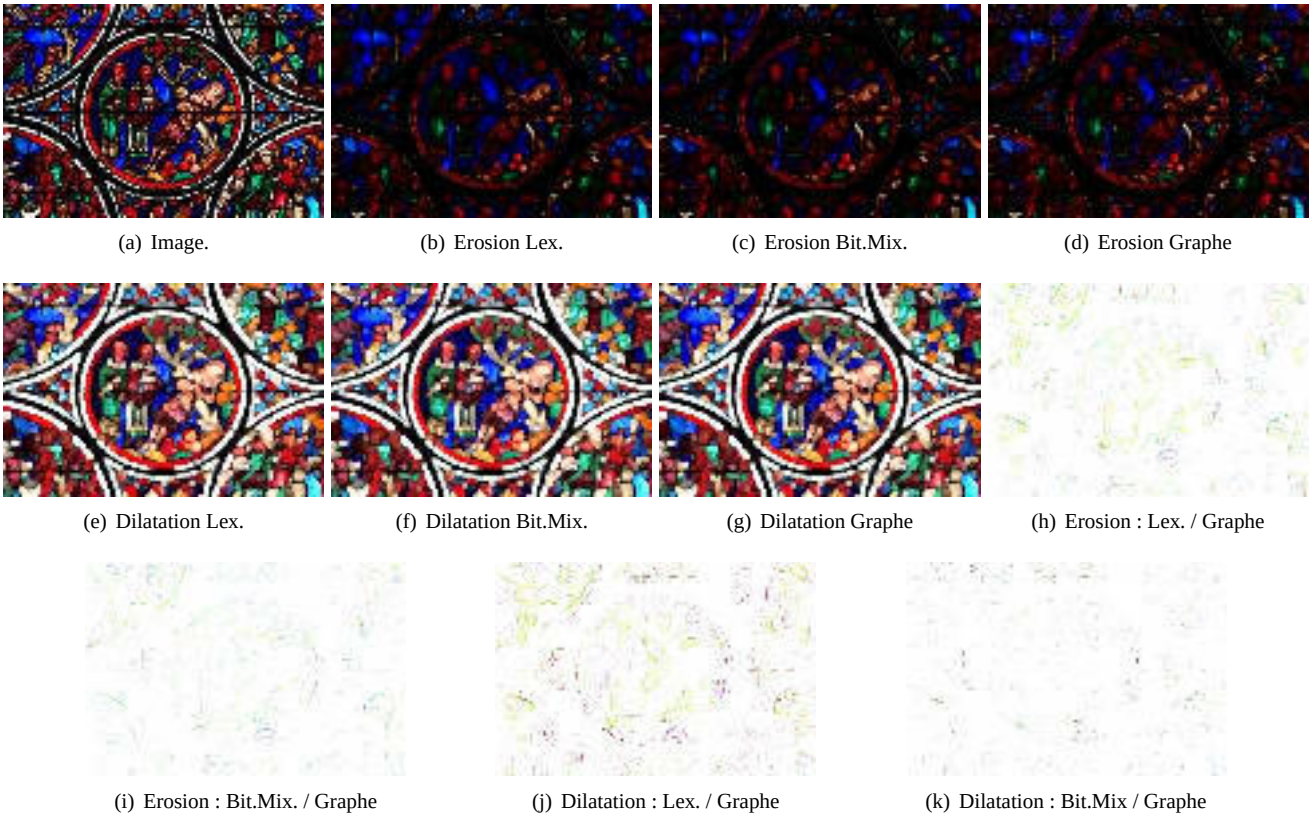
[12] J. Angulo. Morphological color processing based on distances. Application to color denoising and enhancement by centre and contrast operators. Dans *Proc. of VIIP*, pages 314–319, 2005.

[13] M. Schmitt et L. Martignon. On the complexity of learning lexicographic strategies. *JMLR*, 7 :55–83, 2006.

[14] O. Lezoray, C. Meurie, et A. Elmoataz. A graph approach to color mathematical morphology. Dans *Proc. of ISSPIT*, pages 856–861, 2005.

[15] R. Diestel. *Graph Theory*, volume 173. Springer-Verlag, 2005.

[16] Ch. Theoharatos, G. Economou, et S. Fotopoulos. Color edge detection using the minimal spanning tree. *Pattern recognition*, 38 :603–606, 2005.



(a) Image. (b) Erosion Lex. (c) Erosion Bit.Mix. (d) Erosion Graphe

(e) Dilatation Lex. (f) Dilatation Bit.Mix. (g) Dilatation Graphe (h) Erosion : Lex. / Graphe

(i) Erosion : Bit.Mix. / Graphe (j) Dilatation : Lex. / Graphe (k) Dilatation : Bit.Mix / Graphe

Figure 2 – *Comparaison entre ordres pour la morphologie mathématique*

(a) Image. (b) Image bruitée. (c) VMF (d) VMF graphe

(e) VDF (f) VDF graphe (g) VMF/VMF graphe (h) VDF/VDF graphe

Figure 3 – *Comparaison entre ordres pour des filtres médians.*

Codage de vecteurs mouvement par compétition de prédicteurs spatio-temporels dans le standard H.264

G. Laroche[1,2] J. Jung[1] B. Pesquet-Popescu[2]

[1]MAPS/MDG/SIA

France Telecom R&D
38-40 rue du Général Leclerc,
92794 Issy Les Moulineaux

[2] Signal and Image Proc. Department

ENST Paris
46 rue Barrault, 75634 Paris

{guillaume.laroche, joelb.jung}@orange-ft.com
{beatrice.pesquet}@enst.fr

Résumé

Le nouveau standard vidéo H.264/MPEG4-AVC permet une réduction significative du débit par rapport à ses prédécesseurs. Les performances obtenues sur le codage de la texture ainsi que la compensation de mouvement subpixelique ont contribué à augmenter la proportion du débit de l'information allouée au mouvement. Le nouvel objectif de l'ITU-T est de concevoir un codec vidéo réduisant le débit de 50% par rapport au standard H.264. Ce prochain codec augmentera certainement la proportion de l'information de mouvement. Par conséquent la réduction du débit de cette information devient un sujet de recherche essentiel pour le codage vidéo. Dans cet article, une méthode par compétition de prédicteurs spatio-temporels est proposée. La sélection s'effectue à l'aide d'un critère débit-distorsion prenant en compte la nouvelle information liée au mouvement. Cette méthode tire partie des redondances temporelles des champs de vecteurs, non exploitées par le médian spatial présent dans le standard. Les réductions de débit obtenues par rapport au codec H.264/MPEG4-AVC atteignent 20% pour des séquences complexes.

Mots clefs

Codage de vecteurs mouvement, H.264, prédiction spatio-temporelle, compétition, critère débit-distorsion.

1 Introduction

Le récent standard de l'ITU-T/ISO-IEC, H.264 [1], appelé MPEG4-AVC par l'ISO/IEC, obtient un gain de compression significatif par rapport à ses prédécesseurs H.263 et MPEG-4 part 2. Ce gain provient de l'amélioration des outils existants et de l'introduction de nouveaux outils dont les plus importants sont : les multiples prédicteurs Intra, les partitions variables des macroblocs, la compensation de mouvement au $\frac{1}{4}$ de pixel, le "deblocking filter" contenu dans la boucle de codage et le codage arithmétique adapté au contexte (CABAC). De plus un travail conséquent a été entrepris sur le codec de référence [2]. Basé sur un critère de débit-distorsion (RD), il offre ainsi les choix optimaux parmi la multitude des modes de codage possibles. Cette réduction efficace du débit affecté au codage de la texture entraîne une baisse du débit total de H.264 mais augmente la proportion de l'information de mouvement dans ce débit. En effet, pour les bas débits cette proportion atteint 40%.

Le Video Coding Expert Group (VCEG/ITU-T SG16 Q6) atteindra probablement son objectif de normaliser un codec obtenant un gain de compression de 50% par rapport à la référence H.264. Ce prochain standard augmentera certainement la proportion de l'information de mouvement, à l'aide d'une compensation de mouvement beaucoup plus fine et d'une forte diminution du débit des résiduels de bloc. Nous avons donc porté notre attention sur la réduction de cette information de mouvement.

Ce problème a déjà été abordé dans la littérature. Dans cet article seules les méthodes de codage sans perte des vecteurs mouvement, beaucoup plus répandues, sont décrites. Toutefois notons qu'il existe des méthodes de codage avec pertes [3]. L'exploitation de redondances temporelles par prédiction entre deux champs de vecteurs a déjà été entreprise dans [4]. Cette méthode donne de bons résultats pour des séquences contenant des mouvements complexes. Cependant la prédiction temporelle est moins efficace qu'une prédiction spatiale seule lorsqu'un ensemble de vidéos représentatives est utilisé. Dans [5] un prédicteur spatial et un prédicteur temporel sont employés, mais le choix entre ces deux prédicteurs dépend uniquement de la valeur des prédicteurs, ce qui ne constitue pas une méthode par compétition. Dans [6], une sélection entre des prédictions spatiales et spatio-temporelles est proposée, mais uniquement au niveau *slice*. Cette approche est une méthode par compétition. En effet, le meilleur prédicteur, parmi un ensemble de prédicteurs, est sélectionné à l'aide d'un critère d'efficacité de codage et l'indice correspondant à ce prédicteur est codé. Dans [7] une compétition au niveau vecteur est explicitée. Cependant les redondances temporelles ne sont pas exploitées.

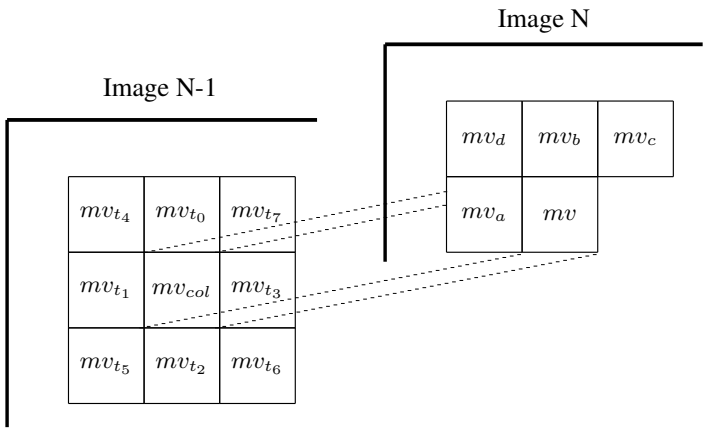


Figure 1 – *Localisation des vecteurs spatiaux et temporels utilisés pour la prédiction.*

Dans cet article nous proposons deux nouvelles techniques pour améliorer ces méthodes. Dans un premier temps, une compétition de prédicteurs spatio-temporels de vecteurs mouvement est introduite, avec la modification du critère de débit-distorsion associée. Puis l'introduction d'un vecteur conditionnel spatio-temporel pour le mode SKIP est explicité. L'ensemble de ces modifications a été implémenté dans le codec de référence du standard H.264.

Cet article est organisé de la manière suivante : un résumé sur le codage des vecteurs mouvement dans H.264 est présenté dans la section 2. Les modifications apportées à ce codage et la proposition d'un nouveau vecteur pour le mode SKIP sont décrites dans la section 3. Enfin la section 4 commente brièvement l'impact des méthodes proposées sur la complexité, et présente les gains de compression situés en moyenne à 4% pour une qualité équivalente, au meilleur profil d'H.264.

2 Etat de l'art du codage des vecteurs mouvement

Tout d'abord décrivons les procédés non-normatifs pour la sélection des vecteurs mouvement, implémentés dans le codec de référence de H.264. Le standard H.264 applique un codage prédictif des vecteurs mouvement, et le résiduel de ce vecteur ε_{mv} est donné par la formule suivante :

$$\varepsilon_{mv} = mv - p \tag{1}$$

dans laquelle mv est le vecteur mouvement, p le médian de 3 vecteurs voisins représentés dans la Figure 1 (mv_a, mv_b, mv_c). Cependant si mv_b n'est pas disponible, la valeur de mv_b est égale à la valeur du vecteur mv_d. Si un ou plusieurs des vecteurs voisins ne sont pas disponibles, p est égal, en fonction des disponibilités, à mv_a ou mv_b ou mv_c ou 0. Le meilleur compromis entre la qualité et le débit est obtenu en minimisant le critère de débit-distorsion :

$$J = D + \lambda R \tag{2}$$

où D est la distorsion calculée dans le domaine spatial ou transformé et λ est une constante positive. R est le débit représentant l'ensemble des composantes du débit [8] :

$$R = R_r + \lambda_m R_m + \lambda_o R_o + \lambda_{mv} R_{mv} \tag{3}$$

où R_r est le débit du résiduel de bloc (luminance+chrominance), R_m le débit du mode de codage, R_{mv} le débit du résiduel de vecteur mouvement et R_o le débit des autres composantes (entête, structure de bloc, bit de bourrage, quantificateur). λ_m, λ_o et λ_{mv} sont des coefficients de pondération dépendant du pas de quantification. Les différentes façons d'estimer la distorsion et le débit sont décrites dans [9]. En terme de complexité de calcul, cette sélection est intensive, mais optimale au sens du compromis débit-distorsion. Cependant elle engendre un champ de vecteurs "non-naturels" (ne correspondant pas aux mouvements réels des objets). Ce processus de sélection est entrepris pour chaque partition de bloc (16x16, ..., 4x4), pour chaque image de référence, pour le pixel entier, le $\frac{1}{2}$ et le $\frac{1}{4}$ de pixels.

Le mode SKIP est un cas particulier du codage Inter. Pour un macrobloc utilisant ce mode, le codeur ne transmet aucun résiduel de bloc, aucun vecteur mouvement, ni aucun indice d'image de référence. La seule information envoyée est le mode de codage (SKIP). Le vecteur mouvement associé à ce codage correspond au vecteur prédicteur d'un Inter 16x16, hormis le cas où mv_a ou mv_b ne sont pas disponibles. Dans ce cas le vecteur est égal à 0.

La Figure 2 représente l'évolution, en fonction du pas de quantification, des débits des différentes composantes de l'Equation 3, pour le profil *High* de H.264. A bas débit, l'information de mouvement R_{mv} représente la plus importante composante du débit total et atteint 38%.

Cette observation a motivé nos recherches basées sur la diminution de l'entropie des résiduels de vecteurs mouvement. Cette diminution utilise une sélection et un codage conjoint optimal au sens débit-distorsion, avec une exploitation adaptative des redondances spatiale et temporelle, à l'aide d'une méthode par compétition.

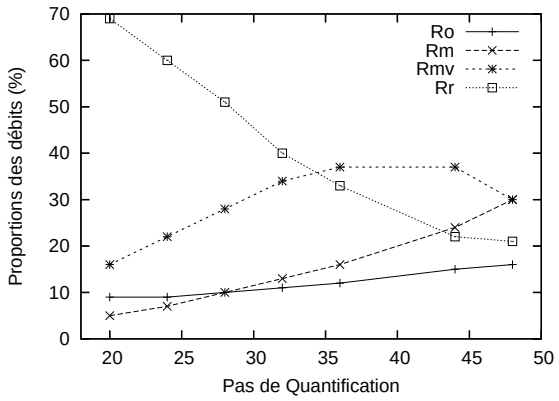


Figure 2 – *Proportions des débits en fonction du pas de quantification pour la séquence Foreman.*

3 Codage des vecteurs mouvement par compétition

Cette section présente nos travaux sur la sélection des prédicteurs des vecteurs mouvement, en comparaison avec la méthode du standard H.264 décrite dans la section 2.

3.1 Sélection du meilleur prédicteur

Lorsqu'un codage sans perte des vecteurs mouvement est utilisé, l'efficacité de codage dépend essentiellement des performances du prédicteur. Une méthode de codage par compétition implique plusieurs prédicteurs possibles pour un vecteur mouvement, et permet de sélectionner le meilleur. Nous avons donc défini un ensemble $\mathcal{P}$ de prédicteurs, incluant des prédicteurs spatiaux, temporels et spatio-temporels. Les prédicteurs spatiaux sont les vecteurs voisins mv_a, mv_b, mv_c, mv_d de la Figure 1 et le médian de la norme H.264 $mv_{H.264}$. Les prédicteurs temporels sont le *collocated* mv_{col} (le vecteur mouvement utilisé pour le codage du bloc ayant la même position dans l'image précédente) et 2 médians temporels mv_{tm5} et mv_{tm9} définis par :

$$mv_{tm5} = median\{mv_{col}, \{mv_{t_j}, 0 \leq j \leq 4\}\} \quad (4)$$

$$mv_{tm9} = median\{mv_{col}, \{mv_{t_j}, 0 \leq j \leq 8\}\} \quad (5)$$

Les prédicteurs spatio-temporels que nous utilisons sont la combinaison de vecteurs spatiaux et temporels, par exemple le vecteur mv_{spt} défini ci-dessous :

$$mv_{spt} = median\{mv_{col}, mv_{col}, mv_a, mv_b, mv_c\} \quad (6)$$

Notons que ce médian donne une plus grande importance au vecteur *collocated* qu'aux vecteurs mv_a, mv_b, mv_c. Un mode de prédiction i et un résiduel ε_{mv_i} sont maintenant associés à chaque prédicteur $p_i \in \mathcal{P}$:

$$\varepsilon_{mv_i} = mv - p_i, \forall i \in \{1, ..., n\} \quad (7)$$

où n est le nombre de prédicteurs de $\mathcal{P}$.

Le mode i doit être transmis dans le bitstream, tout comme le résiduel ε_{mv_i}. Le coût de cette nouvelle information (mode i) n'est pas négligeable (en moyenne 3.5% du débit et 12.5% du débit de l'information allouée au mouvement pour $n = 2$). L'efficacité de cette méthode de codage par compétition dépend toutefois du compromis entre ce coût additionnel et le gain obtenu par une plus grande précision de prédiction.
La sélection du vecteur mouvement de l'Equation 3 est remplacée par l'Equation 8 et le débit du résiduel de vecteur mouvement R_{mv} par $R_{mv/mm}$. $R_{mv/mm}$ contient le coût de la prédiction (ε_{mv_i}) et le coût du mode de prédiction (i).

$$R = R_r + \lambda_m R_m + \lambda_o R_o + \lambda_{mv/mm} R_{mv/mm} \quad (8)$$

$R_{mv/mm}$ est donné par :

$$R_{mv/mm} = \min\{\varsigma(\varepsilon_{mv_i}) + \varsigma(i)\}_{i \in \{1..n\}} \quad (9)$$

où $\varsigma(x)$ est le coût associé à la donnée x dans le bitstream. Notons que les modes de prédiction sont codés à l'aide du codage CABAC. Un élément clé de notre méthode est la capacité du décodeur à "deviner" dans certains cas le mode de prédiction utilisé. Le codeur simule le processus de décodage pour déterminer quel mode peut être deviné et selon le cas, encode ou non le mode. Pour deviner un mode, les informations à la disposition du décodeur sont : la valeur du résiduel ε_{mv_i}, l'ensemble des prédicteurs possibles $\mathcal{P}$, le codage des blocs voisins (spatialement et temporellement). En pratique le mode de prédiction peut être deviné dans 27% des cas lorsque $n = 2$.

3.2 Modification du vecteur mouvement pour le mode SKIP

Comme nous l'avons expliqué dans la section 2, le mode SKIP est très efficace. Le choix de ce mode signifie qu'il est plus intéressant, au sens du critère débit-distorsion, de n'envoyer aucun résiduel de bloc ou de vecteur. Ce mode est largement utilisé, notamment dans les séquences à fond fixe. Notre objectif est par conséquent d'augmenter le plus possible l'occurrence de ce mode. La seule solution pour augmenter le nombre de modes SKIP, sans modifier le critère débit-distorsion, est de modifier son vecteur mouvement par un vecteur qui permet d'obtenir une distorsion plus faible. Le vecteur mouvement de ce mode correspond à un saut, en fonction des disponibilités, entre les valeurs : $median(mv_a, mv_b, mv_c)$, mv_a, mv_b, mv_c ou 0. Nous avons modifié ce processus en définissant un ordre privilégié de prédicteurs spatiaux et temporels. Les sauts entre les différents prédicteurs dépendent des disponibilités des données permettant de calculer ces prédicteurs. Ces disponibilités dépendent de la position du bloc et du codage (Inter/Intra) des vecteurs voisins. Le décodeur est capable de reproduire le même comportement, sans recevoir de nouvelles informations.
L'ordre des sauts que nous avons défini est donné dans le Tableau 1. Cet ordre est le suivant : 1- le médian spatial

Ordre	Prédicteur	Disponibilités
1	$median(mv_a, mv_b, mv_c)$	mv_a, mv_b, mv_c
2	mv_{tm9}	$mv_{t_j}, \forall j < 8, mv_{col}$
3	mv_{tm5}	$mv_{t_j}, \forall j < 4, mv_{col}$
4	mv_{col}	mv_{col}
5	mv_a	mv_a
6	mv_b	mv_b
7	mv_c	mv_c
8	0	

Tableau 1 – *Ordre des prédicteurs pour le vecteur mouvement du mode SKIP.*

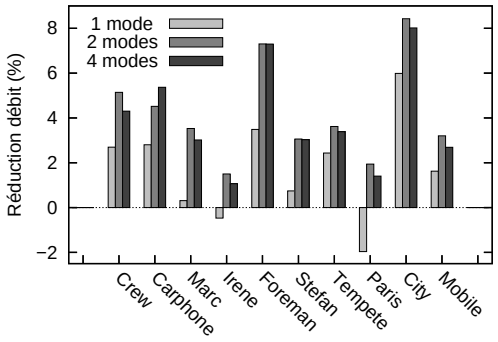


Figure 3 – *Réduction du débit pour les configurations 1, 2 et 4 modes.*

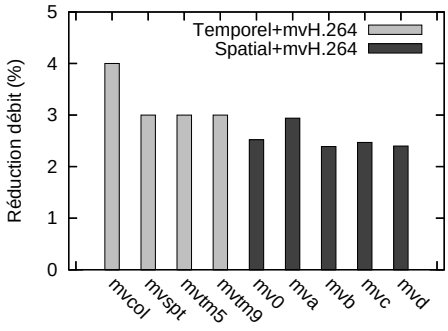


Figure 4 – *Réduction du débit pour chaque combinaison de $mv_{H.264}$ et d'un autre prédicteur.*

est sélectionné si mv_a, mv_b, mv_c sont disponibles, 2, 3- sinon le médian temporel avec 9 puis 5 composantes est retenu si tous les vecteurs permettant le calcul sont disponibles, autrement 4- le vecteur *collocated* est choisi, puis 5- mv_a, 6- mv_b, 7- mv_c, et finalement 8- la valeur '0'. Cet ordre a été défini dans le but de privilégier des compromis entre des vecteurs. Ainsi les trois premiers prédicteurs de cet ordre sont des médians. De plus, des tests effectués sur un ensemble représentatif de séquences ont montré que le médian spatial est plus efficace que chacun des deux médians temporels.
Ce codage des vecteurs mouvement et cette modification du mode SKIP nous ont permis d'obtenir des gains de compression pour une qualité équivalente à la référence. Ces gains sont analysés et décrits dans la section suivante.

4 Résultats expérimentaux et analyse de complexité

Les expériences ont été effectuées dans le codec de référence JM10.0 H.264 [2]. Nous avons utilisé le profil *High* (Fr-Ext) avec une fenêtre de recherche de 32×32, l'option RD-optimisation (RdOpt=1) et le codage entropique CABAC. Cette configuration correspond à la meilleure qualité possible avec les outils normatifs de H.264 et les décisions non-normatives les plus efficaces implémentées dans le récent JM, excepté les images B et les images de référence multiples. Nos modifications ne sont pas encore implémentées pour ces deux options. L'ensemble des séquences choisies est composé de 10 séquences en résolution QCIF, 10 en CIF, et 5 en SD de 100 images chacune, avec des contenus et mouvements variés. Nous avons utilisé les pas de quantification 30, 36, et 42. Comme les résultats obtenus pour les différentes résolutions sont de même ordre de grandeur, nous détaillerons uniquement les gains pour les 10 séquences CIF.

4.1 Impact sur la complexité

Le codec de référence de H.264 permet d'utiliser tous les profils et d'extraire de nombreuses statistiques du codage obtenu. Ce code C n'est pas optimisé ni en terme de complexité de calcul ni en terme de gestion de mémoire. Par conséquent, une mesure quantitative de la complexité ne permettrait pas une évaluation correcte de cette complexité. Nous soulignerons donc uniquement l'impact de nos modifications sur l'algorithme. Cet impact dépend du nombre de prédicteurs de l'ensemble $\mathcal{P}$. En effet, lorsque seul le prédicteur $mv_{H.264}$ est calculé dans le codec de référence, les n prédicteurs de $\mathcal{P}$ sont calculés. De plus pour chaque vecteur candidat de chaque fenêtre de recherche, l'ensemble des résiduels ε_{mv_i} de l'Equation 7 et le débit $R_{mv/mm}$ de l'Equation 9 sont calculés et évalués. L'impact de la modification pour le mode SKIP est négligeable car elle ne modifie pas l'algorithme de recherche. En termes de gestion de mémoire, les prédictions temporelles que nous avons définies requièrent le stockage des vecteurs mouvement et des modes utilisés pour le codage de l'image précédente.

4.2 Nombre de prédicteurs

Les résultats de notre méthode de codage par compétition dépendent du nombre et du type de prédicteurs utilisés. Nos expériences faites pour 1 ($mv_{H.264}$), 2 ($mv_{H.264} + mv_{col}$) ou 4 ($mv_{H.264} + mv_a + mv_{col} + mv_{tm9}$) prédicteurs avec la modification pour le mode SKIP, montrent que le meilleur compromis est obtenu pour 2 prédicteurs. Les résultats de cette expérience sont illustrés dans la Figure 3. Le gain moyen obtenu sur le débit total pour ces 3 configurations

est respectivement 1.8%, 4.2%, et 3.9%. Il est utile de noter que la réduction du débit est plus importante pour 2 prédicteurs, excepté pour une séquence (Carphone). Evidemment la réduction du débit des vecteurs mouvement (R_{mv}) est forte pour 4 modes de prédictions, mais le compromis avec la transmission du mode de codage ($R_{mv/mm}$) entraîne des résultats légèrement plus faibles. Une méthode adaptative, permettant de choisir des ensembles de prédicteurs différents selon des critères statistiques ou de caractéristiques locales, devrait augmenter le gain de compression.
Pour sélectionner le second prédicteur, nous avons associé le médian $mv_{H.264}$ à l'ensemble des prédicteurs décrits précédemment. Les résultats donnés en Figure 4 montrent que la combinaison de $mv_{H.264}$ avec un prédicteur temporel donne de meilleurs résultats que l'association de $mv_{H.264}$ avec un prédicteur spatial. En effet, deux prédicteurs spatiaux ont la plupart du temps des valeurs similaires. Ainsi les résiduels de vecteurs engendrés ont un coût équivalent.

4.3 Configuration avec 2 prédicteurs

La meilleure configuration sélectionnée est la combinaison d'un prédicteur spatial, le médian de la norme H.264, et d'un prédicteur temporel, le vecteur *collocated*. De plus le nouvel ordre de saut pour le vecteur du mode SKIP, défini dans la section 3.2, est employé. Tous les résultats qui suivent sont donnés pour cette configuration.

Réduction du débit de l'information de mouvement. La Figure 5(a) illustre le gain obtenu par la nouvelle information de mouvement ($R_{mv/mm}$) par rapport au coût de l'information de mouvement du standard H.264 R_{mv}. La réduction moyenne est d'environ 10%. Cette diminution est plus élevée pour les bas débits et les séquences contenant du mouvement. Elle s'explique par le fait que l'information de mouvement représente une large part du débit total, comme on peut le voir dans la Figure 2.

Augmentation du nombre de modes SKIP. La Figure 5(b) représente le pourcentage d'augmentation du nombre de modes SKIP, dont la moyenne se situe à 6%. On observe que cette augmentation n'est pas liée aux pas de quantification (QP), mais aux types des séquences. Les séquences à fond fixe (Marc, Irene, Paris) ne profitent pas de l'ordre donné dans la section 3.2. Ainsi pour ce type de séquences la valeur 0 devrait avoir une forte priorité dans l'ordre de choix. Il est aussi intéressant de noter que le gain moyen obtenu par la seule modification du mode SKIP (sans compétition sur les vecteurs) est situé à 1.8% comme le montre la Figure 3. Cette moyenne est certes faible, mais atteint 2.8% pour les séquences contenant plus de mouvement. Ce résultat est important pour une modification apportant une complexité de calcul aussi faible. De plus une adaptation de l'ordre des prédicteurs pour le mode SKIP, basée sur le type de séquence, devrait accroître les gains.

Réduction du débit total. Le Tableau 2 représente le pourcentage de sélection de chacun des deux prédicteurs.

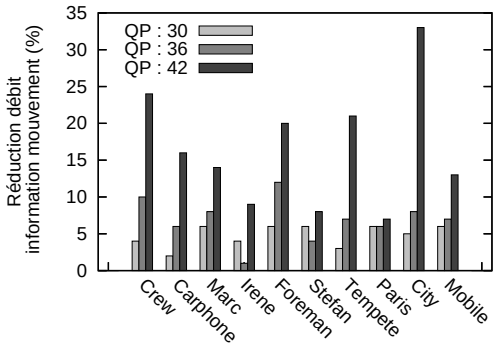
QP	Prédicteur spatial $mv_{H.264}$	Prédicteur Temporel mv_{col}	Prédicteurs égaux (même valeur)
30	62%	38%	16%
36	56%	44%	15%
42	46%	54%	17%
Moyenne	55%	45%	16%

Tableau 2 – *Pourcentage de sélection des prédicteurs.*

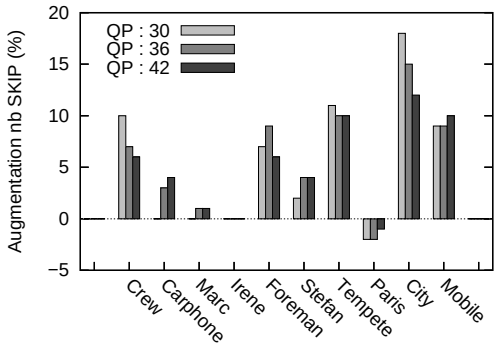
On constate que le prédicteur temporel mv_{col} est sélectionné en moyenne dans 45% des cas. Ce pourcentage exclut les cas où les deux prédicteurs sont égaux (en moyenne 16% des cas). Sachant que la sélection a été effectuée au sens du critère débit-distorsion, ce pourcentage élevé confirme l'utilité du prédicteur temporel. Une caractéristique intéressante est la corrélation entre le pourcentage de prédicteurs mv_{col} sélectionnés et les pas de quantification. Notons enfin que le pourcentage de mv_{col} varie de 24% à 62% en fonction des séquences et des pas de quantification. La Figure 5(c) montre la réduction du débit pour chaque séquence à chaque pas de quantification. On notera le gain positif pour toutes les séquences et tous les pas de quantification, même pour les séquences contenant peu de mouvement, avec une perte de PSNR moyenne de 0.04 dB, par rapport au standard H.264. La qualité visuelle reste équivalente. Le gain moyen sur le débit est d'environ 4% et atteint un maximum de 20%. Evidemment, l'augmentation est plus faible pour les séquences contenant peu de mouvement (e.g. séquences de type visiophonie) car le mode SKIP est déjà largement utilisé. Pour les séquences avec un mouvement rapide et complexe les gains sont plutôt élevés. Finalement les séquences avec un mouvement global et constant et combiné à de hautes fréquences spatiales (comme City), tirent le plus grand avantage de la prédiction temporelle, tandis que le médian spatial échoue. La réduction du débit est bien sûr liée aux pas de quantification, comme nous l'avons vu dans la Figure 2. En effet la proportion de l'information de mouvement dans le débit augmente avec le pas de quantification.

5 Conclusion

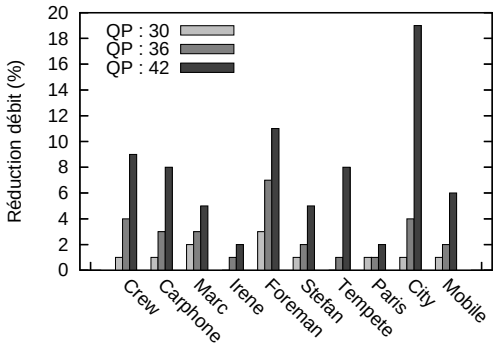
Dans cet article, une méthode de codage par compétition de prédicteurs des vecteurs mouvement est proposée. Les deux prédicteurs, spatial et temporel, sont sélectionnés via un critère débit-distorsion prenant en compte le débit des résiduels et le mode de la prédiction. De plus, une modification est proposée pour augmenter le nombre de macroblocs encodés en mode SKIP. Ces deux techniques combinées et implémentées dans le codec de référence JM10.0 H.264 fournissent un gain systématique de compression (4% en moyenne et atteint 20% avec une dégradation de PSNR négligeable) avec une très légère augmentation de complexité. Il est prévu d'implémenter ces modifications



(a) Réduction du débit de l'information de mouvement.



(b) Augmentation du nombre de modes SKIP.



(c) Réduction du débit total.

Figure 5 – *Résultats expérimentaux pour 2 prédicteurs.*

pour les images B et les images de référence multiples afin d'accroître encore ce gain.

Références

[1] ITU-T Recommendation H.264 et ISO/IEC 14496-10 AVC. Advanced video coding for generic audiovisual services. version 3 : 2005.

[2] H.264/AVC software coordination. *K. Suehring*, http ://iphome.hhi.de/suehring/tml/.

[3] L.A. Da Silva Cruz et J.W. Woods. Adaptive motion vector quantization for video coding. Dans *IEEE ICIP*, volume 2, pages 867–870, Octobre 2000.

[4] J. Yeh, M. Vetterli, et M. Khansari. Motion compensation of motion vectors. Dans *IEEE ICIP*, volume 1, pages 574–577, Octobre 1995.

[5] M.C. Chen et A.N. Willson. A spatial and temporal motion vector coding algorithm for low-bit-rate video coding. Dans *IEEE ICIP*, volume 2, pages 791–794, Octobre 1997.

[6] A.M. Tourapis, F. Wu, et S. Li. Direct mode for bipredictive slices in the H.264 standard. *IEEE Trans. on CSVT*, 15(1), Janvier 2005.

[7] S. Deuk Kim et J. Beom Ra. An efficient motion vector coding scheme based on minimum bitrate prediction. *IEEE Trans. on Image. Proc*, 8(8) :1117–1120, Août 1999.

[8] G.J. Sullivan et T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Proc. Mag.*, pages 74–90, 1998.

[9] K.P. Lim, G. Sullivan, et T. Wiegand. *Text Description of JM Reference Encoding Methods and Decoding Concealment Methods*. JVT-N046 contribution, Hong-Kong, Janvier 2005.

Estimateur de mouvement temps réel multi-DSP pour l'encodage vidéo MPEG-4 AVC/H.264 haute définition

F. Urban[1] R. Poullaouec[1] O. Deforges[2] J-F. Nezan[2]

[1] THOMSON RD France VCL

1 av.belle fontaine, 35576 Cesson-Sévigné

{urbanf, poullaouecr}@thomson.net

[2] IETR - UMR CNRS 6164/groupe Image

20 av. des Buttes de Coësmes 35043 Rennes

{odeforge, jnezan}@insa-rennes.fr

Résumé

Le dernier standard vidéo MPEG-4 AVC/H.264 proposé en mars 2003 s'appuie sur des nouvelles techniques améliorant la compression. En contrepartie, MPEG-4 AVC introduit une complexité rendant problématique les solutions temps réel dans le domaine de l'embarqué. Les performances de ces codeurs vidéo dépendent en grande partie de celles de l'estimation de mouvement. C'est aussi la fonction qui requiert le plus de ressources de calcul et de bande passante mémoire. Le cadre de la vidéo haute définition amplifie cette difficulté d'une exécution temps-réel.

De nombreuses techniques d'estimation de mouvement ont été développées afin de réduire les temps de traitement en gardant la meilleure précision possible. L'objectif de cet article est tout d'abord de faire le point sur ces travaux et de mettre en évidence les meilleurs candidats pour un estimateur MPEG-4 AVC. Les algorithmes HME et EPZS sont ensuite étudiés et implantés sur un processeur de traitement du signal. Leurs performances en terme de qualité d'estimation et de vitesse d'exécution sont finalement comparées.

Mots clefs

Estimation de mouvement, Codeur MPEG-4 AVC temps réel, DSP

1 Introduction

Grâce au développement des techniques de compression vidéo et des systèmes de communications, la diffusion de séquences vidéo est de plus en plus répandue. Néanmoins la bande passante nécessaire pour transmettre une vidéo haute définition reste importante, malgrès le développement de schémas de compression toujours plus efficaces qui permettent de réduire les débits. Le standard de compression vidéo MPEG-4 AVC / H.264 issu de la collaboration entre ITU et MPEG (JVT) permet d'atteindre des débits 50% inférieurs à ceux offerts par MPEG-2.

La compression de donnée dans une vidéo est basée sur l'élimination des redondances spatiales et temporelles. Chaque image peut en effet être reconstruite en utilisant de la prédiction intra-image (I) ou inter-image (P, B). Les images P et B sont reconstruites à l'aide d'une ou plusieurs images précédemment encodées (référence) auxquelles un champ de vecteur est associé. L'opération d'estimation de mouvement consiste à rechercher pour tous les blocs de l'image courante leur mouvement respectif par rapport à une image de référence. Les performances d'un encodeur vidéo dépendent donc beaucoup de la précision de l'estimation de mouvement.

L'estimation de mouvement est une opération qui nécessite une puissance de calcul importante, notamment dans la norme H.264 où les images peuvent être découpées en blocs de tailles variables, les vecteurs de mouvements sont exprimés au quart de pixel près et plusieurs images de référence sont autorisées [1]. De 60 à 80% des ressources matérielles d'un encodeur vidéo sont occupées par l'estimation de mouvement.

L'objectif général de ces travaux est le prototypage d'un estimateur de mouvement temps réel pour un encodage H.264 format HD, et implanté sur une architecture embarquée multi-composants. Cet article présente les résultats de la première phase, à savoir le choix des algorithmes et leur implantation sur DSP (Digital Signal Processor)

La partie 2 dresse un état de l'art rapide des techniques d'estimation de mouvement existantes, la section 3 décrit les implantations embarquées réalisées. La section 4 donne des résultats de comparaison de deux algorithmes d'estimation de mouvement en terme de qualité d'estimation et de temps d'exécution.

2 Méthodes d'estimation de mouvement - état de l'art

L'opération d'estimation de mouvement permet de retrouver les mouvements relatifs entre deux images afin d'éliminer la redondance temporelle. Il existe différentes techniques d'estimation de mouvement et plusieurs façons d'exprimer le résultat. Les algorithmes pixels-récursifs basés gradient [2] produisent un champ de vecteurs dense, c'est à dire qu'une description du mouvement sera donnée

pour chaque pixel de l'image. Une telle description est utile pour des traitements vidéo comme le suivi d'objet, le désentrelacement ou encore la conversion de standard, mais est inadaptée à la compression vidéo où l'image est classiquement divisée en blocs pour être compressée. Les techniques basées sur des transformations fréquentielles comme la corrélation de phase [3, 4] s'appuient sur la théorie mathématique de la transformée de Fourier. Le champ de vecteur résultant a alors des propriétés intéressantes : il correspond aux mouvements réels et est insensible aux variations de luminance. Cependant, la quantité de calculs engendrée est très importante. De plus l'estimation de mouvement a pour but de trouver le bloc le plus ressemblant dans une image de référence, et donc le mouvement réel n'est pas forcément le mieux adapté. Pour un estimateur de mouvement dédié à l'encodage vidéo, un algorithme de mise en correspondance de blocs (BMA : Block Matching Algorithm) est préféré.

2.1 Mise en correspondance de blocs

Les BMA consistent à rechercher pour chaque bloc de l'image courante le bloc qui lui ressemble le plus dans une image de référence. L'image courante est découpée en blocs de taille $M \times N$, puis pour chacun d'entre eux une mise en correspondance est effectuée. Une mesure de distance est alors calculée entre le bloc courant et un certain nombre de candidats. Cette mesure peut par exemple être la somme quadratique des différences (SSE : Sum of Square Errors), la somme des valeurs absolues des différences (SAD : Sum of Absolute Differences) ou un calcul prenant en compte plus précisément le coût de codage du résidu grâce à une transformée comme la SATD (Sum of Absolute Transformed Differences). Pour des raisons de facilité d'implantation et de coût de calcul, la SAD est souvent retenue. Le coût de codage des vecteurs peut aussi être pris en compte dans cette mesure par le biais d'un coefficient de lagrange pour résoudre le problème d'optimisation débit/distorsion [5].

L'algorithme de mise en correspondance le plus simple est la recherche exhaustive. Chaque déplacement possible d'amplitude maximale p est considéré comme candidat. C'est l'algorithme qui demande le plus de puissance de calcul. En effet, $(2p + 1)^2$candidats sont considérés par bloc, induisant pour chacun une SAD $M \times N$ à calculer. Par exemple pour une image 720p (1280×720), des blocs de taille 8×8 et $p = 16$, cela abouti à 15,7 millions de SAD par image. La puissance de calcul nécessaire étant démesurée, plusieurs algorithmes rapides ont été proposés en utilisant principalement trois techniques d'optimisation.

La première consiste à calculer la SAD complète le moins souvent possible [6, 7]. En effet il est possible d'éliminer rapidement certains candidats avant même d'avoir effectué tous les calculs. Ces algorithmes réduisent énormément le nombre de calculs, néanmoins ils impliquent des opérations de conditionnement, coûteuses en temps de calcul et qui rend difficile l'optimisation de l'implantation. De plus, la contrainte temps réel nous conduit à considérer le pire cas d'exécution qui est alors moins bon que la technique de base.

La deuxième technique consiste à réduire le nombre de candidats et à orienter la recherche le plus tôt possible vers les candidats les plus probables. L'hypothèse principale est que la SAD croît lorsque l'on s'éloigne de l'optimum. Cet hypothèse n'est pas toujours vérifiée et conduit certains algorithmes à tomber dans des minima locaux. Chen et Al [8] proposent de rechercher dans une direction à la fois, alors que l'algorithme logarithmique proposé par Jain et Jain [9] et l'algorithme à trois pas de Koga et Linuma [10] font d'abord une estimation grossière puis raffinent le résultat. Dans [11] le mouvement est estimé grâce à une méthode récursive. A chaque étape, les SAD sont évaluées pour quelques candidats suivant un motif en diamant autour de la position courante. Le mouvement est raffiné successivement en suivant la direction de descente.

La troisième technique prend en considération le contenu des séquences vidéos à traiter. En effet il existe une certaine continuité du mouvement (spatialement et temporellement), c'est à dire que le mouvement d'un bloc a une grande chance d'être proche de celui d'un bloc voisin, ou du bloc colocalisé dans l'image précédente. Il est alors possible d'avoir un ensemble de prédicteurs du mouvement recherché grâce aux résultats déjà obtenus. La pertinence des différents prédicteurs est évaluée (en calculant la SAD avec le bloc courant) puis une recherche locale autour du (ou des) meilleur(s) prédicteur(s) est effectuée pour raffiner le résultat. De nombreux algorithmes utilisant cette technique ont été développés [12, 13, 14, 15, 16]. Ils diffèrent par le choix des prédicteurs et la technique de recherche locale.

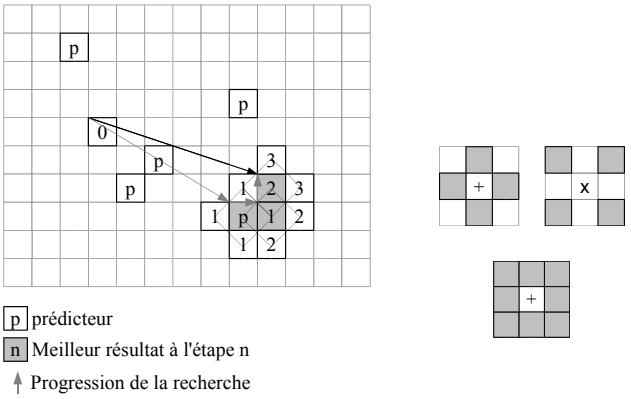
Les algorithmes EPZS [12] et hiérarchique [17] sont particulièrement intéressants pour une implantation DSP. Ceux sont eux qui ont finalement été retenus, et qui sont détaillés plus précisément ci-dessous.

2.2 EPZS

L'algorithme EPZS (Enhanced Predictive Zonal Search) est une amélioration de l'algorithme PMVFAST [13] grâce à des nouveaux prédicteurs. La phase de prédiction est alors rendue plus précise et la recherche locale (figure 1-a)) est par conséquent réduite. Le raffinement grossier suivant un grand motif en diamant réalisé dans PMVFAST est devenu inutile. Le meilleur prédicteur est directement raffiné finement suivant un motif en diamant à 4 ou 8 connexités (figure 1-b)). L'amélioration de l'étape de prédiction réduit sensiblement le temps d'exécution.

Le seuil adaptatif, déjà utilisé dans PMVFAST, permet d'accélérer davantage le traitement en éliminant les calculs inutiles. La recherche est stoppée prématurément si le résultat est "assez bon", c'est à dire inférieur au seuil adaptatif. Le peu de calcul à réaliser font de cet estimateur un bon candidat pour une implantation logicielle temps réel.

Cet algorithme est utilisé actuellement dans les logiciels d'encodage vidéo tel que XviD et dans l'encodeur de



(a) (b)

FIG. 1 – *EPZS*

référence H.264 JM pour sa vitesse d'exécution et la qualité de l'estimation.

2.3 HME

La méthode d'Estimation du Mouvement Hiérarchique (HME) [17] est basée sur un raffinement successif des vecteurs déplacement d'abord estimés grossièrement en sous-échantillonnant l'image.
L'algorithme débute par la construction du couple de pyramides d'images. Le niveau 0 correspond à l'image pleine résolution, l'image de niveau $n+1$ est obtenu en sous-échantillonnant l'image de niveau n d'un facteur 2 après l'application d'un filtre passe bas gaussien (figure 2).

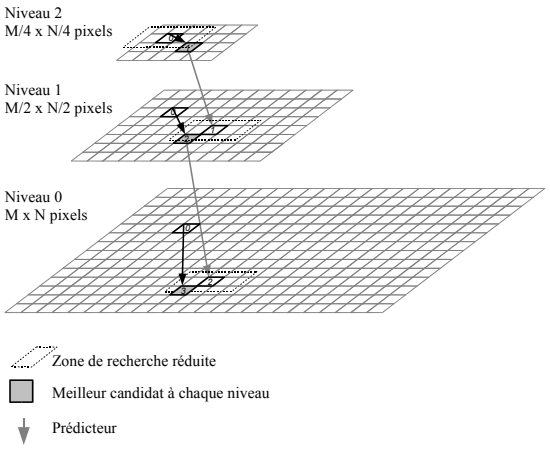


FIG. 2 – *Décomposition pyramidale d'une image*

L'estimation de mouvement est d'abord réalisée sur l'image basse résolution (niveau le plus élevé) puis le champs de vecteur est raffiné successivement. La taille des blocs à travers la pyramide reste constante de sorte que les petits détails n'influent pas sur les mouvements des niveaux supérieurs. Les détails et les mouvements des petits objets sont détectés à mesure que la résolution est augmentée.
Pour chaque niveau, l'estimation de mouvement est réalisée. On retrouve comme dans le cas de EPZS une phase de prédiction et une phase de raffinement. Les différents prédicteurs sont : le déplacement nul, les prédicteurs spatiaux, temporels et hiérarchiques. La phase de raffinement est une recherche exhaustive très réduite centrée autour du meilleur prédicteur. Cet algorithme est robuste grâce aux prédicteurs hiérarchiques qui permettent de détecter les mouvements de grande amplitude, bien que la recherche locale soit très réduite. Cet estimateur offre des performances comparables à un estimateur exhaustif en réduisant énormément le nombre de calculs.
Grâce au mécanisme prédictif et à une zone de recherche locale réduite, le champ de vecteur résultant des estimateurs HME et EPZS sont naturellement homogènes et donc de faible entropie. Ceci est un avantage non négligeable pour un encodeur vidéo.

2.4 Tailles de blocs variables

La norme H.264 réduit le débit des vidéo en ajoutant des modes de codage par rapport aux anciens standards. La compensation de mouvement peut être faite sur des blocs de taille variable. Les macroblocs 16×16 peuvent être divisé en 16×8, 8×16 ou 8×8 et les sous-partitions 8×8 peut être à nouveau divisées en 8×4, 4×8 ou 4×4. Le choix de petits blocs améliore la précision de la compensation de mouvement mais nécessite la transmission d'un plus grand nombre de vecteurs. Un algorithme de décision efficace doit donc être mis en place. Il peut faire partie intégrante de l'estimateur ou bien être séparément exécuté. Dans ce dernier cas l'estimateur de mouvement fourni les vecteurs pour toutes les tailles de blocs.
L'estimation de mouvement à taille de blocs variables peut être réalisée avec différentes approches. La première consiste à recopier le schéma d'estimation pour chaque taille de bloc. Cela permet de aisément de paralléliser les traitements mais ne permet pas d'exploiter la redondance des calculs.
La deuxième approche calcule un champ de vecteur pour une taille de bloc, ce qui permet d'initialiser une recherche beaucoup plus rapide pour les autres tailles [18].
Dans la suite du document, l'accent est porté sur la technique d'estimation pour une taille de bloc fixe. Les mouvement pour les autres tailles sont considérés déduits (cas de la deuxième approche). Les perfomances de l'estimateur global dépendent donc directement de celles de la première passe. De plus, comme l'intérêt est porté sur la qualité intrinsèque des champs de vecteurs, on cherche à s'affranchir de l'influence d'un algorithme de décision.

2.5 Précision quart de pixel

Dans H.264, la précision des vecteurs de mouvement atteint le quart de pixel. Un gain de compression significatif est apporté au prix d'une complexité plus élevée. La définition de l'image est augmentée en interpolant successivement au demi-pixel avec un filtre à 6 coefficients, puis

au quart de pixel avec un filtre linéaire.

La précision quart de pixel peut être obtenue de différentes manières ; la première est de rechercher directement dans l'image interpolée. Le nombre de candidats à prendre en compte se trouve donc augmenté, ou autrement dit, les dimensions horizontales et verticales de l'image sont quadruplées. La deuxième manière consiste à réaliser l'estimation de mouvement en plusieurs étapes [18] : dans un premier temps la recherche est effectuée à la précision pixel entier, puis le vecteur est raffiné au quart de pixel. Cette deuxième solution permet une interpolation de l'image "à la volée", c'est à dire que les positions demi et quart de pixels sont calculées uniquement lorsqu'elles sont utiles. Cela réduit la bande passante mémoire nécessaire mais augmente légèrement les calculs à effectuer. Cette dernière solution semble être un bon compromis pour une implantation DSP. De plus il est alors très aisé de rendre le raffinement subpixel facultatif et donc d'avoir une implantation évolutive.

Les vecteurs sont donc estimés avec un des algorithmes décrits précédemment au pixel entier près (avec HME ou EPZS), puis ils sont raffinés successivement au demi puis au quart de pixel.

3 Implantation temps réel sur DSP

L'implantation d'un estimateur de mouvement pour l'encodage MPEG-4 AVC/H.264 [19] haute définition représente d'énormes contraintes temps réel. Avec des spécificités telles que des tailles de blocs variables ou la précision quart de pixel des vecteurs de mouvement, l'opération d'estimation de mouvement à elle seule pose des problèmes de bande passante mémoire et de puissance de calcul. Dans cette section une implantation sur DSP d'un estimateur de mouvement pour des blocs de taille 8×8 est étudiée. Les algorithmes HME et EPZS sont comparés entre eux en termes de qualité et de temps d'exécution.

Nous ne détaillerons pas dans cet article la manière dont les optimisations d'implantation ont été effectuées. Nous pouvons toutefois préciser que les temps de traitement ont pu être divisés par 5 entre les versions originales et optimisées sur DSP.

3.1 Implantation de EPZS

L'algorithme EPZS implanté possède les caractéristiques suivantes. Les prédicteurs utilisés sont le vecteur nul, 1 prédicteur temporel et 4 prédicteurs spatiaux. L'arrêt anticipé permettant de stopper prématurément l'algorithme n'a pas été implanté de façon à obtenir un temps d'exécution plus constant, ce qui est indenspensable pour une implantation temps réel. De plus la précision des vecteurs s'en trouve améliorée.

La recherche locale est effectuée avec un motif carré (8 connexités). La qualité des vecteurs est alors meilleure au prix d'un temps d'exécution légèrement plus important par rapport à un motif en diamant à 4 connexités.

3.2 Implantation de HME

En plus des opérations d'estimation de mouvement proprement dites, l'implantation de l'algorithme hiérarchique prend en compte la construction de la pyramide d'images sous-échantillonnées. Chaque niveau est obtenu en appliquant un filtre gaussien et en sous-échantillonnant d'un facteur 2 l'image de niveau inférieur.

Le mécanisme de hiérarchie permet d'ajouter des prédicteurs (hiérarchiques) fiables par rapport à EPZS mais avec un coûp de calcul plus important.

3.3 Raffinement quart de pixel

Le point le plus contraignant du point de vue de l'implantation est sans doute le raffinement quart de pixel des vecteurs de mouvement. En effet les filtres d'interpolation sont très coûteux en terme de temps de calcul. Néanmoins le gain de compression apporté est significatif. Son implantation est donc nécessaire.

Le raffinement subpixel est implanté de la même façon, que l'algorithme d'estimation de mouvement choisi soit EPZS ou HME. Ce dernier fournit la meilleure position pixel qui est ensuite raffinée au demi-pixel en interpolant la zone utile de l'image grâce au filtre à 6 coefficients utilisé dans la norme H.264. Comme l'opération d'estimation de mouvement n'est pas normée (contrairement à la compensation) un filtre plus simple permettrait d'accélérer les calculs, cependant la qualité serait inférieure. Lorsque les valeurs demi-pixel sont disponiles, la meilleure des 8 positions demi-pixel adjacentes à la position pixel est retenue en évaluant les SAD. Ensuite le vecteur est raffiné de même au quart de pixel en appliquant un filtre d'interpolation linéaire à l'image demi-pixel.

4 Résultats

Les deux estimateurs de mouvement EPZS et HME ont été implémentés sur un DSP et intégrés dans un encodeur vidéo H.264. L'estimation de mouvement est limitée à des tailles de blocs 8×8 pour pouvoir comparer la qualité des champs de vecteurs fournis.

4.1 Temps d'exécution

Les algorithmes EPZS et HME ont été portés et optimisés sur un DSP Texas Instrument TMSC6416 à 1Ghz. Le tableau 1 donne les différents temps d'exécution obtenus pour des images progressives SD (720x576) et HD (1280x720). Pour chaque algorithme, deux versions ont été envisagées : avec ou sans raffinement quart de pixel.

Dans la version quart de pixel, EPZS est deux fois plus rapide que HME. Ceci est principalement dû à la création de la pyramide d'images et à l'estimation de mouvement des niveaux hiérarchiques, inexistants dans EPZS. Le plus grand nombre de prédicteur pour HME et une recherche locale plus intensive contribuent également à augmenter légèrement le temps d'exécution.

Une implantation de EPZS au quart de pixel est donc temps réel (30 images par secondes) sur un DSP pour de la

Algorithme	SD 720x576	HD 1280x720
HME $\frac{1}{4}$-pel	30 ms	60 ms
EPZS $\frac{1}{4}$-pel	16 ms	32 ms
HME pel	21 ms	41 ms
EPZS pel	7 ms	13 ms

TAB. 1 – *Temps d'exécution des différents algorithmes*

haute définition. Pour implémenter HME en temps réel, on peut envisager un pipeline de 2 DSP (un pour les niveaux hiérarchiques, et un pour la pleine résolution) capable de traiter la vidéo HD à 25 images par secondes.

4.2 Qualité d'estimation

Pour évaluer la qualité des champs de vecteurs fournis par les estimateurs de mouvement, ceux-ci ont été implantés dans l'encodeur vidéo H.264 développé à THOMSON Corporate Research. Seuls les blocs de taille 8×8 et le mode de codage inter sont considérés. Ceci permet de comparer uniquement la qualité des champs de vecteurs.
Les figures 3, 4 et 5 donnent les courbes débit/distorsion correspondant à l'encodage des 200 premières images de différents types de séquences. Pour chaque séquence les courbes de qualité (donnée en PSNR moyen) correspondant à chaque estimateur sont tracées en fonction du débit moyen.

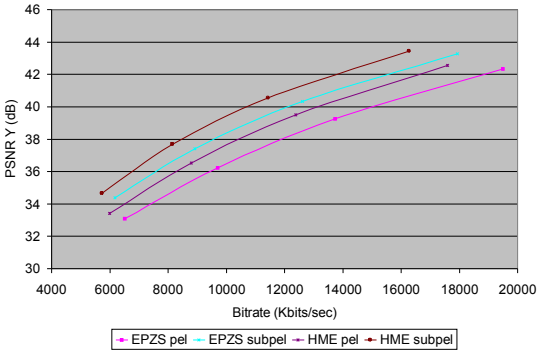


FIG. 3 – *Séquence HD*

La première séquence de test (figure 3) est une succession rapide de différente scènes en haute définition (1280x720). L'algorithme hiérarchique atteint une qualité largement supérieure du fait de la détections des mouvements de grande amplitude offert par les prédicteurs hiérarchiques. Les prédicteurs temporels de EPZS, naturellement inadaptés à chaque début de scène sont une autre explication de cette qualité inférieure. En effet lors d'un changement de scène les vecteurs ne sont pas fiables. Ils ne le sont pas non plus en tant que prédicteurs temporels pour l'image suivante. On voit aussi clairement que le raffinement $\frac{1}{4}$-pixel des vecteurs augmente les performances de l'encodeur.

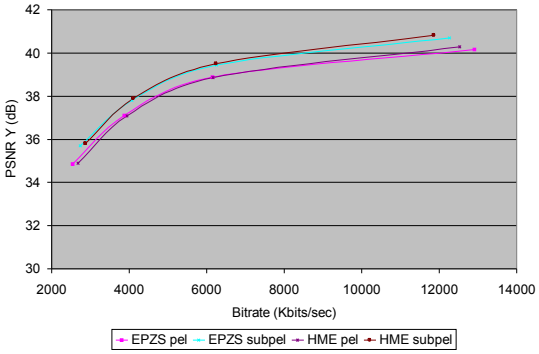


FIG. 4 – *Séquence Hockey HD*

La deuxième séquence de test (figure 4) représente un entrainement de hockey sur glace en haute définition (1280x720). Les mouvements sont assez homogènes et il y a peu de changements de scène. Les performances des estimateurs HME et EPZS sont donc comparables. Le raffinement $\frac{1}{4}$-pixel des vecteurs augmente toujours les performances de l'encodeur.

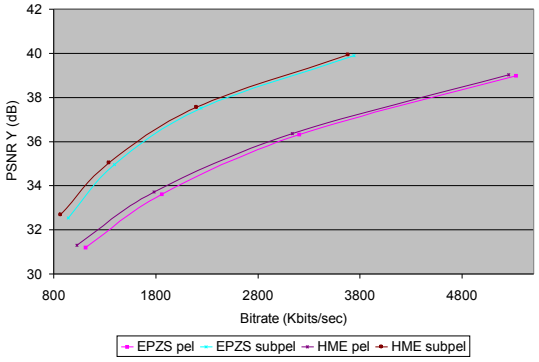


FIG. 5 – *Séquence Raid2Maroc SD*

La dernière séquence de test (figure 4) représente des scènes de sport en définition standard (720x576). Il y a quelques changements de scène mais la réduction de la définition par rapport à la première séquence permet de réduire l'écart des performances entre les deux estimateurs. Ici encore, le raffinement $\frac{1}{4}$-pixel des vecteurs semble indispensable pour offrir de bonnes performances.
Les deux estimateurs sont de qualité équivalentes pour de nombreuses séquences lorsque le mouvement reste faible ou homogène. EPZS fournit un champ de vecteur de qualité inférieure à HME lorsque la séquence contient beaucoup de mouvements. Son temps d'exécution largement inférieur en fait cependant un bon candidat pour une implantation temps réel peut coûteuse.

5 Conclusion

Un état de l'art des méthodes d'estimation de mouvement a été dressé. Les techniques HME et EPZS, intéressantes

en terme de qualité et de vitesse d'exécution ont été plus précisément décrites et implantées sur DSP Texas Instruments C64x à 1Ghz. Leurs performances dans un encodeur MPEG-4 AVC/H.264 ont été évaluées.
L'estimateur EPZS est temps réel (30 images/s) pour des images hautes définition 720 p (1280x720 progressif) au quart de pixel. HME est deux fois plus lent.
Les performances des deux estimateurs dans un encodeur vidéo sont équivalentes sur des séquences comportant peu de mouvement ou des mouvements homogènes (spatialement et temporellement). Pour des séquences plus complexes, HME est plus robuste. Le gain de codage apporté par le raffinement des vecteurs au quart de pixel semble justifier son implantation, malgré un coût de calcul beaucoup plus élevé.
Dans le cas d'un estimateur de mouvement à multiples tailles de bloc, le surcoût apporté par HME est réduit. En effet, celui-ci est dû principalement à la construction de la pyramide et à l'estimation de mouvement pour les niveaux hiérarchiques. Or leur dupliquation n'est pas nécessaire pour un estimateur à taille de blocs variable. De plus les résultats des niveaux hiérarchiques intermédiaires permettent d'initialiser une recherche rapide pour chaque taille de bloc.
On peut donc envisager le prototypage d'un estimateur de mouvement temps réel pour l'encodage MPEG-4 AVC/H.264 avec un DSP pour chaque taille de bloc dans le cas de EPZS et un DSP supplémentaire pour les niveaux hiérarchiques dans le cas de HME. Le nombre de DSP nécessaire peut être réduit en ajoutant de la décision dans l'estimateur de mouvement afin de choisir la taille de bloc et de calculer le vecteur associé.

Références

[1] Iain E.G.Richardson. *H.264 and MPEG-4 Video Compression : Video Coding for Next-generation Multimedia*. John Wiley and Sons, 2003.

[2] A. N. Netravali et J. D. Robbins. Motion-compensated television coding. I. *AT T Technical Journal*, 58 :631–670, mar 1979.

[3] G.A. Thomas. Television motion measurement for DATV and other applications. Rapport technique, BBC RD, 11 1987.

[4] R. Storey. HDTV Motion Adaptive Bandwidth Reduction using DATV. Rapport technique, BBC RD, 1986.

[5] G. Sullivan et T. Wiegand. Rate-Distortion Optimization for Video Compression. *IEEE Signal Processing Magazine*, pages 74–90, Nov 1998.

[6] Salari E. Li W.. Successive elimination algorithm for motion estimation. *IEEE Transactions on Image Processing*, 4 :107–110, 1995.

[7] Y-P Hung Y-S Chen et C-S Fuh. Fast Block Matching Algorithm Based on the Winner-Update Strategy. Dans *IEEE Transactions on Image Processing*, volume 10, August 2001.

[8] T.D. Chiueh M.J. Chen, L.G. Chen. One-dimensional full search motion estimation algorithm for video coding. *IEE Transactions on Circuits and Systems for Video Technology*, 4 :504–509, 1994.

[9] J. R. Jain et A. K. Jain. Displacement measurement and its application in interframe coding. *IEEE Transactions on Communications*, COM-29(12) :1799–1808, 1981.

[10] A.Hirano Y.Iijima T.Koga, K.Linuma et T.Ishiguro. Motion compensated interframe coding for video conferencing. *Proceedings of National Telecommunication Conference*, NTC81 :G5.3.1–G5.3.5, 1981.

[11] M. Ranganath J. Y. Tham, S ; Ranganath et A. A. Kassim. A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation. *IEEE Transactions on circuits and systemes for video technology*, 8, NO. 4, AUGUST 1998.

[12] Alexis Michael Tourapis. Enhanced Predictive Zonal Search for Single and Multiple Frame Motion Estimation. *proceedings of Visual Communications and Image Processing*, pages 1069–79, 2002.

[13] Alexis Michael Tourapis, Oscar C. Au, et Ming Lei Liou. Predictive Motion Vector Field Adaptive Search Technique (PMVFAST). Dans *Proceedings of Visual Communications and Image Processing 2001 (VCIP'01)*, 2001.

[14] P. Zhou Z. Chen et Y. He. Fast motion estimation for JVT. JVT-G016.doc, March 2003.

[15] Lappui Chau Ce Zhu, Xiao Lin et Lai-Man Po. Enhanced Hexagonal Search for Fast Block Motion Estimation. *IEEE Transactions on circuit and systems for video technology*, 14 :1210–1214, OCTOBER 2004.

[16] S. Masud F. Nasim S. Idris K. Virk, N. Khan. Low Complexity Recursive Search Based Motion Estimation Algorithm for Video Coding Applications. Dans *Proceedings of 13th European Signal Processing Conference*, Antalya, Turkey, 2005.

[17] B. Chupeau, P. Robert, M. Pecot, P. Guillotel. Multiscale motion estimation. *Workshop on Advanced Matching in Vision and Artificial Intelligence*, Munich, 5th, 6th June 1990.

[18] Jechang Jeong Woong IL Choi, Byeungwoo Jeon. Fast motion estimation with modified diamond search for variable motion block sizes. Dans *International Conference on Image Processing*, volume 2, pages 371–4, Sept. 2003.

[19] Joint Video Team of ITU-T et ISO/IEC 14496-10. Draft of version 4 of H.264/AVC. Rapport technique, Nov 2004.

Représentation des scènes vidéo par des maillages triangulés

Amal MAHBOUBI

IUT Cherbourg-Manche
LUSAC EA 2607 groupe VAI
Dept. SRC
120, rue de l'exode 50000 Saint-Lô
amal.mahboubi@unicaen.fr

Résumé

Cette étude s'inscrit dans le contexte de la conception de méthodes d'analyse de la vidéo numérique en vue de la représentation de son contenu. Nous pouvons dénombrer plusieurs objets dans une scène vidéo qui se déplacent généralement indépendamment les uns des autres. C'est pourquoi nous avons besoin d'un modèle qui pourra supporter les mouvements non rigides et qui prendra en compte les discontinuités du mouvement aux frontières des objets. Le maillage objet possède des arguments intéressant vis-à-vis de ces deux contraintes. Cet article présente une approche originale de modélisation par maillage triangulaire du contenu des scènes vidéos.. Nous avons développé dans cette étude un maillage triangulé articulé par objet d'intérêt. Cette triangulation s'appuie sur la segmentation spatio-temporelle de la scène afin de résoudre les problèmes liés à l'évolution topologiques des objets vidéos au court du temps. En effet cette représentation conjointe région/maillage permet l'identification des contours et une bonne représentation du mouvement des nœuds.

Mots clefs

Maillage triangulé, suivi temporel du maillage, évolution topologique du maillage.

1 Introduction

Dans cette étude nous nous intéressons à la pertinence d'une représentation conjointe 'région-polygonale'/'maillage-triangulaire' du contenu vidéo. Ainsi la segmentation spatio-temporelle est utilisée pour gérer les problèmes liés aux occultations, en se basant sur l'homogénéité de la luminance et du mouvement alors que le maillage triangulaire est utilisé comme outil de modélisation de cette segmentation saptio-temporelle dans un objectif de codage de type MPEG4. Notons que dans cet article nous nous limitons aux aspects relatifs à la triangulation et supposons que la segmentation spatio-temporelle est déjà disponible. En effet la littérature est riche en méthodes de segmentation spatio-temporelle, il suffit d'en choisir une qui permet l'extraction de ou des objets d'intérêt et le maintient d'un découpage cohérent de chaque objet en zones à mouvement homogène. L'analyse d'image utilise les maillages aussi bien en tant que modèles de déformations, tels les animations faciales, que comme modèles de représentation. On peut classer les maillages en deux catégories : les *maillages réguliers* et les *maillages irréguliers*. Un maillage régulier présente une structure topologique uniforme issue d'une division régulière, à l'inverse d'un maillage irrégulier sur le même ensemble de points. Dans chacune de ces catégories nous pouvons parler de maillage adaptatif ou non adaptatif. Un maillage est dit adaptatif s'il prend en considération le contenu de l'image : la densité des nœuds est modifiée localement suivant les différentes zones de l'image. Alors que le maillage non adaptatif effectue un partitionnement homogène sur le support indépendamment de la distribution de l'information. Nous avons retenu pour notre étude un maillage irrégulier adaptatif car il épouse la forme de l'objet. Plus précisément nous considérerons le maillage triangulaire irrégulier adaptatif par triangulation de Delaunay [1] articulé par objet. Le standard MPEG4 [2] considère la vidéo comme un ensemble d'objets audiovisuels indépendants, ce qui permet leur manipulation individuelle et leur composition. Ce standard introduit la représentation par maillage triangulé de l'objet d'intérêt VOP[1]. Ce concept de « l'objet-maillage » (mesh object) dépeint une entité normalisée, caractérisée par la géométrie (position spatiale des nœuds du maillage) et le mouvement (vecteurs de déplacement nodaux). L'image contiendra un maillage propre à chaque VOP. Ainsi, le maillage se concentre uniquement sur la partie de l'image contenant l'objet. Ceci présente l'avantage d'alléger la géométrie du maillage et procure un meilleur comportement quant à la régularité du mouvement : le maillage ne recouvrant qu'un seul objet, les discontinuités de mouvement seront beaucoup moins nombreuses que dans le cas d'un maillage global.

Cet article est organisé en quatre sections. Dans la section 2, nous passons en revue quelques concepts de

[1] Nous désignerons l'objet d'intérêt par VOP pour Video Object Plane, terme emprunté à MPEG4 et qui désigne la silhouette de l'objet dans la scène animée à l'instant considéré.

triangulation, nous y présentons la construction de notre maillage. Dans la section 3 nous décrivons notre suivi temporel conjoint. En fin, les principaux résultats de notre étude sont présentés dans la section 4.

2 Construction du maillage

La stratégie que nous adoptons dans cette étude consiste à extraire l'objet d'abord et, ensuite, à construire le maillage en s'appuyant sur la géométrie de l'objet affin de profiter de la qualité de la segmentation objet/fond et de pouvoir traiter les problèmes de régularité de maillage par la suite.
Avant de décrire le processus de triangulation, il est nécessaire d'introduire quelques notions relatives à la topologie et la géométrie d'un maillage triangulé articulé. Dans cette étude nous ne considérons que la triangulation 2D.
Trianguler un ensemble de points A consiste à construire une triangulation T dont les sommets sont les points de A et dont le domaine Ω est l'enveloppe convexe de A [3] tel que :

- **H1** L'ensemble des sommets des éléments de T est exactement A.
- **H2** $\Omega = \bigcup_{K \in T} K$, où K désigne un triangle.
- **H3** Tout élément K de T est d'intérieur non vide.
- **H4** L'intersection des intérieurs de 2 éléments est vide.

On dit qu'un ensemble de triangles constitue une triangulation conforme de la surface qu'il recouvre si :

- **H5** l'intersection de deux triangles est soit : l'ensemble vide (pas de triangle inclus dans l'autre ou de chevauchement), un sommet, une arête (sur toute sa longueur).

Enfin une triangulation est valide si les conditions H1, H2, H3, H4 et H5 sont vérifiées.
Dans le plan, si le cardinal de l'ensemble des points A est supérieur à trois alors il existe plusieurs possibilités pour trianguler A. Néanmoins toutes ces triangulations ne sont pas intéressantes, c'est pourquoi il est souhaitable d'en avoir une qui soit optimale vis-à-vis d'un certain critère. Ces critères peuvent être purement géométriques ou liés aux données à traiter. Ainsi dans cette étude nous exploitons l'approximation polygonale du contour des objets de la scène vidéo issue de la segmentation polygonale pour construire l'enveloppe convexe du domaine de points à trianguler et il ne nous reste plus qu'à choisir une triangulation 2D optimale géométriquement. Dans cette optique, notre choix s'est posé sur la triangulation de Delaunay qui présente d'attrayants critères géométriques. [4] ont établi que la triangulation de Delaunay est la plus appropriée pour les approximations de surface, car elle produit des triangles aussi équiangulés que possible. La construction du maillage articulé consiste en une construction géométrique du maillage, suivie du calcul du modèle dynamique du flot optique nodal. Nous allons introduire ci-dessous le modèle à la fois topologique, géométrique et dynamique d'un maillage triangulé articulé.

2.1 Modèle d'un maillage articulé

Soit $S^t=\{R_i^t\}$ la segmentation spatio-temporelle d'un objet à l'instant t. Elle est considérée comme un ensemble connexe de régions polygonales dans une zone limitée du plan-image à l'instant donné t. Le maillage articulé associé à cette segmentation sera défini comme suit.

2.1.1 Modèle géométrique

Soit $S=\{R_i\}$ l'ensemble des domaines polygonaux dans le plan-image. On note *Frontière(R)* l'ensemble des sommets et des arêtes d'un domaine polygonal R donné. On désigne par *Intérieur(R)* l'ensemble des points strictement à l'intérieur de R c'est-à-dire : *{R}-Frontière(R)* où le signe '-' désigne la soustraction ensembliste.
Nous supposons que dans le cas général les domaines R ne sont pas simplement connexes, à savoir que le graphe correspondant à *Frontière(R)* n'est pas connexe. Une de ses composantes connexes correspond à la frontière extérieure du domaine et d'autres à la frontière intérieure encerclant des trous dans un polygone.
Nous supposons systématiquement que pour tout couple (R_i,R_j) *Intérieur*$(R_i) \cap$ *Intérieur*$(R_j) = \varnothing$
Ainsi, nous pouvons introduire le maillage propre d'un polygone et l'articulation des maillages :

1. *Maillage propre d'un polygone* :

Soit R un domaine polygonal avec $N(R)=\{n_k\}_{k=1}^{K_R}$ l'ensemble des sommets du polygone *Frontière(R)* et $M(R)=\{m_l\}_{l=1}^{L_R}$ l'ensemble des points à l'intérieur du domaine polygonal R : $M(R) \subset$ *Intérieur*(R) .
Une triangulation de Delaunay contrainte définie sur l'ensemble des nœuds $N(R) \cup M(R)$ sera appelée *maillage propre d'un polygone*.

2. *Articulation des maillages* :

Soit R_i, R_j deux domaines polygonaux adjacents dans le plan image introduits ci-dessous, tels que *Frontière*$(R_i) \cap$ *Frontière*$(R_j) \neq \phi$.
Soit Δ_i et Δ_j les `maillage propre' respectifs de R_i et R_j.
Soit N_{ij} l'ensemble des sommets polygonaux communs de *Frontière*(R_i) et *Frontière*(R_j) , E_{ij} l'ensemble des segments communs entre les sommets N_{ij}.
Notons par $N\Delta_i$ ($N\Delta_j$ respectivement) l'ensemble des nœuds des maillages Δ_i (Δ_j respectivement) et par E_{Δ_i} (E_{Δ_j} respectivement) l'ensemble des arêtes du maillage Δ_i (Δ_j respectivement). Alors les maillages Δ_i , Δ_j sont articulés si :

- $N_{ij} \in N\Delta_i$ et $N_{ij} \in N\Delta_j$; $E_{ij} \in E\Delta_i$ et $E_{ij} \in E\Delta j$
- et pour tous les nœuds des ensembles $N\Delta_i$ et $N\Delta_j$, aucun n'appartient à l'intérieur des segments E_{ij}..

L'union des maillages propres articulés deux à deux s'appelle le `maillage articulé' fondé sur la segmentation polygonale. La figure 1 illustre ces définitions. Nous représentons le maillage `propre d'un polygone' sur la figure 1-a, alors que les divers cas d'articulation sont présentés sur les figures 1-b,c,d.

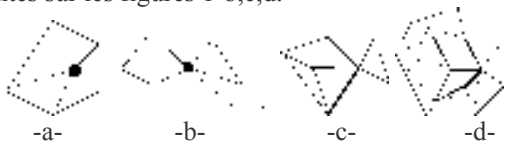


Figure 1 Illustration du maillage articulé. a- maillage 'propre d'un polygone' ; b- articulation par sommet ; c- articulation par frontière extérieur ; d- articulation par frontières intérieurs (trous).

2.1.2 Modèle dynamique du maillage articulé

Soit à l'instant *t*, $\{R_i^t\}$ l'ensemble des régions polygonales de la segmentation d'un objet et $\{\theta_r^t\}_j$ l'ensemble des modèles du mouvement associé à chaque région R_i^t. Le vecteur de déplacement pour tout nœud m_{ie} du maillage à l'intérieur de la région R_i est celui défini par un des modèles $\theta_i^* \in \{\theta_r^t\}_i$: $\vec{d}m_{ie} = \vec{d}(\theta_i^*)$. Pour un nœud $n_{ie} \in N(R_i)$ situé sur la frontière du domaine polygonal R_i nous considérons l'ensemble des domaines polygonaux $RP = \{R_p\}$ pour lesquels ce nœud est commun ($R_i \in RP$); remarquons que si n_{ik} appartient à la frontière extérieure du domaine $S^t = R_1^t \cup R_2^t \cup ... \cup R_i^t \cup R_I^t$ (silhouette de l'objet) alors *RP* ne contient qu'un seul élément R_i. Notons par θ_p^* le modèle du mouvement choisi parmi les modèles $\{\theta_r^t\}_p$. Alors $\vec{d}m_{ie} = \vec{d}(\theta_i^*)$. Remarquons que l'ensemble $\{\theta_r^t\}_i$ des modèles du mouvement d'une même région R_i^t peut être issu de la hiérarchie des segmentations par exemple. Nous expliquerons les critères de choix d'un modèle pour le calcul du vecteur de déplacement nodal au niveau de la section 2.3.
Après avoir introduit le modèle géométrique, et le modèle dynamique pour un maillage articulé, nous allons exposer la méthode de construction géométrique du maillage fondé sur la segmentation.

2.2 Construction géométrique du maillage

La triangulation est réalisée par l'algorithme géométrique de la triangulation contrainte de Delaunay proposé dans [5]. Cet algorithme commence par construire une première triangulation sur le domaine polygonal de départ délimités par les contours du suivi-temporel polygonal en n'utilisant comme nœuds que les sommets polygonaux existants. Ensuite on procède par insertion des nœuds intérieurs afin d'assurer dans la mesure du possible la propriété de Delaunay pour chaque triangle. Ces points dits 'points de Steiner' sont insérés de façon récursive en employant l'algorithme de [6] jusqu'à ce que tous les triangles satisfassent la contrainte de qualité géométrique, à savoir des contraintes sur l'angle minimum et le triangle maximum. Une fois le maillage construit, on calcul le modèle dynamique du flot optique nodal en utilisant les travaux de [6].

2.3 Affectation des vecteurs de déplacement nodaux en fonction du modèle de mouvement des régions

Nous allons décrire à présent la le procédé de calcul du vecteur du déplacement nodal et le choix du modèle du mouvement-région associé. L'objectif de la double représentation `segmentation polygonale'/`maillage triangulé' consiste à exploiter la hiérarchie des segmentations afin d'affecter un meilleur mouvement localement (à un ou plusieurs nœuds à l'intérieur d'une région).
La segmentation en régions polygonales utilisée en amont du maillage [7] permet de maintenir un découpage cohérent au cours du temps d'un VOP en zones de mouvement homogène. Ce modèle est hiérarchique : une région d'un VOP est le sommet de la pyramide des régions correspondant à des niveaux de segmentation plus fins. La méthode d'estimation de mouvement par région polygonale repose sur le modèle de mouvement 2D de type affine réduit : $\Theta = (t_x, t_y, k, \theta)^T$. Le vecteur de mouvement de chaque pixel appartenant à une région R donnée s'exprime à l'aide des quatre paramètres t_x, t_y, k et θ par :

$$\begin{cases} dx = t_x + k(x - x_g) - \theta(y - y_g) \\ dy = t_y + \theta(x - x_g) + k(y - y_g) \end{cases}$$

où (x_g, y_g) sont les coordonnées du centre de gravité de la région R, t_x, t_y, les paramètres de translation, k le paramètre de divergence et θ le paramètre de rotation. La méthode d'estimation choisie s'appuie sur la minimisation de l'erreur quadratique de compensation de mouvement par région (DFD). L'optimisation de cette fonctionnelle est effectuée par une méthode de descente de gradient, associée à une méthode de relaxation déterministe. Ainsi chaque région R est définit par :

- Son modèle C (approximation polygonale de son contour) ;
- Son modèle de mouvement (le vecteur Θ) ;
- La qualité de sa compensation de mouvement DFD.
- Sa profondeur relative par rapport au point d'observation et son appartenance ou pas au fond.

Le suivi temporel du maillage se base donc sur le suivi temporel de la segmentation spatio-temporelle. Il permet

d'obtenir le maillage Δ^{t+1} des VOPs de l'image suivante I^{t+1} à partir du maillage Δ^t des VOPs de l'image I^t et de la segmentation S^{t+1} déjà disponible par le suivi temporel polygonal. Ainsi l'affectation du vecteur de mouvement nodal est réalisée selon la distribution géométrique du nœud de la façon suivante :

- Si le nœud se trouve à la frontière entre plusieurs régions polygonales adjacentes. Le vecteur de déplacement du nœud est calculé à partir du modèle de mouvement affine de la région la plus proche de l'observateur. Si la profondeur des régions est identique, on prend alors la moyenne des vecteurs issus des modèles de mouvements de ces régions voisines. Dans ces calculs les régions du fond ne sont pas considérées.
- Si le nœud se trouve à l'intérieur d'une région polygonale. Le vecteur de déplacement du nœud est calculé à partir du modèle de mouvement affine de la région spatio-temporelle qui optimise sa compensation du mouvement. Cette région est puisée dans la hiérarchie des régions sur lesquelles le nœud donné se projète nécessairement à l'intérieur.

Ce maillage triangulé représente l'objet de façon plus fine que la partition région-polygonale. C'est le suivi de la segmentation polygonale qui permet d'assurer l'adéquation du modèle de représentation au contenu variable d'une scène vidéo. Le maillage triangulaire s'appuie sur ce suivi et permet d'obtenir une représentation plus souple et plus fine au sens du mouvement à chaque instant.

3 Suivi temporel conjoint

Le suivi temporel des VOPs dans une séquence vidéo consiste à suivre conjointement les nœuds du maillage triangulé et les segmentations hiérarchiques associées pour chacun d'eux. Ces deux aspects s'articulent de la façon indiquée dans la figure 2. Connaissant la segmentation spatio-temporelle S^t à l'instant courant t et l'image I^{t+1} à l'instant suivant, le suivi temporel orienté-régions consiste à produire la segmentation spatio-temporelle S^{t+1}.

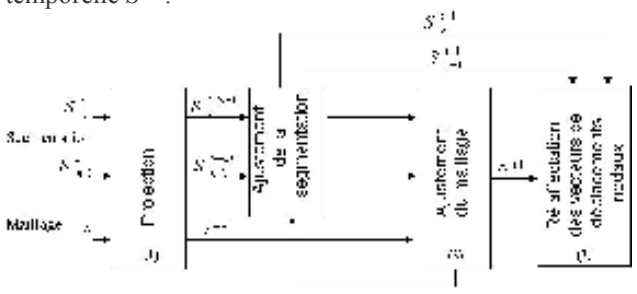


Figure 2 Principe du suivi conjoint

3.1 Cycle de vie d'un maillage

Le cycle de vie du maillage est schématisé en deux phases : 'Initialisation' et 'Suivi'. La phase d'initialisation s'appuie sur la segmentation pour la construction du maillage. Soulignons ici le lien entre le maillage d'un objet et la sémantique de la segmentation. Grâce aux techniques de gestion automatique du contenu que nous avons proposées dans [7], toutes les régions de la segmentation sont catégorisées en tant que *Fond* (fond de la scène), *Forme* (objet) et *Indéfini*. Ces dernières (*Indéfini*) se situent souvent sur les frontières des objets et peuvent comporter des parties des objets. C'est pourquoi nous triangulons systématiquement non seulement les régions de la classe *Forme*, mais également celles de la classe *Indéfini*. La deuxième phase du cycle de vie représente le suivi du maillage qui s'appuie sur la segmentation S^{t+1} disponible. Cette phase comporte des étapes typiques pour le suivi à savoir :

- prédiction ou projection du maillage sur I+1
- ajustement et ré-estimation du mouvement ou, dans notre cas, calcul du flot optique nodal en fonction des modèles du mouvement des régions sous-jacentes.

3.2 Suivi temporel en avant du maillage

Afin d'aboutir au modèle global du maillage (géométrique et dynamique), [6] propose une méthode d'affectation du vecteur de déplacement nodal en fonction du modèle du mouvement de la région sous-jacente. Dans cette étude nous avons approfondi cette approche et proposons une stratégie optimale d'affectation que nous exposons ci-dessous :

Nous cherchons à calculer ces vecteurs de déplacement *en avant* à partir des informations du mouvement et des images disponibles. Pour cela nous projetons le maillage dans le sens du temps et ajustons la position géométrique des nœuds. La somme des mouvements de la projection et de l'ajustement représente le vecteur du déplacement en avant, et permet de reconstruire les nœuds du maillage Δ^{t+1} à partir des nœuds de Δ^t.

- Projection du maillage : la projection consiste à calculer le vecteur $\vec{d}_i^{t\to t+1}=-\vec{d}_i^{t}$, où $\vec{d}_i^{t}$ est le vecteur du déplacement nodal affecté à partir du modèle du mouvement de la région sous-jacente. Ainsi les coordonnées $(x^{t+1/t},y^{t+1/t})$ d'un nœud projeté de l'instant t à $t+1$ seront $\begin{matrix} x^{t+1/t}=x^t+d_x^{t\to t+1}(x^t,y^t) \\ y^{t+1/t}=y^t+d_y^{t\to t+1}(x^t,y^t) \end{matrix}$ où (x^t,y^t) sont les coordonnées du même nœud à l'instant t ; (d_x,d_y) sont les coordonnées du vecteur du déplacement.
- Ajustement géométrique du maillage : le procédé d'ajustement du maillage s'appuie sur l'ajustement de la segmentation spatio-temporelle en régions polygonales qui tient compte à la fois des informations spatiales et

dynamiques de la scène. C'est pourquoi dans l'ajustement du maillage triangulé, nous accordons une confiance absolue aux frontières de la segmentation polygonale « ajustée ». L'ajustement du maillage en est donc simplifié et réalisé de manière purement géométrique par rapport aux contours de la segmentation spatio-temporelle. Il consiste à déplacer le nœud projeté pour correspondre au mieux aux contours réels présents dans le VOP (contours extérieurs et intérieurs). Cet ajustement s'appuie actuellement sur la segmentation à l'instant *t+1* (disponible au codeur). Chaque nœud du bord extérieur d'une région dans un VOP subit un mouvement contraint dans une fenêtre de taille limitée (11*11 pixels) et se déplace sur le bord de la région du niveau supérieur de la segmentation. Les nœuds de Steiner sont ajustés vis à vis des bords des régions des niveaux inférieurs de la segmentation quand cela est possible (contrainte de déplacement). Ainsi le vecteur de déplacement en avant est donné par : $\vec{d}_i^{t^+} = \vec{d}_i^{t \to t+1} + \vec{a}_i^{t+1}$ où $\vec{a}_i^{t+1}$ est le déplacement dû à l'ajustement sur le bord.

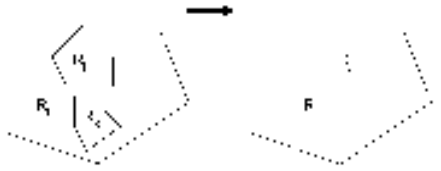
3.3 Evolution de la topologie

Le suivi en avant proposé permet donc non seulement d'ajuster le maillage par rapport à l'évolution de l'objet sous-jacent, mais aussi de fournir les vecteurs de déplacements nodaux nécessaires pour le codage des objets (triangulation) avec le mouvement en avant. Néanmoins l'ajustement géométrique seul s'avère insuffisant si la segmentation a subi des variations de topologie. Celles-ci sont dues à la complexité des mouvements articulés, à l'apparition de nouveaux objets, aux auto-occultations des objets. Afin de pouvoir décrire l'adaptation du maillage à ces situations complexes, nous allons aborder les questions relatives à l'évolution de la topologie des segmentations et des maillages. A cet effet nous définissons la notion de région topologiquement invariable (resp. variable) comme suit :

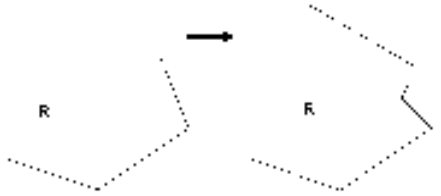
- Région topologiquement invariable (notée TI). Une région R_i est dite TI à l'instant t+1 si elle vérifie les deux conditions suivantes : *i*) elle existait à l'instant précédant t, *ii*) elle ne connaît pas de bouleversement topologique à l'instant t+1, autrement dit aucun trou n'apparaît à cet instant ou ne disparaît (typiquement un trou est une région dans la région).
- Région topologiquement variable (notée TVAR). Une région R_i est dite TVAR à l'instant t+1 si à l'inverse d'une région TI, elle ne connaît pas de bouleversement topologique à l'instant t+1.

La figure 3 illustre quelques exemples de régions topologiquement variables. Dans la figure 3-a, la région est de type TVAR car un trou y apparaît à t+1. Dans la figure 3-b, elle est de type TVAR du fait qu'un trou existant à t a disparu à t+1.Enfin dans la figure 3-c la région est de type TI car elle s'est juste déformée (évolution cohérente).

-a- TVAR



-b- TVAR



-c- TI

Figure 3 Exemple d'évolution de régions

3.4 Suivi temporel avec topologie variable

Le suivi avec topologie variable consiste à trianguler les VOPs de la première image et à les suivre au cours du temps jusqu'à ce que leurs régions deviennent topologiquement variables. La triangulation des régions TI d'une image à l'autre est suivie car le maillage associé à la région TI ne change ni en nombre de nœuds, ni la connexité. Notons qu'ici nous ne traitons pas le problème d'aplatissement des triangles et de la disposition des nœuds en conséquence. Quant aux régions de topologie variable TVAR, la variation de leur topologie signifie une apparition ou disparition des trous et exigerait un remaniement fort du maillage associé. Cela est une tâche complexe compte tenu de la variété des configurations topologiques possibles. Aussi pour simplifier, nous proposons de re-trianguler les régions TVAR comme s'il s'agissait de nouvelles régions, apparues entre des instants consécutifs. Le sur coût de codage dû à cette retriangulation est justifié par la qualité subjective de la scène décodée. Les régions nouvelles, inexistantes dans le passé sont évidemment triangulées et leur flot optique nodal est calculé en mode intra-image. Ainsi le maillage est construit en fonction des modifications de la segmentation, puis affiné localement de façon

hiérarchique. Dès qu'une région R_i est de type TI à l'instant t, son maillage $\Delta^{t-1}(R_i)$ est projeté à l'instant t, donnant lieu à $\Delta^{t}(R_i)$. Pour une région R_j de type TVAR à l'instant t, son maillage $\Delta^{t}(R_i)$ est reconstruit géométriquement comme s'il s'agissait d'une nouvelle région apparue à cet instant. Nous pouvons observer sur la figure 4 le suivi de VOPs avec topologie variable de t=5 à t=27 de la séquence `Children'. Nous y observons la projection du maillage des régions TI et la triangulation des régions TVAR. Par exemple, le maillage associé au ballon est retriangulé à t=25. Le maillage géométrique coïncide avec les contours simplifiés de la segmentation spatio-temporelle polygonale. Le mouvement assigné aux sommets des triangles permet un raffinement supplémentaire de notre représentation. L'articulation du maillage permet de limiter la re-triangulation aux seules régions TVAR à l'intérieur d'un VOP comme aux instants de détachement du ballon.

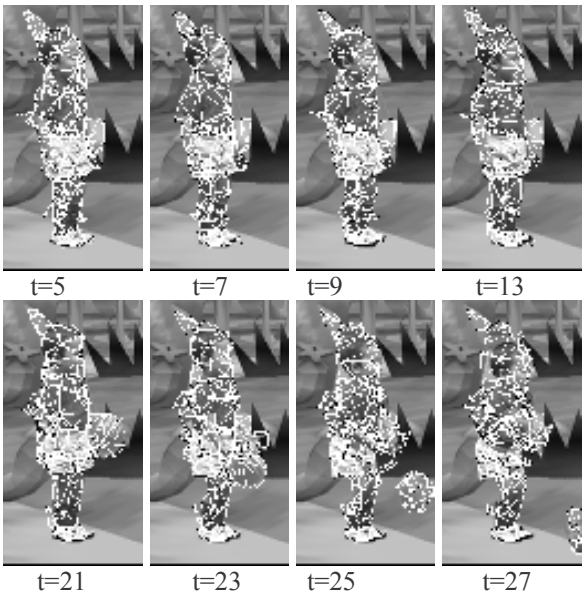


Figure 4 Suivi d'un VOP -topologie variable

4 Conclusions et perspectives

Dans cette étude nous nous intéressons à l'apport de l'association des maillages à la segmentation dans un procédé de suivi temporel. La segmentation spatio-temporelle est utilisée pour gérer les problèmes liés aux occultations, alors que le maillage est utilisé comme support pour la représentation du mouvement. Par construction, l'association de vecteurs de déplacement à chaque nœud du maillage permet l'interpolation immédiate du mouvement localement par rapport à un seul modèle par région. Le maillage utilisé est un maillage objet irrégulier et hiérarchique. Les contours de la segmentation spatio-temporelle polygonale permettent la construction de la triangulation de Delaunay contrainte. La hiérarchie de la segmentation est utilisée pour corriger localement les mailles et ainsi s'affranchir des phénomènes de dégénérescence. Du fait de cette architecture, la zone d'influence de la correction (effectuée pour adapter la maille au contenu) ne concerne que la région mise en défaut. Il n'est pas nécessaire de chercher à découper les mailles des régions voisines afin de préserver la conformité du maillage. Nous obtenons ainsi des mailles emboîtées adaptées au contenu et conformes. La méthode présentée (la représentation conjointe région/maillage) s'affranchit de toute connaissance a priori sur le contenu ce qui représente un atout pour le codage des objets génériques articulés. Cependant nous sommes loin d'avoir exploré l'ensemble des problèmes liés au suivi du maillage tels que l'aplatissement des triangles lors du suivi ou l'adaptation du maillage aux déformations fortes aux frontières de la segmentation. Ces questions seraient à explorer dans des travaux futurs.

Références

[1] B. Delaunay, " Sur la sphère vide. ", Bulg. Acad. Sci. URSS Class. Sci. Nat. 7, pp. 793-800 (1934)

[2] ISO/IEC JTC1/SC29/WG11 N2202. Information technology-Coding of audio-visual objects: Visual. ISO/IEC 14496-2 Committe Draft (MPEG4: Visual). Tokyo, March 1998.

[3] P.L George & H. Borouchaki, «Triangulation de Delaunay et maillage. applications aux éléments finis», Editions Hermès, Paris 1997.

[4] I. Babuzka, A.K. Aziz, "On the Angle Condition in the Finite Element Method", SIAM Journal on Numerical Analysis, vol. 13, no2, p. 214-226, 1976.

[5] J.R. Shewchuk, "Delaunay Refinement Mesh Generation", Ph.D. thesis, Technical Report CMU-CS-97-137, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1997.

[6] J. Benois-Pineau, H. Hornegger, D. Barba ,"Representation of hierarchical VOPs in video sequences by semi-Delaunay adapted meshes." *VLBV'98*, Urbana, IL, pp. 117-120, Oct 8-9 1998.

[7] A. Mahboubi, J. Benois-Pineau, D. Barba "Tracking of Objects in Video Scenes with Time Varying Content", *EURASIP Journal on Applied Signal Processing*, 'Special issue on Image Analysis for Multimedia Interactive Services -Part II', Volume 2002 Number 6 on June 2002, pp.582-594.

Codage conjoint source canal appliqué aux séquences d'images visioconférences

WANG S., CHATELLIER C. et OLIVIER C.

Laboratoire SIC, Université de Poitiers
Téléport 2, Bvd M. et P. Curie, BP 30179, F-86962 Futuroscope-Chasseneuil cedex
Wang+ sic.sp2mi.univ-poitiers.fr

Résumé

Dans cet article, nous nous intéressons au codage conjoint source canal appliqué à des séquences d'images de type visioconférences transmises sur canal gaussien. La méthode proposée repose sur une DWT et une quantification vectorielle adaptée aux différentes sous bandes d'ondelettes considérée. Après avoir rappelé l'algorithme le principe de ce codage appelé WTSOM (Wavelett Transform Self Organize Map) sur des images fixes, nous proposons une extension aux images en mouvement à partir de l'utilisation de dictionnaires 3D, puis une amélioration en terme de distorsion basée sur l'utilisation de différences d'images. L'exécution de cette nouvelle méthode nous permet d'obtenir une bonne qualité visuelle des images en mouvement même avec des TEB très élevés.

Mots clés

Codage conjoint source canal, visio conférence, Quantification vectorielle, transmission d'images.

1. Introduction

Afin de résister aux erreurs de transmission liées au canal gaussien, nous proposons dans cette contribution un codage d'image basé sur une quantification vectorielle (QV) appliquée à des coefficients en ondelettes et couplée à une modulation [1][2][3]. Le nombre d'états de la modulation étant idéalement identique au nombre d'indices du dictionnaire utilisé pour la QV [4]. Si, en terme de performance de débit, cette méthode est inférieure aux standards, elle présente le grand avantage d'être, par son codage à longueur fixe, bien plus robuste aux bruits que les classiques codes VLC usuelles. Déjà présentée dans le cas de transmission d'image fixes [3], la méthode employée est ici étendue et adaptée au codage d'images vidéo.

Dans le paragraphe 2, nous décrivons ainsi la méthode de codage conjoint WTSOM et présentons ses bonnes performances pour des images fixes transmises sur un canal gaussien. Nous proposons un algorithme appelé WTSOM 3D dans le 3ème paragraphe, qui adapte le précédent aux séquences d'images en considérant 3 images successives et WTSOM appliqué directement sur la série de 3 images (3D). Après avoir donné les limites de cette première extension, nous choisissons dans le 4ème paragraphe de travailler avec les différences entre images, en accord avec les stratégies de codages vidéo usuels. Nous proposons un nouvel algorithme basé sur ce principe, puis nous comparons ces deux méthodes en terme de qualité de reconstruction. Dans le 5ème paragraphe, nous montrons des résultats en transmission de cette dernière méthode WTSOM 3D avec différence sur différentes séquences via un canal gaussien caractérisé par son TEB (Taux d'Erreur Binaire). La robustesse de cette méthode est claire dans le cas présenté. Nous terminons cette contribution par une conclusion et les perspectives nombreuses pouvant faire suite à ce travail.

2. WTSOM

L'objectif du codage conjoint source canal appliqué à la transmission d'images est d'améliorer la qualité visuelle de l'image reçue dans des conditions de fort TEB, tout en minimisant la complexité globale de la chaîne de transmission. Afin de réaliser ce codage, nous avons proposé une association entre une transformation en ondelettes classique, une quantification vectorielle par sous bande et une modulation de type M-QAM.

Dans cette application, nous utilisons tout d'abord les ondelettes bi-orthogonales de Daubechies (9/7) [5] sur trois niveaux, pour lesquels nous ne conservons que les cinq sous-bandes les plus significatives : LL_3, HL_3, LH_3, HL_2 et LH_2 représentées sur la figure 1.

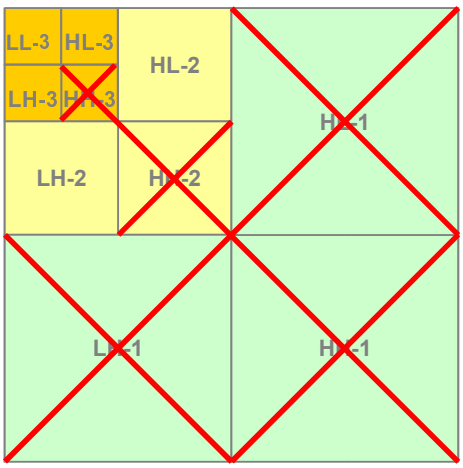


Fig. 1 : Décomposition DWT et sous-bandes retenues.

Dans un second temps, nous appliquons une quantification vectorielle différente pour chaque sous bande en utilisant les cartes topologiques de Kohonen et l'algorithme qui en découle : Self Organize Map (SOM) [6]. Seul les indices des dictionnaires sont transmis par l'intermédiaire d'une modulation choisi en fonction du canal : Gaussien [3] ou Rayleigh [7]. Cette méthode nécessite le calcul de cinq dictionnaires (pour chacune des 5 sous-bandes retenues), connus de l'émetteur et du récepteur. Le nombre et la forme des vecteurs ou blocs de chaque dictionnaire dépendent des sous bandes, du taux de compression et de la qualité voulue de l'image reconstruite. Pour notre application, chaque dictionnaire est composé de 256 vecteurs dont la taille est d'autant plus petite que la sous-bande traitée est plus informative.

La figure 2 représente la chaîne de transmission utilisant le codage WTSOM. On peut y observer la taille et la forme choisie pour les vecteurs ou blocs du dictionnaire suivant les 5 sous-bandes LL_3, HL_3, LH_3, HL_2 et LH_2. Pour une sous-bande donnée, l'indice du vecteur du dictionnaire le plus ressemblant au vecteur extrait de la sous-bande, au sens de la distance euclidienne, est transmis.

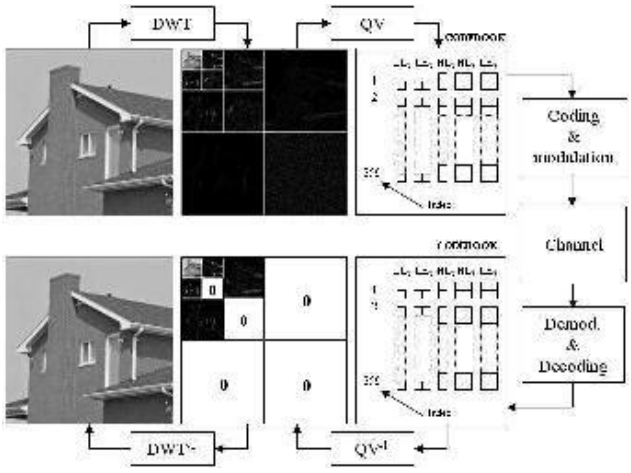


Fig. 2 : chaîne de transmission pour le codage WTSOM.

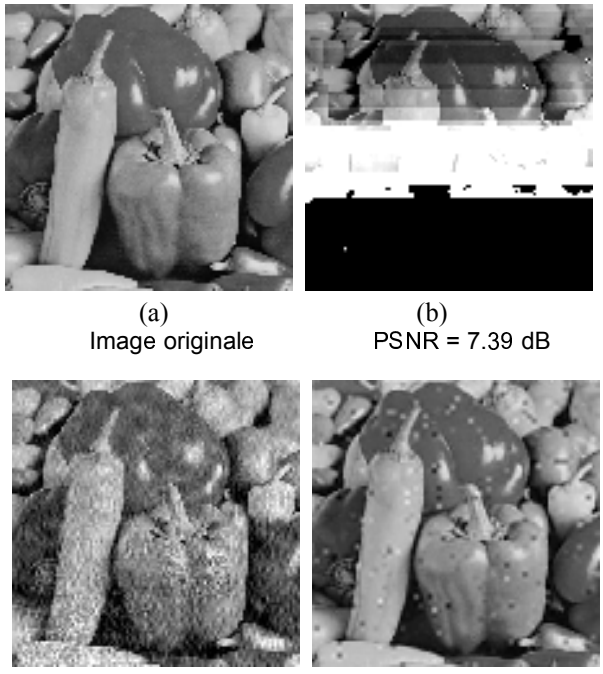
Un exemple de reconstruction d'images qui ne met pas en jeu les aléas dus à une transmission réelle est représenté sur la figure 3. Le taux de compression (T_c) a été fixé ici à 25. Ce résultat montre que la qualité d'image reste bonne même si elle ne peut évidemment pas rivaliser avec les autres codeurs (JPEG, JPEG2000).

La figure 4 permet de mettre en évidence le comportement du codeur WTSOM en présence d'erreurs lors d'une transmission. Pour cette simulation, les erreurs sont générées aléatoirement sur le flux binaire suivant une loi gaussienne. Afin de vérifier la robustesse des différents algorithmes en présence de bruit, nous avons testé JPEG et JPEG2000 pour des TEB faibles de l'ordre de $\mathbf{10^{-4}}$ puis

Image originale Image compressée WTSOM

Fig. 3 : Image compressée WTSOM : PSNR=29,37 dB

WTSOM pour des TEB beaucoup plus élevés de l'ordre de $\mathbf{10^{-2}}$. Ce choix se justifie d'une part, par le fait que pour des TEB trop élevés il n'est pas possible de reconstruire une image JPEG ou JPEG2000, et que d'autre part, pour un TEB trop faible avec WTSOM il n'y a pas d'artefacts visibles sur l'image reconstruite. Ce qui est observé se justifie par ce que nous savons : la grande sensibilité des codeurs usuels en cas d'erreurs. Les résultats montrent clairement que malgré quelques artefacts dus aux erreurs dans la sous-bande LL_3, la qualité de l'image WTSOM est visuellement bien meilleure.



(a) Image originale (b) PSNR = 7.39 dB

(c) PSNR = 19.57 dB (d) PSNR = 25.82 dB

Fig. 4: transmission d'images sur canal gaussien
a - image originale; b - JPEG, TEB = $2x10^{-4}$
c - JPEG2000, TEB = $2x10^{-4}$
d - WTSOM, TEB = $1,6x10^{-2}$

La méthode WTSOM se prête bien aux techniques de protections hiérarchiques d'informations. En effet, les seuls artefacts visibles sur l'image reçue (taches claires ou

foncées) ne sont dus qu'aux erreurs localisées sur la sous bande LL_3. Un code correcteur de type Reed Solomon dédié à la protection de cette sous bande a déjà été testé et a permis d'éliminer la quasi-totalité des artefacts visibles pour des TEB de l'ordre de 10^{-2} sur les deux types de canaux déjà cités [2,3].

3. WTSOM 3D

L'objet de ce paragraphe est de montrer comment il est possible d'appliquer directement le codage WTSOM à des séquences d'images, ce que nous appelons WTSOM 3D. Ce nouvel algorithme permet de traiter des séquences de trois images consécutives sur le même principe que pour les images fixes. Comme dans le cas précédent, cette méthode requière la fabrication de cinq dictionnaires dans lesquels chaque vecteur ou bloc est de taille variable mais de dimension 3 ou volumique. Les dictionnaires sont construits à partir d'une base d'apprentissage de séquences issues des vidéos suivantes : missa, mom, mom_daughter, grandmom, claire, susie. Une fois les dictionnaires connus, le processus de transmission s'effectue de la façon suivante :

1- Chaque image subie une transformation en ondelettes à 3 niveaux de décomposition.

2- Comme l'illustre la figure 5, un vecteur ou bloc 3D est extrait d'une sous bande (ici LL_3) puis comparé à l'ensemble des vecteurs du dictionnaire correspondant.

3- L'indice du vecteur le plus ressemblant, au sens de la distance euclidienne, est transmis.

4- Le récepteur décode l'indice reçu et reconstruit la sous bande à partir des vecteurs d'approximation.

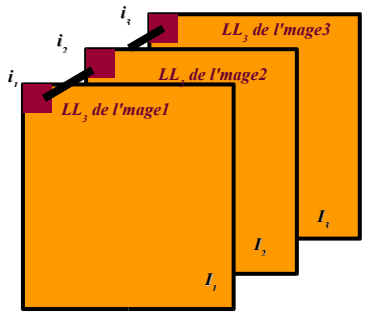


Fig . 5 : représentation d.un vecteur 3D (i_1,i_2,i_3) composé de trois coefficients d.ondelette des sous bandes LL_3 de la séquence I_1,I_2,I_3

La qualité visuelle de la séquence reconstruite dépend de la taille des vecteurs, du nombre de séquences d'apprentissage et du nombre de vecteur par dictionnaire. Pour notre cas, chaque dictionnaire est constitué de 256 vecteurs et nous avons utilisé une petite base d'apprentissage de 21 séquences de trois images issues de 6 vidéos pour construire les 5 dictionnaires. Nous présentons un résultat sur une séquence ne faisant pas parti de la base d'apprentissage. Ces premiers résultats obtenus sur des extraits de la séquence vidéo *Foreman* sont représentés sur la figure 6.

Fig. 6 : Reconstruction d.une séquence de 3 images avec WTSOM 3D. PSNR moyen = 19,81 dB

Ils permettent de conclure sur un point : la qualité de la séquence reconstruite n'est pas très bonne comparée à la séquence originale et ceci est le fait, notamment, du nombre restreint de vecteurs retenus (toujours 256) parmi beaucoup plus de vecteurs que dans le cas 2D, ainsi que du trop petit nombre de séquences d'apprentissages utilisées pour construire chaque dictionnaire. Augmenter le nombre de vecteur par dictionnaire reviendrait à augmenter la complexité de la chaîne de transmission et diminuer le débit utile. C'est pourquoi la seconde stratégie que nous mettons en oeuvre utilise les différences d'images comme cela est déjà utilisé dans d'autres codeurs vidéo et ceci afin de minimiser la variabilité des vecteurs d'approximation.

4. WTSOM 3D avec différence

Afin de minimiser la diversité des vecteurs présents dans les dictionnaires, nous exploitons la faible différence entre deux images consécutives après leur transformation par DWT. Il s'agit donc d'appliquer une transformée en ondelette, comme décrit au chapitre 2, sur trois images successives. On calcule alors une différence sur les coefficients d'ondelette, par sous bande conservée, entre les deux premières images et entre les deux suivantes :

$$Diff_1 = i_1 - i_2,\ Diff_2 = i_2 - i_3,$$

où i_1, i_2, i_3 représentent 3 coefficients à la même localisation spatiale (donc à la même sous-bande) de trois images successives et $Diff_1$, $Diff_2$ sont les différences entre les coefficients. Ainsi nous pouvons construire des dictionnaires à partir des vecteurs des différences. Nous proposons qu'une séquence de trois images soit traitée de la façon suivante :

1- Chaque image subie une transformation en ondelette.
2- On calcule les différences $Diff_1$ et $Diff_2$
3- La première image reste codée WTSOM et est transmise suivant le protocole décrit au paragraphe 3.
4- Les différences $Diff_1$ et $Diff_2$ sont transmises de la même façon, seuls les dictionnaires changent.

La figure 7 illustre les améliorations apportées par cette méthode par rapport à la méthode précédente puisqu'elle permet d'obtenir un gain de PSNR de l'ordre de 3,5 dB.

(a) WTSOM 3D (PSNR moyen = 19.81 dB

(b) WTSOM 3D avec diff. (PSNR moyen = 23.28 dB)

Fig. 7: Reconstruction de la même séquence codée avec WTSOM et avec WTSOM différence

Nous pouvons remarquer que la qualité se dégrade au fur et à mesure que l'image différence est éloignée de l'image de référence. En effet, si nous représentons l'évolution du PSNR des différentes séquences d'images représenté sur la figure 9, nous observons la cohérence des résultats concernant la baisse du PSNR en fonction de l'image traitée. Les améliorations possibles à envisager sont discutées au paragraphe 6.

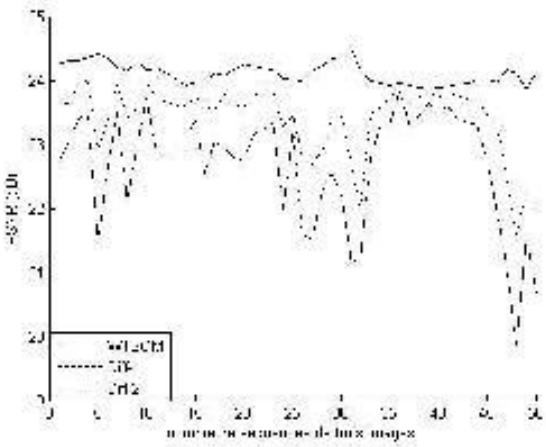


Fig. 8: PSNR des 3 images sur 50 séquences de la vidéo Forman
WTSOM : PSNR moyen=24,1 dB, σ = 0,0253
Diff-1 : PSNR moyen=23,41 dB, σ = 0,558
Diff-2 : PSNR moyen=22,68 dB, σ = 0,683

Nous nous proposons maintenant d'analyser le comportement de ces algorithmes lors d'une transmission sur canal bruité.

5. Résultats sur canal gaussien

Dans cette partie, tous les résultats sont traités par la méthode WTSOM 3D avec différence, et le taux de compression est d'environ $T_c = 35$. Pour justifier l'intérêt de cette méthode, nous transmettons une séquence vidéo (figure 9a: extrait de la vidéo *Mom* originale Mom_033, Mom_034, Mom_035). Nous construisons les dictionnaires dans les mêmes conditions que précédemment (§. 4) relativement au taux de compression choisi. En figure 9b., nous donnons le résultat de la reconstruction de cette séquence par WTSOM avec différence. Les séquences sont visuellement satisfaisantes avec un PSNR moyen de 31.38 dB. En figure 9c, nous transmettons la première séquence via un canal gaussien de TEB égal à $2{,}3.10^{-3}$.

(a)

(b) WTSOM 3D avec différence, sans erreurs
(PSNR moyen = 31.38 dB)

(c) WTSOM 3D avec différence, avec TEB = $2{,}3.10^{-3}$
(PSNR moyen = 29.36 dB)

Fig. 9: Une séquence transmise sur un canal idéal (sans erreurs) et sur un canal gaussien:
(a) la séquence originale
(b) WTSOM 3D avec différence, sans erreurs, PSNR moyen = 31.38 dB
(c) WTSOM 3D avec diifférence, TEB = $2{,}3.10^{-3}$, PSNR moyen = 29.36 dB

La valeur de PSNR moyen diminue de 2dB pour ce TEB de l'ordre de $2{,}3.10^{-3}$, ce qui prouve la robustesse de la méthode présentée et son intérêt pour le codage de vidéo. Les artéfacts observés proviennent uniquement des erreurs localisées sur la sous bande LL_3. Il est à noté que contrairement à la vidéo Foreman, certaines séquences de la vidéo *mom* ont été utilisées dans l'élaboration des dictionnaires. Ce cas peut donc être considéré comme

valide d'un point de vue robustesse vis à vis des erreurs de transmission, mais pas d'un point de vue qualité de reconstruction .

6. Conclusion et perspectives

Dans cet article, nous avons proposés un nouvel algorithme de codage conjoint source canal WTSOM 3D appliqué au codage de séquence d'images de type visio conférence. Il est basé sur une quantification vectorielle de coefficients d'ondelette regroupés par séquences de 3 images. Une amélioration de cette méthode, basée sur le traitement de différences d'images, appelé WTSOM 3D avec différence est également proposée et permet une diminution significative de la distorsion résiduelle sur les images reconstruites. Enfin nous montrons que cette stratégie se prête bien à la transmission d'images sur canal bruité en présence d'un fort TEB.

Ces résultats encourageants vont nous conduire à affiner la méthode (apprentissage, nombre de trames considérées) ainsi que la définition des dictionnaires tant en nombre d'indices que de formes des vecteurs. Différentes stratégies de constructions d'images différences doivent également être abordées. Enfin, la connaissance conjointe d'une part de l'impact visuelle d'une erreur sur l'image reconstruite, et d'autre part de sa localisation doit permettre d'appliquer des algorithmes de traitement d'images afin de restaurer l'image reçue.

7. Bibliographie

[1] Souhard B., Chatellier C., Olivier C. "On The Robustness of Joint Source/Channel Coding for Transmission Through an Ionospheric Channel", *HF2003, IEE Ninth International Conference on HF Radio Systems and Techniques,* Bath (UK), June 2003

[2] Souhard, B. "Codage conjoint source canal : Application à la transmission d'images fixes sur canal ionosphérique", *Thèse de l'Université de Poitiers*, mars 2004.

[3] Chatellier C., Bourdon P., Souhard B., Olivier C. ″An efficient joint source channel coding scheme for image transmission through the ionospheric channel″, *European Conference Propagation and Systems*, Brest (France), March 2005

[4] Aitsab O. "Turbo codes et codage conjoint source canal : application à la transmission d'images". *PhD thesis, Ecole Nationale Supérieure des Télécommunications de Bretagne*, Février 1998

[5] Antonini M., Barlaud M., Mathieu P., Daubechies I., "Image coding using wavelet transform" *IEEE Transactions of Image Processing, Vol. 1, Issue 2, pp 205-220*, April 1992**.**

[6] Kohonen T. "Self Organization and associative memory". *Springer-Verlag*, 1994.

[7] Boeglen H., Chatellier C. ″On the robustness of a joint source-channel coding scheme for image transmission over non frequency selective Rayleigh fading channels", *2nd IEEE International Conference on Information & Communication Technologie: From theory to applications.* Damas (Syria), April 2006

Protection de données à coût nul dans un codeur d'images multirésolution

Concours Jeune Chercheur : Non

Résumé

Les performances des codeurs d'images fixes ne sont plus uniquement évaluées à l'aide des courbes débit-distorsion. Les services fournis sont également un critère de choix. Dans cet article, nous proposons d'intégrer dans un codeur d'images multirésolution offrant des performances supérieures à l'état de l'art avec et sans perte, un schéma de protection de contenu. L'utilisation judicieuse du flux binaire généré par le codeur permet d'obtenir ce service à coût nul. Des éléments tant théoriques que pratiques sont avancés pour justifier l'emploi de ce schéma de protection.

Mots clefs

Protection de données, codage d'image avec et sans perte, multirésolution.

1 Introduction

L'évolution récente des codeurs d'images fixes montre pleinement leur efficacité. Qu'ils soient normalisés (JPEG 2000) ou issus de laboratoires de recherche, leurs performances en terme de courbe débit–distorsion sont meilleures que celles des codeurs de la génération précédentes (JPEG, par exemple). Ceci est également vrai pour les codeurs sans perte.

Said avait résumé en 1999 [1] les caractéristiques qu'il considérait comme souhaitables pour un schéma de codage d'images fixes, à savoir :

– scalabilité ;
– flexibilité et adaptabilité ;
– régulation automatique du débit et contrôle de la qualité ;
– unicité de l'algorithme ;
– décodage de régions d'intérêt ;
– faible complexité ;
– compression efficace ;
– résistance aux erreurs ;
– bonne qualité visuelle.

Les propriétés énoncées concernent pricipalement les qualités du codage de l'image, l'adéquation à la bande passante disponible et la facilité de mise en œuvre.

Depuis, l'Internet haut débit, le faible coût des supports de reproduction, la prise en compte des droits de la propriété intellectuelle et la gestion de la sécurité introduisent de nouvelles demandes pour les schémas de codage. Un codeur ne doit plus seulement être bon en terme de courbe débit-distorsion, il doit également fournir des services supplémentaires.

Ces services concernent, entre autres, la protection du contenu et l'insertion de données cachées. D'une part, la protection du contenu consiste principalement à garantir l'intégrité des données et à masquer le contenu. Les méthodes habituelles utilisées à ces effets sont respectivement le hachage [2] et le chiffrement [3]. D'autre part, l'insertion de données cachées vise soit à protéger les droits de l'image, soit à enrichir le document avec des métadonnées. Les techniques respectives peuvent être le tatouage [4] et la stéganographie.

Masquer le contenu d'une image est le plus souvent effectué via le chiffrement de tout ou partie de l'image. Ce chiffrement peut se faire dans le domaine direct ou le domaine transformé [5]. L'idée est donc d'interdire la visualisation de tout ou partie de l'image en l'absence d'autorisation (la clef de chiffrement).

Cet article propose d'utiliser directement une partie du flux binaire généré par le codeur LAR pour garantir la protection du contenu de l'image. Multirésolution, scalable en débit et en distorsion, avec et sans perte, en niveaux de gris ou en couleurs, ce codeur présente des performances au-delà de l'état de l'art.

Le présent document s'articule comme suit : une présentation succinte de la famille de codeurs LAR (section 2) et une analyse détaillée du flot binaire associée (section 3) permettent de définir une méthode de protection du contenu (section 4). Le raisonnement est suivi d'éléments de justification théorique permettant d'évaluer la pertinence de la méthode (section 5). Enfin, quelques résultats expérimentaux valident l'ensemble (section 6).

2 La famille des codeurs LAR

La famille des codeurs LAR (Locally Adaptive Resolution) adapte la résolution locale en fonction de l'activité de l'image. À une luminance localement uniforme correspond une résolution réduite. À l'opposé, une activité locale importante implique une résolution élevée. En outre, le schéma global de codage considère une image I comme étant la superposition de deux composantes : $I = \bar{I} + \tilde{I}$. Ainsi, $\bar{I}$ est une information globale tandis que $\tilde{I}$ représente la texture.

L'image globale est obtenue à partir d'un partitionnement quadtree par blocs carrés en utilisant un gradient morphologique comme critère d'activité. Un simple seuil d'activité

sert de principal paramètre. On appellera ce partitionnement partition LAR. À chaque bloc de cette partition est associé la valeur moyenne du bloc. La composition des deux fournit l'image $\bar{I}$. Cette image constituée de plateaux constants de taille variable est nommée image plate (flat LAR).

(a) image originale (b) partition LAR

Figure 1 – *Exemple de partition LAR*

Les différents codeurs LAR utilisant cette idée sont soit plats soit hiérarchiques. Ils diffèrent également par la manière d'encoder la texture, avec des approches prédictives, dans le domaine direct ou transformé. Un panorama complet est disponible dans [6] et [7]. Les résultats en terme de qualité de codage sont probants comme le montrent [8], [9] et [10].
Dans la suite de l'article, nous prendrons comme outil le codeur hiérarchique *Interleaved* S+P[10] comme base de travail. Par simplicité, nous l'appellerons LAR-H. La figure 2 donne un aperçu graphique de son fonctionnement.

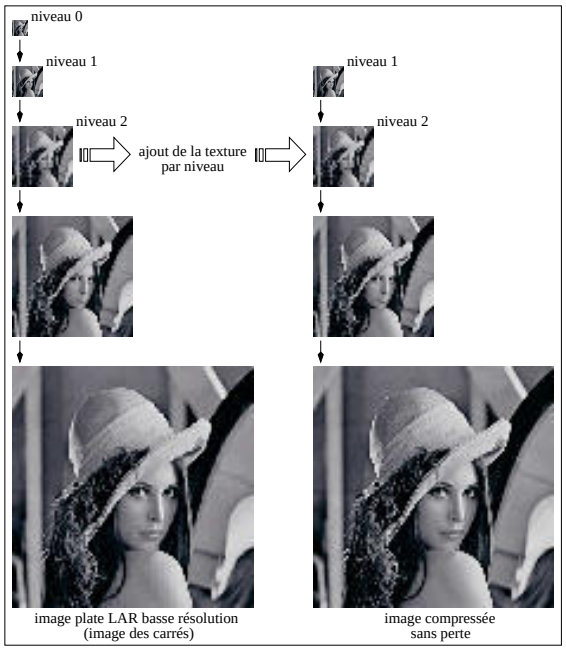


Figure 2 – *Fonctionnement du codeur LAR-H*

La superposition de l'image plate et de la texture se retrouve au niveau du codeur. Une première passe, à gauche sur la figure 2, assure la construction de l'image plate pour un niveau de résolution donnée. La seconde passe consiste à ajouter la texture par niveau, jusqu'au niveau souhaité. Ce procédé permet d'aller graduellement d'un codage avec pertes vers un codage sans perte.

3 Description d'un flux binaire LAR-H

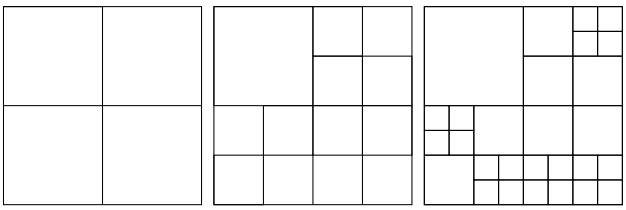
Cette section décrit succintement un flux binaire LAR-H. Ce flux comporte trois composantes entrelacées :

– le codage de la partition LAR ;
– le codage de l'image plate ;
– l'ajout de la texture.

La composante nous intéressant particulièrement est la partition LAR. Nous précisons donc la façon de la représenter.

3.1 Codage de la partition LAR

Le codage de la partition quadtree LAR se fait classiquement à partir d'une partition uniforme au plus haut niveau. À chaque passage au niveau inférieur, un bit transmis indique si le bloc est à découper. Ce codage est donc de type conditionnel. Le balayage se fait selon les lignes. La figure 3 donne un exemple pour lequel le flux binaire est le suivant : 0111 pour le passage entre le premier et le deuxième niveau, 0100 1000 0111 pour le passage entre le deuxième et le troisième niveau.



(a) Premier niveau (b) Deuxième niveau (c) Troisième niveau

Figure 3 – *Codage de la partition LAR*

3.2 Structure du flux binaire LAR-H

Le flux binaire LAR-H est scindé en deux. D'une part la partie permettant de reconstituer l'image plate au niveau de résolution souhaité (passe 1) et d'autre part, la partie ajoutant la texture à l'image plate (passe 2). La figure 4 présente le détail du flux. Dans chaque partie, le flux est décomposé en sous-flux, un par niveau. Chaque sous-flux de la passe 1 comprend le flux de la partition LAR (cf. 3.1) et le flux de construction de l'image plate.

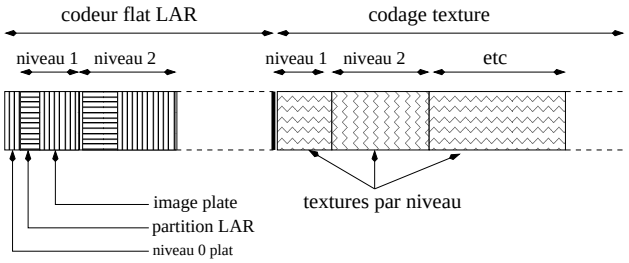
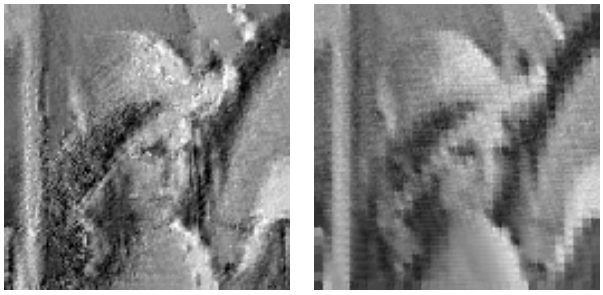


Figure 4 – *Flux binaire LAR-H*

3.3 Influence des erreurs du quadtree sur les images reconstruites

Le flux binaire LAR-H est sensible aux erreurs faites dans le décodage de la partition LAR. La figure 5 présente les effets de deux types d'erreurs différentes. La première est une erreur faite sur la taille d'un bloc au plus haut niveau de la partition LAR. L'effet de l'erreur se propage à toute l'image. La deuxième consiste, en l'absence de la partition LAR, à utiliser comme a priori une partition uniforme.

Deux remarques peuvent être formulées. D'abord, sans une parfaite connaissance de la partition LAR, une seule erreur est interdite, l'image reconstruite est très éloignée de l'image originale. Un schéma de protection par redondance du flux binaire LAR est proposé dans [11]. Ensuite, faire l'hypothèse d'une partition LAR uniforme au niveau le plus haut ne permet pas d'obtenir une image reconstruite de bonne qualité.



(a) erreur sur le premier bloc du niveau le plus haut (16x16)

(b) en considérant une grille uniforme (16x16)

Figure 5 – *Influence des erreurs du quadtree*

4 Un schéma de masquage du contenu

Les sections 2 et 3 présentent un codeur qui décorrèle deux informations : une image plate dérivant de la partition LAR et la texture associée. La section 3.3 montre l'effet d'une connaissance incomplète de la partition LAR sur les images reconstruites.

La partition LAR peut être utilisée pour masquer le contenu de l'image codée, éventuellement par niveau. Le schéma 6 détaille cette méthode. L'idée est d'ôter du flux binaire tout ou partie du flux correspondant à la partition LAR.

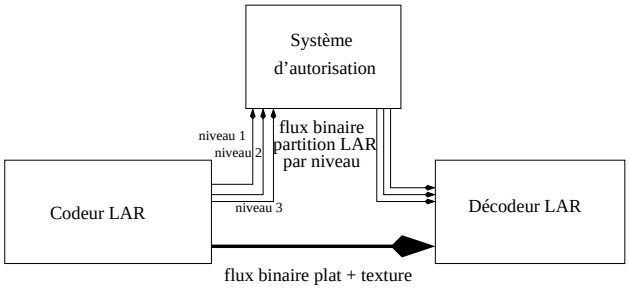


Figure 6 – *Proposition de schéma de masquage d'image par niveaux*

Cette méthode diffère des techniques de chiffrement habituellement proposées. Le schéma de masquage intégré au codeur LAR permet d'autoriser la récupération d'une image à différents niveaux de résolution.

Comment démontrer que la reconstruction de l'image sans connaitre la partition LAR est difficile ? Cette question amène quelques pistes :

- montrer que le nombre de partitions est grand ;
- montrer que, connaissant la partition à un niveau de résolution donné, il est difficile de trouver celle du niveau suivant ;
- montrer que la corrélation restant entre le flux binaire de la partition et les flux binaires de l'image plate et de textures est faible.

La section 5 présente quelques résultats théoriques sur le nombre de partitions LAR (section 5.1) et sur le passage d'un niveau à l'autre de la partition (section 5.2).

5 Étude théorique

5.1 Nombre de partitions LAR

Soit une image I de M lignes et N colonnes avec $S = M \times N$ pixels. Chaque pixel est représenté par q bits et peut donc prendre $Q = 2^q$ valeurs différentes.

$\mathrm{QP}^{[L_T..L_B]}(I)$ est une partition quadtree de l'image I avec des blocs de tailles allant de $L_T \times L_T$ pour le niveau le plus haut (premier niveau) à $L_B \times L_B$ pour le niveau le plus bas (dernier niveau). $\left|\mathrm{QP}^{[L_T..L_B]}(I)\right|$ est le nombre de partitions quadtree par blocs possibles pour une image I.

Soient $l_T = \log_2 L_T$, $l_B = \log_2 L_B$ et $l = l_T - l_B + 1$ le nombre de niveaux dans la partition quadtree, ou partition LAR. On note $\Omega^{[L_T..L_B]}$ le nombre de partitions LAR possibles sur un bloc de taille $L_T \times L_T$. $\omega^{[L_T..L_B]} = \log_2 \Omega^{[L_T..L_B]}$ est donc le nombre de bits nécessaires pour coder cette partition quadtree.

Pour illustrer notre propos, nous considérons une image 256×256 sur 8 bits avec une partition LAR $\mathrm{QP}^{[32..2]}$ sur 5 niveaux.

Le nombre de partitions quadtree $\Omega^{[L_T..L_B]}$ est lié à la fonction récursive Φ comme suit :

$$\begin{aligned} \Phi(0) &= 1 \\ \Phi(n) &= 1 + \Phi^4(n-1) \end{aligned} \quad (1)$$

En fait, $\Omega^{[L_T..L_B]} = \Phi(l-1)$. Le tableau 1 donne les premières valeurs de $\Phi(n)$. Il est à remarquer que $\log_2 \Phi(n)$ est le nombre de bits nécessaires pour représenter la valeur de $\Phi(n)$. Pour $n > 0$, $\Phi(n)$ peut être approché grossièrement par $\Phi_a(n) = 2^{(4^{n-1})}$. Ceci établit un coût de codage du quadtree $C(\text{QP}^{[L_T..L_B]})$ à 4^{l-2} bits par bloc $L_T \times L_T$.

n	$\Phi(n)$	$\log_2 \Phi(n)$	$\Phi_a(n)$
0	1	0	1
1	2	1	2
2	17	4.09	2^4
3	83522	16.35	2^{16}
4	4.86×10^{19}	65.40	2^{64}

Tableau 1 – $\Phi(n)$ *et* $\Phi_a(n)$ *pour les premières valeurs de* n

Pour notre exemple, $\Omega^{[32..2]} = 4.86 \times 10^{19}$ et il faut jusqu'à 65.40 bits pour coder la partition quadtree sur un bloc 32×32.
Le nombre de partitions LAR pour l'image entière I est simplement calculé en considérant que chaque bloc de taille $L_T \times L_T$ contient une partition quadtree $\text{QP}^{[L_T..L_B]}$. $\left|\text{QP}^{[L_T..L_B]}(I)\right|$ est alors donné par :

$$\left|\text{QP}^{[L_T..L_B]}(I)\right| = \Phi(l-1)^{\frac{MN}{L_T^2}} \quad (2)$$

Le coût de codage est souvent exprimé en bit par pixel (bpp). Ce coût est donné par :

$$\frac{\log_2 \Phi(l-1)}{L_T^2} \quad (3)$$

En continuant avec notre exemple, $\left|\text{QP}^{[32..2]}(I)\right| \approx 2^{4186}$. Cela indique que le coût de codage sera au maximum de 4186 bits soit 0.064 bpp.

5.2 Passage d'un niveau à un autre dans la partition LAR

Il s'agit de répondre à la question suivante : connaissant la partition LAR à un niveau donné, sachant le nombre de blocs à décomposer, combien y-a-t'il de partitions LAR de niveau suivant ?
La partition initiale, en blocs de taille maximale $L_T \times L_T$ est toujours la même. Il y a $b = \frac{MN}{L_T^2}$ blocs différents. Parmi ces blocs, d seront découpés en blocs de taille inférieure. Le nombre de partitions LAR de passage, P_{L_T} est donc une combinaison :

$$P_{L_T} = C_b^d \approx 2^{bH_2(\frac{d}{b})}$$

avec $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ l'entropie binaire. La figure 7 présente le tracé de $H_2(p)$. En tenant compte de $0 \leq H_2(x) \leq 1$, de $0 \leq \frac{d}{b} \leq 1$ et $H_2(x)$ maximale pour $x = 0.5$, il vient :

$$0 \leq P_{L_T} \leq 2^b$$

et P_{L_T} maximale pour $d = \frac{b}{2}$. En fait, si zéro ou tous les blocs sont à découper, il n'y a aucune difficulté à retrouver la partition suivante. En revanche, si le nombre de blocs à découper est proche de la moitié des blocs, le nombre de partitions possibles est de l'ordre de 2^b.
En reprenant notre exemple canonique, $b = 512^2/32^2 = 256$ et $0 \leq P_{32} \leq 2^{256}$. Un calcul identique peut également être fait entre deux niveaux quelconques de la partition LAR.

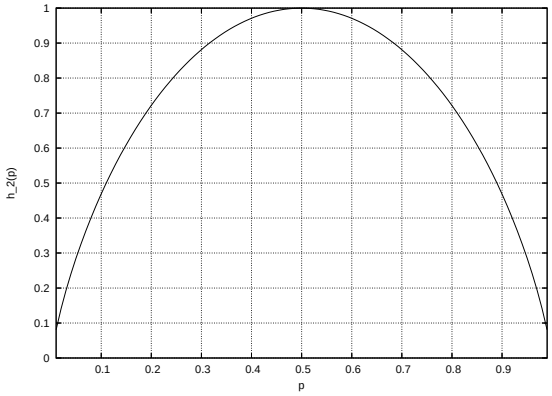


Figure 7 – *Fonction entropie binaire* $H_2(p)$

5.3 Remarques

La section 5.1 montre le grand nombre de partitions LAR possibles sur une image. Deux remarques viennent nuancer ce résultat.
La première concerne le résultat de l'équation (2). Ce résultat repose sur l'hypothèse que les blocs d'une image sont indépendants. Ce qui implique que le coût de codage fourni par (3) est un coût maximal. La présence d'une corrélation entre blocs fera baisser l'entropie du flux binaire généré et donc le coût de codage. L'expérimentation présentée à la section 6 montre que l'entropie restante est toujours importante.
La deuxième remarque indique que l'espace des images est toujours beaucoup plus grand que l'ensemble des partitions LAR. Dans l'exemple utilisé, chaque pixel est représenté par 8 bits, mais le coût maximum de codage de la partition LAR est de 0.064 bit. Ceci explique que la partition LAR ne peut pas servir d'identifiant unique pour une image, comme clef de hachage par exemple. En effet, nombre d'images différentes partagent la même partition LAR.

La section 5.2 indique que le nombre de partitions d'un niveau au suivant dépend fortement de la proportion de blocs à découper. Si cette proportion est proche de $1/2$, le nombre de partitions est immensément grand. Par contre, pour une proportion éloignée de $1/2$, ce nombre peut être réduit à 1. Valider ce résultat revient à vérifier expérimentalement la proportion de blocs à découper.

6 Experimentations

6.1 Présentation des résultats

Les images utilisées pour l'expérimentation sont `lena`, `airplane`, `baboon`, `goldhill`, `man`, `pepper` et `woman`, toutes de taille 512×512 et codées sur 8 bits. Les partitions LAR sont de type $\text{QP}^{[64..2]}$ soit au moins 6 niveaux dans la pyramide. Dans ce cas, l'entropie maximale pour la partition LAR est de 0.06387 bpp.
La figure 8 montre l'influence du seuil d'activité sur l'entropie de la partition LAR. En effet, le seuil pilote le processus de découpage en quadtree. Plus précisement, lorsque le seuil augmente, le nombre de régions diminue, avec des blocs de taille plus grande. La valeur du seuil est fixée par défaut à 30 (sur une échelle de 256 niveaux de gris).
Pour les images de l'expérimentation, l'entropie de la partition LAR reste au-dessus de 0.03 bpp (environ la moitié du coût de codage) pour une large plage de seuil. Avec une telle entropie pour la partition LAR et en prenant une image de taille 512×512, l'entropie de la partition LAR est d'environ 7864 bits pour toute l'image. Ainsi, le nombre de partitions LAR possible est de l'ordre de $2^{7864} \approx 10^{2367}$. Même si décoder un flux binaire ne prenait que 1μs, une attaque de type *brute force* ne serait pas envisageable.
À l'inverse, si la protection de l'image devait durer 100 ans, c'est-à-dire $3.156 \times 10^9 s$, l'entropie que devrait avoir la partition LAR ne devrait être que de 51.5 bits, soit 0.0002 bpp, toujours pour une image 512×512.

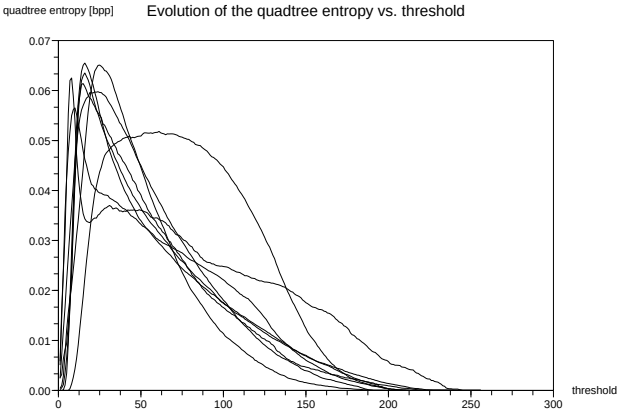


Figure 8 – *Entropie de la parition LAR en fonction du seuil d'activité*

La figure 9 montre l'influence du nombre de blocs sur l'entropie de la partition LAR. Le graphe a l'allure de la fonction entropie binaire $H_2(p)$ comme sur la figure 7. Cela démontre bien que l'entropie de la partition LAR est maximale lorsque le nombre de blocs est égal à la moitié du nombre maximal de blocs possible.

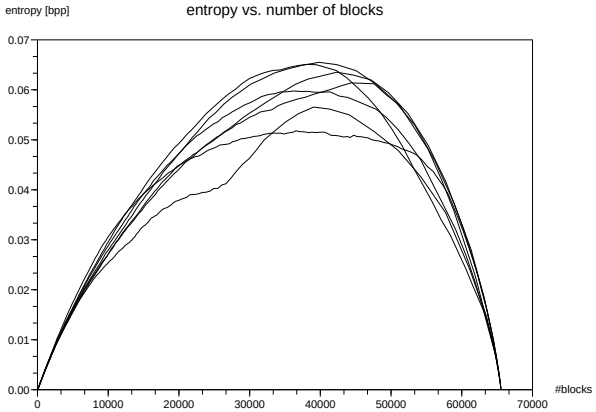


Figure 9 – *Entropie de la partition LAR en fonction du nombre de blocs*

Comme la partition LAR est transmise progressivement (cf. section 3.1), l'entropie de chaque niveau de la partition est calculée et le tableau 2 présente les résultats obtenus lorsque le seuil vaut 30. L'entropie du premier niveau est nulle en l'absence de blocs de taille 64×64 dans la partition LAR. Le second niveau possède une entropie d'environ 50 bits qui peut être suffisante pour fournir une protection raisonnable de l'image. De plus, l'entropie augmente avec la diminution de la taille des blocs, rendant la reconstruction de l'image sans la partition LAR plus difficile. L'image `airplane` fait exception, les 17 bits d'entropie étant insufisants. Dans tous les cas, le niveau le plus haut de la partition LAR peut être décodé, ne fournissant qu'une image de très basse résolution avec une grande distorsion.

image	64×64	32×32	16×16	8×8	4×4
lena	0	61	656	2916	8886
airplane	17	175	601	1922	7243
baboon	0	0	21	1647	10356
goldhill	0	65	218	2659	12218
man	0	35	466	2824	11014
pepper	0	41	667	3148	8245
woman	0	86	560	2810	9483

Tableau 2 – *Entropie des niveaux de la partition LAR en [bits] pour des images de* 512×512

Un dernier résultat expérimental concerne le nombre de partitions LAR possibles en passant d'un niveau à l'autre (cf. section 5.2). Le tableau 3 montre le nombre P_{32} de partitions LAR possibles à partir du décodage du niveau

de taille 32×32. Mis à part l'image baboon qui présente beaucoup de petits blocs (cf. tableau 2), P_{32} est assez élevé pour toutes les autres images.

image	$P_3 2$
lena	2.79×10^{17}
airplane	2.23×10^{51}
baboon	1
goldhill	6.24×10^{18}
man	8.81×10^{9}
pepper	3.69×10^{11}
woman	1.01×10^{25}

Tableau 3 – *Nombre de partitions P_{32} du passage du niveau du haut au suivant*

6.2 Discussion des résultats

La section 5.3 proposait quelques pistes d'expérimentations. Celles-ci mettent en évidence deux choses. La première est que le nombre de partitions LAR possibles pour des images naturelles est très grand. La seconde est que, même en connaissant la partition LAR pour un niveau donné, il reste très difficile, pour ne pas dire impossible, de déterminer la partition LAR pour le niveau suivant. Ainsi, la partition LAR apparaît comme un outil potentiellement utilisable pour effectuer une protection du contenu de l'image, avec un fonctionnement par niveaux. De plus, l'adaptation du seuil d'activité semble être le moyen de choisir le niveau de protection souhaité.

7 Conclusion et perspectives

Cet article présente quelques résultats sur l'intégration dans un codeur d'images fixes avec et sans perte d'un mécanisme de protection de contenu. Ce mécanisme utilise une partie du flux binaire généré par le codeur, sans coût additionnel. Il autorise des niveaux différenciés d'accès au contenu d'images codées avec un outil dont les performances, tant en sans perte que avec pertes, sont au-dela de l'état de l'art.

Le partitionnement LAR autorise à valider d'un point de vue théorique les bonnes propriétés de protection de contenu de la méthode. Les premiers résultats expérimentaux indiquent la faisabilité de l'ensemble. D'autres travaux doivent suivre, notamment la recherche de vulnérabilité de la méthode et la vérification de la faible corrélation entre le flux de la partition LAR et les flux codant l'image plate et de la texture. L'essentiel de la théorie dans ce domaine est inexistant, et l'expérimentation qui lui correspond est lourde.

Une application de ce travail est la mise en œuvre d'un système sécurisé d'archivage pour des images d'art haute résolution. Cette banque numérique d'images fournira un accès à ses fonds de manière sélective, avec différentes qualités d'image. Le projet TSAR[1] (Transfert Sécurisé d'image d'Art haute-Résolution) a pour objectif l'implantation d'un tel système d'archivage.

[1] http ://www.lirmm.fr/tsar

Références

[1] A. Said. Wavelet based image compression. Rapport technique, Imaging Technology Dept., Hewlett-Packard Labs, 1999.

[2] C. De Roover, C. De Vleeshouwer, F. Lefebvre, et B. Macq. Robust image hashing based on radial variance projections of key-frames. Dans *International Conference on Image Processing*, volume 3, pages II–77–80, 2005.

[3] M. Yang, N. Bourbakis, et S. Li. Data, image and video encryption. *IEEE Potentials*, 23(3) :28–34, Aug.-Sept. 2004.

[4] F. Davoine et S. Pateux, éditeurs. *Tatouage de documents audiovisuels numÃ©riques*. HermÃ¨s Science Publications, 2004.

[5] S. Lian, J. Sun, et Z. Wang. A novel image encryption scheme based on jpeg encoding. Dans *International Conference on Information Visualisation*, pages 217–220, 2004.

[6] O. Déforges. *Codage d'images par la méthode LAR et méthodologie Adéquation Algorithme Architecture. De la définition des algorithmes de compression au prototypage rapide sur architectures parallèles hétérogènes*. Habilitation à diriger des recherches de l'université de Rennes 1, 2004.

[7] M. Babel. *Compression d'images avec et sans perte par la méthode LAR (Locally Adaptive Resolution)*. Thèse de doctorat, INSA Rennes, 2005.

[8] O. Déforges et J. Ronsin. Region of interest coding for low bit-rate image transmission. Dans *ICME*, volume 1, pages 107–110, july 2000.

[9] M. Babel, O. Déforges, et J. Ronsin. Lossless and lossy minimal redundancy pyramidal decomposition for scalable image compression technique. Dans *ICME*, volume 3, pages 249–252, 2003.

[10] M. Babel, O. Déforges, et J. Ronsin. Interleaved S+P pyramidal decomposition with refined prediction model. Dans *ICIP*, volume 2, pages 750–753, 2005.

[11] M. Babel, B. Parrein, O. Déforges, et N. Normand. Secured and progressive transmission of compressed images on the internet : application to telemedicine. Dans *SPIE 17th annual symposium/ Electronic Imaging Internet - Internet Imaging*, volume 5670, pages 126–136, 2005.

Indexation d'Objets 3D Basée sur Les Séries de Fourier

E.Ait Lmaati(1) , Ahmed El Oirrak(1) , Driss Aboutajdine(2) Senior Member IEEE, Mohamed Daoudi(3) Member IEEE, M.N. Kaddioui(1)

(1)Faculté des sciences Semlalia, Dept Informatique, Marrakech, Maroc
(2)Faculté des sciences, LEESA-GSCM, BP 1014, RABAT
(3)Département Informatique et Réseaux, ENIC/INT
Cité scientifique - Rue Guglielmo Marconi
Villeneuve d'Ascq cedex France

lmaatimustapha@yahoo.fr, oirrak@yahoo.fr, aboutaj@fsr.ac.ma, daoudi@enic.fr

Concours jeune chercheur : Oui

Résumé

La taille des données 3D utilisées sur le Web devient de plus en plus très grande, par conséquent le développement des applications de reconnaissance d'objets 3D et des moteurs de recherche devient nécessaire. Dans ce papier on propose un nouveau schéma pour extraire la similarité entre les modèles 3D, en se basant sur les rayons maximales entre la surface de l'objet et son centre de masse et les séries de Fourier après l'alignement de l'objet en utilisant l'ACPC (Analyse en Composante Principale Continue).

Les vecteurs caractéristiques construisent par cette méthode sont invariants sous l'action de rotation, translation, réflexion et l'échelle. La méthode proposée est stable pour le bruit et le niveau de détail. Un moteur de recherche développé nous permet de tester la performance de ce descripteur nommé (RFS : Ray with Fourier Series) en utilisant une large base d'objets VRML2.0.

Mots clefs

Modèles 3D, indexation 3D, séries de Fourier, VRML.

1 Introduction

De plus en plus, la taille des données audio-visuel stockées autour du WEB devient énorme, par conséquent la description de ces données (texte, images, audio, vidéo, 3D objets, etc) est l'objectif de plusieurs chercheurs scientifiques. L'indexation de modèles 3D est l'un des domaines les plus récent qui permet la recherche des modèles similaires à un modèle requête dans une large base de données. La tâche la plus intéressante de l'indexation consiste à trouver un descripteur de forme pour extraire un vecteur caractéristique qui nous permet la mesure de similarité entre les éléments d'une base d'objets 3D. Plusieurs travaux sont faits dans ce sens, le descripteur de spectre de forme [1] proposé par Zaharia et Prêteux se base sur les courbures locales de la surface de l'objet 3D. Filali et Daoudi ont proposé un descripteur qui se base sur les vues caractéristiques de l'objet, il s'agit d'une méthode probabiliste qui sélectionne les vues intéressantes parmi plusieurs vues [2]. En se basant sur des statistiques, Osada et al ont proposé le descripteur nomé distribution de forme (D2)[3]. Paquet et Rioux ont proposé la méthode d'histogrammes de corde [4] qui repose sur des segments reliant le centre de gravité de l'objet aux centres de chaque triangle du maillage de l'objet. Le rapport aire/volume est utilisé comme vecteur caractéristique pour décrire les objets 3D par Zhang et Chen [5], malgré que cette description est rapide en calcul elle nécessite un maillage de bonne qualité (surface fermée, triangles orientés). Vranic et Saupe construit les vecteurs caractéristiques à partir de fonction complexe dans la sphère [6]. Parmi d'autre descripteurs, on site la méthode des rayon obtenues du centre de gravité de l'objet et l'intersection avec la surface de l'objet dans des directions données [7], cette approche n'est pas stable au bruit et elle nécessite une grande dimension pour les vecteurs caractéristiques construits, c'est pourquoi Vranic et al ont introduit la méthode de l'harmonique sphérique [8] afin de diminuer la dimension des vecteurs caractéristiques qui sont obtenus dans le domaine fréquentiel et d'avoir la stabilité au bruit.

Dans ce papier on propose de reconstruire une courbe 3D fermée à partir du maillage 3D après l'alignement de l'objets 3D par l'ACPC, puis d'appliquer les séries de Fourier qui sont utilisées dans la littérature [10,11], enfin d'extraire les vecteurs caractéristiques à partir des coefficients de Fourier calculés.

Les séries de Fourier nous permettent de calculer des quantités dans le domaine fréquentiel, puis les

normalisées afin d'avoir l'invariance à l'échelle. Ainsi le descripteur de forme proposé est invariant sous l'action de translation, rotation, réflexion et l'échelle, stable au bruit et aux niveaux de détails.

2 Représentation de l'objet 3D

Les objets 3D sont représentés par un maillage polygonal, avec des facettes triangulaires qui ne sont pas forcements orientés. Chaque objet est constitué de m facettes $\{F_i\}_{i=1...m}$ avec $F_i \subset \Re^3$ et n sommets $\{P_i\}_{i=1...n}$, $P_i = (x_i, y_i, z_i) \in \Re^3$.

Les facettes F_i sont déterminées par des points (P_j, P_k, P_l) avec $j,k,l \in \{1..n\}$, la surface de l'objet et donnée par $S = \cup_{i=1..m} F_i$.

3 Vecteur caractéristique basé sur les séries de Fourier

Le descripteur proposé utilise l'ACPC [9] qui est très souvent utilisé comme prétraitement dans le processus d'indexation 3D, c'est un ensemble de transformations affines appliquées sur l'objet 3D afin d'obtenir l'invariance au translation, rotation, réflexion et l'échelle. Ces transformations sont appliquées en chaque sommet P de l'objet.

Pour avoir l'invariance au translation on fait une translation du système de coordonnées au centre de masse G de l'objet, l'invariance au rotation est assurée en multipliant les coordonnées obtenus par la matrice de covariance V dont les colonnes sont les vecteurs propre unitaires classés selon l'ordre des valeurs propres associées, enfin on multiplie par la matrice F de réflexion afin de déterminer les directions des axes dans le nouveaux repère. Ces transformations affines ϕ sont données par l'équation suivante :

$$\phi(P) = F.V.(P - G).$$

La facilité de calcul des vecteurs caractéristiques invariants pour le facteur d'échelle nous permet de minimiser le temps de calcul et d'éliminer les erreurs dues aux approximations des intégrales dans le calcul du facteur d'échelle.

Le passage d'un objet 3D vers une courbe 3D fermée est la principale idée qui nous permet d'appliquer les séries de Fourier, et de passer ainsi du domaine spatial vers le domaine fréquentiel.

En calculant les distances maximales r_i entre la surface de l'objet et son centre de masse dans des directions u_i unitaires construisent par une hélice sphérique présentée dans la figure 1, on construit la courbe 3D correspondante. Les points d'intersections avec la surface de l'objet sont donnés par :

$$P_i = r_i u_i = (x_i, y_i, z_i),\ i = 1 \ldots N$$

Avec $P_1 = P_N$.

La figure 2, montre une courbe 3D qui correspond au modèle de la figure 3.

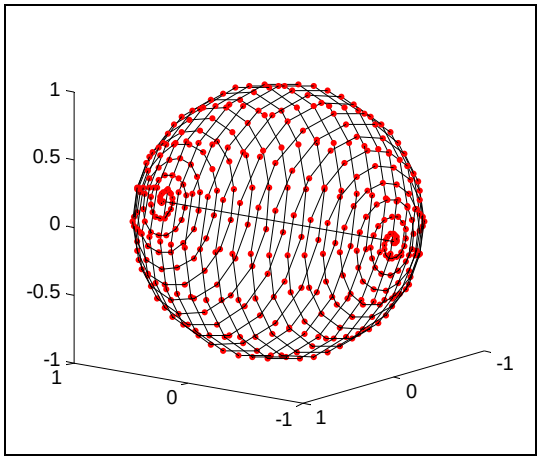


Figure 1 – *Courbe d'une hélice sphérique de nombre de points 400.*

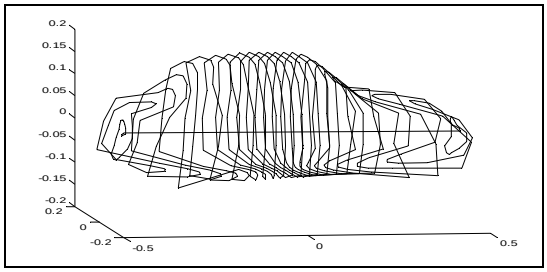


Figure 2 - *Courbe 3D construite par une hélice sphérique.*

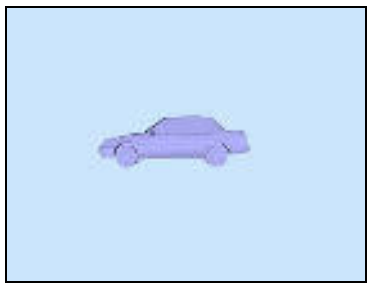


Figure 3 - *modèle m1524.off de la base de princeton shape benchmark.*

Comme résultat on a construit une courbe paramétrique X de période T dans un repère cartésien, cette courbe est donnée par l'équation :

$$X(t) = \begin{cases} x(t) \\ y(t) \\ z(t) \end{cases} \qquad t \in [0, T]$$

La reparamétrisation des courbes est une problématique dans le domaine de reconnaissance des formes. Très souvent connue dans la littérature l'abscisse curviligne normalisée se transforme linéairement sous l'action d'une similitude, tandis que la linéarité sous affinité est conservée pour la longueur affine. Dans notre cas ni le problème de point de départ ni le problème d'invariance aux affinités ne se pose pas puisque on a aligné les objets 3D en utilisant l'ACPC, alors on propose la paramétrisation invariante au facteur d'échelle donnée par l'équation:

$$\tau(t) = \frac{\int_0^t \sqrt{x^2(u)+y^2(u)+z^2(u)}\,du}{\int_0^T \sqrt{x^2(u)+y^2(u)+z^2(u)}\,du} = \frac{\int_0^t r(u)\,du}{\int_0^T r(u)\,du}$$

Pour extraire les vecteurs caractéristiques pour un objet 3D donné on applique la méthode des séries de Fourrier au fonction $r(\tau)$, on obtient la formule :

$$FS(r(\tau)) = a_0 + \sum_{n=1}^{\infty} c_n e^{jn2\pi\tau} + c_{-n} e^{-jn2\pi\tau}$$

avec $\quad a_0 = \int_0^1 r(\tau)\,d\tau$

Les coefficients de Fourrier sont donnés par:

$$c_n = \frac{1}{2}\left(a_n - jb_n\right)$$

avec $\quad \begin{cases} a_n = \int_0^1 r(\tau)\cos(2n\pi\tau)\,d\tau \\ b_n = \int_0^1 r(\tau)\sin(2n\pi\tau)\,d\tau \end{cases} \qquad n \neq 0$.

On constate que a_n et b_n sont des invariants relatifs sous l'action d'échelle, en effet :

Si $\tilde{X}(t)$ est la courbe correspondante à l'objet 3D après l'application de l'ACPC résumé par les transformations affines $\varphi(P) = s^{-1}.F.V.(P-G)$ ou s est le facteur d'échelle égale à la distance moyenne entre le centre de l'objet et ça surface, on a : $\tilde{X}(t) = s^{-1}.X(t)$ donc $\tilde{a}_n = \int_0^1 \tilde{r}(\tau)\cos(2\pi n\tau)\,d\tau = \int_0^1 s^{-1} r(\tau)\cos(2\pi n\tau)\,d\tau$

Alors on a : $\tilde{a}_n = s^{-1} a_n$, ou $\tilde{a}_n$ est le coefficient de Fourier pour la courbe $\tilde{X}(t)$, on aura le même résultat pour le coefficient b_n .

Les quantités complexes I_n données par la formule ci-dessous sont invariantes pour le facteur d'échelle.

$$I_n = a_n / a_1 + j\, b_n / b_1$$

Le vecteur caractéristique de chaque objets 3D est constitué des magnitudes de I_n pour n>1. Dans la pratique les premiers coefficients constituent le vecteur caractéristique car les coefficients de hautes fréquences sont affectés par le bruit.

4 Résultats Expérimentaux

Pour tester la performance des vecteurs caractéristiques proposés ainsi que nos algorithmes, on a formé des classes par aspect géométrique de l'objet dans la base de données du *Princeton Shape Retrieval and Analysis Group* [12] qui contient 1814 objets 3D en format off (*object file format*). La figure 4 montre des objets de quelques classes.

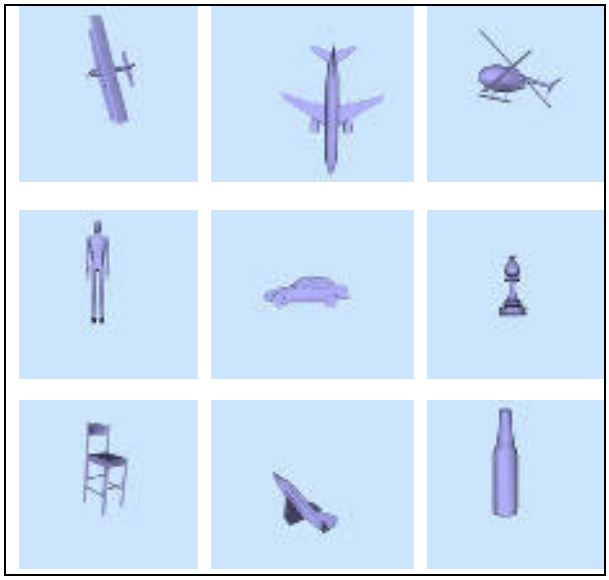


Figure 4 - *Modèles 3D de quelques différentes classes*

L'implémentation des différentes distances l_1, l_p, l_∞ avec $p \succ 1$ nous permet la mesure de similarité entre des paires d'objets et d'extraire les modèles les plus similaires du modèle requête.
Une interface web est développé, dont le serveur apache nous permet de faire des requêtes en ligne, le noyaux du moteur de recherche d'objet 3D est développé en PHP et Java. La figure 5 montre le résultat d'une requête pour une voiture, les 10 premiers objets les plus similaires sont extrais.
La figure 6 montre la courbe rappel - précision sur la classe voiture calculée pour les 20 premiers résultats retournés, en utilisant la norme l_2 et 200 comme dimension de vecteur caractéristique, avec des courbes 3D de 400 points. On remarque que le descripteur proposé donne de bons résultats.

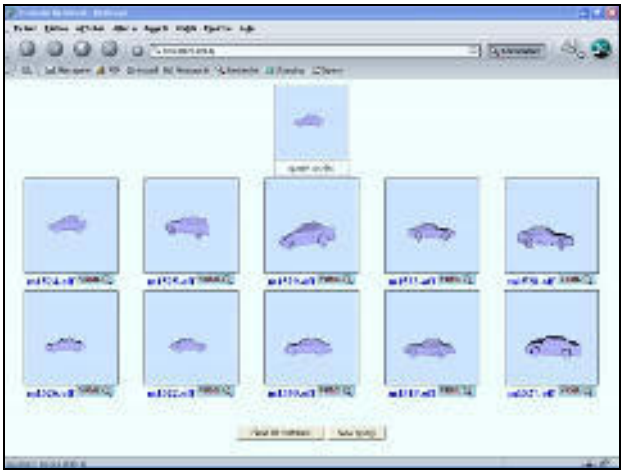


Figure 5- *requête pour une voiture.*

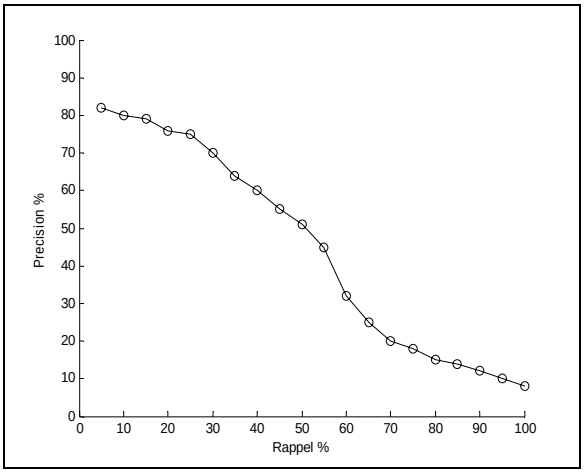


Figure 6-*courbe Rappel /Précision pour la classe voiture.*

5 Conclusion

Un descripteur de forme 3D basé sur les séries de Fourier est introduit dans cet article, efficace pour l'indexation 3D, stable au bruit et aux niveaux de détails, testé avec une large base d'objets 3D grâce au moteur de recherche développé. Les résultats obtenu par ce descripteur sont prometteurs, qui seront comparables à d'autres descripteurs classiques similaires comme l'harmonique sphérique [8] et la transformée de Fourier 3D [9].

Références

[1] MPEG-7 Video Group,"Information Technology - Multimedia Content Description Interface - Part 3: Visual," ISO/IEC FCD 15938-3 / N4062, MPEG-7, Singapore, March 2001.

[2] T. Filali Ansary, M. Daoudi, J.P. Vandeborre, 3D Model Retrieval Based on Adaptive Views Clustering in Proc. ICAPR 2005, Bath, UK, August 22-25, 2005, Part II.

[3] R. Osada, T. Funkhouser, B. Chazelle,, and D. Dobkin, "Matching 3D Models with Shape Distributions," in Proc. SMI 2001, Genova, Italy, May 2001, pp. 154 -166.

[4] E. Paquet, M.Rioux, "a query by content software for three – dimensional models databases management," in Proc. Int. Conf. on Recent Advances, 3-D Digital Imaging and Modeling, pages 345-352.

[5] C. Zhang, T. Chen, "Efficient feature extraction for 2d/3d objects in mesh representation. In IEEE International Conference on Image Processing, ICIP, Thessaloniki, Greece.

[6] D. V. Vranic and D. Saupe, "Description of 3D-Shape Using a Complex Function on the Sphere," in Proc. 2002 IEEE International Conference on Multimedia (ICME 2002), Lausanne, Switzerland, August 2002, pp. 177-180.

[7] D. V. Vranic, D.Saupe, "3D Model Retrieval," in Proc. Spring Conference on Computer Graphics an dits Applications (SCCG2000), B. Falsidieno, Ed.,Budmerice Manor, Slovakia, May 2000, pp. 89-93, Comenius University.

[8] D. Saupe and D. V. Vranic, "3D Model Retrieval with Spherical Harmonics and Moments," in Proc. DAGM 2001, B. Radig and S. Florczyk, Eds., Munich, Germany, September 2001, pp. 392-397, Springer Verlag.

[9] D. V. Vranic and D. Saupe, " 3D shape descriptor based on 3D Fourier transform," In K. Fazekas, editor, EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services, ECMCS 2001.

[10] A. Eloirrak, M. Daoudi, D. Aboutajdine, "Affine invariant descriptors using Fourier series, " in Pattern Recognition Lett. 23, 1109-1118.

[11] A. Eloirrak, M. Daoudi, D. Aboutajdine, “Affine invariant descriptors for color images using Fourier series,” in pattern Recognition Lett. 24, 1339-1348.
[12] P. Shilane, P.Min,M. Kazhdan, et T. Funkhouser. The princeton shape benchmark. Dans *Shape Modeling International*, June 2004.

Contribution à la création d'un moteur de recherche sémiotique : application aux manuscrits latins médiévaux

Y. Leydier[1] F. LeBourgeois[1] H. Emptoz[1]
[1]Laboratoire d'InfoRmatique en Images et Systèmes d'information
INSA de Lyon
20 av. Albert Einstein, 69621 Villeurbanne cedex
{yann.leydier, frank.lebourgeois, huber.emptoz}@liris.cnrs.fr

Résumé

*Cet article présente une méthode de recherche de mots par similarité de formes (*word-spotting*) dédiées aux manuscrits latins médiévaux. Nous proposons une nouvelle méthode de comparaison des formes qui tire avantage de la robustesse du gradient et tolère les variations spatiales. Nous testons notre algorithme sur plusieurs manuscrits latins médiévaux.*

Mots clefs

Manuscrits médiévaux, recherche de mots par similarité de formes, *word-spotting*.

1 Introduction

Le travail présenté dans cet article s'inscrit dans le projet « Formes et couleurs des manuscrits médiévaux : élaboration d'un outil de recherche », piloté par l'IRHT[1]. L'objectif du projet était de définir un outil de recherche facilitant l'accès aux gigantesques bases de données contenant les manuscrits médiévaux de l'IRHT. L'une des pistes explorées dans ce contexte concerne l'accès au contenu textuel des documents.
Certains mots jouent un rôle important dans les manuscrits médiévaux, comme par exemple « incipit » et « explicit » qui bornent les chapitres ou les livres dans un même volume. D'autres mots peuvent avoir un intérêt particulier pour les chercheurs, entre autres certains noms propres, ou, dans le cas de traduction, des mots dont le sens est inconnu et dont on cherche les différents contextes d'utilisation afin de déduire leur signification.

Il n'existe actuellement aucun système automatique capable de lire le texte des manuscrits médiévaux. Les principaux problèmes rencontrés ont pour origine la difficulté de segmenter le texte en mots, l'irrégularité de l'écriture, la complexité et la diversité des styles typographiques et le vocabulaire ouvert pour lequel nous ne disposons pas de dictionnaire.
Les systèmes de reconnaissance de texte utilisés industriellement aujourd'hui, les OCR (*Optical Character Recognition*), ne sont pas prévus pour traiter les images de documents manuscrits (voir figure 1). En fait, les OCR ne donnent des résultats corrects que sur les documents imprimés contemporains utilisant des polices de caractères usuelles (Times, Arial, *etc*.).

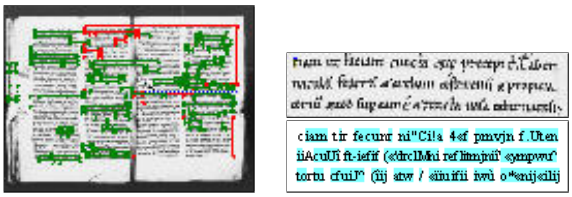


Figure 1 – *Segmentation et reconnaissance d'un manuscrit médiéval par Fine Reader.*

Une solution alternative consiste à limiter le processus à la reconnaissance d'un petit nombre de mots définis par les utilisateurs. Dans ce cas, il est possible de localiser avec précision toutes les occurrences d'un mot dans une image car nous en possédons toujours un modèle précis (défini par l'utilisateur).
La *recherche de mots par similarité de formes* (le terme *word-spotting* est plus souvent employé) est une technique permettant de localiser des mots choisis par un utilisateur dans un texte, écrit ou parlé, sans aucune contrainte [1, 2, 3, 4]. Cette approche générique peut être appliquée à tout type de document écrit, quel que soit son langage et qu'il utilise un alphabet, un syllabaire ou des idéogrammes… Il n'est pas nécessaire de créer une base d'apprentissage adaptée à chaque document ou à chaque scripteur.
Cette technique est utilisée lorsque la reconnaissance de mots est mise en échec, comme par exemple sur les documents très détériorés ou les manuscrits.
Dans notre cas, il s'agira de rechercher toutes les occurrences de l'image d'un mot. Le prototype, ou *mot-clé* est sélectionné par l'utilisateur en l'entourant sur une image du document à l'aide d'une interface graphique. Le mot-clé est comparé à des parties des images des pages du document en utilisant une mesure de similarité ou une distance. En fin de traitement, le système propose une liste d'images de mots triée par ressemblance avec le mot-clé. Cette dernière contient inévitablement des fausses détections que l'utilisa-

[1]Institut de Recherche et d'Histoire des Textes.

teur élague manuellement. Cette technique est, en fait, plus proche du domaine de la recherche d'image par le contenu (*Content Based Image Retrieval*, CBIR) que de la reconnaissance de mots.

Les documents qui nous ont été fournis sont écrits dans différents styles typographiques [5]. Ces derniers correspondent à plusieurs grandes classes (gothique, carolingien…) mais de nombreux manuscrits présentent des variations et des mélanges de ces styles si bien qu'il est impossible de créer des modèles qui pourraient représenter la totalité de notre corpus.
Bien que l'écriture dans les manuscrits médiévaux puisse sembler assez stable au profane, la production d'un même scripteur s'avère parfois très irrégulière. La présence de réglure sur les pages ne suffit d'ailleurs pas toujours à assurer un alignement correct des mots et de fortes courbures de lignes dues aux conditions de prise de vue sont parfois observées. L'espace entre les lettres et les mots est aussi très irrégulier sur une même ligne, souvent guidé par le souci de justifier le texte et d'éviter les césures.
De plus, un livre peut avoir été rédigé par plusieurs copistes. Cela ajoute aux irrégularités de l'écriture, le changement de scripteur étant souvent visible à l'œil nu.

2 État de l'art

La plupart des travaux effectués sur la recherche de mots dans les images de documents repose sur une segmentation du document en mots. Quelques rares auteurs ne segmentent les documents qu'en lignes, opération bien mieux maîtrisée que la segmentation en mots dans le cas des images de manuscrits [3, 6].
Une méthode de segmentation en mots basée sur la théorie de l'espace multi-échelle a été proposée [7] et a donné de bons résultats sur un corpus de manuscrits de George Washington. Elle n'a cependant été testée que sur ce corpus qui possède des caractéristiques très différentes du nôtre dont les documents présentent une structure physique complexe et sont souvent endommagés. Si nous tentons de segmenter nos documents, les résultats ne seront pas optimaux et la qualité des traitements suivants sera compromise.

Il est possible de comparer les pixels directement en utilisant des méthodes sophistiquées comme la distance SLH (*Scott et Longuet-Higgins*) [1] qui est robuste vis à vis des transformées affines, ou bien encore la distance de Hausdorff [4]. La plupart de ces méthodes nécessitent cependant une binarisation de l'image qui cause inévitablement une perte d'information importante.
Des mots segmentés peuvent être représentés par des vecteurs de caractéristiques. Ces caractéristiques sont très proches de celles utilisées pour la reconnaissance de caractères : profiles, projections, lissage gaussien [8], *etc*.
Lesdits vecteurs sont souvent comparés grâce à l'algorithme DTW (*Dynamic Time Warping*) [3, 8, 9] qui est robuste face aux variations spatiales. L'association de ces vecteurs de caractéristiques et de cet algorithme de comparaison est très efficace mais suppose une segmentation parfaite des mots, ce qui ne peut pas être réalisé sur nos documents.
D'autres caractéristiques comme les concavités [2] ou les coins [10] nécessitent des structures de données plus complexes pour les décrire (en général, des arbres ou des graphes). Ces caractéristiques sont très stables sur des documents propres mais elles sont peu adaptées aux documents de notre corpus.

Dans les études précédentes, certains résultats sont présentés en termes de précision [8] ou à l'aide de courbes ROC (*reciever operating characteristics*) qui représentent le taux de succès en fonction des fausses détections.
Certains auteurs préfèrent donner des valeurs numériques (par exemple le nombre de bonnes réponses) plutôt que des taux mais le manque d'information sur ces valeurs les rend peu facile à interpréter et comparer. Certaines études présentent la liste des images des n meilleures réponses pour certaines requêtes en guise d'exemples [11].
Récemment, des auteurs ont commencé à utiliser des courbes Précision–Rang [12] et Précision–Rappel [13], déjà utilisées dans le domaine de la recherche d'images par le contenu.

Les caractéristiques utilisées dans les méthodes présentées sont très classiques et nécessitent pour la plupart une binarisation des images, voire une segmentation en mots. Bien que la possibilité de segmenter certains manuscrits du Moyen Âge a été démontrée, tous ne sont pas segmentables. Afin de garantir que notre méthode pourra être appliquée à l'ensemble des documents de notre corpus, nous avons décidé de développer une méthode qui n'applique aucune segmentation, que ce soit en lignes, en mots ou même en graphèmes. De plus, nous n'effectuerons aucune binarisation afin de conserver un maximum d'information.

3 Comparaison élastique cohésive

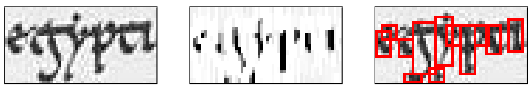
Nous avons testé différentes caractéristiques directement applicables sur les images en niveaux de gris et sans segmentation telles que le gradient, les valeurs propres de la matrice hessienne, la courbure des isophotes, la courbure des lignes de flux, *etc*.
Les expérimentations ont montré que les meilleurs résultats sont donnés par l'orientation du gradient lorsque sa magnitude est significative. Les performances baissent si on baisse le seuil sur la magnitude du gradient : plus le seuil est bas, plus le bruit entre en compte dans les calculs.
La distance entre deux gradients est définie ici par l'angle entre eux si les deux magnitudes sont supérieures au seuil. Si l'un des deux gradients a une magnitude trop faible, la distance prend la valeur d'une pénalité. Cette pénalité est égale au double de l'angle maximal possiblement formé par deux gradients. La distance entre deux gradients de faible magnitude est définie comme nulle. Ainsi, dans l'espace d'échelle σ, $\mathcal{G}$ étant l'opérateur gradient ($\|\mathcal{G}\|$ sa

norme et $\theta(\mathcal{G})$ son argument), la distance entre deux pixels a et b est définie par :

$$d_\sigma(a,b) = \begin{cases} \min(|\theta(\mathcal{G}^a) - \theta(\mathcal{G}^b)|, \\ \quad 128 - |\theta(\mathcal{G}^a) - \theta(\mathcal{G}^b)|), \\ \quad \text{si } \|\mathcal{G}^a\| > \varepsilon \text{ et } \left\|\mathcal{G}^b\right\| > \varepsilon \\ 0, \text{ si } \|\mathcal{G}^a\| < \varepsilon \text{ et } \left\|\mathcal{G}^b\right\| < \varepsilon \\ 255, \text{sinon} \end{cases}$$

De manière générale, pour qu'un algorithme de comparaison soit robuste face aux variations spatiales, chaque pixel du prototype doit être comparé à tous les pixels d'un voisinage sur l'image. Un tel processus est très lourd en calculs mais peut être simplifié. En ne comparant que des zones d'intérêt idoines, le temps de calcul diminue alors drastiquement.
Dans les manuscrits médiévaux, la plupart de l'information du texte est localisée sur des traits verticaux. Illustrons cela avec la lettre « u ». La distance entre les deux traits verticaux de deux occurrences de « u » manuscrits est toujours différente, ce qui met en échec les algorithmes de comparaison naïfs. Il est donc plus pertinent de comparer les pixels autour des traits verticaux que de comparer les deux formes dans leur intégralité.

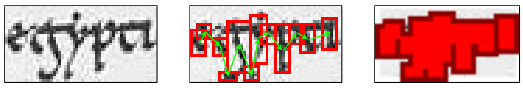
Nous calculons une signature des images par morphologie mathématique appliquée aux niveaux de gris. Nous effectuons une ouverture avec un élément structurant choisi en fonction du type de document. Un élément structurant idoine pour les manuscrits médiévaux latins est une ligne verticale dont la longueur n dépend de la taille des caractères. n est déterminé visuellement et de façon simple et intuitive par l'utilisateur grace à une interface graphique. D'autres éléments structurants, obliques par exemple, pourraient être utilisés pour compléter la signature du texte au risque d'y ajouter du bruit.
Nous obtenons ainsi une signature du texte composée de *guides* verticaux. En agrandissant le rectangle englobant de ces guides, nous obtenons des *zones d'intérêt* (voir figure 2). Notons que l'extraction de zones d'intérêt (ou ZI) n'est pas une segmentation en lignes ni en mots. En effet, nous ne cherchons pas à extraire des entités cohérentes d'un point de vue sémantique.



1) *Prototype* 2) *Guides* 3) *ZI*

Figure 2 – *Guides et zones d'intérêt du mot "egỳpti".*

Le prototype est divisé en morceaux suivant ses zones d'intérêt. La distance et l'orientation entre les centres successifs desdites zones sont stockées (voir figure 3.2). Les liens entre elles sont lâches (voir figure 3) si bien que le prototype peut être déformé afin de mieux s'apparier aux occurrences les plus déviantes (voir figure 4). Le déplacement possible de chaque zone d'intérêt permet un léger décalage vertical dépendant de la hauteur des caractères. Le déplacement horizontal doit être plus large sans pour autant permettre à des zones d'intérêt de se croiser ; ainsi en général une ZI ne doit pas être décalée de plus la moitié de la largeur moyenne des caractères. Ces paramètres sont réglés par l'opérateur car nous ne pouvons pas segmenter le texte en caractères.



1) 2) 3)

Figure 3 – *1) Un prototype. 2) Les liens entre les ZI. 3) En rouge, les ZI, en rouge foncé leur aire de déplacement possible.*

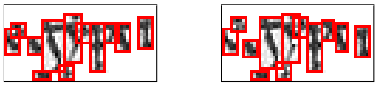


Figure 4 – *Un déplacement possible des ZI du prototype « egỳpti ». Le prototype est déformé afin d'être comparé à des mots dont la forme diffère légèrement.*

Afin de ne pas comparer le prototype naïvement à la totalité de l'image, nous avons décidé d'orienter la comparaison grâce aux aux guides de l'image. Ces derniers servent de points de départ, ensuite les liens entre les zones d'intérêt du prototype sont utilisés pour déterminer l'ensemble des zones à comparer
Le processus est illustré sur la figure 5. Les rectangles transparents sont les zones d'intérêt de prototype, les rectangles gris pointillés sont les guides de l'image. Seul la première ZI du prototype doit être calée sur un guide de l'image.

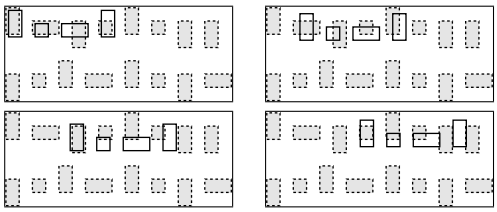


Figure 5 – *Exemple de comparaison cohésive.*

```
pour chaque guide G de l'image
    pour chaque déplacement d dans le voisinage
        comparer la 1ère ZI Z du prototype
                [avec l'image en coord(G) + d
    score(G) = min(resultats)
    X = coord(Z)
```

```
pour chaque autre zone d'intérêt Z du
                                [prototype
    pour chaque déplacement d dans le
                                [voisinage
        comparer Z avec l'image en
             [coord(G) + coord(Z) - X + d
    score(G) = score(G) + min(resultats)
    X = coord(Z)
```

4 Résultats

Nous avons créé une interface permettant de lancer une recherche sur un groupe d'images. Le mot clé est sélectionné par l'utilisateur en l'encadrant sur une image avec la souris.

4.1 Test MS14

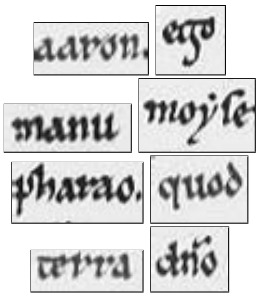
La première série de tests présentée a été effectuée sur 24 pages du manuscrit MS14 de la bibliothèque d'Amiens (voir figure 6.1), soit environ 11000 mots. Nous avons choisi les mots-clés selon deux critères : leur pertinence sémantique, pour simuler une utilisation réelle du système, et leur fréquence d'apparence, pour des raisons statistiques. Étant donné que le texte est en latin, nous avons coupé les déclinaisons des mots-clés afin d'en garder les racines. Des recherches ont été effectuées pour les mots-clés suivants : « aaron », « quod », « terra », « ego », « manu », « moyse- », « dño » et « pharao- ». Nous avons compté le nombre d'occurrences de chaque mot-clé et noté les occurrences césurées, avec une majuscule ou une lettrine séparément. En effet, notre méthode n'est pas prévue pour gérer ce genre de cas et nous avons voulu en tenir compte dans nos statistiques. Ainsi notre algorithme est sensé ne pouvoir localiser que 488 occurrences au mieux sur les 530 des huit mots-clés, soit 92,1%.
Au final, notre algorithme a localisé 437 occurrences de mots-clés (82,5%) soit 9,6 points de moins que la maximum théorique (voir table 1). La courbe P–R est donnée figure 6.3.

	Total	Localisé	%age
Complet	488	433	88,7%
Lettrine	1	1	Ø
Majuscule	34	1	2,9%
Césure	7	2	Ø
Total	530	437	82,5%

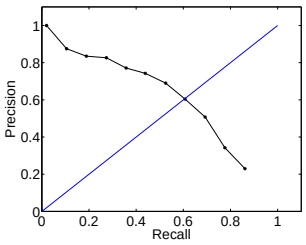
Tableau 1 – *Résultats sur le MS14.*

Les mots-clés donnant les plus mauvais résultats en termes de rappel sont « aaron » et « manu ». Le premier s'explique par la présence de deux « a ». Cette lettre est probablement celle qui a la plus grande variabilité dans le texte de ce document. De plus elle ne possède aucune hampe ou jambe pour la caractériser. Le second mot-clé contient un « m », un « n » et un « u ». Ces trois lettres peuvent facilement être confondues avec d'autres à cause de la faible résolution des images (par exemple, « m » et « nı », « nu » et « mı », *etc.*).

1) *Une page du manuscrit MS14.*



2) *Les huit prototypes.*



3) *La courbe P–R.* $P = R = 0.6$, $P(R = 1) = 0.1$

Figure 6 – *Le test MS14.*

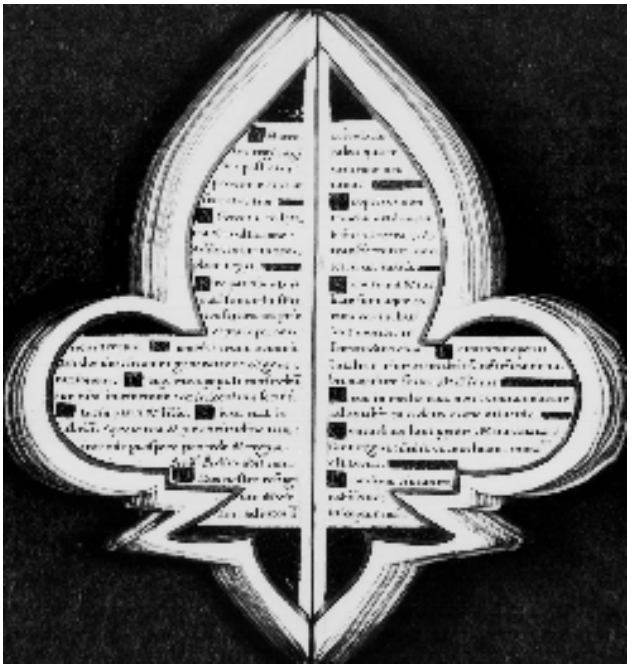
Les mots commençant par des majuscules n'ont pas été localisés à cause de la trop grande différence morphologique entre les lettres majuscules et minuscules dans ce type d'écriture.
Les résultats sont globalement très bons compte tenu de la résolution de l'image et de l'irrégularité de l'écriture.

4.2 Test *Escalopier* 22

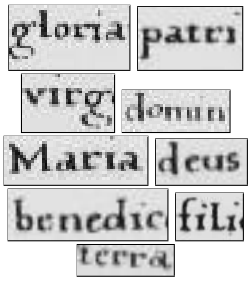
Nous avons ensuite extrait 24 pages (soit environ 2000 mots) dans le manuscrit *Escalopier 22* (voir figure 7.1) de la bibliothèque d'Amiens. Nous avons utilisé les mêmes critères que précédemment pour sélectionner les mots-clés et avons choisi : « benedic- », « deus », « domin- », « fili », « gloria », « Maria », « patri », « terra » et « virg- ». À cause des césures, des majuscules et des lettrines, notre système n'est sensé pouvoir localiser que 153 occurrences sur les 195 occurrences des neuf mots-clés, soit 78,5%.
Nous avons pourtant réussi à localiser 171 mots-clés (87,7%) soit 9,2 points de plus que le maximum théorique (voir table 2). La courbe P–R est donnée figure 7.3. Tous les mots-clés ont été localisés à des taux de précision et rappel équivalents.
Notons qu'un plus grand nombre de mots en majuscules et de mots césurés ont été localisés sur ce manuscrit que sur le précédent. Cela semble indiquer que le style d'écriture

	Total	Localisé	%age
Complet	153	151	98,7%
Lettrine	27	12	44,4%
Majuscule	3	3	Ø
Césure	12	5	41,7%
Total	195	171	87,7%

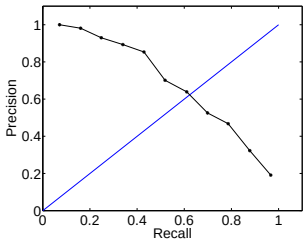
Tableau 2 – *Résultats sur* Escalopier 22.



1) *Une page du manuscrit* Escalopier 22.



2) *Les neuf prototypes.*



3) *La courbe P–R.* $P = R = 0.62$, $P(R = 1) = 0.19$.

Figure 7 – *Le test* Escalopier 22.

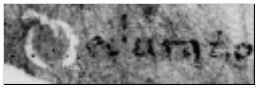
de ce manuscrit est plus discriminant. Cela montre de plus que cette écriture est plus régulière.
Cette fois encore les résultats sont très bons avec une précision relativement élevée pour de très hautes valeurs de rappel.

4.3 Test MS18

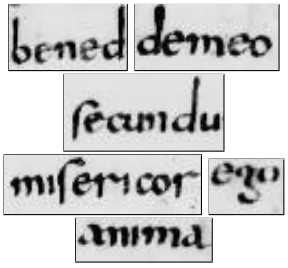
Nous avons lancé une autre série de tests sur le manuscrits MS18 de la bibliothèque d'Amiens. C'est un document fortement endommagé contenant de nombreuses taches et dont l'encre est souvent estompée.

Nous avons extrait 24 pages (soit environ 3100 mots) pour notre test. Malheureusement, ce document ne contient pas beaucoup de mots redondants et nous avons dû limiter notre test à la recherche de six mots (« demeo », « ego », « bened- », « secundu- », « misericor- », « anima »). Étant donné que même ces mots ne sont pas très fréquents, nous ne présenterons pas de courbe P–R car elle ne serait pas significative.

1) *Une page du MS18*



2) *Le microfilmage ne rend pas correctement les couleurs des lettrines. Celle-ci n'est visible que parce que le parchemin est sombre à cet endroit.*



3) *Les six prototypes.*

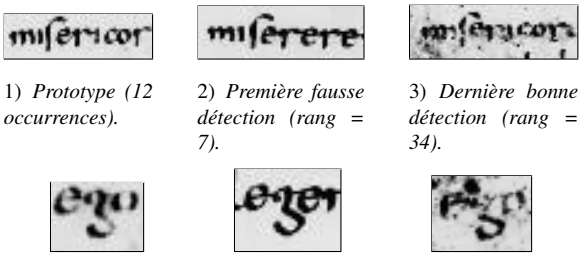
Figure 8 – *Le test MS18.*

Les résultats sont très bons étant donné l'état du document (voir table 3). Si le rappel est encore une fois très satisfaisant, notons que la précision est assez faible. Certaines occurrences des mots-clés sont si gravement endommagées qu'elles n'apparaissent que très loin dans la liste des réponses (voir figure 9).

	Total	Localisé	%age
Complet	49	48	98%
Lettrine	9	1	11,1%
Majuscule	0	0	Ø
Césure	2	0	Ø
Total	60	49	81,7%

Tableau 3 – *Résultats sur le MS18.*

Les mots commençant par une lettrine n'ont pas été locali-



1) *Prototype (12 occurrences).* 2) *Première fausse détection (rang = 7).* 3) *Dernière bonne détection (rang = 34).*

4) *Prototype (12 occurrences).* 5) *Première fausse détection (rang = 2).* 6) *Dernière bonne détection (rang = 181).*

Figure 9 – *Prototypes, premières fausses détections et dernières bonnes détections de deux mots-clés sur le MS18.*

sés. Dans ce document, les lettrines apparaissent en blanc (voir figure 8.2). Si le parchemin n'est pas taché, les lettrines sont invisibles, le cas échéant, elles sont blanches : aucun guide n'en est jamais extrait. Ainsi, lors de la comparaison du prototype, la première ZI de ce dernier est calée sur le second guide du mot sur l'image. L'élasticité de notre algorithme n'a pas été conçue pour surmonter un tel décalage car si les liens entre les ZI étaient trop lâches, l'algorithme sauterait des morceaux de mots et pourrait, par exemple, apparier « sale » avec « sa*b*le ».

5 Conclusion

Nous avons décrit une nouvelle méthode de localisation de mots adaptée aux manuscrits latins médiévaux. Cette méthode est particulièrement originale dans le sens où elle ne nécessite aucune segmentation ou binarisation du texte.
Nous avons choisi comme caractéristique l'orientation des gradients dans les zones où ces derniers ont une magnitude significative. Cette mesure est robuste aux faibles variations géométriques et rend compte de la structure locale des formes.
Nous avons proposé un algorithme de comparaison robuste aux irrégularités géométriques du texte manuscrit. Cet algorithme a trois principaux avantages : il est élastique et cohésif afin de gérer de grandes déformations des caractères et il est plus rapide que les méthodes de corrélation simples.
Nous avons testé notre méthode sur des manuscrits latins et avons obtenu de très bons résultats.

Des tests sont en cours sur d'autres types de documents, notamment des documents imprimés très dégradés (binarisés avec tramage), des manuscrits arabes, hébreux, tibétains et japonais.
Nous avons prévu d'améliorer l'extraction des guides afin de la rendre indépendante du type des documents traités.
Une méthode de raffinement des résultats sera mise au point afin de trier la liste des réponses de manière plus fine.
Nous réfléchissons par ailleurs à un moyen de rendre notre algorithme de comparaison robuste aux homothéties afin de régler le problème de la normalisation.

Références

[1] R. Manmatha, C. Han, et E.M. Riseman. Word spotting : A new approach to indexing handwriting. Dans *CVPR*, pages 631–637, San Francisco, Etats-Unis, 1996.

[2] P. Keaton, H. Greenspan, et R. Goodman. Keyword spotting for cursive document retrieval. Dans *Workshop on Document Image Analysis*, pages 74–81, San Juan, Puerto Rico, 1997.

[3] A. Kołcz, J. Alspector, M. Augusteijn, R. Carlson, et G. Viorel Popescu. A line-oriented approach to word spotting in handwritten documents. *PAA*, 3 :153–168, 2000.

[4] Y. Lu et C.L. Tan. Word spotting in chinese document images without layout analysis. Dans *ICPR*, pages 57–60, Quebec, Canada, 2002.

[5] A. Derolez. *The Paleography of Gothic Manuscript Books*. Cambridge Univerity Press, Cambridge, Grande-Bretagne, 2003.

[6] J. Edwards, Y.W. Teh, D. Forsyth, R. Bock, et M. Maire. Making latin manuscripts searchanle using ghmm's. Dans *Neural Information Processing Systems*, pages 385–392, Cambridge, Etats-Unis, 2004.

[7] R. Manmatha et J.L. Rothfeder. A scale space approach for automatically segmentting word from historical handwritten documents. *IEEE TPAMI*, 27(8) :1212–1225, 2005.

[8] T.M. Rath et R. Manmatha. Features for word spotting in historical manuscripts. Dans *ICDAR*, volume 1, pages 218–222, Edinbourg, Ecosse, 2003.

[9] T.M. Rath et R. Manmatha. Word image matching using dynamic time warping. Dans *IEEE Computer Vision and Pattern Recognition*, pages 521–527, Madison, Etats-Unis, 2003.

[10] J.L. Rothfeder, S. Feng, et T.M. Rath. Using corner feature correspondences to rank word images by similarity. Dans *Conference on Computer Vision and Pattern Recognition Workshop*, pages 30–35, Madison, Etats-Unis, 2003.

[11] R. Manmatha, C. Han, E.M. Riseman, et W.B. Croft. Indexing handwriting using word matching. Dans *ACM First Intl. Conf. on Digital Libraries*, pages 151–159, Bethesda, Etats-Unis, 1996.

[12] T.M. Rath, R. Manmatha, et V. Lavrenko. A search engine for historical manuscript images. Dans *ACM SIGIR conference on Research and development in information retrieval*, pages 369–376, Sheffield, Royaumes-Unis, 2004.

[13] N.R. Howe, T.M. Rath, et R. Manmatha. Boosted decision trees for word recognition in handwritten document retrieval. Dans *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 377–383, Salvador, Brésil, 2005.

Affichages distribués et usage collaboratif

Maxime Collomb Mountaz Hascoët

LIRMM, UMR 5506 du CNRS. 161 rue Ada, 34 392 Montpellier Cedex 5

{collomb, mountaz}@lirmm.fr

Résumé

Cet article propose une vue d'ensemble des systèmes permettant de gérer des environnements d'affichage distribués. La distribution de l'affichage peut se manifester par l'utilisation de plusieurs affichages mais aussi par le partage d'un affichage entre plusieurs utilisateurs. Dans un premier temps, la notation UDP/C est utilisée. Nous proposons ensuite de la compléter en ajoutant les notions de réplication et de mixage pour mieux expliciter points communs et différences entre les systèmes présentés.

Mots clefs

multi écrans, multi pointeurs, collecticiels, distributed display environment.

1 Introduction

Il est de plus en plus courant pour un utilisateur de disposer de plusieurs surfaces d'affichage. En effet, les cartes graphiques actuelles disposent très souvent de plusieurs sorties et il n'est pas rare de réutiliser un écran provenant d'un ancien ordinateur pour se constituer une configuration disposant d'un affichage distribué.

Il n'est pas rare non plus de disposer de plusieurs machines, le cas le plus courant étant un couple ordinateur portable / ordinateur de bureau.

Enfin, une autre situation courante d'affichage distribué est l'utilisation d'un vidéo projecteur dans une salle de réunion.

Cependant, les différentes configurations multi affichages dont on peut disposer amènent des problématiques différentes. En effet, les problèmes techniques sont différents si l'on veut utiliser un écran à plusieurs ou tout seul ou encore si deux écrans sont reliés à une ou deux unités centrales.

Nous allons présenter dans cet article un ensemble représentatif de systèmes mettant en œuvre des environnements d'affichage distribués. L'utilisation de la majeure partie de ces systèmes se traduit par le transfert d'évènements d'entrée ou de flux d'affichage via un réseau (LAN ou WAN). Nous nous intéresserons tout particulièrement à la caractérisation de ces systèmes afin de mettre en évidence les propriétés importantes qui les différencient ou les rapprochent.

2 Notation UDP/C

La notation **UDP/C** introduite par Lecolinet [1] utilise les abréviations suivantes : **U** pour utilisateur, **D** pour dispositif d'affichage (ou *display*), **P** pour pointeur et **C** pour unité centrale (ou *computer*).

La notion de dispositif d'affichage nécessite une clarification : dans cette notation, un dispositif d'affichage est assimilé à une carte graphique plutôt qu'à un écran. En effet, le fait qu'il y ait un ou plusieurs écrans en sortie d'une même carte graphique ne fait quasiment aucune différence. La carte graphique se contente de séparer les signaux vidéo et il n'y a pas de changement notable dans l'interface utilisateur.

Il convient également de noter que la notion de pointeur est liée à **D**. Il faut parler de pointeur par dispositif d'affichage. En effet, un pointeur est généralement lié à un écran (et plus particulièrement lorsque l'utilisateur dispose d'un dispositif de pointage direct comme un écran tactile).

Précisons finalement que seule la notion de pointeur est utilisée. Celle-ci renvoie à l'utilisation d'un système de pointage (souris, surface tactile) mais pas à un dispositif de saisie (clavier). On suppose en fait que les 2 sont couplés et que le clavier "suit" la souris.

La notation **UDP/C** a été initialement utilisée pour la classification des systèmes multi pointeurs mais nous l'utiliserons ici en nous intéressant également aux capacités multi affichages d'un ensemble de systèmes.

3 Les systèmes personnels (1-*-*/*)

3.1 La configuration courante

Le cas classique d'un ordinateur disposant d'1 écran et d'1 souris utilisés par 1 utilisateur correspond à la notation 1-1-1/1. Précisons que cette notation couvre également le cas d'un double affichage (deux écrans connectés sur la même carte graphique).

3.2 Les bureaux à distance

Une variante intéressante est l'utilisation de systèmes qui permettent d'interagir avec des applications qui ne s'exécutent pas sur l'unité centrale C qui contrôle D et P mais sur d'autres machines reliées à C via un réseau. On peut par exemple citer le serveur X-window utilisé dans le monde UNIX, VNC (*Virtual Network Computing*) [2] (figure 1) ou la "connexion au bureau à distance" de MS Windows.

Il est également intéressant de citer ici l'architecture INDIGO [3] qui s'appuie sur un ensemble de standards (XML, XSLT, SVG, SOAP, etc.) et qui propose la réalisation d'interfaces distribuées au travers de deux types de composants :

Figure 1: *Un bureau MS Windows contenant un bureau MacOS via VNC. MacOS ne s'exécute pas sur la même machine que MS Windows, mais l'affichage du bureau MacOS est reproduit dans une fenêtre de MS Windows.*

des serveurs *d'objets* et des serveurs *d'interaction et de rendu*. Un serveur d'objets est en fait le noyau fonctionnel d'une application qui expose ses données à un ou plusieurs serveurs d'interaction et de rendu. Les serveurs partagent un modèle conceptuel que le serveur d'interaction et de rendu transforme en un modèle perceptuel. Le serveur d'interaction et de rendu contrôle donc l'interface utilisateur et peut interpréter les interactions de l'utilisateur sur le modèle perceptuel. Ces interactions donnent lieu à des commandes à appliquer au modèle conceptuel qui sont transmises au serveur d'objets.

3.3 Les pointeurs mobiles

Des systèmes comme Synergy [4] ou PointRight [5] permettent d'utiliser un même dispositif de pointage sur plusieurs machines. Les unités centrales doivent être reliées via un réseau, et lorsque le pointeur atteint le bord d'un des écrans, le curseur "saute" sur un autre écran connecté à une autre unité centrale. Ces systèmes correspondent à la notation 1-N-1/N, c'est à dire que 1 utilisateur utilise 1 dispositif de pointage qui peut se déplacer sur N écrans (connectés sur N machines).
De tels systèmes permettent par exemple de contrôler un ordinateur portable et un ordinateur de bureau simplement en utilisant la souris et le clavier de l'ordinateur de bureau. Lorsque le pointeur atteint un bord de l'écran de l'ordinateur de bureau, il ne s'arrête pas mais continue son mouvement et entre sur l'écran du portable. Tout se passe comme si les écrans de l'ordinateur portable et de l'ordinateur de bureau formaient un espace continu. Cependant, l'illusion a ses limites : il n'est pas possible de faire un *glisser-déposer* pour déplacer une fenêtre ou une icône. Il n'y a pas d'interaction entre les deux espaces de travail (bien que ces Synergy et PointRight gèrent un presse papier partagé qui permet de copier des données sur une machine et de les coller sur une autre).

4 Les logiciels collaboratifs (N-*-*/*)

4.1 Single display groupware (logiciel collaboratif sur un seul écran)

Dans le cadre de cet article, les *Single display groupware* ont la particularité de fonctionner sur une machine unique et de ne pas nécessiter de transferts réseau. Ces systèmes sont cependant très intéressants car ils permettent la manipulation de plusieurs pointeurs sur un même espace de travail
Dans cette catégorie de systèmes, plusieurs utilisateurs interagissent avec le même dispositif d'affichage mais en utilisant plusieurs dispositifs d'interaction. Ces systèmes sont décrits par la notation N-1-N/1. Un exemple de *Single display groupware* est DiamondSpin [6] (figure 2).

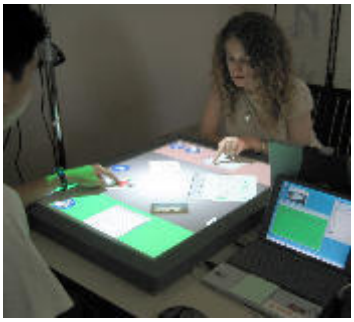


Figure 2: *Deux utilisateurs interagissant simultanément avec une table interactive et le logiciel DiamiondSpin [6].*

Il faut noter que l'utilisation de plusieurs pointeurs simultanément introduit des complications importantes au niveau des interfaces. En effet, que faire si deux personnes naviguent en même temps dans le menu d'une application par exemple ?

4.2 Logiciels collaboratifs synchrones

Cette catégorie de logiciels regroupe les logiciels permettant une collaboration entre des utilisateurs n'étant pas à proximité immédiate les uns des autres. Chaque utilisateur dispose donc de sa propre machine, toutes les unités centrales étant reliées via un réseau. Chaque utilisateur dispose d'une vue sur un espace de travail commun et peut voir les actions des autres utilisateurs.
Lecolinet différencie ici les pointeurs actifs et passifs. En fait, sur un dispositif d'affichage donné, on est susceptible de voir N pointeurs. Un seul de ces pointeurs (celui qui est lié au dispositif d'affichage) sera actif alors que les N-1 autres seront passifs. Il en découle la notation N-N-(1, N-1)/N. Sur un dispositif d'affichage donné, les pointeurs passifs permettent à l'utilisateur du pointeur actif de savoir ce que font les autres et ainsi d'anticiper les conflits pour mieux les éviter.

5 Les interfaces distribuées (*-N-*/N

Les systèmes présentés jusqu'ici présentent encore des limites. En effet, il est possible d'avoir plusieurs curseurs simultanément sur un même écran, il est possible grâce à une connexion réseau d'utiliser une souris et un clavier sur une autre machine que celle où ils sont connectés. Il est également possible de voir sur son écran personnel des programmes qui s'exécutent sur d'autres machines. Cependant, il serait intéressant de pouvoir faire toutes ces choses dans un même système.

Deux systèmes, encore au stade de recherche, permettent d'intégrer tous ces comportements. Le premier système est UBIT [1] (grâce à l'*Ubit Mouse Server*) qui permet de contrôler plusieurs pointeurs, les différents dispositifs de pointage n'étant pas obligatoirement connectés à la même machine. De plus, l'Ubit Mouse Server permet le contrôle d'une application depuis une machine distante ou encore la migration d'une application d'un affichage à un autre.

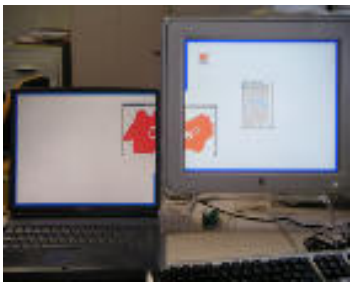


Figure 3: *Avec le système I-AM, la fenêtre d'une application s'exécutant sur une machine A peut être affichée en partie sur le dispositif d'affichage de la machine A et en partie sur le dispositif d'affichage d'une machine B, les machines A et B étant reliées via un réseau. Il est bien sur également possible d'afficher cette fenêtre uniquement sur un des deux dispositifs d'affichage [7].*

Le second système est I-AM [7] qui permet lui aussi de contrôler plusieurs dispositifs de pointage agissant sur plusieurs dispositifs d'affichage. Il est également possible de transférer l'affichage d'une application sur un dispositif d'affichage distant (figure 3).

6 Les notions clés : Réplication et mixage

Tous les systèmes présentés permettent à des niveaux différents de répliquer (ou parfois simplement de rediriger) et/ou de mixer les dispositifs de pointages et/ou les dispositifs d'affichage.

6.1 Les dispositifs de pointage

Réplication / redirection. La réplication au niveau des dispositifs de pointage consiste à capturer les mouvements d'un pointeur sur une machine, à les transmettre à une autre machine à travers un réseau, puis à reproduire ces mouvements sur cette autre machine.

Mixage. Le mixage au niveau des dispositifs de pointage consiste à la gestion simultanée de plusieurs pointeurs sur une même surface d'affichage. Ceci introduit d'importants problèmes car il est pratiquement impossible d'utiliser les interfaces actuelles en utilisant plusieurs pointeurs (il est par exemple impossible de dessiner plusieurs formes simultanément dans les logiciels de dessin courants).
En effet, l'utilisation simultanée de plusieurs pointeurs remet en cause les systèmes de fenêtrage actuels qui résident sur la notion d'un focus unique (un seul composant de l'interface est actif à un moment donné).

6.2 Les dispositifs d'affichage

Réplication / redirection. Au niveau des dispositifs d'affichage, la réplication consiste à afficher l'interface d'un programme sur un écran qui n'est pas directement connecté à l'unité centrale qui contrôle l'exécution du programme.

Mixage. Le mixage des dispositifs d'affichage se traduit par la possibilité d'intégrer les interfaces. Intégrer une interface permet de déplacer l'affichage d'un programme d'un dispositif d'affichage à un autre sans que cela n'ait de conséquence sur l'exécution du programme.

6.3 La réplication et le mixage dans les systèmes présentés

Le tableau 1 montre les capacités des différents systèmes présentés dans cet article pour gérer la réplication et le mixage au niveau des dispositifs de pointage et d'affichage.

	Pointage		Affichage		Multi
	Répli-cation	Mix-age	Répli cation	Mix-age	utili-sateurs
Double-écrans					
X-Window, VNC, Indigo	✓		✓		
Synergy, PointRight	✓				
SDG, DiamondSpin		✓			✓
log. collab. synchrones	✓	✓	✓		✓
UBIT I-AM	✓	✓	✓	✓	

Tableau 1: *Gestion de la réplication et du mixage dans les systèmes présentés*

On remarque que les systèmes les plus avancés sont UBIT et I-AM. Cependant, ces derniers présentent encore une limitation : ils ne permettent pas une utilisation collaborative.

6.4 Granularité

Cependant, définir les notions de réplication et mixage pour caractériser les environnements d'affichage distribués n'est pas suffisant. En effet, les données échangées lors de la réplication, que ce soit au niveau des dispositifs de pointage ou d'affichage, peuvent être de natures très différentes.
Par exemple, prenons le cas d'une application affichant une scène 3D qui s'exécute sur une machine A et s'affiche sur une machine B (figure 4). On peut distinguer deux cas extrêmes au niveau des données échangées entre les machines A et B pour la distribution de l'affichage. Une première possibilité consiste à effectuer le maximum de calculs sur la machine A contrôlant l'exécution du programme. La machine B recevant les données à afficher sous la forme d'un champ de pixels. La machine B ne manipulant qu'un champ de pixels, elle ne sait pas où se trouvent les objets de l'interface (les boutons, par exemple). De ce fait, la machine B ne peut pas traiter les évènements issus des dispositifs de pointage et les transmet donc sans interprétation à la machine A. Les transmissions de ce type ne requièrent pas une

machine puissante du coté des dispositifs d'affichage et de pointage (machine B) mais sont exigeantes en ressources réseaux (débit important et temps de latence faible).
La seconde possibilité consiste à un échange de données de plus haut niveau entre les machines A et B. La machine A transmet à la machine B une description de la scène 3D. Et c'est la machine B qui effectue le rendu de la scène. Dans ces conditions, la machine B est en mesure d'interpréter certains évènements issus des dispositifs de pointage sans en référer à la machine A. Par exemple, pour naviguer dans la scène 3D (zoom, panoramique, etc.), la machine B peut effectuer elle même le déplacement de la caméra. Il peut cependant être nécessaire de notifier la machine A de ce déplacement de caméra. Apparaissent alors les questions de synchronisation entre les données de la machine A et de la machine B, questions récurrentes dans le cadre des environnements d'affichage distribués.

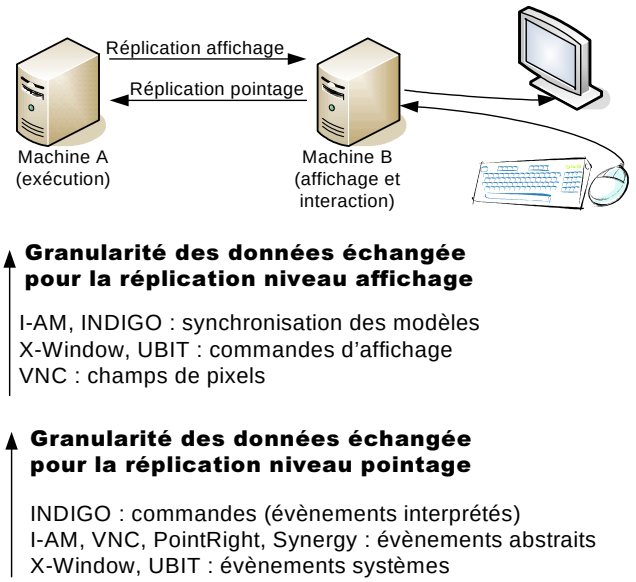


Figure 4: *Différences de granularité de la réplication au niveau de l'affichage et du pointage pour les systèmes présentés.*

La figure 4 présente les différentes granularités des systèmes présentés précédemment dans cet article, au niveau de l'affichage est du pointage. Ainsi, au niveau de l'affichage, le protocole de VNC transmet des informations de très bas niveau puisqu'il s'agit de pixels. X-Window et UBIT (qui est basé sur X-Window) transmettent quand à eux des commandes d'affichage (exemple : le tracé d'un rectangle se traduit par le transfert de la commande *XDrawRectangle*, commande qui est exécutée par la machine B). Et enfin, I-AM et Indigo procèdent à une synchronisation des modèles d'interface entre les différentes machines. Par conséquent, la machine B contrôle les éléments de l'interface et leur affichage.
Au niveau des évènements de pointage, le client et le serveur X-Window s'échangent des évènements systèmes de bas niveau (tous du même type XEvent), tandis que des systèmes comme I-AM, VNC, PointRight ou Synergy procèdent à une abstraction des évènements avant de procéder à leur réplication. Ceci est du à la nature multi plateforme de ces systèmes (des évènements issus d'un système MS Windows nécessitent une interprétation pour être répliqués sur un système MacOS, par exemple). Et enfin, Indigo gère l'interaction du coté de la machine B et seules les commandes interprétées sont transmises à la machine A.

7 Conclusion

On dispose de plus en plus souvent de plusieurs affichages dans notre environnement de travail. Certains systèmes simplifient l'utilisation de ces affichages simultanément. Il existe également des systèmes permettant d'utiliser une surface d'affichage de manière collaborative.
Cependant, les environnements d'affichage distribués permettant une utilisation collaborative sont confrontés au frein que sont les interfaces graphiques logicielles. En effet, dans l'état actuel des choses, elles représentent un verrou au développement d'applications collaboratives car elles ne permettent que très rarement d'être utilisées par plusieurs utilisateurs simultanément.
Les environnements d'affichages distribués et le travail collaboratif font souvent l'objet de domaines de recherche dissociés. Nous avons présenté dans cet article un ensemble représentatif de systèmes se rapportant à ces deux domaines et nous avons proposé des propriétés qualitatives permettant de caractériser ces systèmes. Nous espérons ainsi clarifier des domaines qui sont de plus vastes.

Références

[1] Eric Lecolinet. Multiple pointers : a study and an implementation. Dans *IHM 2003*, pages 134–141, New York, NY, USA, 2003. ACM Press.

[2] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, et Andy Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1) :33–38, 1998.

[3] Renaud Blanch, Michel Beaudouin-Lafon, Stéphane Conversy, Yannick Jestin, Thomas Baudel, et Yun Peng Zhao. Indigo : une architecture pour la conception d'applications graphiques interactives distribuées. Dans *IHM'05*. ACM Press, 2005.

[4] Chris Schoeneman. Synergy. http ://synergy2.sourceforge.net/.

[5] Brad Johanson, Greg Hutchins, Terry Winograd, et Maureen Stone. Pointright : experience with flexible input redirection in interactive workspaces. Dans *UIST '02*, pages 227–234, New York, NY, USA, 2002. ACM Press.

[6] Chia Shen, Frédéric D. Vernier, Clifton Forlines, et Meredith Ringel. Diamondspin : an extensible toolkit for around-the-table interaction. Dans *CHI '04*, pages 167–174, New York, NY, USA, 2004. ACM Press.

[7] Christophe Lachenal. *Modèle et infrastructure logicielle pour l'interaction multi-instrument multisurface*. Thèse de doctorat, 2004.

Une nouvelle méthode de description et d'indexation des grandes bases de formes

H. SILKAN[1], S. E. OUATIK[1], A. LACHKAR[2], M. MEKNASSI[1]
[1]LISQ, Faculté des Sciences Dhar EL Mahraz, Fès
[2]E.S.T.M, Université Moulay Ismail, Meknès, lachkar@est-umi.ac.ma,
Silkan_h@hotmail.com, souatik@fsdmfes.ac.ma, m_meknassi@yahoo.com

Résumé

Dans ce papier, nous proposons une nouvelle méthode de description et d'indexation des grandes bases de formes. Elle apporte deux contributions. La première consiste à calculer un nouveau descripteur de forme invariant par rapport à certaines transformations géométriques, notamment la rotation. Ce dernier est une version multi-échelle du descripteur proposé par Berretti et al. dont le principal inconvénient est la non invariance par rapport à la rotation. Dans la deuxième contribution, nous proposons de stocker l'ensemble des index de toutes les formes de la base dans une seule structure d'index appelée M-tree, cette structure en arbre est liée à toutes les formes de la base et non pas à chaque objet de celle-ci. Les résultats obtenus par l'application de notre approche sur une grande base de formes montrent l'intérêt de celle-ci. Nous montrerons aussi que notre système d'indexation et de recherche est plus performant en temps de réponse que les autres systèmes basés sur le parcours séquentiel.

Mots clefs

Descripteur invariant, Indexation, M-tree, Recherche par similarité, Bases de formes.

1 Introduction

La quantité d'images produites et archivées chaque jour ne cesse d'augmenter. Pour pouvoir exploiter ces archives colossales ainsi créées, il faut être capable d'indexer ce qui est stocké, et le retrouver. Ceci nécessite, d'une part, le choix d'une représentation pertinente des images de la base par le biais de primitives visuelles significatives et fiables qui traduisent le contenu de la base. D'autre part, une structure d'index efficace est nécessaire pour optimiser le temps de la recherche.

Généralement, les critères employés pour décrire une image sont des descripteurs de bas-niveau, appelés aussi *vecteurs caractéristiques*, tels que la couleur [1,2] et la texture [3,4]. La forme [5,6] tient une place à part parmi ces descripteurs dans la mesure où elle apporte une information moins ambiguë sur l'objet, car elle est plus proche de sa signification.

Différents descripteurs de formes existent dans la littérature [7]. Récemment, Berretti et al. [8] ont proposé un nouveau descripteur. Les formes sont partitionnées en des portions appelées *tokens*. Chaque token est représenté par deux attributs qui sont la *courbure* et *l'orientation*.

Comme il est mentionné par Zhang et al. [9], ce descripteur est invariant par rapport à la translation et l'homothétie, mais il présente l'inconvénient de ne pas être invariant par rapport à la rotation. Pour pallier à ce problème, nous proposons de réaliser une rotation de tous les vecteurs d'orientations des tokens suivant un angle calculé en utilisant le principe d'équilibre des vecteurs de forces. De plus, les deux attributs courbure et orientation sont calculés à plusieurs valeurs d'échelles. Il s'agit alors d'une version multi-échelle du descripteur de Berretti et al. [8], ayant l'avantage d'être invariant par rapport à la rotation.

Après l'étape de l'extraction des descripteurs la question qui se pose est de déterminer comment retrouver les images les plus similaires à une image requête. Plusieurs travaux ont été proposés se basant principalement sur deux étapes : (i) organiser les vecteurs caractéristiques dans des structures de données adaptées, puis (ii) trouver les algorithmes de parcours efficaces pour améliorer le temps de la recherche. Les méthodes d'accès spatial SAM (Spatial Access Methods) comme R-Tree [10] et ses variantes [11,12] ont été utilisées pour répondre aux requête de similarité. Il s'agit de retrouver les objets de la base les plus similaires à un objet requête. Ainsi, ce processus nécessite une fonction de distance pour mesurer la similarité entre les vecteurs caractéristiques. Cependant, l'application des méthodes SAM a relevé deux limitations (i) le nombre de composantes des vecteurs caractéristiques doit être réduit, (ii) la fonction de distance doit être de type L_p comme la distance euclidienne L_2 et Manhattan. Berretti et al. [8] ont utilisé l'arbre métrique M-tree [13]. A chaque forme est associé un arbre. Deux mesures ont été définies, une pour calculer la similarité entre les tokens et une autre non métrique pour les formes. D'autres méthodes d'indexation ont été proposées. Mahmoudi et Daoudi [14] proposent de caractériser les objets 3D par des vues caractéristiques. Ils ont utilisé des index calculés à partir des vues et basés sur le CSS (Curvature Scale Space). Ces index sont organisés autour d'une structure d'arbre M-tree. De même, ils ont associé un arbre M-tree à chaque vue de la base.

Pour une base de N formes, les deux derniers travaux cités précédemment produisent N arbres M-tree. Ainsi, la recherche est rapide au niveau de chaque forme, par contre elle est séquentielle sur la base. Pour éviter cet inconvénient, nous proposons d'utiliser l'arbre M-tree pour la totalité de la base et non pas pour chaque forme.

Le reste de cet article est organisé de la façon suivante. La section 2 présente le descripteur de forme proposé. Dans la section 3, nous décrivons la technique d'indexation adoptée. Les résultats expérimentaux sont présentés en section 4. Dans la section 5, nous donnons une conclusion et quelques perspectives de ce travail.

2 Descripteur proposé

Nous proposons un descripteur invariant par rapport à la rotation. Ce dernier est une version multi-echelle du descripteur proposé par Berretti et al. [8]. Chaque token est représenté par deux attributs : la courbure et l'orientation.

Soit $f(u) = \langle (x(u), y(u)) | u \in [0, T] \rangle$ la représentation paramétrique de la courbe correspondant à la forme, où u est l'abscisse curviligne et T est la longueur de la courbe. La famille des courbes $\langle f(u,\sigma)/\sigma \geq 0 \rangle$ obtenues par la convolution de f(u) avec la gaussienne $g(u,\sigma)$ définie par : $g(u,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-u^2}{2\sigma^2}}$ engendre l'espace d'échelle euclidienne. Soit aussi $K(u,\sigma)$ la courbure multi-echelle correspondant à la famille des courbes $\langle f(u,\sigma)/\sigma \geq 0 \rangle$.

$$K(u,\sigma) = \frac{x_t(u,\sigma) y_{tt}(u,\sigma) - x_{tt}(u,\sigma) y_t(u,\sigma)}{\left(x_t^2(u,\sigma) + y_t^2(u,\sigma)\right)^{3/2}}$$

Tel que x_t, y_t et x_{tt} ,y_{tt} sont respectivement les dérivées premières et secondes de x et y. Posons $P = \{P_{i(\sigma)}\}_{i=1}^{N}$ l'ensemble de points tels que $K(u,\sigma)=0$. Si $K(u,\sigma)$ est continue entre deux points consécutifs $P_{i(\sigma)}$ et $P_{(i+1)(\sigma)}$, il existe toujours entre eux un extremum de $K(u,\sigma)$ nommé $m_{i(\sigma)}$ au point $P_{mi(\sigma)}$. Pour chaque valeur d'échelle σ on obtient une forme lissée qu'on partitionne en un ensemble de tokens, selon les points P_i. Chaque token i de $f(u,\sigma)$ est représenté par le vecteur $E_{i(\sigma)}(m_{i(\sigma)}, O_{i(\sigma)})$ (figure-1), où $m_{i(\sigma)}$ est la courbure de $P_{mi(\sigma)}$ comprise entre -180 et 180, et $O_{i(\sigma)}$ l'orientation du token en coordonnées polaires variant entre 0 et 360°.

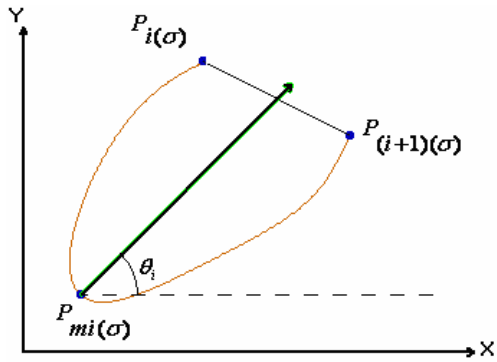


Figure 1 – *Représentation d'un token : l'angle θ_i mesure l'orientation du token.*

Pour chaque valeur de lissage σ d'une forme, on obtient l'ensemble des vecteurs :

$\Gamma_\sigma = \left\langle \vec{E}_{i(\sigma)}(m_{i(\sigma)}, O_{i(\sigma)}) \middle| 1 \leq i \leq n_\sigma \right\rangle$, Où n_σ est le nombre de tokens de la forme lissée notée F_σ. Finalement le vecteur caractéristique obtenu pour une forme de la base est donné par :

$$\Gamma = \bigvee_{j=1}^{M} \Gamma_{\sigma j} = \bigvee_{j=1}^{M} \left\langle \vec{E}_{i(\sigma j)}(m_{i(\sigma j)}, O_{i(\sigma j)}) \middle| 1 \leq i \leq n_{\sigma j} \right\rangle = \left\langle \vec{E}_i(m_i, O_i) \middle| 1 \leq i \leq n \right\rangle$$

Où M est le nombre de valeurs de lissage, et $n = \sum_{j=1}^{M} n_{\sigma j}$ est le nombre total des vecteurs.

Notons que la courbure et l'orientation d'un token sont invariantes par rapport à la translation et à l'homothétie. Et par conséquent, notre descripteur sera également invariant par rapport à ces transformations. Le problème qui reste est celui d'invariance par rapport à la rotation. Prenons un exemple de deux orientations différentes d'un même token (figure-2).

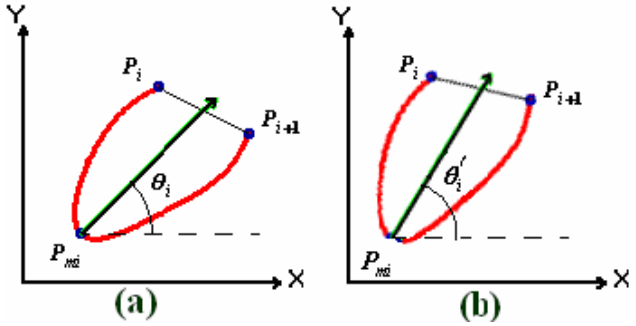


Figure 2 - *Un token representé dans deux orientations différentes.*

Soient $E_1(m_i,\theta_i)$ et $E_2(m'_i,\theta'_i)$ les descripteurs de token avant et après rotation (figure-2), $m_i = m'_i$ et $\theta_i \neq \theta'_i$, donc le descripteur considéré n'est pas invariant par rapport à la rotation.

Pour résoudre ce problème nous proposons la solution suivante : Soit $\vec{a}_0$ et $\vec{b}_0$ deux vecteurs non nuls. D'après le principe d'équilibre des vecteurs de forces, il existe un vecteur $\vec{c}_0$ tel que $\vec{a}_0 + \vec{b}_0 + \vec{c}_0 = \vec{0}$. Le vecteur $\vec{c}_0$ qui vérifie l'équilibre est appelé *vecteur principal* et sa direction par rapport à l'axe (OX) est appelée *direction principale*. Supposons que $\vec{a}_\theta$ et $\vec{b}_\theta$ sont les deux vecteurs obtenus après la rotation de $\vec{a}_0$ et $\vec{b}_0$ par un angle θ. Il existe un vecteur principal $\vec{c}_\theta$ qui vérifie $\vec{a}_\theta + \vec{b}_\theta + \vec{c}_\theta = \vec{0}$. La direction principale θ de $\vec{c}_\theta$ est déterminée par l'équation : $\theta = \arccos\left(\frac{\vec{c}_0 \cdot \vec{c}_\theta}{|\vec{c}_0| \cdot |\vec{c}_\theta|}\right)$

Ce principe peut être généralisé à plusieurs vecteurs. En effet, soit $R(\theta)$ une transformation de rotation par un

angle θ d'un vecteur donné. Pour un ensemble de vecteurs $\Lambda_0 = \{\vec{f}_i \mid 1 \le i \le N\}$, où N est le nombre de vecteurs et $\Lambda_\theta = \{\vec{g}_i \mid \vec{g}_i = R(\theta).\vec{f}_i \ , \ 1 \le i \le N\}$ sa rotation par un angle θ. Alors, si $\sum_{i=1}^{N} \vec{f}_i \neq \vec{0}$, il existe un vecteur principal $\vec{F}_0$ vérifiant $\sum_{i=1}^{N} \vec{f}_i + \vec{F}_0 = \vec{0}$. De même, si $\sum_{i=1}^{N} \vec{g}_i \neq \vec{0}$ il existe un vecteur principal $\vec{F}_\theta$ vérifiant l'équation (1)

$$\sum_{i=1}^{N} \vec{g}_i + \vec{F}_\theta = \vec{0} \qquad (1).$$

La direction principale θ de $\vec{F}_\theta$ est donc déterminée par l'équation (2) : $\theta = \arccos\left(\frac{\vec{F}_0 \cdot \vec{F}_\theta}{|\vec{F}_0| \cdot |\vec{F}_\theta|}\right)$ (2)

A partir de l'équation donnée en (1), on déduit :

$$R(-\theta) \cdot \left[\sum_{i=1}^{N} \vec{g}_i + \vec{F}_\theta\right] = \sum_{i=1}^{N} [R(-\theta) \cdot \vec{g}_i] + R(-\theta) \cdot \vec{F}_\theta = \vec{0}$$

Or, $R(-\theta) \cdot \vec{F}_\theta = \vec{F}_0$ *et* $\vec{F}_0 = -\sum_{i=1}^{N} \vec{f}_i$

Donc $\sum_{i=1}^{N} R(-\theta) \cdot \vec{g}_i = \sum_{i=1}^{N} \vec{f}_i$ (3)

D'après la formule (3), on déduit que la direction initiale de l'ensemble $\Lambda_0 = \{\vec{f}_i \mid 1 \le i \le N\}$ peut être déduite à partir de la rotation des vecteurs $\sum_{i=1}^{N} \vec{g}_i$ et $\vec{F}_\theta$ par l'angle $-\theta$. Cela peut être réalisé par l'application des deux formules suivantes :

$$R(-\theta) \cdot \vec{\Lambda}_\theta = \vec{\Lambda}_0 \qquad (4)$$

$$R(-\theta) \cdot \vec{F}_\theta = \vec{F}_0 \qquad (5)$$

Pour chaque forme de la base, nous proposons de calculer sa direction principale θ, ensuite réaliser une rotation de tous les vecteurs d'orientations des tokens par $-\theta$ de telle sorte que le vecteur principal des vecteurs caractéristiques de chaque forme de la base coïncide avec l'axe (OX). Ceci va permettre de rendre notre descripteur invariant par rapport à la rotation. En effet, pour deux descripteurs $\Lambda_{\theta 1}$ et $\Lambda_{\theta 2}$ d'une même forme avec deux orientations différentes, on calcule d'abord leurs vecteurs principaux respectivement $F_{\theta 1}$ $F_{\theta 2}$. Ensuite, on détermine leurs directions principales θ_1 et θ_2. Puis, on procède à une rotation de l'ensemble des vecteurs $\Lambda_{\theta 1}$ et $\Lambda_{\theta 2}$ respectivement par $-\theta_1$ et $-\theta_2$ en appliquant (4) et (5). On obtient alors pour $\Lambda_{\theta 1}$ et $\Lambda_{\theta 2}$ le même vecteur caractéristique Λ_0 ayant comme vecteur principal F_0, vie celui qui vérifie l'équation de l'équilibre et qui coïncide toujours avec l'axe (OX). La figure 3 présente le résultat d'un exemple produit par notre programme appliqué à une forme dans quatre orientations différentes. On remarque que pour les quatre orientations de la forme, on obtient le même descripteur (2d).

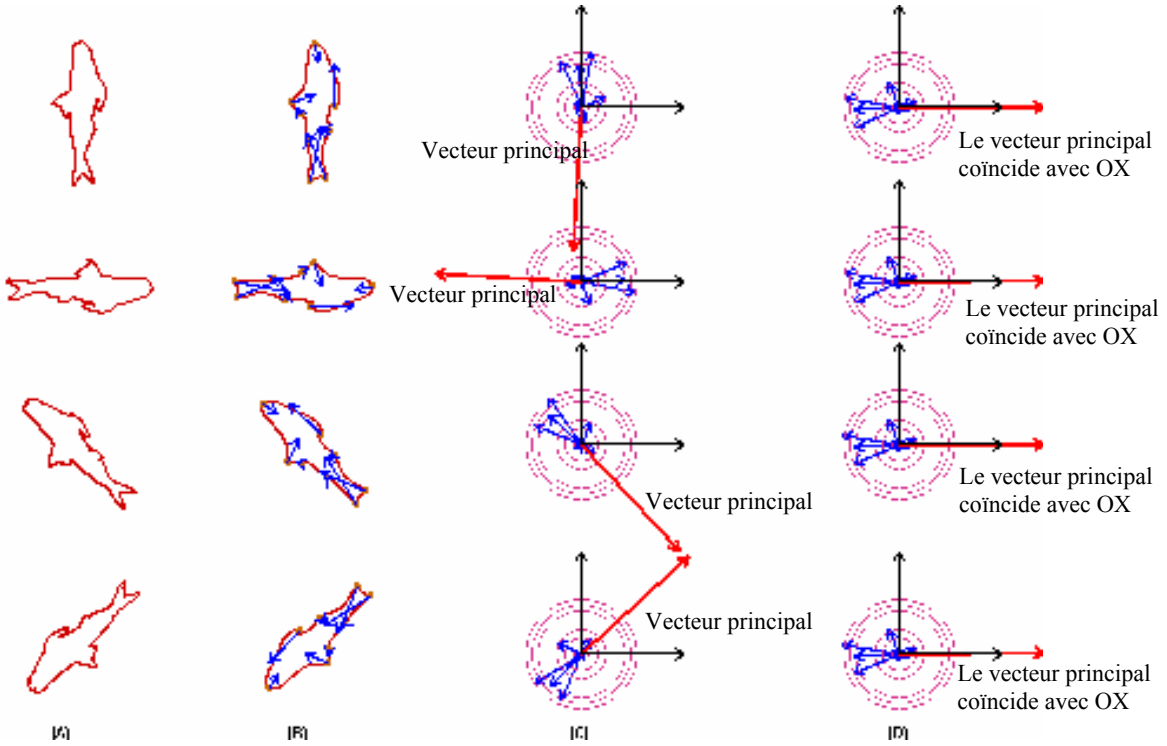


Figure 3 - *Invariance par rapport à la rotation : (A) une même forme dans quatre orienations différentes. (B) forme avec les vecteurs caractéristiques associés . (C) présentation des vecteurs caractéristiques et le vecteur principal dans un repère euclidien. (D) présentation des différents vecteurs après l'application d'une rotation* $R(-\theta)$ *à tous les vecteurs où* θ *est la direction principale.*

Notre algorithme utilisé pour calculer le descripteur est le suivant :

Procedure *calcul_descripteur*(Forme F)
{ Soit A : l'ensemble des vecteurs (courbure, orientation)
Soit A_θ : l'ensemble des vecteurs caractéristiques
$A \leftarrow \emptyset$; $A_\theta \leftarrow \emptyset$;
Pour chaque valeur de lissage σ faire
{ Pour chaque token f_i de F
{ Calculer $E_{i(\sigma)}(m_{i(\sigma)}, O_{i(\sigma)})$.
Ajouter $E_{i(\sigma)}$ à A.
}
}
Calculer la direction principale θ de l'ensemble A.
Pour chaque vecteur E_i de A faire
{ $E_i' \leftarrow R(-\theta).E_i$
Ajouter E_i' à A_θ
}
}

Notons que notre descripteur présente plusieurs avantages :

- Le lissage permet d'avoir la robustesse au bruit.
- L'invariance par rapport à la translation, l'homothétie et la rotation.
- Il permet d'éviter le problème du choix du token de référence utilisé dans le travail de Berretti et al. [8].

3 Méthode d'indexation

Nous avons choisi une méthode d'indexation basée sur les arbres métriques [15], elle nécessite la définition d'un espace métrique M= (D,d) pour mesurer la similarité entre les formes où D est le domaine de vecteurs caractéristiques et d la fonction de distance vérifiant les trois axiomes suivantes :

$d(O_x, O_y) = d(O_y, O_x)$ (symétrie)
$d(O_x, O_y) > 0$ si $O_x \neq O_y$ et $d(O_x, O_x) = 0$ (Positivité)
$d(O_x,O_y) \leq d(O_x,O_z)+d(O_z,O_y)$ (inégalité du triangle)

3.1 Présentation de la structure d'index

L'arbre M-tree organise l'espace de vecteurs caractéristiques en un ensemble de régions ou clusters imbriquées chacune correspond à un nœud de l'arbre et regroupe un ensemble de formes qui sont similaires (ou très rapprochées). Il est composé de noeuds internes et de nœuds feuilles. Chaque noeud comporte un ensemble de M entrées au maximum (la capacité du noeud).

Toute entrée d'un nœud correspond à une forme de la base. Sa structure dépend de la nature du nœud :

- Entrée d'une feuille :

$entry(F_j) = [F_j, oid(F_j), d(F_j; P(F_j))]$ où :
F_j : les descripteurs de la forme F_j.
$oid(F_j)$: l'identifiant de la forme (nom du fichier image).
$d(F_j; P(F_j))$: distance entre F_j et $P(F_j)$ qui est l'objet père .

- Entrée d'un noeud interne :

$entry(F_r) = [F_r, ptr(T(F_r)), r(F_r), d(F_r; P(F_r))]$ où :
F_r : les descripteurs de la forme F_r (routing object).
$ptr(T(F_r))$: pointeur sur un sous arbre $T(F_r)$
$r(F_r)$: le rayon de couverture, c'est la distance maximale entre F_r et les formes de la sous région.
$d(F_r; P(F_r))$: distance entre F_r et $P(F_r)$ qui est l'objet père.

L'arbre est construit par insertions successives. Insérer un nouvel objet (une forme) consiste à trouver le noeud feuille le plus convenable pour ajouter cet objet. Ceci entraîne éventuellement la décomposition du nœud feuille s'il est saturé (appel à la fonction *Split*). Le découpage nécessite deux méthodes *Promote* et *Partition* : la méthode *Promote* consiste a choisir deux objets internes O_{p1} et O_{p2} à insérer dans le nœud père N_p, tandis que la méthode *Partition* consiste à partitionner les M+1 entées en deux sous-ensembles disjoints.

3.2 Recherche par similarité dans M-tree

Pour réduire au minimum le nombre de nœuds consultés et le nombre des distances calculées, toute l'information concernant ces distances sont stockées dans les nœuds de l'arbre. Dans ce travail, deux types de requêtes de similarité sont développés : requête par intervalle (range query) et k-plus proches voisins (k-nearest neighbors query).

Comme le nombre de vecteurs caractéristiques d'une forme requête et celui d'une forme cible ne sont pas toujours égaux, nous avons utilisé la distance de Hausdorff définie entre les ensembles pour mesurer la similarité entre deux formes [16].

Soient $X = \{x_i(m_i, O_i) / 1 \leq i \leq n\}$ l'ensemble de descripteurs d'une forme F et $Y = \{y_j(m_j, O_j) / 1 \leq j \leq m\}$ l'ensemble de descripteurs d'une forme requête Q.

La distance de Hausdorff entre X et Y est définie par :

$$d(X,Y) = \frac{\sum_{i=1}^{n} d(x_i, Y) + \sum_{j=1}^{m} d(y_j, X)}{|X| + |Y|}$$

Où :

$|X|$ est le cardinal de X

$d(x_i, Y) = \min_{y_j \in Y} d(x_i, y_j)$ est la distance entre un vecteur x_i de X et l'ensemble Y tel que $d(x_i,y_j)$ est la distance entre deux vecteurs donnée par :

$$d(x_i, y_j) = \alpha \cdot |m_{x_i} - m_{y_j}| + (1-\alpha) \cdot |O_{x_i} - O_{y_j}|$$

Afin de comparer deux vecteurs x_i et y_j tout en donnant la même importance pour chaque attribut (courbure et orientation), nous prenons α=1/2.

4 Résultats

Pour illustrer l'intérêt de notre méthode, nous avons développé une application en Visual C++. Elle permet à l'utilisateur d'interroger la base de formes à partir d'une interface graphique. Nous avons utilisé une base de 1100 images d'animaux marins [17].

4.1 Construction de l'arbre

La construction de l'arbre est effectuée d'une manière

dynamique da bas vers le haut. Nous avons utilisé l'algorithme de construction présenté dans [13]. Chaque entrée d'un nœud de l'arbre stocke un ensemble de vecteurs caractéristiques correspondant à une forme de la base. La figure 5 montre un exemple de construction de la l'arbre pour 10 formes de la figure 4.

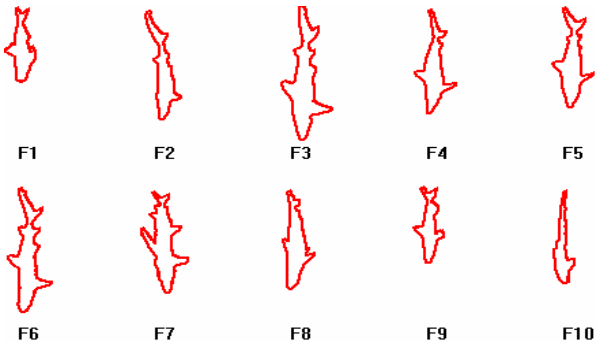


Figure 4 – *Exemples de formes de la base*

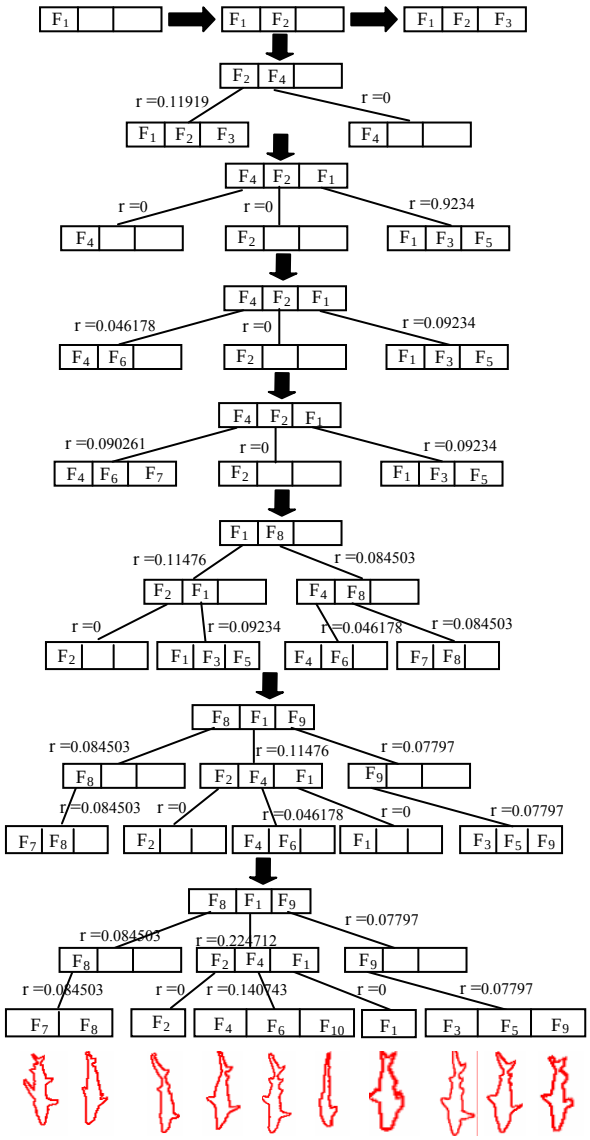
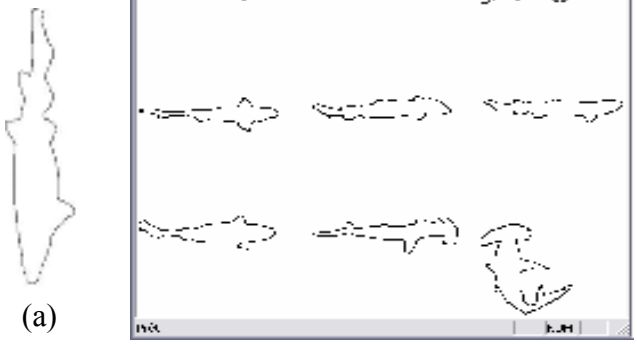


Figure 5 - *Construction de l'arbre M-tree*

4.2 Recherche d'une forme requête

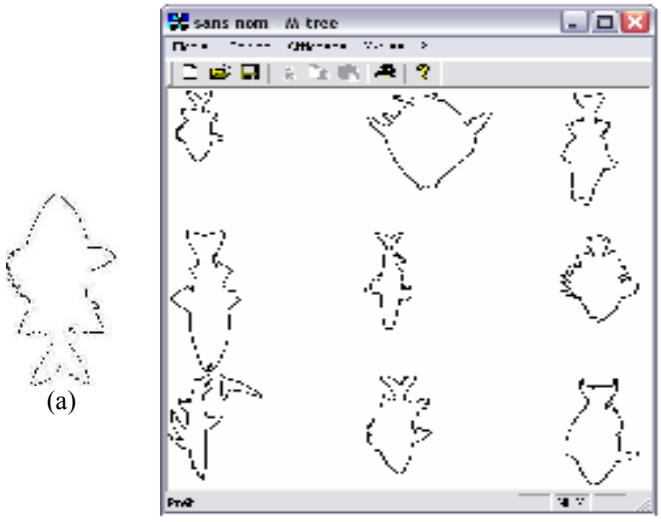
La recherche d'un objet requête se fait en parcourant la structure d'arbre en profondeur tout en éliminant le parcours des branches inutiles.

Nous avons mené plusieurs expérimentations. Les figures 6 et 7 montrent chacune un exemple d'une forme requête et les résultas obtenus de la recherche des k-plus proches voisins en fixant k à 9. Les 9 premières formes les plus ressemblantes à la forme requête sont affichées de gauche à droite puis de haut en bas.



(a) (b)

Figure 6 - *(a) Forme requête (b) les plus proches voisins obtenus.*



(a) (b)

Figure 7 - *(a) Forme requête (b) les 9 plus proches voisins obtenus.*

Notons que les résultats obtenus sont très proches de ce qu'un utilisateur aurait pu trouver visuellement. Les formes obtenues ont différentes orientations.

4.3 Temps de réponse

Nous avons mené également plusieurs tests pour montrer la performance de notre système en comparant notre méthode à la recherche séquentielle. La figure 8 présente le temps de réponse des deux méthodes en fonction de la taille de la base. Les résultats obtenus permettent de

montrer l'intérêt de notre approche au niveau de l'indexation de la base de formes en réduisant d'une façon remarquable le temps réponse.

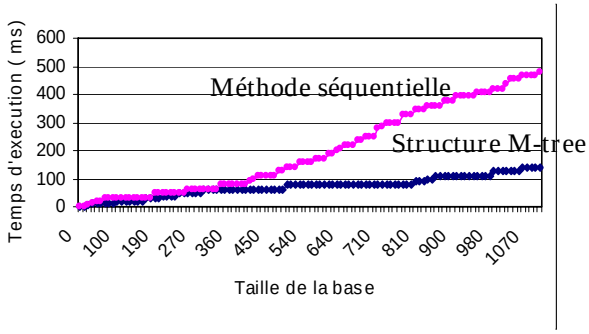


Figure 8 - *Comparaison entre M-tree et la méthode séquentielle*

5 Conclusion et perspectives

Dans ce papier, nous avons proposé une nouvelle approche pour la description, l'indexation et la recherche dans les grandes bases de formes. Le descripteur ainsi proposé est une version multi-échelle de celui de Berretti et al., qui n'est pas invariant par rapport à la rotation. Pour le rendre ainsi, nous avons réalisé une rotation de tous les vecteurs d'orientations des tokens suivant un angle calculé en utilisant le principe d'équilibre des vecteurs des forces. De plus, les attributs d'un token sont calculés à plusieurs valeurs d'échelles. En ce qui concerne l'indexation, au lieu d'associer un arbre M-tree à chaque forme de la base, nous avons stocké l'ensemble des index de la totalité de la base de formes dans une seule structure en arbre M-tree. Ceci permet d'éviter le parcours séquentiel lors de la recherche dans la base. Les résultats obtenus par l'application de notre système sur une grande base de formes, montrent l'intérêt des deux contributions de description et d'indexation de formes. Nous avons montré aussi que le système proposé est plus performant en temps de réponse que les autres systèmes basés sur le parcours séquentiel. D'autres expérimentations menées sur une base de formes de taille plus grande (11 000 formes) ont été présentées dans un autre travail [18].
Parmi les perspectives, nous envisageons d'intégrer d'autres descripteurs de formes pour enrichir notre système. Tester aussi d'autres techniques d'indexation pourra également, faire l'objet d'un travail ultérieur.

Références

[1] E. Binaghi, I. Gagliardi, and R. Schettini, "Image retrieval using fuzzy evaluation of color similarity," Int. J. Pattern Recognit. Artif. Intell., vol.8, no. 4, pp. 945–968, May 1994.

[2] C. Faloutsos, M. Flickner, W. Niblack, D. Petkovic, W. Equitz, and R. Barber, "The Qbic Project: Efficient and Effective Querying by Image Content," IBM Res. Div. Almaden Res. Center, Res. Rep. 9453, Aug. 1993.

[3] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness —Wold features for image modeling and retrieval," IEEE Trans. Pattern Anal. Machine Intell., vol. 18, pp. 722–733, July 1996.

[4] H. Tamura, S. Mori, and T. Yamawaki, "Texture features corresponding to visual perception," IEEE Trans. Syst., Man, Cybern., vol. SMC-6, pp.460–473, Apr. 1976.

[5] A. Del Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 121–132, Feb. 1997.

[6] F.Mokhtarian, S.Abbasi, and J.Kittler, "Efficient and robust retrieval by shape content through curvature scale spac", Proc. International Workshop on Image Databases and MultiMedia Search, Amsterdam, The Netherlands, pp 35-42, 1996.

[7] R.C.Veltkamp et M.Tanase. Content-based image retreival systems : A survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, October 2000.

[8] S. Berretti, A. Del Bimbo, P. Pala "Retrieval by shape Similarity with Perceptual Distance and Effective Indexing". IEEE Transactions On Multimedia, Vol. 2, No 4., December 2000

[9] Dengsheng Zhang , Guojun Lu., "Review of shape representation and description techniques", *Pattern Recognition* 37 (2004) 1 – 19.

[10] N.Guttman, « R-trees : A dynamic index structure for spatial searching ", In Proc. 1984 ACM SIGMOD Int. Conf. Management of Data, pp 47-57, Bostone, MA, June 1984

[11] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," in Proc. ACM SIGMOD Int. Conf. Management of Data, May 1990, pp.322–331.

[12] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The r+ -tree: A dynamic index for multi-dimensional objects," in Proc. 13th VLDB Int. Conf.,Sept. 1987, pp. 507–518.

[13] P. Ciaccia, M. Patella, F. Rabitti, P. Zezula. "Indexing Metric Spaces with M-tree". In SEBD'97, pp 67-86, 1997.

[14] S. Mahmoudi et M. Daoudi, "Une nouvelle méthode d'indexation 3D", 13ème Congrès Francophone de Reconnaissance des Formes et Intelligence Articifielle (RFIA2002), volume 1, pp. 19-27, Angers, France, 8-9 janvier 2002.

[15] J. K. Ulmann.Satisfying general proximity/similarity queries with metric trees. Information Processing Lettres,40(4):175-179,November 1991.

[16] I. Remolar, M. Chover, Ó. Belmonte, J. Ribelles, C. Rebollo," Geometric Simplification of Foliage", EUROGRAPHICS 2002.

[17] http://www.ee.surrey.ac.uk/Research/VSSP/imagedb.

[18] H. Silkan, A. Lachkar, S. E. Ouatik, A. Elkharraz, Une nouvelle approche pour la description, l'indexation et la recherche dans les grandes bases de formes. Soumis à MCSEAI, Agadir December 7-9, 2006, Maroc.

Surface active pour la segmentation d'images 3D : comparaison de méthodes d'évolution

J. Mille, R. Boné, P. Makris, H. Cardot
Université François Rabelais de Tours, Laboratoire Informatique (EA2101)
64 avenue Jean Portalis, 37200 Tours
{julien.mille, romuald.bone, pascal.makris, hubert.cardot}univ-tours.fr

Concours Jeune Chercheur : Oui

Résumé

Les modèles déformables constituent un outil général et puissant pour la segmentation d'images. Au cours des quinze dernières années, les modèles déformables 3D (ou surfaces actives) ont été largement utilisés pour déterminer les frontières de régions d'intérêt dans des images volumétriques. Ils ont été développés à partir de nombreuses représentations et algorithmes d'évolution. Nous considérons ici un type de surface active particulier, le maillage déformable, sur lequel sont appliquées deux méthodes d'évolution différentes. L'article présente le modèle de surface active et ces deux méthodes : l'algorithme glouton (greedy) et l'approche physique. Nous comparons les deux approches en terme de complexité, de rapidité d'exécution ainsi que de qualité de segmentation, évaluée grâce à une fonction de distance comparant la frontière déterminée par la surface active à une vérité terrain. Des expérimentations sont menées sur des images 3D artificielles et réelles.

Mots clefs

Segmentation 3D, surface active, maillage, algorithme glouton, approche physique.

1 Introduction

Depuis leur introduction par Kass *et al.*[1], les modèles déformables tels que les contours actifs (souvent désignés sous le nom de "snakes") ont suscité un vif intérêt parmi les chercheurs et ont trouvé de nombreuses applications en vision par ordinateur, principalement en segmentation d'images. Les modèles déformables sont un outil général et puissant pour obtenir des informations géométriques concernant une région d'intérêt, i.e. les coordonnées des points appartenant aux contours de l'objet à segmenter. Développé initialement pour les images 2D, le modèle de contour actif a été étendu en 3D, donnant ainsi naissance à la surface active, afin de segmenter des objets dans des images volumétriques. A partir d'une position initiale fournie par l'utilisateur, le modèle se déforme itérativement selon un algorithme d'évolution dont le but est d'ajuster la surface aux frontières de l'objet.

Diverses représentations des surfaces actives ont été développées dans la littérature. Pour un état de l'art des modèles déformables 3D, le lecteur pourra se référer à [2] et [3]. Parmi les surfaces 3D, les maillages sont des représentations explicites discrètes. L'information stockée et manipulée est ici un ensemble de sommets interconnectés (les points appartenant à la surface). La surface est déformée par modifications des coordonnées des sommets. Deux principaux types de maillage sont souvent rencontrés : les maillages triangulaires [4] et les maillages simplexes [5]. Par opposition aux représentations explicites, les surfaces implicites, comme le modèle géométrique 3D de Kimmel [6], sont basées sur des formulations par ensembles de niveaux [7]. La surface est alors définie comme le niveau 0 d'une fonction dont le support a la même dimension que l'image (en 3D, la fonction est de type $\mathbb{R}^3 \rightarrow \mathbb{R}$). Un avantage souvent évoqué par rapport aux implémentations explicites est la gestion naturelle des changements de topologie. En effet, pour les maillages, des mécanismes de détection de changement de topologie doivent être mis en place [8], et ce indépendamment de la méthode d'évolution. L'inconvénient majeur des ensembles de niveaux est leur coût de calcul élevé (bien qu'il existe des optimisations telles que la méthode de la bande étroite [7]). Ceci est d'autant plus vrai lorsque la résolution augmente, car leur complexité algorithmique est directement fonction de la taille de l'image.

Dans le cas où le temps de calcul est crucial, les implémentations explicites telles que les maillages sont plus adéquats. De plus, si nous connaissons a priori la topologie globale de l'objet, il n'est pas nécessaire d'implémenter des mécanismes de changement de topologie. Parmi les différentes méthodes d'évolution appliquées aux maillages, nous nous focalisons sur deux approches. D'un coté, l'algorithme glouton ou *greedy* [9] consiste à minimiser itérativement une fonctionnelle d'énergie associée à la surface. De l'autre, l'approche physique [4] applique des forces afin d'atteindre un état d'équilibre, conformément à une loi de mouvement. Nous décrivons une implémentation de ces deux méthodes, que nous considérons comme duales, de par la relation existante entre les notions de force et d'énergie. Sur un plan

théorique, nous évaluons la complexité algorithmique et le paramétrage des deux approches. Lors d'expérimentations sur des images 3D, les deux approches sont comparées en terme de rapidité d'exécution ainsi que de qualité de segmentation. Cette dernière est évaluée grâce à une fonction de comparaison entre la frontière déterminée par la surface active et la frontière réelle fournie par segmentation experte.

2 Modèle de surface active

2.1 Représentation géometrique

Dans un domaine discret, une surface est représentée par un maillage, composé d'un ensemble de sommets (les données géométriques, i.e. les points appartenant à la surface) et d'un ensemble d'arêtes liant les sommets (les données topologiques). Les arêtes forment un ensemble de triangles adjacents. Par la suite, nous noterons $\mathbf{p}_i = (x_i, y_i, z_i)^T$ le $i^{ème}$ sommet, n le nombre total de sommets, et V_i le voisinage de $\mathbf{p}_i$ (l'ensemble de sommets connectés à $\mathbf{p}_i$). A chaque sommet, on associe également un vecteur unitaire $\vec{\mathbf{n}}_i$, normal à la surface au point $\mathbf{p}_i$. Nous calculons la normale du sommet $\mathbf{p}_i$ comme la moyenne des normales des triangles voisins de $\mathbf{p}_i$:

$$\vec{\mathbf{n}}_i = \frac{\sum_{t \in T_i} \vec{\mathbf{n}}_t}{\left\| \sum_{t \in T_i} \vec{\mathbf{n}}_t \right\|} \tag{1}$$

où T_i est l'ensemble des triangles voisins de $\mathbf{p}_i$. La normale $\vec{\mathbf{n}}_t$ d'un triangle est déterminée par le produit vectoriel normalisé de deux vecteurs appartenant au plan défini par le triangle.

$$\vec{\mathbf{n}}_t = s_t \frac{(\mathbf{p}_{t_2} - \mathbf{p}_{t_1}) \wedge (\mathbf{p}_{t_3} - \mathbf{p}_{t_1})}{\|(\mathbf{p}_{t_2} - \mathbf{p}_{t_1}) \wedge (\mathbf{p}_{t_3} - \mathbf{p}_{t_1})\|} \tag{2}$$

où $\mathbf{p}_{t_j}, j = 1...3$ sont les sommets du triangle t ($\mathbf{p}_i$ est obligatoirement l'un d'eux). $s_t = \pm 1$ est le signe qui change l'orientation de $\vec{\mathbf{n}}_t$, pour assurer que le vecteur pointe vers l'intérieur de la surface. Orienter ainsi les normales est nécessaire pour l'implémentation du ballon (cf. section suivante).

Le maillage est construit par subdivision d'un icosaèdre, un polyèdre régulier comportant 12 sommets et 20 faces. Le principe de subdivision, décrit dans [10], consiste à casser chaque triangle en quatre triangles plus petits, en ajoutant de nouveaux sommets et arêtes. En construisant le maillage de cette façon, nous obtenons une sphère discrétisée avec une répartition égale des sommets le long de la surface. Ainsi, le maillage initial est homogène d'un point de vue géométrique et topologique.

2.2 Evolution de la surface

Une surface est dite active si sa forme est modifiée selon une méthode d'évolution dont l'objectif est d'ajuster la surface aux contours de l'objet. Les méthodes d'évolution ont en commun le fait qu'elles font toutes intervenir un ou plusieurs terme(s) interne(s) pour maintenir la régularité de la surface et externe(s) pour attirer la surface vers les contours de l'objet. Ces termes sont des forces ou des énergies selon l'approche considérée. Nous présentons ici deux méthodes d'évolution, qui seront comparées par la suite.

Approche gloutonne. On associe à la surface une fonctionnelle d'énergie à minimiser (elle est généralement non-convexe et possède de nombreux minima locaux). Introduite par Williams et Shah [9] pour les contours actifs 2D, l'approche gloutonne est un algorithme de minimisation d'énergie. Une extension 3D de l'algorithme glouton appliquée sur un maillage triangulaire a été proposée dans [11]. Afin de minimiser l'énergie totale, des optimisations locales sont effectuées successivement. L'énergie de chaque sommet $\mathbf{p}_i$ est minimisée indépendamment de celles des autres. Pour se faire, son énergie est calculée pour chaque voxel appartenant à une fenêtre cubique centrée en $\mathbf{p}_i$, de largeur w. Après normalisation, le sommet est déplacé à la position qui donne l'énergie minimale. A chaque itération de l'algorithme, on note $\mathbf{p}_i$ la position initiale du sommet i, et $\mathbf{p}'_i$ la position testée dans la fenêtre cubique. L'énergie associée à une position testée $\mathbf{p}'_i$ est calculée comme suit :

$$\begin{aligned} E(\mathbf{p}'_i) &= \alpha E_{cont}(\mathbf{p}'_i) + \beta E_{curv}(\mathbf{p}'_i) \\ &+ \gamma E_{grad}(\mathbf{p}'_i) + \delta E_{bal}(\mathbf{p}'_i) \end{aligned} \tag{3}$$

Les paramètres (α, β, γ, δ) pondèrent les énergies et doivent être ajustés empiriquement. Les poids α et β sont respectivement l'élasticité et la rigidité de la surface. Plus leurs valeurs sont faibles, plus l'on autorise la surface à se distendre et se courber (dans une certaine mesure, les poids traduisent la connaissance *a priori* sur la forme de l'objet dont dispose l'utilisateur). L'énergie de continuité est une extension 3D de celle couramment rencontrée en segmentation 2D :

$$E_{cont}(\mathbf{p}'_i) = \sum_{j \in V_i} \left| \overline{d^2} - \|\mathbf{p}'_i - \mathbf{p}_j\|^2 \right| \tag{4}$$

Minimiser cette énergie revient à réduire l'écart-type des distances, de sorte que la distance moyenne entre un sommet et ses voisins soit approximativement la même pour chaque sommet. Par conséquent, les sommets demeurent espacés de façon homogène le long de la surface. $\overline{d^2}$ est la distance moyenne globale :

$$\overline{d^2} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{card(V_i)} \sum_{j \in V_i} \|\mathbf{p}_i - \mathbf{p}_j\|^2 \tag{5}$$

Etant donné un contour 2D implémentant une courbe paramétrique $\mathcal{C} : s \mapsto (x(s), y(s))$, la courbure est l'approximation par différences finies de $\left\| \partial^2 C / \partial s^2 \right\|$. La courbure au sommet $\mathbf{p}_i$ est fonction de la distance entre $\mathbf{p}_i$ et le milieu du segment $[\mathbf{p}_{i-1}\mathbf{p}_{i+1}]$. Par extension à la surface, l'énergie de courbure à la position testée $\mathbf{p}'_i$ est la distance

entre $\mathbf{p}'_i$ et le barycentre des voisins de $\mathbf{p}_i$. En considérant que les voisins sont situés régulièrement autour du sommet, la minimiser a pour effet de lisser la surface.

$$E_{curv}(\mathbf{p}'_i) = \left\| \mathbf{p}'_i - \frac{1}{card(V_i)} \sum_{j \in V_i} \mathbf{p}_j \right\|^2 \quad (6)$$

Pour exprimer l'énergie de gradient $E_{grad}(\mathbf{p}'_i)$, nous utilisons la norme du gradient de l'image. En présence de données bruitées, l'image est lissée avec un filtre gaussien avant le calcul du gradient. Dans les équations suivantes, G_σ est un masque gaussien d'écart-type σ et $*$ est l'opérateur de convolution.

$$E_{grad}(\mathbf{p}'_i) = -\left\| (\nabla I * G_\sigma)(\mathbf{p}'_i) \right\| \quad (7)$$

Pour le calcul de la norme du gradient, nous effectuons une détection de contours 3D, en convoluant l'image avec l'opérateur de Zucker-Hummel [12], composé de trois masques de taille $3 \times 3 \times 3$. Par exemple, le masque suivant filtre l'image selon l'axe x.

$$Z_x = \begin{bmatrix} -k_1 & 0 & k_1 \\ -k_2 & 0 & k_2 \\ -k_1 & 0 & k_1 \end{bmatrix} \begin{bmatrix} -k_2 & 0 & k_2 \\ -k_3 & 0 & k_3 \\ -k_2 & 0 & k_2 \end{bmatrix} \begin{bmatrix} -k_1 & 0 & k_1 \\ -k_2 & 0 & k_2 \\ -k_1 & 0 & k_1 \end{bmatrix}$$

$$k_1 = \frac{\sqrt{3}}{3} \; ; \; k_2 = \frac{\sqrt{2}}{2} \; ; \; k_3 = 1 \quad (8)$$

Nous ajoutons au modèle une énergie ballon $E_{bal}(\mathbf{p}'_i)$, dérivée de la force d'inflation proposée par Cohen [13], basée sur le vecteur normal en chaque sommet. Elle permet notamment à la surface d'être initialisée loin de l'objet.

$$E_{bal}(\mathbf{p}'_i) = \left\| \mathbf{p}'_i - (\mathbf{p}_i + w\vec{\mathbf{n}}_i) \right\|^2 \quad (9)$$

On fait intervenir la taille de la fenêtre w pour que le vecteur pointe en dehors de la fenêtre. Les normales $\vec{\mathbf{n}}_i$ étant orientées vers l'intérieur de la surface, nous garantissons que l'énergie ballon a le même effet sur tous les sommets. Le signe du coefficient de pondération δ contrôle l'orientation du mouvement ballon et doit être initialisé en fonction de la position de départ de la surface par rapport à l'objet : si les sommets sont placés à l'extérieur de l'objet, δ devra être positif pour que la surface puisse se rétracter, et inversement.

Les énergies sont normalisées avant de déterminer où le sommet doit être déplacé, de façon à ce que la contribution de chaque énergie soit approximativement la même pour chaque position testée. L'expression suivante est un exemple de normalisation appliquée sur l'énergie de continuité.

$$E_{cont}(\mathbf{p}'_i) = \frac{E_{cont}(\mathbf{p}'_i) - \min_k E_{cont}(\mathbf{p}'_k)}{\max_k E_{cont}(\mathbf{p}'_k) - \min_k E_{cont}(\mathbf{p}'_k)} \quad (10)$$

Approche physique. Basée sur des principes rencontrés en mécanique, l'approche physique [8, 5] fait intervenir la notion de forces (par opposition aux énergies rencontrées précédemment). La déformation de la surface est régie non plus par la minimisation d'une fonctionnelle d'énergie, mais par la recherche d'un état d'équilibre, conformément à une équation d'évolution. Cette méthode est également appelée dynamique, car la variable temps apparaît dans l'équation d'évolution. Chaque sommet $\mathbf{p}_i$ évolue selon une loi de Newton, qui fait intervenir des termes inertiels du premier et second ordre (respectivement la vitesse et l'accélération du sommet) ainsi que le vecteur force appliqué au sommet :

$$m\frac{d^2\mathbf{p}_i}{dt^2} + \mu\frac{d\mathbf{p}_i}{dt} = \vec{\mathbf{F}}(\mathbf{p}_i) \quad (11)$$

où m et μ sont respectivement la masse et le coefficient de viscosité (égaux pour tous les sommets). On considère la masse comme nulle, ce qui conduit ainsi à une loi Lagrangienne du mouvement. Une fois la discrétisation temporelle d'Euler appliquée (Δt étant le pas temporel), on obtient un schéma d'évolution défini de façon explicite :

$$\begin{aligned} \frac{\mu}{\Delta t}(\mathbf{p}_i^{(t+1)} - \mathbf{p}_i^{(t)}) &= \vec{\mathbf{F}}(\mathbf{p}_i^{(t)}) \\ \mathbf{p}_i^{(t+1)} &= \mathbf{p}_i^{(t)} + \frac{\mu}{\Delta t}\vec{\mathbf{F}}(\mathbf{p}_i^{(t)}) \end{aligned} \quad (12)$$

A chaque itération, le sommet $\mathbf{p}_i$ est translaté le long du vecteur $\vec{\mathbf{F}}(\mathbf{p}_i)$, force globale calculée à partir des forces internes et externes agissant sur $\mathbf{p}_i$. Comme $\vec{\mathbf{F}}(\mathbf{p}_i)$ est déjà une somme pondérée, l'équation 12 est simplifiée : les notions de pas temporel et de viscosité sont inclues dans les poids appliqués aux forces. En conséquence, quatre poids doivent être réglés manuellement par l'utilisateur. La force globale est calculée de la façon suivante :

$$\begin{aligned} \vec{\mathbf{F}}(\mathbf{p}_i) &= \alpha\vec{\mathbf{F}}_{cont}(\mathbf{p}_i) + \beta\vec{\mathbf{F}}_{curv}(\mathbf{p}_i) \\ &+ \gamma\vec{\mathbf{F}}_{grad}(\mathbf{p}_i) + \delta\vec{\mathbf{F}}_{bal}(\mathbf{p}_i) \end{aligned} \quad (13)$$

Les forces $\vec{\mathbf{F}}_{cont}$, $\vec{\mathbf{F}}_{curv}$, $\vec{\mathbf{F}}_{grad}$ et $\vec{\mathbf{F}}_{bal}$ sont les équivalents vectoriels des énergies décrites dans la section précédente. En effet, les notions de forces et d'énergies sont liées dans la mesure où la force est l'opposé de la dérivée spatiale de l'énergie correspondante, à une constante près. Etant donné un opérateur de dérivation par rapport à un point $\partial f/\partial \mathbf{p} = (\partial f/\partial p_x, \partial f/\partial p_y, \partial f/\partial p_z)^T$, la dérivée d'une distance au carré donne :

$$\frac{\partial \left\| \mathbf{p} - \mathbf{q} \right\|^2}{\partial \mathbf{p}} = 2(\mathbf{p} - \mathbf{q}) \quad (14)$$

En suivant ce principe, étant donnés deux points $\mathbf{p}$ et $\mathbf{q}$, le déplacement qu'il faut appliquer à $\mathbf{p}$ pour minimiser l'énergie $\left\| \mathbf{p} - \mathbf{q} \right\|^2$ est $\mathbf{q} - \mathbf{p}$. Lorsque l'on cherche à déplacer un point $\mathbf{p}$ afin de minimiser une énergie E fonction de $\mathbf{p}$, la dérivée $\partial E/\partial \mathbf{p}$ donne la direction du déplacement. Ainsi, l'approche physique peut être considérée comme une descente de gradient de la fonctionnelle d'énergie.

En ce qui concerne les forces internes, la force de continuité est similaire à celle utilisée dans [4]. Elle fait intervenir la même distance moyenne au carré $\overline{d^2}$ utilisée dans notre algorithme glouton. La force de courbure attire le sommet vers le centre de gravité de ses voisins.

$$\vec{\mathbf{F}}_{cont}(\mathbf{p}_i) = \sum_{j \in V_i} (\overline{d^2} - \|\mathbf{p}_i - \mathbf{p}_j\|^2) \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|} \quad (15)$$

$$\vec{\mathbf{F}}_{curv}(\mathbf{p}_i) = (\frac{1}{card(V_i)} \sum_{j \in V_i} \mathbf{p}_j) - \mathbf{p}_i \quad (16)$$

La force de gradient $\vec{\mathbf{F}}_{grad}$ est fonction de la dérivée spatiale de l'énergie de gradient E_{grad}. Elle est calculée *a priori* en chaque point de l'image, générant ainsi un champ vectoriel. Notons qu'un champ vectoriel plus évolué, le *gradient vector flow* (GVF) de Xu *et al.* [14] est souvent utilisé. Il est obtenu par diffusion du gradient et possède donc un champ de capture beaucoup plus large. Cependant, pour conserver l'équivalence entre les deux approches, du point de vue de capacité d'attraction de la surface, nous ne l'avons pas utilisé. Pour exprimer la force ballon, le vecteur normal à la surface défini au sommet $\mathbf{p}_i$ est directement appliqué.

$$\vec{\mathbf{F}}_{grad}(\mathbf{p}_i) = -\nabla \|\nabla I(\mathbf{p}'_i)\| \quad (17)$$

$$\vec{\mathbf{F}}_{bal}(\mathbf{p}_i) = \vec{\mathbf{n}}_i \quad (18)$$

A ce stade, une comparaison préliminaire des deux méthodes est possible. Il s'agit tout d'abord de deux approches duales du point de vue de la formulation des énergies et des forces (à partir de chaque énergie, une force a pu être élaborée). Si l'on exprime la complexité algorithmique en fonction de la quantité de sommets, l'approche physique est en $O(n)$, alors que l'algorithme glouton est en $O(nw)$. Quand au paramétrage, dans l'approche gloutonne, les poids des énergies ont une importance relative (un des poids peut être fixé et les autres ajustés), car ils interviennent dans une fonctionnelle à minimiser. Dans l'approche physique, les poids incluent la notion de pas temporel, et ont une signification absolue (s'ils sont trop élevés, la force résultante risquerait de passer outre les frontières de l'objet ou de créer un phénomène d'oscillations). De plus, l'algorithme glouton est basé sur des mouvements pas à pas, puisque les sommets sont déplacés dans des fenêtres selon un pas spatial égal au voxel (dans une certaine mesure, les coordonnées peuvent être stockées comme des valeurs entières). Ce n'est pas le cas dans l'approche physique, où la précision des déplacements n'est pas limitée.

3 Méthode de remaillage

Afin de maintenir une distribution homogène des sommets quelles que soient les déformations subies par la surface, un remaillage adaptatif est effectué [8]. Des sommets peuvent être créés ou supprimés de façon à conserver une distance homogène entre les sommets voisins. Le remaillage garantit que tout couple de voisins $(\mathbf{p}(i), \mathbf{p}(j))$ satisfasse la contrainte suivante :

$$d_{min} \leq \|\mathbf{p}_i - \mathbf{p}_j\| \leq d_{max} \quad (19)$$

où d_{min} et d_{max} sont deux seuils définis par l'utilisateur, vérifiant $d_{max} \geq 2d_{min}$. Le fait d'ajouter et de supprimer des sommets modifie la topologie locale ; des contraintes topologiques doivent donc être vérifiées. Considérons un couple de sommets voisins ($\mathbf{p}_i$,$\mathbf{p}_i$). Pour que l'ajout ou la fusion de sommets soient possibles, $\mathbf{p}_i$ et $\mathbf{p}_j$ doivent posséder exactement deux voisins en commun :

$$N_i \cap N_j = \{a, b\} \quad (20)$$

Avant l'opération de remaillage, les sommets $\mathbf{p}_a$ et $\mathbf{p}_b$ sont tous deux adjacents à $\mathbf{p}_i$ et $\mathbf{p}_j$ (figure 1 à gauche). Ce cas de figure est le plus courant, cependant il arrive qu'au fur et à mesure des remaillages des configurations topologiques problématiques apparaissent, c'est pourquoi la condition 20 est testée systématiquement. Lorsque la condition $\|\mathbf{p}(i) - \mathbf{p}(j)\| > d_{max}$ est vérifiée, un nouveau sommet (d'indice $n+1$) est créé au milieu du segment $[\mathbf{p}_i\mathbf{p}_j]$ et connecté à $\mathbf{p}_a$ and $\mathbf{p}(b)$ (figure 1 au milieu). La taille du voisinage n'est pas modifiée pour $\mathbf{p}_i$ et $\mathbf{p}_j$, tandis que $\mathbf{p}_a$ et $\mathbf{p}_b$ gagnent un voisin chacun. A l'inverse, lorsque $\|\mathbf{p}_i - \mathbf{p}_j\| < d_{min}$, $\mathbf{p}_i$ est translaté au milieu du segment $[\mathbf{p}_i\mathbf{p}_j]$, puis $\mathbf{p}_j$ est supprimé de l'ensemble des sommets (figure 1 à droite). Les anciens voisins de $\mathbf{p}_j$ deviennent voisins de $\mathbf{p}_i$.

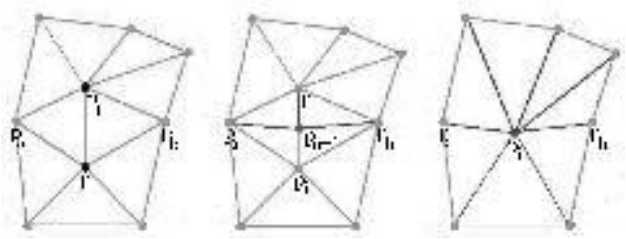


Figure 1 – *Remaillage entre les sommets* $\mathbf{p}_i$ *et* $\mathbf{p}_j$

La création de sommets permet à la surface de garder un échantillonnage suffisant lorsqu'elle accroît son volume (lorsqu'elle se dilate). A l'inverse, la fusion a pour rôle d'empêcher les sommets voisins d'être trop proches, ce qui risquerait de causer des recouvrements et des auto-intersections de la surface (triangles sécants). En employant l'algorithme glouton, deux voisins $\mathbf{p}_i$ and $\mathbf{p}_i$ risquent de se confondre à l'itération t seulement si leurs fenêtres respectives se superposent à l'itération $t-1$. Trivialement, deux cubes de taille w ne se superposent pas si leurs centres sont au moins distants de w voxels dans l'une des trois dimensions. En utilisant la norme infinie $\|\mathbf{p}_i - \mathbf{p}_j\|_\infty = \max(|x_i - x_j|, |y_i - y_j|, |z_i - z_j|)$, nous pouvons formaliser ce principe et ainsi redéfinir le critère de remaillage de l'équation 19.

$$w < \|\mathbf{p}_i - \mathbf{p}_j\|_\infty < 2w \quad (21)$$

Ainsi, le paramètre w contrôle non seulement la largeur de l'espace de recherche de l'algorithme glouton, mais aussi l'échantillonnage du maillage (la surface sera d'autant plus dense que w est petit).

4 Evaluation

La comparaison des approches gloutonne et physique porte sur le nombre d'itérations, le temps de calcul ainsi que la qualité de la segmentation. Afin d'évaluer cette dernière, nous utilisons une fonction de comparaison entre la surface réelle (la vérité terrain fournie par l'expert) et la surface obtenue par segmentation automatique. Cette comparaison a pour objet la distance globale entre le modèle et la surface réelle. Soient $\mathcal{S}$ l'ensemble des voxels appartenant à la surface active finale et $\mathcal{R}$ l'ensemble des voxels appartenant à la surface réelle. Pour chaque voxel de la surface active, nous considérons le plus proche voxel sur la surface réelle, ce qui amène la distance de Hausdorff modifiée $\mathcal{H}$, introduite dans [15].

$$\begin{aligned} \mathcal{H}(\mathcal{S},\mathcal{R}) &= \max(h(\mathcal{S},\mathcal{R}),h(\mathcal{R},\mathcal{S})) \\ h(\mathcal{S},\mathcal{R}) &= \frac{1}{card(\mathcal{S})}\sum_{\mathbf{s}\in\mathcal{S}}\min_{\mathbf{r}\in\mathcal{R}}\|\mathbf{s}-\mathbf{r}\| \end{aligned} \quad (22)$$

h étant la distance de Hausdorff orientée. Comme $\mathcal{H}$ est calculée sur tous les voxels de la surface (et non pas seulement sur les sommets), les facettes du maillage doivent être entièrement discrétisées. On utilise pour cela un algorithme de voxélisation de triangle [16] (un exemple de voxélisation de triangle est illustré sur la figure 2).

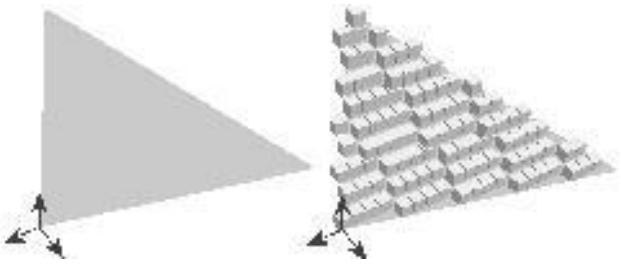


Figure 2 – *Voxélisation de triangle*

Nous présentons tout d'abord les résultats obtenus sur des images 3D artificielles bruitées selon une loi gaussienne : une avec 3 ellipsoïdes et une avec une forme hélicoïdale, toutes deux de résolution $200 \times 200 \times 200$. Un lissage gaussien a été appliqué avant de calculer le gradient et le champ vectoriel. Le poids γ a été fixé à 1 pour toutes les expérimentations. Dans la première image, la surface a été initialisée autour de l'objet, avec 2562 sommets, sans remaillage. Les valeurs des poids sont ($\alpha = 0.5$, $\beta = 0.3$, $\delta = 0.8$) pour la méthode gloutonne et ($\alpha = 0.1$, $\beta = 0.1$, $\delta = 0.15$) pour la méthode physique. Pour l'image de la spirale, la surface a été initialisée à l'intérieur de l'objet, avec 12 sommets. Pour chaque approche, deux valeurs de w ont été testées, conduisant ainsi à différents échantillonnages de la surface. Nous avons utilisé le même critère de remaillage (21) dans les deux approches (w intervient dans ce critère, c'est pourquoi il apparait également dans les résultats de l'approche physique). Les jeux de paramètres sont les suivants : ($\alpha = 0$, $\beta = 0.25$, $\delta = -0.5$) pour l'algorithme glouton et ($\alpha = 0$, $\beta = 0.1$, $\delta = -0.95$) pour l'approche physique. Les résultats figurent dans le tableau suivant (les temps de calcul, exprimés en secondes, ont été obtenus avec une implémentation C++ sur un Pentium IV 2.8GHz).

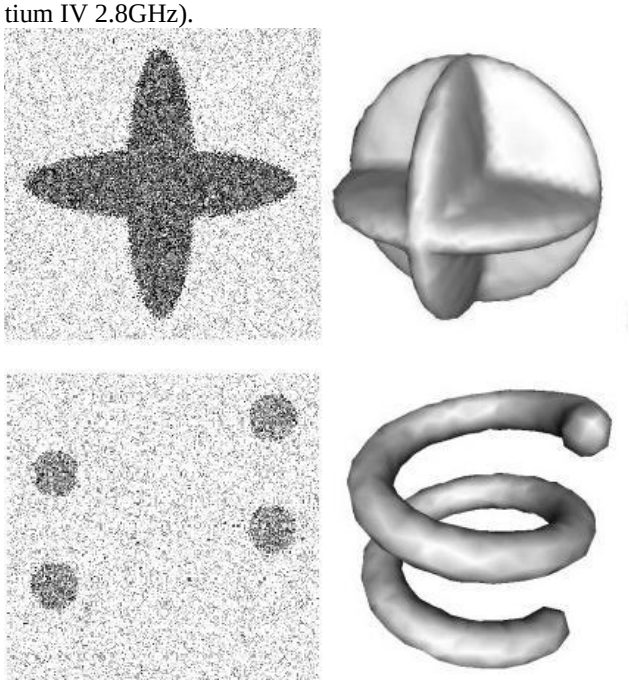


Figure 3 – *Images artificielles 3D. Gauche : coupe 2D (z=100), droite : surface 3D. Haut : ellipsoïdes, bas : spirale*

			Nb. iter.	Temps	Distance
Ellipsoïdes	Glouton	w=3	61	0.73	0.530
		w=5	39	1.00	0.529
	Physique		250	1.71	0.658
Spirale	Glouton	w=3	350	1.91	0.497
		w=5	192	0.73	0.723
	Physique	w=3	440	3.47	0.931
		w=5	448	1.15	1.161

La figure 4 présente une reconstruction 3D du résultat obtenu sur une image par résonance magnétique de l'abdomen, de résolution $512 \times 512 \times 810$. Le maillage a été ici utilisé pour segmenter l'intérieur de l'aorte complète (aorte ascendante + crosse aortique + aorte descendante), de la paroi cardiaque jusqu'à la bifurcation illiaque. Une telle segmentation est effectuée dans la cadre du diagnostic de l'anévrisme de l'aorte abdominale. La surface a été initialisée à l'intérieur du vaisseau et dilatée lors de sa déformation. Les jeux de paramètres sont ($\alpha = 0$, $\beta = 0.06$, $\delta = -0.1$) pour l'algorithme glouton et ($\alpha = 0$, $\beta = 0.25$, $\delta = 0.3$) pour l'approche physique.

		Nb. iter.	Temps	Distance
Glouton	w=3	580	25.30	1.265
	w=5	330	9.65	1.278
Physique	w=3	4050	81.42	2.514
	w=5	3780	24.23	2.335

La comparaison est effectuée sur des formes correctement segmentées (les valeurs de distances correspondent à des résultats satisfaisants quelle que soit l'approche). Il faut noter que les jeux de paramètres diffèrent selon l'approche

utilisée, car ils proviennent de la recherche d'un compromis entre vitesse de convergence et qualité de segmentation (avec d'autres paramètres, il est possible de réduire le nombre d'itérations, mais avec une qualité moindre à l'arrivée). L'approche gloutonne utilise significativement moins d'itérations pour converger, ce qui conduit globalement (et malgré une complexité algorithmique supérieure) à un temps de calcul inférieur. Les distances sont également inférieure avec cette méthode. Cela s'explique par le fait que l'approche physique a tendance à faire osciller les sommets autour des frontières. En effet, les sommets sont systématiquement translatés (la force n'est jamais nulle) et les coordonnées prennent des valeurs réelles quelconques. L'algorithme glouton est quant à lui plus stable de par sa nature, puisque les sommets y sont déplacés selon un mouvement pas à pas (le pas de déplacement est égal au voxel, donc à l'échantillonnage de l'image).

Figure 4 – *Reconstruction 3D de l'aorte*

5 Conclusion et perspectives

Dans cet article, nous avons présenté un modèle de surface active, sur lequel deux méthodes d'évolution (gloutonne et physique) ont été appliquées. Nous avons comparé les méthodes sur un plan théorique puis de façon expérimentale, sur des critères de temps de calcul et de qualité de segmentation, cette dernière étant évaluée par la distance de Hausdorff modifiée. La comparaison se révèle être en faveur de l'algorithme glouton, moins coûteux en temps de calcul et conduisant à de meilleures segmentations. Dans de futurs travaux, une méthode d'évolution hybride est envisagée : afin d'améliorer les performances, le voisinage utilisé en minimisation gloutonne pourrait être guidé et optimisé par un calcul de vecteur force.

Références

[1] M. Kass, A. Witkin, et D. Terzopoulos. Snakes : active contour models. *International Journal of Computer Vision*, 1(4) :321–331, 1987.

[2] J. Mille, R. Boné, P. Makris, et H. Cardot. 3D segmentation using active surface : a survey and a new model. Dans 5^{th} *Int. Conf. on Visualization, Imaging & Image Processing (VIIP)*, pages 610–615, Benidorm, Spain, 2005.

[3] J. Montagnat, H. Delingette, et N. Ayache. A review of deformable surfaces : topology, geometry, and deformation. *Image and Vision Computing*, 19(14) :1023–1040, 2001.

[4] J-Y. Park, T. McInerney, et D. Terzopoulos. A non-self-intersecting adaptive deformable surface for complex boundary extraction from volumetric images. *Computer & Graphics*, 25(3) :421–440, June 2001.

[5] J. Montagnat et H. Delingette. 4D deformable models with temporal constraints : application to 4D cardiac image segmentation. *Medical Image Analysis*, 9(1) :87–100, 2005.

[6] R. Kimmel. Geometric segmentation of 3D structures. Dans *International Conference on Image Processing*, volume 3, pages 639–642, Barcelona, Spain, 2003.

[7] R. Malladi, J.A. Sethian, et B.C. Vemuri. Shape modeling with front propagation : a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2) :158–175, 1995.

[8] J.O. Lachaud et A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. *Medical Image Analysis*, 3(2) :187–207, 1999.

[9] D.J. Williams et M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing : Image Understanding*, 55(1) :14–26, 1992.

[10] T. McInerney et D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1) :69–83, 1995.

[11] A.J. Bulpitt et N.D. Efford. An efficient 3D deformable model with a self-optimising mesh. *Image and Vision Computing*, 14(8) :573–580, 1996.

[12] S.W. Zucker et R.A. Hummel. A three-dimensional edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3) :324–331, 1981.

[13] L.D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing : Image Understanding*, 53(2) :211–218, 1991.

[14] C. Xu et J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3) :359–369, March 1998.

[15] M-P. Dubuisson et A.K. Jain. A modified Hausdorff distance for object matching. Dans *Proceedings of* 12^{th} *International Conference on Pattern Recognition (ICPR)*, pages 566–568, Jerusalem, Israel, 1994.

[16] A. Kaufman. Efficient algorithms for 3D scan-conversion of parametric curves, surfaces, and volumes. *Computer Graphics*, 21(3) :171–179, 1987.

Visualisation 3D temps-réel à distance de MNT par insertion de données cachées basée ondelettes

K. Hayat[1] W. Puech[1,2] G. Gesquière[3] M. Chaumont[1,2]

[1] Laboratoire LIRMM, UMR CNRS 5506,
Université Montpellier II - France

[2] Centre Universitaire de Formation et de Recherche de Nîmes - France

[3] Laboratoire LSIS, UMR CNRS 6168,
Université Aix-Marseille - France

khizar.hayat@lirmm.fr, william.puech@lirmm.fr,
gilles.gesquiere@lsis.org, marc.chaumont@lirmm.fr

Résumé

L'utilisation de photographies aériennes, d'images satellites, de cartes scannées et de modèles numériques de terrains amène à mettre en place des stratégies de stockage et de visualisation de ces données. Afin d'obtenir une visualisation en trois dimensions, il est nécessaire de lier ces images appelées textures avec la géométrie du terrain nommée Modèle Numérique de Terrain (MNT). Ces informations sont en pratiques stockées dans trois fichiers différents : MNT, texture, position et projection des données dans un système géo-référencé. Dans cet article, nous proposons de stocker toutes ces informations dans un seul fichier afin de les synchroniser. Nous avons développé pour cela une méthode d'insertion de données cachées basée ondelettes dans une image couleur. Les images de texture contenant les données MNT cachées peuvent ensuite être envoyées du serveur au client afin d'effectuer une visualisation 3D de terrains. Afin de combiner une visualisation en multirésolution et une compression, l'insertion des données cachées est intégrable dans le codeur JPEG 2000.

Mots clefs

Visualisation 3D, insertion de données cachées, compression JPEG2000, modèle numérique de terrains.

1 Introduction

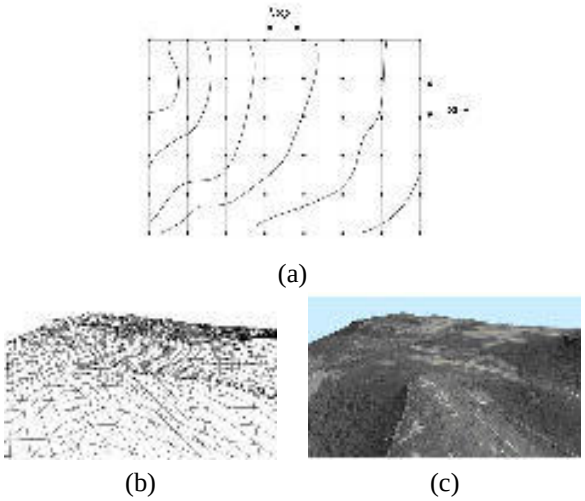
Les Systèmes d'Information Géographique (SIG) sont des outils appréciés dans le domaine de l'aide à la décision dans de nombreuses entreprises et institutions. Ces SIG permettent en effet de combiner des données textuelles, des données vectorielles et des images (rasters). L'utilisation de données comme des photographies aériennes, des cartes scannées ou de modèles numériques de terrains implique de mettre en place des stratégies de stockage et de visualisation de ces données. En particulier, il devient difficile de stocker toutes ces données sur chaque ordinateur, surtout si celui-ci est un média de faible capacité comme par exemple un pocket PC. De plus, ce problème de stockage est amplifié par l'évolution des capteurs qui permettent d'obtenir des images de meilleure qualité et donc de taille de plus en plus importante. Par exemple, il est actuellement possible de disposer d'images dont la précision au sol est inférieure au mètre. Cependant, ces données doivent pouvoir être transférées d'un serveur vers l'application utilisatrice. Pour le département des Bouches du Rhônes par exemple, les photographies aériennes représentent plus de six Giga Octets d'information comprimée par JPEG 2000[1]. De plus, la visualisation en trois dimensions d'un terrain implique de lier les images du sol, appelées textures, avec la géométrie du terrain, appelée modèle numérique de terrain (MNT). Ce lien est possible grâce aux coordonnées géo-référencées de ces éléments (longitude / latitude) qui dépendent du système de projection utilisé. Toutes ces informations sont en général stockées dans trois fichiers différents. L'objectif de nos travaux est de stocker et de synchroniser toutes ces informations dans un seul fichier afin de développer une application client-serveur de visualisation $3D$ temps-réel. A partir de cette approche il est possible de ne transmettre qu'un seul fichier regroupant toutes les informations et dépendant seulement de la zone où l'on se trouve. En fonction des débits de transmission et du point de vue de la visualisation 3D, un niveau de détail à transmettre est sélectionné. Afin de stocker toutes les informations dans un seul fichier, sans avoir à développer un nouveau format propriétaire, et de garder des performances d'un point de vue compression, nous proposons dans cet article d'utiliser des méthodes d'insertions de données cachées dans des images basées ondelettes. Ces travaux sont basés sur une version précédente d'insertion de données basée DCT [1]. Cet article est organisé de la manière suivante. Dans la sec-

[1] L'Institut Géographique National (IGN) fournit les données raster au format JPEG 2000 ou au format Tiff

tion 2 nous introduisons les modèles numériques de terrains et leur visualisation 3D. Section 3, nous décrivons la décomposition en ondelettes d'une image et nous proposons une méthode d'insertion de données cachées. Enfin, section 4, nous présentons notre méthode sur des données réelles et analysons les résultats obtenus.

2 Représentation 3D des terrains

La représentation de terrains en trois dimensions est une composante importante pour la mise en place d'environnements graphiques extérieurs virtuels. Nous pouvons, par exemple, citer les simulateurs de vols et de conduite et de façon plus générale les applications multi-joueurs. Dans le cadre de nos travaux, nous nous intéressons à la visualisation 3D temps-réel de terrains. Ceci nous amène à combiner deux types de données [2]. Nous utilisons tout d'abord un champ d'altitudes. Chaque altitude correspond à une élévation (Figure 1.a). Celles-ci sont utilisées afin de générer la géométrie du terrain, en les connectant par exemple sous la forme de triangles (Figure 1.b). Nous devons ensuite plaquer une image sur ces triangles afin d'obtenir la visualisation voulue (Figure 1.c).



(a) (b) (c)

Figure 1 – *a) Visualisation de la grille uniforme correspondante aux élévations de terrain, b) Surface 3D triangulée reliant les élévations, c) Plaquage d'une texture sur la géométrie.*

2.1 Le modèle numérique de terrain (MNT)

Afin de représenter les altitudes correspondant à chaque point d'un terrain en trois dimensions, un champ d'élévation est utilisé. Celui-ci peut par exemple être représenté à l'aide d'une grille régulière dans laquelle on dispose d'une altitude tous les 50 mètres (Figure 1.a). Les points sont ensuite reliés afin de former des triangles que l'on pourra ensuite afficher. Le nombre de triangles à afficher est très important. Par exemple, plus de 10 millions de triangles sont nécessaires pour le département des Bouches du Rhône (environ $13000\ Km^2$).

De nombreuses méthodes ont été proposées afin de réduire le nombre de triangles fournis par une discrétisation uniforme, tout en préservant une bonne approximation de la surface initiale. Une première approche consiste à utiliser un ensemble de triangles irréguliers (TIN : Triangulated Irregular Network). Un grand nombre de méthodes a été développé afin de créer cet ensemble de triangles, comme par exemple la triangulation de Delaunay [3]. Des représentations hiérarchiques de ces triangulations ont été proposées afin d'obtenir le concept de niveau de détails [4]. Il est alors possible de diminuer le nombre de triangles tout en obtenant une précision identique. Une autre approche consiste à décomposer le terrain en un ensemble de grilles régulières de niveaux de détails différents [5].

2.2 Application de la texture

Une fois la construction géométrique effectuée, il faut plaquer sur les triangles obtenus grâce aux élévations des images des texture. La précision liée à de telles images est en général d'un pixel pour $50\ cm$ sur le terrain, ce qui induit un coût important lié au stockage et au transfert des images vers le client. Il est donc nécessaire de penser à des stratégies de compression, de stockage et de visualisation rapide de ces données. De nombreuses méthodes de compression des images existent, les plus performantes, comme JPEG, induisent une détérioration de la qualité de l'image si le taux de compression est important. Plusieurs architectures client-serveurs ont été proposées afin de stocker un volume important de données utilisées sur plusieurs applications clientes [6]. Afin de visualiser rapidement ces données, une stratégie consiste à couper l'ensemble des images en une grille régulière [7] elle même décomposée en niveaux de détails [8].

3 Insertion de données cachées basée ondelettes

Dans cet article, nous proposons de tirer partie de la compression en ondelettes de JPEG 2000. L'implémentation choisie sera basée sur la méthode de Lifting [9, 10]. Deux types d'ondelettes sont utilisées dans ce travail, une avec pertes (Daubechies $9/7$) et une sans perte (Daubechies $5/3$) [11]. Notre objectif est de visualiser, sur une application cliente, un terrain défini par des élévations ainsi que la texture correspondante. Ces données sont stockées sur un serveur distant. Les données sont envoyées en paquets de petites tailles afin de minimiser les temps d'attentes. Les données que nous utilisons sont en général stockées par l'IGN dans trois fichiers différents : le MNT, la texture et le système de coordonnées employé. Afin de stocker ces informations dans un fichier unique nous proposons d'insérer le MNT, ainsi que les coordonnées géo-référencées dans l'image de texture. Une information d'altitude est alors synchronisé avec un bloc de pixels de la texture.
Nous proposons d'utiliser le format JPEG 2000 afin de profiter pleinement des propriétés de niveaux de détails sous-jascentes. De nombreuses méthodes ont été proposées

pour l'insertion de données cachées basée ondelettes, mais peu sont compatibles avec JPEG2000. Selon [12] pour l'insertion de données, les blocs de code doivent être traités séparément, c'est pourquoi des méthodes proposent des insertions inter-sous-bande [13] ou en multi-résolution hiérarchique [14]. D'autres approches proposent d'insérer les données cachées au niveau du flux binaire comprimé [15].
Afin d'insérer les informations d'altitude dans la carte de texture, nous proposons de suivre le protocole illustré figure 2. A partir de l'image de texture de N^2 pixels et de la carte de m^2 altitudes, nous en déduisons le facteur d'insertion $E = m^2/N^2$ coefficients/pixel. L'image de texture devra donc être découpée en bloc carré de $[1/E]$ pixels. Dans chacun de ces blocs, un coefficient d'altitude est donc caché. Dans un premier temps, la carte de texture est transformée en composantes Y, Cr, Cb. Une transformée en ondelettes discrètes (TOD) est appliquée sur les trois plans obtenus ainsi que sur la carte d'altitude. Nous choisissons d'utiliser une décomposition sans perte sur cette dernière afin de ne pas altérer les valeurs des altitudes.

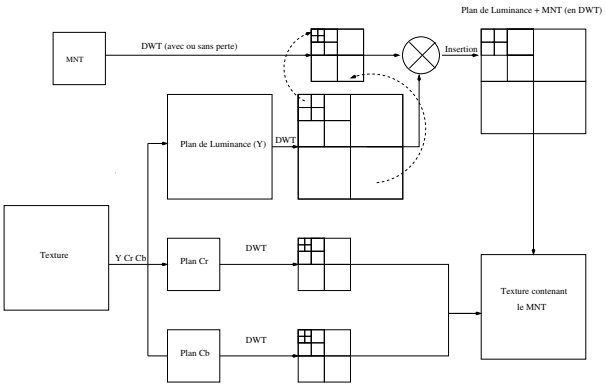


Figure 2 – *Description de la méthode d'insertion des altitudes dans la carte de texture.*

Afin d'assurer une cohérence spatiale entre les altitudes et la carte de texture, nous décomposons le plan de luminance Y, dans le domaine des ondelettes en bloc carré de $[1/E]$ coefficients quel que soit le niveau de décomposition. Dans chaque bloc nous insérons une information d'altitude du même niveau de décomposition. Nous retrouvons donc une mise en correspondance sur l'insertion dans les niveaux d'ondelettes. Par exemple, les basses résolutions de la carte d'altitude sont insérées dans les basses résolutions de la carte de texture. De ce fait, la transmission de la partie basse résolution de la carte de texture permet ainsi d'accéder directement à la partie basse résolution de la carte d'altitude. L'insertion des données se fait en modifiant les bits de poids faibles d'un certain nombre de coefficients du plan de luminance de la carte de texture. Ces coefficients sont choisis avec un générateur de nombres pseudo-aléatoire. A la réception le MNT est extrait de la carte de texture, même si seulement qu'une partie de l'image de texture a été transmise.

4 Résultats

Nous avons appliqué notre méthode sur une carte de texture des bouches du rhône de taille 2048×2048 pixels, illustrée figure 3.a, en y associant la carte d'altitude de 64×64 coefficients, illustrée figure 3.c. Un détail de 128×128 pixels de la carte de texture est représenté figure 3.b. Chaque coefficient d'altitude est codé sur 2 octets et le facteur d'insertion est donc de 1 coefficient pour 32×32 pixels de texture.

(a)

(b)

(c)

Figure 3 – *Données originales : a) Texture, b) Détail texture, c) Altitudes.*

En effectuant des décompositions en ondelettes avec pertes pour la carte de texture et sans perte par la carte d'altitudes nous obtenons les résultats illustrés figures 4 pour le niveau 1, et figures 5 pour le niveau 3 de décomposition. Pour la décomposition en ondelettes en un seul niveau, figure 4.a, l'insertion de données cachées se fera en quatre temps (LL, LH, HL et HH) alors que pour la décomposition

(a)

(b)

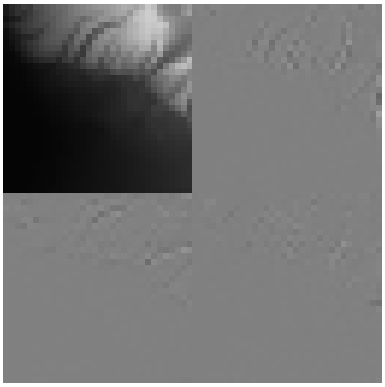


Figure 4 – *TOD, niveau 1 : a) Texture, b) Altitudes.*

(a)

(b)

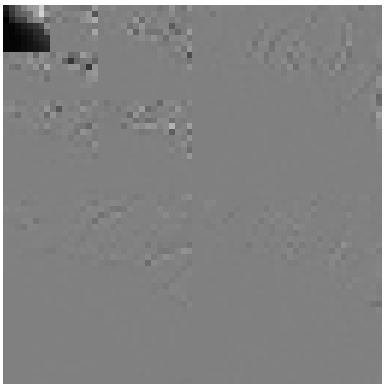


Figure 5 – *TOD, niveau 3 : a) Texture, b) Altitudes.*

en ondelettes en trois niveaux, figure 5.a, l'insertion de données cachées sera réalisée en dix étapes. Quel que soit le niveau de décomposition, la différence entre les images de luminance avant l'insertion des données et après l'insertion nous amène à un PSNR de $69.20\ dB$ puisque nous insérons 16 bits d'un coefficient d'altitude dans 32×32 coefficients de texture sur la composante Y.

(a)

(b)

(c)

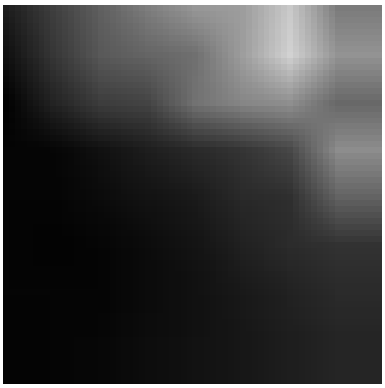
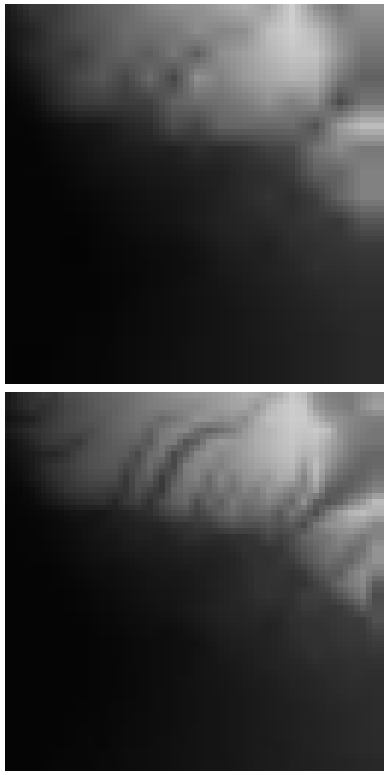


Figure 6 – *Reconstruction avec l'image d'approximation du niveau 3 : a) Texture, b) Détail texture, b) Altitudes extraites.*

A partir de l'image de textures décomposée jusqu'au troisième niveau et marquée avec les altitudes, si nous effectuons une TOD inverse uniquement avec l'image d'approximation du niveau 3, nous obtenons la carte de texture illustrée figure 6.a. En effectuant la différence entre cette carte de texture reconstruite et la carte de texture originale nous obtenons un PSNR de $20.90\ dB$. Un détail de la carte de texture reconstruite (identique à celui extrait figure 3.b)

est illustrée figure 6.b. A partir de l'image d'approximation du niveau 3, si nous effectuons l'extraction des données cachées suivie d'une TOD inverse, nous obtenons la carte d'altitudes illustrée figure 6.c. Le PSNR entre la carte d altitude originale et celle reconstruite est alors de $29.25\ dB$. Notons qu'en utilisant que l'image d'approximation du niveau 3 nous n'avons eu besoin que de $1,6\%$ des données initiales.



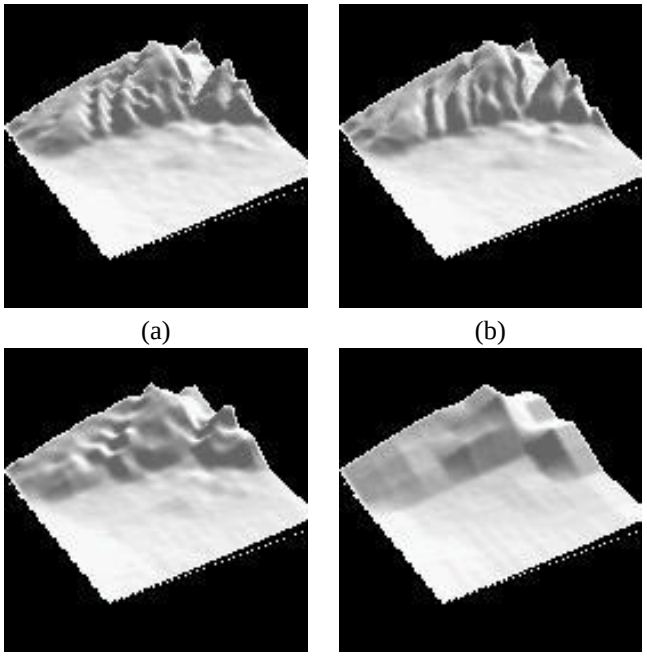
(a) (b)

Figure 7 – *Reconstruction de la carte d'altitudes : a) Avec le niveau 2, b) Avec le niveau 1.*

	Niv. 3	Niv. 2	Niv. 1.	Tout
% données transmises	1.6 %	6.25 %	25 %	100 %
Texture (dB)	20.90	22.79	26.54	37.62
Altitude (dB)	29.25	33.51	40.37	∞
EQM Altitude (m^2)	77.37	29.00	5.97	0

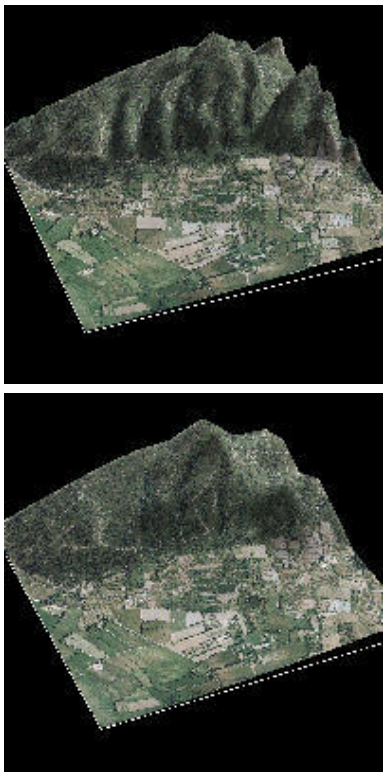
Tableau 1 – *Résultats obtenus pour l'extraction et la reconstruction en fonction de la quantité de données utilisée.*

Le tableau 1 résume les résultats obtenus pour une reconstruction à partir des images d'approximation des niveaux 3, 2 et 1 ou en utilisant toutes les données. Notons que si nous utilisons toutes les données pour la TOD inverse nous obtenons un PSNR de $37.62\ dB$ pour la carte de texture et un PSNR infini pour la carte d'altitude puisque nous avons utilisé une TOD sans perte. Les figures 7.a et b illustrent les cartes d'altitudes reconstruites respectivement avec les images d'approximation des niveaux 2 et 1.



(a) (b) (c) (d)

Figure 8 – *Visualisation 3D des altitudes : a) Avec toutes les données, b) Avec le niveau 1, c) Avec le niveau 2, d) Avec le niveau 3.*



(a) (b)

Figure 9 – *Navigation 3D de la zone : a) Avec toutes les données, b) Avec le niveau 3.*

Les figures 8.a, b, c et d représentent la reconstruction 3D des MNT, respectivement avec toutes les données initiales, qu'avec le niveau 1, qu'avec le niveau 2, et qu'avec le niveau 3. En plaquant la texture sur le MNT les figures 9.a et b permettent de comparer le résultat final entre une visualisation avec toutes les données et une visualisation uniquement avec le niveau 3 composé de 1.6 % des données initiales.

5 Conclusion

Dans cet article, nous avons présenté une nouvelle méthode d'insertion de données permettant de dissimuler un MNT dans l'orthophotographie qui lui est liée. Dans le cadre de l'application client-serveur que nous avons mise en place, nous pouvons ainsi synchroniser ces informations, évitant ainsi toute erreur liée à une combinaison erronée des données ou à un manque dû à leur transfert. De plus, l'utilisation d'images compressées sous la forme d'ondelettes permet de transférer le ou les niveaux nécessaires à une visualisation optimale en fonction du point de vue choisi par l'utilisateur, du débit et du média utilisé (ordinateur, pocket PC, ...) ou du niveau de détail choisi. De plus la méthode d'insertion des données développée est intégrable au codeur JPEG2000, de ce fait les images de texture peuvent être visualisées avec tous les lecteurs d'images JPEG 2000.

Bien que les résultats obtenus nous paraissent déjà intéressants, nous souhaiterions mettre en place, dans la suite de nos travaux, un stockage plus fin du modèle numérique de terrain. Nous pourrons par exemple nous intéresser aux ondelettes géométriques. Nous étudierons aussi la possibilité de ne plus utiliser des grilles uniformes à différents niveaux de détails, permettant ainsi de diminuer le nombre de triangle nécessaires à une bonne représentation du terrain si la variation de terrain est peu importante.

Références

[1] A. Martin, G. Gesquiere, W. Puech, et S. Thon. Real Time 3D Visualisation of DEM Combined with a Robust DCT Based Data-Hiding Method. Dans *Electronic Imaging, Visualization and Data Analysis, SPIE, IS&T, San Jose, CA, USA*, volume 6060, pages 60600G–1–60600G–8, Jan 2006.

[2] R. Raffin S. Thon, G. Gesquière. Visualisation 3D de feux de forêts sur des modèles numériques de terrain de l'IGN. Dans *Rencontre GEO-RISQUE 2006 "La cartographie des risques naturels"*, Montpellier (France), février 2006.

[3] R. J. Fowler et J. J. Little. Automatic Extraction of Irregular Network Digital Terrain Models. Dans *SIGGRAPH '79 : Proceedings of the 6th annual conference on Computer graphics and interactive techniques*, pages 199–207, New York, NY, USA, 1979. ACM Press.

[4] L. De Floriani et E. Puppo. Hierarchical Triangulation for Multiresolution Surface Description. *ACM Trans. Graph.*, 14(4) :363–411, 1995.

[5] F. Losasso et H. Hoppe. Geometry Clipmaps : Terrain Rendering Using Nested Regular Grids. *ACM Trans. Graph.*, 23(3) :769–776, 2004.

[6] P. Baumann, P. Furtado, R. Ritsch, et N. Widmann. Geo/Environmental and Medical Data Management in the RasDaMan System. Dans M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, et M. A. Jeusfeld, éditeurs, *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, pages 548–552. Morgan Kaufmann, 1997.

[7] J. Yu et D. J. DeWitt. Processing Satellite Images on Tertiary Storage : A Study of the Impact of Tile Size on Performance. Dans *5 th NASA Goddard Conference on Mass Storage Systems and Technologies*, 1996.

[8] P. Hansen. OpenGL Texture-Mapping with Very Large Datasets and Multi-Resolution Tiles. Dans *SIGGRAPH '99 : ACM SIGGRAPH 99 Conference abstracts and applications*, page 262, New York, NY, USA, 1999. ACM Press.

[9] W. Sweldens. The Lifting Scheme : a New Philosophy in Biorthogonal Wavelet Constructions. Dans *Electronic Imaging, Wavelet Applications in Signal and Image Processing, SPIE, IS&T, San Diego, CA, USA*, volume 2569, pages 68–79, Sep. 1995.

[10] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[11] I. Daubechies et W. Sweldens. Factoring Wavelet Transforms into Lifting Steps. *Fourier Anal. Appl.*, 4(3), 1998.

[12] P. Meerwald et A. Uhl. A Survey of Wavelet-Domain Watermarking Algorithms. Dans *Electronic Imaging, Security and Watermarking of Multimedia Contents, SPIE, IS&T, San Jose, CA, USA*, Jan 2001.

[13] D. Kundur. Improved Digital Watermarking Through Diversity and Attack Characterization. Dans *ACM Workshop on Multimedia Security'99, Orlando, FL, USA*, pages 53–58, Oct 1999.

[14] D. Kundur et D. Hatzinakos. Digital Watermarking Using Multiresolution Wavelet Decomposition. Dans *International Conference on Acoustic, Speech and Signal Processing (IEEE ICASP 98),Seattle, Washington, USA*, May 1998.

[15] P.-C. Su, H.-J. Wang, et C.-C. J. Kuo. An Integrated Approach to Image Watermarking and JPEG-2000 Compression. *Journal of VLSI Signal Processing Systems, Special issue on multimedia signal processing*, 27(1-2) :35–53, June 1997.

Modélisation 3D d'un objet par un capteur stéréo monté sur un manipulateur mobile

F. Trujillo-Romero M. Devy

LAAS groupe Robotique et Intelligence Artificiel

LAAS, Av du Colonel Roche, 31077 Toulouse Cedex 04

{ftrujill, michel}@laas.fr

Résumé

Ce rapport montre des résultats obtenus sur la modélisation incrémentale d'un objet 3D de forme quelconque, en utilisant un capteur stéréo monté sur un bras manipulateur, lui-même monté sur un robot mobile. Une tel processus nécessite les étapes classiques de recalage entre vues acquises pendant le déplacement du capteur autour de l'objet à modéliser, puis de construction d'un maillage depuis le nuage de points 3D acquis depuis toutes les vues. Nous présentons surtout une technique alternative aux algorithmes de type Marching Cubes exploités pour construire le maillage. Cette technique exploite une paramétrisation sphérique des points 3D afin de construire le maillage d'une façon rapide et efficace, puisque c'est plus facile d'obtenir le maillage d'une sphère que d'un objet dont nous ne connaissons pas la forme.

Mots clefs

Modèle 3D, Maillage, Stéréovision, Recalage, Fusion.

1 Introduction

La modélisation d'objets 3D peut été traitée par de nombreuses méthodes, exploitant divers capteurs pour acquérir des points 3D sur la surface de l'objet. Les progrès technologiques très rapides en ce domaine, donnent accès à des capteurs 3D qui ont chaque fois une sensibilité et une résolution plus grandes que celles qui peuvent être obtenues avec une simple paire de caméras, utilisée dans un banc stéréo. Ces nouveaux capteurs se fondent sur la télémétrie-laser, sur un couple caméra-illuminateur ou sur la mesure du temps de vol (Swiss Ranger) ; ils permettent de modéliser des objets qui ont des surfaces non texturées. Mais ces techniques ont aussi quelques inconvénients ; ils peuvent être encombrants, chers et lents (télémétrie avec balayage deux axes), peuvent être trop fragiles pour être montés sur un manipulateur (Swiss Ranger), ou encore nécessiter un calibrage délicat (caméra avec illuminateur). Par ailleurs une meilleure résolution implique une grande quantité de points, ce qui rend plus complexes et lentes, les méthodes de construction de maillage. Or, nous participons au projet européen *Cogniron*, qui étudie le concept du robot assistant de l'homme. Il a pour but entre autres, d'intégrer sur un robot, des fonctions permettant de modéliser un objet inconnu a priori, puis de le saisir pour le donner à un homme ; dans ce contexte, les méthodes de reconstruction 3D doivent être exécutée en ligne.

Il existe plusieurs travaux dans le domaine de la modélisation 3D. Nous nous sommes inspirés des travaux de Floater [1] qui paramétrise une maille 3D vers une maille 2D sans perte d'information. Bien que la paramétrisation de Floater, comme toutes paramétrisations, exploite une maille déjà construite, cette méthode de reconstruction permet de réduire ou augmenter la quantité de polygones. Gu et al. [2] ont décrit comment comprimer un maillage triangulaire en utilisant la méthode des images géométriques. La méthode que nous utilisons pour modéliser un objet, se base en partie sur la paramétrisation sphérique et sur les travaux de Gu et al. [3], qui ont comparé différentes méthodes pour effectuer une telle paramétrisation. Hormman et Greiner [4] proposent une alternative à la paramétrisation, en exploitant un plan à la place d'une sphère ; cette représentation présente certaines caractéristiques très intéressantes pour traiter de la reconstruction d'objets 3D en temps réel.

La section 2 présente plusieurs méthodes proposées pour la construction de maillage 3D. La section 3 décrit nos propres travaux sur la modélisation, tandis que la section 4 présente quelques résultats. Enfin la section 5 résume l'état actuel de nos travaux et donne quelques perspectives.

2 Problématique : la construction d'un maillage 3D

Une fois qu'un nuage de points est obtenu par recalage de plusieurs vues, il est très difficile de construire un maillage, et cela d'autant plus qu'il existe des contraintes temps réel. Plusieurs problèmes sont posés : (1) les relations topologiques entre points (voisinage) sont perdues après recalage et aggrégation des images de points 3D dans un nuage, (2) les points présentent un bruit non uniforme, (3) la résolution des points est variable dans le nuage et (4) l'objet à modéliser peut avoir une géométrie complexe (concavités, trous...) et peut comporter plusieurs parties non connexes.

Le maillage doit avoir une résolution variable,afin de lisser le bruit et de disposer de modèles multi-résolution.
Citons d'abord les techniques de décimation, exploitées pour réduire le nombre de triangles dans un maillage existant (Garland, Hoppe, Rossignac...). La décimation décrite par Schroeder et al.[5] permet de réduire le nombre de polygones, y compris sur des maillage de très grande taille. Malheureusement ces méthodes ne peuvent pas s'appliquer à un nuage de points 3D, non encore représenté par un maillage. Par conséquent nous devons employer une technique alternative comme celle mentionnée dans le travail développé par Gao et Lu [6].
D'autre part il existe plusieurs méthodes pour construire un maillage depuis un nuage de points. La plus célèbre est la méthode des *Marching Cubes* [7], développé par Lorensen et Cline. C'est un algorithme rapide mais qui a l'inconvénient de laisser les trous dans un maillage là où la résolution du nuage est trop faible. Il existe des améliorations et des variantes de cet algorithme. Citons les *Marching triangles* [8] et Les *Dual Marching Cubes* [9].
A.Restrepo [10] a montré que l'algorithme de *Ball Pivoting* développé par Bernardini et al. [11] est meilleur que *Marching Cubes* mais qu'il a l'inconvénient d'être très lent. Un autre algorithme très classique pour modéliser des objets 3D de forme quelconque ou déformable (en particulier, applications médicales : coeur... Voir un exemple dans [12]) exploite le concept de *Surface déformable*. Cet algorithme part d'un maillage initial, généralement généré par discrétisation d'une sphère, qui doit être déformé itérativement afin de suivre au mieux les points de mesure. La modélisation obtenue est souvent meilleure qu'avec d'autres méthodes, mais avec l'inconvénient que le réglage des paramètres est très délicat et très dépendant de la forme de l'objet à modéliser.
Nous pouvons finalement citer les techniques qui utilisent la triangulation Delaunay commedans les travaux de Fang et Piegl [13]. Ils présentent un algorithme qui est rapide et qui fournit de bons résultats. C'est dans cette dernière catégorie que nous pouvons inclure notre travail, vu que c'est la plus simple à mettre en oeuvre et c'est aussi la plus rapide pour une exécution en temps réel.

3 Nos développements sur la modélisation 3D

3.1 Système utilisé

La figure 1 montre le robot utilisé pour valider notre travail. Ce robot est formé par une plateforme mobile Neobotix sur laquelle est monté un bras robotique PA10-6C développé par Mitsubishi. Ce bras a six degrés de liberté. L'organe terminal est doté d'un capteur d'effort et d'une pince (constituée de trois doigts) qui servira à prendre des objets à partir d'informations visuelles. Ces informations sont acquises depuis une paire stéréo exploitée dans notre travail, pour modéliser l'objet (supposé inconnu) avant de chercher une position de prise pour la pince, et de contrôler la saisie. Le robot JIDO dispose d'autres capteurs montés sur un mât derrière le bras : nous n'utilisons pas ces capteurs pour l'heure.

Figure 1 – *Bras Robotique PA10-6C : JIDO*

Le processus de modélisation se divise en trois fonctions principales : l'acquisition des images, le recalage et la construction du maillage. Dans les sections suivantes nous développerons chacune de ces étapes, qui s'exécutent soit itérativement (recalage et mise-à-jour incrémentale du maillage après chaque acquisition) ou en séquence (acquisition de N images, recalage de ces images et fusion dans un nuage de points 3D, construction du maillage à partir de ce nuage).

3.2 Acquisition des images sur l'objet

La phase d'acquisition des images est démarrée quand le système détecte dans l'environnement, un objet à modéliser. Dans le scénario complet, cette détection nécessite d'abord que le robot localise une table, s'en approche, acquérir une image du plateau et y détecte la présence d'un objet. A partir des points 3D calculés depuis cette image, le centre de gravité P des points perçus sur l'objet est grossièrement localisé sur la table. Ce point P permet de calculer la position d'accostage de la plateforme autour de la table, et les positions du bras pour placer le capteur stéréo et acquérir des images complémentaires sur l'objet.
A partir d'une demi-sphère centrée sur le point P, nous calculons 6 positions différentes pour pouvoir modéliser l'objet. Nous pourrions prendre beaucoup plus d'images mais il est suffisant d'acquérir les vues principales de l'objet, et surtout, de cette manière, le système est plus rapide. On calcule ces positions du capteur stéréo sur la demi-sphère de sorte que les positions correspondantes de l'organe terminal du bras, soient atteignables : pour cette raison, on ne peut acquérir des images sur la partie arrière de l'objet. Dans une extension future, le robot JIDO pourra accoster la table en plusieurs positions afin de voir l'objet sous tous ses points de vue.

3.3 Recalage des images 3D dans un nuage de points 3D

Après chaque acquisition i, le module stéréo est activé pour générer une image I_i de points 3D, reconstruits dans le repère du capteur stéréo. Pour aggréger cette image I_i dans le nuage de points 3D, exprimée dans le repère du robot

JIDO, nous devons calculer la transformation T_i entre le repère stéréo (position du point de vue courant sur l'objet) et le repère JIDO. Une estimée initiale T_i^0 de cette transformation T_i est obtenue en composant plusieurs transformations rigides (position du repère stéréo par rapport au repère Organe terminal, position de la base du bras sur JIDO) avec la transformation qui donne la position de l'organe terminal par rapport à la base du bras, estimée par les codeurs du bras.
Cette estimée T_i^0 n'est pas suffisament précise pour garantir une bonne modélisation. Il faut appliquer la méthode *Iterative Closest Points* (ICP) [14] afin d'améliorer itérativement cette transformation, pour à la fin, obtenir un nuage de points cohérent. Ce processus ICP n'est pas trivial, car il faut choisir une stratégie (recalage incrémental de I_i evec I_{i-1} ou recalage de I_i avec le nuage en cours de formation), il faut conserver uniquement les bons points à apparier dans l'image I_i et les précédentes. Nous utilisons la méthode ICP implementée par A.Restrepo [10]. La convergence n'est pas garantie, mais nous avons vérifié que la précision finale est suffisante.

3.4 Construction du maillage

A partir du nuage de points obtenu après recalage, le maillage est construit en utilisant le processus de paramétrisation sphèrique, qui consiste à projeter tous les points du nuage vers une sphère (en fait une demi-sphère vu que l'objet est posé sur une table). Il n'existait pas de méthode de ce type, car la plupart des travaux existants (voir les sections 1 et 2) se basent sur l'existence d'un maillage initial de triangles ou de polygones, qu'il convient ensuite de décimer.
Notre mise en oeuvre part de l'équation d'une sphère que possède un centre $\vec{C}$ et un rayon R. Le centre de la sphère est dans le centre de gravité du nuage de points et le rayon est la distance qui existe du $\vec{C}$ vers le point le plus éloigné de la nuage plus un petit accroissement ε. Avec ceci nous arrivons à enfermer complètement le nuage. Dans l'équation 1 on peut voir cette relation

$$R = max(x, y, z) + \varepsilon \tag{1}$$

Une fois déterminé le centre et le rayon de la sphère on projette chaque un des points du nuage vers la sphère. Celuici, c'est le processus principal de notre travail dû au fait que s'il n'est pas bien calculé peuvent apparaître effets non souhaités. Comme dans projeter plusieurs points du nuage dans un seul point de la sphère. Il convient de clarifier que doit exister un seul point dans la sphère par un point dans le nuage 3D.
Ceci est obtenu de la manière suivante :
Soit une droite $\vec{L}$ qui passe par le centre $\vec{C}$ et par un point $\vec{N}$ du nuage à triangulaire. On calcule la plus petite distance entre le point $\vec{N}$ et la sphère.
Pour effectuer ce calcul il est nécessaire de trouver l'intersection de la droite $\vec{L}$ et la sphère définie par $\vec{C}$ et R.
Nous considérons d'abord la parametrization de la droite $\vec{L}$ comme

$$\vec{L}(t) = \vec{N} + t\vec{D} \tag{2}$$

où

$$\vec{D} = \vec{N} - \vec{C} \tag{3}$$

et la parametrization de la sphère comme

$$|\vec{L} - \vec{C}|^2 = R^2 \tag{4}$$

En remplaçant l'équation 1 de la droite $\vec{L}$ dans l'équation 4 nous allons obtenir la valeur de t. Lequel doit être positif pour l'intersection de la droite et la sphère.
Avec ceci nous calculons le point dans la sphère comme l'indique l'équation 1 et nous obtenons la projection de $\vec{n}$ dans la sphère.
On effectue la triangulation en utilisant l'algorithme de convex hull. Grâce au fait que tous les points du nuage sont dans la sphère il est possible d'obtenir une surface uniforme.
Une fois triangulaire la sphère nous obtenons les relations des triangles formés pour pouvoir produire le modèle final de l'objet qui est reconstruit.
En bref, c'est l'équation 1 qui est exploitée pour projeter un point de l'objet sur la sphère, en faisant l'hypothèse que la droite reliant le centre de la sphère au point 3D, coupe la sphère en un seul point. Nous calculons cette intersection pour obtenir la paramétrisation sphérique souhaitée. Le maillage peut être construit facilement sur la sphère, et peut être décimé par une méthode classique (Hoppe ou Garland) pour obtenir un maillage avec une résolution adaptée aux traitements suivants (pour nous, recherche des positions de prise pour la pince) et à la complexité de la forme de l'objet.

4 Résultats

Cette section présente plusieurs résultats ainsi que la comparaison de notre méthode avec deux autres algorithmes de reconstruction 3D, *Ball Pivoting* [11] et *Marching Cubes* [7].
Les figures 2 à 5 montrent les résultats obtenus lorsque nous avons appliqué les différents algorithmes pour modéliser une tête de mannequin. Dans la figure 2 nous pouvons observer le résultat de la paramétrisation sphérique appliquée au nuage de points acquis sur le mannequin. Ensuite nous présentons la reconstruction du mannequin avec la paramétrisation sphérique en figure 3, avec *Marching Cubes* en figure 4 et *Ball pivoting* en figure 5.
Il existe beaucoup de différences entre les méthodes, sur la qualité du maillage obtenu et sur les temps d'exécution. Malgré la rapidité d'exécution de *Marching Cubes*, la qualité du modèle final n'est pas bon. En outre, notre algorithme a un temps d'exécution sensiblement équivalent à celui de *Marching cubes*, mais le résultat final est meilleur.

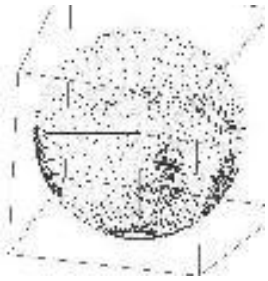


Figure 2 – *Parametrisation sphérique du mannequin*

Figure 3 – *Reconstruction par la paramétrisation sphérique*

Figure 6 – *Paramétrisation sphérique de la boîte*

Figure 7 – *Reconstruction par la paramétrisation sphérique*

Figure 4 – *Reconstruction par Marching cubes*

Figure 5 – *Reconstruction par Ball-Pivoting*

Figure 8 – *Reconstruction par Marching cubes*

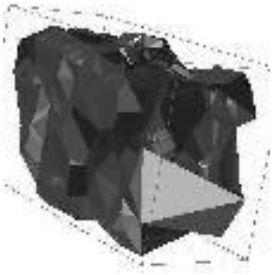


Figure 9 – *Reconstruction par Ball-Pivoting*

Les figures 6 à 9 présentent les résultats obtenus lorsque nous avons appliqué les différents algorithmes de reconstruction décrits dans la section précédente, à un objet réel, en ce cas une boîte parallélépipédique. La figure 6 montre le résultat de la paramétrisation sphérique du nuage de points. Nous présentons la reconstruction de cette boîte, obtenue avec l'algorithme que nous proposons en figure 7, avec *Marching cubes* en figure 8 et avec *Ball pivoting* en figure 9.

Objet	M.C.	B.P.	P.S.
Mannequin	3.82 s.	9.13 m.	5.87 s.
Boîte	2.51 s.	6.32 m.	3.74 s.

Tableau 1 – *Temps de reconstruction*

Le tableau 1 montre le temps d'exécution des algorithmes avec les deux modèles utilisés : nous pouvons voir que le compromis qualité/temps est meilleur dans notre algorithme qu'avec *Marching Cubes* ou *Ball Pivoting*. Ce dernier est très complexe, et nécessite des temps d'exécution incompatibles avec un traitement en temps réel.

5 Conclusions et perspectives

Nous avons présenté une méthode alternative pour reconstruire des objets 3D de forme libre à partir de nuages de points acquis par stéréovision. Cette méthode se fonde sur une paramétrisation sphérique. Cet algorithme est non seulement fonctionnel pour effectuer la triangulation d'un nuage de points 3D, mais il est en plus très rapide. La comparaison de cette méthode avec *Marching Cubes* et *Ball Pivoting* a montré que notre méthode présentait plus d'avantage que d'inconvénients.

Il convient de mentionner que les objets à modéliser doivent être convexes, ce qui est le cas de nombreux objets que nous utilisons tous les jours dans notre maison ou dans le travail. Néammoins, une amélioration en cours est l'extension de cette méthode pour l'appliquer à des objets concaves. Cette mise en oeuvre sera d'une grande utilité pour pouvoir modéliser tout type d'objet.

A court terme, nous allons participer à une expérimentation collective, qui doit valider les capacités du robot JIDO pour : (1) détecter un objet inconnu dans son environnement, (2) le modéliser, (3) le saisir et (4) le tendre à un opérateur humain préalablement perçu.

A moyen terme, cette méthode sera exploitée pour créer une base de données d'objets 3D en exploitant pour cela la méthodologie d'images géométriques développée par Gu et al. [2], présentée aussi par Hoppe et al. [15] pour obtenir la représentation en 2D d'un objet 3D. Pour pouvoir effectuer la reconnaissance d'un objet par classification à partir de ses caractéristiques, il est nécessaire d'effectuer cette paramétrisation.

Remerciements

Cette recherche a été menée grâce à l'appui financier de l'organisme Mexicain CONACYT et a été réalisée dans le cadre du projet Européen COGNIRON .

Références

[1] M. Floater. Parametrization and smooth approximation of surface triangulations. *CAGD*, 14(3) :231–

250, 1997.

[2] X. GU, S. Gortler, et H. Hoppe. Geometry images. *ACM Transaction on Graphics*, 21(3) :355–361, 2002.

[3] X. GU C. Gotsman et A. Sheffer. Fundamentals of spherical parametrization for 3d meshes. *ACM Transaction on Graphics*, 22(3) :358–363, July 2003.

[4] K. Hormman et G. Greiner. Mips : An efficient global parametrization method. *Curve and surface Desing*, pages 153–162, 2000.

[5] J. A. Zarge W. J. Schroeder et W. E. Lorensen. Decimation of triangle meshes. Dans *SIGGRAPH*, pages 65–70. ACM, 1992.

[6] S. Gao et H.-Q. Lu. A fast algorithm for delaunay based surface reconstruction. *The 11-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, February 2003.

[7] W. E. Lorensen et H. E. Cline. Marching cubes : A high resolution 3d surface construction algorithm. Dans *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, July 1987. ACM Press.

[8] S. Akkouche et E. Galin. Adaptive implicit surface polygonization using marching triangles. *Comput. Graph. Forum*, 20(2) :67–80, 2001.

[9] G. M. Nielson. Dual marching cubes. Dans *VIS '04 : Proceedings of the conference on Visualization '04*, pages 489–496, Washington, DC, USA, 2004. IEEE Computer Society.

[10] J.A. Restrepo Specht. *Modélisation d'objets 3D par construction incrémentale d'un maillage triangulaire, dans un contexte robotique*. Thèse de doctorat, LAAS, 7, Av Colonel Roche Toulouse France, Janvier 2005.

[11] F. Bernardini, J. Mittleman, H. E. Rushmeier, C. T. Silva, et G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Vis. Comput. Graph.*, 5(4) :349–359, 1999.

[12] C. Xu et J. L. Prince. Gradient vector flow : A new external force for snakes. Dans *Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 66–. IEEE Computer Society, 1997.

[13] T. P. Fang et L. A. Piegl. Delaunay triangulation in three dimensions. *IEEE Computer Graphics and Applications*, 15(5) :62–69, Sept. 1995.

[14] Y. Chen et G. G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3) :145–155, 1992.

[15] H. Hoppe. Overview of recent work on geometry images. Dans *Geometric Modeling and Processing*, page 12. IEEE Computer Society, 2004.

Suivi 3D à partir d'un modèle basé points

Christophe Dehais[1] Vincent Charvillat[1] Géraldine Morin[1]

[1] IRIT - ENSEEIHT

Site ENSEEIHT
2, rue Charles Camichel - BP4122 - 31071 Toulouse Cedex 7

{dehais, charvi, morin}@enseeiht.fr

Résumé

Cet article présente une déclinaison originale d'un des meilleurs algorithmes de suivi 3D temps réel issu de l'état de l'art. Il s'agit d'un algorithme de suivi basé sur un modèle 3D par facettes planes de l'objet à suivre. Notre contribution essentielle est de substituer ce modèle basé sur des facettes par un nouveau modèle basé sur des points 3D. Cette proposition nous conduit à reformuler le problème en faisant apparaitre plusieurs étapes d'estimation linéaire de paramètres. Ces linéarisations permettent une implantation moins complexe et moins coûteuse de l'algorithme initial. Des expériences montrent finalement que son efficacité est conservée.

Mots clefs

Suivi 3D, modèle basé points

1 Introduction

De nombreuses applications nécessitent le suivi 3D d'objets rigides (ou peu déformables). C'est le cas en particulier de la Réalité Augmentée [1], de la commande visuelle de robots [2] ou du suivi de visage [3].

Dans ces contextes, des contraintes de coût ou des difficultés d'instrumentation de l'environnement conduisent à la recherche de solutions exploitant la vison par ordinateur. Récemment les méthodes s'appuyant sur la connaissance d'un modèle 3D de l'objet d'intérêt et sur la détection et le suivi des caractéristiques naturelles de celui-ci (contours, points d'intérêts, textures) ont montré des résultats très satisfaisants [4, 5].

Les algorithmes proposés ont pour objectifs :

– l'efficacité algorithmique : les applications mentionnées plus haut doivent suivre l'objet à la cadence d'acquisition de la vidéo, entre 10 et 30 images/sec.
– la robustesse : dans la pratique, les données visuelles extraites des images et utilisées pour le suivi sont souvent imprécises et parfois aberrantes. Un algorithme de suivi robuste doit resister à ces contaminations.
– la précision : deux types de problèmes sont fréquemment rencontrés. L'accumulation d'erreurs au cours du suivi conduit d'une part à une dérive progressive du résultat par rapport à la solution recherchée. D'autre part, l'instabilité des paramètres retrouvés induit une cohérence temporelle insuffisante. Ce dernier problème est particulièrement gênant dans le cas de la Réalité Augmentée.

Vacchetti et al. [3] proposent un algorithme temps réel qui remplit ces exigences de robustesse et de précision. Ils s'appuient sur la connaissance a priori d'un modèle composé de facettes (maillage) de l'objet à suivre. Ceci permet de simplifier l'expression d'un problème d'ajustement de faisceaux à partir de mises en correspondance de points d'intérêt. Une faible dérive des paramètres et une bonne stabilité temporelle sont obtenues par l'utilisation conjointe d'un suivi itératif et d'un suivi par rapport à une image clé. Bien qu'une implantation en temps réel de cette technique soit possible [3], elle reste délicate à obtenir pour plusieurs raisons. D'abord les critères à optimiser sont non-linéaires, et présentent de nombreux minima-locaux, ce qui nécessite une initialisation suffisamment proche de la solution. Seules des méthodes d'optimisation itératives relativement couteuses sont utilisables. Ensuite l'environnement de mise en oeuvre fait appel à des calculs intermédiaires nombreux (en particulier pour associer des indices visuels 2D extraits des images aux "bonnes" facettes 3D du modèle). Ces difficultés d'implantations compromettent la viabilité de futures extensions de cette approche, en particulier pour traiter explicitement les déformations.

Par ailleurs, des modèles 3D qui ne sont plus nécessairement complets ou organisés topologiquement et qui incorporent des données texturelles en plus de la géométrie [6, 7] ont montré leur intérêt en vision :

– pour leur capacité en reconnaissance [8]
– pour le suivi avec des approches de type appariement de motif utilisant des modèles d'apparence. [6, 7]

Il faut remarquer que cette tendance est également observable en synthèse d'images où le rendu par points devient performant [9].

Suivant cet élan, nous proposons de bénéficier des avantages des modèles à base de points pour implanter une version simplifiée de l'algorithme présenté par Vacchetti et al. [3] (notre algorithme *de référence* pour cette étude).

Dans la section 2 nous présentons les éléments de modélisation du problème. Les détails algorithmiques de

notre approche sont présentés en section 3. Des expériences sur des données synthétiques et réelles sont enfin reportées en section 4. Nous comparons en particulier nos algorithmes à ceux de notre implantation de l'algorithme de référence.

2 Suivi d'un modèle 3D par facettes

2.1 Modélisation, résolution du problème

On définit l'objectif du suivi comme l'estimation des paramètres de la pose d'un objet par rapport à la caméra au cours d'une séquence d'image $\{I_t\}$.

Le modèle de caméra. La caméra perspective est modélisée par une matrice de projection P 3×4, qui peut se décomposer sous la forme :

$$P = K.[\, R \mid \mathbf{t} \,]$$

où K est la matrice triangulaire supérieure comprenant des paramètres *intrinsèques* de la caméra (distance focale, géométrie du capteur). Nous supposons que K est connue et constante au cours de la séquence. R et $\mathbf{t}$ représentent respectivement la matrice de rotation et le vecteur de translation définissant la transformation d'un repère lié à l'objet au repère de la caméra. R et $\mathbf{t}$ varient aux cours du temps. On notera donc $P_t = K.[\, R_t \mid \mathbf{t}_t \,]$ la pose de l'objet à l'instant t (associée à l'image I_t de la séquence).
L'objectif est de retrouver pour chaque image les paramètres de rotation et de translation, soit 6 paramètres.

Mesures dans les images. Plusieurs types d'indices visuels et méthodes peuvent être utilisés pour retrouver le mouvement 3D à partir de déplacements 2D : contours, motifs texturés, flot optique. Les méthodes présentées plus bas s'appuyent sur l'extraction et la mise en correspondance de points d'intérêt. Cette approche présente plusieurs avantages : elle peut-être rendue robuste aux changements d'illumination, aux occultations locales, et elle supporte des mouvements plus large que les méthodes de flot optique ou de suivi de motifs.
Des points d'intérêt selon le critère de Harris [10] sont mis en correspondance entre deux images consécutives I_{t-1} et I_t. La mesure de similarité utilisée est une corrélation croisée normalisée de fenêtres centrées en les points candidats. Elle est invariante aux changements affines d'illumination.
Soient p^j_{t-1} un point d'intérêt dans l'image I_{t-1} et $p^i_t = p^{v(j)}_t$ le point apparié dans l'image I_t (voir Figure 1). Si p^j_{t-1} appartient à l'image de l'objet, il est la projection selon P_{t-1} d'un point 3D N^j appartenant à la surface définie par le modèle. N^j se projète dans I_t en le point $\tilde{p}^j_t$.

Résolution. On recherche donc $P_t = K.[R_t \mid \mathbf{t}_t]$ qui minimise conjointement les distances $\|\tilde{p}^j_t - p^{v(j)}_t\|^2$ pour toutes les correspondances de points.
Il s'agit en fait d'un problème particulier d'ajustements de faisceaux, mais ici la connaissance des facettes permet de s'affranchir de la détermination explicite des points 3D N^j.

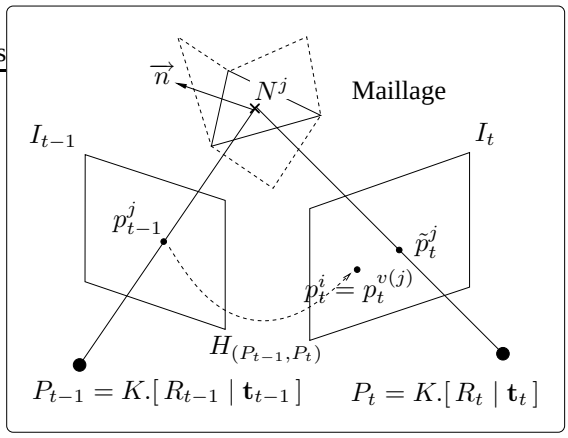


Figure 1 – *Repésentation géométrique du critère de (1)*

En effet, p^j_{t-1} et $\tilde{p}^j_t$ sont reliés par une hommographie qui ne dépend que des poses P_{t-1} et P_t et de la facette $\mathcal{F}$ à laquelle appartient N^j [11]. Ainsi si $\{x \mid \mathbf{n}^T.x - d = 0\}$ est le plan de la facette $\mathcal{F}$ ($\mathbf{n}$ est sa normale), alors on a :

$$\tilde{p}^j_t = H_{(\mathcal{F},P_{t-1},P_t)}.p^j_{t-1}$$

où

$$H_{(\mathcal{F},P_{t-1},P_t)} = K.(\delta R - \delta \mathbf{t}.\mathbf{n}'^T/d').K^{-1}$$

$$\delta R = R_t.R^T_{t-1}\ ,\ \delta \mathbf{t} = -\delta R.\mathbf{t}_t + \mathbf{t}_{t-1}$$

$$\mathbf{n}' = R_{t-1}.\mathbf{n}\ ,\ d' = d - \mathbf{t}^T_{t-1}.(R_{t-1}.\mathbf{n})$$

A chaque image, on est donc amené à résoudre le problème d'optimisation (sur les 6 paramètres de P_t) *non-linéaire* suivant :

$$\widehat{P}_t = \underset{P_t}{\operatorname{argmin}} \sum_j \|H_{(\mathcal{F}(p^j_{t-1}),P_{t-1},P_t)}.p^j_{t-1} - p^{v(j)}_t\|^2 \quad (1)$$

En pratique, les erreurs de localisation sur les points d'intérêt portant à la fois sur p^j_{t-1} et sur p^i_t , on symétrise le problème en transportant de la même manière p^i_t dans l'image I_{t-1}. Le critère est la somme de 2 distances, et à chaque étape l'estimation porte donc sur les 12 paramètres variables de P_{t-1} et P_t.
Ce problème aux moindres carrés non-linéaires est résolu itérativement par la méthode de Levenberg-Marquadt. Le suivi itératif ainsi posé est intéressant car il n'utilise aucune connaissance a priori sur la séquence (sauf la pose initiale). Cependant il souffre de l'accumulation des erreurs d'estimation, ce qui empêche toute utilisation sur des séquences plus longues que quelques centaines d'images. Pour pallier à ce problème [3] propose d'exploiter des informations a priori, constituées hors-ligne sous la forme d'images clés.

2.2 Intégration d'informations a priori

Une image clé est constituée d'une vue de l'objet à suivre et d'un recalage manuel précis du modèle. Les points

d'intérêts p^j_{cle} détectés dans ces images ainsi que leurs antécédents 3D N^j_{cle} sont précalculés.

Pour chaque pairs de points appariés $(p^j_t, p^{w(j)}_{cle})$ entre l'image courante I_t et l'image clé I_{cle}, on peut compléter (1) en sommant également sur les termes :

$$\|p^j_t - \phi(P_t, N^{w(j)}_{cle})\|$$

où $N^{w(j)}_{cle}$ est l'antécedent 3D de $p^{w(j)}_{cle}$ et $\phi(P, .)$ est l'opérateur de projection selon la matrice P.

Il y a deux difficultés : la première est la mise en correspondance des points entre la vue courante et une image clé, qui peuvent être relativement éloignées. Vacchetti et al. [3] résolvent cela en synthétisant une version déformée de l'image clé dans laquelle les mesures de corrélation sont valides. La deuxième difficulté est le choix de l'image clé. On peut choisir celle dont les paramètres de pose sont les plus proches de la pose connue la plus récente P_{t-1}, mais ce choix est parfois sous-optimal pour l'objectif de mise en correspondance. Il est préférable d'utiliser une mesure de similarité entre l'image courante et l'image clé candidate, comme proposé dans [3].

3 Notre approche avec un modèle par points

La solution présentée à la section 2 est intéressante mais la source de ses difficultés d'implantation repose sur l'utilisation de facettes pour définir un modèle de mouvement 2D (homographique) utilisé sur un ensemble peu dense de points (les points d'intérêts). L'idée que nous proposons d'explorer est de rendre homogène l'approche de suivi 2D avec la modélisation de l'objet. Ainsi nous prétendons pouvoir nous passer des facettes, et ainsi gagner en souplesse (voir section 5). Dans la section suivante nous présentons le modèle à base de points que nous utilisons par la suite.

3.1 Le modèle

Le modèle utilisé est composé d'un ensemble de *patchs* définis par un centre $M_i = (x_i, y_i, z_i)^T$, et une normale $\mathbf{n}$ (voir Figure 2). Il s'agit d'un modèle similaire à celui utilisé par Muñoz et al. [7].

3.2 Un modèle de mouvement adapté aux points

Le mouvement 3D entre 2 images consécutives I_{t-1} et I_t est décrit par une transformation euclidienne 3D de matrice 4×4 :

$$M = \begin{bmatrix} \delta R & \delta \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

où δR et $\delta \mathbf{t}$ représentent la rotation et la translation subit par l'objet entre les poses P_{t-1} et P_t. Ainsi :

$$P_t = P_{t-1}.M$$

Pour obtenir un modèle de mouvement 2D adapté à l'utilisation de points, nous linéarisons ce modèle de mouvement à la manière de Drummond [4].

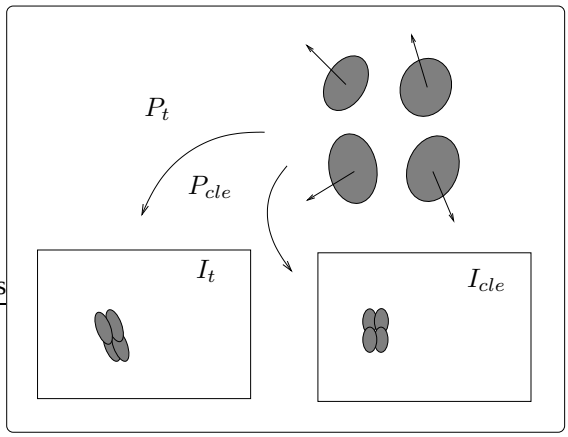


Figure 2 – *Le modèle défini par ensemble de* patches *texturés, projeté selon une pose courante ou selon la pose d'une image clé (voir section 3.5).*

M peut s'écrire sous forme exponentielle :

$$M = exp(\sum_{i=1}^{6} \alpha_i G_i)$$

où les matrices G_i forment une famille génératrice des mouvements 3D élémentaires (rotations et translations par rapport aux axes du repère de l'objet) :

$$G_i = \begin{bmatrix} 0 & -\delta_{i,6} & \delta_{i,5} & \delta_{i,1} \\ \delta_{i,6} & 0 & -\delta_{i,4} & \delta_{i,2} \\ -\delta_{i,5} & \delta_{i,4} & 0 & \delta_{i,3} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ \delta_{i,j} = \begin{cases} 1 \text{ si } i = j \\ 0 \text{ sinon} \end{cases}$$

et $\alpha = (\alpha_1, \cdots, \alpha_6)$ sont les paramètres de translations et de rotations correspondants.

Sous l'approximation :

$$M \approx I + \alpha_i.G_i \tag{2}$$

on peut calculer une famille génératrice $\{L_{j,i}\}_{i=1,\ldots,6}$ de vecteurs de mouvement 2D en tout point $p^j = (u, v, w)$ projection dans le plan image d'un point 3D $N^j = (x, y, z, 1)^T$. On a :

$$\forall i \in [1,6],\ \mathbf{L}_{j,i} = \begin{pmatrix} \frac{u'w - uw'}{w^2} \\ \frac{v'w - vw'}{w^2} \end{pmatrix} \text{ avec } \begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = P.G_i. \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

La famille $\{\mathbf{L}_{j,i}\}$ permet d'exprimer le mouvement 2D induit par un mouvement 3D paramétré par α. La Figure 3 montre le champ des vecteurs $\mathbf{L}_{j,6}$ correspondant à une rotation autour de l'axe z.

Au contraire, notre approche a pour objectif de retrouver le mouvement 3D à partir de nombreuses mesures du mouvement local 2D dans les images. Cette approche a été précédemment utilisée avec des mesures échantillonnées le

long des arêtes d'un modèle [4, 3, 2], à notre connaissance jamais dans le cas de points isolés.
La Figure 4 illustre ce modèle de mouvement. Pour le calcul de la famille $\{\mathbf{L}_{j,i}\}$ correspondant à un point quelconque p^j_{t-1}, voir la section 3.4.
Dans ce schéma, le suivi revient à estimer la matrice de la transformation euclidienne inter-image M, via ses paramètres α. La section suivante montre comment obtenir très simplement ce résultat.

3.3 Résolution

Chaque appariement fournit une évaluation d'un vecteur de mouvement $\mathbf{d}_j$:

$$\mathbf{d}^j = p^j_{t-1} - p^{v(j)}_t$$

Pour une transformation euclidienne M telle que $P_t = P_{t-1}.M$, sous l'approximation de l'équation (2), on a :

$$\mathbf{d}^j = \sum_{i=1}^{6} \alpha_i L_{j,i}$$

Soit $\mathbf{d}^j = (d^j_x, d^j_y)^T$ le vecteur de mouvement, $\mathbf{L}_{j,i}$ le i^{ieme} vecteur de la famille génératrice calculés en p^j_{t-1} et $L_j = (\mathbf{L}_{j,1}, \cdots, \mathbf{L}_{j,6})$ la matrice 2×6 contenant les 6 vecteurs de $\{L_{j,i}\}$. On a la relation suivante :

$$D = L.\alpha = L. \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_6 \end{pmatrix} \quad (3)$$

où

$$D = \begin{pmatrix} d^x_1 & d^y_1 \\ \vdots & \vdots \\ d^x_J & d^y_J \end{pmatrix} \text{ et } L = \begin{pmatrix} L_1 \\ \vdots \\ L_J \end{pmatrix}$$

J est le nombre d'appariements.

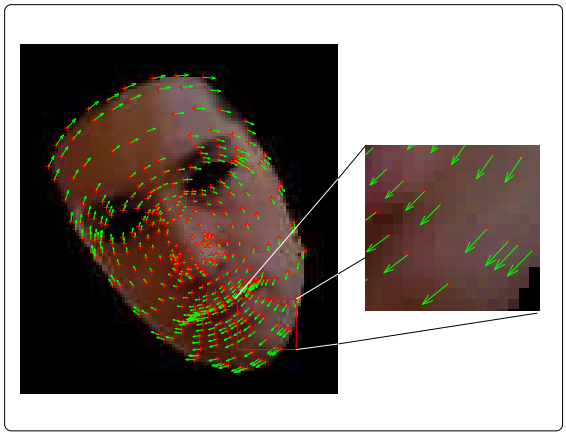


Figure 3 – *Le champ des vecteurs $\mathbf{L}_{j,6}$ correspondant à une rotation autour de l'axe z.*

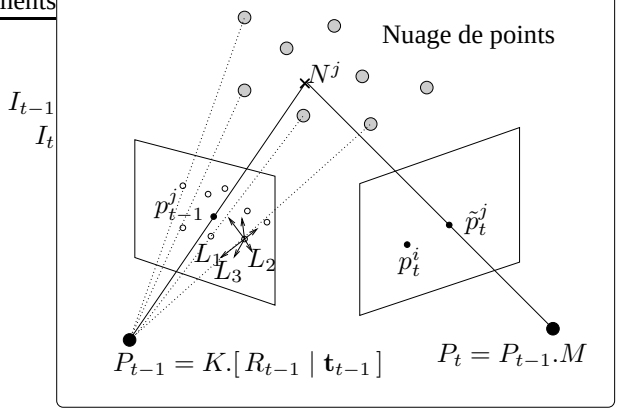


Figure 4 – *Projection des générateurs G_i dans l'image. On a représenté la famille $\{\mathbf{L}_{j,i}\}$ en la projection d'un point particulier du modèle*

L'équation 3 lie de manière linéaire les paramètres du mouvement euclidien 3D et le mouvement apparent D calculé en les points d'intérêts par l'intermédiaire de la matrice L. La résolution de ce problème revient à évaluer la pseudo-inverse de L :

$$\alpha = L^+.D$$

Nous détaillons dans la section suivante les aspects algorithmiques supplémentaires intervenant dans la mise en oeuvre de ce schéma d'estimation.

3.4 Calcul du modèle de mouvement 2D

Nous mesurons les vecteurs de mouvement de manière éparse en des points d'intérêt, que nous considérons comme la projection d'un échantillonnage particulier de la surface définie par le nuage de points du modèle. Pour appliquer le schéma proposé plus haut, nous avons besoin de la connaissance de la famille $\{\mathbf{L}_{j,i}\}$ en chacun des points d'intérêts p_j.

Nous proposons deux approches. La première utilise une projection arrière des points d'intérêts sur le modèle. Pour cela, nous exploitons une technique issue de la synthèse d'images qui consiste à calculer l'intersection du rayon issu du point d'intérêt avec une reconstruction locale de type MLS de la surface définie par le nuage [12]. Le coût de cette technique reste raisonnable pour un nombre de rayons limité (autant que de points d'intérêts appariés).

La deuxième possibilité consiste à considérer un maillage 2D dont les noeuds sont la projection des points du modèle. Le zoom de la Figure 3 montre qu'il est raisonnable d'interpoler localement le champ de mouvement calculé en les noeuds pour le reconstruire en n'importe quel point. Un schéma d'interpolation utilisant un modèle de type Thin Plate Spline [13] est utilisé pour évaluer les familles $\{\mathbf{L}_{j,i}\}$ en tout point.

3.5 Intégration des connaissances a priori

La mise à jour itérative de la pose souffre de l'accumulation d'erreur d'estimation, ce qui peut conduire à l'échec du processus après quelques dizaines d'images (voir les résultats expérimentaux en section 4). Comme en section 2.2, nous introduisons donc des connaissances a priori sur la séquence grâce à des images clés, pour lesquelles la pose de l'objet est déterminée manuellement. Les points d'intérêts ainsi que leurs antécédents 3D sont également calculés. Nous procédons en deux étapes.

Etape 1. Il s'agit du schéma décrit en section 3.3. :
– extraction des points d'intérêts p_t^j,
– formation de D^1 (Eq 3) à partir des appariements $(p_{t-1}^j, p_t^{v(j)})$,
– calcul des vecteurs $L_{j,.}^1$ en les points p_{t-1}^j, formation de la matrice L^1,
– estimation de la mise jour $\widehat{\alpha}^1 = L^{1+}.D^1$ des paramètres du mouvement 3D.

Etape 2. Raffinage de l'estimation courante grâce à une image clé.
– sélection de l'image clé et rendu des *patches* au voisinage de la pose courante (grâce à l'estimation de la pose courante P_t, disponible en sortie de l'Etape 1, cf. Figure 2),
– formation de D^2 à partir des appariements $(p_t^j, p_{cle}^{w(j)})$,
– calcul des vecteurs $L_{j,.}^2$ en les points p_t^j, formation de la matrice L^2,
– estimation raffinée grâce à la correction $\widehat{\alpha}^2 = L^{2+}.D^2$.
Il faut noter que l'etape 2 est itérable.

4 Expériences

4.1 Données synthétiques

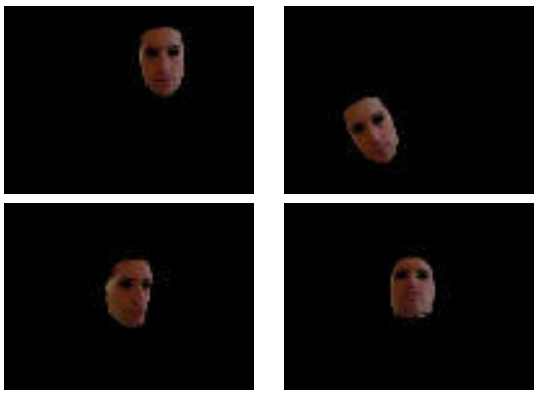


Figure 5 – *4 images issues de la s´equence de synth`ese.*

Nous possédons un modèle CAO réaliste de visage, établi par un graphiste à partir de photographies. Il vient sous la forme d'un maillage composé de facettes triangulaires. Nous avons recalé manuellement ce modèle sur une vue réelle de visage pour obtenir des coordonnées de texture en chaque point du maillage. A partir de ces données, nous avons produit un modèle par points tel que décrit en section 3.1. Celui-ci contient environ 500 points. Ces données nous ont permis de générer une séquence de synthèse par rendu OpenGL du maillage texturé. La séquence est obtenue en faisant évoluer les 6 paramètres de pose de la caméra entre chaque image de la séquence. Les paramètres utilisés pour générer la séquence constituent alors une vérité terrain à laquelle nous confrontons les valeurs estimées par les différentes méthodes implantées pour en évaluer la qualité. Toutes les méthodes évaluées ont été implantées en Matlab. La Figure 5 montre 4 images de la séquences de synthèse utilisée.

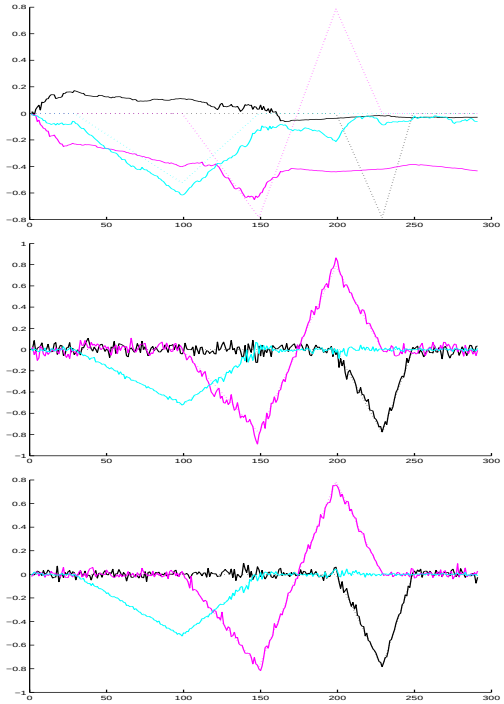


Figure 6 – *Evolution des 3 param`etres de rotation estim´es pour les 300 images de la s´equence. Les courbes en pointill´es correspondent `a la v´erit´e terrain. En haut : notre m´ethode sans image cl´e ; au milieu : notre m´ethode avec images cl´es ; en bas : la m´ethode non-lin´eaire de r´ef´erence, avec images cl´es.*

La Figure 6 présente les résultats du suivi sur une séquence de synthèse de 300 images exhibants différents mouvements de rotation et de translation du modèle devant une caméra perspective dont les paramètres sont fixés. L'erreur de localisation sur les points d'intérêt est gaussienne sans biais d'écart-type 2 pixels. Les courbes en pointillés représentent la vérité terrain telle qu'utilisée pour générer la séquence, les traits pleins sont les valeurs des paramètres estimés pour chaque méthode.
Deux conclusions peuvent être tirées de ces résultats : tout d'abord, que se soit dans la méthode de référence ou dans notre approche linéaire, l'utilisation des images clés corrige efficacement l'accumulation d'erreur.

Ensuite, malgré l'approximation que nous effectuons, notre méthode reste suffisamment stable bien que logiquement moins précise. Les composantes de α étant du même ordre de grandeur, nous pouvons comparer l'erreur médiane absolue maximale ; elle de l'ordre de 10^-2 pour notre méthode contre 10^{-4} pour la méthode de référence. Ceci est obenu pour un cout calculatoire bien plus faible. Ceci s'explique par le fait que notre approche, en linéarisant le problème au plus tôt, ne requière que l'évaluation de la pseudo-inverse de la matrice L, alors que le critère non-linéaire est évalué plusieurs centaines de fois (lors du calcul des jacobiennes) dans la méthode non-linéaire.

4.2 Données réelles

Figure 7 – *Deux images de suivi d'une s´equence r´eelle.*

Nous avons également des résultats portant sur des séquences vidéo réelles acquises par une caméra couleur AVT Marlin F-046C calibrée (voir Figure 7). Le suivi de telles séquences nécessite de rendre robuste les différentes estimations intervenant dans notre algorithme. Nous utilisons une technique classique de moindres carrés pondérés itérés. Ceci consiste à pondérer les mesures issues de l'appariement des points d'intérêts apparaissant dans les lignes des matrices D (cf Eq. 3)

5 Conclusion et extensions envisagées

Nous avons démontré la possibilité d'utiliser un modèle s'appuyant uniquement sur des points pour le suivi visuel 3D. Cette technique est à la fois plus simple à mettre en oeuvre et plus rapide. De plus, elle apporte de la cohérence entre le modèle de l'objet et le traitement bas niveau (extraction de points d'intérêt).
Nous pensons également que l'utilisation d'un modèle sans topologie sera bénéfique dans la prise en compte d'objets déformables et pour le raffinage du modèle au cours du suivi. Voilà vers quoi nous souhaitons orienter nos futurs travaux.

Références

[1] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, et Blair MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6) :34–47, 2001.

[2] Andrew I. Comport, Éric Marchand, et François Chaumette. Robust model-based tracking for robot vision. Dans *IEEE Int. Conf on Intelligent Robots and Systems, IROS04*, Sendai, Japan, September 2004.

[3] Luca Vacchetti, Vincent Lepetit, et Pascal Fua. Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10) :1391–1391, 2004.

[4] Tom Drummond et Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :932–946, July 2002.

[5] Luca Vacchetti, Vincent Lepetit, et Pascal Fua. Combining edge and texture information for real-time accurate 3d camera tracking. Dans *International Symposium on Mixed and Augmented Reality*, Arlington, VA, November 2004.

[6] Charles S. Wiles, Atsuto Maki, et Natsuko Matsuda. Hyperpatches for 3d model acquisition and tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(12) :1391–1403, 2001.

[7] Enrique Munoz, Jose M. Buenaposada, et Luis Baumela. Efficient model-based 3d tracking of deformable objects. Dans *Proceedings of ICCV 2005*, pages 877–882, Beijing, China, October 2005.

[8] Frederick Rothganger, Svetlana Lazebnik, Cordelia Schmid, et Jean Ponce. 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 272–277, Madison, WI, June 2003.

[9] Mark Pauly. *Point Primitives for Interactive Modeling and Processing of 3D Geometry*. Thèse de doctorat, Federal Institute of Technology (ETH) of Zurich, 2003.

[10] Chris Harris et Mike Stephens. A combined corner and edge detector. Dans *Fourth Alvey Vison Conference*, Manchester, 1988.

[11] Richard Hartley et Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN : 0521540518, second édition, March 2004.

[12] Anders Adamson et Marc Alexa. Ray tracing point set surfaces, 2003.

[13] Fred L. Bookstein. Principal warps : Thin-plate splines and the decomposition of deformations. *IEEE Transaction on PAMI*, 11(6), 1989.

Evaluation de la qualité des codeurs vidéo dans le contexte de la vidéo-surveillance.

L. Quintard M.C. Larabi C. Fernandez-Maloigne

SIC Signal Image Communication

Université de Poitiers
Blvd Marie & Pierre Curie, BP30179
86962 Futuroscope Cedex

{quintard, larabi, maloigne}@sic.univ-poitiers.fr

Concours Jeune Chercheur : Oui

Résumé

Nous proposons une étude complète afin de fournir des recommandations de débits pour des codeurs de la famille MPEG-x et le MJPEG2000. Le but étant d'avoir une qualité perceptuelle identique à celle obtenue avec un codeur hardware MJPEG dont le débit est fixé à 5.6 Mbit/s. Cette étude s'applique dans le cadre de la vidéo surveillance. En premier, une étude objective est réalisée. Pour ce faire, trois métriques sont utilisées, le PSNR[1]*, l'index Universel de qualité ainsi qu'une métrique que nous avons développée. Par la suite, nous proposons une étude subjective en accord avec les recommandations ITU*[2] *: le test DSIS*[3] *pour mesurer la dégradation globale de la vidéo par rapport à l'originale, le test DSCQS*[4] *pour mesurer l'impression visuelle globale de chaque vidéo. Finalement, l'étude de la corrélation entre les évaluations subjectives et objectives est réalisée.*

Mots clefs

Evaluation subjective, évaluation objective, qualité vidéo, corrélation.

1 Introduction

La compression joue un rôle très important dans le contexte de la vidéo surveillance. En effet, la vidéo nécessite des capacités de stockage énorme. Pour pallier à ce point, la compression permet de réduire la taille des données et donc de diminuer la capacité de stockage nécessaire. De nombreux codecs sont utilisés, certains utilisent une méthodologie de compression image par image comme le MotionJPEG (MJPEG) et le MJPEG2000 [1] quand d'autres travaillent sur l'aspect temporel présent au sein d'une vidéo, c'est notamment le cas des codecs de la famille MPEG[5] [2, 3, 4]. Dans tous les cas, les codecs introduisent des artefacts dans la vidéo [5]. Ces derniers affectent la qualité visuelle de la vidéo compressée.

Cependant, dans le cadre de la vidéo surveillance, il est nécessaire que le codec fournisse une qualité vidéo suffisante et, ce afin de pouvoir reconnaître le visage des personnes par exemple. La meilleure solution pour mener à bien ce problème, est d'évaluer objectivement et subjectivement les vidéos.

Les méthodes objectives sont basées pour la plupart sur la mesure de différence entre la vidéo originale et la vidéo compressée. Cette mesure peut utiliser un simple calcul mathématique comme le PSNR ou peut intégrer des propriétés du SVH[6].

Les méthodes subjectives exploitent le jugement humain et nécessite des conditions d'expérimentations spécifiques comme par exemple une salle normalisée. Quand l'expérimentation psychophysique est faite, et que les analyses statistiques sont effectuées, les données issues de l'évaluation sont considérées comme cohérentes et peuvent être utilisées.

Une fois ces deux tests effectués, il est nécessaire d'étudier la corrélation existant entre les deux. Pour ce faire, le groupe VQEG[7] fournit des recommandations afin de corréler les tests subjectifs et objectifs [6].

Ce papier décrit l'expérimentation que nous avons conduite pour différents codeurs vidéo. Le propos est de pouvoir fournir des recommandations de débits en fonction du codeur vidéo et ce pour obtenir une qualité visuelle équivalente à celle obtenue avec un codeur hardware MJPEG dont le débit est fixé par l'exploitant à 5.36 MBit/s. Un autre objectif est de pouvoir fournir des seuils de métrique dans le cas où il serait utile de tester différentes implémentations de codeurs hardwares. Les codecs testés sont le MJPEG2000, MPEG-1, MPEG-2 et MPEG-4. Les

[1] Peak Signal Noise Ratio
[2] International Communication Union
[3] Double Stimulus Impairment Scale
[4] Double Stimulus Continuous Quality Scale
[5] Moving Picture Expert Group
[6] Système Visuel Humain
[7] Video Quality Expert Group

vidéos sélectionnées sont représentatives et de format CIF[8]. La figure 1 illustre un exemple de vidéo utilisées.

-a- -b-

Figure 1 – *Les figures -a- et -b- sont représentatives des vidéos utilisées pour l'étude.*

Les résultats subjectifs sont obtenus en utilisant deux methodologies définies dans [7], nommées DSIS et DSCQS.
Les résultats objectifs sont obtenus en utilisant trois métriques. Le PSNR et deux autres métriques intégrant quelques propriétés du SVH, celles-ci sont l'Index Universel de Qualité [8], et une métrique développée lors de cette étude [9].
Ce papier est organisé de la manière suivante : La section 2 présente l'évaluation objective. La méthodologie de l'évaluation subjective est présentée en section 3. Ensuite, la section 4 est dédiée à l'étude de corrélation entre évaluation subjective et objective. Finalement, nous concluons en section 5.

2 Evaluation objective

Lorsque qu'une vidéo est compressée, elle subit des modifications entraînant des artefacts. Ces artefacts sont plus ou moins gênants d'un point de vue perceptuel [10]. Ils sont inhérents à la compression et peuvent prendre plusieurs formes. La figure 2 illustre un artefact de bloc et de débordement de couleurs.

Figure 2 – *Exemple d'artefacts, effet de bloc (gauche), débordement de couleurs (droite).*

Le but d'un test objectif est donc de pouvoir évaluer ces différents artefacts. Ces métriques peuvent être avec référence, sans référence ou avec référence réduite. Les métriques sans référence sont peu nombreuses dans la littérature. Le fait de ne pas avoir d'image originale implique que l'on ne se trouve plus dans une mesure de fidélité mais bien dans une évaluation absolue de qualité. Cette dernière se calcule en général sur des attributs de l'image pour lesquels nous savons que le SVH est sensible, comme le contraste. Il faut de plus, considérer deux sortes de métriques, celles sans pondération, basées sur une mesure de distance comme le PSNR, et les autres sur critères pondérés qui prennent en compte dans leurs calculs quelques propriétés du SVH [11, 12].

Dans cette étude, nous nous proposons de tester trois métriques devant avoir un temps de calcul rapide. Pour ce faire, nous avons opté pour l'utilisation du PSNR, de l'index universel de qualité [8] ainsi qu'une métrique développée lors de cette étude [9].
Ces trois métriques sont avec référence. Le PSNR compare les différences entre les pixels de chaque image, quand deux images sont fidèles sa valeur se situe aux alentours de 40dB. Certes cette mesure n'est pas toujours corrélée avec le jugement humain, mais, dans le cadre de la compression les résultats sont parfois satisfaisants.
Le système visuel humain est un détecteur de contraste [13], l'index universel de qualité [8] et la nouvelle métrique [9] travaillent sur le contraste local d'une image. Elles ont respectivement une échelle de distance comprise entre $[-1; 1]$ et $[0; 1]$. Avec 1 quand l'image de référence et l'image compressée sont fidèles d'un point de vue perceptuel. A l'inverse, les valeurs -1 ou 0 sont obtenues quand la fidélité perceptuelle est mauvaise.
L'index de qualité universel suit l'approche suivante :

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]} \quad (1)$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad (2)$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2, \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \quad (3)$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) \quad (4)$$

Les équations 1, 2, 3 et 4 expriment la corrélation existant entre deux images. La formule comprend en fait trois mesures. La première est la corrélation linéaire entre deux images x et y. La seconde est la différence entre les valeurs moyennes de x et y. La dernière mesure la similarité du contraste.
La fenêtre de calcul de taille $A * A$ se déplace horizontalement et verticalement pixel par pixel.
Quant à la métrique que nous avons proposée elle travaille sur le contraste mais avec une approche différente. Le contraste est calculé par une fenêtre de taille $A * A$ par l'équation 5.

[8]Common Intermediare Format ($352 * 288$)

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (5)$$

$$R_i = \bar{x} - \bar{y} \quad (6)$$

$$R = 1 - \left([\frac{1}{M}\sum_{i=1} M\,|R_i|] * \theta \right) \quad (7)$$

La fenêtre se déplace horizontalement et verticalement par pas de dimension A. La différence est donnée fenêtre par fenêtre entre l'image de référence et l'image compressée comme le décrit l'équation 6. Le résultat final est calculé selon 7 où θ représente le maximum de distorsion admissible.

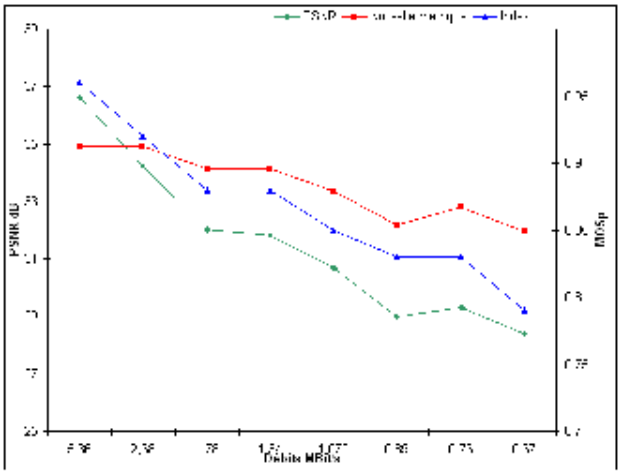


Figure 3 – *Exemple de résultats pour les différentes métriques dans le cas du codeur MPEG-2.*

La figure 3, montre un exemple de résultats pour les différentes métriques dans le cas du codeur MPEG-2.
A partir de cette figure, il est possible d'obtenir des seuils qui pourront servir par la suite à l'évaluation de codec. Néanmoins, ces seuils ne pourront réellement servir que si lors de l'étude de la corrélation entre les tests subjectifs et objectifs, les résultats sont concluants.

3 Evaluation psychophysique

3.1 Conditions

L'évaluation subjective a été réalisée avec des vidéos représentant des conditions de vidéo surveillance (*cf.* figure 1). Les quatre codecs testés sont MJPEG2000, MPEG-1, MPEG-2, MPEG-4. Différents débits ont été choisis et ce de manière linéaire. La table 2 donne les échelles de débits évalués pour chaque codec.
Les conditions d'observation respectent les normes ITU décrites dans [7, 14]. La figure 4 (a) illustre la configuration

Tableau 1 – *Débits utilisés pour les différents codecs.*

MPEG1	0.76 à 5.36 Mbit/s
MPEG2	0.67 à 5.36 Mbit/s
MPEG4	0.357 à 5.36 Mbit/s
MJPEG2000	0.76 à 5.36 Mbit/s

de laboratoire utilisée. La distance entre l'écran et l'observateur est de 60 centimètres afin de respecter les conditions réelles du superviseur. Le moniteur est à tube cathodique, calibré (24" Sony). Chaque session est limitée à 25 minutes afin de ne pas dépasser les capacités de concentration de l'observateur. Le mur est d'un gris neutre. Les vidéos sont visualisées décompressées à 25 images/seconde.

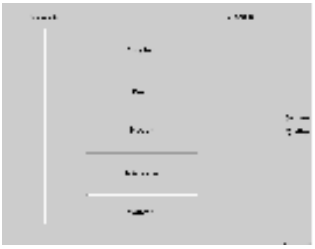
3.2 Méthode d'évaluation

Les deux méthodes utilisées lors de l'évaluation psychophysique sont spécifiées dans ITU-R Recommenadation BT.500 [7]. La qualité perceptuelle doit être la même que celle obtenue pour un codec MJPEG à 5.36MBit/s. Pour ce faire, nous devons comparer les vidéos obtenues par l'utilisation de ce codec avec des vidéos obtenues par d'autres codecs. Ceci est en réalité le sujet du test DSIS. Nous utiliserons aussi le test DSCQS afin d'avoir une comparaison entre les codecs.

– **Le test DSIS :** l'observateur visualise différentes séquences vidéo par paire. La première vidéo étant la référence. Il doit juger la dégradation existant entre la vidéo compressée et la vidéo de référence. Pour ce faire, il dispose d'une échelle discrète de jugement allant de *imperceptible* à *très génant*.
– **Le test DSCQS :** l'observateur visualise des vidéos deux à deux. Il donne son jugement sur chaque vidéo à partir d'une échelle linéaire allant de *mauvais* à *excellent*.

La figure 4 (b), montre l'interface développée pour cette étude. Chaque séquence vidéo dure 12 secondes.

-a-



-b-

Figure 4 – *Exemple d'une salle de laboratoire pour l'évaluation psychophysique (a). Interface dévelopée pour le test DSCQS (b)*

3.3 Les observateurs

Pour cette étude, 23 observateurs non expert ont participé à la session de test. L'acuité visuelle ainsi que la perception des couleurs ont été testés sur chaque observateur par l'intermédiaire des tests de *Snellen* et d'*Ishihara*.

3.4 Analyse de l'évaluation psychophysique

Une fois les tests réalisés, il est de nécessaire de calculer le MOS[9] et l'intervalle de confiance à 95% [7]. Ces résultats ne peuvent être calculés qu'à partir des observateurs dont le jugement est cohérent. Pour ce faire une étude statistique est réalisée en s'appuyant sur le test du *kurtosis*. Afin de faciliter l'analyse, une valeur numérique linéaire est attribuée selon l'échelle de valeur du test.

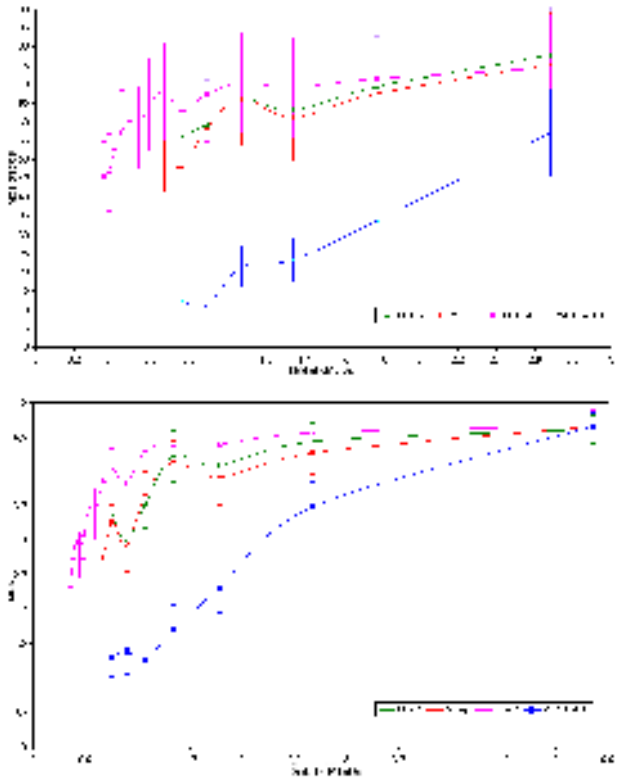


Figure 5 – *MOS DSCQS (haut) MOS DSIS (bas) vs débits. Les barres verticales indiquent l'intervalle de confiance à 95%*

Les figures 5 et 6 sont riches en informations et viennet confirmer des résultats certes prévisibles comme la supériorité du codeur MPEG-4 qui permet d'atteindre des débits records. Cependant, nous avons ici une information sur les débits permettant d'avoir une même fidélité perceptuelle (test DSIS). De plus, étant donné la bonne corrélation existant entre les tests DSIS et DSCQS (coefficient de Pearson de 94%) comme le montre la figure 6, nous pouvons ainsi connaitre par l'intermédiaire du test DSCQS la qualité perceptuelle pour ces différents débits.

[9]Mean Opinion Score / score moyen des opinions

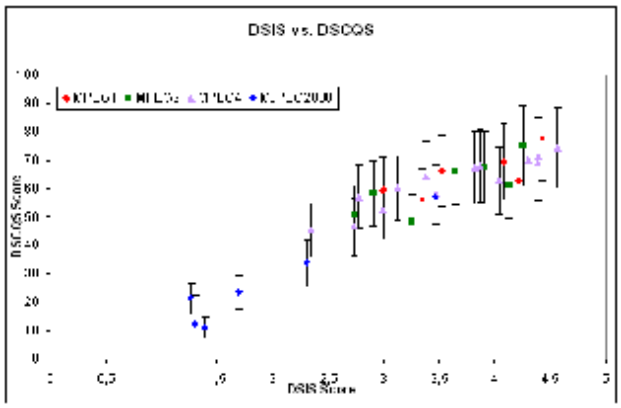


Figure 6 – *Comparaison entre les MOS du test DSIS et DSCQS.*

Tableau 2 – *Débits minimum pour la même qualité perceptuelle obtenue avec un codec MJPEG à 5.36 MBit/s*

CODEC	DEBITS
MPEG1	1 Mbit/s
MPEG2	1 Mbit/s
MPEG4	0.76 Mbit/s
MJPEG2000	3 Mbit/s

Le tableau 2 donne les débits minimum auxquels nous pouvons prétendre pour une qualité perceptuelle équivalente.

4 Etude de la corrélation

Les évaluations subjectives sont très contraignantes. C'est pourquoi une évaluation objective, qui utilise une métrique est bien plus intéressante. Néanmoins, il est nécessaire que la métrique fournisse des informations en concordance avec le jugement humain. Les attributs qui permettent de caractériser la performance d'une métrique objective par rapport aux données subjectives sont :

– Prédiction de l'exactitude
– Prédiction de la monotonicité
– Prédiction de l'uniformité

Nous ne détaillerons ici qu'une partie de la prédiction de la monotonicité, pour de plus amples informations le lecteur pourra se référer à [15] par exemple.

4.1 Modèle de prédiction de la monotonicité

L'analyse de la corrélation nous indique le degré par lequel les valeurs de la variable Y peuvent être prédites, ou expliquées, par les valeurs de la variable X. Une forte corrélation implique qu'il est possible d'effectuer une inférence sur Y en partant de X.

La force et la direction du rapport entre X et Y sont données par le coefficient de corrélation. Il est souvent facile de prévoir s'il y a une corrélation, simplement en examinant les données en utilisant des nuages de points (voir

la fig. 7).

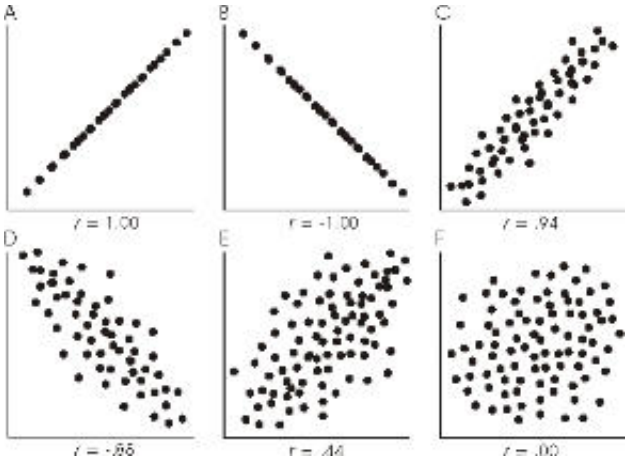


Figure 7 – *Nuage de points tracé indique divers degrés de corrélation linéaire.*

Coefficient de corrélation Pearson Le coefficient de corrélation Pearson r est utilisé pour des données sur des échelles d'intervalle ou de rapport, et est basé sur le concept de la covariance. Quand des échantillons X et Y sont corrélés il est possible de dire qu'ils varient conjointement ; ou qu'ils sont dans des modèles similaires.
Le produit statistique du moment r est donné par :

$$r = \frac{n\sum_{i=0}^{n} X_i Y_i - (\sum_{i=0}^{n} X_i)(\sum_{i=0}^{n} Y_i)}{\sqrt{(\left[n\sum_{i=0}^{n} X_i^2 - (\sum_{i=0}^{n} X_i)^2\right]\left[n\sum_{i=0}^{n} Y_i^2 - (\sum_{i=0}^{n} Y_i)^2\right])}} \quad (8)$$

Où n est le nombre de paires de scores. Le degré de liberté est $df = n - 2$.
Pour une valeur de $df = 5$, toute valeur de corrélation supérieure à 0.754, nous permettra de conclure que X et Y varie conjointement pour un intervalle de confiance à 95%. Les figures 8, 9 et 10 illustrent les corrélations obtenues selon la métrique considérée. Le coefficient de Pearson est proche de 1 dans tous les cas et au dessus de la valeur critique. Nous avons donc une forte corrélation entre les évaluations subjectives et objectives. Ceci nous permettra dans le futur de n'utiliser que les métriques pour une comparaison des qualités perceptuelles.

5 CONCLUSIONS

Cette étude a permis de fournir des recommandations de débits en fonction d'une qualité perceptuelle. Des tests subjectifs et objectifs ont été menés. Quatre codeurs ont été testés : MJPEG2000, MPEG-1, MPEG-2, MPEG-4. Les tests subjectifs ont permis de définir le tableau de recommandations 2. L'étude de la corrélation existant entre les tests subjectifs et objectifs autorise d'utiliser les seuils de métriques comme seuil de qualité sen restant dans la même base de vidéo.

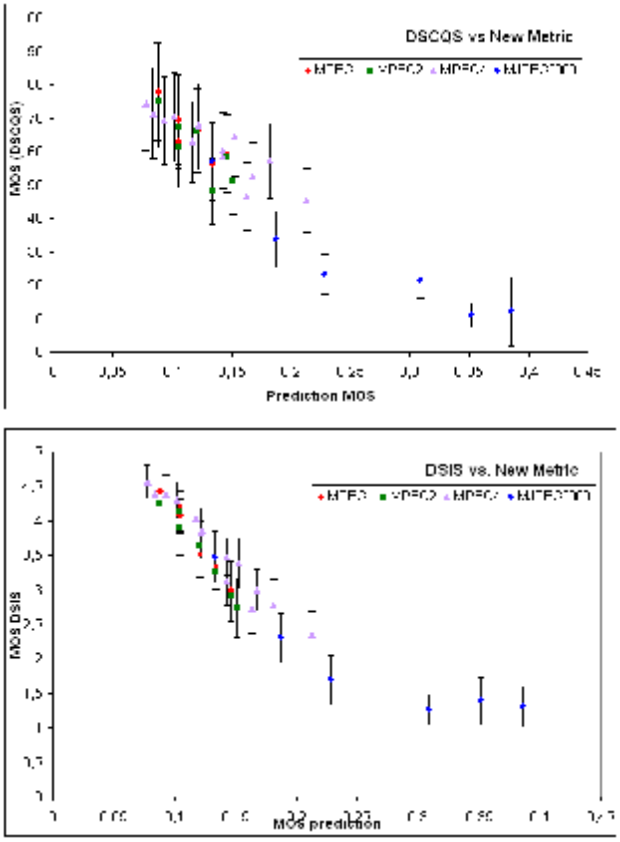


Figure 8 – *MOS prédit vs. MOS subjectif, nouvelle métrique.*

Références

[1] MJPEG2000 ISO/IEC JPEG committee. Information technology-jpeg 2000 image coding system. Rapport technique, Motion JPEG 2000-ISO IEC 15444-3 :2002, 2005.

[2] MPEG-1 ISO IEC. Mpeg-1 standard. Rapport technique, ISO IEC, 1993.

[3] MPEG-2 ISO IEC. Mpeg-2 standard. Rapport technique, ISO IEC, 1993.

[4] MPEG-4 ISO IEC MPEG-98. Mpeg-4 overview. Rapport technique, Requirements Group, 1998.

[5] Peter Symes. *Video compression demystified*. Mc Graw-Hill, New York, 2000.

[6] VQEG. Final report from the video quality expert group on the validation of objective models of video quality assessment. Rapport technique, VQEG, 2000.

[7] ITU-R Recommendation BT.500-10. Methodology for the subjective assessment of the quality of television picture. Rapport technique, International Communication Union, 2002.

[8] Zhou Wang et Alan C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 2002.

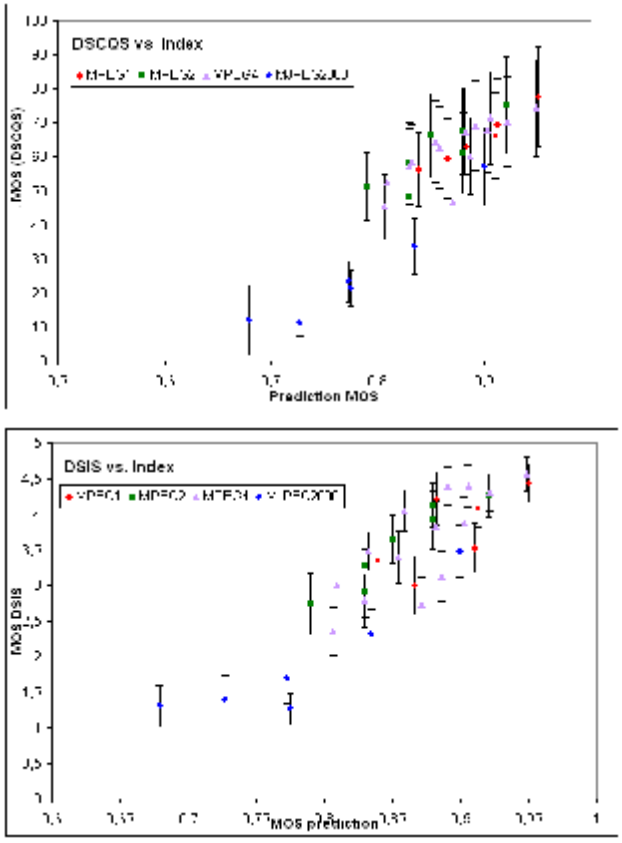


Figure 9 – *MOS prédit vs. MOS subjectif, index universel de qualité.*

[9] ITU-R Recommendation P.910. Subjective video quality assessment methods for multimedia applications. Rapport technique, International Communication Union, 1996.

[10] Z. Wang, H.R. Sheikh, et A.C. Bovik. *Handbook of video databases : design and applications*, chapitre Objective video quality assessment, pages 1041–1078. CRC press, Florida, september 2003.

[11] C. Larabi, A. Stoica, et C. Fernandez-Maloigne. La qualité d'images entre mesures quantitatives, qualitatives et expérience psychovisuelles. *Ecole de printemps - images numériques couleurs*, mars 2003.

[12] B.W Keelan. *Handbook of image quality : characterization and prediction*. Marcel Dekker, Inc, New York, 2002.

[13] Dr. Salmon's. National board part 1 review. Rapport technique, July 2000.

[14] L. Quintard et M.C. Larabi. Qualité vidéo : évaluation subjective et objective. Rapport technique, Université de poitiers, 2005.

[15] D.Freedman, R.Pisani, et R. Purves. *Statistics*. W.W.Norton & Company, New York, 1998.

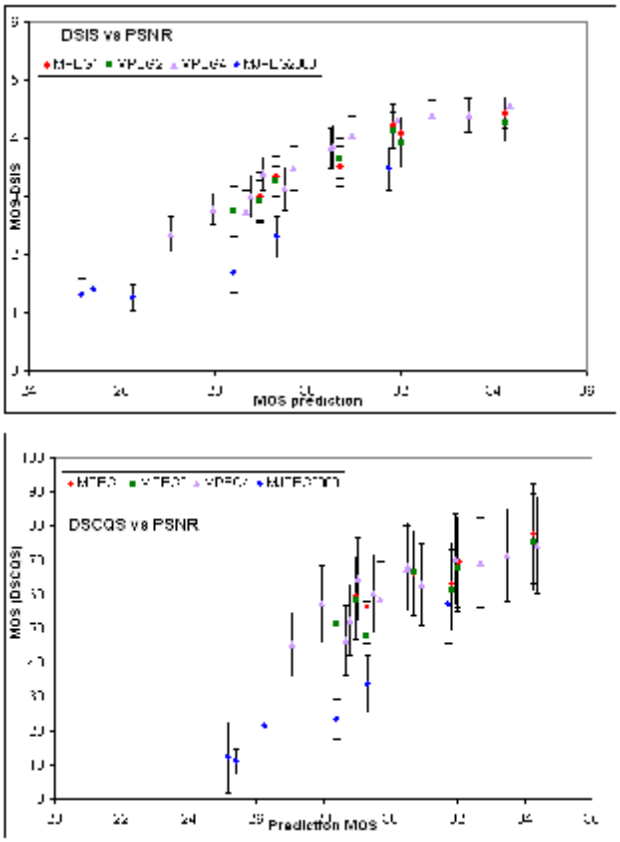


Figure 10 – *MOS prédit vs. MOS subjectif, PSNR.*

Etude préliminaire de l'influence des fréquences spatiales sur l'apparence couleur

Olivier Tulet Mohamed-Chaker Larabi Christine Fernandez-Maloigne

Laboratoire SIC FRE 2731

Université de Poitiers

Blvd Marie et Pierre Curie, BP30179

86962 Futuroscope Cedex

{tulet, larabi, maloigne}@sic.univ-poitiers.fr

Concours Jeune Chercheur : Oui

Résumé

Aujourd'hui, garantir la qualité couleur des produits est un réel challenge. C'est pour cette raison que les modèles d'apparence couleur ont été développés. Ces modèles corrigent et retournent la couleur perçue indépendemment de l'environnement. Ils prennent en compte de nombreux phénomènes qui peuvent altérer notre perception. Cependant ces modèles ne tiennent pas compte de certains phénomènes comme la sensiblité aux fréquences spatiales. Dans cette contribution, une approche basée sur des tests psychophysique pour résoudre ce problème est décrite. Ces tests sont basés sur un ajustement de la clarté, de la chroma et de la teinte de stimuli pour quantifier l'influence des fréquences spatiales sur la perception de l'observateur. Des résultats encourageants ont été obtenus et sont décrits dans ce papier.

Mots clefs

CIECAM, fréquences spatiales, tests psychophysiques, modèle d'apparence couleur.

1 Introduction

De nos jours, la qualité de la couleur est un challenge très important dans l'industrie. En fonction du média sur lequel on se trouve, la couleur ne semble pas toujours identique comme par exemple l'image à l'écran qui semble différente de sa reproduction sur papier via une imprimante. Ce problème concerne le wysiwyg (what you see is what you get) pour lequel de nombreux modèles d'apparence couleur (CAM) ont été développés[1].

Le principal objectif du CAM est d'assurer une bonne reproduction de la couleur à travers différents média en introduisant les caractéristiques du système visuel humain (SVH)[2]. De nombreux CAM existent et sont dédiés à diverses applications (industrie textile, imprimerie, etc.). Ces modeles ont été développés pour répondre à la requête incessante du monde industriel qui a besoin de standards. Ainsi, la CIE (Commission Internationale de l'éclairage) a normalisé en 1997 le CIECAM97 qui a ensuite été amélioré pour devenir le CIECAM02. Le CAM normalisé par la CIE prend en compte l'environnement d'un objet coloré pour compenser l'influence de ce dernier sur notre perception de la couleur.

La description du CIECAM est donnée par la figure 1 [3].

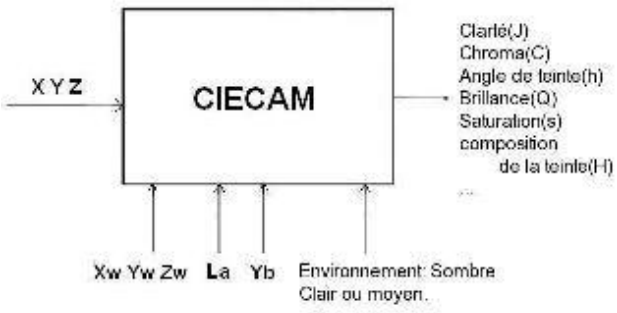


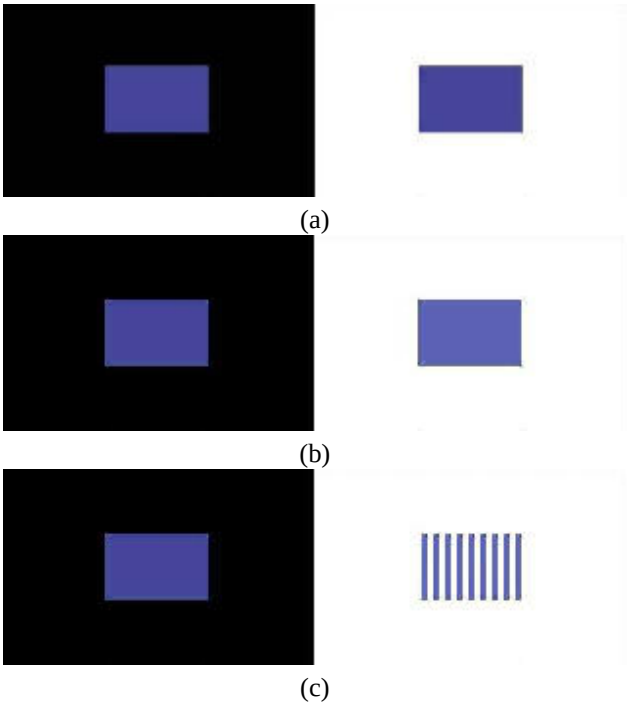
Figure 1 – *Diagramme d'entrée/sortie du CIECAM*

Cette figure décrit les entrées/sorties du CIECAM où les valeurs d'un stimulus d'entrée (dans l'espace de couleur XYZ) sont converties en attributs perceptuels en fonction de l'environnement donné.

Le but de cette transformation est de décorreler la perception de ce stimulus de son environnement. Pour ce faire, le CAM passe par des étapes complexes comme l'adaptation chromatique, le calcul de la réponse des cônes et d'autres pour obtenir des attributs perceptuels.

Le CIECAM02 prend en compte un grand nombre de phenomènes relatifs à l'environnement du stimulus comme

l'effet de Hunt ou l'effet de Stevens, etc. Cependant la sensibilité aux fréquences spatiales n'est prise en compte dans aucun modèle d'apparence couleur. C'est sur cette problématique que nous nous sommes penché. Ainsi, comme le montre la figure 2-a, un stimulus placé dans deux environnements différents (noir et blanc) paraîtra différent d'un point de vue perceptuel. Ce même stimulus est corrigé par le CIECAM02 (figure 2-b) donnant ainsi un aspect similaire quelque soit l'environnement. Cependant en le modulant par une fréquence spatiale donnée, la correction devient inefficace.



(a)

(b)

(c)

Figure 2 – *a : Stimuli uniformes avec différents arrières plan. b : Couleur de (a) corrigée, c : introduction d'une fréquence spatiale à (b)*

Ce problème est un vrai challenge car beaucoup d'images sont naturellement construites de fréquences spatiales (comme pour une texture par exemple). C'est pourquoi bon nombre d'auteurs recommandent d'intégrer ce phénomène aux CAMs [1, 4, 5] tout en mesurant les difficultés liées à cette tâche.
Ainsi, le but de ce travail est d'étudier l'effet des fréquences spatiales sur l'apparence de la couleur et de l'intégrer dans le CAM normalisé par la CIE.
L'approche proposée ainsi que les expériences menées seront décrites dans la section 2. La section 3 est dédiée aux résultats obtenus. Cette contribution se terminera par une conclusion dans laquelle nous décrirons les travaux à venir.

2 Approche proposée

Comme décrit précédemment, le but de cette étude est de quantifier l'influence des fréquences spatiales sur la perception de la couleur.
Cette quantification permettra de corriger les variations de la perception dues aux fréquences tout comme les modèles courant corrigent les variations dues aux données relatives à l'environnement.
Le but de cette étude est d'extraire un modèle du comportement du SVH à l'aide de tests psychophysiques. Ces tests vont permettre de mesurer la différence perçue entre un motif uniforme et un motif avec une certaine fréquence spatiale. Il va falloir réaliser cette mesure sur trois critères de sorties du CIECAM que nous avons choisi pour leur pertinence. Ceci permettra de voir comment ces derniers varient en fonction de la fréquence spatiale. Ces tests seront aussi réalisés sur les couleurs primaires afin de mesurer l'influence des fréquences sur telle ou telle composante couleur.
Les tests psychophysiques nécessitent une préparation rigoureuse autant pour la création des motifs que du choix de l'environnement. Les sections suivantes décrivent cette préparation.

2.1 Salle psychophysique

Pour obtenir une bonne quantification de l'influence des fréquences spatiales avec des tests psychophysiques un environnement normalisé doit être utilisé.

Figure 3 – *Salle psychophysique*

Conformément au standard ISO 3664 [6] cet environnement doit respecter plusieurs conditions comme par exemple la couleur des murs qui doit être neutre ou la chromaticité de l'arrière plan qui doit correspondre à un illuminant D65.
Notre salle psychophysique répond à toutes les conditions de ce standard et est donc utilisée pour notre expérimentation. Un autre point important est le choix et la calibration de l'écran. Pour cette expérience, nous utilisons un SONY® FW900 de ratio 16/10 et de dia-

gonale 24 pouces. Le calibrage couleur du tube CRT a été réalisé avec le calibreur d'écran EYE-ONE monitor Mach 1.1 color calibrator du GretagMacbeth® et vérifié (éventuellement corrigé) à l'aide des mesures effectuées avec un spectro-colorimètre PR-650 SpectraScan.

2.2 Présentation des stimuli

Le cône de vision binoculaire est optimum pour un angle de 10-12 degrés d'angle visuel. C'est pourquoi un rectangle s'étendant sur une surface de 10 degrés d'angle visuel a été choisi pour cette étude. Les stimuli sont construits avec les trois couleurs primaires (rouge, vert, bleu) avec des fréquences variant de 1 à 17 cpd et sont modulés en crénaux. Ces configurations permettent d'obtenir 63 tests différents. La figure 4 donne un exemple de condition de visualisation.

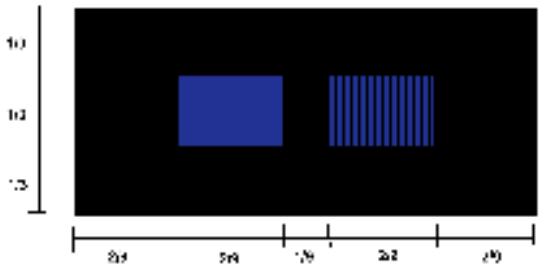


Figure 4 – *Exemple de test psychophysique avec un stimulus bleu à une fréquence donnée*

Cette figure montre la répartition du stimulus sur l'écran. Avec la distance écran-observateur d'un mètre cinquante et la résolution de l'écran on retrouve le cône de vision décrit plus haut.

2.3 Procedure de test

Dans ce test psychophysique, il est demandé à l'observateur d'ajuster un critère sur la couleur comme la clarté (J), la chroma (C) ou la teinte (h) dans le but d'obtenir une similarité entre un motif uniforme et un motif modulé par une fréquence spatiale. Cette première étude a été réalisée avec un arrière plan noir dans le but de réduire le nombre de tests psychophysiques. Ces tests prennent beaucoup de temps d'une part pour la construction des motifs et d'autre part pour le recrutement et le passage des observateurs.

La procédure de test est la suivante :

- Après différentes mesures sur sa vision (tests d'Ishihara, tests d'acuité visuelle) l'observateur est installé dans la salle psychophysique à la distance d'un mètre cinquante de l'écran.
- La procédure de test lui est expliquée ainsi que les différentes tâches à accomplir.
- Le test commence, et l'observateur doit régler un et un seul des trois critères (J,C,h) du motif possédant une fréquence spatiale dans le but d'obtenir la même couleur que le motif de référence. Le critère que l'utilisateur peut régler lui est inconnu pour ne pas influencer son jugement.
- Ce dernier point est répété 63 fois avec trois couleurs primaires, sept fréquences différentes et 3 critères de sortie du CIECAM02 (J,C,H)

De plus la séquence de test est tirée aléatoirement à chaque nouvel observateur.

2.4 Observateurs

Pour obtenir des statistiques correctes, l'ITU [ITU500] recommande d'avoir au minimum 15 observateurs. Le tableau 1 donne La répartition des observateurs qui ont passé le test en fonction de leur sexe et de leurs affections visuelles.

	Normal	Myope	Autres affections	Total
Homme	10	5	1	16
Women	2	3	0	5
Total	12	8	1	21

Tableau 1 – *Tableau des observateurs*

3 Résultats et discussion

Cette section décrit les résultats de notre expérience ainsi que le développement de notre modèle.

Les figures 5, 6 et 7 montrent l'écart perçu par l'observateur entre le motif uniforme et celui avec une fréquence spatiale sur l'un des trois critères J, C et h. Ces figures illustrent le fait que sur un fond noir la différence perçue sur la clarté et sur la chroma augmentent en fonction de la fréquence spatiale. La figure 7 montre un comportement angulaire, ce qui confirme la nature de la teinte.
Ce comportement était prévisible à cause de l'augmentation de la répartition du noir sur le motif. Cependant cette expérience permet de quantifier cette augmentation qui n'est pas linéaire. Quelques phénomènes incompréhensibles sont encore visibles aux fréquences moyennes (9 à 11 cpd). Ces problèmes seront étudiés dans de prochaines expériences. La seconde partie de l'étude permet de regarder l'influence des fréquences spatiales séparemment sur les hommes et les femmes. La figure 8 illustre le fait que les femmes semblent plus sensibles que les hommes pour une variation des fréquences spatiales sur la teinte du rouge.

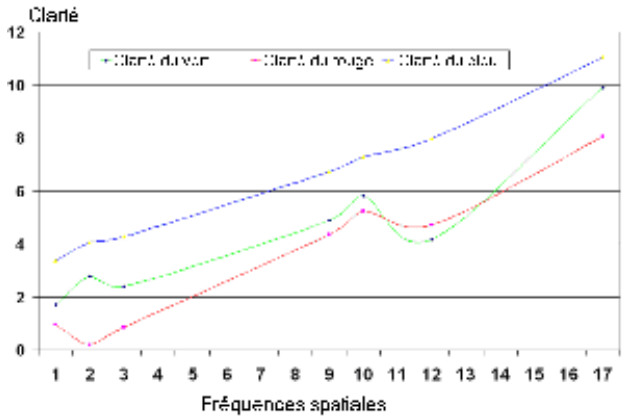


Figure 5 – *Clarté perçue en fonction de la fréquence spatiale sur un arrière plan noir*

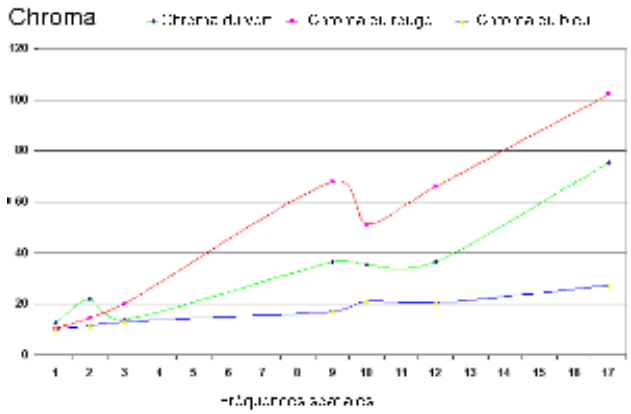


Figure 6 – *Chroma perçue en fonction de la fréquence spatiale sur un arrière plan noir*

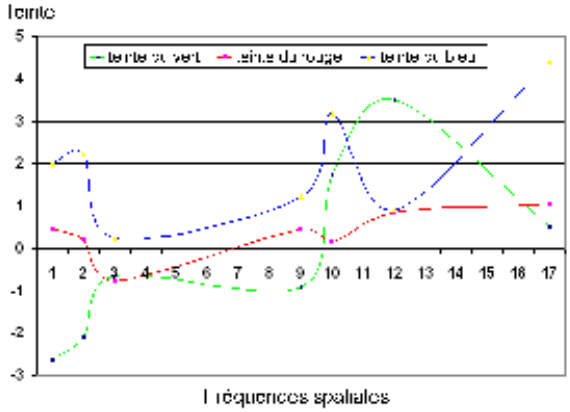


Figure 7 – *Teinte perçue en fonction de la fréquence spatiale sur un arrière plan noir*

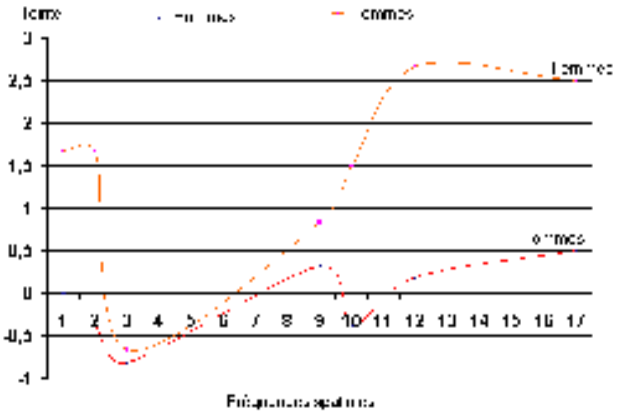


Figure 8 – *Teinte du rouge perçue par les hommes et les femmes en fonction de la fréquence spatiale pour un arrière plan noir*

Les données obtenues par la campagne d'évaluation ont permis la construction d'un premier modèle. Pour certaines fréquences, un grand écart-type a été mesuré entre les différents observateurs, c'est pourquoi le critère de chauvenet [7] a été utilisé pour rejeter les valeurs incohérentes. Le modèle simple choisi est une courbe de degré 2.

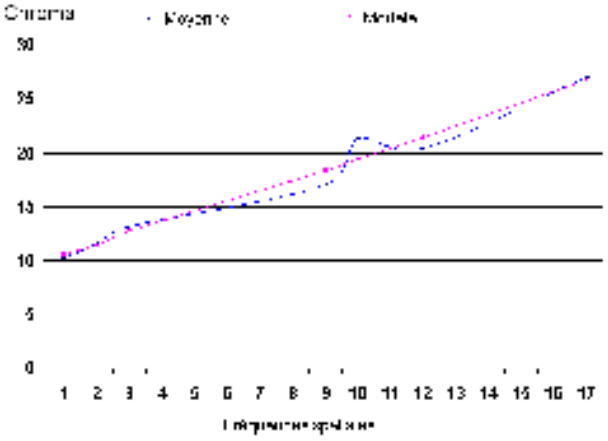
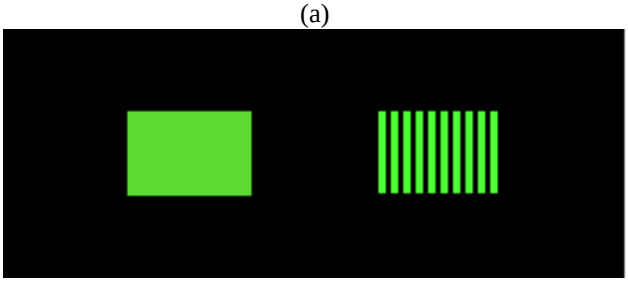


Figure 9 – *Courbe obtenue pour la chroma du bleu et son modèle*

La figure 9 montre un exemple de modélisation de la chroma du bleu. On peut voir que le modèle suit la courbe de résultat obtenue par le test. Cependant on peut observer qu'il y a certains problèmes aux alentours des fréquences spatiales 10 et 12.

Ce modèle a été intégré au CIECAM02 afin de corriger des motifs modulés par une fréquence spatiale. Un exemple est donné par la figure 10. Dans cette figure, il est possible de constater que le motif de la figure 10-b paraît plus similaire que celui de la figure 10-a.

(a)



(b)

Figure 10 – *(a) motif sans correction, (b) motif avec correction*

4 Conclusion

Dans cette contribution, une méthode basée sur des tests psychophysiques qui permettent de prendre en compte l'influence des fréquences spatiales sur l'apparence de la couleur et ses résultats ont été décrits. Résultats avec lesquels un premier modèle a été obtenu et intégré dans le CIECAM02. La correction obtenue avec ce modèle est très encourageante.
Il serait intéressant d'affiner les résulats, l'expérience et le modèle pour prendre plus précisemment en compte un comportement du SVH en fonction des fréquences spatiales. L'influence de l'arrière plan est aussi une étude à mener et des tests sur ce dernier point sont en cours. Une fois toutes ces expériences menées et les résultats obtenus validés, il serait intéressant d'intégrer une correction de l'apparence de la couleur en fonction des fréquences temporelles, ce qui est essentiel dans le domaine de l'apparence pour des images animés.

Références

[1] M.D.Fairchild. *Color appearance model*. Addison-Wesley, Massachussets, 1997.

[2] N. Moroney, M.D. Fairchild, R.W.G. Hunt, C.J Li, M.R. Luo, , et T. Newman. The ciecam02 color appearance model. Dans *IS&T/SID 10th Color Imaging Conference*, pages 23–27, Scottsdale, Novembre 2002.

[3] Garette M. Johnson. The quality of appearance. *Munsell Colour Science Laboratory*, Janvier.

[4] M.R. Luo et R.W.G. Hunt. The structure of the cie 1997 colour appearance model (ciecam97s). Dans *Color Research and Application*, pages 138–146, University of Derby, Mackworth Rd., Derby DE22 3BL, England, Décembre 1998.

[5] Brian A. Wandell. *Foundations of vision*. Sinauer Associates, Inc, Massachussetts, 1995.

[6] ITU-R Recommendation BT.500-10. *Methodology for the subjective assessment of the quality of television pictures*. ITU, Geneva Switzerland, 2000.

[7] John R. Taylor. *The Study of Uncertainties in Physical Measurements*. Softback, Colorado, 1997.

Mesure rapide de similarités musicales

Perception du rythme

Luigi.lancieri, Lucille.Tanquerel

[1] France Telecom R&D

42 rue des coutures 1400 Caen

Résumé

Cet article décrit une technique de caractérisation rapide de documents sonores basée sur une mesure statistique de la variation du signal. Nous avons montré qu'un échantillonnage très limité des morceaux était suffisant pour obtenir une performance de la caractéristique raisonnable tout en étant 300 fois plus rapide à calculer qu'un échantillonnage complet. Nous avons réalisé une première validation de notre approche en mettant en évidence une corrélation de 0,7 entre la perception humaine du rythme et le rendu de notre caractéristique ainsi qu'une erreur de reconnaissance inférieure à 5%.

Mots clefs

Similarité musicale, rythme, variation.

1 Introduction

De nombreuses sources font état des besoins et du fort potentiel commercial lié à la gestion automatisée de documents sonores. Par exemple l'IFPI (International Federation of the Phonographic Industry) a annoncé que le chiffre d'affaires global des services de vente de musique numérique en ligne a été multiplié par 10 en 2004 par rapport à 2003. Les analystes sont très confiants et annoncent qu'en 2005-2006 ce type de services devrait générer un chiffre d'affaires de l'ordre 330 millions de dollars [12].

La description des caractéristiques sonores d'un document est un élément clé pour réaliser des traitements automatiques impliquant des données audio. Ce type de mesure peut être utile non seulement pour caractériser les données mais aussi pour décrire les goûts musicaux des usagers sur la base de leurs activités d'écoute. Ces techniques deviennent critiques compte tenu de la quantité croissante de documents sonores, que ce soit sur le Web ou dans les bases de données musicales des fournisseurs de contenus. De nombreux travaux ont été réalisés dans ce domaine mais les techniques de traitement restent lourdes à mettre en œuvre et manquent de standards.

L'objectif de ce document est de décrire une méthode permettant de caractériser de manière compacte et rapide le rythme associé à un fichier sonore par l'extraction de caractéristiques physiques réparties sur le fichier (analyse spectrale du signal). L'innovation de notre proposition porte sur l'organisation de l'extraction des échantillons et sur le mode d'analyse pour fournir très rapidement une signature représentative de la nature rythmique du contenu musical.

L'organisation de l'extraction définit la manière dont les échantillons sont prélevés. Il paraît possible, par exemple, de déterminer le spectre sur tout le fichier musical ou seulement sur la première minute. Notre proposition vise à réaliser un échantillonnage statistique séquentiel minimal réparti sur le fichier sonore selon une loi de probabilité particulière. Le principe de cette proposition est basé sur le postulat que la collecte d'une faible quantité d'échantillons de petite durée suffit pour avoir une information résumant de manière efficace le rythme perçu. L'état de l'art montre, par exemple, qu'un individu est capable de reconnaître un genre musical dans 70 % des cas après avoir écouté seulement 3 secondes d'une bande son [1]. Notre méthode de validation repose d'une part sur la comparaison de la signature rythmique avec la perception humaine et d'autre par sur une mesure d'erreur de reconnaissance objective. Dans ce dernier cas, nous montrons que la signature rythmique permet de comparer les morceaux entre eux et d'identifier fidèlement les morceaux identiques même si ceux ci ne sont pas complets.

Dans la suite de ce document, après avoir détaillé les différents éléments de notre approche, nous proposons un état de l'art de travaux comparables ainsi qu'une présentation de quelques résultats.

2 Description générale

La figure suivante montre les bases du processus d'obtention de la signature à partir de l'analyse d'échantillons prélevés dans un fichier sonore. L'idée est de capturer l'image du balancement du spectre sonore tel que l'on peut le percevoir en observant le barre-graph d'un lecteur audio. Les échantillons à analyser sont collectés par triplets (E0, E1,..) de spécimen contigus de durée k. Dans cette première étude, chaque triplet est

collecté de manière aléatoire mais en respectant un ordre chronologique. C'est-à-dire que si on décide de prélever 10 triplets, la seule contrainte sera que le premier précède le second qui devra précéder le troisième, etc. L'espace de temps entre chaque triplet pourra être quelconque.

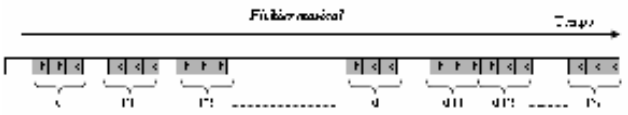


Figure 1 - *Collecte des échantillons par triplets dans un fichier sonore*

Sur chaque échantillon k de chaque triplet est calculée la répartition de fréquences au sens de Fourrier [2] puis, le coefficient directeur *p* de la droite de régression liant le niveau *(y)* à chaque classe de fréquence *(x)* du spectre. Cette droite de régression s'exprime de la manière suivante : $y = px + b$.

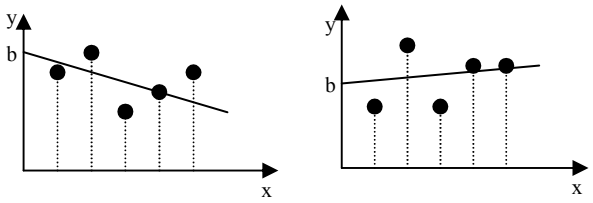


Figure 2 - *La pente du spectre de 2 éléments d'un triplet.*

L'analyse du comportement de *p* (pente de la droite de la figure 2) va contribuer à évaluer le comportement rythmique en mesurant le balancement du spectre sur une période, et en valeur moyenne sur les différents échantillons. Par référence à la mécanique, ce balancement, sa vitesse et son accélération sont évalués de la manière qui suit.

La première étape consiste à identifier le nombre de triplets ainsi que leur position dans le signal. Sur une fraction du fichier sonore, on extrait le premier triplet sur lequel on calcule les 3 spectres puis les coefficients directeurs des droites de régression. On obtient ainsi 3 valeurs de pente $(p1_1, p1_2, p1_3)$. La vitesse du balancement est obtenue en calculant l'écart entre 2 pentes consécutives. On obtient 2 valeurs de vitesses $(v1_1, v1_2)$, pour chaque triplet. L'accélération a_1, unique par triplet, est évaluée sur l'écart des vitesses. On recalcule ces données sur le triplet suivant et ainsi de suite jusqu'à la fin du fichier. A la fin de l'opération on dispose d'un ensemble de valeurs de coefficients $(p1_1, p1_2, p1_3, p2_1, p2_2, p2_3, ..., pn_1, pn_2, pn_3)$, de vitesses $(v1_1, v1_2, v2_1, v2_1, ..., vn_1, vn_2)$ et d'accélérations *(a1, a2, ..., an)* pour n triplets représentatifs du morceau de musique. Le comportement du balancement (position, vitesse et accélération) est obtenu par une combinaison des valeurs moyennes et de l'écart type de toutes ces données (pi, vi, $ai_)$.

Une difficulté importante que nous n'abordons que partiellement ici est de définir la proportion idéale de ces caractéristiques (pi, vi, ai). Pour commencer, nous n'utiliserons que la vitesse comme image du rythme. Dans d'autres travaux, en cours, nous évaluons l'influence des autres grandeurs (pente et accélération) pour optimiser la représentativité ou pour définir d'autres caractéristiques que le rythme. La signature du fichier musical est donc constituée par une valeur numérique combinant la moyenne et l'écart type de la vitesse. Cette valeur sera utilisée dans des métriques de comparaison avec l'évaluation humaine.

3 Etat de l'art

Le procédé décrit dans ce document se distingue de l'art antérieur par une meilleure capacité descriptive rapportée aux ressources de calcul et de stockage nécessaires. La capacité descriptive est liée à l'évaluation de la rythmique par l'analyse de structure de balancement. Ces éléments n'ont pas besoin d'être obtenus sur tout le fichier sonore, un échantillonnage statistique limité suffit. La signature ne nécessite a priori que le stockage d'une quantité très limitée de données numériques (une seule ici). D'autre part, la signature sera quasiment indépendante du format ou de la qualité sonore du morceau, même si ce dernier est incomplet.

Les techniques existantes pour la caractérisation de fichiers musicaux et les recherches de similarités (MIR – Music Information Retrieval) sont très variées. Il existe trois principales approches : celles basées sur le traitement du signal, le filtrage collaboratif, et la fouille de données. Les approches basées sur le traitement du signal consistent à analyser directement le contenu du morceau (signal et spectre) et peuvent être appliquées à n'importe quel fichier audio. En général, ces caractéristiques sont modélisées par des systèmes d'apprentissage, et des comparaisons sont effectuées pour la recherche de similarités [3, 4]. Par exemple, dans ses travaux, Georges Tzenakis [3] extrait une liste de caractéristiques obtenues à partir de l'enveloppe du signal et des données spectrales, notamment le centroïd (mesure de la luminance spectrale), le rollof (mesure de la forme du spectre), le ZeroCrossings (nombre de fois où la courbe du signal passe par le zéro) et parfois même les MFCC (Mel-frequency spectral coefficients) [5], caractéristiques couramment utilisées dans la reconnaissance vocale. Ces caractéristiques sont calculées dans des fenêtres d'analyse successives de taille fixe et seulement sur les 30 premières secondes du morceau. Un autre exemple de technologie en matière d'empreintes acoustiques est la TRM (This Recognizes Music) [11]. Cette technologie a été mise au point par la société américaine Relatable. Concrètement, ce système permet la reconnaissance de morceaux de musique par analogie acoustique exploitant une empreinte de type "code barre audio" qui génère une signature unique. Dès

que l'empreinte numérique a été créée, elle est envoyée au serveur TRM, qui compare l'empreinte à celle d'une chanson existante dans la base de données d'un client. La dernière version commerciale du serveur TRM peut gérer plus de 5000 empreintes par seconde, ou jusqu'à plusieurs milliards de requêtes par jour.

Avec le développement du Web, d'autres techniques basées sur des données publiques ont émergé [6, 7]. Elles utilisent l'analyse de texte et des techniques de filtrage afin de combiner des données provenant de divers individus pour déterminer des similarités basées sur des informations subjectives. Les techniques de filtrage collaboratif sont basées sur la comparaison de profils utilisateurs et représentent la technique principale utilisée aujourd'hui dans les systèmes de recommandations (Amazon, AllMusicGuide, etc.). L'avantage du filtrage collaboratif est que c'est une technique relativement simple à implémenter. Le principal inconvénient est le fait qu'elle requiert un très grand nombre d'utilisateurs d'un système donné pour être significative. Les méta-données culturelles sont des informations décrivant l'opinion publique et les tendances culturelles provenant de divers textes non structurés associés aux contenus et produits par le public. L'utilisation de ces informations pour juger de la similarité entre artistes musicaux a l'avantage d'exploiter des données complémentaires largement distribuées.

En dehors de l'aspect proprement musical, certaines techniques (dont la nôtre) peuvent être utilisées dans un contexte de DRM (Digital Right Management). Un exemple d'avantage est l'identification de fichiers musicaux tronqués ou piratés qui ne pourrait pas forcement être pris en charge par des techniques de DRM plus traditionnelles comme le watermarking. Comparée aux systèmes traditionnels, une DRM basée sur la similarité acoustique a de nombreux avantages (facile à mettre en œuvre, mieux tolérée par l'usager final, ..) même si la fiabilité peut être plus limitée.

4 Mesure de performances

Pour évaluer la pertinence de notre méthode, nous utilisons 2 ensembles de morceaux de musiques différents. L'un contrôlé, l'autre composé aléatoirement. Ces 2 sélections vont, dans un premier temps, être confrontées à l'opinion de 10 évaluateurs humains dont nous comparerons la perception à celle de notre système. Dans un second temps, le premier ensemble sera utilisé pour évaluer le taux d'erreur de reconnaissance et la robustesse de la signature (reproductibilité de la reconnaissance).

Le premier ensemble « calibré » comporte 26 morceaux de musique que nous avons choisis a priori de manière à couvrir une large plage de spectre rythmique. A titre d'exemple de morceaux rythmés citons « la Sonate n 9 pour piano » de Wolfgang Amadeus Mozart ou « The easy winner » de Scott Joplin. Pareillement pour les morceaux peu rythmés citons « l'Allemande » de la Suite pour violoncelle de Jean-Sébastien Bach, ou « Pièce pour haut-bois et harpes » de Gabriel Fauré. Nous avons volontairement limité le spectre des genres à la musique classique et au jazz de manière à ne pas introduire trop de paramètres dans l'étude. Pour augmenter la représentativité de cet ensemble nous avons réalisé 50 mesures de signature pour chacun des 26 fichiers (1300 signatures au total) sachant que chacune de ces 50 signatures peut être différente, en particulier pour les taux de couverture faibles. Il est en effet important de vérifier que toutes les signatures d'un même morceau sont cohérentes. Le second ensemble n'est pas calibré et correspond à 50 morceaux de musique choisis aléatoirement parmi 700 figurant au programme actuel de quelques radios généralistes comme SKY FM.

En plus de la comparaison de ces 2 ensembles avec la perception humaine, nous déterminons la capacité intrinsèque de discrimination de la signature par le biais d'une matrice de confusion. Cette technique permet, en comparant 2 à 2 les signatures, de calculer les erreurs de reconnaissance (similitudes reconnues à tort) et de non reconnaissance (similitudes réelles non reconnues). Cette technique sera appliquée aux 1300 signatures du premier ensemble.

Avant d'évaluer les performances de notre méthode, nous étudions la sensibilité de la signature aux différents paramètres qui y sont liés. Pour mémoire, ces paramètres sont le taux de couverture du morceau (T : de 5 à 75 %) et la taille d'un élément du triplet de base (K : de 1024 à 16384 octets), (voir figure 1).

4.1 Capacité descriptive de la signature

L'écart type est une mesure intéressante de la stabilité et de la performance des résultats d'un processus. En effet un écart type faible implique qu'au travers des nombreux tests, les résultats sont très proches (reproductibilité). Dans notre cas, il se trouve que l'écart type de la vitesse est l'élément de base de la signature. Pour évaluer l'influence de cette composante aux différents paramètres (K, T), nous l'agrégeons pour tous les morceaux. C'est donc l'influence de K et T sur cet agrégat que nous évaluons. L'agrégat est produit de la manière suivante. Pour chaque morceau de musique nous calculons l'écart-type EC correspondant à chaque morceau (i.e écart type de la vitesse). Comme nous calculons 50 signatures pour chaque morceau et pour un taux de couverture donné, nous obtenons par exemple pour une couverture de 5% :

Morceau 1 (T=5%) : EC1-1,…. EC1-50
Morceau i (T=5%) : ECi-1……ECi-50
Morceau 26 (T=5%) : EC26-1….EC26-50

Nous calculons ensuite pour chaque morceau la moyenne Mi des (ECi-1,.., ECi-50) et enfin ME, l'agrégat correspondant à la l'écart-type des Mi. Les courbes qui suivent montrent comment ME est influencé par la taille de l'échantillon élémentaire (K) ainsi que du taux de couverture du morceau (T). Naturellement, cette influence est moyennée mais elle permet de se faire une opinion globale.

Nous interprétons ces courbes de la manière suivante. Une discrimination importante entre les différents morceaux implique une valeur de ME élevée. A la limite, une valeur de ME nulle, indique que chaque morceau produit une caractéristique identique aux autres ce qui implique un pouvoir de discrimination nul.

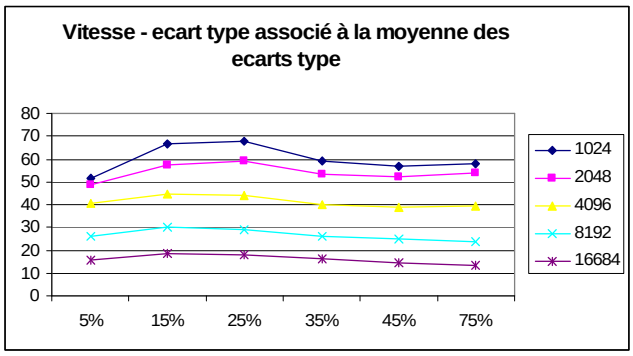


Figure 2 : Evolution de l'agrégat ME (ord) en fonction du taux de couverture (abs) et de la taille de l'échantillon élémentaire (paramètre).

On constate de manière assez prévisible qu'une taille d'échantillon élémentaire importante engendre une stabilité de la métrique. Ceci est vérifié par le fait que ME est au minimum et varie peu en fonction du taux de couverture pour une taille de 16684 octets. Ce résultat s'explique par l'effet d'intégration produit par l'évaluation du spectre sur des échantillons larges. A la limite, une taille d'échantillon élémentaire très grande recouvrant, par exemple, la première moitié d'un morceau de musique aurait une très forte probabilité de produire une métrique quasi identique (écart-type quasi nul) comparée à celle de la seconde moitié. Une taille d'échantillon trop importante est donc à proscrire si l'on souhaite une métrique représentative du contenu. De la même manière, un taux de couverture trop faible ou trop élevé finit par être pénalisant.

Ce qui est aussi intéressant dans ce graphique c'est la mise en évidence d'une relation non linéaire entre le taux de couverture et le niveau de discrimination de la métrique. Ceci implique qu'au-delà d'un certain niveau de taux de couverture, le gain en performance de discrimination s'affaiblit. C'est ce que l'on peut observer par une valeur de ME quasi identique entre 15 et 25 %, puis décroissante ensuite. Ainsi non seulement l'utilisation d'un taux de couverture important est pénalisant en terme de temps de calcul mais en plus il diminue la performance de la métrique. Le raisonnement est de même nature pour des valeurs faibles. L'idéal semble être un taux compris entre 10 et 20 %. En réduisant ce taux, on affaiblit les performances de la discrimination mais de manière très limité comparé au gain en temps de calcul. En effet, en prenant 3 fois moins d'échantillons (passage de 15% à 5% du taux de couverture) on ne réduit la « performance » de la catégorisation que de l'ordre de 20 %. (passage de 68% à 53 % de ME pour une taille d'échantillon de 1024). Ainsi, puisque notre objectif est d'obtenir une mesure rapide, nous utiliserons dans les tests de performances qui suivent un taux de couverture volontairement très faible compris entre 1 et 5 %.

4.2 Matrice de confusion

La matrice de confusion porte sur le premier ensemble et a pour objectif de comparer les échantillons deux à deux afin de tester la capacité de reconnaissance de la mesure de similarité. Pour une caractéristique et une métrique données, il devrait être possible de reconnaître les morceaux identiques et ceux qui sont différents. Le pourcentage de réussite permet d'apprécier la fiabilité du couple caractéristique-métrique. Nous évaluons cette performance pour l'échantillon des 26 morceaux représentés par les 1300 signatures. En plus de tester la fonction discriminante, ceci permet d'évaluer la capacité de l'algorithme à reconnaitre les mêmes morceaux échantillonnés différemment. Ceci est particulièrement intéressant dans le cas des taux de couverture très faible où la probabilité d'échantillonner les mêmes parties de chaque morceau est faible.

La matrice de confusion peut être déterminée pour un taux de recouvrement et une largeur d'échantillon donnés. La similitude entre 2 morceaux est déterminée par l'écart entre la signature de chaque morceau. La décision de similarité est prise en fonction d'un seuil en-deçà duquel les 2 morceaux sont considérés comme identiques. Le choix de ce seuil est naturellement fondamental, nous évaluons donc son influence. La courbe qui suit représente en pourcentage l'évolution de l'erreur d'association (courbe du haut) et de dissociation (en bas) en fonction de ce seuil. L'erreur d'association survient lorsque l'on considère que 2 morceaux sont identiques alors qu'ils sont différents. L'erreur de dissociation survient lorsque l'on considère que 2 morceaux sont différents alors qu'ils sont identiques. Ces deux visions inverses de la notion d'erreur de reconnaissance sont mesurées en fonction du seuil avec un taux de couverture de 5 % et une durée d'échantillon de 1024.

Dans la figure qui suit On observe clairement que les 2 types d'erreurs évoluent de manière inverse avec l'accroissement du seuil. En effet, il est logique de

constater qu'un seuil plus grand donne plus de chance de ne pas omettre de bons morceaux mais augmente aussi les chances de laisser passer de mauvaises associations. En fonction des souhaits on peut donc minimiser les erreurs d'association en utilisant un seuil minimal ou minimiser les erreurs de dissociation en maximisant le seuil.

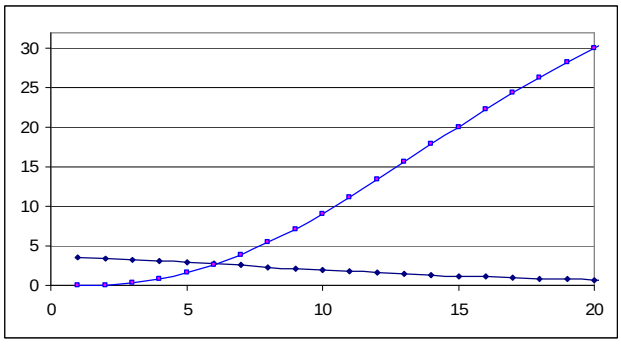


Figure 3 : *Pourcentage d'erreurs d'association (haut) et de dissociation (bas) en fonction du seuil de similarité.*

Un bon compromis semble être obtenu avec un seuil de 6. Pour cette valeur, on obtient une équivalence des 2 types d'erreurs autour de 3 %. Ces résultats sont encourageants car ils mettent en évidence un bon niveau de discrimination compte tenu du temps de calcul.

4.3 Comparaison avec la perception humaine

Pour mieux évaluer la pertinence de notre approche, nous avons soumis nos deux échantillons à un groupe de 10 individus auxquels nous avons demandé s'ils considéraient que les morceaux étaient rythmés ou non. Chaque individu a été interrogé de manière isolée sans contacts avec les autres. Nous avons ensuite calculé la moyenne des 10 avis afin d'obtenir pour chaque morceau une valeur comprise entre 0 et 1. La courbe qui suit exprime la relation entre le rythme perçu par les usagers et la valeur de la signature (normalisée) pour une couverture de 1%. Chaque point sur ce premier graphique représente un des 26 fichiers du premier ensemble.

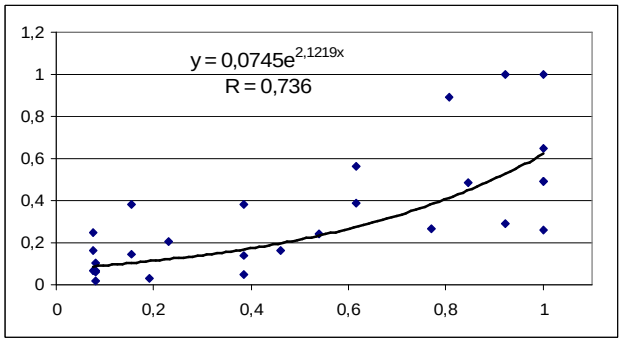


Figure 4-*: Relation entre le rythme perçu par les testeurs (abs) et la valeur de la signature (ord) pour chacun des 26 morceaux (premier ensemble).*

De la même manière le second graphique concerne le second ensemble de fichiers.

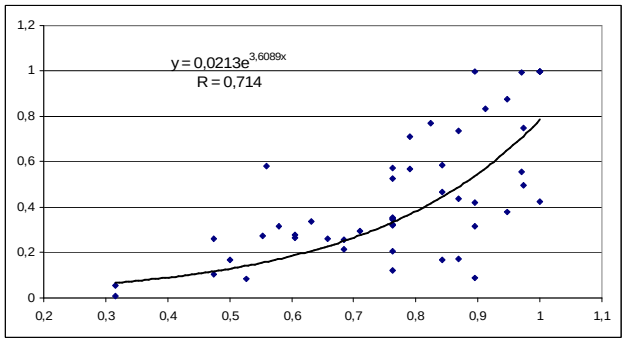


Figure 5-*: Relation entre le rythme perçu par les testeurs (abs) et la valeur de la signature (ord) pour chacun des 50 morceaux (second ensemble).*

Ces courbes permettent de faire plusieurs remarques. Tout d'abord on observe que les 26 échantillons (figure 4) sont répartis de manière assez homogène dans le spectre plus ou moins rythmé. Ceci est confirmé par la valeur moyenne du rythme perçu sur les 26 morceaux se situant à 0,49. On observe aussi que la signature basée sur la moyenne et l'écart-type de la vitesse permet une évaluation raisonnable du rythme avec, dans le cas d'une régression exponentielle, un coefficient de corrélation égal à 0,7. Ceci est confirmé par la seconde courbe avec des niveaux de performances comparables.

Compte tenu de la très faible quantité de signal prélevé, ces coefficients de corrélation doivent être considérés avec précautions. En effet, il est fort possible que 2 signatures de 1 % successives sur le même fichier correspondent en fait à 2 parties du signal complètement différentes. La matrice de confusion montre que malgré les différences, d'une signature sur l'autre, la cohérence est bien présente avec un bon niveau de discrimination. Les résultats de la comparaison avec l'avis des testeurs semblent plus nuancés, même s'ils restent très convenables compte tenu des temps de calcul (taux de couverture 1%). Sur ce point, il faut aussi considérer le caractère aléatoire et ambigu de l'évaluation humaine. Ceci dit, la moyenne des opinions réalisées sur les 10 usagers est de nature à limiter ce facteur.

4.4 Temps de calcul

Il est bien connu que les traitements multimédias sont lourds en temps de calcul. Une de nos motivations en abordant cette étude était d'ailleurs de limiter cette contrainte tout en conservant des performances raisonnables en termes de caractérisation des contenus.

Dans notre cas tous les traitements ont été réalisés sur un PC P4 datant de 2003. A titre d'exemple, le temps unitaire de traitement pour obtenir la moyenne et l'écart type pour

la vitesse et l'accélération avec un taux de couverture de 1% et une taille d'échantillon élémentaire de 2048 octets est de 0,12 secondes (0,08 sec si comme dans nos essais seule la vitesse est nécessaire). Ce temps passe à 36 secondes pour un taux de couverture à 75 % en conservant les autres paramètres identiques.

Sans compter le temps d'évaluation humaine, les mesures de l'influence de tous les paramètres (taux de couverture de 1 à 75 %, taille d'échantillons entre 1024 octets et 131172 octets, calcul des matrices de confusions, etc.) ont nécessité 700 h de traitement (équivalent à un mois de calcul continu). Cette durée importante est une des raisons qui nous ont poussés à limiter le nombre de morceaux de musique distincts évalués dans cette étude.

5 Conclusion

La caractérisation des fichiers musicaux représente un enjeu important dans la mesure où elle permet d'envisager l'indexation et la gestion automatisée et performante des contenus multimédias. Cette automatisation peut être appliquée de plusieurs manières impliquant le document sonore lui-même ou l'usager dans une perspective de modélisation de la perception musicale.

Dans ce contexte, nous avons développé et breveté une technique de caractérisation rapide basée sur la prise en compte de la variation du signal. Nous avons montré qu'un échantillonnage limité de séquences interne était suffisant pour obtenir une performance raisonnable de la caractéristique tout en étant plus de 300 fois plus rapide à calculer qu'un échantillonnage complet. Nous avons abordé la méthodologie de validation suivant deux angles différents : la matrice de confusion et la comparaison avec la perception humaine. Chacune de ces méthodes permet de conclure que la technique offre une représentation cohérente des fichiers sonores.

L'évaluation de notre algorithme en fonction des différentes variables d'influence comme le taux de couverture ou la taille des échantillons internes a nécessité une période de traitement longue. Cette contrainte et la volonté de prendre en compte l'évaluation humaine explique le nombre limité d'échantillons musicaux pris en compte dans cette expérience. Dans les phases ultérieures de nos travaux nous envisageons de valider ces résultats sur la base d'une plus grande quantité de fichiers, mais en limitant l'étendue des variables aux valeurs identifiées comme pertinentes (e.g taux de couverture 1 à 5 %).

Par ailleurs, il nous semble possible d'optimiser la représentativité de la signature en combinant de manière plus pertinente les diverses composantes extraites de notre approche.

Références

[1] Perrot, D., and R. O. Gjerdingen. Scanning the dial: An exploration of factors in identification of musical style. Research notes. Department of Music, Northwestern University, Illinois, USA. 1999

[2] Oppenheim, A. and Schafer, R. Discrete-Time Signal Processing. Prentice Hall. Edgewood Cliffs, NJ. 1989.

[3] George Tzanetakis, George Essl, Perry Cook. Automatic musical genre classification of audio signals. ISMIR, 2001

[4] Cory McKay, Ichiro Fujinaga. Automatic genre classification using large high-level musical. ISMIR, 2004

[5] Hunt, M., Lenning, M., and Mermelstein, P. Experiments in syllabe-based recognition of continuous speech. In Proceedings of International Conference on Acoustics, Speech and Signal Processing, 1996, 880-883

[6] Mark Zadel et Ichiro Fujinaga. Web services for music information retrieval ISMIR 2004

[7] François Pachet, Gert Westermann, Damien Laigre. Musical Data Mining for Electronic Music Distribution ISMIR, 2004

[8] http://www.servicedoc.info/article.php3?id_article=174

[9] http://www.xrml.org

[10] http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm

[11] http://rm.relatable.com/

[11] http://www.clubic.com/actualite-18188-le-marche-de-la-musique-en-ligne-multiplie-par-10-.htmle

Extraction de traits caractéristiques perceptuels dans des images couleur et métriques de similarité associées

Mathieu Carnec Patrick Le Callet Dominique Barba

Équipe Image et Vidéo Communications/IRCCyN

École polytechnique de l'université de Nantes
Rue Christian Pauc, La Chantrerie, 44 NANTES Cedex 3

{mathieu.carnec, patrick.lecallet, dominique.barba}@univ-nantes.fr

Résumé

Dans cet article, plusieurs traits caractéristiques perceptuels sont extraits pour décrire des images. Leurs méthodes d'extraction sont détaillées. Des métriques de similarité permettant de comparer, entre autres, ces traits caractéristiques sont également données. Ces traits caractéristiques et métriques de similarité ont été utilisés dans des applications permettant de mesurer la qualité d'images ou de reconnaître des visages. Les performances obtenues en termes d'évaluation de qualité montrent l'utilité de ces traits caractéristiques et de leurs métriques de similarité qui peuvent être employés dans d'autres domaines comme la description d'images pour l'indexation ou la reconnaissance de formes.

Mots clefs

Traits caractéristiques, métriques de similarité, système visuel humain, description réduite d'images, évaluation de qualité.

1 Introduction

Les images, en tant que tableaux bidimensionnels de pixels, sont bien souvent des ensembles d'informations de trop bas niveau pour pouvoir aider à la décision. Il est alors nécessaire de transformer ces informations dans un espace approprié ou de construire des représentations d'images de plus haut niveau. Dans ce deuxième cas, les types de traits caractéristiques qui représentent une image et leur utilisation sont profondément liés. Pour des applications visant des utilisateurs et pour des décisions ayant un rapport avec la perception humaine, il peut être intéressant d'extraire des traits caractéristiques comparables à ceux utlisés par le système visuel humain (SVH). De tels traits caractéristiques perceptuels ont montré leur utilité dans l'évaluation de qualité des images.

Cet article présente une description du SVH et des stimuli auxquels il est sensible. Puis des traits caractéristiques perceptuels sont présentés et leurs méthodes d'extraction sont détaillées. Ensuite, des métriques de similarité sont données, permettant de comparer ces traits caractéristiques. Puis, des applications possibles de ces traits caractéristiques perceptuels et de ces métriques de similarité sont proposées. Enfin, les performances de certaines de ces applications en évaluation de qualité sont données.

2 Le système visuel humain

Le système visuel humain (SVH) est un ensemble d'éléments fortement reliés les uns aux autres. Si nous négligeons les contre-réactions ("feedbacks"), le SVH peut être décrit par le modèle fonctionnel présenté figure 1 [1] [2].

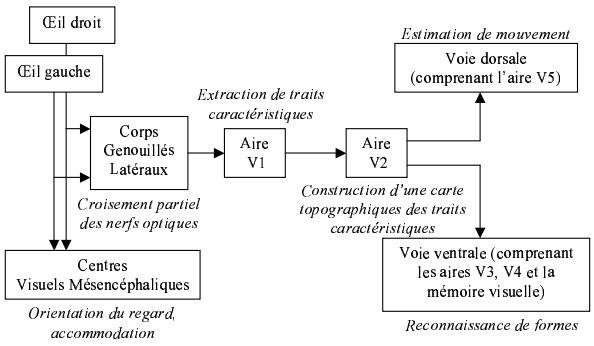


Figure 1 – *Modèle fonctionnel du système visuel humain*

Dans ce modèle, nous pouvons distinguer les éléments suivants :

– les yeux : au fond de chaque oeil, les capteurs photoélectriques de la rétine acquièrent l'information visuelle et l'envoient au cortex visuel par les nerfs optiques ;
– les corps genouillés latéraux (CGL) : ils sont le siège d'un croisement partiel de nerfs optiques tel qu'à leur sortie des CGL, chaque nerf optique transporte l'information de la partie du champ visuel qui lui est opposée (le nerf optique gauche porte les informations de la partie droite du champ visuel *et vice versa*) [1] ;

– l'aire V1 : elle extrait des traits caractéristiques à partir des informations fournies par les nerfs optiques, détecte des lignes orientées [3] [4] et code l'orientation des bordures des objets [1] ;
– l'aire V2 : elle repère les traits caractéristiques (extraits par l'aire V1) les uns par rapport aux autres en construisant une carte d'organisation topographique de ces traits caractéristiques [5] ;
– la voie ventrale : elle effectue une reconnaissance de formes en comparant les représentations fournies par l'aire V2 à celles stockées dans une mémoire visuelle (il existe en fait au moins deux mémoires visuelles différentes : une à très court terme et une à long terme) [6] ;
– la voie dorsale : elle estime le mouvement des objets présents dans le champ visuel grâce aux représentations topographiques fournies par l'aire V2 [5] ;
– les centres visuels mésencéphaliques : ils sont constitués du colliculus supérieur qui contrôle l'orientation du regard et les saccades occulaires mais également du pretectum qui réalise l'accomodation (courbure du cristallin en fonction de la distance entre l'oeil et l'objet regardé) et l'adaptation à la lumière (adaptation de la perception en fonction de la dynamique des signaux parvenant à la rétine) [2].

Parmi les contre-réactions qui n'apparaissent pas dans le modèle présenté figure 1, nous pouvons citer une contre-réaction allant de la voie dorsale vers les centres visuels mésencéphaliques afin d'orienter le regard vers un objet en mouvement. Mais il existe de nombreuses autres contre-réactions comme celles qui permettent de concentrer l'attention sur des objets difficiles à reconnaître ou nécessitant l'exploration d'un champ visuel important.

3 Traits caractéristiques perceptuels

L'extraction de traits caractéristiques dans des images peut permettre de mesurer la qualité de celles-ci. Certains auteurs extraient des traits caractéristiques directement à partir du signal d'image [7] [8]. Afin de prendre en compte les spécificités du SVH, plusieurs traitements consécutifs peuvent être appliqués à une image afin d'en obtenir une représentation perceptuelle appropriée à l'extraction des traits caractéristiques recherchés. Dans cet article, la transformation d'une image en représentation perceptuelle modélise le dispositif d'affichage mais aussi des phénomènes importants de la vision humaine comme la sensibilité au contraste et l'effet de masquage.

3.1 Représentation perceptuelle d'une image

De manière à prendre en compte l'image vue par l'observateur, le comportement du dispositif d'affichage est modélisé par une fonction non linéaire (appelée couramment "fonction gamma"). Cette fonction transforme les données d'une image (représentée en composantes RVB) en luminances physiques (exprimées en candella par mètre carré). Pour la composante rouge, la fonction utilisée est la suivante :

$$L_{\mathrm{R}} = \mathrm{Offset}_{\mathrm{R}} + L_{\mathrm{R,max}} * (\frac{R}{R_{\mathrm{max}}})^{\gamma_{\mathrm{R}}} \qquad (1)$$

avec :
– L_{R} : luminance physique de la composante rouge,
– $\mathrm{Offset}_{\mathrm{R}}$: valeur de la luminance pour une composante rouge nulle (typiquement 0.23 cd/m^2) ,
– $L_{\mathrm{R,max}}(R)$: luminance maximale de la composante rouge,
– R_{max} : valeur maximale de la composante rouge (255 pour un codage sur 8 bits),
– γ_{R} : paramètre dependant de l'écran utilisé (typiquement 2.4).

Les relations entre L_{V} et V (pour la composante verte) et entre L_{B} et B (pour la composante bleue) sont respectivement du même type mais avec des différences dans les valeurs de $L_{\mathrm{R,max}}$, $L_{\mathrm{V,max}}$ et $L_{\mathrm{B,max}}$. Pour un moniteur TV à tube cathodique (CRT) standard calibré selon la recommandation ITU-T BT.500-11 [9]), ces valeurs sont les suivantes :
– $L_{\mathrm{R,max}} = 18.310\, cd/m^2$
– $L_{\mathrm{V,max}} = 58.672\, cd/m^2$
– $L_{\mathrm{B,max}} = 9.376\, cd/m^2$

Dans un deuxième temps, les luminances (L_{R}, L_{V}, L_{B}) sont converties dans l'espace colorimétrique de Krauskopf [10]. Cet espace a été validé comme espace colorimétrique perceptuel [11] car c'est dans cet espace que la présence d'un stimuli sur un axe colorimétrique perturbe le moins possible la perception d'un signal porté par un autre axe. L'espace de Krauskopf contient une composante achromatique notée A et deux composantes chromatiques antagonistes notées $Cr1$ (axe rouge-vert) et $Cr2$ (axe bleu-jaune). La conversion s'effectue à l'aide de la transformation suivante :

$$\begin{pmatrix} A \\ Cr_1 \\ Cr_2 \end{pmatrix} = L_{\mathrm{max}} \begin{pmatrix} \frac{0.2244}{L_{R,max}} & \frac{0.6811}{L_{V,max}} & \frac{0.0942}{L_{B,max}} \\ \frac{0.0891}{L_{R,max}} & \frac{-0.0617}{L_{V,max}} & \frac{-0.0275}{L_{B,max}} \\ \frac{-0.1029}{L_{R,max}} & \frac{-0.2874}{L_{V,max}} & \frac{0.3903}{L_{B,max}} \end{pmatrix} \begin{pmatrix} L_R \\ L_V \\ L_B \end{pmatrix} \qquad (2)$$

avec $L_{\mathrm{max}} = L_{\mathrm{R,max}} + L_{\mathrm{V,max}} + L_{\mathrm{B,max}}$

Ensuite, seule la composante A va subir de nouveaux traitements alors que les composantes $Cr1$ et $Cr2$ resteront inchangées.

Les valeurs de la composante A (représentant les luminances achromatiques) sont divisées par la luminance achromatique moyenne afin de produire une image de contraste ("contraste global" de Daly [12]). Une fonction de sensibilité au contraste (CSF : *Contrast Sensitivity Function*) peut alors être appliquée. Cette CSF est de la forme filtre passe-bande et modélise la sensibilité du

SVH aux fréquences spatiales de l'image. La sensibilité au contraste d'un stimulus de fréquence f et d'orientation θ est notée $CSF(f, \theta)$. Elle est égale à l'inverse du seuil différentiel de visiblité (SDV) pour cette fréquence f et cette orientation θ, le SDV étant la différence minimale d'amplitude entre un stimulus et son voisinage pour que le stimulus soit perçu. Dans le plan fréquentiel 2D, la CSF est modélisée par la tranformation suivante [12] :

$$CSF(f,\theta) = min\left(S(\frac{f}{bf_{\mathrm{a}} * bf_{\mathrm{e}} * bf_{\theta}}, l, s), S(f, l, s)\right) \quad (3)$$

avec :

- f : fréquence spatiale radiale (en cycles par degré visuel),
- θ : orientation (en degrés),
- l : luminance d'adaptation (en cd/m^2),
- s : aire de l'image (en degrés^2),
- $S(f,l,s) = ((3.23 * (f^2 * s)^{-0.3})^5 + 1)^{\frac{1}{5}} * A_{\mathrm{l}} * 0.9 * f * e^{-\mathrm{B_l} * 0.9 * f} * \sqrt{1 + 0.06 * e^{\mathrm{B_l} * 0.9 * \mathrm{f}}}$
- $A_{\mathrm{l}} = 0.801 * (1 + \frac{0.7}{l})^{0.2}$
- $B_{\mathrm{l}} = 0.3 * (1 + \frac{100}{l})^{0.15}$
- $bf_{\mathrm{a}}, bf_{\mathrm{e}}, bf_{\theta}$: paramètres dependant de la distance d'observation, de l'excentricité et de l'orientation.

Les paramètres bf_{a}, bf_{e} et bf_{θ} sont donnés par les équations suivantes :

$$bf_{\mathrm{a}} = 0.856 * d^{0.4} \quad (4)$$

$$bf_{\mathrm{e}} = \frac{1}{1 + 0.24 * e} \quad (5)$$

$$bf_{\theta} = 0.15 * cos(4 * \theta) + 0.85 \quad (6)$$

avec :

- d : distance d'observation (en mètres),
- e : excentricité (en degrés),
- θ : orientation (en degrés).

L'image qui résulte du filtrage par la CSF est ensuite décomposée en sous-bandes. Chaque sous-bande est accordée sur une gamme de fréquences spatiales et une gamme d'orientations. Pour des raisons pratiques, cette décomposition est appliquée dans le domaine spectral à l'aide de 17 filtres Cortex [13] mais des décompositions similaires peuvent être effectuées avec des fonctions de Gabor ou des ondelettes. Le plan fréquentiel 2D est partitionné de la manière représentée sur la figure 2. Ce partitionnement provient d'expériences de physcophysique effectuées dans notre équipe [14].

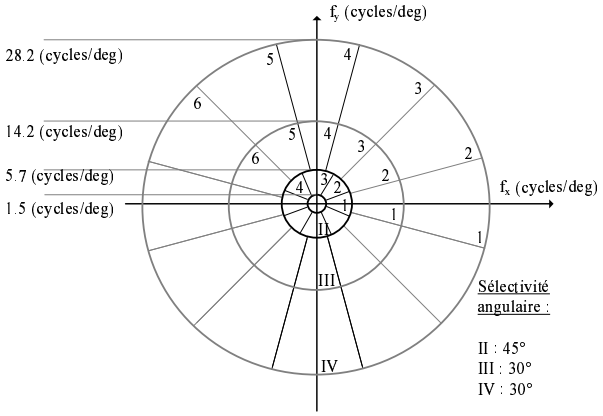


Figure 2 – *Partitionnement du plan fréquentiel pour la décomposition en sous-bandes perceptuelles de la composante achromatique A*

Cette décomposition en sous-bandes de la composante A permet de modéliser l'effet de masquage. Cet effet traduit l'accroissement de la difficulté à percevoir un signal (appelé signal masqué) lorsqu'il est en présence d'un autre signal (appelé signal masquant). Cet effet est maximal quand le signal masqué et le signal masquant sont proches en fréquences spatiales et en orientation. Ici, seul l'effet de masquage entre signaux de même sous-bande est pris en compte. Cette prise en compte consiste à calculer, pour chaque emplacement (x, y) de chaque image en sortie d'une sous-bande, l'élévation locale E du seuil de différentiel visiblité (SDV), cette élévation étant due à la présence du signal masquant. Seule une élévation du SDV est calculée car le SDV a déjà été pris en compte en utilisant la CSF. La forme de la fonction d'élévation par rapport à l'amplitude du signal masquant est représentée sur la figure 3.

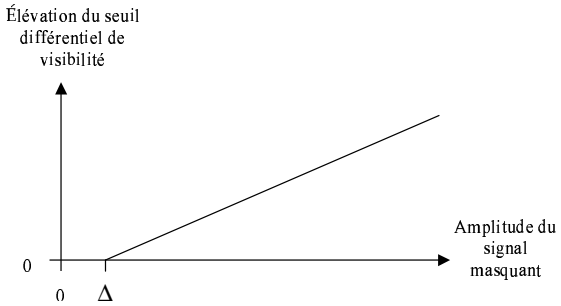


Figure 3 – *Élevation du seuil différentiel de visibilité en fonction de l'amplitude du signal masquant*

Ce calcul de l'élévation E du SDV utilise la relation suivante [12] :

$$\mathrm{E}_{\rho,\theta}(x,y) = (1 + (k_1 * (k_2 * |Isb_{\rho,\theta}(x,y)|)^{\mathrm{s}})^{\mathrm{b}})^{\frac{1}{\mathrm{b}}} \quad (7)$$

avec :
– $k_1 = 0.0153$,
– $k_1 = 392.5$,
– $Isb_{\rho,\theta}(x,y)$: amplitude du signal à l'emplacement (x,y) dans la sous-bande (ρ,θ) (ρ représentant la sélectivité en fréquences radiales et θ indiquant la sélectivité angulaire),
– $s,\ b$: paramètres dépendant de la bande de fréquences radiales considérée.

Cette élévation permet de connaître le seuil de visibilité local d'un stimulus en fonction de sa fréquence spatiale (CSF) mais aussi de son voisinage (masquage). Chaque échantillon des 17 sous-bandes de la représentation perceptuelle peut alors être normalisé par le seuil de visibilité local afin de réprésenter sa valeur perçue.
La représentation perceptuelle finale contient donc :
– 17 sous-bandes (accordées selon une gamme de fréquences radiales et une gamme d'orientations) pour la composante achromatique A,
– une composante chromatique $Cr1$,
– une composante chromatique $Cr2$.

Les composantes $Cr1$ et $Cr2$ n'ont pas été filtrées par une CSF et n'ont pas été décomposées en sous-bandes car elles contiennent peu d'information structurelle or c'est principalement ce type d'information qui nous intéresse. Des traits caractéristiques perceptuels vont maintenant pouvoir être extraits de la représentation perceptuelle.

3.2 Extraction des traits caractéristiques perceptuels

Dans l'aire V1 du SVH, chaque cellule est sensible à une gamme de fréquences spatiales, une gamme d'orientations et une partie du champ visuel (appelée "champ récepteur"). Les cellules de l'aire V1 sont sensibles à des contrastes orientés. Il est donc intéressant d'extraire de telles informations. Pour cela, nous allons extraire des segments orientés à des points P_i situés sur les extrema locaux dans les images en sortie des sous-bandes de la composante A.
Cette extraction utilise un algorithme original de *"stick growing"*. Cet algorithme, présenté figure 4, consiste à essayer de construire un segment centré sur P_i dans toutes les directions. Chaque point du segment doit être situé sur une valeur supérieure à un seuil exprimé en pourcentage de la valeur au point P_i. Une valeur du seuil de 50% de la valeur au point P, déterminée empiriquement, est utilisée car elle produit, sur des images naturelles variées, des segments comparables à ceux que l'on pourrait extraire manuellement pour représenter le contenu de l'image. Le résultat de l'algorithme est le segment le plus long. Une fois que la longueur L_i et l'orientation O_i de ce segment ont été déterminées, l'algorithme de *"stick growing"* est employé, mais cette fois uniquement dans la direction orthogonale au segment trouvé, afin de déterminer sa largeur notée W_i. Enfin, l'amplitude Am_i de la sous-bande au point P_i est mesurée. Cet algorithme de *"stick growing"* présente deux avantages. Tout d'abord, il permet de mesurer l'orientation locale avec une grande précision angulaire. De plus, il est plus rapide que des méthodes classiques comme les banques de filtres car cet algorithme se compose principalement de calculs d'adresses mémoire et de comparaisons à un seuil.
Finalement, les segments orientés sont décrits par leur orientation O, leur longueur L, leur largeur W et leur amplitude Am.
D'autre part, les valeurs moyennes des composantes A_i, $Cr1_i$ et $Cr2_i$ sont extraites au point P_i. Elles sont respectivement notées $\overline{A_i}$, $\overline{Cr1_i}$ et $\overline{Cr2_i}$. Chaque valeur moyenne est calculée sur un voisinage circulaire de rayon 0.1 degré visuel, ce qui correspond à un dixième du rayon de la zone fovéale. Ce rayon a été déterminé afin d'obtenir une valeur moyenne très locale de chaque composante.

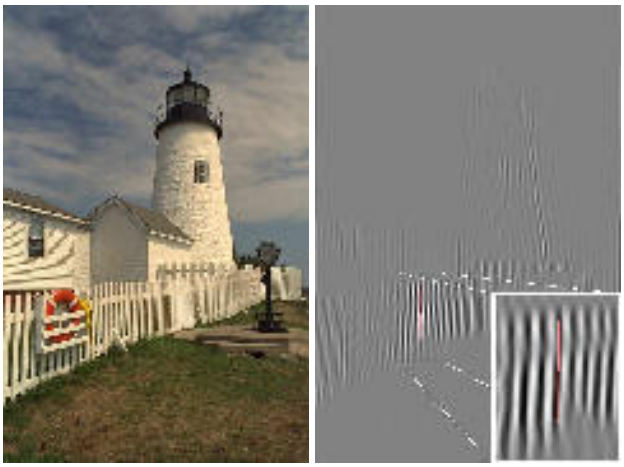


Figure 4 – *Extraction de segments orientés dans l'image d'une sous-bande sur l'image "lighthouse1" de la base d'images LIVE : image testée (gauche), image de la sous-bande d'indices $\rho = III$ and $\theta = 1$ (droite) et exemple de segment extrait (agrandissement)*

4 Métriques de similarité

Chaque type de trait caractéristique a une unité et une dynamique qui lui sont propres. C'est aussi le cas d'une différence en deux traits caractéristiques de même type (entre deux largeurs par exemple ou entre deux valeurs moyennes locales de la composante A). Pour comparer deux traits caractéristiques, il peut alors être intéressant d'avoir une métrique de similarité qui produise des résultats appartenant à une seule dynamique, quelque soit le type de traits caractéristiques comparés.

Pour cela, nous calculons un coefficient de correspondance qui est un différence absolue normalisée. En pratique, nous calculons la correspondance entre un trait caractéristique extrait d'une image de référence $Iref$ et son équivalent extrait d'une image dégradée $Ideg$. Le coefficient de correspondance $C(F_{\mathrm{R,i}}, F_{\mathrm{D,i}})$ entre deux traits caractéristiques ($F_{\mathrm{i,Iref}}$ et $F_{\mathrm{i,Ideg}}$ est défini comme la différence absolue entre les deux traits caractéristiques, normalisée par l'amplitude du trait caractéristique dans l'image de référence $Iref$ comme indiqué dans la relation suivante :

$$C(F_{\mathrm{i,Iref}}, F_{\mathrm{i,Ideg}}) = \max(0, 1 - \left|\frac{F_{\mathrm{i,Iref}} - F_{\mathrm{i,Ideg}}}{F_{\mathrm{i,Iref}}}\right|) \quad (8)$$

Néanmoins, l'équation précédente n'a pas de sens pour comparer deux orientations. En effet, pour une différence entre deux orientations (traits caractéristiques notés O_i), la normalisation ne peut se faire par la valeur de l'orientation dans l'image originale (par exemple, normaliser en divisant par 0° n'aurait aucun sens). Or, le plus grand écart angulaire possible entre deux structures est $\frac{\pi}{2}$. Par conséquent, nous utilisons la fonction $DiffNorm(O_i)$, périodique de période π, représentée sur la figure 5.

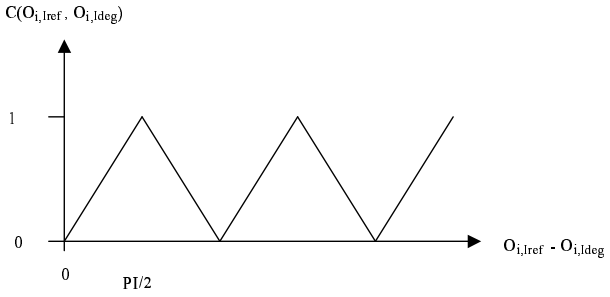


Figure 5 – *Fonction $C(O_{\mathrm{i,Iref}}, O_{\mathrm{i,Ideg}})$ (pour le trait caractéristique "orientation")*

Chaque coefficient de correspondance appartient donc à la dynamique $[0; 1]$. La valeur 1 montre une égalité entre les deux traits caractéristiques alors que la valeur 0 indique que la différence entre les deux traits caractéristiques est au moins égale à la valeur du trait caractéristique dans l'image de référence.

Les coefficients de correspondance ayant la même dynamique, quelque soit le type de trait caractéristique pris en compte, des traits caractéristiques de type différents peuvent alors être combinés au sein d'une métrique de similarité locale en utilisant leurs coefficients de correspondance. Par exemple, une métrique de similarité locale peut combiner l'orientation et la longueur des segments orientés, de la manière suivante :

$$SL_{\mathrm{i}} = \frac{1}{2}[C(0_{\mathrm{i,Iref}}, 0_{\mathrm{i,Ideg}}) + C(L_{\mathrm{i,Iref}}, L_{\mathrm{i,Ideg}})] \quad (9)$$

Les mesures de similarité locale peuvent ensuite être combinées (par une moyenne arithmétique ou géométrique par exemple) pour produire une mesure de similarité globale entre les deux images.

5 Exemples d'application et performances

Des applications logicielles qui utilisent ces traits caractéristiques et métriques de similarité sont disponibles gratuitement sur internet à l'adresse *http ://www.dcapplications.t2u.com/*. Le but de ces applications est de noter la qualité visuelle d'images qui ont été dégradées par différents traitements comme du codage (JPEG, JPEG2000, LAR [15]) ou du flou. La mesure des performances de ces applications montre qu'elles fournissent des notes de qualité qui sont hautement corrélées avec le jugement humain de la qualité. En effet, des coefficients de corrélation linéaires supérieurs à 0.91 ont été mesurés sur trois bases d'images notées par des observateurs humains lors de tests subjectifs. Le coefficient de corrélation linéaire est égal à 0.913 sur 150 images de la base de notre équipe (base IVC). Cette base a été constituée à partir de 10 images originales (de scenes naturelles) ayant subi des codages JPEG, JPEG2000, LAR et l'application d'un flou binomial. Le coefficient de corrélation linéaire est égal à 0.972 sur les 204 images JPEG de la base d'images notées LIVE [16] et égal à 0.957 sur les 198 images JPEG2000 images de cette même base. La base LIVE a été construite à partir de 29 images de scènes naturelles dégradée par un codage JPEG ou JPEG2000. Ces performances montrent que le critère de qualité mis au point (qui combine tous les traits caractéristiques présentés) donne des résultats bien meilleurs que ceux des critères classiques dans l'évaluation de qualité d'images (comme le PSNR ou la MSE) mais également de meilleurs résultats que les critères de l'état de l'art comme UQI [7] ou SSIM [8]. En effet, sur la base IVC et sur les images JPEG et JPEG2000 de la base LIVE, le critère UQI fournit des coefficients de corrélation linéaire de 0.809, 0.907 et 0.881 respectivement. Sur ces mêmes bases d'images notées, les coefficients de corrélation linéaire du critère SSIM sont respectivement 0.779, 0.958 et 0.942. Quant au PSNR, ses coefficients de corrélation sont respectivement de 0.633, 0.858 et 0.880. Les traits caractéristiques présentés et leurs métriques de similarité permettent donc de mesurer précisément de faibles différences (dues au codage ou au flou) entre deux images.

Une application a également été réalisée pour la reconnaissance de visages. Elle a montré que les traits caractéristiques présentés et leurs métriques de similarité présentés peuvent également permettre de reconnaître une

image parmi d'autres [17].

6 Conclusion

Cet article a présenté un modèle simplifié du système visuel humain et la construction d'une représentation perceptuelle d'une image. A partir de cette représentation perceptuelle, plusieurs traits caractéristiques sont extraits. Les méthodes d'extraction ont été décrites. Puis des méthodes de comparaison de ces traits caractéristiques ont été proposées. Ces traits caractéristiques et leurs métriques de similarité ont été utilisés dans plusieurs applications visant surtout à prédire la qualité d'images dégradées par différents codages. Les performances des applications présentées montrent l'utilité des traits caractéristiques perceptuels décrits et des métriques de similarité présentés. Ces traits caractéristiques pourraient donc servir à d'autres domaines comme la description d'images pour l'indexation ou la reconnaissance de formes. Plus généralement, les performances indiquées montrent l'intérêt d'intégrer une modélisation perceptuelle dans une application destinée à des utilisateurs humains.

Références

[1] Jean Bullier. Organisation anatomique et fonctionnelle des voies visuelles. Dans *École de printemps NSI (Neurosciences et Sciences de l'Ingénieur-Association des connexionnistes en thèse)*, pages 1–17, 6-10 mai 1997.

[2] Emmanuel Marilly. *Pré-processeur de vision fovéale. Application á la vision active.* Thèse de doctorat, Université du Havre, 1999.

[3] H.C. Nothduft et C.Y. Li. Texture discrimination : Representation of orientation and luminance differences in cells of the cat striate cortex. Dans *Vision Research, Vol. 25, No 1*, pages 99–113, 1985.

[4] D. H. Hubel et T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. Dans *J. Physiol. Lond., Vol. 160*, pages 106–154, 1962.

[5] William Bechtel et Robert N. McCauley. Heuristic identity theory (or back to the future) : The mind-body Problem Against the background of research strategies in cognitive neuroscience. Dans *Proceedings of the 21st Annual Meeting of the Cognitive Science Society*, 1999.

[6] P.H. Schiller. The effects of v4 and middle temporal (MT) area lesions on visual performances in the rhesus monkey. Dans *Visual Neuroscience, Vol. 10*, pages 717–746, 1993.

[7] Zhou Wang. Universal quality index. Dans *IEEE Signal Processing Letters*, volume 9, pages 81–84, March 2002.

[8] Zhou Wang, A. C. Bovik, H. R. Sheikh, et E. P. Simoncelli. Image quality assessment : From error measurement to structural similarity. Dans *Proceedings of ICIP (IEEE International Conference on Image Processing)*, volume 13, 2004.

[9] I.T.U. Recommandation BT.500-11. Methodology for the subjective assessment of the quality of television pictures. Rapport technique, I.T.U. CCIR, 2002.

[10] J. Krauskopf, D. R. Williams, et D. W. Heeley. Cardinal directions of color space. Dans *Vision Research*, volume 22, pages 1123–1131, 1982.

[11] Abdelhakim Saadane, Laurent Bedat, et Dominique Barba. Perceptual quantization of chromatic components. Dans *Proceedings of SPIE, Human Vision and Electronic Imaging III*, pages 202–209, July 1998.

[12] Scott Daly. *The Visible Differences Predictor : An Algorithm for the Assessment of Image Fidelity*, chapitre 14, pages 179–206. MIT Press, 1993.

[13] Andrew B. Watson. The cortex transform : Rapid computation of simulated neural images. Dans *Computer vision, graphics and image processing*, volume 39, pages 311–327, 1987.

[14] Hakim Senane, Abdelhakim Saadane, et Dominique Barba. Image coding in the context of a psychovisual image representation with vector quantization. Dans *Proceedings of ICIP (IEEE International Conference on Image Processing), Washington*, pages 97–100, October 1995.

[15] O. Déforges et J. Ronsin. Locally adaptative method for progressive still image coding. Dans *IEEE International Symposium on Signal Processing and its Applications*, 1999.

[16] H. R. Sheikh, Z. Wang, L. Cormack, et A. C. Bovik. Live image quality assessment database. http ://live.ece.utexas.edu/research/quality.

[17] Mathieu Carnec et Dominique Barba. Pattern recognition based on the perception and the behavior of the human visual system. Dans *Proceedings of Digital Signal Processing, Santorini, Greece*, 2002.

Contribution d'un modèle d'attention visuelle à l'évaluation sans référence de la qualité des images compressées JPEG

Rémi Barland Abdelhakim Saadane

IRCCyN-IVC, UMR n°6597 CNRS
École Polytechnique de l'Université de Nantes, rue Christian Pauc, La Chantrerie, BP 50609, 44 306, Nantes, France.

{remi.barland ; abdelhakim.saadane}@univ-nantes.fr

Résumé

A bas débit, les standards de compression actuels (JPEG) génèrent des dégradations visuelles pouvant gêner un observateur humain. De tels artéfacts sont généralement exploités pour évaluer la qualité d'une image sans référence. Cependant, même si les dégradations sont réparties dans l'image entière, certaines régions d'intérêt attirent le Système Visuel Humain et donc contribuent fortement à la détermination de la qualité perçue de l'image. Dans ce papier, nous proposons d'utiliser un algorithme simple d'attention visuelle pour pondérer des mesures de distorsions liées respectivement aux effets de bloc et de flou, principaux artéfacts crées par la compression JPEG. Ces mesures sont ensuite combinées afin de prédire une note de qualité. Une étude comparative des résultats prédits avec des notes de qualité issues de tests subjectifs, démontrent l'efficacité de l'approche.

Mots clefs

Évaluation de la qualité image sans référence, attention visuelle.

1 Introduction

L'apparition de nouvelles technologies et la recherche d'algorithmes de compression de plus en plus efficaces ont largement contribué au développement de nouveaux services d'acquisition et de diffusion d'image numérique. Pour tout nouveau service proposé, les standards actuels de compression image tels que JPEG, sont intégrés afin de minimiser la quantité d'information contenue dans une image, tout en offrant à l'utilisateur final, un maximum de qualité. La mesure en ligne de cette qualité perçue, qui dépend aussi bien des méthodes de codage utilisées que des erreurs de transmission, constitue aujourd'hui une des clés de l'explosion du tout numérique.

Les tests subjectifs représentent l'approche naturelle dans l'évaluation de la qualité : ils déterminent la note de qualité moyenne (MOS : « Mean Opion Score ») à partir des jugements humains de qualité. Certaines recommandations [1-3] de l'Union Internationale des Télécommunications (« International Telecommunication Union », ITU) spécifient pour chaque type d'applications les conditions d'observations, le choix des observateurs, le matériel à utiliser, les procédures à utiliser et les méthodes d'analyse des données. Bien que demeurant la référence dans le domaine, les tests subjectifs sont toutefois coûteux, longs et fastidieux à mettre en œuvre.

Les métriques de fidélité simples comme le PSNR, ou élaborées comme les métriques perceptuelles [4-8], qui connaissent également un essor considérable, ne sont pas du tout appropriées car elles nécessitent l'utilisation d'une image de référence qui n'est pas toujours disponible. Les métriques de qualité sans référence (NR : « No Reference ») représentent une alternative intéressante pour toutes les applications où il est nécessaire de contrôler en ligne, l'impact des dégradations induites par le processus de quantification. En général, les métriques NR combinent des mesures distinctes de dégradations en une seule [9], afin de prédire la qualité perçue. Dans le cas de la compression JPEG, deux défauts majeurs sont générés : les effets de blocs et de flou.

Dans la littérature, les mesures quantifiant les effets de blocs sont les plus nombreuses [10-16]. Cependant, peu intègrent des propriétés du Système Visuel Humain (SVH). Dans [13], la mesure d'effets de blocs prend en compte l'effet de masquage, sous la forme de pondérations calculées à partir des moyennes et écart-types des pixels situés aux frontières des blocs. Wang et al [11, 12] proposent de mesurer l'effet de blocs an analysant fréquentiellement le spectre de puissance d'un signal 1D constitué de la différence absolue de pixels consécutifs. Des effets de masquage liés à l'adaptation à la lumière et à la complexité des textures sont incorporés. En ce qui concerne l'effet de flou, la plupart des métriques proposées sont basées sur une approche signal [17-19]: Marziliano et al [18] proposent de mesurer le flou contenu dans une image de luminance, en calculant l'augmentation de la taille des contours à partir des points d'inflexion définissant le début et la fin de ces contours. Une autre approche [17] consiste à mesurer la précision des contours, en calculant le kurtosis local de chaque contour.

Dans [19], Marichal et al mesure l'effet de flou en exploitant l'information issue de l'histogramme des coefficients DCT.

Dans nos travaux antérieurs [20, 21], nous avons défini des mesures des effets de flou et de blocs, basées sur approche purement signal. Nous avons intégré dans le calcul de ces mesures de distorsion, des pondérations générées par un algorithme classant les régions contenues dans une image par importance perceptuelle [22]. Dans ce papier, nous proposons d'étendre ces travaux en intégrant une nouvelle carte d'importance perceptuelle. Un observateur humain n'effectue pas une analyse complète de l'image pour déterminer la qualité perçue mais plutôt, sélectionne certaines zones, apparaissant comme perceptuellement plus importantes. Ces zones ne correspondent pas forcément à des régions : des pixels appartenant à une même région de l'image n'ont pas forcement la même importance perceptuelle. A partir de ces deux observations, nous proposons d'identifier les zones perceptuellement importantes à l'aide d'un algorithme simple simulant l'attention visuelle pré attentive. La carte d'importance résultante va donc permettre d'identifier ces zones et d'intégrer dans le calcul des mesures de distorsion, leur contribution dans l'établissement de la qualité perçue des images compressées JPEG. Le papier est organisé de la manière suivante : la section 2 présente la structure de la métrique de qualité sans référence proposée, en détaillant la génération de la carte d'importance. La section 3 présente les expériences et les résultats. Enfin, des conclusions seront données dans la section 4.

2 Evaluation de la qualité sans référence des images JPEG

2.1 Structure de la métrique proposée

La figure 1 présente la structure de la métrique proposée : après une conversion couleur dans l'espace de Krauskopf [23], une carte d'importance est générée. Celle-ci est utilisée dans le calcul des mesures distinctes d'effets de blocs (BlM) et de flou (BM). La dernière étape de l'algorithme consiste à cumuler ces mesures de distorsion afin de déterminer une note de qualité prédite (pMOS).

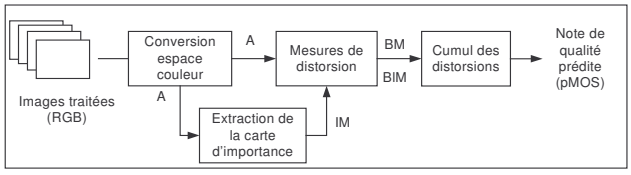


Figure 1 - *Structure de la métrique sans référence proposée*

2.2 Génération de la carte d'importance perceptuelle

Les observateurs humains focalisent leur attention sur certaines zones de l'image afin de déterminer la qualité perçue. Deux mécanismes sont mis en jeu : le premier appelé « bottom up » dépend du contenu de l'image, tandis que le second, « top down », se réfère à la tâche à accomplir. Dans cette section, la carte d'importance perceptuelle est générée par une approche « bottom up ». Dans la littérature, ce mécanisme de la vision humaine est simulé par des approches multi résolution décomposant l'image en différant canaux perceptuels liés aux sélectivités angulaire et radiale du SVH [24-27].

La figure 2 présente le fonctionnement de la génération de la carte d'importance. L'espace de couleur de Krauspoft [23] décompose l'image en trois canaux couleur : un achromatique A et deux chromatiques Cr_1 et Cr_2. Ici, seule la composante achromatique sera utilisée. La composante achromatique est ensuite décomposée en cinq résolutions grâce à une pyramide laplacienne [28] : cette opération permet d'obtenir des images de résolutions plus petites, tout en éliminant la corrélation spatiale existante dans l'image de départ.

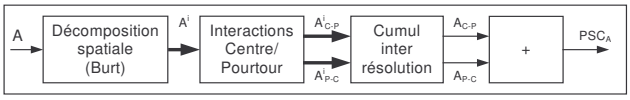


Figure 2 - *Algorithme de génération de la carte d'importance perceptuelle*

Ensuite, les interactions centre/pourtour sont incorporées : ces mécanismes de la vision humaine, réalisés par les champs récepteurs de l'aire corticale, détectent les discontinuités spatiales de l'image. Ainsi, pour chaque résolution *i*, l'émulation des champs récepteurs s'effectue de la manière suivante :

$$A^i_{C-P} = R(FPB(A^i, \sigma_1) - FPB(A^i, \sigma_2))$$

$$A^i_{P-C} = R(FPB(A^i, \sigma_2) - FPB(A^i, \sigma_1))$$

Où R est la fonction définie par $R(x)=0$ si $x \leq 0$, sinon $R(x)=x$. $FPB(.,\sigma)$ est un filtre passe-bas, défini par la convolution de la composante achromatique A^i à la résolution *i*, par un filtre gaussien ($\sigma_1=0.4$ et $\sigma_2=2.4$).

Pour chaque résolution *i*, nous obtenons deux images distinctes modélisant les interactions Centre/Pourtour (A^i_{C-P}) et Pourtour/Centre (A^i_{P-C}). Afin d'obtenir une carte de saillance de même résolution que l'image passée en entrée mais aussi pour cumuler les informations

obtenues par chaque résolution *i*, un cumul inter résolution est réalisé :

$$A^i_{X-Y}(m,n)=\frac{A^i_{X-Y}(m,n)^2}{Max(A^i_{X-Y}(m,n),A^{i-1}_{X-Y}(m,n))^2+\sigma^2}$$

Où σ est une constante assurant la non-nullité du dénominateur. Ce cumul inter résolution entre deux images de même interaction (Centre/Pourtour ou Pourtour/Centre) mais de résolutions différentes, permet de simuler les mécanismes excitateur et inhibiteur du SVH. A la sortie de ce contrôle de gain, nous obtenons deux images distinctes $A_{C\text{-}P}$ et $A_{P\text{-}C}$, qui sont sommées linéairement, afin d'obtenir une carte 2D de points saillants (PSC_A).

La carte de saillance finale est générée à partir de cette carte de points saillants. Les zones où il existe une forte concentration de points saillants, sont les régions ayant une forte probabilité d'attirer le regard de l'observateur humain. Ainsi, un filtrage itératif est réalisé afin de mettre en évidence ces zones. Il est effectué de la manière suivante :

$$IM_{k+1}=\frac{IM_k+FPB(IM_k,\sigma_{PS})}{Max(IM_k+FPB(IM_k,\sigma_{PS}))}$$

Où $FPB(.,\sigma_{PS})$ définit un filtre passe-bas réalisant une convolution avec un filtre gaussien ($\sigma_{PS}=5$). Au départ du schéma itératif, IM_0 est initialisée avec les valeurs de PSC_A ; au bout d'une vingtaine d'itérations, nous obtenons la carte finale d'importance (IM). La figure 3 présente un exemple de carte d'importance perceptuelle, générée par l'algorithme proposé.

Figure 3 – *A gauche, image originale « rapids » ; à droite, la carte d'importance perceptuelle associée*

2.3 Mesures de distorsions

2.3.1 Mesure de l'effet de blocs

L'effet de blocs peut être défini comme une discontinuité artificielle entre deux blocs voisins appartenant à une image compressée par le standard JPEG. Cet artéfact résulte d'une quantification sévère et indépendante des coefficients DCT de chaque bloc. Il est d'autant plus visible si le contraste avec les blocs voisins est important. A partir de ces deux constatations, nous avons défini dans [21], une mesure locale (*LBLM(k,l)*) de l'effet de blocs, pour le bloc (k,l):

$$LBlM\ (k,l)=\frac{R_H(k,l)+R_V(k,l)}{2}.S(k,l)$$

Où *S(k,l)* définit la sévérité de quantification et $R_H(k,l)$ (respectivement $R_V(k,l)$) est une valeur de renforcement liée au contraste horizontal (respectivement vertical) des blocs voisins.

Pour le bloc (k,l), la sévérité de quantification est exprimée de la manière suivante :

$$S(k,l)=\frac{1}{1+a\times STD\ (k,l)}$$

Où *STD(k,l)* est la valeur de l'écart type du bloc considéré et *a*, une constante. Le renforcement de contraste horizontal est calculé par la formule suivante :

$$R_H(k,l)=1+C_H(k,l)$$

Avec $C_H(k,l)=\frac{|\ A(k,l)-A(k,l-1)\ |+|\ A(k,l)-A(k,l+1)\ |}{2\times Max\{|\ A(k,l)-A(k,l-1)\ |,|\ A(k,l)-A(k,l+1)\ |\}}$

A(k,l) (respectivement *A(k,l-1)* et A(k,l+1)) est la valeur moyenne achromatique du bloc (k,l) (respectivement des blocs (k,l-1) et (k,l+1)). Le renforcement de contraste vertical utilise la même formulation mais basée sur les valeurs des blocs verticaux voisins.

La mesure finale de l'effet de blocs est obtenue en sommant les mesures locales d'effets de blocs pondérées par la carte d'importance IM :

$$BlM=\left(\frac{1}{NB_V\times NB_H}\sum_{k=1}^{NB_V}\sum_{l=1}^{NB_H}(IM(k,l)\times LBlM\ (k,l))^p\right)^{1/p}$$

Où NB_V (respectivement NB_H) représente le nombre de blocs verticaux (respectivement horizontaux) contenus dans l'image traitée. Dans notre implémentation, la valeur du paramètre *p* est égale à 2.

2.3.2 Mesure de l'effet de flou

L'effet de flou est engendré par l'atténuation des coefficients hautes fréquences, durant l'étape de quantification JPEG. Cet artéfact est visuellement défini, comme une distorsion globale sur l'image entière, caractérisée par une augmentation de l'étalement des contours et des détails spatiaux. Ainsi, en tenant des pondérations perceptuelles fournies par la carte d'importance IM, nous avons formulé dans [20], la mesure d'effet de flou pour l'image de taille $M\times N$, de la manière suivante :

$$BM=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}IM\ (i,j).A'_{Edge}\ (i,j).I_A^2(i,j)}{\sum_{i=1}^{M}\sum_{j=1}^{N}IM\ (i,j)A_{Edge}\ (i,j).I_A^2(i,j)}.\frac{\frac{N(A'_{Edge})}{M\times N}}{\frac{N(A_{Edge})}{M\times N}}$$

Où A_{Edge} est l'image binaire issue d'une détection de contours (par filtrage Sobel), A'_{Edge} est l'image complémentaire de A_{Edge}. $I_A(i,j)$ définit l'intensité du pixel (i,j) de la composante achromatique. Enfin, $N(A_{Edge})$

(respectivement $N(A'_{Edge})$) est le nombre de pixels non nuls de A_{Edge} (respectivement A'_{Edge}).

2.4 Prédiction de la qualité

Afin de déterminer la qualité perçue de l'image traitée, nous effectuons un cumul des mesures de dégradations BlM et BM. Celui-ci est défini comme une combinaison linéaire des mesures d'effets de blocs et de flou, auquel on ajoute le terme croisé associé. Ainsi, la note de qualité prédite (pMOS) est obtenue par :

$$pMOS=a_0+a_1.BlM+a_2.BM+a_3.BlM.BM$$

Où les a_i (i=0..3) sont des pondérations dont les valeurs ont été ajustées pour optimiser la performance de prédiction.

3 Expériences et résultats

Pour déterminer la performance d'une métrique de qualité, la corrélation entre notes prédites et notes subjectives (MOS) est analysée. Actuellement, VQEG (« Video Quality Expert Group ») préconise certains indicateurs [29] quantifiant des propriétés adéquates à l'analyse de la performance. Le coefficient de Spearman et l'erreur quadratique moyenne (RMSE : « Root Mean Square Error ») définissent l'exactitude de prédiction ; le coefficient de Spearman, la monotonie ; le pourcentage de points aberrants (« Outlier ratio »), l'uniformité et le coefficient Kappa, l'agrément entre notes prédites et subjectives. Cette section présente les résultats de corrélation basés sur le traitement d'une base d'images compressées JPEG [30]. Elle est constituée de 29 images couleurs originales (24 bits/pixel, RGB, 768x512 pixels) et des versions compressées respectives.

Pour les besoins de notre étude, cette base d'images est divisée en deux. La première partie sert à optimiser les pondérations a_i du cumul inter distorsion (section 2.4) : ces poids sont estimés en minimisant l'erreur quadratique moyenne entre les notes prédites à partir des mesures de distorsion (BM et BlM) et les MOS correspondants. La deuxième partie (elle comprend 75 images) sert à analyser la performance de la métrique proposée en calculant les différents indicateurs statistiques de performance.

Le tableau 1 présente les résultats de corrélation de chaque mesure de distorsion liée respectivement aux effets de blocs et de flou. On distingue deux cas :

- le calcul des mesures de distorsion n'intègre pas les pondérations de la carte d'importance perceptuelle (BlM et BM) ;
- ces pondérations sont prises en compte (BlM_{IM} et BM_{IM}).

En comparant la performance de chaque mesure de distorsion, on constate que la plupart des indicateurs proposés par VQEG sont meilleurs lors de l'intégration des pondérations perceptuelles. La corrélation de Pearson et l'erreur quadratique moyenne (RMSE) permettent de quantifier l'habilité à prédire la qualité perçue. Plus la corrélation de Pearson est proche de 1 et plus la RMSE est petite, meilleure est l'habilité de prédiction. En comparant les valeurs de la corrélation de Pearson et de la RMSE, on constate de meilleurs résultats pour les mesures de distorsion intégrant les pondérations perceptuelles. Ceci nous permet d'affirmer que l'utilisation de ces poids permet d'améliorer l'habilité de prédiction. La monotonie de prédiction est analysée à l'aide du coefficient de Spearman. Plus le coefficient de Spearman est proche de 1, meilleure est la monotonie. Les mesures de distorsion intégrant les pondérations perceptuelles obtiennent de meilleures valeurs, la monotonie de prédiction est donc meilleure. Le pourcentage de points aberrants est relativement constant pour toutes les mesures de dégradation. Enfin, le coefficient Kappa représente une mesure d'accord entre notes prédites et notes subjectives. S'il est supérieur à 0.4, on peut affirmer qu'il existe un bon agrément, résultat constaté pour chaque mesure de distorsion. De plus, nous remarquons que la valeur d'agrément est meilleure si les mesures de dégradation intègrent les pondérations perceptuelles.

	Pearson	RMSE	Spearman	Outlier	Kappa
BlM	0.910	0.865	0.921	12%	0.448
BlM_{IM}	0.932	0.626	0.943	13.3%	0.678
BM	0.912	0.711	0.90	13.3%	0.518
BM_{IM}	0.935	0.625	0.949	14.6%	0.732

Tableau 1 - *Performances respectives des mesures de distorsion liées aux effets de blocs (BlM) et de flou (BM), tenant compte ou non, de la carte d'importance perceptuelle IM*

Le tableau 2 présente les résultats des indicateurs de performance appliquée à la mesure conjointe des effets de blocs et de flou. Cette mesure est obtenue en réalisant le cumul inter distorsion défini dans la section 2.4. La métrique CM est basée sur les calculs des mesures de distorsion BM et BlM, n'intégrant pas les pondérations de la carte d'importance perceptuelle IM. Au contraire, la métrique CM_{IM} les intègre, puisqu'elle prend en entrée les mesures de distorsion BlM_{IM} et BM_{IM}.

En comparant les résultats de CM_{IM} par rapport à ceux de CM, on constate une nette augmentation de tous les indicateurs de performance. CM_{IM} présente une meilleure exactitude dans la prédiction de la qualité perçue (meilleurs coefficients de Pearson et RMSE) que CM. Pour les deux métriques, la relation de monotonie est respectée. De plus, CM_{IM} obtient une meilleure uniformité dans la prédiction de la qualité : le pourcentage de points aberrants (Outlier) est très faible. Enfin, la métrique CM_{IM}

obtient un meilleur accord entre notes prédites et subjectives.

	Pearson	RMSE	Spearman	Outlier	Kappa
CM	0.930	0.70	0.922	6.67%	0.518
CM_{IM}	0.965	0.485	0.954	1.3%	0.803

Tableau 2 – *Performances de la métrique proposée : CM utilise le cumul proposé de BlM et BM (mesures de distorsion n'intégrant pas les pondérations perceptuelles IM) ; CM_{IM} utilise le cumul proposé de BlM_{IM} et BM_{IM} (mesures de distorsion intégrant les pondérations perceptuelles IM)*

L'étude des tableaux 1 et 2 permet de confirmer l'apport de la carte d'importance perceptuelle dans un schéma de prédiction de la qualité. Cette carte permet d'identifier et de quantifier l'importance de certaines zones dans une image, susceptibles d'attirer le regard d'un observateur humain. La prise en compte de ces pondérations perceptuelles permet de mieux intégrer les corrélations spatiales existantes entre les pixels de l'image, dans le calcul des mesures de distorsion.

4 Conclusion

Dans ce papier, nous avons présenté la contribution d'une carte d'importance perceptuelle, générée par un algorithme simple simulant l'attention visuelle pré attentive, dans un schéma d'évaluation sans référence de la qualité des images compressées JPEG. La métrique proposée est basée sur la mesure respective des deux artéfacts les plus gênants, engendrés par la compression JPEG : les effets de blocs et de flou. L'apport de cette carte a été démontré en comparant les résultats de prédiction de métriques intégrant ou non, ces pondérations perceptuelles dans le calcul des mesures de distorsion. L'intégration de ces pondérations perceptuelles permet d'identifier des zones susceptibles d'attirer le regard humain mais aussi de quantifier leur importance perceptuelle. Afin d'affiner cette carte et peut être améliorer le schéma d'évaluation de la qualité présenté, nos travaux futurs ont pour but d'intégrer d'autres caractéristiques influençant la vision humaine (couleur, orientation, etc.) et d'analyser leurs apports respectifs.

Références

[1] ITU-R Recommendation BT.500-10, Methodology for the Subjective Assessment of the Quality of Television Pictures. 2000, ITU: Geneva.

[2] ITU-T Recommendation P.910, Subjective Video Quality Assessment Methods for Multimedia Application. 1999, ITU: Geneva, Switzerland.

[3] ITU-T Recommendation P.930, Principles of a Reference Impairment System for Video. 1996, ITU: Geneva, Switzerland.

[4] Teo, P.C. and D.J. Heeger. Perceptual Image Distortion. in IEEE International Conference on Image Processing. 1994. Austin, Texas USA.

[5] Lubin, J., A Visual Discrimination Model for Imaging System Design and Evaluation. Vision Models for target detection and recognition, 1995: p. 245-283.

[6] van den Branden Lambrecht, C. and J. Farrell. Perceptual Quality Metric for Digitally Coded Colour Images. in European Signal Processing Conference. 1996. Trieste, Italy.

[7] Winkler, S. A perceptual distortion metric for digital color images. in International Conference of Image Processing. 1998. Chicago, Illinois USA.

[8] Fontaine, B., A. Saadane, et al. Perceptual Quality Metrics: Evaluation of Individual Components. in International Conference on Image Processing. 2004. Singapore.

[9] Farias, M., S.K. Mitra, et al. Perceptual contributions of blocky, blurry and noisy artifacts to overall annoyance. in International Conference on Multimedia and Expo. 2003. Balitmore, Maryland USA.

[10] Meesters, L. and J.-B. Martens, A Single-ended Blockiness Measure for JPEG-Coded Images. Signal Processing, 2002. 82(3): p. 369-387.

[11] Wang, Z., H. Sheikh, et al. No Reference Perceptual Quality Assessment of JPEG Compressed Images. in IEEE International Conference on Image Processing. 2002.

[12] Wang, Z., A. Bovik, et al. Blind Measurement of Blocking Artifacts in Images. in IEEE International Conference on Image Processing. 2000.

[13] Wu, H.R., Z. Yu, et al. Impairment metrics for MC/DPCM/DCT encoded digital video. in Picture Coding Symposium. 2001. Seoul, Korea.

[14] Caviedes, J. and J. Jung. No-Reference Metric for a Video Quality Control Loop. in World Multi-Conference on Systems Cybernetics and Informatics Broadcasting Convention. 2001.

[15] Vlachos, T., Detection of Blocking Artifacts in Compressed Video. Electronics Letters, 2000. 36(13): p. 1106-1108.

[16] Gao, W., C. Mermer, et al., A De-Blocking Algorithm and Blockiness Metric for Highly Compressed Images. IEEE Transactions on Circuits and Systems for Video Technology, 2002. 12(12): p. 1150-1159.

[17] Caviedes, J. and S. Gurbuz. No-Reference Sharpness Metric Based on Local Edge Kurtosis. in IEEE International Conference on Image Processing. 2002. Rochestern, New York, USA.

[18] Marziliano, P., F. Dufaux, et al., A No-Reference Perceptual Blur Metric. Image Communication, 2002. 19: p. 163-172.
[19] Marichal, X., W.-Y. ma, et al. Blur Determination in the Compressed Domain Using DCT Information. in IEEE International Conference on Image Processing. 1999. Kobe, Japan.
[20] Barland, R. and A. Saadane. Reference-Free Quality Metric Using a Region-Based Attention Model for JPEG-2000 Compressed Images. in IS&T/SPIE Symposium on Electronic Imaging. 2006. San Jose, California, USA.
[21] Barland, R. and A. Saadane. A Reference Free Quality Metric for Compressed Images. in Second International Workshop on Video Processing and Quality Metrics for Consumer Electronics. 2006. Scottsdale, Arizona, USA.
[22] Osberger, W. and A.M. Rohaly. Automatic Detection of Regions of Interest in Complex Video Sequences. in SPIE Human Vision and Electronic Imaging. 2001. San Jose, California, USA.
[23] Williams, D.R., J. Krauskopf, et al., Cardinal Directions of Color Space. Vision Research, 1982. 22: p. 1123-1131.
[24] Itti, L., C. Koch, et al., A Model of Saliency -Based Visual Attention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998. 20(11): p. 1254-12 59.
[25] Le Meur, O., P. Le Callet, et al. From Low Level Perception to High Level Perception, a Coherent Approach for Visual Attention Modeling. in SPIE Human Vision and Electronic Imaging IX. 2004. San Jose, California, USA.
[26] Koch, C. and S. Ullman, Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. Human Neurobiology, 1985. 4: p. 219-227.
[27] Parkhurst, D. and E. Niebur, Texture Contrast Attracts Overt Visual Attention in Natural Scenes. European Journal of Neuroscience, 2004. 19: p. 783-789.
[28] Burt, P.J. and E.H. Adelson, The Laplacian Pyramid as a Compact Image Code. IEEE Transactions on Communications, 1983. 31(4): p. 532-540.
[29] Rohaly, A.M., P. Corriveau, et al. Video Quality Experts Group: Current Results and Future Directions. in Proceedings of Visual Communications and Images Processing. 2000.
[30] Sheikh, H., Z. Wang, et al., LIVE Image Quality Assessment Database. http://live.ece.utexas.edu/research/quality.

Graphes de Reeb de Haut Niveau de Maillages Polygonaux 3D

Julien Tierny, Jean-Philippe Vandeborre* et Mohamed Daoudi*

Laboratoire d'Informatique Fondamentale de Lille (UMR USTL/CNRS 8022)
* GET / INT / Télécom Lille 1
{*tierny, vandeborre, daoudi*}*@lifl.fr*

(a) Sommets caractéristiques. (b) Fonction d'application. (c) Graphe de Reeb de haut niveau. (d) Application à la déformation de maillage.

FIG. 1 – *Principales étapes de notre méthode sur un maillage quelconque.*

Résumé

Cet article présente une méthode originale pour la construction de graphes de Reeb invariants de haut niveau – entités topologiques qui offrent une bonne vue d'ensemble de la structure d'un objet 3D.
Dans ce but, nous proposons un algorithme d'extraction de sommets caractéristiques simple et précis. Ces sommets sont utilisés pour le calcul d'une fonction d'application invariante, visuellement intéressante. De plus, nous proposons un nouvel algorithme de construction de graphe de Reeb, basé sur l'analyse de connexité de lignes de niveau discrètes. Cet algorithme apporte une solution pratique au problème de suppression de points critiques non significatifs, produisant en sortie des graphes bénéficiant de bonnes propriétés descriptives. L'invariance géométrique de ces graphes et leur forte tolérance à la variation de pose du modèle et à la variation d'échantillonnage du maillage en font de bons descripteurs, exploitables dans diverses applications, comme la déformation de maillage (expérimentée dans cet article), la compression, l'indexation 3D, la métamorphose, etc.

Mots clefs

Modélisation de formes 3D, description topologique invariante, graphes de Reeb, sommets caractéristiques.

1 Introduction

Le maillage de polygones est une représentation des formes 3D massivement utilisée. Cependant, bon nombre d'applications en informatique graphique nécessitent des descriptions de formes de plus haut niveau, comme des descriptions structurelles par exemple.

Pour répondre à ce besoin, de nombreuses approches ont été développées, comme la segmentation de maillages [1] ou l'extraction de squelettes [2]. Les approches topologiques basées sur les graphes de Reeb présentent l'avantage de préserver les propriétés topologiques du maillage [3]. Cependant, en pratique, la construction de graphes de Reeb pour la description de haut niveau soulève plusieurs problèmes, comme le non-respect de contraintes d'invariance ou l'identification de points critiques non-significatifs. Ceci peut conduire à des graphes de faible intérêt sémantique [4], encodant des détails non significatifs, que nous désignons par le terme de *graphes de Reeb de bas-niveau*.

Dans cet article, nous présentons une méthode originale pour la construction de graphes de Reeb invariants de haut niveau. Premièrement, nous introduisons l'état de l'art des approches topologiques. Deuxièmement, nous présentons un nouvel algorithme d'extraction de sommets caractéristiques (figure 1(a)). Cet algorithme est utilisé pour le calcul d'une fonction d'application invariante visuellement intéressante (figure 1(b)). Puis, nous présentons un algorithme de construction de graphe qui apporte une solution pratique pour la suppression de points critiques non-significatifs (figure 1(c)). Finalement, nous présentons et commentons des résultats expérimentaux. Nous évoquons également les applications possibles de notre méthode, comme la déformation de maillages (figure 1(d)).

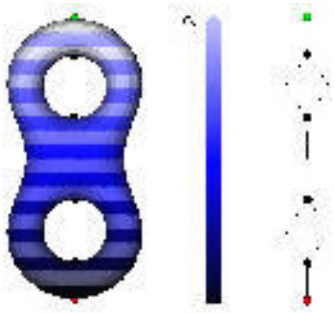


FIG. 2 – *Évolution des lignes de niveau de la fonction* hauteur *sur un bitore, ses points critiques et son graphe de Reeb.*

2 État de l'art

Un graphe de Reeb [5] est une structure topologique définie comme suit :

Definition 1 (Graphe de Reeb) *Soit f une fonction réelle définie sur une variété compacte V $f : V \to \mathbb{R}$. Le graphe de Reeb de f est l'espace quotient de f dans $V \times \mathbb{R}$, par la relation d'équivalence $(p_1, f(p_1)) \sim (p_2, f(p_2))$, vérifiée si et seulement si :*

$$\begin{cases} f(p_1) = f(p_2) \\ p_1 \text{ et } p_2 \text{ appartiennent à la même composante} \\ \qquad \text{connexe de } f^{-1}(f(p_1)) \end{cases}$$

Concrètement, un graphe de Reeb est composé de noeuds représentant les points critiques de f, autrement dit les points de la variété V où les dérivées partielles de f s'annulent. Les arêtes du graphe représentent les composantes connexes de V reliant les points critiques de f. La figure 2 présente un graphe de Reeb calculé sur un bitore, selon la fonction *hauteur*. Les points critiques de cette fonction ont été marqués en rouge pour les *minima*, en vert pour les *maxima* et en noir pour les *points selles*.

Dans le cas des surfaces triangulées, les algorithmes traditionnels de construction de graphes de Reeb [6, 7] identifient dans un premier temps les sommets correspondant aux points critiques de f, puis construisent le graphe en analysant leurs relations de connexité.

Premièrement, ces méthodes présupposent que l'ensemble des points critiques identifiés est significatif, alors qu'en pratique, cette hypothèse peut conduire à des graphes comportant un grand nombre de noeuds et d'arêtes [4], encodant des détails insignifiants et n'offrant donc pas une description globale, de haut niveau. Pour fournir une description structurelle globale d'un objet 3D, il est donc nécessaire d'apporter une solution à cette problématique de sélection de points critiques significatifs. Plusieurs algorithmes pratiques ont été proposés à cette fin [8, 9], mais tous sont conditionnés par un paramètre d'entrée (paramètre de *persistance* ou de *coupe*). Dans notre approche, nous proposons un algorithme unifié de construction et de simplification de graphe, ne prenant aucun paramètre d'entrée.

Deuxièmement, dans certaines applications (comme la modélisation de terrains), la fonction f est donnée par le contexte applicatif. Lorsqu'il s'agit de construire des *squelettes topologiques* [3] ou des graphes de Reeb de haut niveau, il est nécessaire de définir une fonction d'application f qui d'une part présente des propriétés d'invariance et qui d'autre part mette en valeur les parties les plus visuellement intéressantes de l'objet. Lazarus et Verroust [10] ont proposé une telle fonction, définie en tout sommet de la triangulation par sa *distance géodésique*[1] à un sommet source. L'heuristique proposée pour choisir ce point source souffre cependant d'une certaine instabilité, ce qui exclue son utilisation dans des applications où la stabilité est une propriété fondamentale. Pour résoudre ce problème, dans le cadre de l'indexation 3D, Hilaga et al. [11] proposent d'intégrer cette fonction sur tout le maillage au dépend d'un coût de calcul relativement élevé (les auteurs proposent une approximation). Dans notre approche, pour mettre en valeur la structure globale de l'objet, nous utilisons des distances géodésiques dont les points sources sont les sommets caractéristiques du maillage.

3 Fonction d'application

Plusieurs fonctions d'application ont été proposées dans l'état de l'art pour la construction de graphes de Reeb. Le choix de cette fonction détermine les propriétés de stabilité et d'invariance des graphes résultants. Dans notre approche, étant donné une triangulation connexe T, nous déterminons dans un premier temps ses sommets caractéristiques. Puis, nous définissons notre fonction d'application, notée f_a, en chaque sommet $v \in T$ en calculant la distance géodésique de v au sommet caractéristique le plus proche.

3.1 Métrique employée

Tout d'abord, pour garantir l'invariance de notre méthode aux rotations et aux translations, nous définissons f_a en nous basant sur l'estimation de distances géodésiques. De telles mesures ne sont pas définies par un quelconque repère euclidien et sont donc invariantes aux rotations et aux translations. Deuxièmement, pour garantir l'invariance de notre méthode à l'homothétie uniforme, nous utilisons des grandeurs normalisées. Troisièmement, l'utilisation de distances géodésiques garantit la tolérance de notre méthode aux variations de pose du modèle. Par exemple, la distance géodésique entre le nez d'un humanoïde et ses doigts reste la même, que son bras soit plié ou tendu.

D'un point de vue algorithmique, les distances géodésiques peuvent être approchées par l'algorithme de Moore-Dijkstra (minimisation de distance dans les graphes pondérés). Dans le reste de l'article, nous désignerons par $\delta(v_1, v_2)$ la distance géodésique normalisée entre les sommets v_1 et v_2.

[1] Longueur du plus court chemin entre deux sommets d'une triangulation.

3.2 Extraction des sommets caractéristiques

Les sommets caractéristiques d'une triangulation sont les sommets situés aux extrémités des composantes proéminentes de l'objet. Visuellement, leur ensemble donne une vue globale de la structure du modèle 3D. C'est pourquoi nous décidons de les utiliser comme origines pour nos évaluations de distances géodésiques.

Plusieurs algorithmes ont été proposés pour l'extraction de sommets caractéristiques. Par exemple, Mortara et Pantanè [12] proposent de sélectionner comme sommets caractéristiques les sommets où la courbure gaussienne excède un certain seuil. Malheureusement, cette technique ne permet pas d'extraire des sommets caractéristiques sur les zones de courbure constante (sphères, zones planes, etc.). Katz et al. [13] ont développé un algorithme basé sur l'homothétie multi-dimensionnelle, en complexité d'exécution quadratique.

Dans cet article, nous proposons un algorithme relativement direct, basé sur des outils de topologie différentielle. Soient v_{s_1} et v_{s_2} les sommets de T les plus distants l'un de l'autre (au sens géodésique). Ces sommets sont identifiés par l'algorithme de calcul de Diamètre d'Arbre [10]. Sur la figure 3, v_{s_1} est situé à l'extrémité du poignet et v_{s_2} est situé au bout du majeur.

Soient f_{g_1} et f_{g_2} deux fonctions réelles définies sur chaque sommet $v \in T$:

$$f_{g_1}(v) = \delta(v, v_{s_1}) \tag{1}$$

$$f_{g_2}(v) = \delta(v, v_{s_2}) \tag{2}$$

En se basant sur la classification des points critiques proposée dans [7], un *minimum local* est défini comme un sommet dont tous les voisins directs ont une valeur de fonction supérieure. Réciproquement, un *maximum local* est défini comme un sommet dont tous les voisins directs ont une valeur de fonction inférieure. Soit E_1 l'ensemble des extrema locaux (minima et maxima) de f_{g_1} (en jaune sur la figure 3(a)) et E_2 l'ensemble des extrema locaux de f_{g_2} (en cyan sur la figure 3(b)). Les extrémités des composantes proéminentes sont des configurations où f_{g_1} et f_{g_2} tendent vers des extrema (voir figures 3(a) et 3(b)). Par conséquent, l'ensemble des sommets caractéristiques est à la fois inclus dans E_1 et dans E_2. Donc, nous définissons l'ensemble des sommets caractéristiques F de T (figure 3(c)) comme suit :

$$F = E_1 \cap E_2 \tag{3}$$

En pratique, les extrema locaux de f_{g_1} et f_{g_2} qui correspondent à des sommets caractéristiques n'apparaissent pas strictement sur les mêmes sommets, mais dans le même *voisinage géodésique*. Par conséquent, la contrainte d'intersection est relaxée comme suit, avec $\epsilon \in [0, 1]$ le rayon du *voisinage géodésique* (les distances géodésiques sont normalisées) :

$$v \in F \Longleftrightarrow \begin{cases} \exists v_{e_1} \in E_1 \quad / \quad \delta(v, v_{e_1}) < \epsilon \\ \exists v_{e_2} \in E_2 \quad / \quad \delta(v, v_{e_2}) < \epsilon \\ \delta(v, v_{f_i}) > \epsilon \quad \forall v_{f_i} \in F \\ \epsilon \in [0, 1] \end{cases} \tag{4}$$

(a) E_1. (b) E_2. (c) $E_1 \cap E_2$.

FIG. 3 – *Extraction des sommets caractéristiques.*

(a) 25 000 sommets. (b) 5 000 sommets. (c) 1 000 sommets.

FIG. 4 – *Tolérance de l'algorithme d'extraction de sommets caractéristiques face aux variations de résolution du maillage.*

D'après nos expériences, fixer $\epsilon = 0.05$ donne des résultats satisfaisants. L'algorithme de Moore-Dijkstra constitue un goulot d'étranglement en terme de complexité d'exécution. f_{g_1} et f_{g_2} sont toutes deux calculées en $O(n \times log(n))$ étapes, avec n le nombre de sommets dans T.

Dans ce paragraphe, nous avons présenté un algorithme rapide pour l'extraction de sommets caractéristiques, en $O(n \times log(n))$. Cet algorithme est basé sur l'évaluation de distances géodésiques. Par conséquent, il est invariant aux transformations géométriques et robuste aux variations de changement de pose du modèle. De plus, la sélection des sommets caractéristiques est guidée par une analyse de gradient de fonctions d'application. Aucune hypothèse n'a été formulée quant à l'échantillonnage de la triangulation. Par conséquent, cet algorithme est robuste aux variations d'échantillonnage de la surface, comme illustré figure 4.

3.3 Définition de la fonction d'application

La définition de la fonction d'application employée dépend de ce qui souhaite être mis en valeur sur la surface. Par exemple, pour la modélisation de terrains, la fonction *hauteur* présentera des points critiques sur les pics et dans les vallées, offrant ainsi une description topologique pertinente. Dans notre approche, nous souhaitons mettre en valeur la structure globale des objets. C'est pourquoi nous décidons d'utiliser les sommets caractéristiques dans notre calcul de fonction d'application f_a, que nous définissons comme suit :

$$f_a(v) = 1 - \hat{\delta}(v, v_p) \tag{5}$$

avec v_p le sommet caractéristique le plus proche de v :

$$v_p \in F \quad / \quad \hat{\delta}(v, v_p) = min_{v_{f_i} \in F} \quad \delta(v, v_{f_i}) \tag{6}$$

(a) $|F| = 6$, $|C| = 94$. (b) $|F| = 7$, $|C| = 92$.

FIG. 5 – *Évolution des lignes de niveau de f_a et ses points critiques sur des modèles standards.*

La figure 5 présente des exemples de calcul de f_a sur des modèles standards, ainsi que le nombre de sommets caractéristiques identifiés ($|F|$) et le nombre de points critiques ($|C|$, identifiés selon la classification proposée dans [7]). Comme le calcul de f_a est basé sur des évaluations de distances géodésiques, f_a est invariante aux rotations et aux translations. De plus, toutes les grandeurs utilisées sont normalisées, donc cette fonction est invariante aux homothéties uniformes. Comme on peut le voir sur la figure 5, f_a génère un nombre important de points critiques. Par conséquent, les algorithmes traditionnels de construction de graphes de Reeb créeraient des graphes complexes, comportant autant de noeuds que de points critiques (94 pour le modèle figure 5(a) et 92 pour le modèle figure 5(b)). Il s'agit d'un problème majeur pour la description de haut niveau, auquel nous apportons une solution dans le paragraphe suivant.

4 Graphes de Reeb de haut niveau

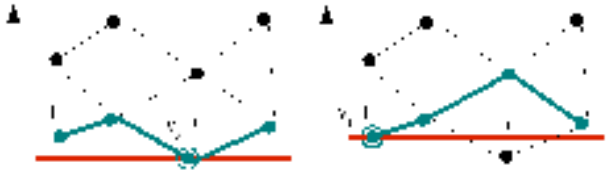
Dans cette section, nous proposons un algorithme pratique pour la construction de graphes de Reeb de haut niveau, basé sur la notion de *lignes de niveau discrètes*.

4.1 Lignes de niveau discrètes

La définition de lignes de niveau d'une fonction réelle f définie sur une triangulation T n'est pas un problème trivial. Dans le cas continu, deux points p_1 et p_2 appartiennent à la même ligne de niveau $f^{-1}(f(p_1))$ si $f(p_2) - f(p_1) = 0$. De plus, p_1 et p_2 appartiennent au même contour si ils appartiennent à la même composante connexe de $f^{-1}(f(p_1))$.

Dans le cas discret, pour un sommet donné $v \in T$, selon l'échantillonnage de T, $f^{-1}(f(v))$ est souvent réduite au sommet v lui même. Au vu de la définition 1, un graphe de Reeb correct ne pourrait pas être calculé avec cette définition des lignes de niveau discrète, car les conditions de la relation d'équivalence seraient rarement vérifiées.

Pour préserver les propriétés topologiques des lignes de niveaux dans le cas discret, nous définissons la *ligne de niveau discrète* $\Gamma(v)$ associée au sommet v par une courbe portée par les arêtes de T, approximant par valeur supérieure la ligne de niveau continue $f^{-1}(f(v))$.



(a) $\Gamma(v_a)$. (b) $\Gamma(v_b)$.

FIG. 6 – *Exemple de lignes de niveau continues (en rouge) et discrètes (en vert, fonction hauteur).*

(a) $\Gamma(v_{600})$, 2 contours. (b) $\Gamma(v_{10\ 000})$, 4 contours. (c) $\Gamma(v_{20\ 000})$, 6 contours.

FIG. 7 – *Exemples de lignes de niveau discrètes sur un maillage de 25 000 sommets (fonction f_a).*

La figure 6 montre des lignes de niveau discrètes traversant une triangulation quelconque au vu de la fonction hauteur. De plus, nous désignons par le terme *contour discret* chaque sous ensemble connexe de $\Gamma(v)$. En particulier, nous définissons le contour discret $\gamma(v)$ associé au sommet v comme le sous-ensemble connexe de $\Gamma(v)$ contenant v. Plus l'échantillonnage de T sera important, plus les lignes de niveau discrètes tendront vers les lignes de niveau continues.

Les lignes de niveau discrètes peuvent être calculées pour tous les sommets du maillage en utilisant un algorithme de remonté de gradient. Ce type d'algorithme manipule deux piles V et C, représentant respectivement l'ensemble des sommets visités et l'ensemble des sommets candidats à la visite. À chaque itération de l'algorithme, C entoure V par valeur supérieure. Une étude plus approfondie montre qu'à chaque itération, $\Gamma(v)$ est équivalent à C avec $v = argmin_{v \in C} \quad f(v)$.

Sur la figure 7, plusieurs exemples de lignes de niveau discrètes sont représentés, à différentes itérations de l'algorithme. L'ensemble de sommets V est affiché en blanc tandis que $\Gamma(v)$ est affichée en rouge. Visiter récursivement $\Gamma(v)$ permet d'en identifier chacun de ses sous-ensembles connexes, et particulièrement $\gamma(v)$.

4.2 Construction des graphes

Les algorithmes traditionnels de construction de graphes de Reeb nécessitent une étape de simplification, afin d'en éliminer les branches non significatives. Ici, nous propo-

(a) (b) (c) (d) (e) (f)

FIG. 8 – *Contextes de bifurcation et de jonction sur un tore (fonction hauteur).*

sons un algorithme unifié pour la construction et la simplification, basé sur notre notion de ligne de niveau discrète.
En suivant la définition 1, nous pouvons établir une relation d'équivalence analogue dans le cas discret, entre deux sommets $v_1, v_2 \in T$:

$$(v_1, f(v_1)) \sim (v_2, f(v_2)) \Longleftrightarrow \begin{cases} v_2 \in \Gamma(v_1) \\ v_2 \in \gamma(v_1) \end{cases} \quad (7)$$

À partir de cette relation d'équivalence, pour chaque ligne de niveau discrète $\Gamma(v)$, il est possible d'identifier chacune des composantes connexes de T traversée par $\Gamma(v)$, et donc de construire un graphe de Reeb.
Soit $N_{\Gamma(v_t)}$ le nombre de sous-ensembles connexes de $\Gamma(v_t)$, avec v_t le sommet visité à l'itération t de l'algorithme de construction de lignes de niveau discrètes. Pour construire un graphe de Reeb, il suffit donc d'observer l'évolution de $N_{\Gamma(v_t)}$ au fil de l'algorithme de construction de lignes de niveau discrètes, en envisageant les variations topologiques suivantes :

1. *bifurcations* :

$$N_{\Gamma(v_t)} > N_{\Gamma(v_{t-1})} \quad (8)$$

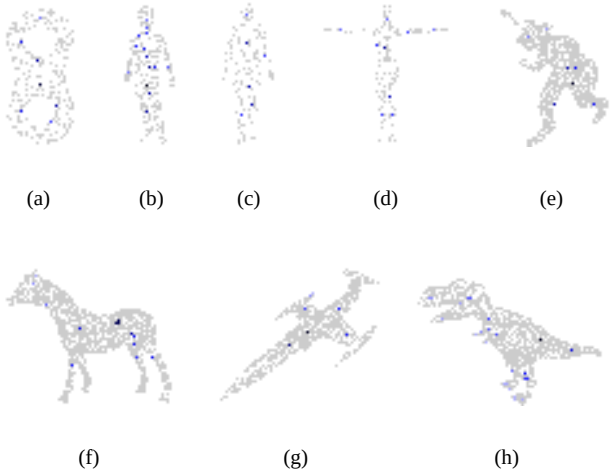
2. *jonctions* :

$$\begin{cases} N_{\Gamma(v_t)} < N_{\Gamma(v_{t-1})} \\ \exists v_n \in Lk(v_{t-1}) \quad / \quad v_n \in \Gamma(v_t) \end{cases} \quad (9)$$

3. *terminaisons* :

$$\begin{cases} N_{\Gamma(v_t)} < N_{\Gamma(v_{t-1})} \\ v_n \notin \Gamma(v_t), \quad \forall v_n \in Lk(v_{t-1}) \end{cases} \quad (10)$$

La figure 8 présente les contextes d'apparition de bifurcations et de jonctions sur un tore, au vu de la fonction hauteur. Sur la figure 8(a), $\Gamma(v)$ n'est composée que d'un contour discret, qui se divise en deux en 8(b) : une bifurcation est donc créée dans le graphe (figure 8(e)). Sur la figure 8(b), $\Gamma(v)$ est composée de deux contours discrets, qui fusionnent en 8(c) : une jonction est donc créée dans le graphe (figure 8(f)). Dans l'équation 9, la seconde condition exprime le fait qu'un nouveau contour apparaît après la jonction, ce qui n'est pas le cas pour une terminaison équation 10 ($Lk(v_t)$ désigne le lien, ou le voisinage direct, de v_t).
Dans notre approche, la construction de graphe de Reeb est donc effectuée durant l'algorithme de construction de lignes de niveau discrètes. Pour cela, nous appliquons les variations topologiques nécessaires sur le graphe en fonction de l'évolution du nombre de sous ensembles connexes des lignes de niveau discrètes. Comme ces lignes de niveau ne se déconnectent pas dans les configurations bruitées de



(a) (b) (c) (d) (e)

(f) (g) (h)

FIG. 9 – *Graphes de Reeb duaux de formes primitives et complexes.*

f_a (zones repérées par des ensembles de points critiques sur la figure 5, en rouge et noir), le graphe ne rend compte que des variations topologiques significatives. Cette propriété est illustrée dans la section suivante, dédiée aux résultats expérimentaux.

5 Résultats expérimentaux

Une étude de la complexité en temps de notre algorithme montre qu'il nécessite $O(n^2)$ étapes, avec n le nombre de sommets dans la triangulation. Cet algorithme a été implémenté en C sous GNU/Linux et expérimenté sur un PC type station de travail (P4-3GHz, 2 Go de RAM). Avec cette configuration, le calcul d'un graphe de Reeb de haut niveau prend 0.23 seconde pour un modèle de 2 000 faces, 2 secondes pour un modèle de 10 000 faces, 17 secondes pour un modèle de 40 000 faces et 86 secondes pour un modèle de 100 000 faces.
La figure 9 présente des graphes de Reeb de haut niveau calculés avec notre algorithme. Ces graphes ont de bonnes propriétés descriptives car ils n'encodent pas de variations topologiques non significatives (avec les algorithmes traditionnels, le graphe du cheval aurait compté 92 noeuds). Sur cette figure, les graphes représentés sont duaux. C'est-à-dire que chaque composante connexe est représentée par un noeud, et leurs relations d'adjacence par une arête. Un exemple de graphe de Reeb non dual est présenté figure 10(a), où un noeud a été placé au centre de chaque contour discret, pour former un *squelette topologique*.
De tels squelettes sont particulièrement adaptés pour des applications de transformation de maillages, comme la déformation, illustrée figure 10(b). Comme la déformation de maillages est un problème qui dépasse le cadre de cet article, pour cet exemple, nous avons utilisé une stratégie simple. Pour un noeud du graphe de Reeb dual sélectionné par un utilisateur, étant donné une origine, un axe et un angle de rotation, une matrice de rotation est calculée. Cette matrice est ensuite appliquée à chaque sommet référencé par le noeud du graphe de Reeb dual. Ainsi, sur

(a) (b)

FIG. 10 – *Graphe de Reeb de haut niveau (a) et une application à la déformation de maillage (b).*

(a) 25 000 sommets. (b) 5 000 sommets. (c) 1 000 sommets.

FIG. 11 – *Tolérance de notre algorithme face aux variations de résolution du maillage.*

la figure 10(b), le poignet et les doigts de la main ont été inclinés par rapport au modèle initial.
Sur les figures 9(a) et 9(b), nous remarquons que notre algorithme prend correctement en compte les surfaces de genre non nul. Sur les figures 9(c) et 9(d), nous observons que les graphes sont similaires, que les bras de l'humanoïde soient couchés ou relevés. Par ailleurs, aucune hypothèse n'a été portée sur la résolution du maillage. Par conséquent notre algorithme est robuste aux variations de résolutions du maillage, comme illustré figure 11. Ces propriétés de tolérance (résolution et pose du modèle) associées aux propriétés d'invariance géométrique font des graphes de Reeb de haut niveau de bons descripteurs pour l'indexation 3D, la compression, etc.

6 Conclusion

Dans cet article, nous avons présenté une nouvelle méthode pour la construction de graphe de Reeb invariants de haut niveau. Cette méthode se décompose en trois étapes : extraction de sommets caractéristiques, calcul d'une fonction d'application et construction de lignes de niveau discrètes. Les graphes obtenus, sans aucun paramètre d'entrée et dans des temps d'exécution satisfaisants, présentent des propriétés d'invariance et de forte tolérance aux variations de pose des modèles et de résolution des maillages. Par ailleurs, ces graphes n'encodent que les variations topologiques significatives et donc ont de bonnes propriétés descriptives. Ces atouts en font donc de bons descripteurs pour diverses applications comme la déformation, l'indexation 3D ou encore la compression.

Références

[1] Ariel Shamir. A formalization of boundary mesh segmentation. Dans *IEEE 2nd International Symposium on 3DPVT*, pages 82–89, 2004.

[2] Harry Blum et Roger N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10 :167–180, 1978.

[3] Silvia Biasotti, Simone Marini, Michela Mortara, et Giuseppe Patanè. An overview on properties and efficacy of topological skeletons in shape modelling. Dans *Shape Modeling International*, pages 245–254, 2003.

[4] Xinlai Ni, Michael Garland, et John Hart. Fair Morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics, SIGGRAPH*, 23 :613–622, 2004.

[5] Georges Reeb. Sur les points singuliers d'une forme de pfaff complètement intégrable ou d'une fonction numérique. *Comptes-rendus de l'Académie des Sciences*, 222 :847–849, 1946.

[6] Sergey Tarasov et Michael Vyalyi. Construction of contour trees in 3D in O((n) log (n)) steps. Dans *Symposium on Computational Geometry*, pages 68–75, 1998.

[7] Kree Cole-McLaughlin, Herbert Edelsbrunner, John Harer, Vijay Natarajan, et Valerio Pascucci. Loops in Reeb graphs of 2-manifolds. Dans *Symposium on Computational Geometry*, pages 344–350, 2003.

[8] Marco Attene, Silvia Biasotti, et Michela Spagnuolo. Shape understanding by contour-driven retiling. *The Visual Computer*, Volume 19 :127–138, 2003.

[9] Bremer Peer Timo, Herbert Edelsbrunner, Bernd Hamann, et Valerio Pascucci. Topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, Volume 10 :385–396, 2004.

[10] Francis Lazarus et Anne Verroust. Level set diagrams of polyhedral objects. Rapport technique 3546, Institut National de Recherche en Informatique and en Automatique (INRIA), 1999.

[11] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, et Tosiyasu Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. Dans *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 203–212, 2001.

[12] Michela Mortara et Giuseppe Patanè. Affine-invariant skeleton of 3D shapes. Dans *Shape Modeling International*, pages 245–252, 2002.

[13] Sagi Katz, George Leifman, et Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer (Pacific Graphics)*, 21 :865–875, 2005.

Génération automatique de marqueurs pour la ligne de partage des eaux 3D

S. Delest G. Pageot R. Boné H. Cardot

Laboratoire Informatique

Université François-Rabelais de Tours
64, avenue Jean Portalis, 37200 TOURS

{sebastien.delest, romuald.bone, hubert.cardot}@univ-tours.fr

Concours Jeune Chercheur : Oui

Résumé

Cet article présente une nouvelle méthode de segmentation de maillage triangulaire basée sur la ligne de partage des eaux, initialisée par des marqueurs issus de la squelettisation de l'objet 3D. Dans cette méthode, le modèle est d'abord transformé en une représentation de voxels ; un algorithme de squelettisation est ensuite utilisé pour extraire le squelette constitué de voxels. Chaque branche du squelette est labellisée et les voxels de surface prennent le label des voxels du squelette qui leur sont associés. Les voxels de surface non ambigus peuvent ensuite servir de marqueurs pour la ligne de partage des eaux 3D. Cette méthode, qui associe la décomposition en partie et la décomposition en patches surfaciques, est particulièrement bien adaptée aux problématiques de segmentation d'objet qui comportent des parties significatives.

Mots clefs

Ligne de partage des eaux, squelettisation, marqueurs.

1 Introduction

Les maillages polygonaux sont couramment utilisés pour représenter des surfaces 3D ; en particulier les maillages triangulaires, qui offrent une structure simple et qui sont présents dans de nombreuses applications. Cet article concerne la segmentation de maillages triangulaires, cependant la méthode est tout aussi bien adaptée aux autres types de maillages.

La segmentation de maillages a de nombreuses applications dans les domaines de la visualisation et de la modélisation. La forme des modèles est importante et peut amener à différentes approches de segmentation selon qu'il s'agisse de formes naturelles ou de parties mécaniques. Les méthodes de segmentation de maillages sont classées principalement en deux groupes : la décomposition en patches surfacique qui tient compte des propriétés de planéité, de taille et de convexité et la décomposition en parties qui cible davantage le partitionnement en parties significatives de l'objet.

La décomposition en parties intervient dans de nombreux domaines comme l'appariement de formes, l'indexation et la reconstruction de formes par la reconnaissance des objets 3D, le morphing, la compression et la simplification de forme, la détection de collision, le mappage de texture, etc. Katz et al. [1] ont proposé une décomposition d'objets à partir du squelette pour permettre la déformation et l'animation du modèle. Wu et Levine [2] font intervenir les propriétés des charges électriques qui s'accumulent dans les zones de fortes convexités et disparaissent dans les zones de fortes concavités pour décomposer un modèle en parties. Koschan [3] utilise des opérateurs morphologiques comme outils de marquage et la ligne de partage des eaux pour segmenter les objets en parties. Lavoué et al. [4] ont fait intervenir des procédés de classification et de croissance de régions pour identifier les parties les plus significatives des objets 3D. Bruner et al. [5] offrent une décomposition du maillage par squelettisation et par association des branches du squelette aux faces du modèle.

Nous proposons ici une segmentation de maillages par décomposition en parties basée sur la squelettisation puis la ligne de partage des eaux (LPE). Le maillage 3D est d'abord voxelisé ; chaque face du modèle est convertie en groupe de voxels et la surface fermée qui en résulte est ensuite remplie. Nous avons alors utilisé l'algorithme de squelettisation proposé par Kálmán Palágyi [6] pour obtenir le squelette du modèle. Chacune de ses branches obtient un label différent et les voxels du squelette sont associés à des voxels de surfaces de l'objet, eux même associés aux faces du maillage d'origine. Les faces qui sont connectées à des voxels non ambigus reçoivent leur label et vont servir de marqueurs pour la LPE.

2 La Ligne de Partage des Eaux

La méthode de Ligne de Partage des Eaux proposée par Digabel et Lantuéjoul [7] est un outil morphologique qui a été longtemps considéré comme l'étape finale d'un processus de segmentation. De nombreux pré-traitements (filtres, opérateurs morphologiques) ont eu alors pour but de réduire le nombre de régions non significatives tout en conservant les contours réels. Néanmoins, les travaux de

Beucher [8] ont démontré les limites de ces méthodes et ont mis en avant une approche de niveau supérieur, en considérant alors la LPE comme un opérateur morphologique de base. De cette approche a résulté l'apparition des algorithmes de LPE hiérarchique et de LPE par marqueurs. Dans ce qui suit, nous proposons une LPE par marqueurs générés à partir du squelette du modèle 3D.

2.1 La LPE 3D

En deux dimensions, l'algorithme de la LPE consiste à simuler la montée des eaux sur le gradient (ou bien un autre paramètre associé aux pixels) de l'image d'entrée depuis ses minima locaux ou ses marqueurs (fig. 1). Cela permet de générer des lignes de partage aux endroits où les bassins se rejoignent, définissant ainsi un SKIZ géodésique de l'image. Un SKIZ géodésique est un ensemble de lignes continues divisant une image en un ensemble de régions d'influence équivalentes. Ainsi, on obtient au final un découpage de l'image en régions dans lesquelles l'intensité lumineuse des pixels est relativement homogène.

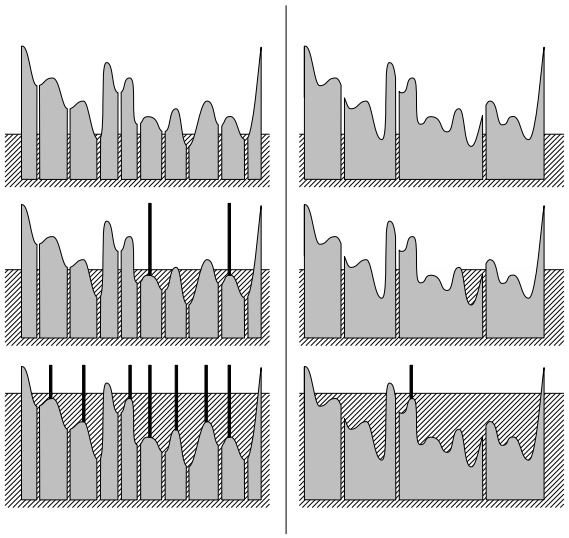


Figure 1 – *Trois niveaux d'inondation pour la LPE avec minima à gauche et la LPE par marqueurs à droite. Les minima et les marqueurs sont les sources d'innondation.*

En trois dimensions, l'intensité ou le gradient des pixels est remplacé par la courbure des vertex, de même que la connexité fixe des pixels devient alors variable (fig. 2).

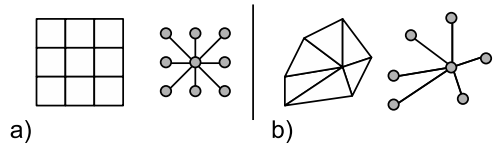


Figure 2 – *a) Structure de pixels connectés, b) structure de vertex connectés*

Pour construire la LPE, nous avons utilisé l'algorithme de calcul rapide de la LPE 2D par Files d'Attentes Hiérarchique (FAH) sans biais proposé par Serge Beucher [9] et adapté en trois dimensions dans [10]. Cette méthode consiste à créer autant de piles qu'il y a de niveaux de courbure dans le modèle traité. Les vertex observés dans le voisinage des vertex traités tout au long de l'immersion seront ainsi placés dans la pile correspondant à leur niveau. Les vertex labellisés comme minima ou marqueurs sont les premiers empilés. Un label différent est attribué aux vertex ou groupes de vertex isolés. Leurs voisins sont ensuite extraits et rangés dans la file correspondant à leur niveau de courbure (fig. 3). Ils reçoivent le label de leur vertex parent et lorsqu'un conflit apparaît, le vertex est marqué comme ligne de partage des eaux.

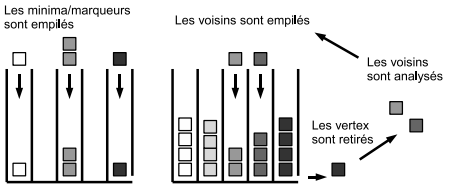


Figure 3 – *Ligne de partage des eaux basée sur les Files d'Attentes Hiérarchiques.*

2.2 Critère de courbure

Plusieurs approches ont été explorées pour obtenir une estimation de la courbure ; Meyer et al. [11] ont proposé une étude intéressante sur les opérateurs discrets pour estimer la courbure et Mangan et Whitaker ont mis en avant l'efficacité de la norme de la matrice de covariance dans [12]. Dans notre cas, cette dernière méthode s'avère la plus adaptée pour caractériser la courbure des vertex qui correspond au critère de hauteur pour la ligne de partage des eaux. Le calcul de la courbure par la matrice de covariance repose sur un concept statistique qui consiste à évaluer les variances et covariances des coordonnées des différents vertices appartenant au voisinage. Pour un vertex donné, la courbure C est définie par la norme de la matrice de covariance :

$$C = \|M\| \text{ avec } M = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

$$\sigma_{uu}^2 = \frac{1}{N}\sum_{i=0}^{N}(u_i + \bar{u})^2$$

$$\sigma_{uv}^2 = \frac{1}{N}\sum_{i=0}^{N}(u_i + \bar{u})(v_i + \bar{v})$$

où σ_{uu} représente l'écart type des coordonnées en u du voisinage du vertex, and σ_{uv} représente la racine carrée de la covariance entre les composantes en u et les composantes en v. N correspond au nombre de triangles associés au vertex et $[x_t\ y_t\ z_t]^T$ est le vecteur de la normale du triangle t.

2.3 La sur-segmentation

L'utilisation de la ligne de partage des eaux seule ne permet pas réellement une bonne segmentation car beaucoup trop de régions sont détectées. Il existe deux principales méthodes pour limiter cette sur-segmentation : la segmentation hiérarchique et l'utilisation de marqueurs.

La segmentation hiérarchique L'approche hiérarchique peut consister à générer un arbre de régions à partir du résultat de la LPE. Les régions et les lignes de partages des eaux sont d'abord indexées, puis le processus de segmentation hiérarchique fait fusionner les régions dont les frontières communes sont les plus faibles. Il en résulte un arbre dans lequel il est possible d'explorer les différents niveaux de fusion des régions. La figure 4 propose deux segmentations avec des niveaux de fusions différents. Le modèle de gauche contient 208 régions et celui de droite 57 régions.

Figure 4 – *Segmentation hiérarchique à partir de la ligne de partage des eaux. Modèle Cow avec 208 régions à gauche et 57 régions à droite.*

Les marqueurs Ils vont définir les sources depuis lesquelles l'algorithme de la LPE va simuler la montée des eaux. Afin d'éviter la création de bassins au niveau des minima locaux, il est nécessaire d'effectuer une modification de l'homotopie de la structure d'entrée qui consiste à mettre les zones marquées au niveau le plus bas de la structure (fig. 1).

Figure 5 – *marqueurs et résultat de la LPE.*

Cette technique fournit des caractéristiques très intéressantes en terme de qualité de segmentation, de robustesse et de temps de calcul, aussi bien sur des objets industriels que naturels. Les faces ont été marquées manuellement dans l'exemple de la figure 5. Nous allons voir dans ce qui suit comment définir ces marqueurs de façon automatique.

3 Génération des marqueurs

Le marquage des régions qui vont servir de source d'inondation pour la LPE est réalisé à partir du squelette du modèle. La génération du squelette fait intervenir plusieurs procédés tels que la transformation du maillage en contour fermé de voxels, le remplissage du volume (voxelisation) et enfin la squelettisation du modèle.

3.1 La voxelisation

Brunner et Brunnet [5] ont proposé une méthode efficace pour stocker les voxels et réaliser la voxelisation sur un maillage fermé. La structure qui contient les voxels ne correspond pas à une image 3D mais à un plan ou tableau en deux dimensions qui, pour chaque case, intègre des couples de voxels. Ces voxels peuvent être associés aux entrées et sorties du rayon qui traverserait l'objet suivant une direction perpendiculaire au plan (fig. 6).

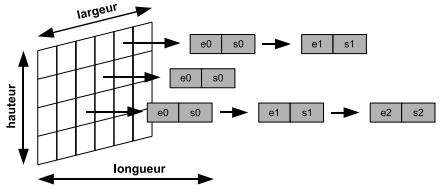


Figure 6 – *Structure de stockage des voxels. Seuls les couples de voxels (entrée/sortie) sont référencés.*

La voxelisation revient ici à déterminer l'intersection du rayon avec les faces du maillage. Le rayon entre dans l'objet puis en ressort, il peut y entrer à nouveau et en ressortir. Karabassi et al. [13] ont proposé un algorithme très rapide de voxelisation basé sur l'utilisation de 6 z-buffers (un z-buffer est lié à une direction de visualisation de l'image 3D) ; cependant, cette méthode ne prend pas en compte les parties internes ou cachées de l'objet.

3.2 La squelettisation

Pour extraire le squelette de voxels, nous avons utilisé l'algorithme de squelettisation de Palágyi dans [6] qui présente des avantages de rapidité et d'efficacité en terme d'érosion des différentes couches du volume. Cet algorithme supprime successivement les voxels dans l'image 3D selon certaines contraintes géométriques. Pour réaliser une érosion symétrique, six érosions sont successivement appliquées sur les voxels de surface dans les directions Haut, Bas, Nord, Sud, Est et Ouest (fig. 7a). A chaque érosion, seul les voxels directement visibles à partir de la direction donnée sont testés ; si ceux-ci peuvent être enlevés sans que leur suppression ne modifie la topologie de l'objet, alors ils sont rangés dans une liste sans être encore retirés et sont considérés comme *points simples*. Cette liste est ensuite consultée et ces voxels peuvent être supprimés si leur simplicité n'a pas été affectée pas la suppression des autres points simples. Un voxel p de l'objet est appelé point

simple si sa suppression ne modifie pas la topologie de l'objet, c'est-à-dire si le nombre de composantes connexes et le nombre de trous de l'objet et de son complémentaire, dans le voisinage $N_{26}(p)$, restent inchangés après suppression de p. Le point p est simple s'il réuni les conditions suivantes :

1. Le point p ne doit pas être un point isolé ni un point extrémité.
2. Les voisins pleins dans le groupe $N_{26}(p)\backslash\{p\}$ sont *26-connectés* à ce même groupe. Le nombre de composantes connexes doit rester le même.
3. Le point p est *6-adjcacent* à un point blanc. C'est un point de bord.
4. Les voisins vides dans le groupe $N_6(p)\backslash\{p\}$ sont *6-connectés* au groupe de voxels vides $N_{18}(p)\backslash\{p\}$.

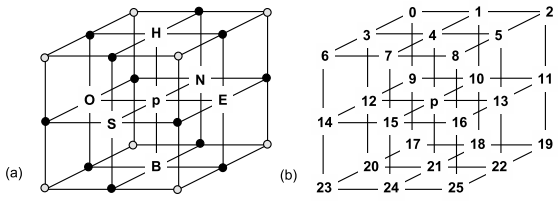


Figure 7 – *(a) Le groupe $N_6(p)$ contient le point central p et les six points marqués H, B, N, S, E et O. Le groupe $N_{18}(p)$ contient le groupe $N_6(p)$ et les 12 points marqués "•". Le groupe $N_{26}(p)$ contient le groupe $N_{18}(p)$ et les 8 points marqués "∘". (b) Indices assignés aux points dans le groupe $N_{26}(p)\backslash\{p\}$.*

L'algorithme de squelettisation fait intervenir une fonction qui réalise la suppression successive des points simples suivant l'ordre Haut, Bas, Nord, Sud, Est et Ouest jusqu'à l'obtention du squelette. A chaque itération, seuls les voxels de surface par rapport à la direction donnée sont traités, ce qui valide la condition 3 ; ces voxels ne doivent pas être isolés ou extrémités pour valider la condition 1. Les conditions 2 et 4 peuvent alors être testées suivants les algorithmes **Cond2Satisfaite** et **Cond4Satisfaite**. Ces algorithmes sont extraits de [6] et corrigés dans notre article car la phase de labellisation n'était pas complète.

La fonction **Cond2Satisfaite** utilise deux structures de données auxiliaires : la première correspond au tableau d'entiers L, où $L[i]$ stocke les labels assignés aux éléments représentés par $Np[i](i = 0, \ldots, 25)$. La seconde est la clé du processus de labellisation : $S26$ est un tableau d'indice, où $S26[i] = \{j | j \in N_{26}[i]$ et $0 \leq j < i\}(i = 0, \ldots, 25)$. On aura ainsi $S26[0] = \emptyset$, $S26[1] = \{0\}$, $\ldots S26[25] = \{13, 15, 16, 21, 22, 24\}$ (voir fig. 7b). Tous les groupes $S26[0], \ldots, S26[25]$ peuvent être stockés dans un tableau prédéfini. Les voisins pleins du voxel p sont 26-connectés si le même label est attribué à chacun des voisins pleins de p.

Fonction `Cond2Satisfaite(`Np`)`

début
- $label \leftarrow 0$
- $lst \leftarrow$ nouvelle liste vide
- **pour** $i \leftarrow 0$ **à** 25 **faire** $L[i] \leftarrow 0$
- **si** $Np[0] = 1$ **alors**
  - $label \leftarrow 1$
  - $L[0] \leftarrow label$
- **pour** $i \leftarrow 1$ **à** 25 **faire**
  - **si** $Np[i] = 1$ **alors**
    - $label \leftarrow label + 1$
    - $L[i] \leftarrow label$
    - **pour chaque** $j \in S26[i]$ **faire**
      - **si** $L[j] > 0$ **alors**
        - **pour** $k \leftarrow 0$ **à** $i - 1$ **faire**
          - **si** $L[k] = L[j]$ **alors** Insert(lst,k)
    - **tant que** $lst \neq \emptyset$ **faire**
      - $l \leftarrow$ Retire(lst)
      - $L[l] \leftarrow label$
- **pour** $i \leftarrow 0$ **à** 25 **faire**
  - **si** $Np[i] = 1$ **et** $L[i] \neq label$ **alors**
    - **retourner** *[FAUX]*
- **retourner** *[VRAI]*

fin

Fonction `Cond4Satisfaite(`Np`)`

début
- $label \leftarrow 0$
- $lst \leftarrow$ nouvelle liste vide
- **pour** $i \leftarrow 0$ **à** 25 **faire** $L[i] \leftarrow 0$
- **si** $Np[4] = 0$ **alors**
  - $label \leftarrow 1$
  - $L[4] \leftarrow label$
- **pour** $i \leftarrow 1$ **à** 17 **faire**
  - $indice \leftarrow N18[i]$
  - **si** $Np[indice] = 0$ **alors**
    - $label \leftarrow label + 1$
    - $L[indice] \leftarrow label$
    - **pour chaque** $j \in S18[i]$ **faire**
      - **si** $L[j] > 0$ **alors**
        - **pour** $k \leftarrow 0$ **à** $indice - 1$ **faire**
          - **si** $L[k] = L[j]$ **alors** Insert(lst,k)
    - **tant que** $lst \neq \emptyset$ **faire**
      - $l \leftarrow$ Retire(lst)
      - $L[l] \leftarrow label$
- **pour** $i \leftarrow 0$ **à** 5 **faire**
  - $indice \leftarrow N6[i]$
  - **si** $Np[indice] = 0$ **et** $L[indice] \neq label$ **alors**
    - **retourner** *[FAUX]*
- **retourner** *[VRAI]*

fin

La fonction **Cond4Satisfaite** utilise les mêmes principes mais cette fois-ci, ce sont les connexions entre points blancs qui sont analysées et ces points, dans le groupe $N_6(p)$, doivent être *6-connectés* au groupe de points blancs $N_{18}(p)$. Les voisins *6-adjcacents* ont les indices 4, 10, 12, 13, 15 et 21 dans la figure 7b. La clé de labellisation S18 intègre les groupes suivants $S18[0] = \emptyset$, $S18[1] = \emptyset$, $\ldots S18[2] = \{1, 3\}$, $S18[3] = \{4\}$, $\ldots S18[17] = \{15, 21\}$

3.3 Création du marquage

En considérant le squelette du modèle comme un graphe, il apparaît deux principaux types d'éléments : les arcs et les sommets. Les arcs contiennent tous les voxels étant connectés à un ou deux autres voxels et les sommets correspondent aux voxels de jonction qui sont connectés à au moins trois autres voxels.

Figure 8 – *Le modèle "Cow" à gauche et son squelette à droite. Les branches du squelette apparaissent d'une couleur différente.*

Le marquage peut commencer par l'attribution d'un label différent à chaque arc comme sur la figure 8. Lors de la création du contour fermé de voxels, les voxels de surface deviennent liés aux faces du maillage qui leur correspondent. Ces connexions sont transmises aux nouveaux voxels de surface lors de chaque érosion (Figure 9).

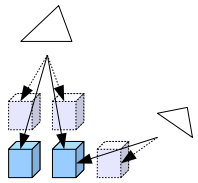


Figure 9 – *Transmission des connexions de faces aux nouveaux voxels de surface les plus proches*

Le squelette du modèle contiendra alors des voxels liés aux faces du maillage et les labels des voxels du squelette pourront être directement transmis aux faces (Figure 10). Les faces qui sont liés à des voxels ambigus ne seront pas labellisées à cette étape mais le seront après la segmentation par ligne de partage des eaux.

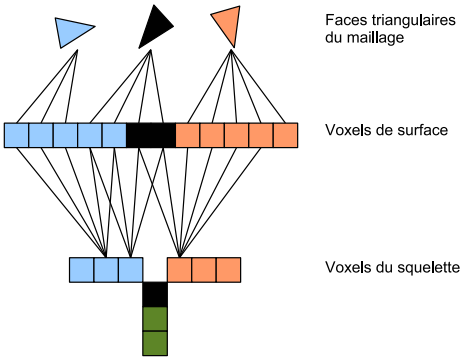


Figure 10 – *Marquage des faces triangulaires du maillage. Les voxels de surface ambigus sont labelisés en noir ; leurs faces associées ne vont pas servir de marqueurs pour la LPE. Seules les faces liées à des voxels de couleur non noire dans l'exemple pourront servir de marqueurs.*

4 Résultats

Figure 11 – *Marqueurs générés par squelettisation à gauche et résultat de la segmentation par LPE avec marqueurs à droite.*

Les expérimentations ont été réalisées avec un ordinateur équipé d'un processeur cadencé à 2.8Ghz et de 512Mo de mémoire. La figure 11 montre les résultats de la génération automatique des marqueurs et de la segmentation par ligne de partage des eaux. La résolution utilisée correspond à 100 voxels pour la dimension longueur, hauteur ou largeur maximale. Les marqueurs font ressortir les principales régions et la méthode de ligne de partage des eaux établie une frontière entre ces régions par rapport à la courbure de la surface du modèle.
La segmentation apparaît comme très efficace avec la combinaison de la génération automatique des marqueurs et la

ligne de partage des eaux. Les processus de squelettisation et de marquage offrent d'excellents marqueurs pour la ligne de partage des eaux. Les principales parties des modèles sont repérées et la finesse du marquage peut être paramétrée à partir de la résolution de voxels.

Modèles	Nombre de vertex	Volume en voxels	Temps de calcul (s.)	
			Marquage	LPE
Cow	2903	38321	15.6	0.062
Triceratops	4152	22282	6.9	0.125
Dinosaur	42146	15582	5.2	9.25

Tableau 1 – *Temps de calcul du processus de marquage et de la segmentation par ligne de partage des eaux.*

Le nombre de vertex d'un modèle va directement influencer le temps de calcul de la LPE. Le temps de calcul du marquage (qui inclut la squelettisation) dépend fortement du volume en voxels du modèle. Le tableau 1 fait apparaître une comparaison des différentes caractéristiques des modèles. Le modèle "Cow" contient le plus petit nombre de vertex et le calcul de la ligne de partage des eaux sera pour cela le plus rapide (62ms), cependant son volume est 1.7 fois plus important que celui du "Triceratops" et 2.4 fois plus grand que celui du "Dinosaur" ; le temps de calcul du marquage sera donc assez conséquent. Le modèle "Dinosaur" est avantagé par son faible volume mais pénalisé par son nombre important de vertex, il en résultera une squelettisation rapide mais un temps de calcul plus grand pour la LPE.

5 Conclusions

Dans cet article, nous avons présenté une nouvelle méthode pour calculer automatiquement les marqueurs pour la segmentation par ligne de partage des eaux sur des maillages triangulaires. Les marqueurs permettent de faire apparaître les régions importantes du modèle et les régions incertaines sont laissées au processus de segmentation de surface. Les marqueurs générés automatiquement sont bien positionnés et permettent ainsi de s'affranchir de la lourde tâche du marquage manuel. La méthode offre de bons résultats et les principales parties des modèles sont détectées. Nos expérimentations ont montré qu'une résolution d'une centaine de voxels, pour la dimension la plus grande l'objet, était suffisante au processus de segmentation pour repérer les principales parties du modèle.

Remerciements

Les modèles utilisés proviennent des sites Internet de Cyberware (www.cyberware.com) et Avalon.

Références

[1] Sagi Katz et Ayellet Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics*, 22(3) :954–961, July 2003.

[2] Kenong Wu et Martin D. Levine. 3d part segmentation using simulated electrical charge distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11) :1223–1235, 1997.

[3] D. L. Page, A. Koschan, et M. Abidi. Perception-based 3d triangle mesh segmentation using fast marching watersheds. Dans *Proceedings of the Computer Vision and Pattern Recognition Conference*, volume II, pages 27–32, June 2003.

[4] Guillaume Lavoué, Florent Dupont, et Atilla Baskurt. A new cad mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*, 37(10) :975–987, September 2004.

[5] David Brunner et Guido Brunnett. Mesh segmentation using the object skeleton graph. Dans *International Conference on Computer Graphics and Imaging*, pages 48–55, Kauai, Hawaii, USA, August 2004.

[6] Kálmán Palágyi, E. Sorantin, E. Balogh, A. Kuba, Cs. Halmai, B. Erdôhelyi, et K. Hausegger. A sequential 3d thinning algorithm and its medical applications. Dans *in Proc. 17th Int. Conf. Information Processing in Medical Imaging*, 2001.

[7] H. Digabel et C. Lantuéjoul. Iterative algorithms. Dans *2nd European Symp. Quantitative Analysis of Microstructures in Material Science*, pages 85–99, Caen, France, 1978. Biology and Medicine.

[8] Serge Beucher. Watershed, hierarchical segmentation and waterfall algorithm. Dans *Mathematical morphology and its applications to image processing*, pages 69–76. Kluwer Academic Publishers, 1994.

[9] Serge Beucher. Algorithmes sans biais de ligne de partage des eaux. Rapport technique, Centre de Morphologie Mathématique de l'École des Mines de Paris, Avril 2004.

[10] Jonathan Betser, Sébastien Delest, et Romuald Boné. Segmentation 3d hiérarchique par ligne de partage des eaux sans biais. Dans *Actes de 15ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2006)*, Tours, Janvier 2006.

[11] Mark Meyer, Mathieu Desbrun, Peter Schröder, et Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. Dans Hans-Christian Hege et Konrad Polthier, éditeurs, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg, 2003.

[12] Alan P. Mangan et Ross T. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *IEEE Transactions On Visualization And Computer Graphics*, 5(4) :308–321, October-December 1999.

[13] Evaggelia-Aggeliki Karabassi, Georgios Papaioannou, et Theoharis Theoharis. A fast depth-buffer-based voxelization algorithm. *J. Graph. Tools*, 4(4) :5–10, 1999.

Recalage de masses de données 3D/couleur non structurées

Sébastien DRUON, André CROSNIER et Marie-José ALDON

[1] LIRMM, UMR 5506 Université Montpellier II - CNRS

[2] 161 rue Ada, 34000 Montpellier - France

[3] {druon, crosnier, aldon}@lirmm.fr

Résumé

Cette étude concerne le recalage de nuages de points denses et non structurés pour la construction de modèles 3D/couleur d'objets complexes. Notre objectif est d'accroître les performances de l'algorithme ICP classique en utilisant l'information couleur attachée aux points, de manière à pouvoir traiter de grandes masses de données, mais aussi recaler précisément des données issues d'objets pour lesquels l'information géométrique n'est pas discriminante. Après avoir rappelé brièvement le principe du recalage itératif basé sur la recherche du point le plus proche dans l'espace 3D (Iterative Closest Point) et des travaux de recherche réalisés dans ce domaine, nous proposons une nouvelle variante de cet algorithme qui permet d'améliorer la sélection des points. L'information couleur y est utilisée comme une contrainte pour réduire la taille de l'espace de recherche durant la phase de mise en correspondance. La validation expérimentale est réalisée avec les images 3D/couleur résultant de la numérisation à haute définition et sous divers points de vue de trois types d'objets.

Mots clefs

Images 3D/couleur, recalage, mise en correspondance.

1 Introduction

La construction automatique de modèles d'objets ou de scènes 3D ou 3D texturés est un problème récurrent pour des applications nombreuses et diversifiées. En effet, ces modèles peuvent être utilisés pour représenter l'environnement dans lequel navigue un robot, pour construire des scènes de réalité virtuelle à partir de l'observation d'objets réels, pour représenter des monuments historiques qui ont été numérisés en vue de leur restauration, ou encore pour alimenter une base de données d'objets de musée consultable par un large public. La construction d'un modèle 3D comporte trois étapes : l'acquisition d'un ensemble d'images 3D sous différents points de vue, leur recalage dans un repère unique, la simplification des données obtenues pour éliminer les redondances résultant du recouvrement des images et le calcul d'une représentation surfacique de l'objet.

Deux grandes catégories de méthodes sont utilisées pour acquérir les données. La première qui est basée sur la vision active en 3D (mesure du temps de vol, vision en lumière structurée, triangulation laser) fournit des images denses de la scène observée (images de distance et parfois images vidéo). La deuxième catégorie qui fait appel à une ou plusieurs caméras vidéo (stéréovision, vision en mouvement) permet d'acquérir la texture de la scène ainsi qu'une structure 3D minimale (par exemple, un ensemble de points d'intérêt). Mais en règle générale, on ne dispose pas d'une image 3D dense décrivant précisément la géométrie de la scène. Pour numériser complètement un objet ou une scène 3D il faut acquérir, sous des points de vue différents, un ensemble d'images qui se recouvrent partiellement. Le nombre de vues nécessaires dépend de la complexité et de la taille de l'objet, mais également de la résolution du capteur. On obtient donc un ensemble d'images qu'il faut recaler. Ainsi, un scanner 3D fournit des nuages de points non structurés qu'il faut ramener dans un repère unique. Lorsque ces images sont recalées deux par deux on parle de recalage simple, lorsqu'elles sont recalées toutes ensemble, il s'agit d'un recalage global [1].

Ce travail de recherche est réalisé dans le cadre de la création de modèles 3D/couleur à haute résolution d'objets de musée (peintures à l'huile, sculptures, figurines archéologiques, …). Dans ce but, nous développons une approche intégrée pour la construction de modèles 3D texturés qui tire parti du caractère complémentaire des données vidéo et 3D fournies par les scanners à lumière structurée. Ce type de capteur présente l'avantage de fournir simultanément ces deux types de données dans le même repère (repère attaché à la caméra).

Le sujet de cette publication est le recalage de deux nuages de points 3D/couleur denses et non structurés. Notre objectif est d'améliorer les performances de l'algorithme ICP (Iterative Closest Point) classique [2,3] en exploitant les données couleur attachées aux points 3D fournis par le capteur, afin d'être en mesure de traiter :

- de grands ensembles de données (plusieurs millions de points par nuage),

- des objets pour lesquels l'information géométrique n'est pas suffisante (relief peu marqué, symétries conduisant à des solutions multiples, …).

Le papier est organisé de la manière suivante. Dans la section suivante, nous présentons l'algorithme ICP utilisé pour recaler des nuages de points 3D et les travaux de recherche développés pour améliorer ses performances. En section 3, nous proposons une nouvelle méthode qui utilise l'information couleur comme une contrainte dans la phase d'appariement des points 3D. La dernière partie décrit les résultats d'une validation expérimentale réalisée avec des images 3D/couleur issues de la numérisation à haute définition et sous divers points de vue de 3 types d'objets : une statue en bois peint, une boite cylindrique en ivoire avec des motifs colorés, et une peinture à l'huile.

2 Etat de l'art

Recaler deux ensembles de données consiste à estimer la transformation rigide qui permet de les représenter dans un repère unique [4]. Les données à recaler peuvent être de nature diverse : images d'intensité, images surfaciques, images volumiques ou images multimodales (par exemple : images de distance texturées). Nous présentons ci-après un aperçu des techniques de recalage basées sur l'algorithme ICP qui sont applicables aux images de distance prises de différents points de vue pour numériser un objet ou une scène. Le recalage itératif basé sur la recherche du point le plus proche (Iterative Closest Point) [2,3] est une des techniques les plus couramment utilisées pour aligner deux nuages de points dans l'espace 3D.

2.1 L'algorithme ICP

Cet algorithme estime la transformation de mouvement rigide entre les deux formes 3D en considérant que :

- elles se recouvrent partiellement, c'est à dire qu'un sous-ensemble de points est commun aux deux nuages,
- elles sont a priori partiellement alignées, autrement dit que l'on dispose d'une estimation de la transformation initiale.

L'algorithme comporte deux étapes : la première génère un ensemble de correspondances temporaires entre points des deux nuages, la deuxième estime la transformation rigide qui relie ces points appariés.

Considérons deux nuages de points $\{p_i\}$ et $\{p'_i\}$ qui représentent les images d'un même objet acquises par un capteur de distance sous deux points de vue différents. D'un point de vue géométrique, on peut considérer que le point p_i correspondant au point p'_i, son plus proche voisin dans $\{p_i\}$, peut être défini par :

$$p_i = \arg\left[\min_{p_j \in \{p_i\}} \left\| p_j - p'_i \right\|\right] \qquad (1)$$

Pour chaque paire de points mis en correspondance p_i et p'_i, nous devons estimer une matrice de rotation R (3x3) et un vecteur de translation t (3x1) tels que :

$$p_i = R\, p'_i + t \qquad (2)$$

Pour simplifier l'écriture, nous utiliserons la notation en coordonnées homogènes qui fait appel à la matrice T fonction des 6 paramètres de mouvement. Théoriquement, 3 paires de points sont suffisantes pour identifier la transformation T. En pratique, compte tenu du bruit qui entache les données et de la présence de zones d'occultation dans les images, des erreurs d'appariement subsistent. Pour estimer les paramètres du mouvement, on fait appel à une technique de minimisation d'erreur. Généralement [5], on recherche la solution au sens des moindres carrés, d'un système surdéterminé de N équations ($N > 3$).

Le critère à minimiser s'écrit :

$$\varepsilon = \frac{1}{N}\sum_{i=1}^{N} \left\| p_i - T\, p'_i \right\|^2 \qquad (3)$$

Cette procédure en deux étapes est itérée jusqu'à la convergence, c'est à dire jusqu'à ce que la variation du critère (3) reste inférieure à un seuil fixé a priori en fonction de la précision souhaitée pour le recalage. L'algorithme ICP peut être résumé de la façon suivante :

Soit T_0 une estimation de la transformation initiale

Répéter, pour $k = 1 \ldots k_{\max}$, ou jusqu'à ce que le critère soit atteint

Trouver un ensemble de N_k paires de points les plus proches dans $\{p_i\}$ et $\{p'_i\}$, selon (1);

Estimer la transformation T_k qui minimise le critère de distance (3)

Appliquer la transformation T_k à tous les points de $\{p'_i\}$.

Besl et Mc Kay [2] montrent que l'algorithme ICP converge de manière monotone vers le minimum local le plus proche. Lorsque l'initialisation n'est pas assez précise, ce type de faux appariement est fréquent durant les premières itérations de l'algorithme et conduit généralement à une estimation biaisée.

2.2 Variantes de l'algorithme ICP

De nombreuses variantes ont été proposées pour améliorer l'algorithme ICP introduit par Besl et Mc Kay. Le lecteur trouvera dans [5] une classification et une comparaison expérimentale de ces différentes solutions. Nous donnons ici un aperçu des principales approches qui ont pour objectif de réduire le volume des calculs nécessaires pour la mise en correspondance et/ou de rendre les appariements plus robustes vis à vis des différentes sources de bruit et de l'imprécision de l'initialisation. Alors que dans [2] tous les points disponibles sont utilisés dans la phase d'appariement, beaucoup de méthodes

n'utilisent qu'une partie des points de l'une ou des deux images. Ainsi, dans [7] un tirage aléatoire permet d'utiliser des échantillons de points différents à chaque itération. Un k-D tree est parfois mis en œuvre pour accélérer la recherche des points les plus proches [9].
Des attributs invariants par rapport au point de vue peuvent être utilisés pour caractériser les points durant la phase d'appariement. Ils peuvent être calculés soit à partir de l'image de distance, soit à partir d'informations complémentaires fournies par le capteur de numérisation (intensité, couleur, …). Dans [8], un vecteur d'attributs géométriques et photométriques est attaché à chaque point de l'image. Il inclut les amplitudes des courbures principales calculées à partir de l'image de distance, ainsi que des données photométriques extraites de l'image couleur (coefficients de réflexion diffuse). Des données photométriques sont également exploitées dans [10] et [11]. Parmi les autres types de descripteurs on trouve : la courbure locale, la texture [9], la forme (attributs différentiels 3D) [7] et l'angle entre les normales [11].
Différentes solutions ont été explorées pour éliminer les faux appariements qui empêchent l'estimateur de converger vers la bonne solution. Dans [9], Zhang élimine les paires de points dont la distance est supérieure à un seuil qui est remis à jour de manière adaptative en fonction des paramètres statistiques de la distribution des distances obtenue au pas précédent. Dans [7], un estimateur LMS (Least Mean Square) minimise la médiane des carrés des résidus et permet de classer les points 3D de chaque image en fonction de la qualité de l'appariement qu'ils génèrent. Les points occultés, mal appariés ou qui n'ont pas de correspondant sont ainsi éliminés de la liste qui sera utilisée pour l'estimation de la transformation. Une autre solution pour accroître la robustesse du processus d'appariement consiste à rajouter des termes supplémentaires qui mesurent la similarité des points dans la fonction de distance (5). Ainsi, dans [12] une distance dans l'espace couleur est rajoutée à la distance dans l'espace 3D. La difficulté dans cette approche est le choix des coefficients de pondération affectés à chacun des termes.
Nous proposons dans cette publication une approche nouvelle pour exploiter la couleur dans l'appariement d'images 3D/couleur. Elle est basée sur une classification préliminaire des données en fonction de leur teinte, afin de sélectionner un nombre réduit de points pour réaliser la mise en correspondance.

3 Appariement 3D contraint

La contribution de cette recherche porte donc essentiellement sur l'utilisation de la couleur pour sélectionner les points à apparier. L'approche utilisée consiste à classifier les points en fonction de leur couleur, en ayant pour objectifs :

- d'améliorer la robustesse de la mise en correspondance en autorisant uniquement l'appariement de points de couleur compatible,
- de réduire le nombre de données à traiter,
- de traiter le cas d'objets pour lequel le recalage ne peut pas être résolu en utilisant uniquement l'information 3D (formes symétriques, relief assez plat, …)

3.1 Classification basée sur la couleur

La segmentation de chaque nuage est effectuée à partir de la teinte H des points calculée selon la définition de l'espace HSV proposée dans [13]. La saturation S et l'intensité V ne sont pas considérées car elles varient avec les conditions d'acquisition. En effet, lorsque la numérisation est réalisée en éclairage ambiant, la variation de couleur d'un objet 3D provient essentiellement des variations de zones d'ombre entre les différents points de vue. Ces variations affectent seulement l'intensité des points, mais pas leur couleur intrinsèque. De plus, l'utilisation de la teinte comme critère de sélection des points ne nécessite pas de modèle d'illumination comme dans [8] où les auteurs calculent les paramètres de réflexion de la surface.
Nous considérons un espace couleur où la teinte H varie de 0 à 6, chaque valeur entière étant associée à une couleur pure. Une classe couleur est caractérisée par un petit intervalle de teinte centré sur une de ces valeurs. Pour chaque nuage de points, nous générons les six sous-nuages correspondant à chaque classe, les points restants n'étant pas pris en considération. Lorsque la saturation S est inférieure à un seuil, la composante teinte ne semble pas significative, c'est pourquoi nous filtrons la composante S.

3.2 Critère pour la sélection des classes

Actuellement l'algorithme de recalage utilise uniquement les points appartenant à une seule classe C, choisie en fonction de deux critères :

- Le ratio *Rc* de points contenus dans la classe. Une bonne classe doit avoir un nombre raisonnable de points. Par exemple, dans les expérimentations qui suivent, nous avons :

$$5\% < Rc < 15\% \quad (4)$$

Si une classe contient moins de 5% des données initiales, nous considérons que c'est un mauvais représentant de l'ensemble des données. La limite supérieure permet de réduire le temps de calcul en favorisant la sélection de régions d'intérêt de faible taille.

- Le ratio *Nc* de points de la classe C initialement inclus dans la boite englobante du nuage 2. Ce critère est destiné à favoriser les classes inclues dans la zone de recouvrement des deux nuages. Etant donné que nous n'avons aucune information sur cette zone, nous

utilisons la transformation initiale fournie par l'utilisateur, et considérons que l'intersection entre les deux boites englobantes doit être représentative de cette région. Une classe possédant un grand nombre de points dans cette intersection est considérée comme plus favorable à un bon recalage.

Un score Sc est calculé pour chaque classe selon la relation :

$$Sc = Nc + f(Rc) \quad (5)$$

où $f(Rc) = 100$ si (4) est respectée, 0 sinon.

3.3 Recalage itératif

Notre algorithme ICP utilise un seuil adaptatif *Dmax* pour éliminer les mauvais appariements, comme dans [9].

- Soit T_0 l'estimation initiale de la transformation
- Initialiser *Dmax* à 20 fois la résolution des images
- Répéter pour $k = 1 \ldots k_{max}$ ou jusqu'à ce que le critère soit atteint
  - Initialiser à vide la liste L_k des paires retenues
  - Pour chaque point $\{\overline{p}_i\}$ de la classe sélectionnée
    - Rechercher son plus proche voisin dans $\{\overline{p}'_i\}$.
    - Si *distance* ($\overline{p}_i$, $\overline{p}'_i$) < *Dmax,*
      ajouter la paire à la liste L_k des paires enregistrées
  - Calculer la moyenne μ et l'écart type σ des distances dans L_k
  - Faire $Dmax = \mu + \sigma$
    - Estimer la transformation T_k à partir des paires de L_k
    - Appliquer la transformation T_k à tous les points de $\{\overline{p}'_i\}$
- Répéter la boucle jusqu'à ce que la moyenne μ et l'écart type σ soient stabilisés.

Pour accélérer le processus de recherche du plus proche voisin, on utilise un k-D tree, selon la méthode décrite dans [9]. La complexité de l'algorithme de recalage est :

$$n_1 \, . \, n_2 \, . \log (n_2)$$

où n_1 et n_2 sont les nombres de points de chaque nuage. Si k_{max} itérations sont exécutées sans atteindre la convergence, un nouveau recalage est exécuté avec la classe qui a le second meilleur score, et ainsi de suite.

4 Validation sur données réelles

Nous décrivons ici une évaluation de l'algorithme avec des données réelles provenant de la numérisation de trois types d'objets avec le système TRITOS développé par la société Breuckmann [14]. Ce capteur utilise une caméra couleur de 1280x1024 pixels, et un projecteur de franges en lumière blanche. Pour chaque point de mesure, il délivre les coordonnées 3D et les composantes RGB.

4.1 Description des données

Les jeux de données utilisées dans cette évaluation sont représentatifs de trois catégories d'objets de musée. Dans ces données, il existe un bruit sur la couleur du essentiellement à la numérisation (chaque composante couleur est codée seulement sur 5 bits), et aux variations d'éclairement pendant l'acquisition.
En ce qui concerne la qualité des données 3D, on peut remarquer que la statuette Bali (fig.4) a une forme suffisamment complexe pour générer un certain nombre de zones d'ombre pour le capteur. Par contre Ivoire (fig. 5) et Wallis (fig. 6) sont des exemples difficiles à traiter du fait de leur symétrie (cylindre et plan) et de l'absence de point d'intérêt 3D.

Nom de l'objet	Bali (f ig 4)	Ivoire (fig 5)	Wallis (fig 6)
Nature	Statue en bois peint	Boite en ivoire colorée	Peinture à l'huile
Taille du nuage 1 (Nbre de points)	209628	312950	1030658
Taille du nuage 2	247461	306978	1036735
Résolution du capteur (µm) X Y Z	180x180x6	60x60x3	60x60x3
Information 3D	Riche	Pauvre (cylindre)	Pauvre (plan)
Information couleur	Bruitée (reflets)	Riche	Riche

Table1 : Description des données

4.2 Résultats

Les résultats obtenus avec notre algorithme de recalage sont résumés dans la table 2. Pour chaque expérience, nous donnons, le nombre de points dans la classe sélectionnée pour chaque nuage, l'erreur moyenne finale en mm, le nombre d'itérations nécessaires pour atteindre la convergence, et le nombre final de paires de points dans L_k. Nous considérons que la convergence est atteinte quand les deux conditions suivantes sont vérifiées pour deux itérations successives de l'algorithme :

- le nombre de paires dans L_k reste constant,
- les variations de l'erreur moyenne et de l'écart type sont inférieures à 1e-6 mm.

On constate que, dans chaque expérience, l'erreur finale moyenne est proche de la résolution du capteur. Ceci confirme la qualité du recalage final.
Exécuté sur un Pentium Dell 4- 2,9 GHz – 1Go Ram, sans optimisation du code source, le recalage de deux nuages de points a demandé quelques minutes pour Ivoire et Bali. Par contre, le temps de calcul reste important pour Wallis. En effet, l'absence de relief significatif dans la direction *Z* se traduit par une structure de kD-tree dégénérée et donc un temps de mise en correspondance beaucoup plus long. Les courbes présentées dans la figure 1 montrent

l'évolution de la moyenne et de l'écart type des distances entre les nuages de points recalés, au cours des itérations successives lors du recalage de Bali. La figure 2 montre l'évolution du nombre de paires dans L_k.
Ces tests montrent qu'un recalage approximatif est obtenu au bout de 25 à 30 itérations. Les figures 3 à 6 visualisent les différents nuages de points 3D avant et après recalage. Il faut souligner que dans le cas d'Ivoire ou de Wallis, il serait impossible d'obtenir une convergence correcte avec un algorithme de recalage purement géométrique.

Nom de l'objet	Bali	Ivoire	Wallis
Classe sélectionnée	Rouge	Vert	Cyan
Pourcentage de points sélectionnés	5.95 %	7.06 %	9.10 %
Erreur moyenne (mm)	0.289311	0.06311	0.140817
Itérations	58	102	201
Nbre final de paires dans L_k	5677	11419	40473

Table 2 : Bilan des résultats

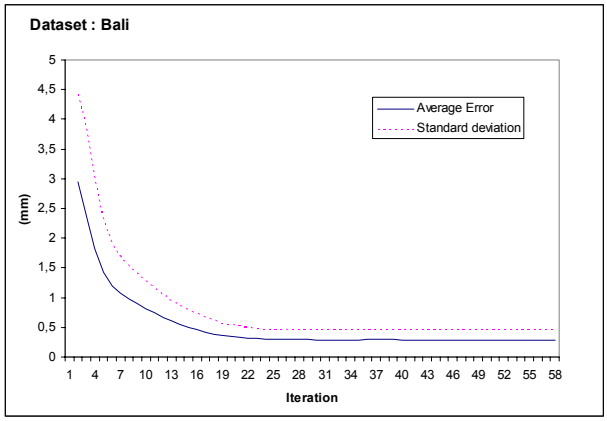


Figure 1 : Evolution de l'erreur de recalage avec Bali

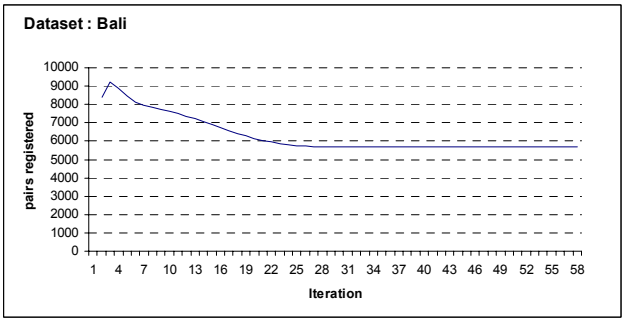


Figure 2 : Evolution du nombre de paires avec Bali

5 Conclusion

Nous avons proposé une nouvelle variante de l'algorithme ICP qui exploite l'information couleur pour permettre le recalage de grands ensembles de points 3D/couleur. Une segmentation préliminaire des données permet de sélectionner des points appartenant à une ou plusieurs classes couleur intéressantes, de façon à accélérer le processus d'appariement 3D, et à augmenter la vitesse de convergence de l'algorithme de recalage itératif.
La validation expérimentale effectuée avec des données réelles montre que :

- le temps de calcul est réduit,
- la méthode fournit une estimation précise de la transformation rigide, y compris dans le cas où le recalage géométrique est difficile à résoudre avec un algorithme ICP classique.

Les améliorations en cours portent sur le processus de segmentation des données couleur. Un algorithme de clustering automatique est développé pour optimiser la sélection des classes. Par ailleurs, il est possible de réduire les temps de calcul en parallélisant un certain nombre de tâches. Nos travaux futurs porteront également sur une validation expérimentale plus complète, et en particulier sur une comparaison avec d'autres méthodes [7, 8, 9].

6 Références

[1] R. Ben-Jemaa, F. Schmitt, "Recalage 3D", *Images de profondeur*, éditions Hermès, 2002.

[2] P. Besl and N.D McKay, "A Method for Registration of 3-D Shapes", *Proc. of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (2): 239-256, 1992

[3] Y. Chen and G. Medioni, "Object Modelling by Registration of Multiple Range Images", *Image and Vision Computing*, 10 (3): 145-155, 1992.

[4] S. Seeger, X. Laboureux, " Feature Extraction and Registration", *Principles of 3D Image Analysis and Synthesis*, Kluwer Academic Pub, 2000.

[5] S. Rusinkiewicz and M. Levoy, "Efficient Variant of the ICP Algorithm". *Actes 3DIM*, pages 145-152, Canada, Québec, juin 2001.

[6] G. Blais and M. D. Levine, "Registrating Multi-View Range Data to Create 3-D Computer Objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8) : 820-824, 1995.

[7] T. Masuda and N. Yokoya, "A Robust Method for Registration and Segmentation of Multiple Range Images", *Computer Vision and Image Understanding*, 61(3): 295-307, mai 1995.

[8] G. Godin, D. Laurendeau, R. Bergevin, " A method for the registration of Attributed Range Images", *Actes 3DIM*, pages 179-186, Québec, Canada, juin 2001,.

[9] Z. Zhang, "Iterative Point Matching for Registration of Free Form Curves", *International Journal of Computer Vision*, 13 (2): 119-152, 1994.

[10] S. Weik, "Registration of 3D partial surface models using luminance and depth information", *Actes 3DIM*, pages 93-100, Ottawa, Canada, mai 1997.

[11] K. Pulli, "Multiview Registration for Large Data Sets", *Actes 3DIM*, pages 160-168, Ottawa, Canada, octobre 1999.

[12] A. Johnson and S. Bing Kang, "Registration and Integration of Textured 3-D Data", *Actes 3DIM*, pages 234-241, Ottawa, Canada, mai 1997.

[13] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, "Computer graphics: principles and practice", *Addison Wesley*, 1992.

[14] http://www.breuckmann.com/HTML/engl/tritos.html, site web du capteur Tritos, société Breuckmann

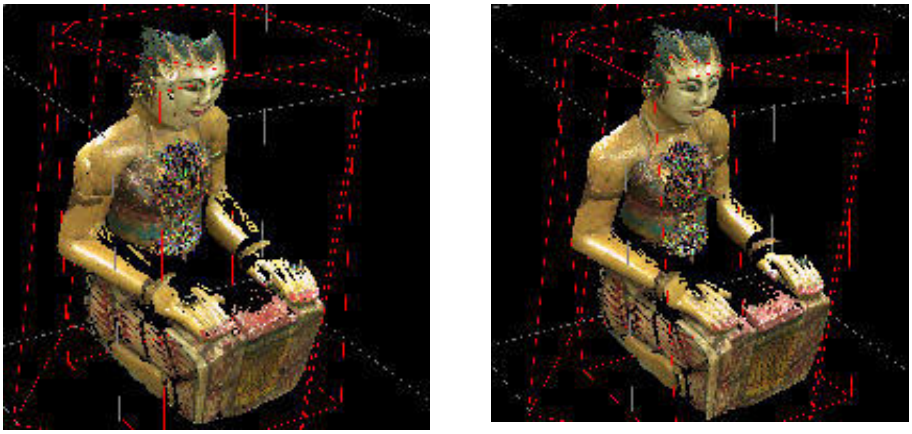


Figure 3 : Bali avant et après recalage

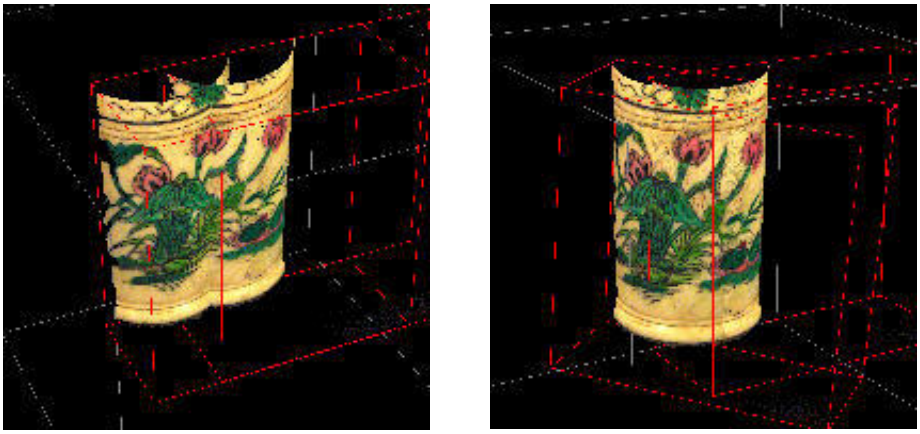


Figure 4 : Ivoire avant et après recalage

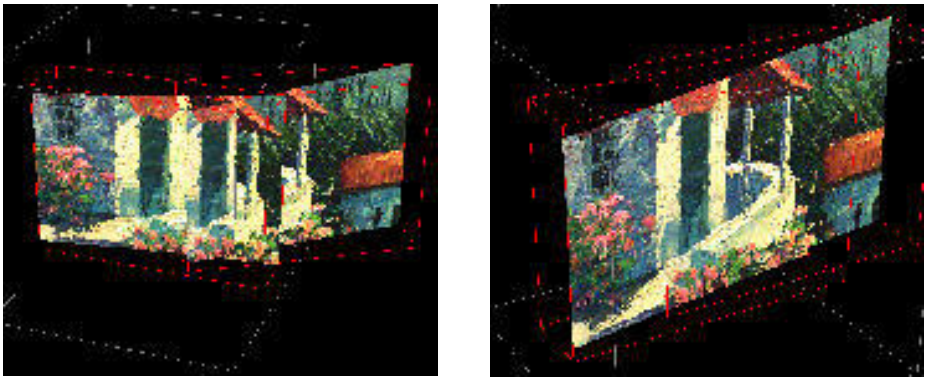


Figure 5 : Wallis avant et après recalage

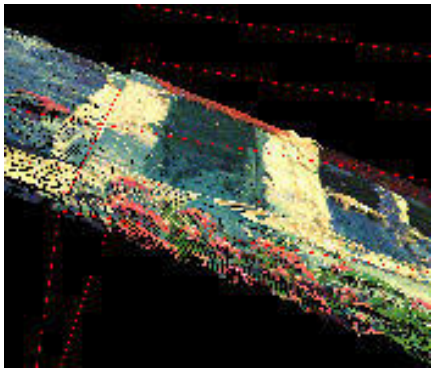


Figure 6: Vue agrandie de la surface de Wallis après recalage

Reconnaissance de Visages 3D Utilisant l'Analyse de Formes des Courbes Faciales

C. Samir[1] M. Daoudi[1] A. Srivastava[2]

[1] FOX-MIIRE Research Group (LIFL UMR USTL-CNRS 8022)

(GET/INT - ENIC Telecom Lille 1)
Rue G. Marconie Cité Scientifique 59650 Villeneuve d'Ascq

{daoudi, chafik}@enic.fr
[2] Department of Statistics

Florida State University, Tallahassee
FL 32306, USA

anuj@stat.fsu.edu

Résumé

Dans ce papier, nous présentons une nouvelle méthode pour la reconnaissance de visages 3D. Nous proposons de comparer deux surfaces faciales à travers les formes des courbes faciales. L'idée de base est d'approximer grossièrement la surface faciale S par un ensemble fini de courbes de niveau, appelés courbes faciales, d'une fonction F sur S.
En utilisant la géométrie riemannienne nous définissons la notion de chemin géodésique entre deux surfaces, et la distance entre deux surfaces.
Des résultats expérimentaux sur la base Notre Dame démontrent l'efficacité de notre approche pour la reconnaissance de visages 3D. En effet, la courbe ROC montre que si on accepte 1% de faux positif alors on obtient 97%.

Mots Clef

reconnaissance de visages 3D, chemin géodésique, image de profondeur, métriques sur les formes de visages, courbes faciales.

1 Introduction

La reconnaissance automatique de visage humain basée sur le traitement des images 2D s'est bien développée ses dernières années, et plusieurs techniques ont été proposées. Malgré les résultats obtenus dans ce domaine, la reconnaissance robuste de visage reste un problème très difficile. Les méthodes actuelles sont efficaces lorsque les conditions de prise de vue des images tests sont similaires à celles des images d'apprentissage. Cependant, la grande variabilité générée par le changement de luminosité et le changement de prise de vue causent de sérieux problèmes pour de nombreux systèmes de reconnaissance existants. Une solution à ce problème, est l'utilisation de l'information tridimensionnelle du visage car elle permet une meilleure information sur les caractéristiques du visage humain dans l'espace 3D [6]. Cette information procure une invariance relative à la lumière et aux conditions de prise de vue. En plus, les avancées récentes en imagerie 3D (outils d'acquisition, modeleurs, cartes graphiques, etc.) rendent possibles la création et le stockage des visages 3D.
Etant donné un visage scanné 3D, le but de ce travail est de développer des algorithmes pour comparer les formes des visages 3D dans un objectif de reconnaissance de visages[11].
Plusieurs approches utilisent les propriétés différentielles des surfaces comme les courbures principales maximales, minimales permettant la segmentation de la surface [3] en plusieurs régions selon leur concavité, convexité et points de selles. Ces caractéristiques locales de la surface peuvent offrir des outils intéressants pour la reconnaissance d'un visage 3D. Dans [10] un minimum de trois points est nécessaire pour établir un alignement de la surface du visage 3D, les points choisis dans ce travail sont le nez et les yeux. Les auteurs utilisent l'index de courbure sur le maillage pour l'extraction de ces points. Après l'alignement, une transformation rigide est estimée par l'algorithme ICP en proposant un algorithme combinant les résultats obtenus dans [5] et [4].
Chafik et al. [2] proposent une approche topologique, ils utilisent les graphes de Reeb en enrichissant leurs noeuds par la courbure moyenne comme information géométrique.
Les méthodes utilisées actuellement pour la reconnaissance des visages 3D ont plusieurs inconvénients. D'une part, elles utilisent des propriétés différentielles des surfaces telles que les courbures. Elle sont donc très sensibles aux bruits. Et d'autre part, deux surfaces représentant deux visages avec des expressions faciales différentes se-

ront classés d'une manière similaire car elles sont basées essentiellement sur la comparaison de surfaces, en essayant de trouver la meilleur transformation euclidienne. En effet, les visages ne peuvent pas être traités comme des objets rigides puisqu'ils peuvent subir des déformations dues aux expressions faciales.
Le but de notre travail est de caractériser les formes des surfaces des visages modulo des déformations qui correspondent aux déformations faciales.
Dans ce papier, nous proposons un cadre où la comparaison entre formes peut être potentiellement indépendante du choix de la représentation de la forme. Notre approche est de *représenter une surface en utilisant une famille de courbes fermées* [8][1]. Ces courbes seront calculées comme les courbes à niveau d'une fonction continue F de la surface du visage, et les indexes correspondent aux valeurs de cette fonction. Nous pouvons utiliser par exemple **la fonction de profondeur** (valeur des coordonnées z) comme fonction où les courbes à niveau fournissent les courbes des visages désirées. Les formes de deux surfaces de visages seront comparées par la comparaison des formes de courbes faciales des visages. Cette approche utilisant les formes des courbes de niveau pour analyser les formes des surfaces est plus générale et elle est indépendante des limitations associées au choix de F. En fait, elles existent certaines fonctions dont les courbes de niveau sont invariantes aux transformations rigides des surfaces, et sont adaptées pour ce type d'analyse [8]. Autrement dit, notre méthode est adaptée pour comparer des courbes planes, une généralisation dans le cadre des courbes 3D est en cours de rédaction dans un autre papier.
Le reste du papier est organisé de la façon suivante : la section 2 décrit notre représentation faciale utilisant les formes des courbes faciale et les métriques pour comparer les courbes extraites à partir du maillage 3D du visage. La section 3 présente des résultats expérimentaux sur la base Notre Dame[7]. Nous terminerons par une conclusion résumant les points forts de notre approche.

2 La représentation de la forme des courbes faciales

Soit S une surface dénotant le visage scanné. Bien qu'en pratique la surface S soit représentée par un maillage triangulaire définie par des sommets et des arrêtes, nous commencerons notre discussion en supposant la continuité de la surface S . D'une manière précise, elle est plongée dans l'hémisphère supérieure $\mathbb{S}^2_+$ dans $\mathbb{R}^3$. Dans cette définition, nous avons ignoré (ou rempli) les trous dans S associés aux yeux, et à la bouche. La figure 1 montre des exemples de la surface S, de la même personne avec différentes expressions faciales.

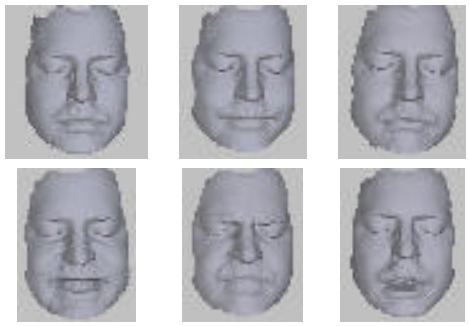


FIG. 1 – *Exemple de surface faciale d'une personne avec différentes expressions faciales.*

FIG. 2 – *Exemple de courbes faciales C_λ pour une surface S. Système local de coordonnées attaché à un visage.*

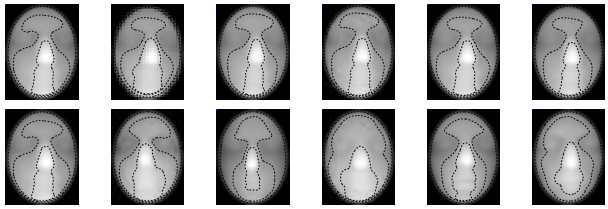


FIG. 3 – *Trois courbes faciales pour chaque surface. Haut : six expressions faciales, la même personne. Bas : la même expression faciale, six personnes différentes.*

Soit $F : S \mapsto \mathbb{R}$ une fonction continue définie sur S. Soit C_λ la ligne de niveau de F, appelée aussi **courbe faciale**, pour la valeur $\lambda \in \mathbb{R}$, i.e.

$$C_\lambda = \{p \in S | F(p) = \lambda\} \quad \subset S,$$

Nous pouvons reconstruire S à partir des courbes de niveau $S = \cup_\lambda C_\lambda$. La figure 2 montre des exemples de courbes de niveau d'un visage. En principe, la collection des courbes de niveau $\{C_\lambda | \lambda \in \mathbb{R}_+\}$ contient toutes les informations sur S et nous pouvons analyser la surface S à travers les courbes C_λ. En pratique, cependant, un échantillonnage fini de λ restreint la représentation à une approximation grossière de la forme de la surface S.

Dans ce papier, nous avons choisi de représenter F par **la fonction de profondeur**. Par conséquent $F(p) = p_3$, la composante z du vecteur $p \in \mathbb{R}^3$. Notre but est d'analyser

la forme de S invariante sous l'action du groupe des similitudes $\mathbb{SE}(3) \equiv \mathbb{SO}(3) \ltimes \mathbb{R}^3$ sur la surface S ($\ltimes$ implique un produit semi direct, qui signifie que la rotation est toujours appliquée avant la translation). Par conséquent, nous préférons former les ensembles de niveau C_λ dont les formes sont invariantes à l'action du groupe $\mathbb{SE}(3)$ sur S. Nous allons maintenant étudier la variabilité des courbes de niveaux de F respectant ces transformations. Récrivons $\mathbb{SE}(3)$ comme $(\mathbb{SO}(2) \times \mathbb{S}^2) \ltimes (\mathbb{R}^2 \times \mathbb{R}^1)$, où nous pouvons interpréter $\mathbb{SO}(2) \ltimes \mathbb{R}^2$ comme une transformation rigide dans le plan $x-y$, i.e. perpendiculaire à l'axe z, $\mathbb{S}^2$ comme la direction de l'axe z, et $\mathbb{R}$ la translation dans la direction de z. Nous supposons que les axes $x-y-z$ forment un *système de coordonnées cartésiennes* lié au modèle, tel que l'axe z est aligné avec la direction de la camèra, comme le montre la figure 2. Comme nous allons décrire plus tard, notre technique de comparaison des formes des contours fermés est invariante aux transformations planes dans $\mathbb{SO}(2) \ltimes \mathbb{R}^2$ et la translations selon z dans $\mathbb{R}$. Cependant, nous n'avons aucun moyen direct pour supprimer les variabilités dues aux changements dans la direction de z qui varie autour de $\mathbb{S}^2$; nous supposons que nous avons un moyen pour l'alignement de la surface du visage de la rotation de telle sorte que les axes z soient toujours alignés.

La figure 3 montre des surfaces faciales représentées par trois courbes faciales. On peut voir que celles-ci sont robustes aux déformations dues aux expressions faciales.

2.1 La forme des courbes faciales

Considérons les courbes faciales C_λ fermées, les courbes planes $\mathbb{R}^2$ sont paramétrisées par l'abscisse curviligne. La fonction coordonnée $\alpha(s)$ de C_λ liée à la direction de la fonction $\theta(s)$ selon $\dot{\alpha}(s) = e^{j\,\theta(s)}$, $j = \sqrt{-1}$. Pour rendre les formes invariantes aux rotations planes, nous allons nous restreindre aux fonctions angulaires de telle sorte que, $\frac{1}{2\pi}\int_0^{2\pi} \theta(s)ds = \pi$. Il faut aussi que les courbes soient fermées, θ satisfasse la *condition de fermeture* : $\int_0^{2\pi} \exp(j\ \theta(s))ds = 0$. En résumé, nous allons nous restreindre à l'ensemble $\mathcal{C} = \{\theta |\ \frac{1}{2\pi}\int_0^{2\pi} \theta(s)ds = \pi,\ \int_0^{2\pi} e^{j\theta(s)}ds = 0\}$. Pour supprimer la re-paramétrisation du groupe (différents placements de l'origine, les points avec $s = 0$, sur la même courbe), définie l'espace quotient $\mathcal{D} \equiv \mathcal{C}/\mathbb{S}^1$ comme l'espace des courbes planes continues.
Soient C_λ^1 et C_λ^2 deux courbes faciales associées à deux visages différents, extraites à partir du même niveau λ. Nous nous intéressons à la quantification de la dissimilarité entre ces deux courbes. Soient θ_1 et θ_2 les fonctions angulaires associées aux deux courbes, respectivement. Un outil important de l'analyse Riemannienne des formes est de construire les chemins géodésiques entre les formes et d'utiliser la longueur géodésique comme distance entre formes. Klassen et al. [12] approxime les géodésiques dans $\mathcal{D}$ en dessinant des segments infinitésimales dans $\mathbb{L}^2$ et les projeter dans $\mathcal{S}$. Pour deux courbes $\theta_1, \theta_2 \in \mathcal{S}$, ils utilisent une méthode pour construire des géodésiques entre eux. L'idée de base est de chercher une direction tangente g à la première forme θ_1, de telle sorte que de telle géodésique dans cette direction atteigne la seconde forme θ_2, appelée forme cible, dans une unité de temps. Cette recherche est réalisée en minimisant une "fonction de perte" définie comme une distance dans $\mathbb{L}^2$ entre la forme atteinte et θ_2, en utilisant "'la méthode du gradient "'. La géodésique, respectant la métrique définie dans $\mathbb{L}^2$: $\langle g_1, g_2 \rangle = \int_0^{2\pi} g_1(s)g_2(s)ds$, est dans l'espace tangent de $\mathcal{D}$. Ce choix implique que la géodésique entre deux formes est le chemin qu'utilise le minimum d'énergie pour déformer une forme vers une autre. La figure 4 montre deux exemples de chemins géodésiques entre surfaces faciales. Le premier correspond à un chemin géodesique entre deux surfaces correspondant à la même personne mais avec deux expressions faciales différentes. Le second exemple correspond au chemin géodesique entre deux visages de deux personnes différentes.

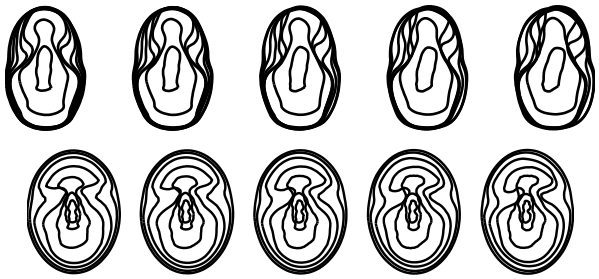


FIG. 4 – *Chemin géodesique entre deux surfaces faciales. Haut : même personne, différentes expressions faciales. Bas : différentes personnes.*

2.2 Une métrique pour la comparaison des courbes faciales

Maintenant que nous avons défini une métrique pour comparer les formes des courbes faciales, il est devenu facile de comparer les formes des surfaces. Supposons que $\{C_\lambda^1 | \lambda \in \Lambda\}$ et $\{C_\lambda^2 | \lambda \in \Lambda\}$ soient deux collections de courbes faciales associées aux deux surfaces, une métrique possible entre ces deux surfaces est : $d_g(S^1, S^2) = \left(\prod_{\lambda \in \Lambda} d(C_\lambda^1, C_\lambda^2)\right)^{1/|\Lambda|}$ où Λ est un ensemble fini de valeurs utilisées pour approximer la surface faciale par les courbes faciales. Les résultats expérimentaux obtenus par [1] montrent que d'autres distances sont possibles. Le choix de Λ est important dans les performances que nous obtiendrons. L'amélioration de la métrique est fonction de la taille de Λ, mais comment choisir les éléments de Λ ?. Dans ce papier, nous avons pris toutes les valeurs de profondeur et nous les avons échantillonnées uniformément pour obtenir Λ.

3 Résultats expérimentaux

En utilisant le cadre général que nous venons de décrire, différentes expériences vont nous permettre d'analyser les formes des surfaces faciales.

3.1 Pré-traitement et extraction des courbes

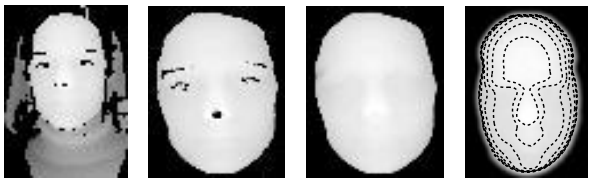
Nous allons maintenant appliquer l'algorithme décrit dans les sections précédentes sur la base publique *Notre dame* [9], [7], utilisée pour comparer différents algorithmes de reconnaissance de visages 3D.

C'est une base de visages 3D représentés sous forme d'images de profondeur. Cependant, cette base nécessite un pré-traitement afin qu'on puisse l'utiliser en pratique. En effet, les visages 3D contiennent des trous, des cheveux et des habilles.

Les Figures 3.1(a)-(d) montrent un exemple de pré-traitement effectué sur les visages pour les rendre utilisables par notre algorithme.

La figure 3.1(a) montre un exemple d'une image de profondeur avec trous, des cheveux et des habilles. La figure 3.1(b) représente le masque associé pour supprimer les cheveux et les habilles, la figure 3.1(c) représente l'algorithme d'interpolation qui permet de remplir les trous et enfin l'étape d'extraction de courbes faciales lisses représentées dans la figure 3.1(d).

En résumé, notre algorithme suit les étapes suivantes : (i) Extractions des courbes faciales à partir des images de profondeur. (ii) Calcul d'une fonction d'angle pour chaque courbe extraite, paramétrisation par l'abscisse curviligne, (iii) calcul des longueurs géodésiques entre les courbes faciales respectives, et calcul des distances entre les surfaces faciales. En terme de temps de calcul, ces trois étapes prennent moins d'une seconde sur un PC de bureau.



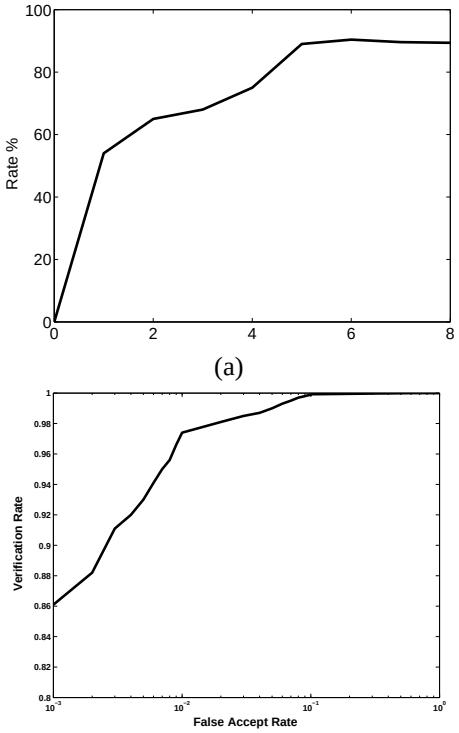
(a) Image de profondeur (b) masque (c) remplissage de trous (d) Extraction de courbes faciales

3.2 Performance de l'algorithme

Bien qu'il y ait 953 visages corespondant à 277 personnes à l'origine, nous avons supprimé des visages dont le maillage avait des grandes parties manquantes. Nous avons divisé les 740 visages scannés restants en une base d'apprentissage de 470 visages et une base test de 270 visages 3D.

Nous avons utilisé l'algorithme du plus proche voisin comme classifieur pour calculer le taux de reconnaissance. La figure 5(a) montre le taux de reconnaissance obtenu pour différentes courbes faciales pour représenter un visage. Cette figure montre que le meilleur taux de reconnaissance de 90.4% est obtenu pour six courbes faciales. L'ajout d'autres courbes faciales n'pas amélioré la performance de notre classifieur. Par la suite, nous utiliserons six courbes faciales pour représenter les visages 3D.

Nous présentons également la courbe ROC (Receiver Operating Characteristic) 5(b) pour évaluer les performances de notre algorithme. Nous avons divisé notre base précédente en trois parties. La première de taille 470 dite base *d'apprentissage*, la seconde de taille 270 dite base *test* dont les visages ont des correspondants dans la base d'apprentissage et enfin une base de taille 30 dite *d'imposteurs* qui n'ont aucun correspondant dans la base d'apprentissage. La courbe ROC est calculée en suivant le même protococle défini dans [14]. Cette courbe montre que si on accepte 1% de faux positif alors on obtient 97%.



(a)

(b)

FIG. 5 – *(a) Taux de reconnaissance en fonction de λ, (b) La courbe ROC utilisant six courbes faciales pour représenter une surface faciale.*

4 Conclusion

Dans ce papier, nous avons décrit une approche géométrique pour comparer deux surfaces faciales à travers les formes des courbes faciales. L'idée de base est d'approximer grossièrement la surface faciale S par un ensemble fini de courbes de niveau, appelés courbes faciales, d'une fonction F sur S. En choisissant F comme

la fonction de profondeur, on utilise des techniques standards d'analyse d'images pour l'extraction des courbes faciales à partir des images de profondeur. Nous comparons les courbes faciales, de même niveau, des surfaces par une analyse de formes décrites dans [12]. Une métrique sur les formes faciales est déduite en cumulant les distances entre les courbes faciales. Les résultats obtenus de reconnaissance et de classification des surfaces faciales sont présentés en utilisant cette métrique.
Nous avons montré que la représentation des surfaces faciales par six courbes faciales permet d'obtenir un taux de reconnaissance de 90.4%.
Des résultats expérimentaux sur la base Notre Dame démontrent l'efficacité de notre approche pour la reconnaissance de visages 3D. En effet, la courbe ROC montre que si on accepte 1% de faux positif alors on obtient 97%.
Des travaux sont en cours pour généraliser les résultats obtenus dans ce papier au cas où les surfaces sont représentés par un ensembles de courbes 3D [13].

Références

[1] C. Samir, A. Srivastava, and M. Daoudi. Automatic 3D Face Recognition Using Shapes of Facial Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence To appear 2006*

[2] C. Samir, M. Daoudi and J.P Vandeborre Automatic 3D face recognition using topological techniques. *IEEE ICME 2005*

[3] A. B. Moreno, A. Sanchez, J. F. Velez, and F. J. Diaz *Face recognition using 3D surfaces-extended descriptors, IMVIP 2003*

[4] J. Lee and E. Milios Matching Range images for Human Faces *Proceedings of the International Conference on Computer Vision pp. 722-726 1990*

[5] Y. Chen and G. Medioni Object Modeling by Registration of Multiple Range Images *Preceedings of the International conference on Robotics and Automation, 1991*

[6] C. Beumier and M. Acheroy. Automatic face authentication from 3D surface. In *British Machine Vision Conference*, 1998.

[7] T. Maurer, D. Guigonis, I. Maslov, B. Pesenti, A. Tsaregorodtsev, D. West and G. Medioni. Performance of Geometrix ActiveID^{TM} 3D Face Recognition Engine on the FRGC Data. In *IEEE Conference on Computer Vision and Pattern Recognition - Workshops*, page 154, 2005.

[8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision*, 64(1) : 5–30, 2005.

[9] K. Chang, K. W. Bowyer, and P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. *Computer Vision and Image Under.*, 101(1) :1–15, 2006.

[10] X. Lu and A. K. Jain. Matching 2.5D Face Scans to 3D Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(1) :31–43, 2006.

[11] P. W. Hallinan, G. G. Gordon, A. L. Yuille, P. Giblin, and D. Mumford. *Two- and Three-Dimensional Patterns of Face*. A. K. Peters, 1999.

[12] E. Klassen, A. Srivastava, W. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(3) :372–383, March, 2004.

[13] A. Menucci, A. Yezzi. Metrics in the space of curves. *Preprint, Scuola Normal Superior*, 2004.

[14] P. Grother, R. Micheals and P. J. Phillips *Face Recognition Vendor Test 2002 Performance Metrics Fourth International Conference on Audio-Visual Based Person Authentication, June 2003*

Compression de maillages 3D dynamiques par modèle de *skinning*

Khaled Mamou Titus Zaharia Françoise Prêteux

Groupe des Ecoles des Télécommunications

Institut National des Télécommunications / Département ARTEMIS
9, rue Charles Fourier 91011 Evry, France

{Khaled.Mamou, Titus.Zaharia, Francoise.Preteux}@int-evry.fr

Résumé

Cet article introduit un nouveau schéma de compression de maillages 3D dynamiques avec topologie constante et géométrie variable. La technique proposée exploite un prédicteur affine par morceaux couplé à un modèle de skinning et à une représentation par transformée en cosinus discrète des erreurs résiduelles. Les performances de cette nouvelle approche sont objectivement évaluées sur un corpus de séquences animées de diverses tailles, géométries et topologies et correspondant à des mouvements rigides et élastiques. Les résultats expérimentaux montrent que la méthode proposée offrent des gains importants en termes de débits (de 37% à 67%) par rapport aux approches GV, RT, MPEG-4/AFX, D3DMC, PCA et Dynapack.

Mots clefs

Compression, maillage dynamique, animation 3D.

1 Introduction

Les industries du film d'animation et des jeux vidéo exploitent aujourd'hui à grande échelle les contenus dynamiques 3D.

Ces contenus sont produits en utilisant un large spectre de techniques allant des simulations physiques aux techniques d'acquisition et de clonage de mouvement. Les créateurs des contenus multimédias exploitent le plus souvent les techniques d'animation par modèle de peau (*skinning*) comme celles proposés par le standard MPEG-4/AFX [1] et par les environnements professionnels de modélisation 3D comme 3DS MAX ou Maya.

Pour des raisons de propriété intellectuelle, les créateurs de ces contenus sont souvent réticents à transmettre ces modèles d'animation coûteux et susceptibles de réutilisation illicite et préfèrent une représentation par trames-clés. L'animation est alors stockée sous forme d'une séquence de maillages 3D successifs représentant les trames-clés. Les trames intermédiaires sont obtenues à l'aide de procédures d'interpolation.

La représentation par trames-clés permet de décrire un large spectre d'animations et répond aux problèmes de propriété intellectuelle. De plus, cette représentation est indépendante de la technique d'animation utilisée pour générer le contenu et permet ainsi de disposer d'un format générique d'animation 3D.

L'inconvénient majeur de cette représentation est en revanche lié aux coûts exorbitants de stockage et de transmission. En effet, même pour de courtes séquences de quelques minutes, des milliers de modèles 3D sont nécessaires. L'élaboration de techniques de compression efficaces et adaptées à ces contenus dynamiques devient alors un enjeu majeur comme en témoigne l'important nombre de travaux de la littérature émergente consacrée à ce sujet (voir [2] pour un état de l'art).

Dans [3], Lengyel introduit pour la première fois le concept de compression de maillages dynamiques. L'auteur propose de représenter l'animation à l'aide d'un ensemble de transformées affines et d'erreurs de prédiction associées. Dans [4], les auteurs proposent une extension de [3], appelée RT (*Rigid Transform*). Le mouvement des sommets du maillage est ici modélisé uniquement par des transformées rigides. La technique D3DMC (*Dynamic 3D Mesh Compression*), proposée dans [5], introduit une approche différente. Le champ de mouvement des sommets est représenté par un ensemble de vecteurs de mouvement associés à une structure volumique d'arbre octal (*octree*). Toutes ces approches nécessitent la mise en oeuvre d'une procédure de segmentation qui vise à regrouper les sommets du maillage en parties pouvant être décrites par un unique modèle de mouvement. Cette procédure de segmentation au sens du mouvement est complexe en temps de calcul et peut induire des discontinuités à bas débit au niveau des frontières entre les *patchs*.

Le schéma de compression IC [6] (*Interpolation Compression*) récemment adopté par le standard MPEG-4/AFX [1] exploite une procédure de sous-échantillonnage des trames-clés combinée à une stratégie de prédiction spatio-temporelle locale. L'approche Dynapack [7] propose une approche similaire avec des prédicteurs plus élaborés, appelés *ELP* et *Replica*. Les techniques MPEG-4/AFX-IC et Dynapack présentent l'avantage d'un faible coup de calcul, ce qui les rend particulièrement adaptées aux applications de codage/décodage en temps-réel. Toutefois, le parcours déterministe du maillage utilisé les rend inadaptées pour

des fonctionnalités plus avancées comme la transmission progressive ou le rendu scalable.

Dans [8], Alexa et Müller introduisent une famille différente d'approches, fondée sur une analyse en composantes principales (ACP) du champ de déformation du maillage. Le même principe est repris et étendu dans [9], où un schéma de prédiction supplémentaire est introduit, ainsi que dans [10] où le maillage est préalablement segmenté en parties optimisées pour une représentation par ACP. Les approches de compression par ACP sont spécifiquement adaptées à des séquences longues et répétitives avec un nombre de sommets petit par rapport au nombre de trames. Ces approches sont en revanche très complexes en temps de calcul (cubique avec le nombre de sommets du maillage).

Dans [11], les auteurs proposent un schéma de compression de maillages dynamiques 3D par ondelettes irrégulières construites à l'aide d'une structure de maillage progressif [12]. Le codeur proposé permet d'atteindre des bas débits tout en offrant des fonctionnalités de transmission progressive et de rendu scalable. Cette approche est en revanche inadaptée aux maillages avec un nombre réduit de sommets par composante connexe.

Une représentation différente, dite GV (*Geometry Video*), est proposée dans [13]. Le principe consiste à convertir la géométrie 3D dynamique sous forme d'une séquence d'images 2D. L'approche GV exploite un découpage du maillage (nécessaire pour obtenir une topologie homéomorphe à un disque) et une paramétrisation [14] sur un domaine 2D carré. La topologie initiale du maillage est complètement abandonnée et remplacée par une topologie régulière, obtenue en échantillonnant uniformément le domaine paramétrique. Les images géométriques ainsi construites sont compressées par des techniques traditionnelles de codage d'images 2D. La représentation GV offre des performances de compression compétitives ainsi que des fonctionnalités de transmission progressive et de rendu scalable. En revanche, les objets traités doivent être homéomorphes à un disque ou à une sphère, ce qui restreint sensiblement le domaine d'application de la méthode. De plus, la procédure de remaillage peut conduire à une perte de détails de la surface ainsi qu'à l'apparition d'artéfacts visuels.

Dans cet article, nous proposons un nouveau schéma de compression de maillages 3D dynamiques, fondé sur une modélisation du mouvement par des techniques de *skinning*. La contribution majeure de ce travail concerne l'étape de compensation du mouvement qui est formulée comme un problème inverse. A partir d'une représentation arbitraire par trames-clés, un modèle de *skinning* est tout d'abord dérivé et ensuite exploité dans le cadre d'une stratégie de prédiction. Les erreurs résiduelles résultantes sont compressées à l'aide d'une représentation par Transformée en Cosinus Discrète (TCD).

L'approche de compression par modèle de *skinning* proposée étend l'approche RT en introduisant :

– Une nouvelle technique de segmentation au sens du mouvement avec un nombre fixe de *patchs* qui permet d'éviter une sur-segmentation du maillage dynamique,
– Un modèle de *skinning* permettant d'éviter les discontinuités au niveau des bords des *patchs*,
– Un nouveau bloc de codage par TCD des erreurs résiduelles.

Le paragraphe suivant décrit la méthode de compression proposée en détaillant ses principales étapes. Les performances en terme d'efficacité de compression sont ensuite évaluées objectivement, comparées et discutées (paragraphe 3). Enfin, la dernière partie conclut l'article et ouvre les perspectives de recherche future.

2 Compression par *skinning*

Le schéma synoptique de l'algorithme de compression par modèle de *skinning* est présenté Figure 1.

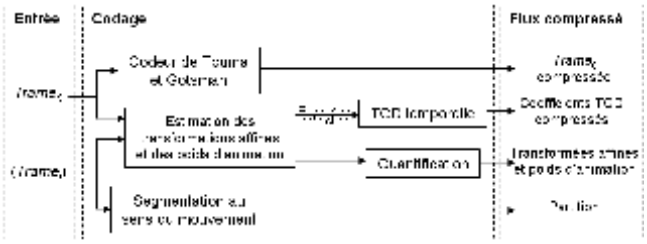


Figure 1 – *Schéma synoptique de l'algorithme de compression par modèle de skinning.*

Tout d'abord, les sommets du maillage sont partitionnés en un ensemble de *patchs* de façon à ce que le mouvement de chaque *patch* puisse être décrit par une unique transformée affine 3D. Un modèle de mouvement affine est alors associé à chaque *patch* et pour chaque trame. Enfin, le mouvement de chaque sommet est exprimé comme une combinaison linéaire pondérée des mouvements des différents *patchs*.

Le mouvement affine est toujours estimé par rapport à la première trame de l'animation afin (1) de permettre un accès aléatoire aux trames de l'animation et (2) de gérer efficacement les pertes d'information dans le contexte de transmission en environnement bruité. La première trame de la séquence est compressée par un codeur statique. Dans ce travail, nous avons utilisé le codeur classique de Touma et Gotsman [15]. Les erreurs résiduelles de compensation de mouvement sont finalement codées à l'aide d'une représentation par TCD temporelle.

Détaillons à présent les différents blocs du schéma de codage proposé.

2.1 Segmentation au sens du mouvement

L'objectif de l'étape de segmentation est d'obtenir une partition $\pi = (\pi_k)_{k\in\{1,\ldots,K\}}$ des sommets du maillage en K parties dont le mouvement peut être fidèlement représenté par des transformées affines 3D. L'approche proposée, détaillée ci-dessous, prend en compte le mouvement de

l'ensemble des sommets du maillage au cours de toute la séquence.
Tout d'abord, une transformée affine A_i^v décrivant le mouvement d'un voisinage local du sommet $v \in \{1, ..., V\}$ (V étant le nombre des sommets) à la trame $i \in \{0, ..., F-1\}$ (F étant le nombre de trames) par rapport à la première trame est calculée comme décrit dans l'équation (1) :

$$A_i^v = \arg\min_A \left(\sum_{v \in v^*} \| A\chi_0^v - \chi_i^v \|^2 \right), \qquad (1)$$

avec A une matrice 4×4 représentant une transformée affine, v^* le voisinage du sommet v d'ordre trois (*i.e.*, sommets connectés à v par un chemin composé au plus de trois arêtes), et χ_i^v un vecteur 4D représentant les coordonnes homogènes du sommet v à la trame i.
L'ensemble $(A_i^v)_{i \in \{0,...,F-1\}}$ est ensuite stocké dans un seul vecteur $\alpha^v \in \mathbb{R}^{12 \times F}$ (une transformée affine étant complètement définie par 12 coefficients réels). Enfin, la partition π est obtenue en appliquant l'algorithme de segmentation *k-means* [16] sur l'ensemble des vecteurs $(\alpha^v)_{v \in \{1,...,V\}}$.
La Figure 2 présente les résultats de segmentation obtenus pour les maillages animés "Dance", "Chicken" et "Snake". Le nombre de parties K a été fixé à 20 dans toutes les expérimentations. Cela permet de prendre en compte les différentes parties articulées des objets sans générer de sur-segmentation. Remarquons que pour les maillages dynamiques articulés, l'algorithme arrive à retrouver les parties anatomiques en mouvement. Dans tous les cas, la procédure conduit à des *patchs* constitués de sommets topologiquement connexes, même si aucune information topologique n'a été directement prise en compte par l'algorithme *k-means*. Intuitivement, ces résultats montrent que la modélisation affine par morceaux, détaillée dans la section suivante, est bien adaptée pour décrire ces mouvements complexes.

(a) "Dance" (b) "Chicken" (c) "Snake"

Figure 2 – *Segmentation de maillages dynamiques avec différents mouvements, formes et complexités.*

2.2 Estimation du mouvement affine

Une fois la partition π déterminée, pour chaque trame i le champ de mouvement est modélisé par K transformées affines, notées $(H_i^k)_{k \in \{1,...,K\}}$. Avec les notations de la section précédente, la transformée affine H_i^k associée au *patch* k à la trame i est définie par :

$$H_i^k = \arg\min_A \left(\sum_{v \in \pi_k} \| A\chi_0^v - \chi_i^v \|^2 \right). \qquad (2)$$

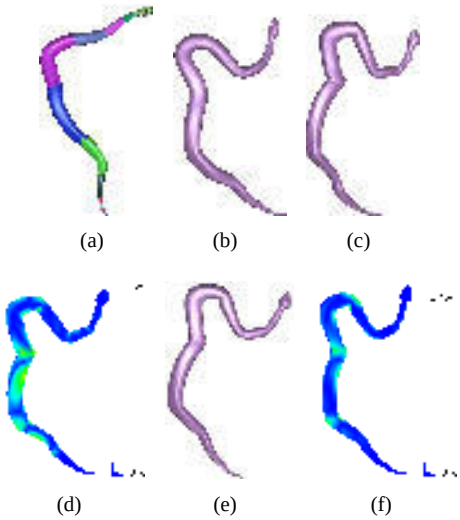
L'ensemble des transformées affines $(H_i^k)_{k \in \{1,...,K\}}$, avec la partition π, fourni un prédicteur affine par morceaux de la trame i à partir de la trame 0 défini comme suit :

$$\forall\, v \in \{1, ..., V\}, \quad \widehat{\chi}_i^v = H_i^{k(v)} \chi_0^v, \qquad (3)$$

où $k(v)$ désigne le *patch* contenant le sommet v.
Les erreurs de prédiction $e_i^v = (e_i^{v,x}, e_i^{v,y}, e_i^{v,z}, 0)^t$ sont définies par :

$$\forall\, v \in \{1, ..., V\}, e_i^v = \chi_i^v - \widehat{\chi}_i^v. \qquad (4)$$

La Figure 3 montre la trame originale 36 de la séquence "Snake", sa version prédite et la distribution des erreurs de prédiction correspondantes, exprimées en pourcentage de la diagonale de la boîte englobante de l'objet et représentées en fausses couleurs.
Nous remarquons que le prédicteur proposé modélise efficacement le mouvement des différentes parties du maillage. Ici, l'erreur de prédiction maximale est de l'ordre de 4% de la diagonale de boîte englobante de l'objet. Notons également que les erreurs les plus importantes sont obtenues au niveau des frontières des *patchs*. Cela montre les limitations du modèle de mouvement affine par morceaux, qui introduit des discontinuités de bords.
Afin de nous affranchir de cette limitation, nous proposons de raffiner le modèle de mouvement en introduisant un modèle de *skinning* similaire à ceux utilisés dans le domaine de l'animation 3D.



(a) (b) (c)

(d) (e) (f)

Figure 3 – *Prédicteur affine par morceaux vs. modèle de skinning : (a) première trame segmentée, (b) trame 36, (c) trame 36 prédite par transformées affines, (d) distribution des erreurs de (c), (e) trame 36 prédite par modèle de skinning et (f) distribution des erreurs de (e).*

2.3 Modèle de *skinning* et estimation des poids d'animation

Le principe de l'animation par modèle de *skinning* est de dériver un champ de mouvement continu sur l'ensemble du maillage en combinant linéairement les mouvements affines des *patchs* avec des poids appropriés. La position prédite $\widehat{\chi}_i^v$ du sommet v à la trame i s'exprime alors comme :

$$\widehat{\chi}_i^v = \sum_{k=1}^{K} \omega_k^v H_i^k \chi_0^v, \quad (5)$$

où ω_k^v est un coefficient réel qui contrôle l'influence du *patch* k sur le mouvement du sommet v.
L'objectif est de déterminer, pour chaque sommet v, le vecteur des poids optimaux $\omega^v = (\omega_k^v)_{k\in\{1,\dots K\}}$ défini par :

$$\omega^v = \arg\min_{b\in\mathbb{R}^K} \sum_{i=0}^{F-1} \left\| \sum_{k=1}^{K} b_k H_i^k \chi_0^v - \chi_i^v \right\|^2. \quad (6)$$

En pratique, il est raisonnable de considérer qu'un *patch* influence uniquement ses propres sommets et ceux appartenant à un *patch* voisin. Soient $k(v)$ le *patch* contenant le sommet v et $\theta(v)$ l'ensemble des *patchs* incluant $k(v)$ et ses voisins. L'équation (6) est résolue sous les contraintes suivantes :

$$\forall k \notin \theta(v), \quad b_k = 0. \quad (7)$$

L'équation (6) avec les contraintes (7) conduit à un problème de minimisation au sens des moindres carrés qui peut être résolu par une méthode de pseudo-inverse [17].
Les Figures 3.e et 3.f présentent les résultats obtenus pour la séquence "Snake" en utilisant le modèle de *skinning*. Cette fois, le maillage obtenu par compensation de mouvement est lisse. De plus, les erreurs maximale et moyenne ont été réduites de 35% et 31%, respectivement.
Les erreurs résiduelles $(e_i^{v,x}, e_i^{v,y}, e_i^{v,z})$ sont enfin compressées en utilisant un schéma de codage par TCD temporelle, comme décrit dans le paragraphe suivant.

2.4 Compression par TCD

Pour chaque sommet v du maillage dynamique, nous considérons les erreurs de prédiction associées $(e_i^{v,x})_i$, $(e_i^{v,y})_i$, et $(e_i^{v,z})_i$ ($i \in \{1, ..., F\}$) comme trois séquences temporelles. Les spectres $(s_i^{v,x})_i$, $(s_i^{v,y})_i$, et $(s_i^{v,z})_i$ de chaque séquence sont alors déterminés en appliquant une TCD monodimensionnelle temporelle.
Cette procédure est appliquée à l'ensemble des sommets du maillage. Les coefficients spectraux obtenus sont enfin multiplexés dans un seul vecteur S de dimension $3\times V\times F$, défini comme suit :

$$S = \coprod_{i\in\{1,\dots,F\}} \coprod_{v\in\{0,\dots,V-1\}} (s_i^{v,x}, s_i^{v,y}, s_i^{v,z})^t, \quad (8)$$

où $\coprod$ désigne l'opérateur de concaténation.

Le vecteur S ainsi obtenu est finalement codé à l'aide du codeur arithmétique proposé dans [18].
Notons enfin que le processus de concaténation décrit par l'équation (8) rend possible la transmission progressive des coefficients de le TCD à partir des basses fréquences, et ainsi la reconstruction d'une version approchée de l'animation à toute étape du processus de décodage.

3 Résultats expérimentaux

3.1 Corpus de test et critères d'évaluation

Afin de réaliser des comparaisons objectives, nous avons considéré un corpus de test de 4 séquences animées, utilisées par la majorité des travaux de la littérature et appelées "Dance", "Chicken", "Humanoid" et "Snake". Le Tableau 1 résume leurs propriétés exprimées en termes de nombre de sommets (V), de trames (F) et de composantes connexes (CC).

Séquence d'animation	V	F	CC
"Dance"	7061	201	1
"Humanoid"	7646	154	1
"Chicken"	3030	400	41
"Snake"	9179	134	1

Tableau 1 – *Propriétés des séquences animées considérées pour l'évaluation.*

Les taux de compression sont exprimés en bits par sommet par trame (bpst). Les distorsions de compression sont mesurées en utilisant l'erreur RMSE [19] entre maillages initiaux et reconstruits (décodés). L'erreur RMSE entre deux séquences d'animation est définie comme étant la moyenne des erreurs RMSE sur l'ensemble des trames.

3.2 Résultats de compression

Dans nos expérimentations, les coefficients réels décrivant les transformations affines ainsi que les poids d'animation ont été quantifiés sur 16 bits. La première trame de l'animation a été compressée en utilisant le codeur de Touma et Gotsman [15] avec une quantification sur 12 bits pour la géométrie. Les erreurs de prédiction ont été quantifiées sur 7 bits.
Les résultats obtenus par les approches D3DMC, AFX-IC, RT, PCA, Dynapack et GV sont ceux rapportés dans [5], [4], [7] et [13].
La Figure 4 présente les résultats du codage par modèle de *skinning* et l'approche GV pour la séquence "Dance". La technique de compression par *skinning* montre des gains en débits pouvant atteindre 65% (avec débits de 8 et 2.8 bpst resp. à une RMSE de 0.00025). Les faibles performances de l'approche GV s'expliquent par les distorsions de paramétrisation importantes et le sur-échantillonnage mis en oeuvre par ce codeur.
La Figure 5 présente une comparaison des performances de l'approche par modèle de *skinning* et celles des codeurs AFX-IC et D3DMC, pour la séquence "Humanoid". Notre

technique offre des gains en taux de compression par rapport à D3DMC jusqu'à 67% (avec débits de 3 et 1 bpst resp. à une RMSE de 0.0006). Les performances de l'approche AFX-IC sont de loin les plus faibles. Cela est sans surprise, compte tenu de la simplicité du schéma de prédiction locale utilisé, trop élémentaire pour prendre en compte les corrélations spatio-temporelles.

La Figure 6 compare les performances du codeur par *skinning* aux approches D3DMC, AFX-IC, RT, et Dynapack pour la séquence "Chicken". Le codeur D3DMC et la technique de compression par modèle de *skinning* surclassent les autres techniques, permettant d'atteindre de très bas débits (moins de 4 bpst). Par rapport à D3DMC, le codage par modèle de *skinning* conduit à des gains en débit jusqu'à 37% (avec débits de 3.8 et 2.4 bpst resp. à une RMSE de 0.0013). Cela s'explique par la non-optimalité de la décomposition par structure d'arbre octal considérée par D3DMC.

Enfin, la Figure 7 illustre les performances des approches de codage par *skinning*, par RT et par ACP pour la séquence "Snake". Ici, le codeur par ACP présente les performances les plus faibles. Cela est dû au fait que l'approche par ACP est optimisée pour des animations avec un nombre de trames beaucoup plus important que le nombre de sommets ($F \gg V$), hypothèse qui n'est manifestement pas vérifiée par la séquence "Snake" ($F = 134$, $V = 9179$). Par rapport à la technique RT, notre approche permet d'obtenir des gains en débits jusqu'à 45% (avec débits de 5.3 et 2.9 bpst resp. à une RMSE de 0.00035). Cela prouve que la modélisation du mouvement par un modèle de *skinning* combinée à une TCD temporelle est plus efficace que celle par transformées rigides adoptée par RT.

Ces résultats montrent que l'approche de codage par modèle de *skinning* est particulièrement efficace dans le cas des maillages dynamiques articulés, où la stratégie de compensation de mouvement proposée est bien adaptée. De plus, notre approche permet d'obtenir des séquences reconstruites de haute qualité à partir de très bas débits (1 bpst).

4 Conclusion et perspectives

Dans cet article, nous avons présenté une nouvelle technique de compression de maillages 3D dynamiques, avec topologie fixe et géométrie variable. Le coeur de notre méthode est un modèle de *skinning*, inspiré des techniques de création de contenus 3D, qui permet la mise en oeuvre d'un schéma de prédiction hybride avec compensation de mouvement et représentation en TCD des erreurs résiduelles de prédiction. Les différentes étapes intervenant dans la construction automatique de ce modèle (segmentation du maillage au sens du mouvement, modélisation de mouvement affine par morceaux et estimation optimale des poids d'animation) sont présentées en détails.

Les expérimentations, conduites sur une base de séquences de test traditionnellement utilisées dans la littérature pour l'évaluation des algorithmes de compression, montrent que l'approche proposée offre des gains en débit considérables (de 37% à 67%) par rapport aux autres techniques de l'état de l'art (GV, D3DMC, RT, Dynapack et MPEG-4/AFX-IC). Le schéma de codage par modèle de *skinning* se révèle particulièrement efficace dans le cas des séquences d'animation présentant des personnages au mouvement articulé.

Les perspectives de recherche concernent principalement l'optimisation de l'étape de segmentation au sens du mouvement. En particulier, nous étudierons comment déterminer un nombre optimal de parties, sous contraintes aussi bien anatomiques (*e.g.* prise en compte d'un squelette d'animation) que géométriques (*e.g.* erreur de prédiction maximale).

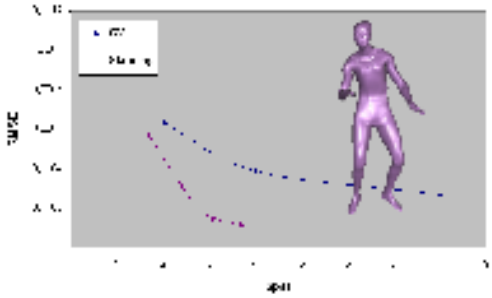


Figure 4 – *Modèle de skinning vs. GV pour la séquence "Dance".*

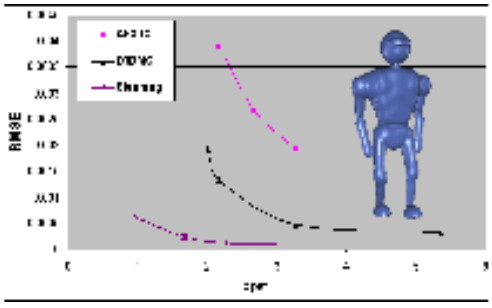


Figure 5 – *Modèle de skinning vs. AFX-IC et D3DMC pour la séquence "Humanoid".*

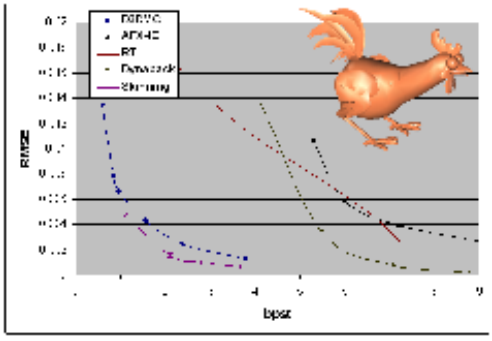


Figure 6 – *Modèle de skinning vs. AFX-IC, D3DMC, RT et Dynapack pour la séquence "Chicken".*

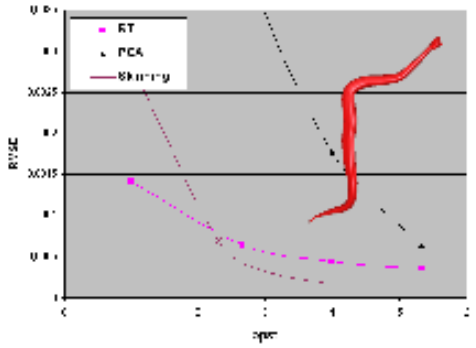


Figure 7 – *Modèle de skinning vs. RT et PCA pour la séquence "Snake".*

Références

[1] M. Bourges-Sevenier et E.S. Jang. An introduction to the MPEG-4 animation framework extension. *IEEE transactions on Circuits and Systems for Video Technology*, 14(7) :928–936, Juillet 2004.

[2] K. Mamou, T. Zaharia, et F. Preteux. A preliminary evaluation of 3D mesh animation coding techniques. Dans *Actes de la conférence SPIE Conference on Mathematical Methods in Pattern and Image Analysis*, pages 44–55, San Diego, États-Unis, Avril 2005.

[3] J. Lengyel. Compression of time-dependent geometry. Dans *Actes de la conférence ACM Symposium on Interactive 3D Graphics*, pages 89–96, Atlanta, États-Unis, Avril 1999.

[4] G. Collins et A. Hilton. A rigid transform basis for animation compression and level of detail. Dans *Actes de la conférence Vision, Video, and Graphics*, pages 21–28, Édimbourg, Royaume-Uni, Juillet 2005.

[5] K. Müller, A. Smolic, M. Kautzner, P. Eisert, et T. Wiegand. Predictive compression of dynamic 3d meshes. Dans *Actes de la conférence IEEE International Conference on Image Processing*, pages 621–624, Genève, Suisse, Septembre 2005.

[6] E. S. Jang, J. D. K. Kim, S. Y. Jung, M. J. Han, S. O. Woo, et S. J. Lee. Interpolator data compression for MPEG-4 animation. *Circuits and Systems for Video Technology*, 14(7) :989–1008, Juillet 2004.

[7] L. Ibarria et J. Rossignac. Dynapack : space-time compression of the 3D animations of triangle meshes with fixed connectivity. Dans *Actes de la conférence SIGGRAPH/Eurographics symposium on Computer animation*, pages 126–133, San Diego, États-Unis, Juillet 2003.

[8] M. Alexa et W. Müller. Representing animations by principal components. *Computer Graphic Forum*, 3(19) :411–418, Août 2000.

[9] Z. Karni et C. Gotsman. Compression of soft-body animation sequences. *Computers and Graphics*, 28(1) :25–34, 2004.

[10] M. Sattler, R. Sarlette, et R. Klein. Simple and efficient compression of animation sequences. Dans *Actes de la conférence SIGGRAPH/Eurographics symposium on Computer animation*, pages 209–217, Los Angeles, États-Unis, Juillet 2005.

[11] I. Guskov et A. Khodakovsky. Wavelet compression of parametrically coherent mesh sequences. Dans *Actes de la conférence ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 183–192, Grenoble, France, Août 2004.

[12] H. Hoppe. Progressive meshes. *Computer Graphics*, 30(Annual Conference Series) :99–108, Juin 1996.

[13] H. M. Briceño, P. V. Sander, L. McMillan, S. Gortler, et H. Hoppe. Geometry videos : a new representation for 3D animations. Dans *Actes de la conférence Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 136–146, San Diego, États-Unis, Juillet 2003.

[14] P. Sander, P. Gortler, J. Snyder, et H. Hoppe. Signal-specialized parameterization. Dans *Actes de la conférence Eurographics workshop on Rendering*, pages 87–98, Pisa, Italie, Juin 2002.

[15] C. Touma et C. Gotsman. Triangle mesh compression. Dans *Actes de la conférence Graphics Interface*, pages 26–34, Vancouver, Canada, Juin 1998.

[16] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, et A. Y. Wu. An efficient k-means clustering algorithm : Analysis and implementation. *Pattern Analysis and Machine Intelligence*, 24(7) :881–892, Juillet 2002.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, et B. P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge Press, New York, États-Unis, 2^{nd} édition, 1992.

[18] I. H. Witten, R. M. Neal, et J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6) :520–540, Juin 1997.

[19] N. Aspert, D. Santa-Cruz, et T. Ebrahimi. MESH : Measuring errors between surfaces using the hausdorff distance. Dans *Actes de la conférence IEEE International Conference in Multimedia and Expo (ICME)*, pages 705–708, Lausanne, Suisse, Août 2002.

Indexation et Recherche de Vidéo à Travers leur Script

Emna Fendri *, Narjes Hajjem, Hanene Ben-Abdallah **
Laboratoire Miracl
Institut Supérieur des Etudes Technologiques de Sfax;Faculté des Sciences Economiques et de Gestion de Sfax*
{Fendri.msf,hanene}@gnet.tn

Concours jeune chercheur : Oui

RÉSUMÉ

Dans cet article, nous présentons un système d'indexation et de recherche des documents vidéo à travers leur script. Ce système se base sur les concepts et étapes d'indexation des documents textuels pour générer une base d'index au document vidéo à partir du script. Notre approche d'indexation se base sur l'adaptation des techniques d'indexation empiriques en vue de déterminer des index reflétant le contenu sémantique particulier aux documents vidéo.

Mots clefs

Indexation textuelle, indexation vidéo, alignement, script XML, pondération

1 Introduction

Face à l'abondance des documents audiovisuels, les utilisateurs sont en quête de techniques et outils efficaces pour l'indexation et la recherche de documents textuels et, de plus en plus, des images et séquences vidéo.
Plusieurs techniques d'indexation et de recherche de documents textuels (c.f., [1], [2]) ont atteint un haut niveau de maturité et d'efficacité. Cependant, la nature opaque des images et de la vidéo limite l'efficacité des méthodes, jusque là proposées, pour l'indexation et la recherche basées sur le contenu de la vidéo. En effet, ces méthodes se heurtent à une rupture entre les résultats des analyses bas niveau du flux vidéo (e.g., les descripteurs MPEG7[1]) et les interprétations sémantiques de haut niveau. La maturité des techniques textuelles nous a incité à explorer la possibilité de les réutiliser pour la vidéo. La réutilisation de ces techniques (d'indexation/ recherche/ extraction) peut être réalisée selon deux approches : appliquer les techniques sur le flux vidéo, ou bien les appliquer sur le script de la vidéo. La deuxième approche de réutilisation parait plus simple et efficace. Elle exploite, d'une part, le document textuel base de la production de toute vidéo (appelé script) et, d'autre part, les techniques existantes pour l'analyse bas niveau de la vidéo.
Dans cet article, nous proposons une approche de réutilisation des techniques d'indexation textuelle pour indexer la vidéo à travers son script. En effet, cette proposition de réutilisation directe des techniques d'indexation textuelle pour la vidéo repose sur le fait que toute vidéo est produite sur la base d'un document textuel structuré, appelé *script*. Ce dernier, décrit avec détails le contenu de la vidéo permettant ainsi de fournir des informations reflétant l'aspect narratif (les dialogues et les événements) et productif (scènes, séquences, etc.) de la vidéo. Comme exploité dans le système de recherche vidéo à base de script (SRV) [3], grâce à des analyses bas niveau du flux vidéo, un script peut être augmenté par des points d'entrée à sa vidéo. Ceci permet d'indexer/ rechercher indirectement un document vidéo. Ainsi notre approche de réutilisation directe des techniques textuelles à travers les scripts vidéo offre une couverture sémantique de la vidéo plus riche puisqu'elle couvre simultanément les aspects audio, visuel et textuel (descriptif) de la vidéo.
Dans Section 2, nous présenterons une étude sur les différentes méthodes d'indexation pour les documents textuels et les vidéos. Section 3 est consacrée pour la présentation de notre approche globale utilisée pour l'indexation des documents vidéo à travers leur script. Dans Section 4, nous détaillerons les méthodes utilisées pour la structuration du script et la détermination des termes d'index. Avant de conclure, dans Section 5, nous présenterons des résultats expérimentaux de la méthode proposée.

2 Etat de l'art

2.1 Indexation d'un Document Textuel

Dans la littérature, les diverses méthodes d'indexation sont basées sur soit des traitements linguistiques qui utilisent une analyse rhétorique (c.f. [8]) ou une analyse syntaxique (c.f. [4]) minimale; soit des calculs statistiques (c.f.,[6]) basés sur une analyse statistique de la fréquence d'apparition de certains mots et/ou de la distribution de certains termes sans effectuer une analyse linguistique préalable; ou encore des traitements combinés (linguistique et statistique) (c.f. [6]).
En outre, ces méthodes diffèrent dans leur modèle de représentation des documents à indexer. Les modèles les plus utilisés sont : à plat, pondéré, à rôles ou à facettes, et structuré. Indépendamment du modèle utilisé, les différentes méthodes d'indexation proposées utilisent une étape préliminaire de classification du document à indexer selon son volume, sa structure ou son domaine de références. Les types de classification dépendent des objectifs finaux de l'indexation (recherche, résumé, etc.) et du niveau de précision à atteindre. Suite à la classification du document, les méthodes procèdent aux étapes de segmentation, étiquetage, lemmatisation et élimination de mots vides, en vu de produire un

"ensemble" de termes représentant/indexant le document. Selon l'objectif de l'indexation, un terme représentatif peut être 1) des mots choisis en fonction d'un score calculé à base d'une méthode de pondération appropriée (c.f. [1]) ; 2) des phrases contenant les mots les plus fréquents pour représenter la thématique du texte ; ou encore, 3) des paragraphes à chacun est associé un vecteur dont les coordonnées sont le nombre d'occurrences de chaque mot du texte retenu après prétraitement.

2.2 Indexation d'un document vidéo

L'indexation d'un document vidéo consiste à extraire puis à structurer toutes les informations disponibles dans ce document [17]. Tout comme les documents textuels, deux structurations linéaire et relationnelle sont essentielles à une représentation complète du contenu d'un document vidéo. Toutes les informations issues de l'indexation du document sont greffées sur cette double structure.

La structuration linéaire d'un document vidéo fait apparaître une structure hiérarchique, capable de représenter les composants du document et d'atteindre un niveau sémantique élevé. Dans cette structure, les plans constituent des unités fondamentales. Suite à un macro découpage du document vidéo (découpage en une suite de plans), un sous découpage des plans en morceaux plus petits, ayant une certaine cohérence syntaxique ou sémantique, et une extraction d'images représentatives de leur contenu, constitue le micro découpage temporel du document vidéo. Les images représentatives issues du découpage temporel du document vidéo seront par la suite découpées en régions possédant un contenu sémantique propre, et formant ainsi une structuration linéaire spatiale du document.

La structuration relationnelle d'un document vidéo met en évidence des relations pouvant exister entre des entités précédemment extraites et qui ne sont pas forcément voisines ni de même type pour former un "graphe de relations". La mise en relation s'effectue naturellement par extraction de points communs entre deux entités données (similarité entre images clés de deux prises de vue différentes, persistance d'incrustations et de bandeaux, persistance de présentateurs, détection d'un fond immobile,…).

D'autre part, pour avoir un taux de pertinence satisfaisant suite à une requête utilisateurs, des traitements sémantiques des documents audiovisuels s'avèrent nécessaires. Ces traitements donnent naissance à de nouvelles représentations des informations audiovisuelles encodées non sous la forme de valeurs de pixels, mais selon un format d'objets associés à des mesures physiques et des informations temporelles appelées *structures symboliques*. Plusieurs travaux sont menés dans l'objectif de répondre aux besoins d'une création d'une structure symbolique relative à un flux vidéo. Parmi ces travaux, se classent les activités de standardisation MPEG-7(c.f. [10], [11]), Dublin Core ainsi que le projet IVR [11] qui a donné naissance à la nouvelle structure symbolique de documents vidéo. Cette structure symbolique montre l'existence des entités "classe" qui décrivent des éléments significatifs comme des personnages, des éléments de décors, des objets ou encore des objets de granularité plus fine représentant des parties d'objets. Chaque apparition d'une "classe" dans les images consécutives est appelée une entité d'occurrence. Chaque "occurrence" est une suite de zones caractérisées chacune par des entités de forme, de couleur, de position et de mouvement de caméra.

2.2.1 Étapes d'indexation vidéo

L'indexation des documents vidéo par le contenu est une opération de traitement par laquelle nous choisissons les unités les plus appropriées pour représenter le contenu d'un document (par exemple personnage, lieu, incrustation, etc.). Selon l'objectif visé, différents niveaux de précision sont retenus suite à une étape de segmentation du flux vidéo en unités structurelles, d'extraction d'images représentatives d'un plan et de décomposition d'une image clé en une liste d'objets ou de blocs informationnels.

a- La segmentation en unités structurelles

Deux niveaux de segmentation sont possibles : un macro découpage de la vidéo en une suite de plan et un micro découpage d'un plan en une suite d'images. Pour chaque niveau de découpage, diverses techniques peuvent être utilisées. En se basant sur le fait qu'un plan représente une unité atomique en terme de montage, un macro découpage de la vidéo en une suite de plan permet de déterminer une structuration linéaire d'un document vidéo. Dans un document vidéo, deux plans successifs sont séparés par des transitions. De ce fait, pour extraire un plan à partir d'un document vidéo il suffit d'avoir les moyens de détections de transitions encadrant ce plan. Dans la littérature, quatre groupes de transitions sont mis en évidence (c.f. [9]) : Les coupures, les groupes 2, les groupes 3, les fondus (i.e., fondu enchaîné, fondu noir, fondu blanc, etc.). La détection de chaque type de transition utilise des techniques différentes.

b- L'extraction d'images clés

Cette étape de traitement consiste à extraire pour chaque plan des images représentatives de leur contenu informationnel. Dans un document vidéo, deux cas sont possibles : soit l'information sémantique la plus intéressante est placée en dehors des transitions. Soit que, ces transitions signifiant l'apparition de nouvelle information et c'est justement à ces moments là où juste après, qu'il convient de sélectionner les images clés. De même, le choix d'images clés dépend fortement du type de contenu du flux vidéo traité : par exemple une seule image clé suffit pour représenter un plan de présentateur de journal télévisé, alors que plusieurs seront nécessaires dans le cas d'un plan contenant plusieurs objets en mouvement. De nombreuses techniques sont utilisées pour

l'extraction des images représentatives d'un plan. Ces techniques sont classées en trois catégories :
– Les techniques à base d'un choix arbitraire d'images clés (c.f., [12])
– Les techniques basées sur des critères de mouvement, de couleur, de présence de visage, etc.
– Les techniques basées sur une étude des bords entrants et sortants dans les images
c- L'extraction d'objets représentatifs
Vu le volume très important d'es informations qui peuvent être présentées au sein d'une image, une extraction des objets représentatifs d'une image (en particulier les images clés) permet d'enrichir la description structurelle d'un document vidéo. L'extraction des objets représentatifs dans une image commence par une segmentation de l'image en régions homogènes suivie par un repérage des zones représentatives du contenu de l'image.

2.2.2 Méthodes d'indexation vidéo

Diverses méthodes opèrent dans le domaine compressé. Elles sont généralement classées en quatre catégories suivant la nature des indices qu'elles manipulent : Les méthodes utilisant les coefficients DCT (Discrete Cosine Transform) (c.f.,[13]) ; les méthodes utilisant les vecteurs de mouvement (c.f., [14]) ; les méthodes utilisant les coefficients DCT et les vecteurs de mouvement ; et les méthodes de décomposition en sous bande (c.f., [13]).
Plusieurs autres techniques opèrent dans le domaine non compressé, par exemple en utilisant les histogrammes (couleur ou luminance x2) (c.f., [13]), les formes, les mouvements (c.f. [13]), etc.

2.2.3 Méthodes d'indexation de la vidéo à travers son script

L'idée d'indexation de la vidéo à travers son script a été exploitée pour développer le Système de Recherche de Vidéo SRV [3]. Dans ce système, l'indexation utilise la technique implémentée par le moteur Niagara [28] qui se base sur la spécification :
– d'une DTD définissant la grammaire des scripts,
– d'une liste d'éléments relatifs aux divers balises dans les documents du corpus. Pour chaque élément, Niagara associe l'identificateur du document contenant cet élément, la position début et la position fin de cet élément.
– d'une liste de mots clés retenus après l'élimination des mots vides. A chaque mot clé dans cette liste, Niagara associe l'identificateur du document contenant le mot et la position du mot dans ce document.
Selon le processus d'indexation de Niagara, un mot est défini comme mot vide s'il appartient à une liste de mots vides, décrite à priori dans son code source. Les mots clés sont retenus avec leur forme fléchie sans prise en considération de la notion de forme canonique ni de relation sémantique entre mots. Un score fréquence mot est calculé pour chaque mot clé retenu. Ce score est pris comme critère de classification des résultats suite à un processus de recherche.

3 Notre Approche d'indexation de la vidéo

Notre approche d'indexation des scripts alignés à la vidéo (voir Figure 1) utilise deux niveaux d'indexation complémentaires : une indexation *sémantique structurée locale* (l'étape de raffinement dans la figure 1) et une indexation *statistique globale* (les étapes d'extraction, détermination de lemme/synonyme, pondération).

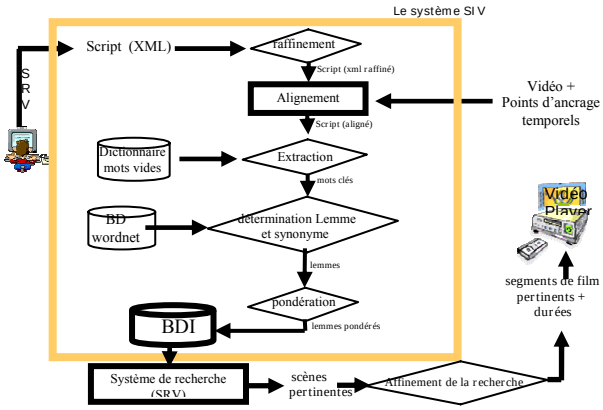


Figure 1 - *Approche d'indexation script aligné dans SIV*

Cette approche a été implémentée dans l'outil Système d'Indexation de Vidéo (SIV) qui est associé au Système de Recherche de Vidéo (SRV) [3] qui nous a facilité l'évaluation expérimentale de notre approche.
Comme illustré dans la figure 1, notre approche accepte un script en tant qu'un document XML (automatiquement formaté par SRV). Après l'étape de raffinement du script, ce dernier est augmenté par des points d'entrée à la vidéo permettant de l'aligner manuellement au flux vidéo segmenté. Ainsi, outre la réutilisation des techniques d'indexation textuelle, notre approche profite aussi des analyses bas niveau de la vidéo qui existent déjà dans la littérature afin d'extraire des points d'encrage entre la vidéo et son script. Par exemple, nous citons les travaux du groupe LIP6 qui a proposé des méthodes d'analyse audio permettant d'aligner les dialogues d'un script avec les phrases parlées dans le flux vidéo. En outre, les travaux de Ronfard et Thuong [11] qui se basent sur la détection des transitions et des sous titres pour aligner un script aux plans et aux segments de dialogues dans le flux vidéo. De même, les travaux de Mahdi et al [3] qui présentent des méthodes de détection de scènes filmées à l'intérieur, l'extérieur durant le jour ou la nuit, permettant ainsi d'aligner un script aux scènes vidéo.
Une fois aligné, un script aligné subit une indexation selon une méthode statistique qui génère des index pondérés. Ces derniers sont utilisés pour créer une base indexée de scripts, qui peut être interrogée par le système SRV. Ce

système produit une liste de scènes pertinentes. Cette dernière est alors explorée par un module assurant un affinement de la recherche pour obtenir les portions de vidéo les plus pertinents répondant à la requête utilisateur.

3.1 Structure des scripts

Un script structuré est un document décomposé en des éléments qui reflètent l'aspect narratif (exemple dialogue, action,…) et productif (scène, plan,…) de la vidéo. Une étude statistique de scripts réalisée dans le projet SRV [3], a pu dégager la structure générale de script vidéo sous format de la DTD de la figure 2.

```
<!ELEMENT script (Titre, Auteur*, Scenariste?, Producteur?, Directeur?, Ouvrage_base?, Annee?, Cast?,Introduction?,Sequence)>
<!ELEMENT Titre (#PCDATA)>
<!ELEMENT Auteur (#PCDATA)>
<!ELEMENT Scenariste (#PCDATA)>
<!ELEMENT Producteur (#PCDATA)>
<!ELEMENT Directeur (#PCDATA)>
<!ELEMENT Ouvrage_base (Oeuvre?, Ecrivain*)>
<!ELEMENT Oeuvre (#PCDATA)>
<!ELEMENT Ecrivain (#PCDATA)>
<!ELEMENT Annee (#PCDATA)>
<!ELEMENT Cast (Nomreel, Nomrole?)*>
<!ELEMENT Nomreel (#PCDATA)>
<!ELEMENT Nomrole (#PCDATA)>
<!ELEMENT Introduction (#PCDATA)>
<!ELEMENT Acteur (Nom?, Desc_acteur?, Dialogue?, Description?)>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Intext (#PCDATA)>
<!ELEMENT Lieu (#PCDATA)>
<!ELEMENT Moment (#PCDATA)>
<!ELEMENT Duree (#PCDATA)>
<!ELEMENT Desc_dansscene ( Desc_scene?, Acteur*) >
<!ELEMENT Desc_scene (#PCDATA)>
<!ELEMENT Desc_acteur (#PCDATA)>
<!ELEMENT Dialogue (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
```

Figure 2- *structure d'un script vidéo (DTD) aligné [3]*

La DTD de la figure 2 est formée principalement par des éléments. L'élément racine est appelé « script ». Ce dernier est décomposé en plusieurs éléments (par exemple Titre, Auteur, Scénariste, etc.) dont certains forment une donnée XML valide (PCDATA : Parseable Character DATA), alors que d'autres sont formés par plusieurs sous éléments dont chacun est formé de zéro, un ou plusieurs sous éléments. Cette structure à été augmentée par le système SRV [3], par l'ajout d'un élément « Durée » dans le but d'aligner le script à sa vidéo au niveau des scènes. Dans la suite du papier, un script est dit *aligné* lorsqu'il contient des points d'entrée au flux vidéo. Chaque point d'entrée est un intervalle de temps qui représente les instants de début et de fin de la partie vidéo correspondant à la partie du script annotée avec l'intervalle. Grâce à l'alignement du script à sa vidéo, une telle décomposition permet d'avoir une description structurelle physique (séquence, scène, plan,…) et symbolique (événement action, relation,…) de chaque vidéo traitée.

3.2 Raffinement

Après une étude empirique de 116 scripts structurés selon la DTD de la Figure 2, nous avons remarqué que cinq éléments des scripts structurés selon la méthode de SRV restent relativement grands : introduction, description scène, description acteur, dialogue et description. Ces éléments peuvent engendrer un taux de précision faible et un taux de bruit élevé. Cependant, plus l'élément indexé est fin, plus la recherche est performante ; par exemple, une indexation de l'élément *description acteur* comme unité élémentaire offre à l'utilisateur des informations noyées aux milieux d'autres sujets. Ainsi, une décomposition de cet élément en sous éléments cohérents (body, action, relation sociale,…) permet d'enrichir l'index et par conséquence, d'avoir des résultats plus précis dans un processus de recherche par exemple.

D'autre part, notre étude empirique a aussi souligné la présence de deux types d'éléments dans un script XML :

– des *méta-balises* dont le contenu représente des informations complémentaires sur la vidéo ou de nature descriptive; par exemple : le genre, l'auteur, le producteur, etc. Ces informations sont vues comme des entités élémentaires et sont donc prises directement comme des index ;

– des *non méta-balises* dont le contenu représente de grandes quantités d'information sur le contenu de la vidéo. Ces éléments doivent être segmentés, classifiés selon des cas sémantiques, et représentés par des termes d'index. Ces traitements sont réalisés par les étapes de lemmatisation et de pondération dans la Figure 1.

L'étape de raffinement vise à homogénéiser les cinq éléments du script, qui ont été identifiés comme porteurs de diverses informations. Le raffinement d'un script est essentiellement une indexation locale au niveau des cinq éléments à homogénéiser. Comme indiqué dans Figure 3, notre approche de raffinement utilise quatre étapes :

– Extraction des éléments à raffiner dans la DTD de la Figure 2

– Segmentation du contenu de chaque élément à raffiner en phrases : Afin de décomposer encore chaque élément retrouvé dans l'étape précédente il est nécessaire de les segmenter en phrases.

– Analyse de chaque phrase retrouvée dans un élément afin de lui attribuer une nouvelle balise.

– Restructuration du script XML en insérant les nouvelles balises.

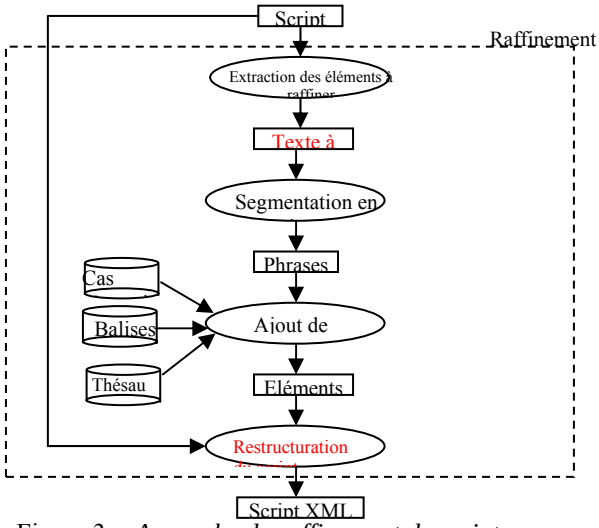


Figure 3 - *Approche de raffinement de script*

L'extraction des éléments à raffiner est réalisée par une analyse syntaxique du script pour trouver les balises marquant le début et la fin de chacun d'eux. L'attribution des balises à chaque phrase retrouvée est essentiellement de l'indexation (ou classification) de phrases. A la fin de cette étape chaque macro balise sera représentée par un ensemble d'unités élémentaires et cohérentes permettant ainsi d'avoir une structuration plus fine du script.

Notre méthode de classification de phrases se base, d'une part, sur une analyse sémantique des mots et, d'autre part, sur une étude empirique qui permet de dégager les cas sémantiques, chacun représenté par une (nouvelle) balise. Expérimentalement, à travers une étude des 116 scripts, nous avons déterminé un ordre de priorité des cas sémantiques par macro-balise. Cet ordre est utilisé lors de l'affectation d'une balise pour chaque phrase.

Selon cette étude, une phrase peut être classée dans un cas sémantique parmi dix cas possibles dérivés en combinant les cas lexicaux offerts par la base de données lexicales Wordnet [15]. Il s'agit de "Evènement action", "Objet animal", "Objet", "Corps", "Evènement naturel", "lieu", "Objet personne", "Relation temporelle", Relation sociale" et "relation spatiale". Nous avons trouvé que le cas sémantique à attribuer à une phrase dépend de l'élément (i.e., macro balise) du script où elle se trouve. Par exemple, si une phrase dans l'élément «description acteur» contient des mots représentant un événement action, un corps et une relation spatiale, alors la phrase est classée dans le cas «événement action» ; cependant cette même phrase sera classée dans le cas «relation spatiale» si elle se trouve dans l'élément introduction.

Une fois qu'un macro-élément a été raffiné (subdivisé), le script initial est réécrit afin de remplacer le contenu de chaque macro-élément par sa version raffinée. Notons que, dans cette étape, afin d'éliminer des index redondants, deux phrases successives ayant une même balise sont regroupées sous une seule balise. A la fin de cette étape, nous obtenons un script finement structuré et pouvant être indexé selon notre démarche illustrée dans la figure 1.

3.3 Détermination des termes d'index

La troisième étape de l'indexation d'un script raffiné et aligné est la détermination des termes d'index (voir Figure 1). L'extraction est restreinte au contenu des nœuds feuilles de la DTD. Notre approche réutilise les méthodes classiques de détermination des mots clés dans les documents textuels. En général, ces méthodes commencent par une élimination des mots vides (à base d'un dictionnaire mots vides), et utilisent des calculs de fréquence pour déterminer des mots clés. Notre approche diverge de ces méthodes classiques en prenant tous les mots qui restent comme mots clés, vue la granularité fine de l'élément analysé.

A fin d'optimiser cette étape, elle est précédée par l'application d'une étape de détermination des synonymes et lemmatisation. Notre approche vise à finaliser la liste des termes déjà retenus par un remplacement d'un groupe de mots ayant un lien sémantique par un seul mot nommé mot index. La fréquence de chaque mot en relation avec un mot index sera considérée lors du calcul de la fréquence du mot index. En outre, la lemmatisation consiste à prendre les termes avec leur forme canonique. Ceci étant toujours dans l'objectif de réduire le volume des tables d'index.

4 Pondération et structuration d'index

Notons que, les techniques linguistiques utilisent des règles d'analyse qui dépendent de la langue du document. Tandis que les techniques statistiques jouissent d'une indépendance de la langue. Cet avantage nous a incité à adopter une approche statistique et donc à base de pondération comme critère d'indexation.

Les techniques de pondération sont aujourd'hui les plus dominantes dans le domaine textuel (cf. [6], [8], [7], [16]). Elles consistent à représenter chaque terme déjà retenu par un score représentant l'importance du terme d'indexation associée à cette dimension dans le document (c.f., [7]).

4.1 Score terme

Comme souligné par Salton [7] et Sauvagnat, l'importance d'un terme se traduit par une valeur de score qui dépend de la fréquence du terme dans le document, de sa position et éventuellement d'informations globales (comme sa fréquence dans le corpus). Tenant compte de cette dépendance, nous avons fixé le score d'un terme comme fonction de sa fréquence dans le document et dans le corpus. La fréquence d'un terme dans le script est calculée en tenant compte du nombre de synonymes et de formes fléchis mis en relation avec le terme. Tandis que la fréquence d'un terme dans le corpus est la somme des fréquences de ce terme dans chaque script du corpus.

Pour déterminer le score position d'un terme, nous avons mené une étude empirique sur les scripts qui a montré que le score position varie selon l'emplacement du terme dans le script et le genre cinématographique de la vidéo. Par exemple, pour un script de journal télévisé, les termes présents dans les *titres* sont plus importants que ceux présents dans la description des acteurs. Suite à notre étude statistique, nous avons fixé le score position d'un terme comme la moyenne des scores des éléments contenant ce terme. De sa part, le score d'un élément du script dépend du genre de la vidéo.

4. 2 Score élément

Dans le domaine structuré, les techniques les plus populaires accordent à chaque élément des niveaux d'importance différents selon sa position hiérarchique dans le document. Différentes méthodes sont possibles : soit selon la profondeur dans la structure hiérarchique, soit selon la profondeur et la position séquentielle dans un niveau de la structure hiérarchique [16]. Un premier tableau décrit le niveau d'importance affecté à chaque macro-balise dans le script. Ces niveaux ont été accordés suite à une étude du corpus et avec prise en considération du genre cinématographique de la vidéo.

Tout comme les macro-balises, un niveau d'importance est affecté à chaque micro-balise (événement action, personne,…) en fonction de son importance dans le corpus et de sa position dans le document. Une technique probabiliste est utilisée afin d'accorder des valeurs d'importance à ces micro-balises. Sur la base d'une étude du corpus, les taux de probabilité présentés un deuxième tableau, ont été retenus pour chaque micro-balise. Notons qu'une phrase peut décrire une ou plusieurs cas sémantiques (micro- balises) des taux de probabilités dont la somme est supérieur à 1 sont donc retrouvés.

Le produit scalaire des valeurs retenues dans le premier tableau et celles dans le second tableau permet de donner comme résultat les scores relatifs à chaque micro balise selon sa position dans le document.

5 Evaluation expérimentale

Une étude expérimentale a été menée pour évaluer notre approche sur quatre étapes à savoir : le raffinement, la détermination des mots clés, la recherche à travers le système SRV et l'attribution des scores.

– Pour le module de raffinement, nous avons obtenu un taux de pertinence de 70% sur 870 phrases provenant de divers scripts du corpus.

– Concernant les mots clés, notre étude expérimentale effectuée sur un corpus de 53 scripts a donné un taux de rappel de 88 % et un taux de précision de 91% pour la détermination des mots clés et un taux de pertinence de 74% pour l'accord des scores des différents mots clés.

– Enfin, pour une évaluation globale du système SIV, l'application de 31 requêtes sur un deuxième corpus de 116 scripts recherchés via le système SRV (après intégration du système SIV) a donné les résultats présentés dans Tableau1.

Films entiers	Rappel 86%	Précision 90%
Segments de films	Rappel 95%	Précision100%

Tableau 1 : Taux de rappel et précision fournis par SRV après intégration de SIV.

6 Conclusion

Dans cet article, nous avons passé en revu brièvement un état de l'art sur l'indexation des documents textuels et de la vidéo, puis nous avons présenté une nouvelle approche pour l'indexation et la recherche des documents vidéo à travers leur script. Cette méthode réutilise les concepts d'indexation textuelle appliqués au script du document vidéo. Une première évaluation expérimentale, après intégration de notre système d'indexation au système de recherche SRV, a donné des taux de pertinence et de rappel encourageant. Ces taux devraient être validés à travers une évaluation plus étendue. D'autre part, nous sommes en train d'investiguer l'automatisation de l'alignement du script au flux vidéo tout en bénéficiant des travaux existant pour la segmentation physique de la vidéo.

Références

[1] http://www.dsi-info.ca/mot-cle.html

[2] http://www.movie-page.com/movie_scripts.htm

[3] W. Magrebi, *système de recherche vidéo.* Mémoire de DEA SINT FSEG de Sfax Tunisie. Mars, 2003.

[4] http://www-poleia.lip6.fr/~slodzian/sberland/Chapitre1.html

[6] N. Masson, *Méthodes pour une génération variable de résumé automatique : vers un système de réduction de texte*. Thèse de doctorat Paris XI, Orsay LIMSI 1998.

[7] J.T. Minel, *Filtrage sémantique du résumé automatique a la fouille de textes*.

[8] M. M. Amini, *Apprentissage automatique et recherche de l'information : application à l'extraction d'information de surface et au résumé de texte*. Thèse de doctorat de l'Université Paris XI. 13 Juillet, 2001.

[9] http://www.irit.fr/ASSTICIM/irit.html

[10] F. Prêteux, E*njeux et technologies des standard MPEG-4 et MPEG-7*. médianet 2002.

[11] http://opera.inrialpes.fr/people/Tien.Tran-Thuong/DEAThese99/RapportDEAOrg2506.html

[12] L. Chen, D. Fontaine, et R. Hammoud. *La segmentation sémantique de la vidéo basée sur les indices spatio-temporels*. CORESA, pages 67-75, Lannion, Juin 1998.

[13] M.Ardebilian Fard, *une contribution à l'indexation par le contenu de la vidéo.* doctorat de l'Université de Technologie de Compiegne 2001.

[14] F.Salazar, *analyse automatique des mouvements de caméra dans un document vidéo*. rapport IRIT/95-33-R. Septembre, 1993.

[15] http://www.lip6.fr/Laboratoire/Rapport1998/Projets.pdf

[16] *Actes de la première Conférence en Recherche d'Information et Applications (CORIA'04) 10–12 mars 2004.*

[17] C.G.M. Snoek and M. Worring. Multimodel vide indexing: A review of the state of the art. ISIS Technical Report Series, 2001(20), 2001.

Connexions entre descripteurs locaux et globaux pour la reconnaissance d'objets dans les vidéos

Bruno Lameyre Valérie Gouet-Brunet
CEDRIC/CNAM - 292, rue Saint-Martin - F75141 Paris Cedex 03
{bruno.lameyre,valerie.gouet}@cnam.fr

Concours Jeune Chercheur : Oui

Résumé

Dans ce travail, nous présentons une approche de reconnaissance d'objets génériques à partir de flux vidéos, basée sur la construction d'un catalogue de caractéristiques visuelles hétérogènes. Notre première contribution porte sur la description de l'apparence visuelle des objets, en proposant l'utilisation conjointe de primitives génériques et complémentaires de différentes natures : d'un côté, un ensemble de descripteurs locaux, aux propriétés bien connues, telle leur robustesse à l'arrière-plan ; de l'autre côté, un contour actif comme descripteur global, fournissant une description haut-niveau de la forme de l'objet. Notre seconde contribution propose de structurer efficacement ces descripteurs, notamment en établissant des connexions entre eux. Cette approche est comparée à une approche classique et évaluée sur plusieurs séquences contenant 20 objets. Nous montrons sa pertinence pour l'annotation automatique de contenus vidéo, où de bons taux de reconnaissance sont atteints, tout en préservant des performances compatibles avec le temps-réel.

Mots clefs

Indexation d'images par contenu visuel, Reconnaissance d'objets, Descripteurs locaux, Contours actifs, Flux vidéo.

1 Introduction

La reconnaissance d'objets est depuis longtemps un domaine de recherche actif. La plupart des approches rencontrées base la phase d'apprentissage sur un ensemble d'images fixes. Depuis peu, un petit nombre d'approches propose d'exploiter la richesse de la vidéo, sur la base des observations suivantes : les humains reconnaissent mieux un objet quand il est en mouvement plutôt qu'à partir de simples vues ; techniquement, une vidéo fournit de multiples vues de l'objet, facilement reliables par une méthode de suivi entre trames consécutives. Ces approches exploitent l'information temporelle des vidéos en extrayant une ou plusieurs primitives visuelles dans chaque trame et en le(s) suivant le long de la séquence. Suivre une primitive permet notamment de produire une description plus robuste en modélisant sa variabilité le long de la trajectoire et en structurant l'espace de description engendré pour regrouper les caractéristiques redondantes par objet et entre objets. Dans [1] par exemple, des modèles 3D sont construits en exploitant le suivi de patches à partir d'objets en mouvement dans les vidéos. Dans [2], une approche probabiliste de suivi et de reconnaissance est proposée pour la reconnaissance de visages à partir de vidéos.

Lorsque l'on considère la reconnaissance d'objets *génériques* dans les vidéos, les approches rencontrées impliquent comme primitives des *descripteurs locaux* basés sur l'extraction de points d'intérêt, voir par exemple [3, 4, 5, 6]. Ces primitives sont suivies le long de la séquence, de manière à exhiber les plus robustes en sélectionnant celles survivant sur plusieurs trames et en modélisant leur variabilité le long des trajectoires [5]. A noter que ces descripteurs ont été aussi été utilisés pour la reconnaissance d'objets spécifiques, comme les visages, après une étape de détection [7].

En parallèle, avec les récentes propositions de nouveaux descripteurs locaux impliquant différentes natures de support (incluant les patches de texture, les régions homogènes, les formes locales et les points de symétrie), certains travaux [3, 6, 8] ont proposé d'améliorer la description de l'objet pour la reconnaissance dans les images fixes et les vidéos, en exploitant la combinaison de différents descripteurs locaux complémentaires.

Avec le même objectif, d'autres approches ont proposé d'associer un *contexte* plus global aux descripteurs locaux. Dans [9], le vecteur SIFT décrivant chaque point est renforcé par une information de forme dans un voisinage plus large. Dans [10], un contexte (exprimé par des corrélogrammes) est ajouté au descripteur local, intégrant une description des relations spatiales entre le point et ses voisins.

Toutes les approches génériques de description venant d'être décrites sont locales, et tirent ainsi partie des propriétés bien connues de ces descripteurs, comme leur robustesse aux transformations de l'image, aux occultations et aux arrière-plans. Malheureusement, par définition ces primitives ne peuvent pas fournir une description *globale* de l'apparence visuelle de l'objet, pourtant si informative.

D'un autre côté, une description plus globale, comme par exemple la forme ou les couleurs dominantes de l'objet, pourrait être grandement informative pour la reconnaissance. Cela est d'ailleurs démontré dans [11], où la reconnaissance d'organismes marins est améliorée par l'utilisation conjointe de primitives locales et globales. Mais en règle générale, quand les objets sont mêlés à un arrière-

plan, les primitives globales requièrent une étape préliminaire de segmentation de l'image ou de détection de l'objet. Ces traitements sont incompatibles avec la reconnaissance d'objets génériques à partir d'images ordinaires. En effet, si une segmentation peut être facilement réalisée avec des images spécifiques, comme dans [11], elle reste une étape délicate nécessitant des connaissances a priori sur l'image. La détection d'objets requière quant à elle un modèle de l'objet à détecter, par exemple un détecteur de visages [7].

Principe de notre approche. A partir de ces observations, nous proposons une approche de reconnaissance d'objets génériques à partir de primitives visuelles *hétérogènes*. L'objectif est de combiner le potentiel des descripteurs locaux, principalement *leur robustesse* aux occultations et à l'arrière-plan, à celui de descripteurs plus globaux *très informatifs*, sans avoir à considérer un pré-traitement de segmentation de l'image ni de détection de l'objet.
Notre première contribution porte sur la construction *indépendante* de deux espaces de description : l'un associé à la description locale par points d'intérêt, l'autre dédié à la description globale des objets. Dans les expériences menées, nous avons choisi un contour actif comme descripteur global, qui décrit bien la forme de l'objet. Comme énoncé plus haut, les descripteurs locaux permettent la reconnaissance d'objets quel que soit l'arrière-plan, alors que les contours actifs ne sont pas directement utilisables dans un tel contexte. Dans notre approche, les descripteurs locaux sont vus comme la *source primaire* dans une première étape de reconnaissance. Les points appariés obtenus sont vus comme des *ancres* et donnent la possibilité d'aller plus loin dans la reconnaissance. En effet, elles permettent d'*indexer* un ou plusieurs contours actifs, qui viendront alors confirmer ou infirmer la reconnaissance. Pour permettre cela, notre seconde contribution définit des *connexions* entre descripteurs locaux et globaux.
Nous n'avons pas choisi d'enrichir la description locale en ajoutant des composantes aux vecteurs de points, comme dans les approches définissant un contexte de points [9, 10], conduisant à des espaces de grande dimension (188 dans [9] et 960 dimensions dans [10]). Construire séparément les espaces de description et établir des connexions entre eux permet non seulement de garder des dimensions modérées (20 dimensions dans notre cas) restant compatibles avec des applications temps réel, mais également de préserver les avantages de chaque type de descripteur.
L'article est organisé comme suit : dans la section 2, nous présentons les primitives visuelles choisies pour décrire l'apparence des objets. Les descripteurs obtenus sont stockés et structurés dans un catalogue, dont le processus de construction est décrit dans la section 3. Enfin, nous évaluons notre approche de reconnaissance d'objets et démontrons sa pertinence dans la section 4, avant de conclure.

2 Description visuelle des objets

Nous donnons ici les techniques que nous avons employées pour extraire, décrire et suivre les structures locales et globales utilisées comme descripteurs visuels génériques. Ces approches sont classiques, la contribution principale de cet article consiste à structurer efficacement l'espace des descripteurs obtenus et à les utiliser conjointement pour améliorer la reconnaissance.

2.1 Description locale par points intérêt

Les points d'intérêt sont très populaires en vision par ordinateur comme en indexation d'images. Beaucoup d'approches ont été proposées, comme le montre l'étude comparative [12]. Appliquées à la vidéo, il existe aussi des approches temporelles, citons en particulier [13]. Nous avons choisi d'extraire les points d'intérêt trame par trame, puis de les suivre le long de la séquence. L'algorithme d'extraction et de poursuite que nous avons utilisé est similaire à l'algorithme KLT [14].
Pour la reconnaissance, nous caractérisons les points avec les 20 premiers coefficients d'une DCT. Dans le reste de cet article, l'espace des descripteurs locaux sera noté V_n^{point}, ou n est la dimension ($n = 20$). Notons que d'autres descripteurs locaux pourraient être utilisés sans changer le concept de notre approche, en particulier le descripteur SIFT [15], reconnu pour ses performances [12].

Figure 1 – *Exemples de descripteurs locaux et globaux associés à une tête en mouvement.*

2.2 Description globale avec un contour actif

Caractériser l'apparence d'un objet avec des descripteurs de haut niveau peut être effectué avec plusieurs primitives telles que : formes locales, régions, contours actifs, etc. Pour évaluer notre prototype, nous avons choisi de décrire la forme globale de l'objet avec un *contour actif*. Plusieurs raisons nous ont conduit à faire ce choix :
– Un contour actif décrit bien la forme globale de l'objet, et en fournit ainsi une description visuelle très informative (la forme seule de l'objet permet souvent de le reconnaître). Puisque calculés localement, les points d'intérêt ne contiennent pas d'information globale et ne caractérisent pas les mêmes zones de l'objet.
– Un contour actif est assez facile à suivre dans une séquence vidéo ;
– Un contour actif peut aider durant la poursuite des points d'intérêt et vice versa [16].
La théorie des contours actifs fut introduite dans [17], un état de l'art peut être trouvé dans [18]. L'implémentation discrète des contours actifs que nous avons choisie est classique : trois forces sont appliquées à chaque point de contrôle (élongation, courbure et une force externe déduite des contours de l'image), sans connaissance a priori sur

l'objet. L'espace des descripteurs associé est basé sur les descripteurs de Fourier [19]. Dans la suite de cet article, il sera noté V_m^{snake}, où m est sa dimension (ici $m = 20$). A la figure 1, on peut voir un exemple de la caractérisation obtenue pour un objet particulier. Il illustre la complémentarité et la richesse de ces deux descripteurs, puisque la simple vue de ces descripteurs suffit à reconnaître la nature de l'objet sans ambiguïté.

3 Structuration des descripteurs

Nous proposons ici de construire et de structurer un catalogue de caractéristiques visuelles hétérogènes, à partir des catégories de descripteurs venant d'être introduites.

3.1 Construction du catalogue

La figure 2 illustre la structure globale du catalogue. Tous les descripteurs visuels sont collectés lors de la poursuite des objets dans une séquence d'entraînement (voir [A]).

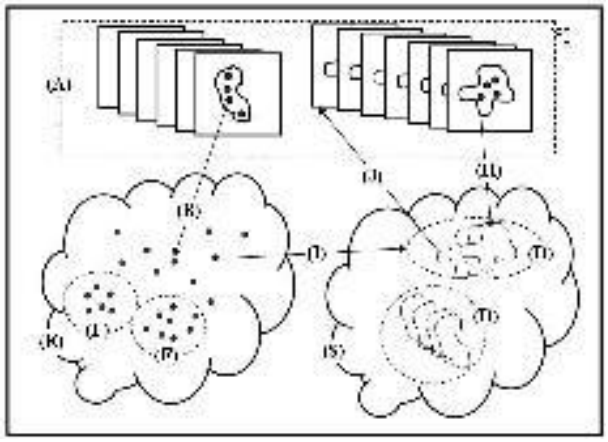


Figure 2 – *Structure globale du catalogue.*

Structuration de l'espace des descripteurs locaux. Pour chaque objet, l'ensemble des descripteurs locaux collectés est inséré [B] dans l'espace V_n^{point}. Chaque point garde le lien avec l'objet associé. Les éléments similaires contenus dans cette espace [E] sont ensuite agglomérés afin de fournir des clusters représentant un vocabulaire visuel (noté " Elementary Local Patterns " ou ELPs) [F]. La construction de vocabulaire visuel à partir de descripteurs locaux a déjà été utilisée dans plusieurs travaux relatifs à la reconnaissance d'objets issus de vidéo, par exemple [5, 6, 7]. Les approches utilisent classiquement un algorithme de regroupement de type k-means, qui fixe le nombre de classes à obtenir et part d'initialisations aléatoires. Dans nos expérimentations, nous avons préféré une approche non supervisée (Competitive Agglomeration - CA) où le nombre de classes est automatiquement déterminé durant le déroulement de l'algorithme, qui est initialisé avec des clusters de points issus de la même trajectoire.

Structuration de l'espace des descripteurs globaux. De façon similaire, tous les descripteurs associés aux contours actifs extraits des objets sont collectés [H] dans l'espace de description V_m^{snake} [S]. Chaque descripteur présent dans V_m^{snake} garde le numéro de trame d'où il est extrait [J]. Tous ces descripteurs de forme sont également soumis à l'algorithme CA [D]. Cette agglomération génère plusieurs clusters que nous appellerons *"Elementary Global Shapes"* ou EGSs.
La structuration des espaces de description V_n^{point} et V_m^{snake} a deux principaux avantages : elle permet une réduction de la redondance spatiale et temporelle des descripteurs, fournissant ainsi des descripteurs plus compacts. Cette compacité a pour conséquence de réduire efficacement le temps de recherche dans ces espaces. Le second avantage est de permettre l'enrichissement dynamique de la description d'objets lors de la reconnaissance d'objets à partir de nouvelles séquences, avec un accroissement minimal du catalogue.

Connexion entre descripteurs locaux et globaux. La dernière étape consiste à *connecter* les espaces de description V_n^{point} et V_m^{snake}. Chaque ELP contient un ensemble de descripteurs locaux similaires et chaque chacun d'entre eux est lié à une trame dans laquelle un contour actif a été extrait et suivi. Ce contour actif est également lié à l'EGS auquel il appartient. Par conséquent, on peut définir une connexion logique entre chaque point de V_n^{point} et un EGS. Les connexions établies entre descripteurs locaux et descripteurs globaux représentent la principale contribution de ce travail : ils sont l'unique façon d'exploiter des descripteurs globaux sans avoir recours à une phase préliminaire de segmentation ou de détection d'objet. Grâce à ces connexions, les descripteurs locaux font fonction d'*index* permettant de déterminer et d'exploiter les descripteurs globaux appropriés.

3.2 Reconnaissance à partir d'une trame

Soit I la trame en cours d'analyse où un objet doit être recherché. La reconnaissance est effectuée en trois étapes :

Étape 1 : Recherche des points candidats. Soit $\{P_1$ ... $P_k\}$ l'ensemble des descripteurs locaux extraits de l'image I. La première étape consiste à rechercher les plus proches voisins de ces points dans l'espace de description V_n^{point}. Pour chaque P_i, les ELPs les plus similaires sont recherchés dans V_n^{point} dans une sphère S_{ϵ,P_i} de rayon ϵ centrée en P_i. Pour tous les P_i considérés, si le nombre de sphères S_{ϵ,P_i} qui intersectent un ou plusieurs ELPs est supérieur à un seuil nommé $T_{anchors}$, alors on suppose que I contient *potentiellement* l'objet (que l'on nomme alors objet candidat). $T_{anchors}$ permet de ne pas détecter un objet systématiquement ; sa valeur est discutée à la section 4.2. Les P_i qui sont appariés dans le catalogue sont nommés M_i ($i \leq k$). D'autres alternatives de classification équivalentes existent pour l'appariement de groupes de points, comme par exemple les classifieurs SVM dédiés aux descripteurs locaux [20, 21]. Nous ne les avons pas utilisées car, à notre connaissance, il n'existe pas de manière efficace de déterminer les M_i à l'issue de la reconnaissance.

Les ELPs du catalogue qui sont appariés avec des M_i sont vues comme des *ancres*. Elle autorisent en effet l'initialisation d'une analyse plus approfondie, car plus globale, des objets candidats, en exploitant les descripteurs globaux EGSs et surtout les connexions établies entre ELPs et EGSs.

Étape 2 : Recherche des formes candidates. Ici, le but est de rechercher dans V_m^{snake} la meilleur forme candidate correspondant à l'objet candidat : l'EGS ayant le plus de connexions avec les ancres est considéré comme le meilleur candidat de forme (on pourrait bien sûr considérer plusieurs candidats de forme). On note SV_{best} le prototype (médoïd) associé au meilleur EGS.

Étape 3 : Validation de la forme candidate. Lors de la dernière étape, il faut vérifier si SV_{best} correspond à l'objet présent dans l'image testée I. Il est nécessaire d'estimer la transformation $\mathcal{T}$ qui existe entre la forme réelle de l'objet dans I et SV_{best}. Puisque SV_{best} est lié à la trame F d'où il vient, nous pouvons apparier l'ensemble des points d'intérêt initialement détectés dans F avec M_i. $\mathcal{T}$ est estimée de ces appariements et permet de placer la forme $SV_I = \mathcal{T}(SV_{best})$ dans I, qui, idéalement, devrait entourer l'objet s'il est présent. Dans le catalogue, chaque point de contrôle pc_i de SV_{best} est décrit par la direction de son gradient $\vec{\nabla} pc_i$. Afin de confirmer si SV_I a une réalité dans I, nous cherchons, dans le voisinage de chacun de ses points de contrôle, si un pixel de l'image possède une direction de gradient proche de $\mathcal{T}(\vec{\nabla} pc_i)$. Si de tels pixels existent pour plus d'un tiers des points de contrôle de SV_I, alors SV_I est déclaré valide, l'objet est déclaré présent dans I et sa localisation est donnée très précisément par SV_I. Le seuil du tiers (noté T_{snake} dans la suite de cet article) a été choisi en fonction de la proportion des occultations autorisées durant la reconnaissance (un exemple de reconnaissance en présence de fortes occultations est montré à la figure 5).
A la reconnaissance de l'objet est associé un taux de confiance $CR(I, O_j)$, où O_j est l'objet reconnu. Ce taux dépend du nombre de points appariés, des distances d'appariement et du nombre de points de contrôle du contour actif qui possèdent un gradient similaire.

4 Evaluation de l'approche

L'approche proposée a été évaluée sur 20 objets aux différentes apparences visuelles en termes de contenu et de forme (des jouets, des visages, des boîtes, etc). Les séquences d'entraînement (au format 352×288 pixels) contiennent chacune 400 trames filmant une rotation complète de l'objet avec un fond uniforme. Nous avons évalué l'approche sur 8000 trames avec les mêmes objets, à une échelle similaire (les descripteurs locaux utilisés n'étant invariants qu'à de faibles changements d'échelle), mais sous des points de vue différents, avec un arrière-plan chargé et une caméra mobile. Dans un premier temps, la reconnaissance est effectuée trame par trame et pour chaque trame, les points intérêt sont extraits de la totalité de l'image. Notre approche est comparée à une approche de référence qui consiste à exploiter seulement les descripteurs locaux pendant la reconnaissance. Les différents résultats de reconnaissance sont présentés et comparés aux sections 4.1, 4.2 et 4.3. Une courbe ROC moyenne y est calculée par objet ; le paramètre de cette courbe est le seuil de détection ϵ de la recherche des plus proches voisins dans V_n^{point} (section 3.2). Puis, dans la section 4.4, nous évaluons notre approche pour un scénario "video-to-video" où la reconnaissance est faite à partir de plusieurs trames. Finalement, la section 4.5 donne une idée des temps de calcul associés.

4.1 Reconnaissance à partir des points seuls

Dans cette évaluation, seuls les descripteurs locaux sont utilisés. Soit $\{P_1 \ ... \ P_k\}$ l'ensemble des descripteurs locaux extraits de la trame I où l'objet est cherché. Les plus proches voisins de chaque P_i sont recherchés dans V_n^{point} à l'intérieur d'une sphère ϵ centrée autour de P_i. Chaque point trouvé vote pour l'objet auquel il est associé. O_j est déclaré présent s'il est associé au vote le plus fort. Remarquons que, comme dans la première phase de la section 3.2 avec $T_{anchors}$, nous imposons un seuil minimal T_j, qui est fonction du nombre moyen N_j de points d'intérêt extraits de toutes les vues O_j de la séquence d'entraînement. Dans cette expérience, $T_j = N_j/3$.
La figure 3 montre plusieurs des 20 courbes ROC obtenues (lignes fines), ainsi que la courbe ROC représentant la moyenne sur les 20 objets (courbe en pointillés épais).

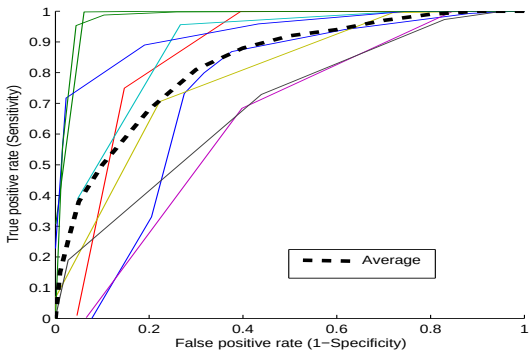


Figure 3 – *Reconnaissance avec les points seuls.*

Le taux d'erreur (*ROC equal error rate*) obtenu est de 74%[1]. Ce résultat a été obtenu en appariant chaque point d'intérêt indépendamment, de sorte qu'il pourrait être amélioré en ajoutant une étape de recalage afin de limiter les mauvais appariements (algorithme Ransac ou Hough par exemple). Ici, il constitue seulement la méthode référence pour évaluer notre approche.

4.2 Apport du descripteur global

La même évaluation a été effectuée en utilisant l'approche complète. Dans la figure 4, la courbe fine en pointillés est celle de la figure 3, présentée ici comme référence et affichée à une échelle adaptée. Les courbes ROC (en fin) illustrent la reconnaissance de plusieurs des objets, alors que

[1] Taux de faux positifs = taux de faux négatifs = 26%.

la courbe plus épaisse représente la courbe ROC moyenne obtenue avec les 20 objets.

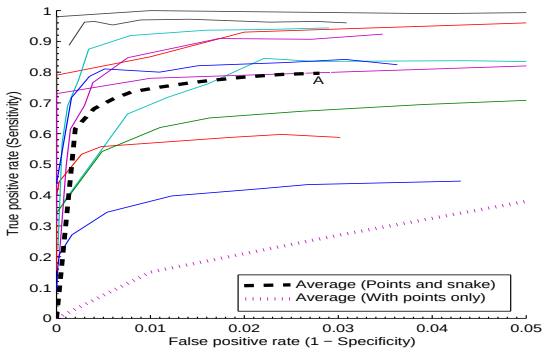


Figure 4 – *Reconnaissance avec les descripteurs globaux.*

Les résultats obtenus sont bien meilleurs lorsque les descripteurs globaux sont utilisés. Par exemple, avec un taux de faux positifs de 2.85 % (point A sur la figure), le taux de faux négatifs est divisé par 3.72 (il est de 74.4% avec l'approche de base et de 20 % avec l'approche complète). Plusieurs raisons expliquent cette amélioration :

Détection automatique et suppression des mauvais appariements. Les points extraits dans l'image et appariés à des vecteurs de V_n^{point} qui ne pointent pas vers l'EGS majoritaire sont déclarés illicites et sont automatiquement supprimés. Ce contrôle efficace et peu coûteux permet de supprimer de nombreux faux appariements.

Détection et suppression des mauvais candidats de forme. Les contours actifs permettent de vérifier la validité de l'objet supposé présent, comme expliqué à la section 3.2. Cette seconde étape de reconnaissance permet de supprimer beaucoup de fausses alarmes, inévitable lorsque les descripteurs locaux sont utilisés seuls.

Ajustement des seuils. Dans les séquences d'entraînement, les diverses vues d'un même objet ne contiennent pas, en moyenne, le même nombre de points. Dans la version basique (sans contour actif), le seuil T_j, représentant le nombre minimum d'ancres requises, est statique pour un objet donné O_j. Dans la version complète, ce seuil (noté $T_{anchors}$ dans la première partie de la section 3.2) peut être ajusté de façon plus fine pour chacune des vues de l'objet. Pour la vue v, $T_{anchors}$ correspond à une fraction α de la moyenne du nombre de points d'intérêt situés dans l'objet 0_j pour l'ensemble des trames qui ont participé à l'EGS associée à v. Cette adaptation contribue à améliorer les résultats de reconnaissance. Dans notre expérimentation, nous avons choisi $\alpha = 1/8$.

La figure 5 illustre un résultat de reconnaissance en présence de fortes occultations. L'importance des occultations tolérées dépend de la valeur des seuils $T_{anchors}$ et T_{snake}. Ici, les résultats ont été obtenus en utilisant les valeurs des seuils données dans l'article.

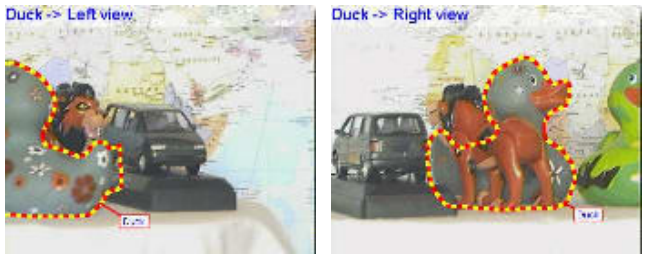


Figure 5 – *Robustesse à de fortes occultations.*

4.3 Choix d'un point de fonctionnement

Pour l'annotation de flux vidéos, il est préférable de choisir un point de fonctionnement (ici ϵ) qui permette de réduire les faux positifs, et donc le nombre d'annotations erronées. Le nombre de vrais positifs s'en retrouvera certainement lui aussi diminué, mais, sous l'hypothèse réaliste qu'un objet est présent sur plusieurs trames d'une séquence, la probabilité de rater cet objet (et donc de ne pas l'annoter) reste faible. En conséquence, le point de fonctionnement que nous avons choisi correspond au cas où le coût des faux positifs est 200 fois plus important que celui des faux négatifs. Les fausses détections correspondantes, déduites des figures 3 et 4, sont résumées dans la table 1.

Approche	Faux Négatifs	Faux Positifs
Points seuls	92%	0.46%
Points + contour actif	38%	0.19%

Table 1 – *Fausses détections pour un point de fonctionnement de rapport 1/200.*

Les résultats obtenus avec l'approche complète démontrent que, lorsque l'objet est présent, il est détecté 15.5 fois par seconde en moyenne (9.5 détections sont manquées chaque seconde, pour un flux vidéo à 25 fps). Cette fréquence de détection (15.5 Hz) permet, de façon quasi certaine, de détecter l'objet s'il passe dans la séquence. Remarquons, qu'avec ce faible taux de faux positifs, nous avons tout de même une fausse détection toutes les 21 secondes en moyenne. Dans la section suivante, nous allons voir qu'en intégrant l'information temporelle, on contribue aussi à réduire le nombre de fausses détections isolées.

4.4 Reconnaissance sur plusieurs trames

Des approches récentes exploitant les flux vidéo proposent d'intégrer les réponses des classifieurs sur plusieurs trames consécutives, comme par exemple [22] qui définit un contexte temporel probabiliste. Pour le moment, nous avons choisi une voie plus simple qui consiste à pondérer les taux de confiance $CR(I_k, O_j)$ (section 3.2) obtenus lorsque l'objet O_j est reconnu dans la trame I_k par les taux de confiance obtenus dans la fenêtre temporelle $[I_{k-w}..I_{k-1}]$ de taille w (dans nos expérimentations, $w = 5$). En intégrant cette information, nous avons amélioré les résultats de la section précédente, voir la table 2.

Scénario	Faux Négatifs	Faux Positifs
mono-trame	38%	0.19%
multi-trames	33.62%	0.069%

Table 2 – *Contribution de l'intégration temporelle avec l'approche complète. Le taux de faux négatifs a été divisé par 1.13 et le taux de faux positifs par 2.74.*

4.5 Temps de calcul

Notre prototype n'est pas entièrement optimisé, mais nous pensons que le temps réel peut être atteint. En particulier, pour le moment aucune structure d'index n'est utilisée pour accélérer la recherche dans V_n^{point}. Les informations suivantes donnent une idée des performances actuelles basées sur un Intel P4 avec une fréquence de 3.2GHz : pour une résolution vidéo de 352×288, avec une moyenne de 100 points d'intérêt extraits dans chaque trame et un catalogue contenant 20 objets (soit environ 240.000 points d'intérêt), le système analyse entre 2 et 3 trames par seconde.

5 Conclusions et perspectives

La principale contribution de ce travail est l'utilisation conjointe de descripteurs locaux et globaux pour la reconnaissance d'objets génériques. Les connexions établies entre eux permettent d'utiliser conjointement des descripteurs locaux *robustes* et un descripteur global *informatif*. Nous avons montré que ces connexions apportent une amélioration significative du processus de reconnaissance dans des vidéos. L'approche ne requiert aucune étape initiale de segmentation d'image ni de détection d'objet afin d'isoler l'objet, la rendant robuste à des arrière-plans quelconques. La mise en place d'espaces de description *indépendants* permet non seulement d'envisager l'annotation temps-réel, de part les dimensions modérées des espaces engendrés, mais aussi de préserver les atouts de chaque technique de description. Finalement, en plus de la richesse de sa description, le choix d'un contour actif comme descripteur global fournit une localisation précise de l'objet retrouvé dans l'image, comme l'illustre la figure 5.

Un pas vers l'annotation fine des objets dans les vidéos. La structure du catalogue autorise *plusieurs niveaux d'annotations* : en plus des labels sémantiques associés aux objets (ici leurs noms), il est en effet possible d'en affecter aux clusters de forme EGSs. Le système a donc la capacité de reconnaître l'objet mais aussi de donner, découlant de la forme reconnue, une idée de sa pose 3D et donc de son comportement dans la séquence. Cette idée est illustrée à la figure 5, avec les annotations "left", "right", "back" et "face" attribuées aux 4 EGSs obtenus pour cet objet.
Nous allons maintenant nous consacrer à l'enrichissement dynamique du catalogue : une fois reconnu, l'objet sera poursuivi dans la séquence, fournissant ainsi de nouveaux descripteurs à ajouter au catalogue. Ces mises à jour permettront de prendre en compte de nouvelles vues de l'objet, rendant la reconnaissance de plus en plus robuste.

Références

[1] F. Rothganger, S. Lazebnik, C. Schmid, et J. Ponce. Segmenting, modeling and matching video clips containing multiple moving objects. Dans *ICCV*, 2004.

[2] S. Zhou, V. Krueger, et R. Chellappa. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, 91 :214–245, 2003.

[3] J. Sivic et A. Zisserman. Video Google : A text retrieval approach to object matching in videos. Dans *ICCV*, 2003.

[4] J. Sivic et A. Zisserman. Video data mining using configurations of viewpoint invariant regions. Dans *IEEE CVPR*, Washington, DC, 2004.

[5] M. Grabner et H. Bischof. Extracting object representation from local feature trajectories. Dans *1st Cognitive Vision Workshop*, 2005.

[6] A. Opelt, J. Sivic, et A. Pinz. Generic object recognition from video data. Dans *1st Cognitive Vision Workshop*, 2005.

[7] J. Sivic, M. Everingham, et A. Zisserman. Person spotting : video shot retrieval for face sets. Dans *CIVR*, 2005.

[8] F. Jurie et C. Schmid. Scale-invariant shape features for recognition of object catgories. Dans *IEEE CVPR*, 2004.

[9] E.N. Mortensen, H. Deng, et L. Shapiro. A SIFT descriptor with global context. Dans *IEEE CVPR*, 2005.

[10] J. Amores, N. Sebe, et P. Radeva. Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors. Dans *IEEE CVPR*, 2005.

[11] D.A. Lisin, M.A. Mattar, et M.B. Blashcko. Combining local and global image features for object class recognition. Dans *IEEE CVPR*, 2005.

[12] K. Mikolajczyk et C. Schmid. A performance evaluation of local descriptors. *IEEE CVPR*, 2003.

[13] I. Laptev et T. Lindeberg. Space-time interest points. Dans *ICCV*, 2003.

[14] C. Tomasi et T. Kanade. Detection and tracking of point features. Technical report CMU-CS-91-132, Avril 1991.

[15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[16] V. Gouet et B. Lameyre. SAP : a robust approach to track objects in video streams with snakes and points. Dans *BMVC*, Kingston University, London, UK, Septembre 2004.

[17] M. Kass, A. Witkin, et D. Terzopoulos. Snakes : Active contours models. *IJCV*, pages 321–331, 1988.

[18] A. Blake et M. Isard. *Active Contours*. Springer, 1998.

[19] D. S. Zhang et G. Lu.. A comparison of shape retrieval using fourier descriptors and short-time fourier descriptors. Dans *PCM*, pages 855–860, 2001.

[20] C. Wallraven, B. Caputo, et A. Graf. Recognition with local features : the kernel recipe. Dans *ICCV*, 2003.

[21] S. Boughorbel, J.P. Tarel, et N. Boujemaa. The intermediate matching kernel for local image features. Dans *IJCNN*, 2005.

[22] O. Javed, M. Shah, et D. Comaniciu. A probabilistic framework for object recognition in video. Dans *ICIP*, 2004.

Construction de partitions géo-temporelles à partir d'une divergence de Kullback-Leibler modifiée en vue de la navigation dans une collection d'images personnelles

Marc Gelgon Antoine Pigeau Afshin Nikseresht

LINA FRE 2729 CNRS / INRIA ATLAS group

2, rue de la Houssinière 44322 Nantes - France

{prénom.nom}@univ-nantes.fr

Concours Jeune Chercheur : Non

Résumé

L'usage d'appareils mobiles équipés de capteurs photographiques entraîne le problème de la gestion de larges collections d'images personnelles. Suivant les études sur les besoins d'utilisateurs, nous proposons une technique d'organisation basée sur les méta-donnes géo-temporelles des images. Notre algorithme permet de construire incrémentalement une hiérarchie de deux partitions, l'objectif étant de représenter la structure temporelle ou géographique de la collection dans deux hiérarchies distinctes. Pour chaque partition de la hiérarchie, des critères statistiques (la vraisemblance complétée intégrée et une divergence de Kullback-Leibler modifiée) sont optimisés afin de déterminer ou mettre à jour leurs paramètres. L'approche probabiliste permet une flexibilité dans l'évolution des partitions.

Mots clefs

Classification d'images, Application sur les terminaux mobiles, Méta-données temporelles et géographiques, Classification statistique

1 Objectif et état de l'art

L'utilisation courante d'appareils mobiles équipés de capteurs photographiques (téléphone mobile, appareil photographique numérique) permet à un utilisateur de concevoir de grandes collections d'images. Un besoin essentiel étant de fournir aux utilisateurs des solutions pour rechercher ses images parmi des milliers, l'indexation de ce type de données multimédia est un domaine de recherche présentant de nombreux intérêts. Les produits Lifeblog de Nokia et MyLifeBits de Microsoft sont deux réponses récentes de la part d'industriels.

Les particularités de la tâche, comparées aux approches basées sur le contenu des données multimédia, réside dans les méta-données disponibles des images fournies par l'appareil d'acquisition (date, localisation géographique, paramètres de prise de vue) et les critères d'organisation préférés des utilisateurs. Des études [1] ont montré sans surprise que les interactions sociales, les événements, la date et les lieux sont des critères pertinents.

Dans ce papier, nous utilisons seulement les méta-données temporelles et géographiques attachées à chaque image. Nous faisons l'hypothèse que les coordonnées géographiques sont fournis par un système GPS/E-OTD. L'ensemble des données à structurer est ainsi un flux $\{(t,(x,y)) \in \mathbb{R} \times \mathbb{R}^2\}$. Notre contribution est une technique (les critères statistiques et l'algorithme incrémental pour les optimiser) pour construire automatiquement une classification d'images sous forme de hiérarchie de deux partitions (une partition *générale* et une partition *fine*), en se basant sur la date et le lieu de prise de vue. Deux hiérarchies distinctes sont construites pour la date et le lieu, respectivement à partir des méta-données temporelles et géographiques, avec une méthode presque identique. Les hiérarchies sont construites incrémentalement, au fur et à mesure que les méta-données sont ajoutées, les phases d'acquisition et de parcours de la collection étant très liées.

Notre approche est orientée vers la navigation dans la collection contrairement à une approche orientée requête. En effet l'utilisateur connaît la structure de la collection et les interactions homme-machine sur les appareils mobiles sont limitées. Notre objectif est de permettre à un utilisateur de parcourir l'organisation proposée à partir de résumés des classes, définis chacun par un petit sous-ensemble d'images visuellement représentative de leur classe. Le choix des images représentatives n'est pas résolu dans ce papier, mais des solutions existes dans le contexte de résumés de données vidéos. Notons de plus que les structures d'organisation proposées peuvent aussi améliorer la gestion des mémoires caches dans un contexte d'application mobile client-serveur [2].

La plupart des solutions d'organisation existantes utilise comme critère la date des images, celle-ci étant intuitive, disponible et fiable. La segmentation incrémentale d'un flux de données temporelles est proposée dans [3]. Les travaux les plus proches des nôtres organisent aussi les images à partir de la date et du lieu de prise de vue. Leur princi-

pal avantage est la prise en compte simultanée des deux critères, mais la solution proposée est basée sur des paramètres arbitraires et n'est pas incrémentale. Dans notre proposition nous évitons de tels paramètres grâce à une approche probabiliste et nous fournissons un algorithme incrémental. De notre point de vue, cette dernière propriété est importante pour organiser la collection, l'utilisateur n'ayant pas besoin de régulièrement penser à mettre à jour sa classification. En tournant en tâche de fond sur l'appareil mobile de l'utilisateur avec une faible priorité, la complexité de mise à jour de notre proposition est loin d'être aussi coûteuse que des codeurs vidéo en temps réels fonctionnant sur de telle plate-forme.
Ce papier est organisé comme suit. La section 2 présente la technique d'organisation, en détaillant les constructions des deux niveaux de la hiérarchie. La section 3 fournit des résultats expérimentaux et la section 4 résume notre contribution et propose quelques perspectives.

2 Construction de la hiérarchie de deux partitions

Initialisation :
ajout d'une nouvelle donnée dans l'ensemble D et itérer l'algorithme EM, initialisé avec le modèle obtenu après l'ajout de la nouvelle donnée précédente. Soit $\mathcal{M}_f$ le modèle correspondant.
Phase de division :
ordonner les candidats $\{S_1, \ldots, S_d\}$ à diviser selon le critère d'entropie $E = -\sum_{i=1}^{n} t_{ik} \cdot log(t_{ik})$ (nous divisons les classes avec des paramètres se chevauchant afin de faire apparaître un nouveau groupe de données).
pour les α premières composantes ordonnées par entropie décroissante :
– diviser la composante et mettre à jour le modèle à partir de $\mathcal{M}_f$;
– itérer l'algorithme EM jusqu'à convergence.
garder parmi les modèles testés le nouveau modèle $\mathcal{M}_f$ optimisant le critère ICL.
Phase de fusion :
ordonner les paires de composantes à la fusion en fonction de leur distance de Mahalanobis (en quelque sorte, nous testons les fusions des composantes les plus proches).
pour les α premières composantes ordonnées selon une distance croissante de Mahalanobis :
– fusionner les composantes et mettre à jour le modèle à partir de $\mathcal{M}_f$;
– itérer l'algorithme EM jusqu'à convergence.
garder le modèle $\mathcal{M}_f$ optimisant le critère ICL.

Figure 1 – *Algorithme pour déterminer la partition* fine.

Notre objectif étant de faciliter le parcours de la collection sur un appareil mobile, nous devons prendre en compte deux critères sensibles : les contraintes matérielles, la taille de l'écran en particulier, et le nombre de groupes pour représenter la structure des données. Notre solution consiste ainsi à construire une hiérarchie à deux partitions :
– en définissant manuellement le nombre de classes dans la partition *générale* selon les contraintes d'interfaces ;
– en déterminant, au fur et à mesure de l'ajout de nouvelles méta-données, le nombre de groupes dans la partition *fine* à l'aide d'un algorithme non-supervisé.
Pour chaque nouvelle prise de vue, nous devons ainsi mettre à jour 4 partitions : une partition *générale* et une partition *fine* pour chacune des hiérarchies temporelle et spatiale. Nous présentons ici les grandes lignes de notre approche, et explicitons les modèles et algorithmes dans les sections suivantes.

1. la construction de la partition *fine* suit la technique proposée dans [4]. Elle est basée sur une représentation des données avec des modèles de mélange probabiliste gaussien. Le critère ICL (la vraisemblance complétée intégrée) permet de déterminer le nombre de classes dans le modèle. Il présente l'avantage d'être robuste face aux données non gaussiennes. Un algorithme d'optimisation de ce critère, basé sur des fusion et divisions de classes, permet une recherche semi-locale dans les valeurs des paramètres pour mettre à jours le modèle incrémentalement. Enfin une estimation bayésienne des matrices de covariance préserve de l'instabilité des paramètres dans le cas où les classes sont associées à un trop petit échantillon de données. Le choix de l'approche basée sur les modèles de mélange présente deux avantages :
   – il permet d'éviter le problème d'explosion combinatoire inhérente aux problèmes de regroupement de données ;
   – il convient bien à la nature incrémentale de la tâche. Les affectations des données aux classes sont flexibles et permettent une évolution souple des partitions au fur et à mesure de l'ajout de nouvelles données.
2. la partition *compacte* des données est déterminée en identifiant des groupes de classes pertinents dans la partition *fine* (la technique diffère sensiblement de celle proposée dans [4]). Tandis que le nombre de classes dans la partition *fine* est déterminé automatiquement à partir des données, la partition *compacte* a un nombre de classes plus faible défini explicitement en prenant en compte les contraintes matérielles. L'objectif est de fournir un résumé de la collection à partir duquel l'utilisateur peut ensuite obtenir du détail avec la partition *fine*. Le regroupement des classes de la partition *fine* est formulé comme l'identification d'un modèle de mélange plus général, obtenu en minimisant une distance entre ces deux partitions. Cette distance est définie par une divergence de Kullback-Leibler modifiée. Le problème combinatoire du regroupement des classes de la partition *fine* peut être

résolu à l'aide d'un algorithme itératif opérant en alternant la phase d'estimation des paramètres du modèle général et la phase de ré-assignement des classes de la partition *fine* aux classes de la partition *compacte*. L'approche est similaire aux techniques classiques de partitionnement, mais elle opère sur les paramètres des classes et non pas sur les données initiales. L'approche présente les propriétés suivantes :

- la complexité en temps de calcul est faible puisque seuls les paramètres, et non pas les données initiales, de la partition *fine* sont utilisés ;
- la mise à jour de la partition *compacte* est incrémentale. L'initialisation des regroupements de la partition *fine* peut être fixée à partir de la configuration obtenue à l'étape précédente. En plus de réduire la complexité en temps de calcul, cela permet une meilleure stabilité de la partition *compacte* au cours du temps, un point important pour l'utilisateur ;
- la méthode n'est pas affectée par l'aspect non-gaussien des méta-données initiales, ce qui est le cas d'autres approches essayant de trouver une hiérarchie de modèles à partir des données.

2.1 Recherche de la partition *fine*

Nous optons pour une approche basée sur les modèles de mélange, les modèles probabilistes permettant d'obtenir les classes et les affectations des données aux classes. Ce dernier point est pertinent pour notre aspect incrémental, puisque les affectations des données aux classes sont flexibles et peuvent évoluer facilement lorsque l'ajout de nouvelles données dans la classification suggère une reconsidération de la structure précédemment trouvée.
Les données D (soit les lieux (x, y) ou les dates t) sont ainsi supposées être générées aléatoirement à partir d'un modèle de mélange gaussien.

Critère de sélection de modèles :

Dans le cadre des modèles de mélange, un bon critère de comparaison entre plusieurs hypothèses de classification ayant un nombre différent de classes est la vraisemblance complétée intégrée (ICL) [5]. Pour une hypothèse H_k, le critère ICL est défini par :

$$p(D, Z|H_K) = \int p(D, Z|\Theta_K, H_K)p(\Theta_K|H_K)d\Theta_K \quad (1)$$

où $\Theta_K = (\theta_1, \ldots, \theta_K)$ est le vecteur de paramètres de H_K et $\theta_i = (w_i, \mu_i, \Sigma_i), 1 \leq i \leq K$ (w_i est la proportion de mélange de la composante i, Σ_i sa covariance et μ_i son centre). Le calcul pratique de cette expression exploite une approximation du critère BIC (1), exprimée par :

$$ICL = -ML + \frac{\nu_K log(n)}{2} - \sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik} \cdot log(t_{ik}) \quad (2)$$

où ML est la log-vraisemblance du modèle de mélange optimisé, ν_K est le nombre de paramètres libres du modèle à K composantes, n est le nombre de données et t_{ik} est la probabilité à posteriori pour une observation i d'avoir pour origine la composante k. Les t_{ik} sont en fait les espérances des affectations binaires probabilisées z_{ik}. En pratique, les z_{ik} sont déterminés pendant l'étape E de l'algorithme EM, décrit dans la suite.
En comparaison avec le critère BIC approximant la vraisemblance marginalisée des données, ce critère optimise conjointement la vraisemblance des données et les labels inconnus z d'affectations des données au modèle. L'introduction de ces variables auxiliaires permet de prendre en compte la qualité de la partition lors de l'estimation des paramètres. L'expression (2) est une variation du critère BIC : le terme de droite ajouté a un comportement entropique et favorise les classes bien séparées [5].
Il est fréquent qu'une classe soit assignée à un petit échantillon de données, entraînant une mauvaise estimation de sa covariance. Nous traitons le problème en introduisant, à l'étape M de l'algorithme EM, des estimations de covariances régularisées calculées à partir des espérances des distributions à posteriori des matrices de covariances initiales (utilisant les modèles à priori conjugués de Gamma pour le temps (une dimension) et de Wishart pour le lieu (deux dimensions)).

Algorithme d'optimisation :

La recherche de partitions de données incrémentale suppose de pouvoir modifier facilement l'affectation des données aux classes et d'ajuster le nombre de composantes en fonction des nouvelles données. Notre proposition consiste à utiliser la partition obtenue au temps t comme initialisation de l'optimisation du critère ICL pour la partition à $t + 1$: cela garantit la stabilité de la partition au cours du temps et facilite ainsi son exploration par l'utilisateur.
L'algorithme Expectation-Maximization (EM) [6] permet d'optimiser localement le critère ICL à un nombre constant de composantes. Il se décompose en deux étapes, l'étape E, dans laquelle les probabilités d'affectation des données à chaque composante sont calculées, conditionnellement aux paramètres des modèles, et l'étape M, dans laquelle les paramètres des modèles sont estimés en se basant sur l'estimation courante des affectations des données aux modèles.
Il reste deux problèmes à résoudre :

- l'évolution du nombre de composantes ;
- l'optimisation du critère ICL (équation 2) : l'utilisateur prenant généralement ses images par paquets, le flux de données ne peut pas être modélisé comme une série de données indépendantes. De plus grands efforts de réorganisation des classes sont ainsi nécessaires que si les données étaient totalement indépendantes. Par exemple, au fur et à mesure que les données du même lieu sont ajoutées, les classes sont optimisées localement sur ce lieu spécifique et peuvent être ainsi plus difficiles à modifier si un nouveau lieu apparaît. L'optimisation au cours du temps est ainsi délicate et des minima locaux sont souvent obtenus si seul un algorithme EM classique

est utilisé.

Notre algorithme EM incrémental consiste à tester plusieurs divisions de composantes suivies de plusieurs fusions pour chaque nouvelle donnée rajoutée [4]. On alterne ainsi les phases de recherche semi-locale avec des itérations de l'algorithme EM, jusqu'à convergence. Les deux étapes permettent de minimiser le critère ICL et servent le même objectif : éviter les minima locaux et permettre l'évolution du nombre de composantes au cours du temps. L'algorithme est détaillé par la figure 1.

2.2 Recherche de la partition *compacte*

Soit M_f le modèle de mélange obtenu pour la partition *fine*. Trouver des groupes pertinents de ces classes peut être exprimé par la recherche d'un modèle M_c maximisant la log-vraisemblance des données D supposées être générées à partir de M_f (voir (3)). Cela revient à minimiser la divergence de Kullback-Leibler $KL(M_f \| M_c)$ [7], définie par (4), qui représente la perte d'information due à l'approximation de M_f par M_c.

$$\hat{M}_c = \arg\max \quad E_{M_f} \; [\; ln \; p(D|M_c) \;] \tag{3}$$

$$\hat{M}_c = \arg\min \quad \left[- \int M_f(x) \; ln \; \frac{M_c(x)}{M_f(x)} \; dx \right] \tag{4}$$

Un problème est l'absence d'une forme approchée pour cette divergence dans le cas des modèles de mélange gaussien. Il est résolu en se basant sur une variation de cette divergence, proposée récemment par [8], qui consiste à minimiser la mesure de similarité suivante

$$d(M_f, M_c) = \sum_{i=1}^{m_f} w_f^i \min_{j=1}^{m_c} KL(N_f^i \| N_c^j) \tag{5}$$

où N_f^i (resp. N_c^i) est la $i^{\text{ème}}$ composante de M_f (resp. de M_c), m_f, m_c représente respectivement le nombre de composantes dans M_f et M_c et w_f^i est la proportion de mélange de la composante i du modèle M_f.

Cette mesure de similarité présente deux propriétés :

– la complexité en temps de calcul est faible puisque la divergence de Kullback-Leibler entre deux gaussiennes, dont les paramètres sont (μ_1, Σ_1) and (μ_2, Σ_2), est exprimée par l'expression simple suivante :

$$\frac{1}{2}(log\frac{|\Sigma_2|}{|\Sigma_1|} + Tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2) - \delta) \tag{6}$$

où δ est la dimension de l'espace des données. Nous disposons ainsi d'une expression analytique facilement calculable pour déterminer une divergence de Kullback-Leibler modifiée.

Ainsi, suivant [8], nous optimisons localement un critère avec un algorithme itératif, détaillé par la figure 2. Il peut être comparé à un algorithme K-means appliqué aux composantes d'un modèle de mélange.

À partir de l'initialisation $\hat{\pi}^0$ obtenue à convergence au temps $t-1$ (si nécessaire en prenant en compte les apparitions ou disparitions de composantes de la partition *fine*).
$it = 0$
Répéter
Mise à jour des composantes pour le modèle M_c :
connaissant la classification courante $\hat{\pi}^{it}$, obtenue initialement ou calculer à l'étape précédente, mettre à jour les paramètres du modèle de mélange comme suit :

$$\hat{M_c}^{it} = \arg \min_{M_c \in \mathcal{M}_{m_c}} d(M_f, M_c, \hat{\pi}^{it}) \tag{7}$$

où $\mathcal{M}_{m_c}$ est l'espace de tous les modèles de mélange avec m_c composantes pouvant être formé à partir des regroupements des composantes de M_f. Cette ré-estimation revient à mettre à jour chaque composante de M_c comme suit. Pour la composante j, le calcul conduit à l'expression suivante :

$$\hat{w}_c^j = \sum_{i \in \pi^{-1}(j)} w_f^i, \tag{8}$$

$$\hat{\mu}_c^j = \frac{\sum_{i \in \pi^{-1}(j)} w_f^i \mu_f^i}{\hat{w}_c^j}, \tag{9}$$

$$\hat{\Sigma}_c^j = \frac{\sum_{i \in \pi^{-1}(j)} w_f^i (\Sigma_f^i + (\mu_f^i - \hat{\mu}_c^j)(\mu_f^i - \hat{\mu}_c^j)^T)}{\hat{w}_c^j} \tag{10}$$

où $\pi^{-1}(j)$ est une notation simplifiée de $\hat{\pi}^{-1,it}(j)$, l'ensemble des composantes de M_f projetées dans la composante j de M_c.

Optimisation de la divergence de Kullback-Leibler modifiée 5 :
soit le modèle de mélange $\hat{M}_c^{it}$ obtenu à l'étape précédente, rechercher la transformation π^{it+1}, définie à partir de $\{1, \ldots, m_f\}$ vers $\{1, \ldots, m_c\}$, qui regroupe au mieux les composantes de M_f pour construire les composantes $\hat{M}_c^{it}$, de la façon suivante :

$$\hat{\pi}^{it+1} = \arg\min_{\pi} \; d(M_f, \hat{M}_c, \pi) \tag{11}$$

En d'autres termes, chaque composante i de M_f est projetée vers la composante j de $\hat{M}_c^{it}$ la plus proche, selon la convergence de Kullback-Leibler modifiée ((12) ci-dessus). Cette phase revient à une recherche exhaustive parmi les composantes sources et présente une complexité en temps de calcul faible grâce à la disponibilité de (6).

$$\pi^{it+1}(i) = \arg\min_j KL(N_f^i || N_c^j) \tag{12}$$

it=it+1
jusqu'à convergence (i.e. $\pi^{it+1} = \pi^{it}$)

Figure 2 – *Algorithme pour déterminer la partition* compacte.

3 Expérimentation

Nous avons réalisé des expériences sur une collection personnelles réelles de 721 images. Elle regroupe des événements pris sur trois ans dans plusieurs pays. Le nombre de composantes de la partition *compacte* est fixé à 4.

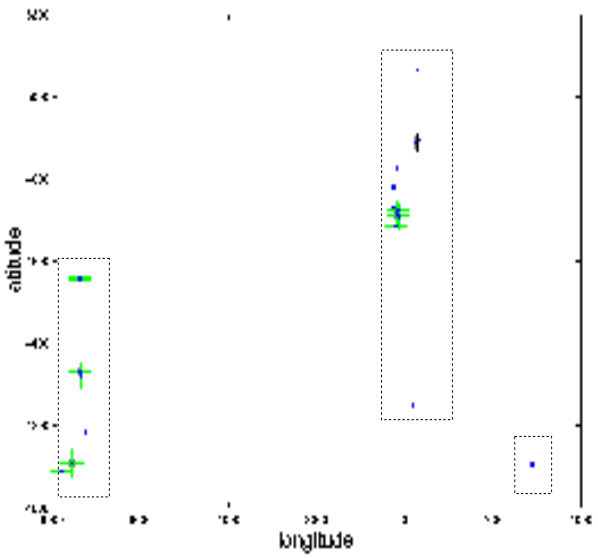


Figure 3 – *Hiérarchie spatiale : les classes de la partition fine sont représentées par le signe* $+$*. La partition compacte est définie par les lignes en pointillées.*

La figure 3 présente la partition spatiale obtenue. La partition *compacte* est définie par les lignes en pointillées et la partition *fine* par les signes $+$. Cette dernière comprend des classes de données très compactes, dû à la configuration très regroupées des données. L'utilisateur a pris en général beaucoup d'images dans des lieux précis et chaque lieu a été convenablement retrouvé dans 25 classes. La partition *compacte* est quant à elle composée de trois classes. Notre algorithme permet, dans le cas où une composante de la partition *compacte* n'est associée à aucune classe de la partition *fine*, d'adapter la complexité du modèle. Ainsi le modèle général obtenu ne contient que 3 composantes ce qui semble cohérent au vu de la structure des données.
La figure 4 présente la classification temporelle finale obtenue pour 150(a), 300(b) et 721(c) images. La partition *compacte* est définie par les lignes en pointillées et la partition *fine* par les lignes continues. La partition *fine* finale (figure 4(c)) comprend 41 classes et nous avons vérifié que chaque limite de classes correspondait bien à un changement d'événements dans la collection. Les partitions obtenues sont stables dans le temps puisque les différents états obtenus présentent des similarités. Les résumés proposés dans chacune des partitions sont correctes, les limites de classes étant visuellement justifiées. Les principaux groupes de données sont bien mis en valeur.

4 Conclusion

Ce papier propose une technique d'organisation d'une collection d'images personnelles acquises sur un mobile, à partir des méta-données temporelle et spatiale. Le choix de ces méta-données est motivé par leur disponibilité et leur bonne interprétation par les utilisateurs. D'autres critères, par exemple sur le contenu de l'image, peuvent bien sur être utilisés en conjonction avec ces méta-données.
La contribution de ce papier est une technique de classification automatique, limitant les paramètres arbitraires critiques et dépendant ainsi seulement de la structure des données. Le principe est de construire progressivement une hiérarchie de modèles de mélange de deux partitions, en fonction de l'ajout de nouvelles images dans la collection. Le critère ICL maintient une définition uniforme de la partition *fine* et un critère de divergence de Kullback-Leibler modifiée est utilisé pour résumer la partition *fine* à partir de ses paramètres. L'intérêt de l'approche est le très faible coût de calcul de la partition *compacte*, mise à jour incrémentalement au fur et à mesure de l'ajout de nouvelles images. Elle présente de plus l'avantage de s'emboîter parfaitement dans la partition *fine*.
Une perspective de travail est de proposer une méthode pour déterminer le nombre de classes dans la partition *compacte* en fonction de la structure des données. Pour chaque modification de la partition *fine*, il serait nécessaire de tester plusieurs solutions de résumés avec un nombre variable de classes et de sélectionner la partition *compacte* la plus pertinente. Une méthode basée sur un critère statistique, par exemple le critère AIC, est en cours d'étude.

Références

[1] K. Rodden. How do people manage their digital photographs ? Dans *ACM Conference on Human Factors in Computing Systems*, pages 409 – 416, Fort Lauderdale, 2003.

[2] A. Myka, J. Yrjanainen, et M. Gelgon. Enhanced storing of personal content. US Patent 16660/10502275, Nokia corp., Juillet 2004.

[3] J. C. Platt et B. A. Field M. Czerwinski. PhotoTOC : Automatic clustering for browsing personal photographs. Rapport technique MSR-TR-2002-17, Microsoft Research, Février 2002.

[4] A. Pigeau et M. Gelgon. Building and tracking hierarchical partitions of image collections on mobile devices. Dans *ACM Multimedia conference*, pages 141–150, Singapore, nov 2005.

[5] C. Biernacki, G. Celeux, et G. Govaert. Assessing a mixture model for clustering with the integrated classification likelihood. Dans *IEEE Transaction on pattern analysis and machine intelligence*, volume 22, pages 719–725, Juillet 2000.

[6] A. P. Dempster, N. M. Laid, et D. B. Rubin. Maximum likelihood for incomplete data via the EM algorithm. *J. R. Stat. Soc.*, pages 1–38, 1977.

[7] C. Bishop. *Neural networks for Pattern Recognition*. Oxford University Press, 1995.

[8] J. Goldberger et S. Roweis. Hierarchical clustering of a mixture model. Dans *Proc. of Neural Information Processing Systems (NIPS'2004)*, pages 505–512, 2004.

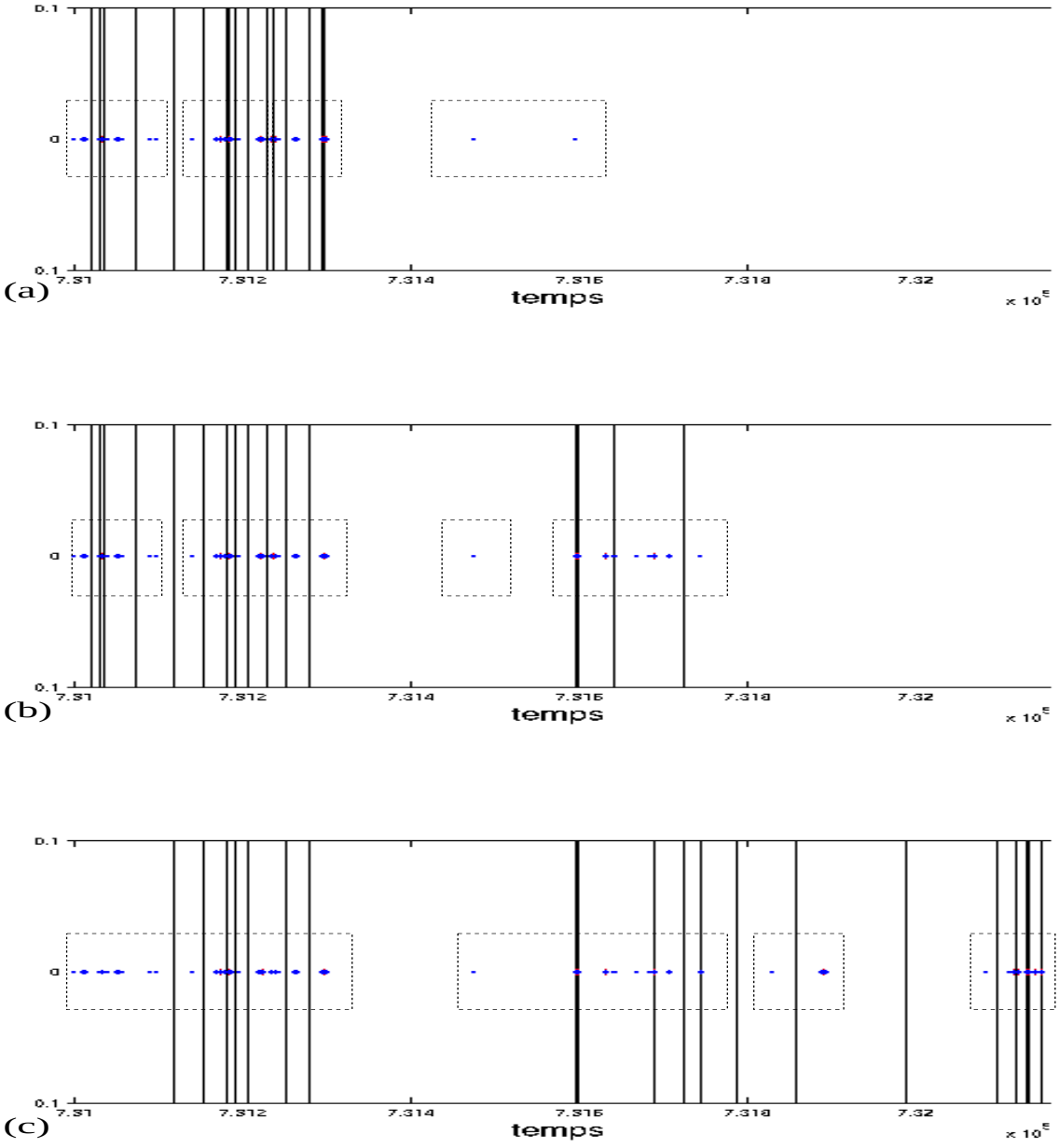


Figure 4 – *Hiérarchie temporelle : plusieurs états de la partition obtenue pour 300, 450 et 721 données. Les limites de classes dans la partition fine sont représentées par les lignes continues. La partition compacte est définie par les lignes en pointillées.*

Énergies ASSOM pour la détection d'objets

Grégoire Lefebvre[1] Christophe Garcia[1] Jean Marc Salotti[2]

[1] France Telecom R&D
4, Rue du Clos Courtel
35512 Cesson Sévigné
{prénom.nom}@orange-ft.com

[2] Institut de Cognitique
146, rue Léo Saignat
33076 Bordeaux Cedex
salotti@idc.u-bordeaux2.fr

Résumé

Cet article présente une nouvelle méthode caractérisant le contenu visuel des images. Cette problématique est récurrente dans de nombreuses applications telles que la classification d'images ou la reconnaissance d'objets. Nous proposons d'utiliser des cartes auto-organisatrices ASSOM [1] *(*Adaptive-Subspace Self-Organizing Map*) pour évaluer les caractéristiques communes entre images. Pour atteindre cet objectif, des signatures locales sont extraites des objets à traiter et sont injectées dans plusieurs cartes ASSOM. Pendant la phase d'apprentissage, un processus de compétition intervient, au sein de chaque carte, pour représenter au mieux les concepts des différentes classes d'objets à identifier. La convergence des réseaux nous assure en fin de traitement la constitution de cartes d'activation caractéristiques des différentes catégories. Ainsi, chaque image introduite dans le système active les cartes avec différentes intensités. Ces énergies d'activation peuvent être représentées par un histogramme d'activation des cellules des réseaux ASSOM et constituent le vecteur caractéristique de l'image pour une classification supervisée par* Support Vector Machine *(SVM). Une approche multi-échelle complétée d'un parcours de l'image par une fenêtre glissante, permet la localisation et la classification des objets présents dans la scène. Ce schéma offre des résultats prometteurs avec* $85,08\%$ *de bonne classification pour un exemple de base de* 689 *images à trier en quatre catégories*[1].

Mots clefs

Indexation, classification, ASSOM, SVM.

1 Introduction

Selon les études psychovisuelles de Hoffman [2], le système visuel humain exécute des mouvements saccadées entre des régions saillantes pour capturer le contenu des images. De nombreux travaux en vision par ordinateur s'inspirent de cette observation pour décrire l'information visuelle des images dans une optique d'indexation, de classification ou de détection d'objets. Contrairement aux approches globales, pour lesquelles une signature unique est calculée en considérant tous les pixels de l'image avec la même importance, ces approches locales représentent le contenu de l'image par un ensemble de signatures locales centrées autour de points saillants. La détection de ces points d'intérêt [3, 4, 5] se focalise ainsi dans les zones perceptuellement significatives de l'image.

Les travaux de Tversky [6] montre également que comparer deux images revient à détecter des concepts d'appartenance et d'exclusion entre ces régions saillantes. Notre méthode tente de reproduire cette extraction de concepts par la constitution d'une carte d'activation ASSOM [1] pour chaque catégorie d'objets. La distinction entre ces concepts est alors réalisée par la comparaison des activations des cartes ASSOM respectives.
Cette architecture offre des résultats prometteurs avec $85,08\%$ de bonne classification pour une base de 689 images à trier en quatre catégories.

Ce papier est organisé comme suit : la section 2 présente notre système de détection d'objets, construit sur les énergies d'activation des cartes ASSOM. Ensuite, les résultats expérimentaux de la section 3 illustrent les performances d'une telle architecture. Finalement, nous dressons quelques conclusions.

2 Détection d'objets basée sur les énergies ASSOM

2.1 Énergies des cartes d'activation ASSOM

Comme le souligne R.O. Duda [7], un processus de classification se déroule en trois étapes : une étape de prétraitements, une seconde d'extraction de caractéristiques et finalement une étape de classification. Dans cette étude, nous nous intéressons principalement aux deux premières phases, la dernière étant réalisée par un classifieur SVM.

L'architecture du système se décompose en six processus complémentaires dans la phase d'apprentissage (cf. figure 1) :

[1] http ://www.pascal-network.org/challenges/VOC/voc2005/

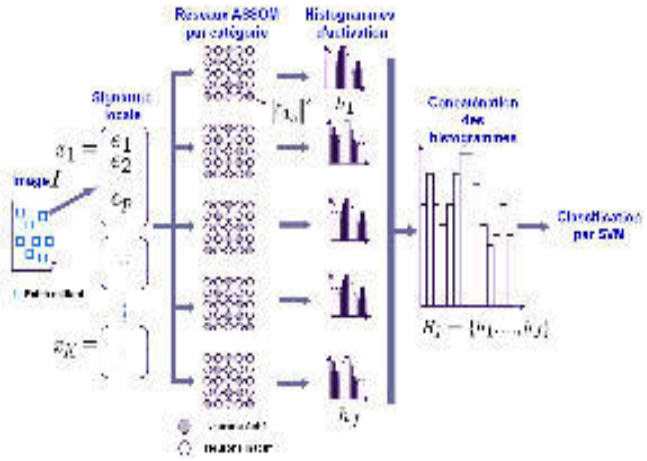


Figure 1 – *Architecture globale du système*

– Nous opérons, tout d'abord, une recherche des zones saillantes, en se focalisant sur les frontières anguleuses obtenues par transformation de l'image à traiter en ondelettes de Haar.
– Ensuite, des caractéristiques visuelles locales sont calculées pour décrire l'orientation et la régularité des singularités présentes dans les zones saillantes.
– Ces informations sont alors injectées au fur et à mesure dans des réseaux ASSOM spécifiques à chaque classe pour déterminer en sortie des prototypes sous forme de cartes d'activation neuronale. Ces cartes d'activation synthétisent les structures informatives au sein de chaque classe d'observation.
– Les énergies d'activation sont représentées sous la forme d'un histogramme pour chaque classe.
– Les histogrammes sont concaténés pour constituer le vecteur caractéristique global de l'image.
– Ce dernier est le support de l'information pour un apprentissage supervisé avec un classifieur SVM.

Dans la phase de test du système, un parcours multi-échelle d'une rétine caractéristique est réalisé. Pour chaque imagette prélevée par la rétine, le système reproduit les six étapes précédentes, pour déterminer les sorties du classifieur SVM. Ces sorties permettent la construction de cartes de vote pour la détection et la localisation des objets recherchés.

2.2 Localisation de points d'intérêt

Selon les mécanismes de la vision humaine active, la détection de points saillants révèlent les localisations perceptuellement importantes au sein des images. De nombreux descripteurs sont proposés dans la littérature [3, 4, 5]. Le détecteur de Harris [4] localise les zones de saillance sur des coins, en cherchant les maxima d'une fonction basée sur la matrice d'auto-corrélation locale du signal. Le détecteur de contraste [5] propose de situer les points d'intérêt dans les zones fortement contrastées. L'étude de la position des points saillants dans l'article [3] utilise une décomposition en ondelettes de Haar pour déterminer les pixels sur les régions frontières anguleuses.
L'utilisation de cette base d'ondelettes est inspirée par le système visuel humain, qui donne une grande importance à une analyse des orientations et des fréquences en multi-résolution. Afin d'extraire ces points pertinents, la transformée ondelette est réalisée sur l'image en niveaux de gris. Puis, les coefficients d'ondelettes obtenus sont présentés sous la forme d'un arbre *zerotree* introduit par Shapiro[8]. Cet arbre est alors parcouru une première fois des feuilles à la racine pour calculer la valeur saillante de chaque noeud. Puis, un second parcours de la racine vers les feuilles détermine pour chaque noeud la valeur de plus forte saillance. Cette détection focalise les points d'intérêt sur les frontières de forte luminance et les ombres. Pour réduire les fausses localisations causées par les conditions d'illuminations, les auteurs utilisent les invariants couleurs démontrés par Geusebroek et co-auteurs [9].

Les résultats expérimentaux montrent que ces pixels sont localisés dans les régions saillantes, c'est pourquoi nous combinons ces travaux avec notre approche de sélection de caractéristiques par réseaux ASSOM. L'apprentissage neuronal peut être alors réalisé sur des informations locales centrées sur ces points d'intérêt.

2.3 Description des régions saillantes

La plupart des descripteurs caractérisent le voisinage local des points saillants par les contours présents dans cette région [10]. Pour décrire ces contours, l'orientation et la magnitude des gradients de l'image sont utilisés. Dans de récentes études [11], le contour est considéré comme une singularité décrite par les coefficients de Hölder.

Définition 2.3.1. *La fonction* $f : [a, b] \rightarrow \Re$ *est Hölder-*α *(*$\alpha \geq 0$*) en* x_0 *si* $\exists K > 0, \delta > 0$ *et un polynôme* P *de degré* $m = \lfloor \alpha \rfloor : \forall x, x_0 - \delta \leq x \leq x_0 + \delta, |f(x) - P(x - x_0)| \leq K |x - x_0|^{\alpha}$.

Définition 2.3.2. *L'exposant de Hölder* $h_f(x_0)$ *de* f *en* x_0 *est définie par :* $h_f(x_0) = \sup\{\alpha, f$ *Hölder-*α *en* $x_0\}$.

$h_f(x_0)$ mesure la régularite locale de f en x_0. Plus cette valeur est faible, plus le signal est considéré comme singulier. Pour une image, l'exposant de Hölder est calculé dans la direction de la régularité minimale de la singularité. Pour décrire une région d'intérêt, à la fois l'orientation et la régularité des singularités sont utilisées. Ainsi l'orientation $\theta(x, y)$ et la magnitude $m(x, y)$ sont calculés pour chaque pixel.

$$\begin{cases} m(x,y)^2 = & (I_j(x+1,y) - I_j(x-1,y))^2 \\ & +(I_j(x,y+1) - I_j(x,y-1))^2 \\ \theta(x,y) = & \tan^{-1}(\frac{I_j(x,y+1) - I_j(x,y-1)}{I_j(x+1,y) - I_j(x-1,y)}) \end{cases} \quad (1)$$

L'exposant de Hölder est estimé selon des ondelettes fovéales comme décrit dans l'étude [12]. Les orientations et

les exposants de Hölder sont utilisés conjointement pour construire des histogrammes $3D$. Chaque région d'intérêt est partitionnée en bloc 4×4. L'histogramme est calculé dans chaque bloc et normalisé par la taille du bloc. La signature locale est obtenue par la concaténation des histogrammes $3D$. Sa taille est $n \times r \times o$ avec n le nombre de sous-régions, r le nombre d'exposants de Hölder compris entre $[-1.5, 1.5]$ et o le nombre d'orientation appartenant à l'intervalle $[-\pi/2, \pi/2]$. Cette signature forme le descripteur RFD (Regularity Foveal Descripteur).

2.4 Apprentissage des réseaux neuronaux ASSOM

Les réseaux ASSOM [1] sont une combinaison d'une méthode de sous espaces linéaires et d'un apprentissage compétitif et coopératif du traditionnel algorithme SOM [1]. Ces réseaux ASSOM diffèrent des autres méthodes de sous espaces en générant un ensemble de sous espaces linéaires topologiquement ordonnés. Chaque cellule des réseaux ASSOM, appelée « module », est composée de plusieurs vecteurs de base qui décrivent ensemble un sous espace linéaire. Ainsi, deux modules proches dans la carte ASSOM représentent deux sous espaces proches dans l'espace des caractéristiques.
L'apprentissage ASSOM parvient à extraire les données caractéristiques de l'ensemble des observations, sans hypothèse sur leur représentation mathématique, contrairement aux méthodes basées sur les transformées de Gabor ou en ondelettes. En d'autres termes, la forme des fonctions de filtrage est construite directement à partir des données.
L'entrée d'un réseau ASSOM est un groupe de vecteurs, appelée « épisode ». Un vecteur de chaque épisode est supposé illustrer une transformation affine (rotation, translation, ou changement d'échelle) d'une observation à modéliser.

Il existe deux principales étapes dans ce processus d'apprentissage :

1. Pour chaque épisode, localiser le module « gagnant » dans la carte ASSOM ;
2. Ajuster le sous espace linéaire « gagnant » et les modules de son voisinage, afin de représenter au mieux l'épisode en entrée.

Pour un sous espace linéaire Λ de dimension M, on peut trouver un ensemble de vecteurs de base $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_M\}$, tel que tout vecteur peut être représenté par une combinaison linéaire de ces vecteurs de base. Ces ensembles ne sont pas uniques, mais sont équivalents dans le sens qu'ils décrivent le même sous espace linéaire. Pour des convenances de mesures mathématiques, les vecteurs de bases sont orthonormalisés par le procédé de Gram-Schmidt.
La projection orthogonale d'un vecteur arbitraire $\mathbf{x}$ sur un sous espace Λ, notée $\hat{\mathbf{x}}_\Lambda$, est une combinaison linéaire sur les vecteurs de base et peut être calculée par :

$$\hat{\mathbf{x}}_\Lambda = \sum_{m=1}^{M} (\mathbf{x}^{\mathrm{T}} \mathbf{b}_m) \mathbf{b}_m. \tag{2}$$

Si $\hat{\mathbf{x}}_\Lambda = \mathbf{x}$, alors $\mathbf{x}$ appartient au sous espace Λ. On peut définir la distance de $\mathbf{x}$ à Λ par $\|\tilde{\mathbf{x}}_\Lambda\| = \|\mathbf{x} - \hat{\mathbf{x}}_\Lambda\|$, selon la norme euclidienne. En comparant la distance entre un vecteur et plusieurs sous espaces linéaires, nous sommes en mesure de définir le sous espace le plus proche, c'est-à-dire le module gagnant.
Dans la réalisation de Kohonen, un sous espace est représenté par une architecture neuronale à deux couches, comme l'illustre la figure 2. Les neurones de la première couche réalisent les projections orthogonales $\mathbf{x}^{\mathrm{T}} \mathbf{b}_m$ du vecteur d'entrée $\mathbf{x}$ sur les vecteurs de bases $\mathbf{b}_m$. Le second est composé d'un neurone quadratique, produisant le carré de la somme des sorties de la couche précédente.
La sortie finale d'un module est alors $\|\hat{\mathbf{x}}_\Lambda\|^2$, le carré de la norme de la projection orthogonale de $\mathbf{x}$. Cette mesure peut être regardée comme le degré de correspondance entre le vecteur d'entrée $\mathbf{x}$ et le sous espace linéaire Λ. C'est cette énergie qui est utilisée pour mettre à jour les histogrammes d'activation de la section 2.5.

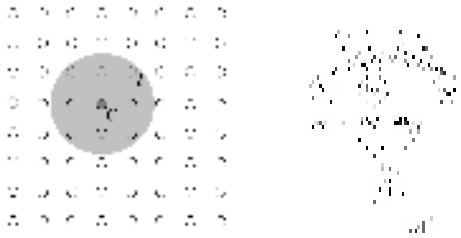


Figure 2 – *A gauche : un réseau ASSOM, son module c gagnant et le voisinage associé. A droite : la structure d'un module.*

La première phase achevée, le module gagnant et son voisinage sont ajustés pour représenter au mieux les données d'observation. Une fonction de voisinage $V_c^{(i)}$ est définie sur le treillis rectangulaire des réseaux ASSOM, où c est l'indice du module gagnant et i celui d'un module arbitraire sur le treillis. Le voisinage rétrécit avec les époques d'apprentissage, afin d'assurer un état topologiquement ordonné, où les modules proches décrivent des sous espaces semblables.
L'algorithme classique de Kohonen pour l'apprentissage d'un réseau ASSOM opère comme suit :

Pour chaque étape d'apprentissage t,

1. Introduire l'épisode d'entrée $\mathbf{x}(s)$, $s \in S$, où S est l'ensemble des indices des vecteurs de l'épisode. Trouver le module gagnant indicé par c :

$$c = \arg\max_{i \in I} \sum_{s \in S} \|\hat{\mathbf{x}}_{\Lambda_i}(s)\|^2, \tag{3}$$

où I est l'ensemble des indices des modules de ASSOM.

2. Pour chaque module i du voisinage de c, incluant c lui-même, et pour chaque vecteur d'entrée $\mathbf{x}(s)$, $s \in S$, ajuster le sous espace Λ_i par la mise à jour des vecteurs de base $\mathbf{b}_m^{(i)}$, selon la procédure :
   (a) Rotation de chaque vecteur suivant la formule :

$$\mathbf{b}_m^{(i)} = \mathbf{P}_c^{(i)}(\mathbf{x}, t)\mathbf{b}_m^{'(i)} \ . \tag{4}$$

   Dans cette règle de mise à jour, $\mathbf{b}_m^{(i)}$ est le nouveau vecteur après rotation de l'ancien $\mathbf{b}_m^{'(i)}$. $\mathbf{P}_c^{(i)}(\mathbf{x}, t)$ est la matrice de rotation définie par :

$$\mathbf{P}_c^{(i)}(\mathbf{x}, t) = \mathbf{I} + \lambda(t)V_c^{(i)}(t)\frac{\mathbf{x}(s)\mathbf{x}^{\mathrm{T}}(s)}{\|\hat{\mathbf{x}}_{\Lambda_i}(s)\|\|\mathbf{x}(s)\|} \ , \tag{5}$$

   où $\mathbf{I}$ est la matrice identité, $\lambda(t)$ le taux d'apprentissage décroissant avec le temps t.
   (b) Disperser les composants $b_{mj}^{(i)}$ des vecteurs de base $\mathbf{b}_m^{(i)}$ pour améliorer la stabilité des résultats :

$$\tilde{b}_{mj}^{(i)} = \mathrm{sgn}(b_{mj}^{(i)})\max(0, |b_{mj}^{(i)}| - \varepsilon), \tag{6}$$

   où ε est la quantité de dispersion, choisi comme étant proportionnelle à la force de correction des vecteurs de base.
   (c) Orthonormaliser les vecteurs du module i.

2.5 Construction des vecteurs caractéristiques

L'architecture du système repose sur un réseau ASSOM pour chaque catégorie à apprendre. Chaque cellule des réseaux ASSOM se spécialise selon les patches extraits vers un concept lié à une classe donnée. Cette idée fût introduite dans les travaux de Zhang et co-auteurs [13] pour la reconnaissance de chiffre manuscrit. Dans leur étude, dix réseaux ASSOM sont employés, un pour chaque chiffre. Pour la classification d'un chiffre test, l'image entière de ce chiffre de taille réduite (25×20 pixels) est envoyée en parallèle à tous les réseaux, résultant dix erreurs de classification. Le réseau possédant l'erreur minimale de reconstruction détermine la classe du chiffre test. Dans cette approche, il n'existe pas d'interaction entre les réseaux ASSOM pendant la phase d'apprentissage. Chaque réseau de neurones se focalise sur les caractéristiques de sa catégorie, mais n'apprend pas à distinguer les informations des autres catégories. La surface de décision optimum entre classe n'est donc pas garantie.
Dans notre contexte, les images à traiter sont de taille supérieure. C'est pourquoi, nous choisissons une approche locale pour extraire l'information pertinente au voisinage des points saillants. Notre stratégie est ainsi de créer le vocabulaire de chaque catégorie à partir de l'activation des différents réseaux ASSOM. Les études de Csurka [14] et Quelhas [15] s'intéressent également à la constitution de ce vocabulaire sous la forme de *codebooks* ou *bags of keypoints*. Ici, nous adoptons une stratégie basée « objet » afin de focaliser l'apprentissage sur les zones de l'images pertinentes. On supposons ainsi que les objets sont annotés dans la base d'images d'apprentissage et les images à apprendre sont ces objets.

Pour construire le vecteur caractéristique $\mathbf{H}_{\mathcal{I}}$ de l'objet $\mathcal{I}$, nous opérons comme suit (cf. figure 1 pour les notations) :
– On sélectionne les points d'intérêt de plus forte saillances de l'image $\mathcal{I}$(cf. section 2.2).
– Pour chaque région d'intérêt :
  – On calcule la signature locale du patch grâce au descripteur RFD (cf. section 2.3).
  – On constitue un épisode pour la signature locale en lui appliquant des transformations affines arbitraires.
  – Pour chaque vecteur de l'épisode :
    – Les $|J|$ réseaux ASSOM reçoivent une signature et établissent une énergie $\|\hat{\mathbf{x}}_{kj}\|^2$ définie par :

$$\|\hat{\mathbf{x}}_{kj}\|^2 = \max_{i \in I_j} \|\hat{\mathbf{x}}_{k\Lambda_i}\|^2 \ , \tag{7}$$

    où I_j est l'ensemble des indices des modules du j^e réseau ASSOM. $\|\hat{\mathbf{x}}_{kj}\|^2$ est la valeur maximale des carrés des projections orthogonales du patch $\mathbf{x}_k$ sur les sous espaces linéaires du j^e réseau ASSOM.
    – Chaque histogramme d'activation h_j correspondant à chaque réseau est alors mis à jour. L'énergie dégagée par la sortie maximale du réseau incrémente le bin correspondant pour son histogramme. Cet incrément est défini par :

$$h_j[i^*](t+1) = h_j[i^*](t) + \|\hat{\mathbf{x}}_{kj}\|^2 \tag{8}$$

    avec $i^* = \arg\max_{i \in I_j} \|\hat{\mathbf{x}}_{k\Lambda_i}\|^2$ et $h_j[i^*](t)$ la valeur de $h_j[i^*]$ à l'instant t.
– Chaque histogramme d'activation h_j, ainsi formé par l'ensemble des patches, est fusionné en un histogramme globale $\mathbf{H}_{\mathcal{I}}$, qui correspond au vecteur caractéristique introduit avec le label de l'image $\mathcal{I}$ pour un apprentissage supervisé du SVM.

2.6 Détection d'objets par multi-résolution

L'apprentissage étant ajusté sur des objets normalisés des images d'exemples, la procédure de classification des images tests consiste en une recherche d'objets. Cette recherche s'effectue par un parcours d'une fenêtre glissante dans une pyramide multi-résolutionnelle.
La détection des objets se réalise sur trois niveaux d'échelles, avec un parcours de la fenêtre glissante d'un pas de la moitié de sa largeur (cf. figure 3).
Pour chaque échantillon, nous observons la sortie du classifieur SVM et lorsqu'il reconnait un objet appris, la zone correspondante est marquée dans la carte de vote de la catégorie reconnue comme étant pertinente. Le poids crédité

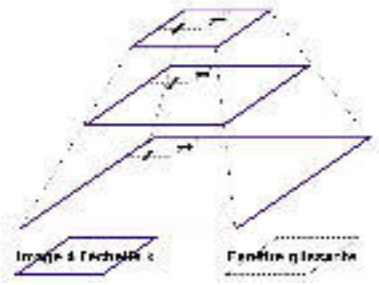


Figure 3 – *Pyramide de résolution et sa fenêtre glissante.*

à cette zone est proportionnel à la sortie du SVM. En effet, la sortie i du classifieur SVM représente la probabilité *a posteriori* d'appartenir à la catégorie C_i.
La fin du parcours de la fenêtre glissante, sur les différentes échelles, établit des zones de votes de différentes intensités. Une phase de regroupement de ces intensités de votes permet de situer des localisations pertinentes pour la présence d'objets dans l'image.
Ainsi nous obtenons des cartes de votes qui encodent la présence des objets (cf. figure 6). Une dernière classification conclue la présence ou l'absence d'objet dans la scène.

3 Expériences et résultats

Notre protocole expérimental repose sur la base d'images proposée lors du challenge PASCAL 2005[1]. La classification des 689 images de tests doit être réalisée à partir de 684 images d'apprentissage. Les classes à étudier sont : « bicycle »,« car »,« motorbike » et « people ».
L'expérience montre que la configuration optimale est une carte ASSOM de dimension $N = 20 \times 20$ comportant $M = 2$ neurones par module. Les signatures locales des images sont extraites autours des points saillants suivant des patches de dimension 32×32. Le descripteur associé RFD comporte 8 orientations, 16 sous-régions et 3 exposants de Hölder. La taille du descripteur est donc : $16 \times 8 \times 3 = 384$.

Pour cette expérimentation, nous configurons notre architecture avec les règles suivantes pour atteindre un bon taux d'apprentissage, afin d'obtenir une représentation des données d'entrées la plus fiable possible :
- le nombre d'époque pour l'apprentissage des réseaux ASSOM est : $T = 500 \times N$;
- le taux d'apprentissage forme une fonction décroissante monotone : $\lambda(t) = \frac{T}{T+99t}$;
- la fonction de voisinage est définie par :
  $V_c^{(i)}(t) = \begin{cases} 1, ||r_c - r_i|| < \mu(t) \\ 0, sinon. \end{cases}$
  Ici, nous choisissons la norme euclidienne et r_i est la position 2D du i^{eme} neurone dans le réseau. $\mu(t)$ spécifie la largeur du voisinage décroissant linéairement avec le temps de $\frac{\sqrt{2}}{2}N$ à 0.5 .

La tâche de classification peut être jugée par une courbe ROC (*Receiver Operating Characteristic*) et son aire sous la courbe AUC (*Area Under Curve*).

Les performances de classification sont présentées par la matrice de confusion (cf. tableau 1). Les aires AUC sont présentées dans le tableau 2. La relative faiblesse de la reconnaissance de la catégorie « people » peut s'expliquer par une très grande variabilité des images à apprendre.

Quelques exemples de bonnes et mauvaises classifications sont montrées en figure 4. On peut ainsi observer sur la figure 5, que l'exemple positif « bicycle » de la figure 4 stimule fortement la carte d'activation de la catégorie « bicycle », ce qui explique ce bon résultat. De plus, on peut noter que les neurones de la carte « motorbike » sont également stimulés, ce qui montre le pouvoir de généralisation de l'approche proposée.

Tableau 1 – *Matrice de confusion. B=Bicycle, C=Car, M=Motorbike, P=People.*

Classé en →	B	C	M	P
B	82	12	14	6
C	2	243	5	13
M	3	11	199	3
P	9	17	6	52

Tableau 2 – *AUC pour chaque catégorie.*

Classes	AUC
bicycle	$0,863$
car	$0,938$
motorbike	$0,944$
people	$0,879$

Figure 4 – *Exemples de bonnes et mauvaises classifications (respectivement 1ère et 2ème ligne).*

Figure 5 – *Cartes d'activations pour un objet « bicycle ».*

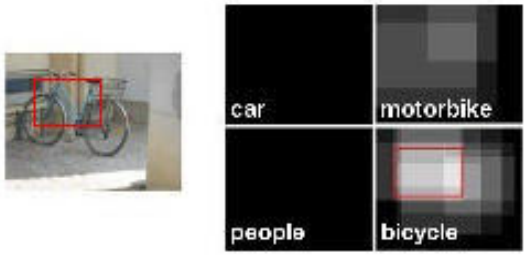


Figure 6 – *Détection et cartes de votes correspondantes.*

Cette architecture apporte ainsi un meilleur taux de classification globale ($85,08\%$) que l'utilisation d'un seul réseau ASSOM ($76,81\%$). Ce taux représente les instances correctement classées pour les quatre catégories. La compétition entre réseaux ASSOM permet ainsi d'obtenir des vecteurs caractéristiques plus discriminants pour la classification SVM[2].
Un exemple de détection d'objets avec ses cartes de votes associées est présenté par la figure 6. On distingue un vote important en intensité pour un objet de la classe « bicycle ».

4 Conclusion

Cet article propose un système original de classification d'images et de détection d'objets, utilisant l'information des singularités contenue dans les régions de forte saillance. Sur la base des trois principales propriétés des réseaux ASSOM, qui sont : la réduction de dimension, la préservation de la topologie et l'émergence de caractéristiques invariantes, notre architecture offre des résultats prometteurs au regard des résultats publiées lors du challenge PASCAL 2005[1].

Références

[1] Kohonen T. *Self-Organizing Maps*. Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[2] Hoffman J.E. et Subramanium B. The role of visual attention in saccadic eye movements. *Perception and Psychophysics*, 57 :787–795, 1995.

[3] Laurent C., Laurent N., Maurizot M., et Dorval T. In depth analysis and evaluation of saliency-based color image indexing methods using wavelet salient features. *Multimedia Tools and Application*, 2004.

[4] Harris C. et Stephens M. A combined corner and edge detector. *Proc. Fourth Alvey Vision Conf.*, pages 147–151, 1988.

[5] Bres S. et Jolion J.M. Detection of interest points for image indexation. *In 3rd Int. Conf. on Visual Information Systems*, pages 427–434, June 1999.

[6] Tversky A. Features of similarity. *Psychological Review*, 4(84) :327–352, 1977.

[7] Duda R.O., Stork D.G., et Hart P.E. *Pattern Classification*. Wiley Interscience, 2000.

[8] Shapiro J.M. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 12(41) :3345–3462, 1993.

[9] Geusebroek J.M., Boomgrad R., Smeulders W.M., et Geerts H. Color invariance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(23) :1338–1350, 2001.

[10] David G. Lowe. Object recognition from local scale-invariant features. Dans *ICCV*, pages 1150–1157, 1999.

[11] Julien Ros, Christophe Laurent, et Grégoire Lefebvre. A cascade of unsupervised and supervised neural networks for natural image classification. Dans *CIVR*, pages 92–101, 2006.

[12] Mallat S. Foveal approximations for singularities. *Applied and Computational Harmonic Analysis*, 14(2) :133–180, 2003.

[13] Zhang B., Fu M., Yan H., et Jabri M.A. Handwritten digit recognition by adaptive-subspace self-organizing map (ASSOM). *IEEE Transactions on Neural Networks*, 4(10) :939–945, 1999.

[14] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, et Gabriela Csurka. Visual categorization with bags of keypoints. Dans *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[15] Pedro Quelhas, Florent Monay, Jean-Marc Odobez, Daniel Gatica-Perez, Tinne Tuytelaars, et Luc Van Gool. Modeling scenes with local descriptors and latent aspects. Dans *IEEE Int. Conf. on Computer Vision*, 2005. IDIAP-RR 04-79.

[2]La classification SVM est ici réalisée suivant les directives du projet WEKA. `http://www.cs.waikato.ac.nz/~ml/index.html`

Accélération de surfaces actives

Julien OLIVIER Julien MILLE Romuald BONÉ Jean-Jacques ROUSSELLE

Laboratoire Informatique

Université François-Rabelais de Tours
64, avenue Jean Portalis, 37200 TOURS

{julien.olivier, julien.mille, romuald.bone, rousselle}@univ-tours.fr

Concours Jeune Chercheur : Oui

Résumé

Les contours actifs et surface actives sont des modèles déformables utilisés respectivement pour la segmentation d'images 2D et 3D. Dans cet article, nous présentons deux méthodes développées afin d'améliorer la vitesse de ces processus de segmentation d'image. Elles reposent sur des adaptations à la 3D de méthodes développées en 2D pour les contours actifs. Nous les appliquons ici sur un modèle de surface 3D discrète (maillage) dont l'évolution est guidée par l'algorithme greedy, bien qu'elles puissent s'utiliser avec d'autres types d'implémentation des contours actifs, tels les courbes de niveaux.

Mots clefs

Segmentation d'images, contours actifs, surfaces actives, accélération.

1 Introduction

Les contours actifs ou *snakes* on été à l'origine développés par Kass *et al* dans [1]. Ce sont des outils de segmentation puissants, notamment grâce à leur robustesse au bruit.
De nombreux algorithmes ont été développés pour les contours actifs. L'algorithme *greedy* introduit dans [2] demeure l'un des plus populaire de par son efficacité et sa facilité d'implémentation. Bulpitt et Efford propose dans [3] une adaptation de l'algorithme *greedy* aux surfaces 3D. De nombreuses méthodes d'accélération des contours actifs 2D ont été développées dans [4]. Nous présentons dans cette article une adaptation des ces méthodes pour accélérer les surfaces actives 3D.
Dans la section 2 nous décrirons tout d'abord le modèle des surfaces active et ses énergies. Nous verrons aussi l'algorithme d'évolution pour les surfaces actives et le principe du remaillage. Les sections 3 et 4 décrivent les deux méthodes d'accélération que nous avons adapté aux surfaces actives : la méthode du voisinage décalé et la méthode du *line search*. La section 5 décrit nos résultats expérimentaux sur des modèles 3D et la section 6 conclut sur notre travail en envisageant les développements futurs.

2 Le modèle de surface active

Dans un domaine continu, un modèle déformable 3D est représenté par une surface paramétrée S qui, à un couple de paramètres (u, v), associe un point $(x, y, z)^T$:

$$\begin{aligned} S \quad : \quad & \Omega^2 \rightarrow \mathbb{R}^3 \\ & (u, v) \longmapsto (x(u, v), y(u, v), z(u, v))^T \end{aligned} \qquad (1)$$

Afin d'implémenter la surface active, nous utilisons la représentation explicite discrète décrite dans [5]. Cette dernière est un maillage triangulaire composé de n sommets connectés, dont les arêtes forment un ensemble de triangles adjacents. Afin de représenter la notion de connectivité (la topologie locale), nous considérons que chaque sommet p_i possède un ensemble de sommets adjacents notés A_i. Le maillage est construit à partir de plusieurs divisions successives d'un icosaèdre [6, 7], menant ainsi à une surface de géométrie sphérique avec une distribution homogène des sommets. La surface possède une fonctionnelle d'énergie discrète obtenue en sommant les énergies de chaque sommet (nous décrirons les différentes énergies des sommets plus loin). La surface va évoluer de manière à minimiser cette fonctionnelle, attirant les sommets vers les bordures de l'objet à segmenter tout en conservant une stabilité géométrique. Initialement développé pour les contours 2D par Williams et Shah [2], l'algorihme *greedy* est une méthode de minimisation d'énergie proposée à l'origine comme une alternative rapide à la méthode variationnelle [1] et à la programmation dynamique [8]. Il a récemment été utilisé pour la segmentation et le suivi 2D dans [9, 10]. Dans [3], cet algorithme est étendu à la 3D pour l'évolution de maillages dans des images volumétriques.

La minimisation globale de la fonctionnelle d'énergie est réalisée par plusieurs minimisations locales successives. En effet, à chaque itération de l'algorithme un voisinage cubique de taile w autour de chaque sommet est étudié (voir fig. 1). Pour chaque voxel du voisinage, l'algorithme calcule la fonctionnelle d'énergie correspondante et affecte alors le sommet de la surface active au voxel ayant

l'énergie minimale. Dans l'algorithme *greedy* classique le fenêtre de voisinage est centrée autour de chaque sommet. Dans les méthodes que nous présentons plus loin nous autorisons cette fenêtre à ne plus être centrée.

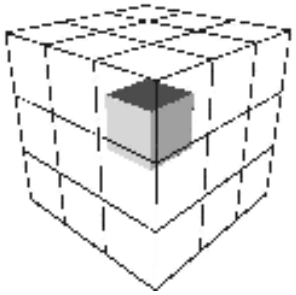


Figure 1 – *Sommet centré dans sa fenêtre de voisinage cubique* $3 \times 3 \times 3$

Afin de définir la position de chaque sommet dans son voisinage nous définissons un vecteur de décalage $\vec{\mathbf{s}}_i^{(t)} = (s_x, s_y, s_z)^T$, représentant les coordonnées du i^{ieme} sommet de la surface active à l'itération t par rapport à un voxel origine choisi dans la fenêtre de voisinage.
Au lancement de l'algorithme toutes les fenêtres de voisinage sont centrées autour de leurs sommets respectifs. Ainsi nous avons $\vec{\mathbf{s}}_i = (w/2, w/2, w/2)^T$.
Nous pouvons désormais définir le voisinage du i^{eme} sommet de la surface active à l'itération t :

$$\mathcal{N}_i^{(t)} = \left\{ \mathbf{p}_i^{(t)} + \mathbf{r} - \vec{\mathbf{s}}_i^{(t)} | \mathbf{r} \in [0, w-1]^3 \right\} \qquad (2)$$

Soit p_i la position initiale dans la fenêtre de voisinage $\mathcal{N}_i^{(t)}$, nous définissons p'_i comme la position du voxel testé par l'algorithme. Une fois que tous les voxels de $\mathcal{N}_i^{(t)}$ ont été explorés l'algorithme choisit la nouvelle position du sommet p_i :

$$\mathbf{p}_i^{(t+1)} = \arg \min_{\mathbf{p}'_k \in \mathcal{N}_i} E(\mathbf{p}'_k) \qquad (3)$$

L'énergie d'un sommet au voxel p'_i est la somme pondérée de différentes énergies internes et externes. Ces dernières sont normalisées sur le voisinage entier.

$$\begin{aligned} E(\mathbf{p}'_i) &= \alpha E_{cont}(\mathbf{p}'_i) + \beta E_{curv}(\mathbf{p}'_i) \\ &+ \gamma E_{grad}(\mathbf{p}'_i) + \delta E_{bal}(\mathbf{p}'_i) \end{aligned} \qquad (4)$$

Les paramètres (α, ..., δ) représentent les différents poids permettant de contrôler l'influence de chaque énergie. La continuité E_{cont} et la courbure E_{courb} sont les énergies internes qui permettent de garantir la stabilité géométrique de la surface. Ces deux énergies impliquent une prise en compte de la distance euclidienne et de l'ensemble de sommets adjacents. Dans un soucis de rapidité lors de l'execution, nous choisirons dans les équations suivantes de calculer la distance au carré plutôt que la distance elle même. Nous distinguons également $\|.\|$ la norme d'un vecteur de $|.|$ la valeur absolue d'un scalaire. Ainsi $\|\mathbf{p}_i - \mathbf{p}_j\|$ représente la distance euclidienne entre les sommets $\mathbf{p}_i$ et $\mathbf{p}_j$.

Nous allons maintenant décrire l'adaptation des différentes énergies à notre modèle 3D. Nos énergies sont des extensions intuitives de celles du modèle 2D, adaptées à la gestion d'un maillage.
L'énergie de continuité E_{cont} est une énergie interne permettant aux différents sommets d'être espacés de façon régulière sur la surface active. En minimisant cette énergie, l'écart entre la distance moyenne au carré $\overline{d^2}$ et la distance entre le sommet considéré p'_i et ses sommets adjacents est réduite.

$$\begin{aligned} E_{cont}(\mathbf{p}'_i) &= \sum_{j \in A_i} \left| \overline{d^2} - \|\mathbf{p}'_i - \mathbf{p}_j\|^2 \right| \\ \overline{d^2} &= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{card(A_i)} \sum_{j \in A_i} \|\mathbf{p}_i - \mathbf{p}_j\|^2 \end{aligned} \qquad (5)$$

La deuxième énergie interne utilisée est l'énergie de courbure E_{courb}. Minimiser cette dernière revient à appliquer un effet lissant sur la surface active en permettant au sommet considéré de se rapprocher du centre de gravité des sommets adjacents.

$$E_{curv}(\mathbf{p}'_i) = \left\| \mathbf{p}'_i - \frac{1}{card(A_i)} \sum_{j \in A_i} \mathbf{p}_j \right\|^2 \qquad (6)$$

Afin d'attirer la surface active vers les bords de l'objet à segmenter, nous utilisant l'énergie de gradient E_{grad}. Cette énergie externe est basée sur l'amplitude de la norme du gradient g de l'image I en chaque voxel. En cas d'image bruitée, un filtre gaussien est appliqué en amont sur l'image. Dans les équations suivantes, G_σ est un noyau gaussien d'écart type σ et $*$ est l'opérateur de convolution.

$$\begin{aligned} g(\mathbf{p}) &= \|\nabla I(\mathbf{p}) * G_\sigma\| / g_{max} \\ E_{grad}(\mathbf{p}'_i) &= -g(\mathbf{p}'_i) \end{aligned} \qquad (7)$$

Pour le calcul de la norme du gradient, nous effectuons une détection de contours 3D, en convoluant l'image avec l'opérateur de Zucker-Hummel [11], composé de trois masques de taille $3 \times 3 \times 3$. Par exemple, le masque suivant filtre l'image selon l'axe x.

$$Z_x = \begin{bmatrix} -k_1 & 0 & k_1 \\ -k_2 & 0 & k_2 \\ -k_1 & 0 & k_1 \end{bmatrix} \begin{bmatrix} -k_2 & 0 & k_2 \\ -k_3 & 0 & k_3 \\ -k_2 & 0 & k_2 \end{bmatrix} \begin{bmatrix} -k_1 & 0 & k_1 \\ -k_2 & 0 & k_2 \\ -k_1 & 0 & k_1 \end{bmatrix} \qquad (8)$$

$$k_1 = \frac{\sqrt{3}}{3} \; ; \; k_2 = \frac{\sqrt{2}}{2} \; ; \; k_3 = 1 \qquad (9)$$

Afin d'augmenter le rayon d'action de la surface active, nous utilisons une énergie ballon E_{grad} basée sur la force d'inflation présentée par Cohen dans [12].

$$E_{bal}(\mathbf{p}'_i) = \|\mathbf{p}'_i - (\mathbf{p}_i + k\vec{\mathbf{n}}_i)\|^2 \qquad (10)$$

où $\vec{\mathbf{n}}_i$ est le vecteur normal à la surface, défini au sommet p_i. Nous calculons la normale du sommet $\mathbf{p}_i$ comme la moyenne des normales des triangles voisins de $\mathbf{p}_i$. Par abus de langage, la normale d'un triangle désigne le vecteur unitaire orthogonal au plan auquel appartient le triangle. Dans les expressions suivantes, T_i désigne l'ensemble des triangles voisins de $\mathbf{p}_i$.

$$\vec{\mathbf{n}}_i = \frac{\sum_{t \in T_i} \vec{\mathbf{n}}_t}{\left\| \sum_{t \in T_i} \vec{\mathbf{n}}_t \right\|} \tag{11}$$

La normale d'un plan est déterminée par le produit vectoriel normalisé de deux vecteurs de ce plan. Ainsi, $\vec{\mathbf{n}}_t$ est calculé comme suit :

$$\vec{\mathbf{n}}_t = s_t \frac{(\mathbf{p}_{t_2} - \mathbf{p}_{t_1}) \wedge (\mathbf{p}_{t_3} - \mathbf{p}_{t_1})}{\|(\mathbf{p}_{t_2} - \mathbf{p}_{t_1}) \wedge (\mathbf{p}_{t_3} - \mathbf{p}_{t_1})\|} \tag{12}$$

où $\mathbf{p}_{t_j}, j = 1...3$ sont les sommets du triangle t ($\mathbf{p}_i$ est obligatoirement l'un d'eux). $s_t = \pm 1$ est le signe qui change l'orientation de $\vec{\mathbf{n}}_t$, pour assurer que le vecteur pointe vers l'intérieur de la surface. Orienter les normales de cette façon est nécessaire pour une implémentation correcte du ballon.

Le mouvement induit par la minimisation de l'énergie ballon est soit une dilatation soit une rétractation de la surface, selon le signe du coefficient δ. Celui-ci doit être choisi en fonction de la position initiale de la surface par rapport à l'objet.

Afin de permettre à la surface de s'adapter localement à la géométrie de l'objet à segmenter, un remaillage est effectué après chaque itération de l'algorithme *greedy*. Pour conserver la distance entre les sommets adjacents homogène, le maillage peut ajouter ou enlever des sommets, garantissant une distribution stable des sommets [7, 13, 14]. Ainsi chaque couple de sommets adjacents vérifie la contrainte :

$$d_{min} \leq \|\mathbf{p}_i - \mathbf{p}_j\| \leq d_{max} \tag{13}$$

où d_{min} et d_{max} sont deux seuils définis par l'utilisateur tels que $d_{max} \geq 2d_{min}$. Ajouter ou supprimer des sommets modifie localement la topologie de la surface, des contraintes topologiques doivent donc être vérifiées. Pour pouvoir ajouter ou supprimer des sommets, $\mathbf{p_i}$ et $\mathbf{p_j}$ doivent posséder deux sommets adjacents en commun : $|A_i \cap A_j| = 2$.
Si $\|\mathbf{p}_i - \mathbf{p}_j\| > d_{max}$ alors l'algorithme crée un nouveau sommet au milieu du segment $\mathbf{p}_i\mathbf{p}_j$ et le connecte à $\mathbf{p}_a$ et $\mathbf{p}_b$ (voir figure 2.b).
Si $\|\mathbf{p}_i - \mathbf{p}_j\| < d_{min}$, le sommet $\mathbf{p}_j$ est supprimé et $\mathbf{p}_i$ est alors déplacé au milieu du segment que formaient $\mathbf{p}_i$ et $\mathbf{p}_j$ (voir figure 2.c).

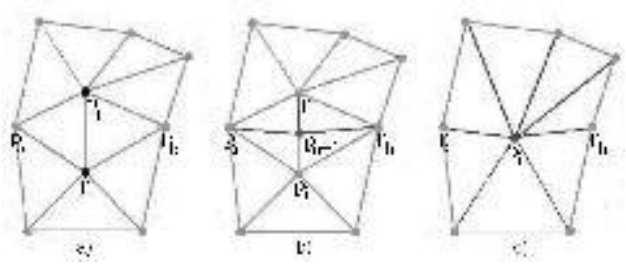


Figure 2 – *Remaillage sur la surface active : a) Maillage initial b) Ajout d'un sommet c) Suppression d'un sommet*

Maintenant que nous avons étudié l'algorithme *greedy* pour les surfaces actives, nous allons présenter les deux méthodes initialement développées en deux dimensions que nous avons adaptées aux surfaces actives 3D.

3 La méthode du voisinage décalé

La méthode du voisinage décalé a été initialement développée pour les contours actifs 2D dans [4]. Dans cette section nous allons décrire comment nous avons adapté cette méthode aux surfaces actives 3D.
A chaque itération nous allons agir sur le voisinage de chaque sommet de manière à orienter son espace de recherche dans la direction paraissant la plus intéressante. Afin de définir quelles sont ces directions, nous allons utiliser l'information déduite de l'itération précédente de l'algorithme *greedy*. En effet, grâce aux itérations précedentes nous sommes en mesure de connaître pour chaque sommet la direction qui a été suivie. Ainsi, nous allons décaler le voisinage de chaque sommet de la surface active d'un voxel dans la direction suivie précédemment. A chaque itération et pour tous les sommets, le prochain décalage à réaliser sera obtenu par :

$$\vec{\mathbf{d}}_i^{\,(t+1)} = \mathcal{B}(-1, 1, \mathbf{p}_i^{(t+1)} - \mathbf{p}_i^{(t)}) \tag{14}$$

$\vec{\mathbf{d}}_i^{(t)}$ représente le déplacement appliqué au i^{eme} sommet à l'itération t. $\mathcal{B}$ est une fonction de limitation de décalage, permettant de borner le décalage entre deux entiers :

$$\mathcal{B}(b_1, b_2, \vec{\mathbf{u}}) = \begin{pmatrix} \max(b_1, \min(b_2, u_x)) \\ \max(b_1, \min(b_2, u_y)) \\ \max(b_1, \min(b_2, u_z)) \end{pmatrix} \tag{15}$$

Connaissant le déplacement $\vec{\mathbf{d}}_i^{(t+1)}$ grâce à l'équation (14), nous pouvons définir le prochain décalage $\vec{\mathbf{s}}_i^{(t+1)}$ à appliquer à chaque sommet de la surface active :

$$\vec{\mathbf{s}}_i^{(t+1)} = \mathcal{B}(1, w-2, \vec{\mathbf{s}}_i^{(t)} - \vec{\mathbf{d}}_i^{(t+1)}) \tag{16}$$

Nous pouvons maintenant définir l'algorithme pour la méthode du voisinage décalé, qui consiste à calculer à la fin de chaque itération et pour chaque sommet la nouvelle fenêtre de voisinage avec les équations (14) (16) et (2), une fois que tous les sommets de la surface active ont été déplacés. En incluant la méthode du voisinage décalé dans l'algorithme *greedy* nous obtenons :

Algorithme 1 Méthode du voisinage décalé : modèle 3D

Pour $t \leftarrow 0$ à $T-1$ **faire**
  Pour $i \leftarrow 0$ à $n-1$ **faire**
    $\mathbf{p}_i^{(t+1)} = \underset{\mathbf{p'}_k \in N_i}{\arg\min}\, E(\mathbf{p'}_k)$
    $\vec{\mathbf{d}}_i^{(t+1)} = \mathcal{B}(-1, 1, \mathbf{p}_i^{(t+1)} - \mathbf{p}_i^{(t)})$
    $\vec{\mathbf{s}}_i^{(t+1)} = \mathcal{B}(1, w-2, \vec{\mathbf{s}}_i^{(t)} - \vec{\mathbf{d}}_i^{(t+1)})$
    $\mathcal{N}_i^{(t+1)} = \left\{ \mathbf{p}_i^{(t+1)} + \mathbf{r} - \vec{\mathbf{s}}_i^{(t)} | \mathbf{r} \in [0, w-1]^3 \right\}$
  Fin Pour
Fin Pour

4 La méthode du *line search*

La méthode du *line search* a elle aussi été développée dans [4] pour les contours actifs 2D. Nous proposons ici son adaptation aux surfaces actives. Son principe est similaire à la méthode du voisinage décalé : il s'agit d'anticiper la prochaine itération de l'algorithme *greedy* en utilisant les informations obtenues lors des précédentes itérations. La différence majeure est que nous n'allons plus modifier le voisinage de chaque sommet, mais explorer une ligne de voxels dans la direction suivie précédemment.

Cette méthode est lancée à la fin de chaque itération de l'algorithme *greedy*, une fois que tous les sommets ont été déplacés. La direction suivie par chaque sommet $\mathbf{p}_i$ est conservée et un nombre fixé de voxels est alors observée vers celle ci. L'énergie globale de chaque voxel de la ligne d'exploration est alors calculée avec l'équation (4). Le sommet courant $\mathbf{p}_i$ est alors affecté au voxel d'énergie minimale, si elle est inférieure à la sienne.

L'algorithme 2 représente l'intégration de la méthode du *line search* dans l'algorithme *greedy*. Définissons $\mathcal{T}$ comme le nombre d'itérations de l'algorithme *greedy* et l comme le nombre de voxels de la ligne d'exploration du *line search*.

Algorithme 2 Méthode du *line search* : modèle 3D

Pour $t \leftarrow 0$ à $T-1$ **faire**
  Pour $i \leftarrow 0$ à $n-1$ **faire**
    $\mathbf{p}_i^{(t+1)} = \underset{\mathbf{p'}_k \in N_i}{\arg\min}\, E(\mathbf{p'}_k)$
    Déterminer la direction $\vec{\mathbf{v}} = \frac{\mathbf{p}_i^{(t+1)} - \mathbf{p}_i^{(t)}}{\left\| \mathbf{p}_i^{(t+1)} - \mathbf{p}_i^{(t)} \right\|}$
    Line Search $d = \underset{s \in [0,l]}{\operatorname{argmin}} \left\{ E(\mathbf{p}_i^{(t)} + s\vec{\mathbf{v}}) \right\}$
    Mise à jour : $\mathbf{p}_i^{(t)} \leftarrow \mathbf{p}_i^{(t)} + dv$
  Fin Pour
Fin Pour

5 Résultats expérimentaux

Nous avons réalisé une série de tests visant à comparer l'algorithme *greedy* aux méthodes du voisinage décalé et du *line search*. Chaque image de test est composée de plusieurs coupes d'un objet en niveaux de gris. Nous avons volontairement ajouté un bruit gaussien de moyenne nulle. Plusieurs largeurs de voisinage ont été testées sur chaque image. La méthode du voisinage décalé n'a pas été testée avec un largeur $w = 3$ afin de respecter les contraintes de la méthode (le voisinage n'aurait jamais été décalé). La première image représente une spirale. Elle a été choisie afin de tester la validité de nos deux méthodes lorsque la surface évolue en dilatation avec création de sommets (le remaillage est activé). La surface active a été initialisée à l'intérieur du modèle 3D avec seulement 12 sommets. Pour les 3 méthodes le maillage final contient environ un millier de sommets. Les paramètres choisis sont $\alpha = 0, \beta = 0.5, \gamma = 2$ et $\delta \in [-1.1, -0.6]$. La deuxième image est le modèle 3D d'un vase. Celle-ci présente l'intérêt de pouvoir tester l'infiltration de la surface active dans les concavités, ce qui a toujours été un des principaux inconvénients de l'algorithme *greedy*. Les paramètres sont $\alpha = 0$, $\beta = 0.5$ pour l'algorithme *greedy*, 0.4 pour la méthode du voisinage décalé et 0.3 pour le *line search*, $\gamma = 2$ et $\delta = 0.8$. Le remaillage n'a pas été autorisé et la surface active a été initialisée avec 2562 sommets. La troisième image représente trois ellipsoïdes imbriquées et permet de tester aussi bien la reconstruction d'angles saillants que d'angles faibles. Les paramètres étaient $\alpha = 0.5$, $\beta = 0.3$ pour l'algorithme *greedy* et 0.4 pour les méthodes du voisinage décalé et du *line search*, $\gamma = 2$ et $\delta \in [0.3...0.7]$.

Chaque exécution a été réalisée sur un Intel Pentium IV 2.8Ghz avec 512Mo RAM. La figure 4 donne les résultats en termes de temps d'exécution. La figure 3 représente dans sa partie supérieure une coupe en niveaux de gris de chaque image à segmenter et dans sa partie inférieure leurs reconstructions 3D respectives obtenues avec la méthode du *line search*.

Nos deux méthodes nous ont permis de réduire de manière significative le nombre d'itérations nécessaires à l'algorithme pour segmenter les frontières de l'objet. Pour les contours actifs 2D la meilleure méthode d'accélération est la méthode du voisinage décalé. Nos résultats nous permettent d'estimer qu'en trois dimensions la méthode du *line search* est la plus performante.

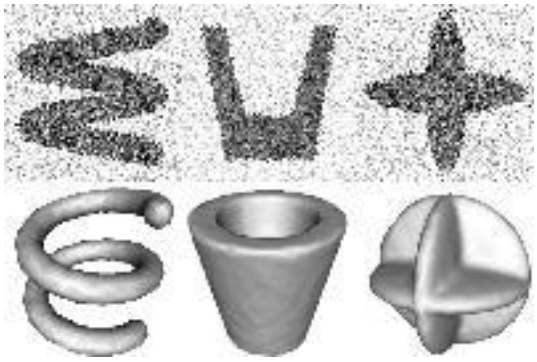


Figure 3 – *Résultats obtenus avec la méthode du line search.*

Image	Voisinage	Méthode	Itérations	Temps d'exécution (s)
Spirale	3	Greedy	400	0.3
		LS	65	0.16
	5	Greedy	195	0.52
		LS	63	0.3
		VD	138	0.31
	7	Greedy	145	0.63
		LS	60	0.57
		VD	97	1.11
3-Elipsoides	3	Greedy	47	0.63
		LS	19	0.41
	5	Greedy	25	1.05
		LS	12	0.69
		VD	16	0.98
	7	Greedy	18	2.24
		LS	12	1.86
		VD	14	2.07
Vase	3	Greedy	155	0.98
		LS	68	0.72
	5	Greedy	120	1.27
		LS	50	0.85
		VD	92	1.08
	7	Greedy	109	1.98
		LS	43	1.13
		VD	60	2.2

Figure 4 – *Comparaison de trois méthodes : algorithme greedy (greedy), voisinage décalé (VD) et line search (LS)*

En effet, le passage en trois dimensions favorise la méthode du *line search* car au lieu d'explorer un nombre fixe de pixels, on explore le même nombre de voxels donc l'augmentation du temps de calcul apportée par la 3D n'est pas significative. En ce qui concerne la méthode du voisinage décalé, les calcul de décalage ne se font plus sur une fenêtre carrée mais cubique. L'augmentation du temps de calcul par rapport à la 2D est donc beaucoup plus importante.

6 Conclusion et perspectives

Dans cet article nous avons présenté deux méthodes d'accéleration des surfaces actives qui sont sont des adaptations à la 3D de méthodes 2D existantes. En termes de temps d'exécution la méthode du *line search* apparaît comme étant la plus performante. Nous avons également testé la troisième méthode décrite dans [4] (le voisinage déformé) mais ses performances en 3D sont réduites pour les mêmes raisons que la méthode du voisinage décalé, nous n'avons donc pas jugé utile de l'inclure à cette étude.

Nous étudions actuellement le possibilité de réaliser un modèle hybride de surface active s'appuyant à la fois sur une approche physique et sur l'algorithme *greedy*. Nous pourrons alors comparer ce modèle aux algorithmes d'évolution accélérés que nous venons de décrire.

Références

[1] M. Kass, A. Witkin, et D. Terzopoulos. Snakes : active contour models. *International Journal of Computer Vision*, 1(4) :321–331, 1987.

[2] D.J. Williams et M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision and Image Processing : Image Understanding*, 55(1) :14–26, January 1992.

[3] A.J. Bulpitt et N.D. Efford. An efficient 3D deformable model with a self-optimising mesh. *Image and Vision Computing*, 14(8) :573–580, 1996.

[4] J. Olivier, R. Boné, et J.J. Rousselle. Comparison of active contour acceleration methods. Dans 5^{th} *Conf. on Visualization, Imaging & Image Processing (VIIP)*, pages 518–523, Benidorm, Spain, 2005.

[5] J. Mille, R. Boné, P. Makris, et H. Cardot. 3D segmentation using active surface : a survey and a new model. Dans 5^{th} *Conf. on Visualization, Imaging & Image Processing (VIIP)*, pages 610–615, Benidorm, Spain, 2005.

[6] T. McInerney et D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1) :69–83, 1995.

[7] J-Y. Park, T. McInerney, et D. Terzopoulos. A non-self-intersecting adaptive deformable surface for complex boundary extraction from volumetric images. *Computer & Graphics*, 25(3) :421–440, June 2001.

[8] A.A. Amini, T.E. Weymouth, et R.J. Rain. Using dynamic programming for solving variational problem in vision. *IEEE Trans. PAMI*, 12(9) :855–867, September 1991.

[9] L. Ji et H. Yan. Attractable snakes based on the greedy algorithm for contour extraction. *Pattern Recognition*, 35(4) :791–806, April 2002.

[10] C.L. Lam et S.Y. Yuen. An unbiased active contour algorithm for object tracking. *Pattern Recognition Letters*, 19(5-6) :491–498, April 1998.

[11] S.W. Zucker et R.A. Hummel. A three-dimensional edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3) :324–331, 1981.

[12] L.D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing : Image Understanding*, 53(2) :211–218, 1991.

[13] J.O. Lachaud et A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. *Medical Image Analysis*, 3(2) :187–207, 1998.

[14] G. Slabaugh et G. Unal. Active polyhedron : surface evolution theroy applied to deformable meshes. Dans *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 84–91, San Diego, 2005.

Reconstruction topologique et géométrique d'objets complexes sur grilles isothétiques irrégulières

Antoine Vacavant[1] David Coeurjolly[1] Laure Tougne[1]

[1] LIRIS - UMR 5205
Université Claude Bernard Lyon 1
43, boulevard du 11 novembre 1918
69622 Villeurbanne cedex, France

{antoine.vacavant,david.coeurjolly,laure.tougne}@liris.cnrs.fr

Concours Jeune Chercheur : Oui

Résumé

Dans cet article, nous abordons le problème de la vectorisation d'images binaires sur grilles isothétiques irrégulières. La représentation d'un objet par segments de droites a été largement développée dans le cadre de l'analyse de documents, où une image est organisée sur une grille discrète régulière. Sans considérer l'application finale, nous proposons de décrire en premier lieu la topologie d'un objet irrégulier à deux dimensions avec son graphe de Reeb associé. De plus, nous recodons cet objet avec des arcs discrets irréguliers. La seconde phase de notre algorithme consiste à réaliser une reconstruction polygonale de l'objet avec des morceaux de droites discrètes, grâce à ces arcs élémentaires. Enfin, nous présentons l'efficacité de notre méthode sur des exemples divers, puis nous discutons de ses applications futures et de son amélioration.

Mots clefs

Grilles irrégulières, géométrie discrète, topologie discrète, reconstruction.

1 Introduction

La représentation, la description et la classification de caractères et de symboles sont des tâches nécessaires dans de nombreuses applications actuelles. Elles sont appliquées sur des images généralement organisées sur des grilles régulières, $i.e.$ tous les pixels ont la même taille, et leur position est indexée de manière simple. Cependant, il est fréquent de diviser successivement une image en sous-images, $e.g.$ avec un *quadtree* [1]. Ces techniques décrivent des régions intéressantes de l'image par un ensemble de pixels irréguliers. Dans cet article, nous introduisons le concept de représentation de formes sur une *grille isothétique irrégulière* (notée $\mathbb{I}$-grille) [2]. Les pixels sont définis par des tailles et des positions variables, et peuvent être déterminées par des règles de subdivision. Nous proposons de représenter la topologie des éléments contenus dans l'image irrégulière à deux dimensions (2-D) en construisant leur graphe de Reeb associé [3]. Puis, nous les décrivons par une structure polygonale simple qui respecte le modèle de supercouverture discret étendu défini dans [2]. De plus, cette structure préserve la topologie que nous détaillons dans la phase précédente. Nous nous intéressons clairement au problème de la *vectorisation* sur grilles isothétiques irrégulières, et pas uniquement dans le cadre de l'analyse de documents. En effet, nous pouvons aussi considérer une subdivision d'une partie de $\mathbb{R}^2$ représentant les solutions d'une fonction donnée $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. Les algorithmes développés en arithmétique d'intervalles sont des approches intéressantes pour aborder ces problèmes [4, 5].

Les techniques de vectorisation développée jusqu'à maintenant sur le plan discret régulier dépendent de l'application finale de la méthode et de la culture scientifique des auteurs [6, 7]. Nous nous concentrons ici sur quelques méthologies de vectorisation, largement développées pour des applications d'analyse de document. A notre connaissance, il n'existe aucune extension de ces approches sur grilles isothétiques irrégulières. Les méthodes basées sur le *run length encoding* (RLE) construisent d'abord une décomposition en cellules allongées suivant un axe de l'image. Dès lors, on construit un graphe d'adjacence de droites (*line adjacency graph* ou LAG) [8]. Ces méthodes cherchent à décrire la topologie des objets rencontrés dans l'image, mais la structure géométrique qu'on en déduit doit être améliorée par de nombreux post-traitements. Les méthodes de squelettisation et d'amincissement sont assurément les plus souvent employées en vectorisation. Un état de l'art des méthodes de vectorisation basée sur le squelette peut être lu dans [9]. Le but est ici de calculer un axe médian de l'objet qui représente de manière minimale sa forme [10]. Néanmoins, ces techniques modifient la géométrie originale de l'objet pour obtenir une description minimale de celui-ci. De plus, elles nécessitent une phase de pré-traitement par filtrage ou par lissage pour réduire le

bruit pouvant perturber l'axe médian calculé. De manière plus générale, un objet peut contenir des trous, et peut être composé d'*arcs épais*. Dans les travaux de Debled et al. [11, 12], la définition de *segment flou* rejoint ce concept d'arc régulier épais. Mais, au-delà de cette représentation géométrique d'arcs, la structuration globale n'est pas discutée, ainsi il n'y a pas de description de la topologie des objets reconnus.
Dans cet article, nous introduisons d'abord les concepts de $k-$arcs et de $k-$objets en rappelant quelques définitions, puis nous énonçons le modèle de supercouverture étendu sur une $\mathbb{I}-$grille. Nous rappelons également le principe de la reconstruction inversible de $k-$arcs décrite dans [13]. Dans la troisième partie, nous détaillons les deux principales phases de notre système : la reconstruction homotope d'un objet complexe basée sur le graphe de Reeb [3], et sa reconstruction polygonale. Puis, nous présentons divers résultats pour illustrer les deux phases de l'algorithme. Nous montrons également la robustesse de la reconstruction polygonale par un test sur une image de dessin technique de grande taille. Nous discutons finalement des applications de notre contribution, et de l'amélioration globale de ses performances.

2 Préliminaires

Nous définissons tout d'abord une grille isothétique irrégulière, notée $\mathbb{I}$, comme un pavage du plan avec des rectangles isothétiques. Nous rappelons juste que chaque rectangle P (également appelé *cellule*) de $\mathbb{I}$ est défini par son centre $(x_P, y_P) \in \mathbb{R}^2$ et sa taille $(l_P^x, l_P^y) \in \mathbb{R}^2$. Dans notre étude, la relation d'adjacence est une notion importante que nous décrivons par les définitions suivantes.

Définition 2.1 ($ve-$adjacence et $e-$adjacence)**.** *Soit P et Q deux cellules. P et Q sont $ve-$adjacentes* (vertex and edge adjacent) *si :*

$$ou \begin{cases} |x_P - x_Q| = \frac{l_P^x + l_Q^x}{2} \; et \; |y_P - y_Q| \leq \frac{l_P^y + l_Q^y}{2} \\ |y_P - y_Q| = \frac{l_P^y + l_Q^y}{2} \; et \; |x_P - x_Q| \leq \frac{l_P^x + l_Q^x}{2} \end{cases}$$

P et Q sont $e-$adjacentes (edge adjacent) *si nous considérons un "ou" exclusif et des inégalités strictes dans la définition de $ve-$adjacence ci-dessus. Par la suite, k indique une $k-$adjacence avec $k = e$ ou $k = ve$.*

Définition 2.2 ($k-$arc)**.** *Soit $\mathcal{E}$ un ensemble de cellules, $\mathcal{E}$ est un $k-$arc si et seulement si pour tout élément de $\mathcal{E} = \{P_i, i \in \{1, ..., n\}\}$, P_i a exactement deux cellules $k-$adjacentes, sauf P_1 et P_n qui sont appelées extrémités du $k-$arc.*

Définition 2.3 ($k-$objet)**.** *Soit $\mathcal{E}$ un ensemble de cellules, $\mathcal{E}$ est un $k-$objet si et seulement si pour tout couple de cellules (P, Q) appartenant à $\mathcal{E} \times \mathcal{E}$, il existe un $k-$chemin entre P et Q dans $\mathcal{E}$.*

Nous considérons maintenant l'extension du modèle de supercouverture sur grilles isothétiques irrégulières [2] pour discrétiser des objets euclidiens sur $\mathbb{I}$.

Définition 2.4 (Supercouverture sur grilles isothétiques irrégulières)**.** *Soit F un objet euclidien dans $\mathbb{R}^2$. La supercouverture $\mathbb{S}(F)$ est définie sur une grille isothétique irrégulière $\mathbb{I}$ par :*

$$\begin{aligned} \mathbb{S}(F) &= \left\{P \in \mathbb{I} \mid \mathbb{B}^\infty(P) \cap F \neq \emptyset\right\} \\ &= \left\{P \in \mathbb{I} \mid \exists (x, y) \in F, |x_P - x| \leq l_P^x/2 \right. \\ &\quad \left. et \; |y_P - y| \leq l_P^y/2\right\} \end{aligned}$$

où $\mathbb{B}^\infty(P)$ est le rectangle centré sur (x_P, y_P) de taille (l_P^x, l_P^y) (si $l_P^x = l_P^y$, $\mathbb{B}^\infty(P)$ est la boule centrée en (l_P^x, l_P^y) de taille l_P^x pour la norme L_∞).

Nous présentons maintenant l'algorithme de reconstruction d'un $k-$arc que nous utilisons dans notre phase de représentation géométrique d'objet complexe (section 3.2). De plus, cette approche respecte le modèle de supercouverture nous venons d'énoncer. L'algorithme proposé dans [13] pour décomposer une courbe en segments est d'abord basé sur la définition suivante d'une droite discrète irrégulière.

Définition 2.5 (Droite discrète isothétique irrégulière)**.** *Soit S un ensemble de cellules dans $\mathbb{I}$, S est appelé est un morceau de droite discrète irrégulière (ou DDI) ssi il existe une droite euclidienne l telle que :*

$$S \subseteq \mathbb{S}(l)$$

En d'autres termes, S est un morceau de DDI ssi il existe l telle que pour tout P ∈ S, $\mathbb{B}^\infty(P) \cap l \neq \emptyset$.

L'algorithme inspiré de [14] utilise les procédures de construction et de mise à jour d'un *cône de visibilité*. On fixe d'abord l'extrémité p_0 du premier segment tel que $p_0 \in P_0$. On note e_0 l'arête partagée par P_0 et P_1, et on considère le premier cône $C_0(p_0, s, t)$ où s et t sont les extrémités de e_0. Puis, pour chaque cellule P_i, on considère e_i l'arête commune à P_{i-1} et P_i, et on met à jour le cône $C_j(p_j, s, t)$ si le triplet $\{p_0, s, t\}$ est ordonné dans l'ordre trigonométrique. Sinon, un nouveau cône $C_{j+1}(p_{j+1}, s, t)$ est construit, et on ajoute le point p_{j+1} à la reconstruction : les auteurs de [13] définissent p_{j+1} comme le milieu de l'intersection entre la bissectrice du cône et la cellule P_{i-1}. La figure 1 illustre la construction progressive des cônes dans un $k-$arc, et la segmentation en portions de droites en résultant.

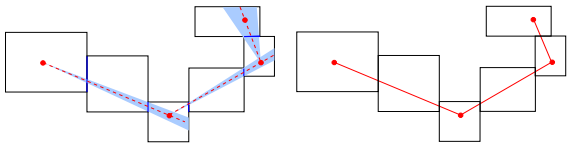


Figure 1 – *Un exemple de construction progressive de cônes dans un $k-$arc (gauche), et la reconstruction en segments que l'on obtient (droite).*

3 Représentation d'objets complexes sur grilles isothétiques irrégulières

3.1 Reconstruction topologique d'objets complexes

Pour représenter la forme d'un $k-$objet $\mathcal{E}$, nous avons choisi une approche incrémentale directionnelle. Elle permet de construire son graphe de Reeb associé G, comme dans le domaine continu (voir la figure 2). Ce graphe, basé sur la théorie de Morse [15], est aussi utilisé dans diverses applications de description de courbes et de surfaces [16, 17]. Le graphe de Reeb G est associé à une *fonction de hauteur* f définie sur $\mathcal{E}$, et les noeuds de G représentent les points critiques de f. De plus, nous voulons représenter un arc entre deux noeuds par un $k-$arc, pour avoir une information topologique minimale dans la représentation de $\mathcal{E}$. Ces arcs seront segmentés dans l'étape de description polygonale de $\mathcal{E}$ (section 3.2).

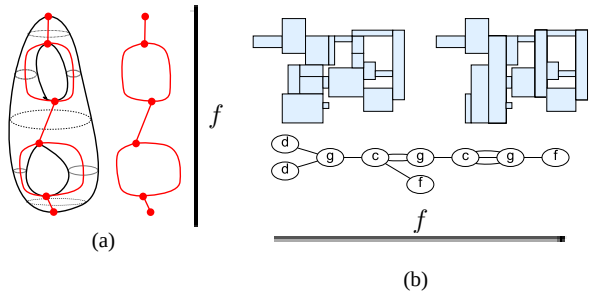


Figure 2 – *(a) : un exemple de graphe de Reeb G d'un objet continu $\mathcal{E}$. Les noeuds de G représentent les points critiques de f (maxima, minima, points d'inflexion), et un arc est une composante connexe de $\mathcal{E}$ entre deux points critiques. (b) : un exemple d'un objet irrégulier $\mathcal{E}$ (gauche), la structure finalement recodée avec des $k-$arcs (droite) et le graphe de Reeb associé à la fonction de hauteur f définie sur $\mathcal{E}$ (bas). Les noeuds d, f, g et c représentent les cellules début, fin, groupe et coupe respectivement dans le graphe.*

Nous notons les bords gauche, droit, haut et bas d'une cellule P respectivement P^L, P^R, P^T et P^B. On a, par exemple, l'abscisse de P^L égale à $x_P - (l_P^X/2)$ (que l'on note par commodité $P^L = x_P - (l_P^X/2)$). Nous dirons également qu'un $k-$arc A et une cellule P sont $k-$adjacents s'il existe une cellule Q dans A telle que P et Q sont $k-$adjacentes. Soit $\mathcal{E} = \{P_i\}_{i=0,...,n}$ un ensemble 2-D de cellules donné. On choisit en premier lieu une direction pour traiter les cellules de $\mathcal{E}$. Sans perte de généralité, on peut supposer que l'on prend l'orientation de gauche à droite, $i.e.$ la fonction de hauteur f est définie suivant l'axe des X. Au temps $t = 0$, on fusionne ensemble toutes les cellules $k-$adjacentes P de $\mathcal{E}$ avec le plus petit bord gauche $x_{t=0} = x_0$, $e.g.$ $P^L = x_0 = 0$. Cette étape de fusion est réalisée par la procédure de mise à jour décrite ci-après. Ces m collections de cellules définissent les *cellules début* des $k-$arcs initialement reconnus $A_0, A_1, ..., A_m$.

Procédure de mise à jour. Soit A un $k-$arc, et P_1, P_2 deux cellules adjacentes de $\mathcal{E}$ telles que $P_1 \in A$, $P_1^L < P_2^L$, et P_2 doit être ajoutée à A. Si $P_2^L = P_1^R$, on ajoute juste P_2 à A, sinon la procédure *met à jour* l'arc A avec P_2, et recode éventuellement A. Pour cela, on construit d'abord le *plus grand rectangle commun* F_2 de P_1 et P_2.

Définition 3.1 (Plus grand rectangle commun)**.** *Soit P_1 et P_2 deux rectangles adjacents. F_2 est le plus grand rectangle commun (ou PGRC) de P_1 et P_2 ssi*

i) $F_2 \subseteq P_1 \cup P_2$,
ii) $F_2 \cap P_1 \neq \emptyset$,
iii) $F_2 \cap P_2 \neq \emptyset$,
iv) il n'existe pas de rectangle plus grand que F_2 par inclusion respectant i), ii) and iii).

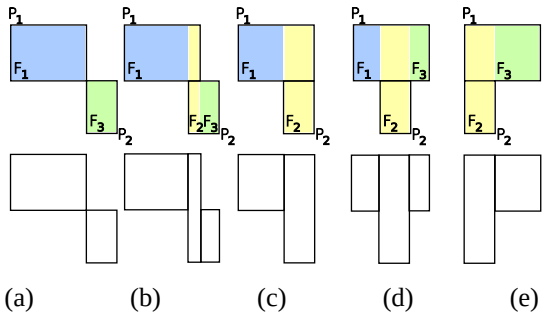


Figure 3 – *Les rectangles F_1, F_2 et F_3 dans la procédure de mise à jour (haut), et les cellules associées (bas). Lorsque $P_1^R < P_2^R$ (a et b), $P_1 - F_2 = F_1$ et $P_2 - F_2 = F_3$, sinon $P_1 - F_2 = \{F_1, F_3\}$ (d et e). Si $P_1^R = P_2^L$, $F_2 = \emptyset$, quand $P_1^R = P_2^R$, $F_3 = \emptyset$ et finalement $F_1 = \emptyset$ dans le cas où $P_1^L = P_2^L$.*

Ensuite, nous considérons les rectangles $P_1 - F_2$ et $P_2 - F_2$. Si $P_1^R < P_2^R$, on note $P_1 - F_2 = F_1$ et $P_2 - F_2 = F_3$, sinon on préférera $P_1 - F_2 = \{F_1, F_3\}$. On peut remarquer que ces rectangles peuvent être vides, $e.g.$ $F_3 = \emptyset$ si $P_1^R = P_2^R$, puisque dans ce cas $F_3^L = F_3^R$. La figure 3 présente les cinq configurations générales de la procédure de mise à jour (il en existe cinq autres, obtenues par symétrie quand $P_2^T > P_1^T$), et le recodage de l'arc que l'on doit réaliser. De plus, nous proposons de réduire le nombre de cellules dans A en joignant les deux rectangles F_1 et F_3 si $F_1^T = F_3^T$, $F_1^B = F_3^B$ et $F_2 = \emptyset$. Cette jonction est traitée en remplaçant F_1 et F_3 par le rectangle $F_1 \cup F_3$. Enfin, la procédure se termine en supprimant P_1 de A, et en ajoutant les cellules correspondant aux rectangles F_1 et F_2 à A. F_3 est empilé dans $\mathcal{E}$, et sera traité ultérieurement ; plus exactement au temps t tel que $x_t = F_3^L$.
Au temps $t + 1$, notre algorithme consiste à fusionner les cellules adjacentes avec le même bord gauche x_{t+1} en k cellules $C_1, C_2, ..., C_k$ (voir la procédure de mise à jour). Ces cellules candidates peuvent être ajoutées à un ou plusieurs arcs parmi A_i, $i \in \{1, ..., m\}$ si elles sont adjacentes à A_i. Il est clair que seule une cellule Q

construite au temps t et ayant un bord droit Q^R égal à x_{t+1} peut respecter l'adjacence avec une cellule C_j, $j \in \{1, ..., k\}$. Une cellule C_j peut être traitée de plusieurs manières :

- C_j n'est adjacente à aucun $k-$arc A_i. On initialise un nouveau $k-$arc A_{m+1} et on l'affecte avec la cellule C_j. C_j représente la *cellule début* de A_{m+1}.
- Si C_j est adjacente avec un $k-$arc A_i, alors on met à jour A_i avec C_j.
- Quand C_j est $k-$adjacent avec p $k-$arcs $A_i, A_{i+1}, ..., A_{i+p}$, c'est une *phase de regroupement*. Premièrement, on met à jour chaque arc avec C_j. La cellule C_j est ensuite marquée comme *cellule groupe* et indique que chaque arc $A_i, ..., A_{i+p}$ possède un arc $A_{m+1} = \{C_j\}$ lié comme *arc suivant*.
- Le cas où p cellules $C_j, C_{j+1}, ..., C_{j+p}$ sont $k-$adjacentes avec un arc A_i est une *phase de découpe*. On met d'abord à jour A_i avec C_j par la procédure de mis à jour. Ensuite, on note Q la cellule dans A_i telle que $Q^R = x_{t+1}$. Nous définissons aussi p nouveaux arcs suivants $A_{m+1}, ..., A_{m+p}$ de A_i tels que $A_{m+1} = \{Q, C_j\}, ..., A_{m+p} = \{Q, C_{j+p}\}$. Dans ces p arcs et dans A_i, Q est marqué comme une *cellule coupe*.

Quand l'algorithme se termine, au temps t tel que x_t est le plus grand bord gauche dans $\mathcal{E}$, on définit la dernière cellule ajoutée dans tous les arcs A_i comme une *cellule fin*. Nous illustrons dans la figure 4 la construction progressive du graphe et le recodage du $k-$objet presenté dans la figure 2 (b) dans cinq étapes de l'algorithme.

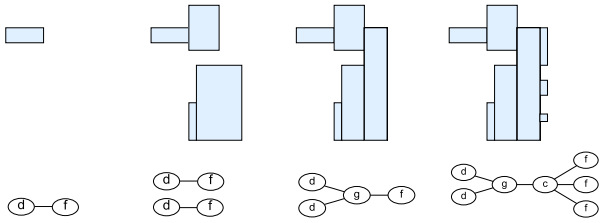


Figure 4 – *Les arcs reconnus et le graphe de Reeb associé pour quelques itérations de notre méthode sur l'objet présenté figure 2 (b). D'abord, on initialise un arc avec la cellule de plus petit bord gauche. Puis, on met à jour et on recode progressivement les arcs. La troisième et la quatrième image présentent les phases de regroupement et de découpe. On peut noter que l'étape de recodage n'est pas détaillée dans cette figure, et que les arcs $g - c$ représentent un $k-$arc avec une seule cellule dans cet exemple.*

Notre algorithme construit finalement une représentation homotope complète de $\mathcal{E}$ avec le graphe de Reeb G en reconnaissant et en liant les cellules d, g, c et f (voir également la figure 2). Il existe neuf configurations possibles d'arcs dans G : $d-c$, $d-g$, $d-f$, $c-c$, $c-g$, $c-f$, $g-c$, $g-g$ et $g-f$. Le nombre de points critiques dans f peut être lié au nombre d'Euler χ de $\mathcal{E}$ [3]. Considérons l'équation suivante, où G est noté comme le couple d'ensembles de sommets et d'arêtes (V, E) :

$$\chi = \sum_{n \in V, (n=d) \vee (n=f)} (deg(n)) - \sum_{n \in V, (n=c) \vee (n=g)} (deg(n) - 2)$$

où $deg(n)$ est le *degré* du noeud n dans G, donc $deg(n) = 1$ si n est un noeud *début* ou *fin*. Le nombre d'Euler permet de décrire la topologie d'un objet par une valeur unique. Pa exemple, pour un tore, $\chi = 0$, pour un disque, $\chi = 2$, et l'objet décrit dans la figure 2 *(b)* possède un nombre d'Euler $\chi = -4$; on peut également dire que cette forme est homéomorphe à un tore avec trois trous, où $\chi = 2 - 2 \times \#(trous) = -4$. Au-delà de ces invariants topologiques obtenus par les points critiques, la structure du graphe dépend clairement de la direction choisie pour la fonction de hauteur f. Une partie des noeuds et des arcs peut changer, mais l'information sur la topologie de $\mathcal{E}$, $i.e.$ les noeuds internes de G, n'est pas modifiée. Le nombre d'Euler est un exemple de l'emploi du graphe de Reeb pour la description de forme. Soit maintenant $\mathcal{E}'$ l'objet dessiné dans la quatrième image de la figure 4. Les trois cellules ajoutées durant la dernière itération peuvent représenter du bruit modifiant le contour de $\mathcal{E}'$. Le graphe de Reeb est changé par une phase de découpe, trois noeuds sont créés, alors que ces cellules sont peut-être nuisibles. En fait, le problème de la perturbation du contour de $\mathcal{E}'$ pourrait être certainement réduit si l'objet est d'abord filtré ou lissé. Ce genre de pré-traitement est souvent adopté, quelque soit l'approche choisie pour la représentation de forme, $e.g.$ la squelettisation. Enfin, avec la procédure de mise à jour, nous recodons les cellules de $\mathcal{E}$ afin qu'un $k-$arc soit toujours representé entre deux noeuds de G. Ce réarrangement géométrique dépend assurément de la direction de f, mais ne change ni la topologie ni le contour du $k-$arc reconnu. La structure topologique est simple, et prépare la phase suivante de notre système global de reconstruction d'objets complexes.

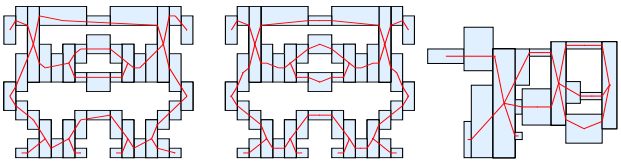
3.2 Reconstruction polygonale d'objets épais

Comme la reconstruction en polylignes affecte toujours le premier point p_0 comme le centre de la première cellule traitée (section 2), nous proposons de commencer la reconstruction de tous les $k-$arcs calculés dans la phase précédente par les noeuds *groupe* et *coupe* détectés dans le graphe de Reeb G. Nous assurons ainsi que chacun de ces noeuds particuliers de G sera représenté par un seul point dans la polygonalisation finale. Les segments sont reconnus de l'intersection entre plusieurs parties de l'objet $\mathcal{E}$ vers ses extrémités, $i.e.$ nous considérons les arcs $g-f$, $c-f$, $g-d$ et $c-d$ de G. De plus, puisque l'algorithme de reconnaissance est glouton, les éventuelles erreurs induites par notre approche de cône de visibilité sont propagées vers les extrémités de $\mathcal{E}$, et non pas vers ces intersections qui décrivent la forme de l'objet. Pour les arcs $c-c$, $g-g$, $g-c$

et $c-g$ de G, nous proposons de réaliser une reconstruction bidirectionnelle qui démarre de chaque noeud de l'arc, et se termine en son centre. Par conséquent, l'éventuelle erreur de reconstruction serait concentrée au milieu de ces arcs. Cette approche confirme que les noeuds g et c de G représentent les lieux où la description de sa géométrie doit être précise. Enfin, nous choisissons de traiter les arcs $d-f$ avec la même reconstruction bidirectionnelle, qui apparaît comme le moyen le plus efficace de garantir une reconstruction robuste.

Nous ne traitons pas le problème de liaison entre deux reconstructions sur le $k-$arc (reconstruction avec $patch$), car une technique générale et efficace de jointure entre deux droites discrètes impliquerait que notre algorithme ne serait plus linéaire [18]. Par conséquent, nous ajoutons juste un segment entre les deux polylignes. Cette phase de notre système ne peut être considérée sans patch, puisque nous utilisons les points internes de la forme de $\mathcal{E}$ pour guider la reconstruction géométrique.

Dans la figure 5, nous illustrons le comportement de notre algorithme dans le cas de l'objet $\mathcal{E}$ présenté dans la section précédente. Nous montrons également l'intérêt de notre approche pour un objet complexe symétrique.



(a) (b)

Figure 5 – *Si nous considérons l'orientation originale des $k-$arcs, la forme du $k-$object présenté dans la section suivante (a-gauche) n'est pas correctement définie puisque la symétrie n'est plus préservée. Donc, nous proposons de démarrer la reconstruction par les noeuds g et c (a-droite). Cette structure respecte le modèle de supercouverture, et la forme symétrique de cet objet. Nous montrons également le résultat de notre algorithme sur le $k-$object presenté dans la section précédente (b).*

Contrairement aux méthodes conventionnelles de vectorisation, nous proposons une technique qui respecte le modèle de supercouverture sur une $\mathbb{I}-$grille. Nous ne traitons pas de la qualité de la structure globale déduite de cette seconde étape de notre système. Pour introduire cette notion de qualité dans le cadre de l'analyse de document, on pourra se référer à [19].

4 Expériences et résultats

Dans la figure 6, nous présentons la structure polygonale obtenue sur une image réarrangée par une approche basée sur le quadtree. La reconstruction des $k-$arcs reste à l'intérieur de l'objet, et les noeuds $groupe$ et $coupe$ sont représentés par un seul point dans la reconstruction. La représentation polygonale permet aussi de mesurer des caractéristiques géométriques ($e.g.$ la longueur) d'une fonction complexe $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ (figure 7). f est d'abord discrétisée par un algorithme de calcul par intervalles [5] sous la forme d'un ensemble de cellules $\mathcal{E}$, puis nous utilisons notre système pour décrire de manière minimale les courbes de $\mathcal{E}$. Enfin, pour montrer la robustesse de notre système, nous présentons dans la figure 8 les reconstructions polygonale et topologique d'une grande image de dessin technique.

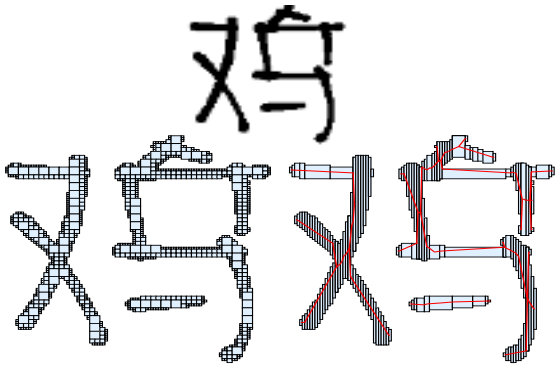


Figure 6 – *Une image d'un caractère chinois (haut), compressée par une approche basée sur un quadtree (gauche). Nous montrons le recodage final en $k-$arcs et la polygonalisation (droite).*

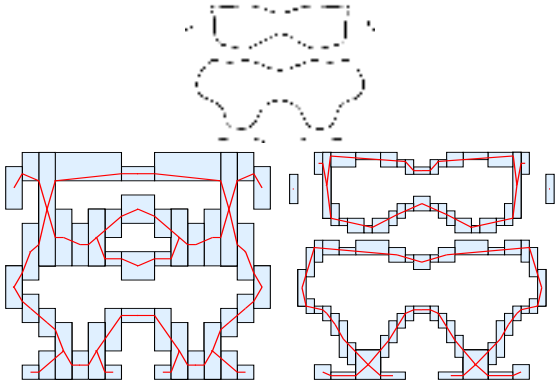


Figure 7 – *La fonction $x^2+y^2+\cos(2\pi x)+\sin(2\pi y)+\sin(2\pi x^2)\cos(2\pi y^2) = 1$ sur $[-1;1]\times[-1;1]$ (haut) discrétisée par un algorithme décrit dans [5] avec deux résolutions différentes, puis recodée et polygonalisée (bas).*

5 Bilan et perspectives

Dans cet article, nous proposons d'élargir le point de vue des méthodologies de vectorisation à la représentation isothétique irrégulière des données binaires. Selon l'application finale de notre système, nous pouvons traiter l'image initiale avec des opérations de pré-traitement, réorganiser le graphe de Reeb (contraction d'arcs, etc.), ou réarranger les segments finalement calculés dans la seconde phase. La reconstruction géométrique reste

à l'intérieur de l'objet, $i.e.$ elle respecte le modèle de supercouverture discrète irrégulière. De plus, cette reconstruction préserve la topologie décrite par le graphe de Reeb. Donc, notre système est robuste, et topologiquement et géométriquement correct. Le graphe de Reeb peut être étendu à la description d'objets en trois dimensions (3-D), avec une approche incrémentale similaire. Cependant, la reconstruction basée sur le cône de visibilité est difficilement adaptable à de tels objets irréguliers. Notre système devrait être modifié pour fournir une polygonalisation 3-D basée sur le graphe de Reeb.

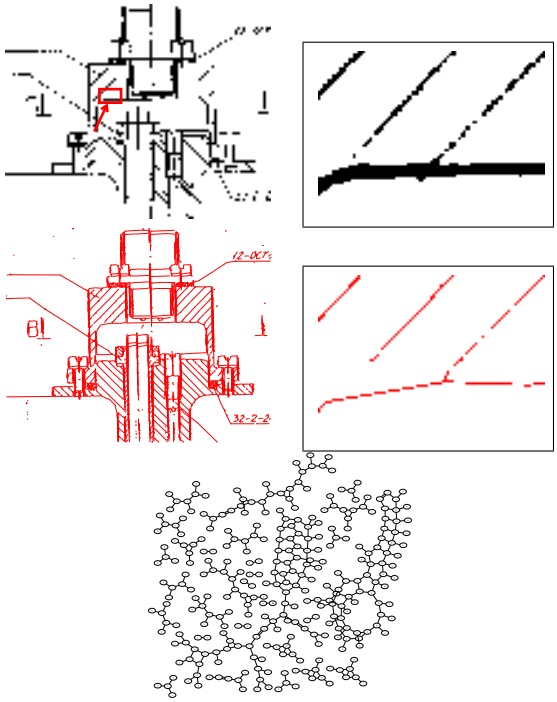


Figure 8 – *Une image de dessin technique de taille 1765 x 1437 pixels soumise à notre système, et une partie zoomée, indiquée par la flèche (haut). La polygonalisation que nous obtenons et le zoom associé sont présentés (centre). Le graphe de Reeb complet (environ 300 noeuds) est également illustré dans un format circulaire (bas).*

Références

[1] H. Samet. The quadtree and related hierarchical data structures. *ACM Computer Survey*, 16(2) :187–260, 1984.

[2] D. Coeurjolly. Supercover model and digital straight line recognition on irregular isothetic grids. Dans *12th International Conference on Discrete Geometry for Computer Imagery*, pages 311–322. LNCS 3429, 2005.

[3] G. Reeb. Sur les points singuliers d'une forme de pfaff complément intégrable ou d'une fonction numérique. *Comptes Rendus de L'Académie des Sciences*, pages 847–849, 1946.

[4] B. Kearfott. Interval computations : introduction, uses, and resources. *Euromath Bulletin*, 2(1) :95–112, 1996.

[5] J.M. Snyder. Interval analysis for computer graphics. *Computer Graphics*, 26(2) :121–130, juillet 1992.

[6] L.P. Cordella et M. Vento. Symbol recognition in documents : a collection of techniques ? *International Journal on Document Analysis and Recognition*, 3(2) :73–88, 2000.

[7] X. Hilaire et K. Tombre. Robust and accurate vectorization of line drawings. Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[8] M. Burge et W.G. Kropatsch. A minimal line property preserving representation of lines images. *Computing*, 62 :355–368, 1999.

[9] L. Lam, S.W. Lee, et Suen C.Y.. Thinning methodologies - a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9) :869–885, 1992.

[10] R. Klette et A. Rosenfeld. *Digital geometry*. Elsevier, San Fransisco, 2004.

[11] I. Debled, S. Tabbone, et L. Wendling. Fast polygonal approximation of digital curves. Dans *International Conference on Pattern Recognition*, volume 1, pages 465–468, Cambridge, United Kingdom, aout 2004.

[12] I. Debled, F Feschet, et J. Rouyer-Degli. Optimal blurred segments decomposition in linear time. Dans *12th International Conference on Discrete Geometry for Computer Imagery*, pages 311–322. LCNS 3429, 2005.

[13] D. Coeurjolly et L. Zerarga. Supercover model, digital straight line recognition and curve reconstruction on the irregular isothetic grids. *Computer and Graphics*, 30(1) :46–53, 2006.

[14] I. Sivignon, R. Breton, F. Dupont, et E. Andres. Discrete analytical curve reconstruction without patches. *Image and Vision Computing*, 23(2) :191–202, 2005.

[15] A. Gramain. *Topologie des surfaces*. Presses Universitaires Françaises, 1971.

[16] F. Hétroy. *Méthodes de partitionnement de surfaces*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, France, septembre 2003.

[17] T. Tung. *Indexation 3D de bases de données d'objets 3D par graphes de Reeb améliorés*. Thèse de doctorat, Telecom Paris, ENST/TIC, Paris, France, juin 2005.

[18] R. Breton. *Reconstruction inversible d'objets discrets 2D*. Thèse de doctorat, Université de Poitiers, Poitiers, France, décembre 2003.

[19] L. Wenyin et D. Dori. A protocol for performance evaluation of line detection algorithms. *Machine Vision and Applications*, 9(5/6) :240–250, 1997.

Régularisation sur graphe pour le traitement d'images couleur

O. Lezoray A. Elmoataz S. Bougleux
[1] LUSAC EA 2607, IUT SRC, 120 Rue de l'exode, 50000 SAINT-LÔ
[2] GREYC CNRS UMR 6072, Groupe Image, ENSICAEN, 6 Bd. Maréchal Juin, F-14050 CAEN
olivier.lezoray@unicaen.fr,{abder.elmoataz,sebastien.bougleux}@greyc.ensicaen.fr

Résumé

Le traitement d'images couleur est un problème important en vision par ordinateur et les formulations variationnelles fournissent un cadre formel pour celui-ci. Les solutions de modèles variationnels peut être obtenues en minimisant des fonctions d'énergies appropriées et cette minimisation est habituellement réalisée à l'aide d'équations aux dérivées partielles (EDP). Le problème est considéré comme un problème de régularisation. Dans cet article, nous proposons un cadre général de régularisation discrète opérant sur des graphes pondérés de topologie arbitraire, ceci peut être vu comme un analogue discret de la théorie classique de regularisation. A l'aide de cette formulation nous proposons une famille de filtres linéaires et non linéaires anisotropes simples et rapides qui ne nécéssitent pas d'EDP. L'approche que nous proposons peut être utile pour des problèmes de restauration, de simplification ou bien de segmentation.

Mots clefs

Régularisation, couleur, graphe, simplification, segmentation.

1 Introduction

Beaucoup d'approches existent pour le traitement d'images couleur et les modèles variationnels ont été extremement utilisés pour une grande variété de problèmes de vision par ordinateur. Les solutions de modèles variationnels peuvent être obtenues en minimisant des fonctions d'énergies appropriées et cette minimisation est habituellement réalisée à l'aide d'équations aux dérivées partielles (EDP). Les EDP sont définies en continu et lorsque l'existence et l'unicité d'une solution ont été prouvées, elles sont discrétisées afin d'obtenir une solution numérique. Soit $f : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ une image (nous supposons que Ω est un rectangle de $\mathbb{R}^2$) et soit f^0 une version dégradée (bruitée) de f :

$$f^0 = Lf + \eta \tag{1}$$

L est un opérateur linéaire représentant le flou et η est un bruit additif de variance σ^2. Pour restaurer l'image (le but est de reconstruire f à partir de f^0), nous devons résoudre (1) pour f qui est un problème inverse mal posé [1]. Ceci peut être résolu par une approche variationnelle et trouver f^0 peut être formulé comme la minimisation d'une énergie de la forme suivante, où $\lambda \geq 0$ est un multiplicateur de Lagrange [2],

$$E(f, f^0, \lambda) = E_{smooth}(f) + \lambda E_{fidelity}(f, f^0) \tag{2}$$

Cette idée a été initialement introduite par Tikhonov [3]. E_{smooth} est un terme de régularisation, $E_{fidelity}$ est un terme d'attache aux données et λ est un poids qui contrôle le ratio entre les deux termes. Dans la suite, nous nous restreignons à la famille suivante de problèmes variationnels ($p \in \{1, 2\}$) :

$$\min_{f \in \mathcal{H}} \left\{ E_p(f, f^0, \lambda) = \int_\Omega \left(\|\nabla f\|^p + \lambda \|f - f^0\|^2_{\mathcal{H}} \right) \right\} \tag{3}$$

Le terme de régularisation dans E_p mesure la p-régularité de f sur le domaine Ω de l'image. Le terme de fidélité est la norme au carré dans un espace de Hilbert donné $\mathcal{H}$. L'image régularisée f qui minimise (3) peut être obtenue par une descente de gradient avec l'équation d'Euler-Lagrange de E_p. L'EDP résultante est discrétisée et résolue par des algorithmes de calculs numériques. Les EDP ont été très utilisées ces dernières années (voir dans [4] pour une revue très complète) et se différencient par l'utilisation ou non d'un terme d'attache aux données. Parmi les travaux les plus connus citons ceux de Perona et Malik [5] pour la diffusion non linéaire basée uniquement sur un terme de régularisation et la variation totale (TV variation) proposée par Rudin, Osher et Fatemi [6] basée sur la régularisation et l'attache aux données afin de préserver les contours. Dans (3), quand $p = 1$ et $\mathcal{H} = L^2$, E_{smooth} est la variation totale (TV), et E_1 correspond au modèle non linéaire de Rudin, Osher et Fatemi (ROF). Quand $p = 2$, E_2 correspond à la régularisation de Tikhonov. Nous n'effectuerons pas ici de revue des modèles variationnels car ceci est hors de la portée de cet article, le lecteur intéressé se réfèrera à [4] et à [7, 8, 9] pour le cas de la couleur. Les modèles variationnels ont été étudiés d'un point de vue théorique et pratique [4]. Ils ont cependant un certain nombre de désavantages. Tout d'abord ils sont définis en continu et leur résolution nécéssite une discrétisation. Or le domaine de l'image est par nature discret, et peut se représenter par des graphes réguliers ou de topologie arbitraire, sur lesquels les méthodes classiques ne fonctionnent pas (des nuages de points par exemple). Deuxièmement les EDP ont un comportement extrèmement local et même avec des schémas numériques

efficaces, les algorithmes les mettant en oeuvre sont relativement lents. Troisièmement les schémas numériques mis en oeuvre pour la discrétisation des EDP nécéssitent que les données à régulariser aient une répartition selon une grille et les EDP ne sont donc pas adaptées pour effectuer la régularisation de données arbitraires, par exemple dans le cas où celles-ci sont représentées par des graphes de topologies arbitraires : la régularisation en continu n'est pas directement appliquable. Les méthodes discrètes semblent donc plus appropriées que les EDP dans certains cas. L'idée est de considérer la régularisation discrète d'image sur des graphes, qui peut se réduire à la résolution de systèmes linéaires ou non-linéaires par des méthodes itératives. Nous transcrivons donc le cadre de la regularisation continue par EDP dans le domaine discret défini sur des graphes de topologies arbitraires. La minimisation discrète est un problème analogue au problème (3) et est formalisée par ($p \in \{1, 2\}$) :

$$\min_{f \in \mathcal{H}(V)} \left\{ E_p(f, f^0, \lambda) = \sum_{v \in V} \|\nabla f(v)\|^p + \lambda \|f - f^0\|^2_{\mathcal{H}(V)} \right\} \quad (4)$$

où V est un ensemble de noeud, $\mathcal{H}(V)$ est un espace de Hilbert défini sur V et $\|\nabla f(v)\|$ est la norme d'un gradient discret donné. Le principal avantage de notre approche est que nous n'avons pas à résoudre une EDP et nous pouvons appliquer notre méthode sur n'importe quel type de données pour peu qu'elles puissent être représentées sous forme de graphes dont la topologie peut être arbitraire. Nous proposons donc un formalisme général de régularisation discrète sur graphes. La régularisation discrète sur graphe a déjà été utilisée sur des graphes réguliers (TV Digital filter [10]) pour le traitement d'images et sur des graphes arbitraires pour la classification semi-supervisée [11]. En nous basant sur ces travaux, nous proposons un cadre général pour la régularisation opérant sur des graphes pondérés de topologies arbitraires qui mène à une famille de filtres anisotropiques linéaires et non-linéaires simples et rapides. Nous rappellons tout d'abord les définitions usuelles sur les graphes puis nous définissons une géométrie différentielle sur ceux-ci. Enfin nous proposons un cadre général pour la régularization discrète sur graphes et nous montrons comment ceci peut être appliqué à différents problèmes de traitement d'images couleur.

2 Graphes pondérés

Un graphe est une structure de données utilisée afin de représenter un ensemble et les relations deux à deux entre des éléments de cet ensemble. Les éléments de l'ensemble sont appellés des noeuds et les relations des arêtes. Nous rappellons quelques éléments concernant la théorie des graphes [12]. Un graphe $\mathcal{G}$ est un couple $\mathcal{G} = (V, E)$ où V est un ensemble de noeuds et $E \subseteq V \times V$ un ensemble d'arètes. Deux noeuds u et v sont adjacents si l'arête $(u, v) \in E$, une arête $e \in E$ est incidente à un noeud u si u est une de ses extrémités. Un chemin p est un ensemble de noeuds $p = (v_1, v_2, \cdots, v_k)$ tel qu'il existe une arête entre deux noeuds sucessifs du chemin ($\forall i \in [1, k[$, l'arête $(v_i, v_{i+1}) \in E$). Un chemin est simple si chaque arête n'est parcourue qu'une seule fois. Un cycle $c = (v_1, \cdots, v_k)$ est un chemin simple terminant en son noeud de départ ($v_1 = v_k$). Un graphe est connexe ssi il existe un chemin entre chaque paire de sommets. Un graphe est non orienté quand l'ensemble des arêtes est symmétrique : à chaque arête $(u, v) \in E$ correspond une arête $(v, u) \in E$. Dans tout le reste de cet article nous ne considérerons que des graphes simples pour lesquels maximum une arête peut joindre deux noeuds. Ces graphes sont supposés être connexes, non orientés et sans cycles. Ces graphes peuvent être pondérés si une fonction de pondération w est associée à chaque arête, $w : E \to \mathbb{R}^+$ satisfaisant $w(u, v) > 0$ si $(u, v) \in E$, $w(u, v) = 0$ si $(u, v) \notin E$ et $w(u, v) = w(v, u)$. Une fonction de pondération mesure donc la similarité entre deux noeuds. Le degré d'un noeud $\delta : V \to \mathbb{R}^+$ est défini comme $\delta(v) = \sum_{u \sim v} w(u, v)$ où $u \sim v$ désigne l'ensemble des noeuds u adjacents au noeud v. Nous pouvons maintenant définir l'espace des fonctions sur graphes. Soit $\mathcal{H}(V)$ l'espace de Hilbert des fonctions à valeurs réelles sur les noeuds dans lequel chaque fonction $f : V \to \mathbb{R}^+$ attribue une valeur réelle $f(v)$ à chaque noeud v. L'espace des fonctions $\mathcal{H}(V)$ peut être muni du produit scalaire usuel $\langle f, g \rangle_{\mathcal{H}(V)} = \sum_{v \in V} f(v)g(v)$ où f et g sont deux fonctions de $\mathcal{H}(V)$. Une fonction f de $\mathcal{H}(V)$ peut être vue comme un vecteur de $\mathbb{R}^{|V|}$. La norme d'une fonction f obtenue à partir du produit scalaire est $\|f\| = \sqrt{\langle f, f \rangle}$. De même, nous pouvons définir $\mathcal{H}(E)$ l'espace des fonctions à valeurs réelles sur les arêtes dans lequel chaque fonction $h : E \to \mathbb{R}^+$ attribue une valeur réelle à chaque arête e. Cet espace de fonctions peut être muni du produit scalaire usuel $\langle h, l \rangle_{\mathcal{H}(E)} = \sum_{(u,v) \in E} h(u, v)l(u, v)$ où h et l sont deux fonctions de $\mathcal{H}(E)$. Les graphes sont des structures de données très utiles pour représenter des données provenant d'images. Une image est une grille discrète à laquelle on peut associer un graphe, les arêtes dépéndant de la connexité considérée entre les pixels (4 où 8 connexité) : c'est le plus bas niveau de représentation d'une image par un graphe. Des représentations de plus haut niveau, sous forme de graphe, de données provenant d'images sont par exemple le graphe d'adjacence de régions ou le graphe d'adjacence de couleurs [13]. Un graphe d'ajacence de régions est le graphe associé à une partition d'une image (la connexité spatiale des régions est considérée). A chaque représentation d'une image couleur on peut donc associer un graphe et nous considérons le traitement d'images couleur comme celui de graphes dont les modèles de noeuds et d'arêtes sont liés aux propriétés colorimétriques de l'image. La topologie du graphe dépend du problème que l'on veut résoudre : des grilles pour la restauration, des graphes d'adjacence de régions pour la simplification et la segmentation, des pyramides irrégulières pour les hiérarchies de partitions et des graphes

d'adjacence de couleurs pour la coalescence. La topologie des graphes considérés est donc arbitraire mais le principe de la régularisation discrète que nous proposons restera le même.

3 Géométrie Différentielle sur Graphes

Nous avons introduit l'espace des fonctions sur graphes et dans cette section nous présentons l'expression d'une géométrie différentielle sur graphes [11, 14, 15] qui est une discrétisation de la géométrie différentielle classique en continu.

3.1 Gradient et divergence

Dans cette section nous présentons le gradient et la divergence sur graphes. L'opérateur de différence $d : \mathcal{H}(V) \rightarrow \mathcal{H}(E)$ sur $\mathcal{G} = (V, E)$ d'une fonction $f \in \mathcal{H}(V)$ sur une arête (u, v) liant deux noeuds u et v est définie $\forall (u, v) \in E$ par

$$(df)(u, v) = (df)_{uv} = \sqrt{w(u, v)}\,(f(v) - f(u)) \quad (5)$$

La dérivée $\left.\frac{\partial f}{\partial e}\right|_v : \mathcal{H}(V) \rightarrow \mathbb{R}^+$ d'une fonction f au noeud v selon une arête $e = (u, v)$ est définie par

$$\partial_v f(u) = \left.\frac{\partial f}{\partial e}\right|_v = (df)(u, v)$$

Pour une fonction $f \in \mathcal{H}(V)$ et un noeud v, le gradient de f au noeud v est l'opérateur vectoriel défini par $\nabla : V \rightarrow \mathbb{R}^N$

$$\nabla f(v) = \nabla_v f = (\partial_v f(u) : (u, v) \in E, u \sim v)^T \quad (6)$$

La norme du gradient sur graphe ∇f au noeud v est défini par $\|\nabla\| : \mathbb{R}^N \rightarrow \mathbb{R}^+$

$$\|\nabla_v f\| = \sqrt{\sum_{u \sim v} (\partial_v f(u))^2} = \sqrt{\sum_{u \sim v} w(u, v)\,(f(v) - f(u))^2} \quad (7)$$

La norme du gradient mesure la régularité d'une fonction autout d'un noeud. Soit $\mathcal{R}$ une fonctionnelle sur $\mathcal{H}(V)$, pour chaque $p \in [1, +\infty)$, définie par $\mathcal{R}_p(f) = \sum_{v \in V} \|\nabla_v f\|^p$. Cette fonctionnelle $\mathcal{R}_p$ peut être vue comme une mesure de régularité de f puisque c'est la somme des variations locales en chaque noeud. La divergence sur graphe est un opérateur $div : \mathcal{H}(E) \rightarrow \mathcal{H}(V)$ qui satisfait

$$\langle df, h\rangle_{\mathcal{H}(E)} = \langle f, -div(h)\rangle_{\mathcal{H}(V)}$$

avec $f \in \mathcal{H}(V)$ et $h \in \mathcal{H}(E)$. Cet opérateur $-div$ est l'opérateur adjoint d^* de l'opérateur de difference d. A partir de la définition des produits scalaires dans $\mathcal{H}(V)$ et $\mathcal{H}(E)$ et l'équation (5), on peut prouver que la divergence sur graphe d'une fonction $h \in \mathcal{H}(E)$ au noeud v peut s'exprimer comme

$$(-div(h))(v) = \sum_{u \sim v} \sqrt{w(v, u)}\,(h(u, v) - h(v, u)) \quad (8)$$

3.2 Le p-Laplacien

La laplacien sur graphe peut être vu comme un analogue discret de l'opérateur de Laplace-Beltrami pour des variétés Riemanniennes. Le laplacien sur graphe est un opérateur $\Delta : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ défini par $\Delta f = -div(df) = d^*(df)$. La laplacien sur graphe est un opérateur linéaire car le gradient et la divergence sont tous deux linéaires. En outre il est auto adjoint :

$$\langle \Delta f, g\rangle_{\mathcal{H}(V)} = \langle df, dg\rangle_{\mathcal{H}(E)} = \langle f, \Delta g\rangle_{\mathcal{H}(V)}$$

et semi-défini positif :

$$\langle \Delta f, f\rangle_{\mathcal{H}(V)} = \langle df, df\rangle_{\mathcal{H}(E)} = \mathcal{R}_p(f) \geq 0$$

ce qui implique que

$$\Delta f = \frac{\partial \mathcal{R}_p(f)}{\partial f} \quad (9)$$

La courbure sur graphe est un opérateur non linéaire $\kappa : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ défini par $\kappa f = -div\left(\frac{df}{\|\nabla f\|}\right) = d^*\left(\frac{df}{\|\nabla f\|}\right)$. Nous pouvons généraliser le laplacien et la courbure sur graphe en un opérateur qui peut être vu comme un analogue discret du p-Laplacien. Le p-Laplacien sur graphe est un opérateur $\Delta_p : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ avec $p \in [1, +\infty)$ défini par

$$\Delta_p f = -div\left(\|\nabla f\|^{p-2}\, df\right) = d^*\left(\|\nabla f\|^{p-2}\, df\right) \quad (10)$$

Clairement nous avons $\Delta_1 = \kappa$ et $\Delta_2 = \Delta$. En substituant (5) et (8) dans la définition (10) de $\Delta_p f$, nous obtenons :

$$(\Delta_p f)(v) = \sum_{u \sim v} \gamma(u, v)(f(v) - f(u)) \quad (11)$$

où $\gamma(u, v)$ est la fonction définie par

$$\gamma(u, v) = w(u, v)\left(\|\nabla f(v)\|^{p-2} + \|\nabla f(u)\|^{p-2}\right) \quad (12)$$

ce qui généralise le laplacien et la courbure classiques sur graphes. En général Δ_p est non linéaire (sauf dans le cas où $p = 2$) et est positif semi-défini. Afin déviter d'avoir un dénominateur à zéro lors du calcul de la courbure, le gradient sur graphe $\|\nabla_v f\|^{p-2}$ est remplacé par sa version régularisée $\left(\epsilon + \|\nabla_v f\|^{p-2}\right)$ avec $\epsilon \rightarrow 0$.

4 Régularisation sur graphes

Dans cette section nous proposons un cadre général pour la régularisation sur graphes, un graphe étant une réprésentation discrète de données dont la dimensionnalité et la topologie sont arbitraires. Dans un soucis de clarté, nous présentons ce cadre pour le cas des images scalaires mais le principe reste le même pour des images couleur. Dans ce cas une image couleur f est composée de trois composantes f_1, f_2, f_3 et la régularisation est appliquée sur chaque composante indépendemment.

4.1 Cadre général

Etant donné un graphe $\mathcal{G} = (V, E)$ auquel est associé une fonction de pondération $w : E \to \mathbb{R}^+$, nous désirons effectuer la p-régularisation d'une fonction $f^0 \in \mathcal{H}(V)$ en utilisant le p-Laplacien. Cela consiste à chercher une fonction $f^* \in \mathcal{H}(V)$ qui est non seulement suffisamment lisse sur $\mathcal{G}$ mais également suffisamment proche de la fonction f. Ceci peut être formalisé par le problème d'optimisation suivant comme la minimisation d'une somme pondérée de deux énergies :

$$f^* = \min_{f \in \mathcal{H}(V)} \left\{ E_p = \mathcal{R}_p(f) + \lambda \left\| f - f^0 \right\|^2 = \sum_{v \in V} \|\nabla_v f\|^p + \lambda \sum_{v \in V} \left\| f - f^0 \right\|^2 \right\} \tag{13}$$

Le premier terme est le terme de régularité qui impose que f ne change pas trop entre des noeuds proches. Le second terme est l'attache aux données qui impose que f ne soit pas trop éloignée de f^0. $\lambda \geq 0$ est un paramètre de fidélité aux données initiales qui spécifie le compromis entre ces deux termes d'énergie. Les deux termes de l'énergie E_p sont des fonctions strictement convexes de f et ce problème a une solution unique pour $p = 1$ or 2 qui satisfait :

$$\left. \frac{\partial E_p}{\partial f} \right|_v = 0, \forall v \in V \tag{14}$$

En utilisant la propriété (9) du p-Laplacien pour calculer la dérivée du premier terme de E_p, l'équation (14) peut être réécrite ainsi :

$$(\Delta_p f^*)(v) + 2\lambda \left(f^*(v) - f^0(v) \right) = 0, \forall v \in V \tag{15}$$

La solution f du probleme (13) est aussi la solution de (15). En substituant l'expression du p-Laplacien dans (15), on obtient, $\forall v \in V$:

$$\left(2\lambda + \sum_{u \sim v} \gamma_{uv} \right) f^*(v) - \sum_{u \sim v} \gamma(u, v) f^*(u) = 2\lambda f^0(v) \tag{16}$$

Parmi toutes les méthodes de résolution de (16), nous utilisons l'algorithme itératif de Gauss-Jacobi. Soit t le numéro d'itération et $f^{(t)}$ la solution de (16) à l'itération t. La fonction initiale $f^{(0)}$ peut être initialisée avec f^0. L'itération correspondante est donnée par, $\forall v \in V$:

$$f^{(t+1)}(v) = \frac{2\lambda}{2\lambda + \sum_{u \sim v} \gamma^{(t)}(u, v)} f^0(v) + \frac{\sum_{u \sim v} \gamma^{(t)}(u, v) f^{(t)}(u)}{2\lambda + \sum_{u \sim v} \gamma^{(t)}(u, v)} \tag{17}$$

où $\gamma^{(t)}$ est la fonction $\gamma(u, v)$ à l'itération t. On notera que la valeur de $f(v)$ à une itération donnée $(t + 1)$ dépend de deux quantités : la valeur originale de f en v (soit $f^0(v)$) et les valeurs à l'itération t dans le voisinage de v. Ces quantités sont pondérées par des poids dont les valeurs dépendent de la somme des variations locales. L'algorithme complet pour calculer la solution de (13) prend en entrée un graphe $\mathcal{G} = (V, E)$, la fonction de pondération w, le paramètre λ, le degré de régularité p, la fonction initiale $f^0 = f^{(0)}$ et le nombre d'itérations i. Tout d'abord, les poids sont initialisés pour chaque arête de E. Ensuite, pour chaque itération $t = 0, \ldots, i$:

(a) pour chaque $v \in V$, $f^{(t+1)}(v)$ est calculé selon (17),

(b) pour chaque arête $(u, v) \in E$, la fonction $\gamma^{(t+1)}(u, v)$ est mise à jour selon (12).

Ce filtre est non linéaire à part pour $p = 2$ qui est linéaire et dont les coefficients γ n'ont pas à être mis à jour à chaque itération. Dans ce cas et si $\lambda \neq 0$, la solution optimale peut être obtenue en répétant (a) jusqu'à convergence, c'est à dire jusqu'à ce que $|f^{(t+1)} - f^{(t)}| < \epsilon$, avec $\epsilon \to 0$.

4.2 Régularisation pour $p = 2$

Quand $p = 2$, en utilisant (15) on en déduit que la solution de (13) est basée sur le Laplacien et satisfait

$$\Delta f^* + 2\lambda(f^* - f^0) = 0 \tag{18}$$

Ceci peut être vu comme un analogue discret de l'equation d'Euler-Lagrange. Dans ce cas le filtre de la section 4.1 est linéaire et si $\lambda \neq 0$, il converge vers la solution de (13). Le schéma itératif utilisé pour résoudre ceci est exprimé par, $\forall v \in V$:

$$\begin{cases} f^0 = f \\ f^{(t+1)}(v) = \frac{2\lambda}{2\lambda + \sum_{u \sim v} w(u,v)} f^0(v) + \frac{\sum_{u \sim v} w(u,v) f^t(u)}{2\lambda + \sum_{u \sim v} w(u,v)} \end{cases} \tag{19}$$

où t indique le numéro de l'itération. Nous définissons la fonction $c : V \to \mathbb{R}^+$ par $c(v) = \frac{1}{2\lambda + \sum_{u \sim v} w(u,v)}$ et le schéma itératif s'exprime alors ainsi, $\forall v \in V$:

$$f^{(t+1)}(v) = 2\lambda c(v) f^0(v) + c(v) \sum_{u \sim v} w(u, v) f^t(u) \tag{20}$$

On peut constater qu'à chaque itération, la nouvelle valeur d'un noeud est obtenue à partir des valeurs pondérées de ses voisins et de la sienne. Le filtre correspondant est un filtre passe-bas dont le comportement s'adapte à l'image à traiter grâce aux valeurs de $w(u, v)$ caculées à partir de f^0. Quand $\lambda = 0$ et $w(u, v) = 1 \; \forall (u, v) \in E$, c'est l'analogue discret de la diffusion sur des variétés Riemmanniennes [7].

4.3 Régularisation pour $p = 1$

Quand $p = 1$, en utilisant (15) on en déduit que la solution de (13) est basée sur l'opérateur non linéaire de courbure κ et satisfait

$$\kappa f^* + 2\lambda(f^* - f^0) = 0 \tag{21}$$

Le schéma itératif utilisé pour résoudre ceci est exprimé par, avec $f^0 = f$,$\forall (u,v) \in E$,$\forall v \in V$:

$$\begin{cases} f^0 = f \\ \gamma^{(t+1)}(u,v) = \\ w(u,v) \left(\frac{1}{\epsilon + \left\| \nabla f^{(t+1)}(v) \right\|} + \frac{1}{\epsilon + \left\| \nabla f^{(t+1)}(u) \right\|} \right) \\ f^{(t+1)}(v) = \frac{2\lambda}{2\lambda + \sum\limits_{u \sim v} \gamma^t(u,v)} f^0(v) + \frac{\sum\limits_{u \sim v} \gamma^t(u,v) f^t(u)}{2\lambda + \sum\limits_{u \sim v} \gamma^t(u,v)} \end{cases} \tag{22}$$

De même que pour $p = 2$, nous pouvons définir la fonction $c : V \to \mathbb{R}^+$ par $c^t(v) = \frac{1}{2\lambda + \sum\limits_{u \sim v} \gamma^t(u,v)}$ et le schéma itératif utilisé pour calculer les nouvelles valeurs de f^t s'exprime ainsi, $\forall v \in V$:

$$f^{(t+1)}(v) = 2\lambda c^t(v) f^0(v) + c^t(v) \sum_{u \sim v} \gamma^t(u,v) f^t(u) \tag{23}$$

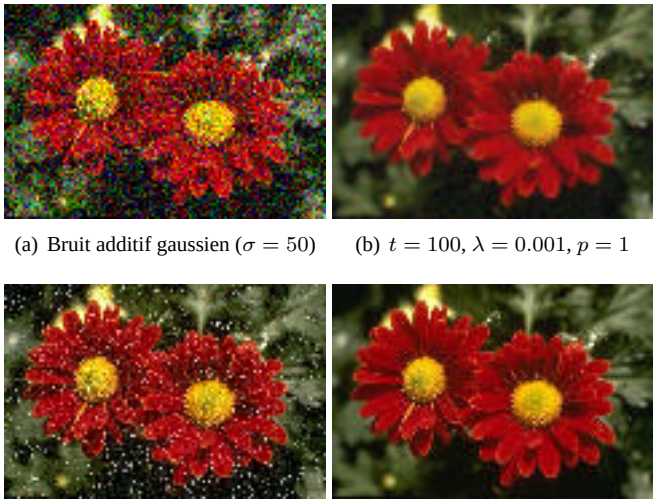
Si l'on compare ceci avec l'algorithme itératif dans le cas de $p = 2$, c'est également un filtre passe-bas mais dont les coefficients sont mis à jour de façon adaptative au cours des itérations en plus de mettre à jour la fonction f. Pour $p = 1$, si $\forall (u,v) \in E, w(u,v) = 1$, ce filtre itératif sur un graphe de type grille régulière correspond au TV digital filter [10] (TV+L^2).

5 Applications

Nous montrons dans cette section différentes applications possibles de la méthode de régularisation sur graphe proposée pour le traitement d'images couleur. Soit $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ une image couleur. Pour appliquer la régularisation sur graphe à une image couleur, il faut lui associer une structure de graphe. Pour la restauration, nous considérons des graphes de type grille régulière en 4-connexité. A chaque noeud est associé la couleur du pixel correspondant, donc pour $f \in \mathcal{H}(V)$, nous avons $f : V \to \mathbb{R}^3$. La régularisation est effectuée sur chaque composante de l'image couleur indépendemment et donc le p-Laplacien est différent pour chaque composante, on a $\gamma_i(u,v) = w(u,v) \left(\|\nabla f_i(v)\|^{p-2} + \|\nabla f_i(u)\|^{p-2} \right)$ pour la i^{ime} composante. Dans le cas de $p = 2$ il est identique pour toutes les composantes, mais pour $p = 1$, il est différent. Afin d'éviter ce problème de non-couplage de la restauration entre les composantes et prendre en compte l'aspect vectoriel des données colorimétriques, le p-Laplacien est considéré comme étant le même pour les trois régularisations, mais en utilisant une norme vectorielle dans le cas où $p = 1$:

$$\gamma(u,v) = w(u,v) \left(\|\nabla f(v)\|_{3D}^{p-2} + \|\nabla f(u)\|_{3D}^{p-2} \right)$$
$$\|\nabla f(v)\|_{3D} = \sqrt{\|\nabla f_1(v)\|^2 + \|\nabla f_2(v)\|^2 + \|\nabla f_3(v)\|^2}$$

La méthode de régularisation s'applique donc sur chaque composante indépendamment avec une pondération des arêtes et un gradient vectoriel qui agissent tous deux comme un couplage entre les composantes. Nous avons considéré la même fonction de pondération pour toutes les applications présentées dans cet article, à savoir $w(u,v) = \frac{1}{\epsilon + \|f_i(u) - f_i(v)\|}, \epsilon \to 0$ et $\|f(u) - f(v)\| = \sqrt{\sum\limits_{i=1}^{3} \|f_i(u) - f_(v)\|^2}$. La figure 1 présente deux exemples de restauration d'images couleur dans le cas de bruit gaussien ou poivre et sel. En comparaison avec les EDP, notre méthode de régularisation est aussi efficace tout en étant plus simple et plus rapide. Notre méthode



(a) Bruit additif gaussien ($\sigma = 50$) (b) $t = 100$, $\lambda = 0.001$, $p = 1$

(c) Bruit poivre et sel (d) $t = 3$, $\lambda = 0.$, $p = 1$

Figure 1 – *Exemples de Restauration.*

régularisation peut s'avérer également extrêmement utile pour la simplification d'images. Partant d'une partition fine d'une image (une partition sur-segmentée mais proche de l'image originale), nous pouvons, en contruisant le graphe d'adjacence des régions, opérer une régularisation directement sur ce graphe. A chaque noeud est associé la couleur moyenne de la région et à chaque arête la distance entre deux régions connexes. Ceci permet d'effectuer une simplification d'image sans déplacement des contours (Figure 2. Utiliser la régularisation sur graphe est une alternative rapide et intéressante à la simplification classique d'images. Enfin la régularisation peut servir à la construction de hiérarchies de partitions : la régularisation tend à faire se rapprocher des modèles similaires et ceux-ci peuvent fusionner au fur et à mesure des itérations. Le principe (voir [16]) revient à alterner diffusion (regularisation sans attache aux données, soit $\lambda = 0$) et fusion de régions selon un critère donné (voir dans [16] pour différents critères). Les critères peuvent être fixes, evolutifs ou adaptatifs. Ceci permet de générer des partitions de finesses différentes à partir d'une partition initiale, ce qui consitue une hiérarchie de partitions utile pour la segmentation. La figure 3 présente des segmentations obtenues pour différents critères de fusion à savoir un critère fixe (seuil égal à 1), un critère évolutif

(a) Image originale (150072 pixels) (b) Zones homogènes (3011 régions)

(c) $t = 5, p = 1, \lambda = 0.01$ (d) $t = 5, p = 2, \lambda = 0.$

Figure 2 – *Simplification d'une image couleur à partir de son graphe d'adjacence de régions.*

(qui augmente selon le niveau de la pyramide irrégulière) et un critère adaptatif [17] ($p = 2$ et la partition fine a été générée par les zones homogènes de niveau 1 [16]).

Figure 3 – *Une hiérarchie de partition par régularisation et fusion par différents critères : les frontières des régions.*

6 Conclusion

Nous avons proposé un cadre général de régularisation discrète basé sur une géométrie différentielle sur graphes. Une famille de filtres linéaires et non linéaires en dérive et ceci permet le traitement d'images couleur via l'utilisation de graphes pour représenter des données colorimétriques. Nous n'avons considéré dans cet article que des connexités dans le domaine de l'image pour définir le graphe (adjacence spatiale dans l'image). L'algorithme de régularisation d'images couleur que nous avons proposé est générique, efficace et facile à mettre en oeuvre. Nous comptons l'appliquer à différents problèmes de vision tels que la coalescence colorimétrique (connexité dans l'espace colorimétrique), les contours actifs, la retouche d'image, la décomposition structure-texture, etc.

Références

[1] A. Bakushinsky et A. Goncharsky. *Ill-Posed Problems : Theory and Applications*. Kluwer Academic Publishers, 1994.

[2] G. Aubert et P. Kornprobst. *Mathematical Problems in Image Processing*. Springer-Verlag, 2002.

[3] A.N. Tikhonov et V.Y. Arsenin. *Solution of Ill-posed Problems*. Winston & Sons, 1977.

[4] T.F. Chan et J. Shen. *Image Processing and Analysis - Variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.

[5] P. Perona et J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis Machine Inteligence.*, 12 :629–639, 1990.

[6] L. Rudin, S. Osher, et E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60 :259–268, 1992.

[7] G. Sapiro et D. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Transactions on Image Processing*, 10(5) :1582–1586, 1996.

[8] P. Blomgren et T.F. Chan. Color tv : total variation methods for restoration of vector-valued images. *IEEE Transactions on Image Processing*, 7(3) :304–309, 1998.

[9] A. Brook, R. Kimmel, et N. Sochen. Variational restoration and edge detection for color images. *Journal of Mathematical Imaging and Vision*, 18 :247–268, 2003.

[10] T. Chan, S. Osher, et J. Shen. The digital tv filter and nonlinear denoising. *IEEE Transactions on on Image Processing*, 10 :231–241, 2001.

[11] D. Zhou et B. Schölkopf. A regularization framework for learning from graph data. Dans *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, pages 132–137, 2004.

[12] R. Diestel. *Graph Theory*, volume 173. Springer-Verlag, 2005.

[13] J. Matas, R. Marik, et J. Kittler. The color adjacency graph representation of multi-coloured objects. Rapport technique VSSP-TR-1/95, University of Surrey, Surrey, Great Britain, 1995.

[14] A. Bensoussan et J-L. Menaldi. Difference equations on weighted graphs. *Journal of Convex Analysis*, 12, 2005.

[15] S. Bougleux et A. Elmoataz. Image smoothing and segmentation by graph regularization. Dans *International Symposium on Visual Computing*, volume LNCS 3804, pages 745–752, 2005.

[16] O. Lezoray, C. Meurie, P. Belhomme, et A. Elmoataz. Multi-scale image segmentation in a hierarchy of partitions. Dans *EUSIPCO*, 2006.

[17] R. Nock et F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11) :1452–1458, 2004.

Segmentation hiérarchique et espaces couleurs

C. Meurie O. Lezoray

LUSAC - EA 2607, Groupe Vision et Analyse d'Image

Université de Caen Basse-Normandie
Site Universitaire, BP 78, F-50130 Cherbourg-Octeville

{cmeurie,olezoray}@info.unicaen.fr

Résumé

Dans cet article, nous étudions l'influence de l'espace couleur sur différentes méthodes de segmentation hiérarchiques. Pour ce faire, nous nous appuyons sur une série de tests effectués sur 100 images de la « Berkeley Segmentation Dataset and Benchmark (BSDB) ». Nous montrons l'influence de quatre espaces couleurs caractéristiques à savoir l'espace RGB, YC_bC_r, $L^*a^*b^*$ et $IHSL$ sur différentes segmentations hiérarchiques produites par les zones quasi-plates, la Ligne de Partage des Eaux hiérarchique non paramétrique et une nouvelle méthode que nous proposons ie. les zones homogènes stratifiées. Nous montrons que ces différentes méthodes de segmentation hiérarchiques ne sont pas toutes égales devant leur sensibilité aux différents espaces couleur. Nous concluons en proposant un ordre de préférence des espaces couleur à utiliser en fonction de la méthode de segmentation hiérarchique.

Mots clefs

Segmentation d'images couleur, hiérarchie de partitions, espace couleur, zones homogènes stratifiées.

1 Introduction

Le paradigme de la segmentation d'images consiste à partitionner une image en régions homogènes c'est-à-dire en un ensemble connexe de points de l'image ayant des propriétés communes. Différentes méthodes de segmentation s'appliquant aux images couleur existent et peuvent être regroupées en deux catégories à savoir les techniques de segmentation bas-niveaux travaillant au niveau du pixel et les techniques de haut-niveaux travaillant au niveau supérieur qui est celui de la région. La plupart du temps, les méthodes de segmentation bas-niveaux ne peuvent pas construire directement une bonne partition finale d'une image et il faut donc avoir recours à des méthodes de segmentation hiérarchiques offrant de meilleurs résultats. Cependant l'application de certaines de ces techniques aux images couleur pose quelques difficultés et notamment dans le cadre de la morphologie mathématique comme le signale MEURIE [1]. Face à cette constatation et aux derniers travaux de ANGULO sur l'influence des conditions d'éclairage dans la segmentation morphologique couleur par LPE [2], il nous a semblé intéressant de montrer l'influence de l'espace couleur sur différentes méthodes de segmentation hiérarchiques. Nous commençons tout d'abord par rappeler les transformations permettant de passer d'un espace RGB ou XYZ à l'espace désiré. Dans un deuxième temps, nous rappelons la notion de partition, de hiérarchie de partitions, deux méthodes de segmentation hiérarchiques à savoir les zones quasi-plates [3, 4, 5] et la Ligne de Partage des Eaux (LPE) hiérarchique non paramétrique [6, 7] et proposons une nouvelle méthode appelée zones homogène stratifiées. Nous présentons dans une troisième section, une série de tests établis sur 100 images de la « Berkeley Segmentation Dataset and Benchmark (BSDB) »[8] et mettons en avant l'influence de l'espace couleur sur les trois méthodes de segmentation hiérarchiques mentionnées dans cet article.

2 Les espaces couleur

L'objectif de notre étude étant de montrer l'influence de l'espace couleur sur différentes méthodes de segmentation hiérarchiques, nous rappelons quelques notions de bases sur les différents espaces que nous allons utiliser dans cet article. Afin d'être le plus exhaustif possible sans pour autant tester tous les espaces existants, nous allons utiliser quatre espaces faisant partie des grandes familles d'espaces de représentation couleur présentées par VANDENBROUCKE dans [9] à savoir l'espace RGB, un espace de luminance-chrominance (l'espace YC_bC_r), un espace perceptuellement uniforme (l'espace $L^*a^*b^*$) et un espace de coordonnées perceptuelles (l'espace IHSL).

2.1 L'espace YC_bC_r

L'espace YC_bC_r est le standard international dédié au codage digital des images de la télévision numérique et a la particularité par rapport aux autres espaces dédiés à la télévision de ne pas imposer de règle sur le blanc de référence à utiliser. La transformation de l'espace RGB en l'espace YC_bC_r est donnée par la relation suivante :

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.2989 & 0.5866 & 0.1145 \\ -0.1688 & -0.3312 & 0.5 \\ 0.5 & -0.4184 & -0.0816 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

2.2 L'espace $L^*a^*b^*$

Le système $L^*a^*b^*$ est une approximation de l'espace d'Adams-Nickerson dans lequel l'amplitude perceptuelle de la couleur est définie en termes d'échelles de couleurs opposées couvrant l'intégralité du spectre visible par l'oeil humain. Le passage au modèle $L^*a^*b^*$ s'obtient à partir du modèle XYZ par les relations non linéaires données ci-dessous :

$$L^* = \begin{cases} 116 \times \left(\frac{Y}{Y_0}\right)^{\frac{1}{3}} - 16 & \text{si} \quad \frac{Y}{Y_0} > 0.008856 \\ 903.3 \times \left(\frac{Y}{Y_0}\right) & \text{si} \quad \frac{Y}{Y_0} \leq 0.008856 \end{cases}$$

$$a^* = 500\left(f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right)\right)$$

$$b^* = 300\left(f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right)\right)$$

$$f(x) = \begin{cases} x^{\frac{1}{3}} & \text{si} \quad x > 0.008856 \\ 7.787x + \frac{16}{116} & \text{si} \quad x \leq 0.008856 \end{cases}$$

ou L^* représente la luminance et par conséquent l'opposition noir-blanc par une valeur comprise entre 0 (noir) et 100 (blanc). a^* mesure l'opposition vert-rouge par une valeur comprise entre -100 et $+100$ (a^* est positif si la couleur contient du rouge, négatif si la couleur contient du vert et nulle si aucun des deux). b^* mesure l'opposition bleu-jaune par une valeur comprise entre -100 et $+100$ (b^* est positif si la couleur contient du jaune, négatif si la couleur contient du bleu et nulle si aucun des deux). X_0, Y_0, Z_0 désignent les coordonnées XYZ de l'illuminant (illuminant E pour notre étude).

2.3 L'espace $IHSL$

Le système $IHSL$ proposé par HANBURY [10] est une amélioration de l'espace HSI. Le passage de l'espace RGB à celui-ci est donné par les relations suivantes :

$$L = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B$$

$$S = \max(R, G, B) - \min(R, G, B)$$

$$H = \begin{cases} 360° - H_1 & \text{si} \quad B > G \\ H_1 & \text{si} \quad B \leq G \end{cases}$$

$$H_1 = \arccos\left[\frac{R - \frac{1}{2}G - \frac{1}{2}B}{(R^2 + G^2 + B^2 - RG - RB - BG)^{\frac{1}{2}}}\right]$$

3 Segmentation hiérarchique d'images couleur

Dans cette section, nous rappelons deux méthodes de segmentation hiérarchiques d'images couleur que l'on retrouve le plus souvent dans la littérature à savoir les zones quasi-plates [3, 4, 5] et la Ligne de Partage des Eaux hiérarchique non paramétrique [6, 7] et terminons par proposer une nouvelle méthode faisant référence au critère connectif des zones homogènes introduit par LEZORAY ET AL. [11]. Mais avant toute chose, rappelons les définitions de partition d'une image et de hiérarchie de partitions.

3.1 Partition d'une image

En traitement d'images, une image I est considérée dans la plupart des cas comme étant un ensemble de pixels : $I = \{p_1, p_2, \cdots, p_n\}$. Lorsque nous parlons d'image segmentée, nous faisons référence à une image divisée en régions disjointes selon un critère donné où chaque région R est un sous-ensemble de pixels connexes de l'image constituée de $|R|$ pixels répondant à un même critère d'homogénéité. Mais cette même image segmentée se trouve être le résultat d'un algorithme de segmentation et donc une partition du domaine de l'image.

Définition 1 (Partition) Une partition P est un ensemble de composantes connexes ou régions $P = \{R_1, R_2, \ldots, R_k\}$ tel que : l'union des régions de la partition donne l'ensemble de départ : $I = \bigcup_{i=1}^{k} R_i$, les régions ont une intersection nulle : $\forall i, j, i \neq j, R_i \cap R_j = \emptyset$

Différents algorithmes existants peuvent être utilisés pour segmenter une image et donc créer ce que nous venons d'appeler une partition. Mais ces même algorithmes peuvent également définir, en jouant sur leurs paramètres, un empilement de partitions de niveaux croissants appelé hiérarchie de partitions. Nous comprendrons qu'il est alors important de définir une relation d'ordre entre deux partitions : une partition P est incluse dans une partition Q si toute région R_j^P est incluse dans une région R_i^Q. Ceci nous amène alors à définir une hiérarchie de partitions emboîtées d'une image. Soit H un ensemble de partitions associées à une image, H forme une hiérarchie de partitions s'il est possible d'établir un ordre d'inclusion parmi toute paire d'éléments de l'ensemble H. Deux régions quelconques appartenant à des partitions différentes de la hiérarchie sont soit disjointes soit incluses l'une dans l'autre.

Définition 2 (Hiérarchie de partitions emboîtées) Une hiérarchie de partitions emboîtées d'une image est un ensemble de partitions $H = \{P_1, P_2, \ldots, P_l\}$ tel que les régions de la partition $P_i = \left\{R_1^i, R_2^i, \ldots, R_k^i\right\}$ sont incluses dans les régions de la partition $P_j = \left\{R_1^j, R_2^j, \ldots, R_{k'}^j\right\}$ avec $j > i, k' > k$ et $R_m^i \subseteq R_p^j$ ou $R_m^i \cap R_p^j = \emptyset$

La notation généralement utilisée consiste à appeler P_i le niveau i de la hiérarchie. P_0 représente le niveau inférieur de la hiérarchie et la partition la plus fine d'où son appellation « partition fine ». P_l constitue quant à lui le niveau supérieur de la hiérarchie et la partition la plus grossière. D'après la définition même de la hiérarchie de partitions emboîtées, les régions des niveaux inférieurs étant incluses dans les régions des niveaux supérieurs, une partition de niveau $i+1$ peut être obtenue par une fusion de plusieurs régions de niveau i.

En morphologie mathématique, le fait d'avoir un ordre entre les partitions implique que la hiérarchie de partitions forme un treillis complet. Les principaux critères morphologiques permettant de définir une hiérarchie de partitions sont basés sur la notion de connexion. Cette notion de connexion réside dans la définition d'un critère puisqu'une image est segmentée en zones au regard d'un critère donné. Les zones plates ou quasi-plates et la LPE hiérarchique non paramétrique que nous allons présenter ci-dessous sont les principaux critères connectifs de segmentation.

3.2 Les zones plates ou quasi-plates

Les zones plates d'une image I sont les composantes connexes ayant une valeur constante ce qui constitue un critère connectif de segmentation. Elles furent introduites par SALEMBIER ET SERRA [3, 4]. L'utilisation brute d'une image en zones plates n'est pas très intéressante en soi puisque nous sommes face à une image très sur-segmentée. Une simplification au préalable de l'image où une fusion selon un certain critère de zones plates *a posteriori* permet de réduire le nombre de régions de l'image afin d'être utilisées par exemple comme marqueurs pour la Ligne de Partage des Eaux. Pour palier cet inconvénient MEYER [5] a proposé d'étendre le concept de zone plate à celui de zone quasi-plate.

Définition 3 (Zone quasi-plate) Deux points p et q appartiennent à la même zone quasi-plate d'une image I ssi il existe un chemin connexe $(p_1, p_2, \cdots, p_n)$ entre ces deux points tel que $p_1 = p$ et $p_n = q$ et pour tout i : $\|I(p_i) - I(p_{i+1})\| \leq \lambda$

avec $\|.\|$ représentant une norme L_2 et λ le critère de seuil. Notons bien évidemment qu'un critère de seuil $\lambda = 0$ revient à considérer une zone plate au sens strict du terme et qu'une utilisation croissante de ce critère permet de définir une hiérarchie de partitions. Le nombre de régions diminue au fur et à mesure de la progression dans la hiérarchie pour arriver vers une image où la perte d'information est très importante, il est alors nécessaire d'être attentif sur la détermination du critère de seuil afin de simplifier l'image initiale mais sans trop la dégrader. Une méthode permettant de définir le meilleur niveau de la hiérarchie en terme de compromis entre fidélité aux données et complexité du modèle a été proposée par MEURIE [1].

La figure 1 illustre plusieurs segmentations produites par les zones quasi-plates pour différents niveaux de la hiérarchie (pour différents λ) et dans deux espaces couleur différents.

3.3 La LPE hiérarchique non paramétrique

La Ligne de Partage des Eaux (LPE) est un opérateur de croissance de régions définissant une connexion par cheminement basée sur le gradient morphologique d'une image. Les germes de la LPE étant les minima du gradient morphologique (il s'agit ici du gradient de DIZENZO

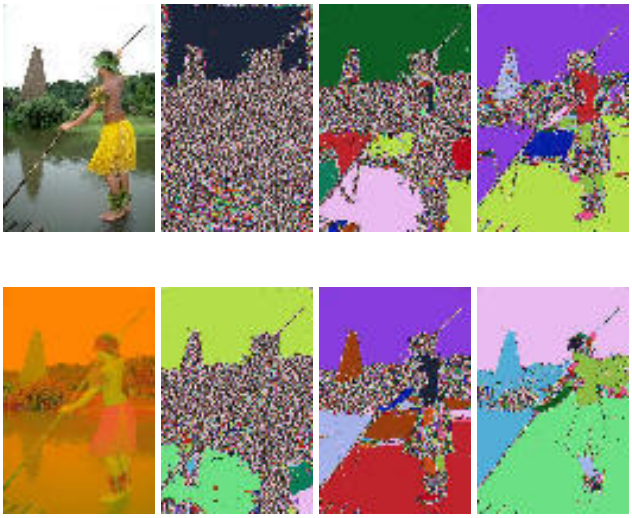


Figure 1 – *Hiérarchies de partitions produites par les zones quasi-plates (image initiale et niveaux 1, 5, 15) dans l'espace couleur RGB (ligne du haut) et $L^*a^*b^*$ (ligne du bas)*

[12]). C'est une méthode ayant fait ses preuves et très utilisée dans le domaine de la segmentation d'images mais l'inconvénient majeur réside dans l'obtention d'une sur segmentation due à un nombre important de minima. Une alternative pour pallier cet inconvénient consiste à ne plus utiliser les minima comme germes de LPE mais des marqueurs correspondant aux régions à segmenter. L'arrivée de techniques de segmentation hiérarchiques a probablement engendré l'intérêt d'une LPE hiérarchique non paramétrique. L'algorithme des cascades de la LPE que l'on peut retrouver dans [6, 7] permet de construire cette LPE hiérarchique non paramétrique qui procède à une fusion des bassins versants. Il est ainsi basé sur la reconstruction de la fonction gradient de l'image mosaïque avec sa LPE. En réitérant un certain nombre de fois cette procédure de sorte à obtenir une cascade de LPE, nous obtenons une hiérarchie de partitions.

La figure 2 illustre plusieurs segmentations obtenues par la LPE hiérarchique non paramétrique pour différents niveaux de la hiérarchie et dans les espaces couleur RGB et YC_bC_r.

3.4 Une nouvelle méthode de segmentation hiérarchique : les zones homogènes stratifiées

Après avoir rappelé deux approches de segmentation hiérarchiques communément utilisées, nous proposons une nouvelle méthode permettant de créer une hiérarchie de partitions et basée sur le critère connectif des zones homogènes défini par LEZORAY ET MEURIE [11, 1]. Si l'on désire construire une hiérarchie de partitions de

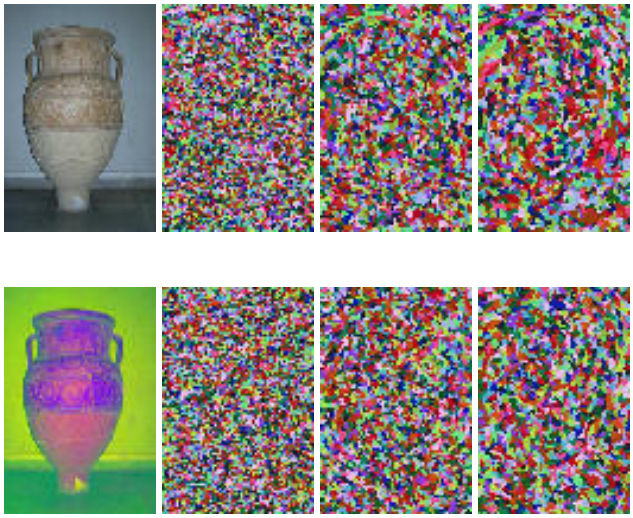


Figure 2 – *Hiérarchies de partitions produites par la LPE hiérarchique non paramétrique (image initiale et niveaux 1, 10, 20) dans l'espace couleur RGB (ligne du haut) et YC_bC_r (ligne du bas).*

zones homogènes qui soit stratifiée, il faut se contraindre à respecter le principe d'inclusion des régions entre deux niveaux successifs de la hiérarchie.Une façon de réaliser ceci est d'appliquer le principe des zones homogènes sur un graphe d'adjacence de régions obtenu par une partition fine par zones homogènes. Chaque région du graphe étant décrite par sa moyenne, on peut appliquer la même règle de croissance, mais cette fois sur le graphe.

Définition (Zones homogènes stratifiées) Deux noeuds N_p et N_q d'un graphe d'adjacence de régions G appartiennent à une même zone homogène d'une image I ssi $\left\|\overline{I}(N_p) - \overline{I}(N_q)\right\| \leq k \times \lambda(Germe(N_p))$, avec $Germe(N_p)$ le noeud germe de la région de N_p et $\lambda(N_p)$=$\frac{1}{n_v}\sum_{N_{p_v}\in V(N_p)} \left\|\overline{I}(N_p) - \overline{I}(N_{p_v})\right\|$

avec $\overline{I}(N_p)$ la couleur moyenne des pixels du noeud N_p, $V(N_p)$ désigne l'ensemble des noeud voisins du noeud N_p et n_v le cardinal de cet ensemble. Chaque noeud N_p du graphe est enfilé dans une file hiérarchique avec pour priorité la valeur de $\lambda(N_p)$. L'algorithme de construction d'une hiérarchie de partitions de zones homogènes i.e. les zones homogènes stratifiées est alors donné par l'algorithme 1.

L'algorithme prend deux paramètres k et k'. k définit la finesse de la partition initiale et k' définit la finesse des partitions successives de la hiérarchie. Le réglage de k et k' est primordial pour la performance de l'algorithme. Pour nos expérimentations, nous avons ici, arbitrairement fixé $k = 0.5$ et $k' = 1$.

λ : **entier** ; k : **réel** ; k' : **réel** ;
$\lambda \leftarrow 1$; Définir λ_{end}
$P_\lambda \leftarrow$ Zones homogènes de finesse k de l'image initiale.
$G_\lambda = (N_\lambda, A_\lambda)$ pour une partition initiale P_λ.
Tant que ($\lambda \leq \lambda_{end}$) **faire**
  $G_{\lambda+1} \leftarrow$ zones homogènes de finesse k' de G_λ
  $\lambda \leftarrow \lambda + 1$
Fait

Algorithme 1 – *Hiérarchie de partitions par zones homogènes.*

Figure 3 – *Hiérarchie de partitions produites par les zones homogènes stratifiées (image initiale et niveaux 1, 5, 15) dans l'espace couleur RGB (ligne du haut) et $IHSL$ (ligne du bas).*

Une illustration de hiérarchie de partitions d'images segmentées par les zones homogènes stratifiées, pour différents niveaux de la hiérarchie et pour deux espaces couleur différents à savoir l'espace RGB et $IHSL$ est proposée sur la figure 3.

4 Résultats expérimentaux

Dans cette section, nous présentons une série de tests effectués sur 100 images de la « Berkeley Segmentation Dataset and Benchmark (BSDB) »[8] et dont les résultats sont donnés sur la figure 5. De par ces tests, nous avons testé l'influence de l'espace couleur sur deux méthodes de segmentation hiérarchique souvent utilisées dans la littérature ainsi qu'une nouvelle méthode proposée appelée zones homogènes stratifiées. Les images segmentées produites ont été évaluées à l'aide de trois critères à savoir le Mean Square Error (MSE), le Normalized Color Difference (NCD) et le Peak Signal to Noise Ratio (PSNR).

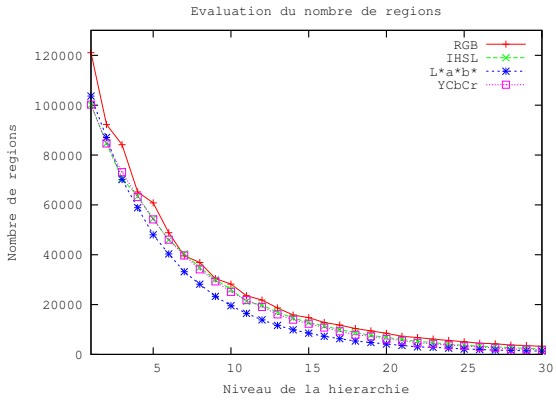


Figure 4 – *Evaluation du nombre de régions dans une pyramide de segmentations obtenues à partir des zones quasi-plates*

Les figures 1, 2 et 3 illustrent bien l'influence de l'espace couleur sur les segmentations hiérarchiques produites. En effet, pour un même niveau de la hiérarchie, quelque soit la méthode de segmentation, le nombre de régions finales dépend grandement de l'espace couleur utilisé. De ce fait, il existe une qualité de segmentation différente entre l'espace classique RGB et tout autre espace de représentation couleur. Nous appuyons cette dernière remarque par l'illustration de la figure 4 où nous remarquons ce comportement. Nous pouvons aller jusqu'à conclure qu'une segmentation hiérarchique produite par la méthode des zones quasi-plates dans l'espace $L^*a^*b^*$ fournit, pour un même niveau de hiérarchie, un nombre de régions inférieur aux autres espaces couleur.

Aux vues des résultats présentés sur la figure 5, nous pouvons conclure que le MSE augmente au fur et à mesure que l'on progresse dans la hiérarchie de partition. Ceci s'explique par le fait que nous perdons de l'information et que la segmentation devient de plus en plus grossière. Nous pouvons aussi constater que les zones quasi-plates sont très sensibles à l'espace couleur dans lequel celles-ci sont utilisées. A contrario, l'espace couleur a très peu d'influence sur les segmentations produites par la LPE hiérarchique non paramétrique. Les variations très légères entre les différents espaces peuvent s'expliquer par le fait que le nombre de régions n'évolue pas de manière importante selon l'espace. Nous appuyons cette remarque par l'illustration de la figure 2. Les zones homogènes stratifiées se trouvent être un bon intermédiaire entre les deux autres méthodes testées en ce qui concerne la qualité de la segmentation comme cela a pu être montrée dans [1] mais aussi en ce qui concerne leur sensibilité à l'espace couleur. Nous pouvons conclure cette analyse en montrant que le choix de l'espace couleur n'est pas primordiale pour l'utilisation de la Ligne de Partage des Eaux hiérarchique non paramétrique. En ce qui concerne l'utilisation des zones quasi-plates et des zones homogènes stratifiées, il faut être attentif au choix de l'espace couleur. Pour facilité ce choix, nous pouvons utiliser l'ordre cité ci-après définissant un ordre de préférence sur les espaces couleur pour un nombre de régions décroissant : $IHSL, RGB, YC_bC_r, L^*a^*b^*$. Notons que cet ordre est quasiment identique selon la méthode de segmentation hiérarchique utilisée si ce n'est que l'espace $IHSL$ considéré comme le meilleur espace pour les zones quasi-plates se trouve être relégué en dernière position pour les deux autres méthodes testées.

5 Conclusion et perspectives

Dans cet article, nous avons présenté l'influence de l'espace couleur dans la segmentation hiérarchique d'images couleur. Pour ce faire, nous avons testé deux méthodes très utilisées dans ce domaine à savoir les zones quasi-plates et la LPE hiérarchique non paramétrique (algorithme des cascades) et proposé une nouvelle méthode appelée zones homogènes stratifiées produisant des segmentations intermédiaires à celles produites par les deux autres méthodes citées précédemment. Les résultats mettent en évidence que les différentes méthodes n'ont pas la même sensibilité aux différents espaces couleur. Les zones quasi-plates réagissent fortement à ces derniers, les zones homogènes stratifiées en moindre mesure alors que la LPE hiérarchique non paramétrique reste quasiment insensible. Pour conclure, nous pouvons définir un ordre de préférence sur les espaces couleur quasi-identique selon la méthode de segmentation hiérarchique utilisée et correspondant à : $IHSL, RGB, YC_bC_r, L^*a^*b^*$. Notons que seul l'espace $IHSL$ est relégué de la première position pour les zones quasi-plates à la dernière position pour les autres méthodes. En terme de perspectives, il serait intéressant d'étendre cette étude aux méthodes hiérarchiques de simplification et de fusion sur graphe ainsi qu'aux différentes méthodes de calcul de gradient pour les opérations de morphologie mathématique.

Références

[1] C. Meurie. *Segmentation d'images couleur par classification pixellaire et hiérarchies de partitions*. Thèse de doctorat, Université de Caen Basse-Normandie, Octobre 2005.

[2] J. Angulo et B. Marcotegui. Sur l'influence des conditions d'éclairage dans la segmentation morphologique couleur de lpe. Dans *COmpression et REprésentation des Signaux Audiovisuels*, pages 313–318, 2005.

[3] P. Salembier et J. Serra. Morphological multiscale image segmentation. Dans *SPIE Visual Communications and Image Processing*, pages 620–631, 1992.

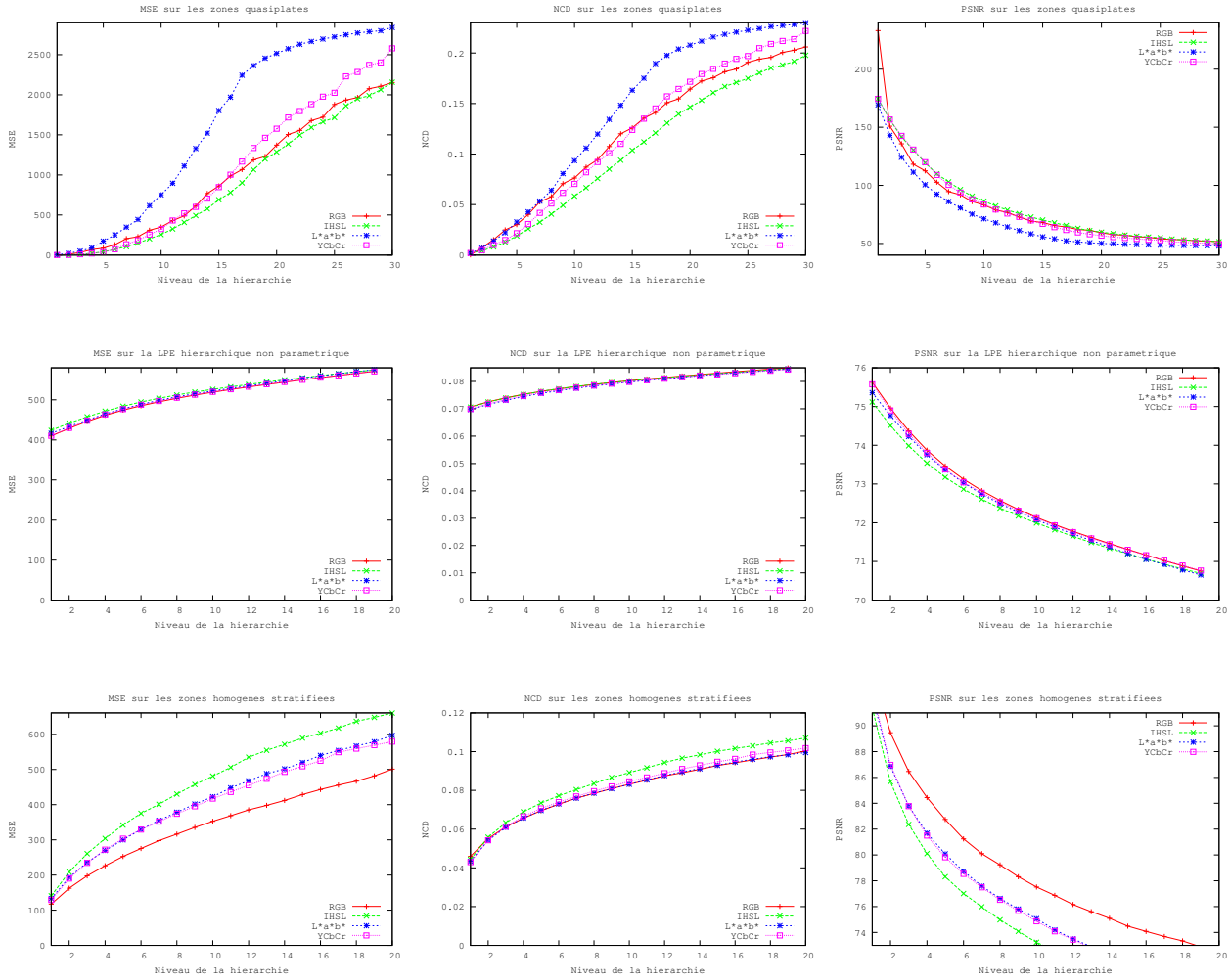


Figure 5 – *Evaluation d'images segmentées par différentes méthodes de segmentation hiérarchiques et dans quatre espaces couleur différents (segmentation par les zones quasi-plates (en haut), LPE hiérarchique non paramétrique (au centre) et les zones homogènes stratifiées (en bas), toutes évaluées à l'aide du MSE (à gauche), le NCD (au centre) et le PSNR (à droite)).*

[4] P. Salembier et J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Trans on Image Processing*, 4(8) :1153–1160, 1995.

[5] F. Meyer. From connected operators to levellings. *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 191–199, 1998.

[6] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. *Mathematical Morphology adns its Applications to Image and Signal Processing*, pages 69–76, 1994.

[7] J. Angulo et J. Serra. Color segmentation by ordered mergings. Dans *Proc. of ICIP 2003*, volume 2, pages 125–128, 2003.

[8] Berkeley. The berkeley segmentation dataset and benchmark. http ://www.cs.berkeley.edu/projects/vision/ grouping/segbench.

[9] N. Vandenbroucke. *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application à l'analyse d'image de football*. Thèse de doctorat, Université de Lille 1, Décembre 2000.

[10] A. Hanbury. A 3d-polar coordinate colour representation well adapted to image analysis. Dans *SCIA*, 2003.

[11] O. Lezoray, C. Meurie, P. Belhomme, et A. Elmoataz. Hiérarchie de partitions pour la simplification et la segmentation d'images couleur. Dans *COmpression et REprésentation des Signaux Audiovisuels*, pages 231–236, 2005.

[12] S. DiZenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing*, 33 :116–126, 1986.

Fingerprint audio robuste pour la gestion de droits

Jérôme Lebossé
France Télécom R&D,
32 rue des coutures,
14000 Caen, France
jerome.lebosse@orange-ft.com

Luc Brun
GREYC UMR 6072,
ENSICAEN, 6 boulevard du Maréchal
Juin,
14050 Caen, France
luc.brun@greyc.ensicaen.fr

Jean Claude Paillès
France Télécom R&D,
32 rue des coutures,
14000 Caen, France
jeanclaude.pailles@orange-ft.com

Résumé

Le fingerprint audio permet d'identifier un document audio éventuellement corrompu, à partir d'un court exemple. Ces méthodes peuvent être utilisées dans le cadre de la gestion des droits numériques (DRM) dans le but d'associer les informations de gestion et de contrôle à chaque document. Dans cet article, nous proposons un nouveau mode de calcul de fingerprint audio qui combine une méthode de segmentation avec un nouveau schéma de construction des codes définissants le fingerprint. La méthode proposée est robuste aux altérations du document audio telles la compression et la suppression de parties ou décalages temporels.

Mots clefs

Audio fingerprint, segmentation, Digital Rights Management, identification.

1 Introduction

Les méthodes de gestion des droits numériques (DRM) empêchent les copies illégales de contenus multimédias et leur distribution par Internet. Cependant, les contenus déjà transmis et copiés avant l'avènement de la DRM sont à jamais perdus pour leurs créateurs. De plus, la conversion numérique-analogique-numérique permet de contourner et de s'affranchir des protections par DRM. Les contenus peuvent alors être transmis sur un réseau non protégé. Des solutions à base de watermarking (ou tatouage) ont alors été proposées [1]. Une marque digitale (watermark) est un message imperceptible ajouté au contenu audio sans altération de sa qualité. Cependant, si l'ajout d'une marque n'altère pas la qualité perceptuelle du document, sa suppression peut généralement s'effectuer également sans altérer la qualité du signal. De plus, à notre connaissance, toutes les méthodes de sécurité basées sur des techniques de watermarking reposent sur la non divulgation de la méthode utilisée pour apposer la marque dans le document. La divulgation ou la découverte de ces méthodes compromet donc la sécurisation des documents basés sur ces techniques de watermarking.

L'identification audio à base de fingerprint représente une approche alternative pour traiter des problèmes de protection de copyrights. Les systèmes de fingerprint multimédia permettent de déterminer la similarité perceptuelle entre deux contenus en utilisant une représentation condensée du signal (le fingerprint). Dans la plupart des applications de fingerprint, un grand nombre de données multimédia sont stockées dans une base de données et associées à leurs métadonnées respectives telles que le nom de l'auteur, le titre, … Le fingerprint peut alors être vu comme un index permettant d'effectuer des requêtes sur le contenu perceptuel des données. Dans le cadre des DRM, les métadonnées associées à un document peuvent inclure des informations sur les opérations autorisées sur celui-ci (e.x. nombre de copies).

L'application de l'audio fingerprint pour la gestion des droits numériques implique certaines exigences. Tout d'abord, le fingerprint se doit d'être le plus invariant possible aux altérations du contenu audio comme la compression ou les décalages temporels. Ensuite, l'algorithme doit respecter des contraintes pour permettre son intégration au sein d'un appareil portable ou d'un ordinateur familial. Plus précisément, la taille de chaque fingerprint doit être la plus concise possible afin de répondre aux exigences de stockage dans la base de données tout en contenant suffisamment d'information discriminante pour caractériser et identifier individuellement chaque document. De plus, le calcul du fingerprint doit pouvoir être réalisé en parallèle à la lecture du document audio. Finalement, l'algorithme de fingerprint doit être capable d'identifier un document à partir d'un échantillon de seulement quelques secondes de signal.

Notons de plus que si l'identification d'un document donne accès à sa lecture, la non reconnaissance d'un document dont le fingerprint est stocké dans la base de données est équivalent à un refus de service. Par conséquent, le taux de faux négatifs du système d'identification se doit d'être très bas.

Dans ce papier, nous décrivons une méthode d'extraction de fingerprint robuste qui réponde aux exigences précédemment citées. Après une description des approches alternatives (Section 2), nous décrivons notre méthode de calcul d'identifiant audio dans la Section 3. Sa capacité à satisfaire les contraintes d'une application de gestion des droits numériques est enfin évaluée en Section 4.

2 Etat de l'art

Comme précisé dans la Section 1, une méthode de fingerprint audio doit être capable d'identifier un court échantillon de quelques secondes de signal audio. Un échantillon sur lequel sera appliquée le processus d'identification est appelé *segment*. Typiquement, la taille d'un segment peut varier entre 5 à 10 secondes. De ce fait, un nombre suffisant de caractéristiques discriminantes doit être extrait du segment pendant un intervalle très court. La première phase d'une méthode d'extraction de fingerprint consiste à diviser le signal en intervalles de temps (appelé *frame*) de quelques millisecondes. Une valeur (appelée *sous fingerprint*) est alors associée à chaque frame en codant les caractéristiques acoustiques du signal sur le frame.

La décomposition du signal en frames (appelé *enframing*) doit être robuste aux suppressions de parties du signal et aux décalages temporels qu'elles induisent. Une méthode habituelle pour répondre à cette robustesse consiste à utiliser une fenêtre recouvrante (e.g. [6] utilise des frames de 0,37s avec un taux de recouvrement de 31/32). Cependant, l'utilisation de fenêtres recouvrantes revient seulement à réduire l'influence d'altérations temporelles que peut subir le signal (Section 4). Par exemple, la dégradation d'un signal par un décalage de 12ms d'une suite de frames de 50ms se recouvrant de 50% décalera toutes les fenêtres de 12ms.

Une solution alternative à l'enframing consiste à trouver des positions particulières dans le signal (appelées onsets). Les onsets ([2]) sont définis par un fort gradient calculé sur des caractéristiques perceptuelles du signal traduisant l'apparition d'un changement brusque du signal. Généralement, les techniques de détection d'onsets se basent sur des mesures d'énergies impliquant souvent une pondération fréquentielle. Cette méthode a récemment été améliorée en incorporant une prise en compte de sous bandes fréquentielles ([4]).

Dans [3], les auteurs proposent un schéma de détection d'onsets dans un document musical basé sur les informations apportées par l'énergie fréquentielle du signal combinée à sa phase. En effet, l'utilisation de l'énergie du signal a déjà prouvé son efficacité à détecter d'importants changements du signal, plus particulièrement dans des signaux avec des changements de notes à fortes consonance percussive comme la batterie, puisque l'énergie dénote alors un fort gradient. L'information de phase quant à elle permet de détecter les onsets dans des signaux aux sources mixtes et aux transitions moins franches.

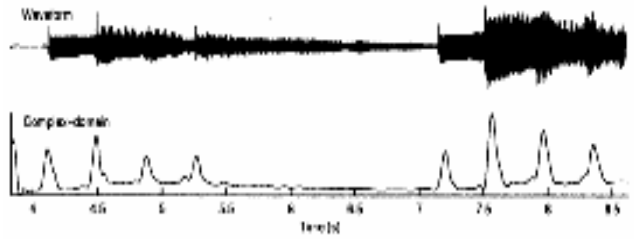


Figure 1 – *Détection d'onsets.*

Comme le montre la figure 1, cette méthode fournit une courbe temporelle décrivant des pics à l'emplacement des onsets et aplanie le reste du temps. L'utilisation d'un filtre médian appliqué à cette courbe permet d'extraire les positions des onsets.

Cependant, l'inconvénient principal de cette approche est dû au fait que le nombre d'onsets détectés dans un laps de temps est imprévisible et est, dans bien des cas, trop faible pour caractériser efficacement un segment. Donc, même si l'approche par onsets permet de synchroniser deux signaux audio basés sur le même contenu, cette approche ne peut pas être utilisée pour le fingerprint audio.

L'approche par frames est alors généralement utilisée pour décomposer le signal en courts intervalles. Une fois le signal divisé, des algorithmes d'extraction de caractéristiques sont alors appliqués pour chaque intervalle. La suite de caractéristiques calculées tout au long du signal définie le fingerprint. Certaines approches combinent la pondération des frames par une fenêtre de Hamming avec l'utilisation de caractéristiques extraites à partir du spectre fréquentiel du signal, comme les Mel Frequency Cepstral Coefficients ([5], [6]). Dans [7], l'auteur associe à chaque frame un bit égal à 1 si l'énergie totale d'un frame est supérieure à celle du frame précédent. Le bit est mis à 0 sinon. D'après l'auteur, cette méthode peut servir à accélérer le processus de recherche en éliminant les mauvais candidats. Mais l'utilisation d'un seul bit par frame ne fournit pas assez d'informations discriminantes pour identifier un segment de façon robuste.

Dans [8], une méthode appelée Distorsion Discriminant Analysis est utilisée pour transformer le signal audio en un vecteur de caractéristiques de plus faible dimension. Tout d'abord, une Modulated Complex Lapped Transform (MCLT) est appliquée sur chaque frame. Cette transformée est un cas particulier de la transformée de Fourier discrète puisqu'elle prend des segments du signal, recouvrants à 50% puis calcule les coefficients

d'amplitude de la décomposition spectrale pour un nombre de bandes de fréquences déterminé. Puis, une Analyse en Composantes Principales Orientées (OPCA) est utilisée pour trouver un ensemble de projections du signal qui maximise le ratio Signal sur Bruit. L'auteur combine alors plusieurs couches d'OPCA pour créer un réseau qui extrait alors des caractéristiques robustes au bruit sur un segment. Finalement, pour 20 secondes de signal audio, cette méthode calcule un vecteur de 64 valeurs. Un vecteur de ce type est alors généré toutes les 243,6ms. L'identification est alors effectuée en calculant la distance Euclidienne entre un vecteur et ceux contenus dans une base de données de vecteurs pré-calculés. Cette méthode ne peut toutefois pas être utilisée dans notre application puisqu'elle nécessite un ensemble d'apprentissage pour apprendre les modèles de distorsions. De plus, au moins 20 secondes de signal sont nécessaires pour produire un vecteur caractéristique. Leur algorithme ne peut donc identifier un segment audio de durée plus réduite.

Haitsma et Kalker [9] associent à chaque frame un nombre codé sur 32 bits défini à partir de la décomposition du spectre de chaque frame en bandes de fréquences avec un espacement logarithme. La séquence de bits de chaque frame est définie d'après le signe de la différence d'énergie calculée entre deux bandes consécutives d'un même frame et entre deux frames consécutifs. Plus précisément, définissons EB(n,m) comme étant l'énergie de la m^{ième} bande du n^{ième} frame et $\Delta EB(n,m) = EB(n,m) - EB(n,m+1)$ comme la différence d'énergie de deux bandes successives d'un même frame. La valeur du m^{ième} bit du n^{ième} frame (F(n,m)) est alors définie par :

$$F(n,m) = \begin{cases} 1 & Si \quad \Delta EB(n,m) - \Delta EB(n-1,m) \geq 0 \\ 0 & Sinon \end{cases}$$

Une table de correspondance est alors crée afin d'associer à chaque sous-fingerprint de la base de donnée la liste des documents audio le contenant et la position du sous-fingerprint dans chaque document. Chaque sous-fingerprint d'un fichier audio inconnu est alors comparé à la table de correspondance pour retrouver la liste de chansons et positions auxquelles elles apparaissent. La distance de Hamming est calculée entre le segment d'entrée et les chansons aux positions sélectionnées. Finalement, un seuil sur la distance de Hamming permet de décider si deux chansons sont dérivées d'un même document. Les expérimentations présentées par les auteurs montrent que leur méthode obtient de bons résultats, même après de forts taux de compression. Cependant, comme nous l'avons déjà mentionné, le framing ne garantie pas des performances robustes en cas de suppression ou de décalage temporel (section 4). Les auteurs ne montrent pas d'expérimentations pour ce type de dégradations.

3 Extraction de fingerprint robuste

Comme mentionné en Section 2, la conception d'un fingerprint à partir d'un contenu audio nécessite deux étapes. La première consiste à décomposer le signal en séquence d'intervalles. Puis, le calcul d'une valeur de sous-fingerprint intervient pour chaque intervalle. Dans cette section, nous proposons une nouvelle méthode pour chacune des étapes précédentes.

3.1 Segmentation audio

La méthode de framing assure qu'un nombre suffisant de frames est sélectionné à partir d'un signal d'entrée (Section 2). Cependant, la sélection d'une séquence de frames contigus est sensible aux opérations de suppressions de parties du signal et de décalages temporels qui peuvent être appliquées au document (Section 2 et 4). Cet inconvénient est atténué grâce au recouvrement entre frames mais n'est pas complètement résolu. D'un autre côté, les méthodes de segmentation, à base d'onsets par exemple, sont moins sensibles à ces opérations mais ne garantissent pas que suffisamment d'intervalles seront extraits dans un intervalle de temps imparti.

L'idée de base de notre méthode est de combiner les avantages respectifs des méthodes de framing et d'onsets en sélectionnant un court intervalle de temps à partir d'un intervalle plus large. L'intervalle plus court permet la détection de caractéristiques particulières du signal alors que l'intervalle plus large assure un taux minimum d'intervalles sélectionnés. Le procédé pourrait être décomposé en trois étapes (Fig. 1):

- Dans la première étape, un intervalle, appelé Intervalle d'Observation (I_o) est sélectionne au début du signal. La taille de cet intervalle est usuellement égale à quelques centièmes de secondes.
- Le signal interne à I_o est analysé pendant la seconde étape. Pendant cette étape, nous parcourons l'intervalle I_o à l'aide d'un sous intervalle de quelques millisecondes appelé Intervalle d'Energie (I_e). L'énergie de chaque intervalle est définie par l'amplitude moyenne des échantillons sur l'intervalle. L'intervalle I_e d'énergie maximale (I_{emax}) sur I_o est alors sélectionné.
- Dans la troisième étape, un dernier intervalle, appelé Intervalle de Caractérisation (I_c) est défini autour de I_{emax}. Finalement, un algorithme d'extraction de caractéristiques est appliqué sur I_c pour calculer une valeur de sous-fingerprint.

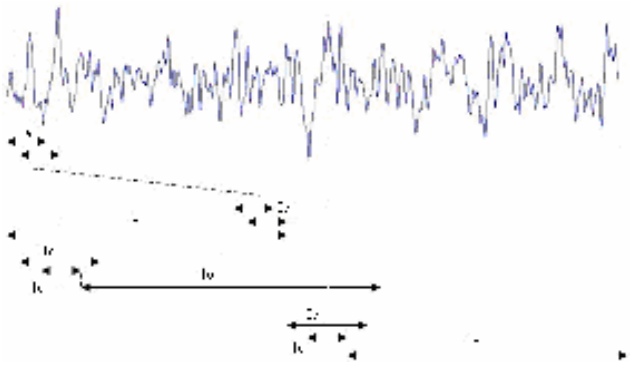


Figure 2 - *Notre méthode de segmentation.*

Partant d'un intervalle I_c sélectionné, le début de l'intervalle I_o suivant est choisi à la fin de I_{emax} (Fig. 2). La distance entre deux intervalles I_c varie alors entre I_e et $I_o - I_e$. Cette méthode apporte une plus grande robustesse envers les décalages temporels par rapport aux statégies de base qui consistent à selectionner une séquence consécutive d'intervalles I_o. En effet, en utilisant cette dernière stratégie, un I_{emax} situé à la transition entre deux intervalles I_o pourrait ne pas être détecté. De plus, notre stratégie permet de détecter plusieurs intervalles I_e, avec des énergies proches, au sein d'un même I_o. Cette dernière propriété permet d'améliorer la robustesse de notre méthode d'extraction de fingerprints. En effet, la statégie de base ne permettrait de sélectionner qu'un seul intervalle I_c. Or, une dégradation de signal pourrait échanger la sélection de deux I_e dont les énergies seraient proches. Notre stratégie renforce donc aussi la robustesse envers d'autres types de dégradations, et plus précisément la compression qui nous intéresse tout particulièrement.

3.2 Calcul de fingerprint

Notre méthode pour calculer un sous-fingerprint pour chaque intervalle I_c est basée sur le même principe que celle de Haitsma et Kalker [9] (Section 2). Comme ces auteurs, nous utilisons donc une décomposition du spectre de I_c en une suite de bandes de fréquences avec un espacement logarithme. Cependant, comme le montrent nos expérimentations (Section 4), un fort taux de compression peut significativement altérer la robustesse de cet algorithme d'extraction de sous-fingerprint. Ce dernier inconvénient interdit une comparaison directe de deux documents audio qui soit simplement basée sur le nombre de sous-fingerprint communs aux deux signaux. En effet, l'altération du signal par du bruit, une compression, ou une opération de suppression réduit drastiquement le nombre de valeurs identiques entre un document et le même document dégradé. Haitsma et Kalker résolvent ce problème en utilisant la distance de Hamming entre deux séquences de sous-fingerprint[9]. Cette stratégie impose toutefois de nombreux calculs de distance de Hamming.

Nous nous proposons d'améliorer la robustesse de l'algorithme d'extraction de caractéristiques en se basant sur les deux remarques suivantes:

- L'utilisation de deux intervalles successifs afin de calculer la valeur du sous-fingerprint implique la corruption de deux sous-fingerprint si une erreur se produit dans l'extraction des caractéristiques de l'intervalle Ic qu'ils ont en commun.
- La comparaison des énergies de deux bandes successives d'un spectre est sensible aux erreurs qui peuvent se produire sur une seule bande. On observe alors le même inconvénient qu'au point précédent entre deux valeurs basées sur l'énergie d'une même bande.

Nous résolvons la première source d'erreurs en n'utilisant qu'un seul intervalle pour chaque calcul de sous-fingerprint. La seconde source d'erreur est liée au fait que l'énergie d'une bande du spectre de I_c ayant subit trop de variation implique une valeur de sous-fingerprint erronée. En effet, en utilisant la même notation que dans la section 2, l'altération de la mesure de l'énergie d'une seule bande (EB(n,m)) altère les valeurs de $\Delta EB(n,m-1)$ et $\Delta EB(n,m)$. Cette altération des bandes d'énergie peut être considérée comme la présence d'un bruit aléatoire sur le signal EB(n,m) $_{m \in \{1,..,M\}}$ où M représente l'index de la dernière bande d'énergie.

Si on suppose que le bruit est non corrélé entre les différents échantillons du signal EB(n,m) $_{m \in \{1,..,M\}}$, une méthode basique pour réduire l'influence du bruit consiste à remplacer chaque mesure EB(n,m) par le calcul d'une valeur moyenne de EB(n,m) fonction de m. Nous définissons alors l'énergie moyenne S(n,m) d'une bande m, d'un intervalle n, comme la moyenne de toute les énergies des bandes de 0 à m:

$$S(n,m) = \frac{1}{m}\sum_{j=1}^{m} EB(n,i)$$

On remplace alors EB(n,m) par S(n,m) dans le calcul des différences des énergies des bandes. Le $m^{ième}$ bit du sous-fingerprint associé à l'intervalle n (F(n,m)) est donc défini par:

$$F(n,m) = \begin{cases} 1 & Si \quad S(n,m) - S(n,m-1) \geq 0 \\ 0 & Sinon \end{cases}$$

Notons que F(n,m) utilise uniquement les informations de l'intervalle n, Les erreurs ne se propagent donc pas. On peut alors facilement montrer que S(n,m)-S(n,m-1)=(EB(n,m)-S(n,m-1))/m. La formule précédente peut alors être simplifiée comme suit:

$$F(n,m) = \begin{cases} 1 & Si \quad EB(n,m) - S(n,m-1) \geq 0 \\ 0 & Sinon \end{cases}$$

Le sous-fingerprint pour chaque frame n est alors défini par la concaténation des M bits $F(n,m)_{m \in \{1,..,M\}}$. Le paramètre M est fixé à 32 dans nos expérimentations (Section 4). Le fingerprint du document audio est défini comme la concaténation de la séquence de sous-fingerprint.

4 Expérimentations

Notre base de données contient 357 chansons de tous genres d'approximativement 4 minutes chacune (environ 5300 valeurs par chanson). Toutes ces chansons ont été soumises à une compression/décompression MP3 à 128kps. Les versions compressées ont ensuite été décalées temporellement en ajoutant un silence d'environ 6ms au début de chaque chanson. Les intervalles I_o, I_e et I_c ont été définis respectivement à 100ms, 1ms, 80ms pour ces expérimentations.
Les taux minimum et maximum d'extraction d'intervalles I_c pour une seconde sont alors respectivement égaux à 10 et 1000 intervalles par seconde. Le taux de détection moyen d'intervalles I_c sur l'ensemble de la base de données est égal à 21,9 intervalles par seconde. L'écart type associé à cette moyenne est égal à 3,5. Les valeurs minimales et maximales calculées sur notre base de chansons sont respectivement égales à 18 et 34.

	Moyenne	**Ecart-type**	**Min**	**Max**
Nb I_c par seconde	21,9	3,5	18	34

Table 1 – *moyenne, écart-type, valeurs min et max du nombre d'intervalles I_c détectés par secondes sur notre base de données*

Les deux premières colonnes de la Figure 2 montrent la séquence de sous-fingerprint calculées par notre méthode sur une version originale puis altérée par compression d'un même contenu audio. Pour chaque valeur de F(n,m), F(n,m)=1 est représenté par un point blanc sur la ligne n de la colonne m. La troisième colonne de cette figure représente la différence (ou exclusif) entre les deux premières colonnes. Les lignes blanches signifient qu'un intervalle I_c détecté dans un signal ne l'était pas dans l'autre. On considère alors que le sous-fingerprint est erroné. On remarque alors que très peu d'erreurs apparaissent entre les deux séquences de fingerprint. Les principales différences entre les colonnes 1 et 2 sont induites par des non correspondances des intervalles. La quatrième colonne représente le fingerprint obtenu à partir d'une version compressée puis décalée du même contenu original. La dernière colonne est une comparaison entre les colonnes 1 et 4. On peut noter visuellement que l'ajout d'un décalage temporel n'augmente pas le nombre d'intervalles détectés erronés (représentés par des lignes blanches). Sur cet exemple, le taux de bit erronés est égal à 0,22%.

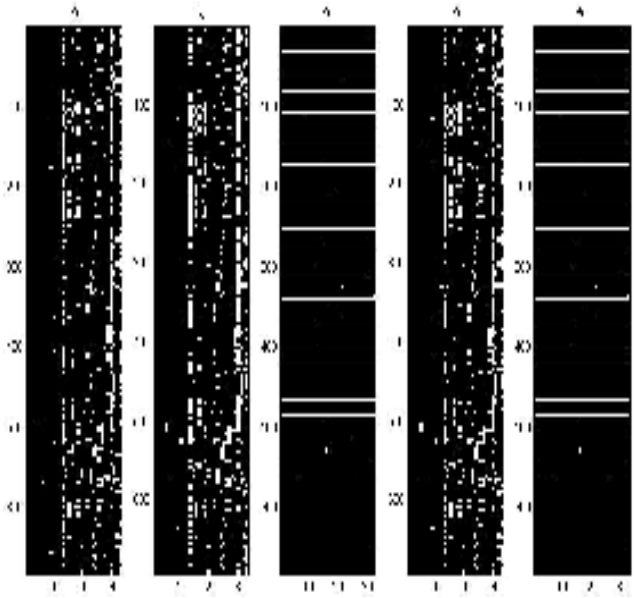


Figure 3 - *(a) fingerprint du document audio original. (b) fingerprint de la version compressée, puis décalée (d). Erreurs entre (a) et (b) puis entre (a) et (d) respectivement représentées par (c) et (e).*

Nous comparons dans la Table 1 les documents audio originaux contenus dans notre base de données avec leurs versions compressées (COM) et compressées/décalées (C&D). Les trois premières colonnes de cette table (SC, SFC et BER) sont divisées en deux sous-colonnes, chaque sous colonne représentant les performances d'un algorithme vis-à-vis des fichiers audio compressés (colonne COM) ou compressé et décalés (colonne C&D). Les colonnes SC (Segmentation Correcte) et SFC (Sous-Fingerprint Correct) représentent respectivement le pourcentage d'intervalles I_c en commun et d'intervalles communs avec des valeurs de sous-fingerprint identiques, c'est-à-dire sans un seul bit erroné, entre le fichier audio original et ses versions dégradées. La colonne BER (Bit Error Rate) représente le pourcentage de bit erronés entre les fingerprints des signaux comparés. La dernière colonne (Ko/min) correspond au nombre moyen de kilo octets nécessaires pour chaque méthode pour coder le fingerprint d'une minute de signal.

	SC		SFC		BER		Ko/ min
	COM	C&D	COM	C&D	COM	C&D	
Kalker [6]			29.9	16.9	5.8	7.3	20
Méthode Hybride	90.7	88.3	16.6	16.3	6.7	7.1	5.3
Notre Méthode	90.7	88.3	66.8	66.4	1.1	1.1	5.3

Table 2 – *Résultats d'expérimentations*

La première ligne de ce tableau illustre les performances de la méthode de Haitsma [9]. Cette méthode utilise des frames de 370ms avec un taux de recouvrement de 31/32. Les différences entre les sous-colonnes COM et C&D à l'intérieur des colonnes SFC et BER montrent la dégradation des performances de cette méthode induites par le décalage temporel. La colonne SC est laissée vide car elle n'a aucune signification pour la méthode de framing.

La seconde ligne présente les performances d'une méthode hybride combinant notre méthode de segmentation avec les calculs de sous-fingerprint proposé par Haitsma. Cette ligne montre que notre méthode de segmentation obtient en moyenne un taux d'extraction d'intervalles communs d'environ 90% (colonne SC). De plus, même si le taux de bits erronés est assez bas (colonne BER), le nombre de valeurs de sous-fingerprints sans erreurs est aussi très bas (16,6%, colonne SFC). Les erreurs sont alors parsemées sur la plupart des sous-fingerprints. On peut, de plus, noter que les performances de cet algorithme chutent légèrement après décalage temporel (colonne C&D à l'intérieur des colonnes SFC et BER). Ce comportement est principalement dû à la méthode de segmentation et au calcul de sous-fingerprint qui nécessite deux intervalles consécutifs. Notons de plus, que l'utilisation de la méthode de segmentation divise aussi par un facteur de 4 le nombre de sous fingerprints calculés (colonne Ko/min).

La dernière ligne montre les performances de notre méthode (Section 3). Le BER est beaucoup plus bas qu'en utilisant l'algorithme de calcul de sous-fingerprint de Haitsma, et plus de la moitié des sous-fingerprint (66,8%) sont extraits sans erreur. Ces performances peuvent donc permettre une comparaison de fichier audio en comptant simplement le nombre de valeurs de sous-fingerprints identiques en commun.

5 Conclusion

Nous avons présenté un nouvel identifiant audio basé sur un algorithme de segmentation du signal audio et un nouveau calcul des identifiants définissant les sous fingerprint L'algorithme de segmentation détermine des positions caractéristiques à l'intérieur du signal tout en maintenant une fréquence d'extraction de telles positions relativement constante. La fréquence d'extraction est également suffisamment élevée pour permettre une identification efficace du signal.
Notre méthode de calcul d'identifiants est basée sur l'énergie des bandes de fréquence calculées sur un court intervalle autour des positions sélectionnées. La suite des énergies de chacune des bandes est considérée comme un signal dont déduit des caractéristiques robustes au bruit susceptible de l'affecter.
La méthode de calcul d'identifiants proposée renforce la robustesse de notre méthode globale vis-à-vis de la compression du contenu. Notre méthode de sélection des positions ajoute à cela une robustesse vis-à-vis des altérations temporelles. Enfin, comparé à une simple sélection de frames consécutifs, notre méthode réduit la taille de la base de données de fingerprints.
Dans nos prochains travaux, nous envisageons de proposer une méthode d'indexation et de recherche appropriée qui, combinée à notre méthode de calcul d'identifiant, permettra une identification rapide de documents audio avec un très faible taux de faux négatif.

Références

[1] Secure Digital Music Initiative (SDMI), http://www.sdmi.org, 2001.

[2] S.Hainsworth and M.Macleod. Onset Detection in Musical Audio Signal. Dans *Proceeding of the Internationl Computer Music Conference*, 2003.

[3] F.Gouyon, A.Klapuri, S.Simon, M.Alonso, G.Tzanetakis, C.Uhle and P.Cano. An Experimental Comparison of Audio Tempo Induction Algorithms. IEEE Transactions on Audio, Speech and Language Processing. 2006.

[4] J.P.Bello, C.Duxbury, M.Davies, M.Sandler. On the Use of Phase and Energy for Musical Onset Detection in the Complex Domain. IEEE Signal Processing Letters, vol. 11, NO.6, Juin 2004.

[5] B.Logan. Mel Frequency Cepstral Coefficients for Music Modelling. Dans *Proceeding of the International Symposium on Music Information Retriaval (ISMIR)*, 2001.

[6] S.E.Johnson and P.C.Woodland. A method for direct audio search with application to indexing and retrieval. IEEE International Conference on Acoustics, Speech and Signal. 2000.

[7] F.Kurth. A ranking technique for fast audio identification. Dans *The International Workshop on Multimedia Signal Processing*, 2002.

[8] C.Burges, J.Platt and S.Jana. Distorsion Discriminant Analysis for Audio Fingerprinting. IEEE Transactions on Speech and Audio Processing. 2002.

[9] J.Haitsma and T.Kalker. A highly robust audio fingerprinting system. *ISMIR*, pages 144-148, 2002

Stratégies d'insertion informée pour un algorithme de tatouage utilisant l'interpolation bilinéaire

Vincent Martin Marie Chabert Bernard Lacaze

IRIT/ENSEEIHT

2, rue Camichel, 31000 TOULOUSE

{martin, chabert, lacaze}@enseeiht.fr

Résumé

Les techniques d'interpolation d'images ont pour propriété de préserver la qualité visuelle, qui est l'une des principales contraintes du tatouage d'image numérique. Cet article présente un algorithme de tatouage utilisant l'interpolation bilinéaire. Il s'agit d'une technique substitutive et de codage informé. Ses propriétés d'imperceptibilité, de robustesse et de sécurité ont été démontrées et comparées avec des méthodes classiques. Il est possible d'établir ses performances théoriques en présence de bruit. On s'intéresse plus particulièrement à l'utilisation de cette expression théorique dans des stratégies d'insertion informée.

Mots clefs

tatouage numérique, interpolation, insertion informée

1 Introduction

Le tatouage numérique consiste à insérer une information dans le contenu d'un document, sous les contraintes d'imperceptibilité, de sécurité et de robustesse aux attaques. Ses applications vont de la gestion des droits d'auteurs numériques à la protection d'intégrité. Dans le tatouage à étalement de spectre à Séquence Directe (DS), on module le message par une séquence pseudo-aléatoire avant de l'ajouter au document. Un corrélateur est utilisé au décodage, parfois associé à un préfiltrage de Wiener (DS+W) [1]. Le tatouage informé consiste à profiter de la connaissance du document lors de l'insertion [2]. On parle de codage informé lorsque le tatouage est construit en fonction de l'image, notamment pour respecter l'imperceptibilité. Si de plus on connaît le décodeur lors de l'insertion, on peut utiliser une stratégie d'insertion informée pour atteindre un objectif fixé en réception. Le principe du tatouage informé a notamment été utilisé dans les techniques d'étalement de spectre amélioré linéaire (LISS) [3] et dans les techniques de *binning* aléatoire [4], dont la plus populaire est le Schéma de Costa Scalaire à Transformation d'étalement (ST-SCS) [5]. Dans ces techniques, le document original n'est plus une source d'interférence.

L'interpolation [6] est souvent considérée comme une source d'erreur dans les schémas de tatouage d'image. En effet, elle est associée à un ré-échantillonnage lors d'attaques géométriques ou lors d'une insertion dans un domaine transformé. Au sein d'un algorithme de tatouage, l'interpolation n'a été utilisée que dans le cas d'objets 3D [7] ou dans un but cryptographique [8].

Les notations suivantes seront utilisées : soit $M = [m(l)]_{l\in\{1,\dots,L\}}$ le message binaire de taille L. Soit I l'image originale, W le tatouage et $I_W = I + W$ l'image tatouée. On utilise la notation matricielle suivante : $I = [i(n_1, n_2)]_{n_1\in\{1,\dots,N_1\},n_2\in\{1,\dots,N_2\}}$. I_W est transmise et peut être attaquée, devenant I'_W. Certaines attaques sont modélisées par un bruit additif blanc gaussien (AWGN) : $I'_W = I_W + B$ où $b(n_1, n_2) \sim \mathcal{N}(0, \sigma_B^2)$. Soit σ_W^2 la variance de $w(n_1, n_2)$. On définit le Rapport Document à Tatouage (DWR), le Rapport Document à Bruit (DNR) et le Rapport Tatouage à Bruit (WNR) :

$$\text{DWR} = \frac{\sigma_I^2}{\sigma_W^2},\ \text{WNR} = \frac{\sigma_W^2}{\sigma_B^2},\ \text{DNR} = \frac{\sigma_I^2}{\sigma_B^2}$$

Dans la partie 2, on introduit un algorithme de tatouage utilisant l'interpolation bilinéaire. On étudie ses performance théoriques dans la partie 3. La partie 4 s'intéresse plus particulièrement à l'utilisation de stratégies d'insertion informée. Les propriétés d'imperceptibilité, de robustesse et de sécurité sont étudiées dans la partie 5.

2 Algorithmes de tatouage informé utilisant l'interpolation

2.1 Principe général

Afin de mettre à profit les propriétés perceptuelles de l'interpolation dans un schéma de tatouage, on se propose de générer un tatouage à partir du résultat d'une interpolation. Certains pixels sont inchangés et sont utilisés pour interpoler la valeur d'autres pixels. Le schéma général de cette classe d'algorithmes, appelée W-interp, est présenté Fig. 1. Ce schéma a été proposé initialement dans [9]. Deux variantes ont été étudiées en détail : W-bilin, utilisant l'interpolation bilinéaire [10] et W-spline, utilisant l'interpolation par splines bicubiques [9].

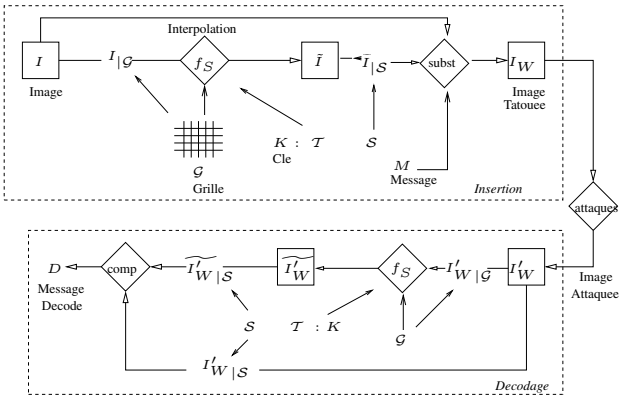


Figure 1 – *Classe de méthodes de tatouages W-interp*

Le principe de W-interp est le suivant. On sélectionne dans I deux ensembles disjoints de coordonnées respectives $\mathcal{G}$ et $\mathcal{S}$. $\mathcal{G}$ représente la grille. Le tatouage est inséré dans $\mathcal{S} \subset \{1,\dots,N_1\} \times \{1,\dots,N_2\} \setminus \mathcal{G}$. Soit $N_\mathcal{S}$ le cardinal de $\mathcal{S}$ et $P_\mathcal{S} = N_\mathcal{S}/L$ la redondance. $\mathcal{S}$ est divisé en L ensembles disjoints et choisis aléatoirement, de taille $P_\mathcal{S}$: $\mathcal{S} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_L$, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset \quad \forall i \neq j$. $\mathcal{S}_k$ est associé au bit $m(k)$ du message. De plus, tout algorithme de tatouage est associé à une clé secrète K, connue à l'insertion et au décodage, qui empêche les pirates de décoder le tatouage. Ici, soit $\mathcal{T}$ un ensemble de paramètres aléatoires spécifiquement introduits pour garantir la sécurité de l'algorithme. K est composée des coordonnées du tatouage $\mathcal{S}$ et des paramètres de sécurité associés ($K = \{\mathcal{S}, \mathcal{T}\}$). Soit $I_{|\mathcal{G}}$ la restriction de I à $\mathcal{G}$. Enfin, soit g une fonction

$$g(I_{|\mathcal{G}}; \mathcal{G}, \mathcal{T}) = \widetilde{I}$$

qui produit $\widetilde{I}$ telle que $\widetilde{I}_{|\mathcal{G}} = I_{|\mathcal{G}}$ et que I et $\widetilde{I}$ soient proches perceptuellement. Remarquons que $I_{|\mathcal{S}}$ n'est pas fournie à g. g estime des échantillons manquants à partir d'un sous-ensemble de I. g peut donc être considérée comme une fonction d'interpolation.

A l'insertion, si $m(l) = 1$ les valeurs $I_{|\mathcal{S}_l}$ des pixels de $\mathcal{S}_l$ sont substituées par leur équivalent $\widetilde{I}_{|\mathcal{S}_l}$ fourni par g, donc $I_{W|\mathcal{S}_l} = \widetilde{I}_{|\mathcal{S}_l}$. Si $m(l) = -1$, $I_{W|\mathcal{S}_l} = I_{|\mathcal{S}_l}$.
Après d'éventuelles attaques, on compare au décodage $I'_{W|\mathcal{S}}$ et $\widetilde{I'_{W|\mathcal{S}}}$. Soit $R = \widetilde{I'_{W|\mathcal{S}}} - I'_{W|\mathcal{S}}$. Pour un bit donné $m(l)$, l'erreur quadratique moyenne $\rho^2(l) = \frac{1}{|\mathcal{S}_l|}\sum_{(n_1,n_2)\in\mathcal{S}_l} r(n_1,n_2)^2$ est comparée à un seuil ν dépendant de l'image. Si $\rho^2(l) < \nu$, la décision est $d(l) = +1$, sinon $d(l) = -1$. ν est choisi empiriquement à partir du résultat du décodage : $\nu = \frac{1}{L}\sum_{l=1}^{L}\rho^2(l)$.

g sera souvent une fonction linéaire des éléments de $\mathcal{G}$. Elle agira donc comme un filtre local. L'imperceptibilité impose que W modifie les hautes et moyennes fréquences de I. g agit donc comme un filtre passe-bas, et le tatouage consiste à modifier les coefficients passe-haut de I.

Ce cadre général fournit des algorithmes de tatouage substitutifs aveugles, car I n'est pas utilisée au décodage. W-interp est une méthode à rejet des interférences de l'hôte car en l'absence d'attaque, on obtient un décodage parfait. On peut alors insérer jusqu'à $N_\mathcal{S}$ bits (débit accessible de $N_\mathcal{S}/N$). W-interp est un algorithme de tatouage informé. En effet, il utilise I pour générer W afin de respecter un modèle perceptuel, donc il s'agit de codage informé. Par contre, la seule stratégie d'insertion informée est ici le rejet des interférence de l'hôte, i.e. maximiser la détection à distortion constante et en l'absence d'attaque.

Une variante de W-interp est caractérisée par le choix d'une fonction g, d'une grille $\mathcal{G}$, des positions $\mathcal{S}$ des points à tatouer, ainsi que des paramètres de sécurité $\mathcal{T}$.

2.2 Variante W-bilin

On se limite dans cet article à la variante W-bilin, où g réalise une interpolation bilinéaire. L'interpolation bilinéaire au point (x, y) est la moyenne de ses 4 plus proches voisins sur la grille, pondérée par leur distance à (x, y) :

$$\begin{aligned} i_{int}(x,y) = \frac{y-y_1}{y_2-y_1}\left(\frac{x-x_1}{x_2-x_1}i(x_2,y_2) + \frac{x_2-x}{x_2-x_1}i(x_1,y_2)\right) \\ + \frac{y_2-y}{y_2-y_1}\left(\frac{x-x_1}{x_2-x_1}i(x_2,y_1) + \frac{x_2-x}{x_2-x_1}i(x_1,y_1)\right) \end{aligned}$$

Dans W-bilin, on substitue $i(n_1, n_2)$ par

$$\widetilde{i}(n_1,n_2) = i_{int}(n_1 + \tau_x(n_1,n_2), n_2 + \tau_y(n_1,n_2))$$

où $\tau_x(n_1, n_2)$ et $\tau_y(n_1, n_2)$ sont des variables aléatoires i.i.d. uniformément distribuées sur $]-a, +a[$. Une augmentation de a bénéficie à la robustesse et à la sécurité, mais la distortion augmente elle aussi. On choisira comme grille $\mathcal{G} = ((2\mathbb{Z}+1)\times 2\mathbb{Z}) \cup (2\mathbb{Z}\times(2\mathbb{Z}+1))$, qui a la forme d'un damier.

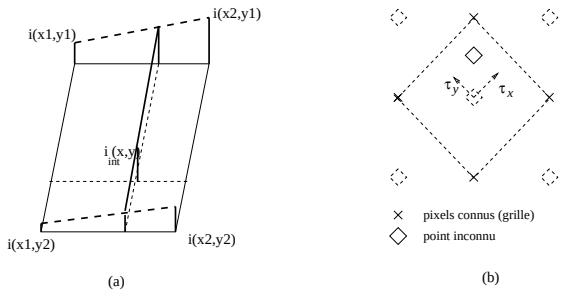


Figure 2 – *(a) Interpolation bilinéaire (b) Décalages aléatoires*

3 Performances théoriques face au bruit additif blanc gaussien

3.1 Influence de $\mathcal{T}$ sur la détection

Soit $g_{k,l}$ le poids du pixel $i(n_1-k, n_2-l)$ dans $\widetilde{i}(n_1, n_2)$. Le décodage compare $R = \{r(n_1, n_2)\}$ à un seuil, avec

$$r(n_1, n_2) = \epsilon_{I_W}(n_1, n_2) + \epsilon_B(n_1, n_2), \text{ où}$$
$$\epsilon_X(n_1, n_2) = \tilde{x}(n_1, n_2) - x(n_1, n_2)$$
$$= \sum_{k=-1}^{1} \sum_{l=-1}^{1} g_{k,l}(x(n_1-k, n_2-l) - x(n_1, n_2))$$

$$\text{Alors } \mathrm{E}[\epsilon_B] = 0 \text{ et } \sigma^2_{\epsilon_B} = (1+\Delta_{\mathcal{T}})\sigma^2_B$$

où la constante $\Delta_{\mathcal{T}}$ dépend de $\mathcal{T}$: $\Delta_{\mathcal{T}} = \sum_{k=-1}^{1}\sum_{l=-1}^{1} \mathrm{E}[g^2_{k,l}]$. Pour W-bilin, comme les élément des $\mathcal{T}$ sont uniformément répartis sur $[-a, a]$,

$$\Delta_{\mathcal{T}} = 4(\frac{1}{4} + a^2/3)^2$$

Sans décalage, $\Delta_{\mathcal{T}} = 0.25$. De plus, on peut déterminer a si $\Delta_{\mathcal{T}}$ est donné.

3.2 Détection

ϵ_I suit une loi gaussienne généralisée, ce qui permet de construire un décodeur optimal [9]. Pour simplifier, on modélise ici l'erreur d'interpolation par une distribution gaussienne : $\epsilon_I(n_1, n_2) \sim \mathcal{N}(0, \sigma^2_{\epsilon_I})$. La détection consiste en un test d'hypothèse binaire :

• hypothèse H_0 : absence de tatouage,

• hypothèse H_1 : présence d'un tatouage.

Soit P_d la probabilité de détection et P_{fa} celle de fausse alarme. Le détecteur de Neyman-Pearson maximise P_d à P_{fa} donnée. La statistique de test correspondante est ici

$$T = \sum_{\mathcal{S}} r(n_1, n_2)^2$$

Sous H_0, $R \sim \mathcal{N}(0, (1+\Delta_{\mathcal{T}})\sigma^2_B + \sigma^2_{\epsilon_I})$.
Sous H_1, la substitution à l'insertion conduit à $\epsilon_{I_W}(n_1, n_2) = 0$, donc $R \sim \mathcal{N}(0, (1+\Delta_{\mathcal{T}})\sigma^2_B)$.
R étant gaussienne centrée, T suit une distribution du χ^2_P sous les deux hypothèses. Soit $\mathrm{F}_{\chi^2_P}$ la fonction de répartition de χ^2_P. On décide H_1 quand $T < \eta$ avec

$$\eta = (1+\Delta_{\mathcal{T}})\sigma^2_B \mathrm{F}^{-1}_{\chi^2_P}(1 - \mathrm{P_{fa}})$$

Dans ce cas, $P_d = 1 - \mathrm{F}_{\chi^2_P}(\eta/((1+\Delta_{\mathcal{T}})\sigma^2_B + \sigma^2_{\epsilon_I}))$.
La distance de Kullback-Leibler D_{KL} entre les distributions de T sous H_0 et H_1 est utilisée comme borne supérieure des performances de détection, similaire à la capacité dans le problème du décodage [13]. Soit f_T la densité de probabilité de T. Ici,

$$D_{\text{KL}} = \int_{-\infty}^{+\infty} f_{T|H_0}(t) \log \frac{f_{T|H_0}(t)}{f_{T|H_1}(t)} \mathrm{d}t$$

Pour W-interp, approximons T par une loi normale, si $P_{\mathcal{S}}$ est grand (théorème central limite). On compare $\mathcal{N}(\mu_{T|H_0}, \sigma^2_{T|H_0})$ et $\mathcal{N}(\mu_{T|H_1}, \sigma^2_{T|H_1})$, donc

$$D_{\text{KL}} = \frac{1}{2}\left(\log \frac{\sigma^2_{T|H_1}}{\sigma^2_{T|H_0}} + \frac{(\mu_{T|H_0} - \mu_{T|H_1})^2 + \sigma^2_{T|H_0} - \sigma^2_{T|H_1}}{\sigma^2_{T|H_1}}\right)$$

On compare sur la Fig. 3 les performances de W-bilin et des algorithmes classiques DS, DS+W et LISS. DS et DS+W ont une performance bornée par les interférences de l'image hôte. Pour les techniques de tatouage informé W-interp et LISS, $D_{\text{KL}} \to +\infty$ lorsque le bruit diminue. W-interp est la meilleure technique en cas de bruit faible. Par contre, W-interp n'est pas robuste à un fort bruit, auquel les techniques de type DS sont plus robustes.

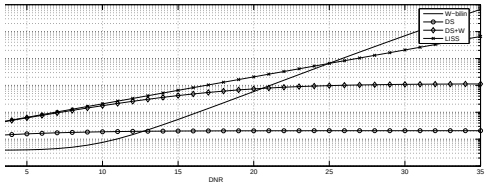


Figure 3 – *Comparaison des D_{KL}, Lena, DWR=38 dB, $N = 2^{18}$*

3.3 Décodage

Pour le décodage, on estime M à partir de I'_W grâce au Taux d'Erreur Bit (TEB) : $\text{TEB} = (1 - \sum_{l=1}^{L} \delta(d(l), m(l)))/L$. Le seuil de décision optimal η_{th} minimise le TEB. Si les bits $\{-1, +1\}$ sont équiprobables, η_{th} est solution de :

$$\frac{1}{\sigma^2_{R|H0}} f_{\chi^2_P}(\frac{\eta_{\text{th}}}{\sigma^2_{R|H0}}) = \frac{1}{\sigma^2_{R|H1}} f_{\chi^2_P}(\frac{\eta_{\text{th}}}{\sigma^2_{R|H1}})$$

La Fig. 4 montre l'amélioration apportée par η_{th} par rapport à un décodage sous-optimal avec le seuil empirique. La Fig. 5 montre qu'à WNR raisonnable, W-interp offre de très bonnes performances face au bruit AWGN.

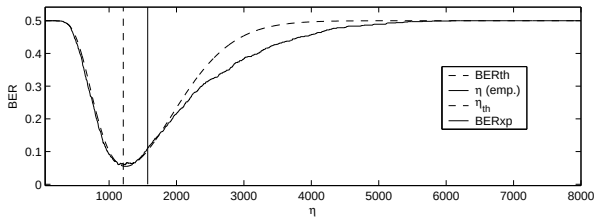


Figure 4 – *Choix de η : DWR=38 dB, $L = 1024$, WNR=-10 dB*

4 Liens avec l'insertion informée

Une contribution originale de cet article consiste à utiliser la relation entre W-interp et le tatouage informé pour améliorer ses performance. Deux principes sont abordés : la compensation des distortions et l'insertion informée.

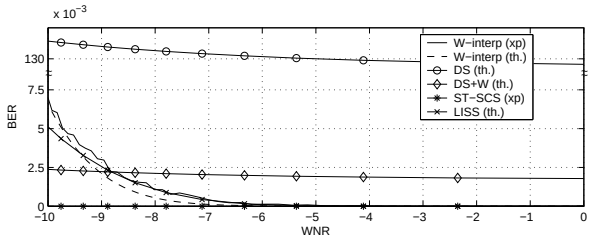


Figure 5 – *Robustesse au bruit AWGN, $L = 300$, DWR=38 dB*

4.1 W-interp et Compensation des Distortions

Dans la technique de *binning* QIM [14], l'insertion consiste à quantifier I selon un pas de quantification Δ :

$$i_W(n_1, n_2) = Q_\Delta(i(n_1, n_2) + d) - d$$

où d contient l'information sur le message et Q_Δ est l'opérateur de quantification. La technique de compensation des distortions (DC-QIM) permet d'améliorer ses performances en présence de bruit AWGN. Son principe est le suivant : changer Δ en Δ/α, avec $\alpha < 1$, permet d'augmenter la robustesse d'un facteur $1/\alpha^2$, mais la distortion augmente également en $1/\alpha^2$. Pour conserver une distortion constante, on réintroduit donc une fraction $(1 - \alpha)$ de l'erreur de tatouage :

$$\begin{aligned} i_W(n_1, n_2) &= Q_{\Delta/\alpha}(i(n_1, n_2) + d) - d \\ &\quad + (1 - \alpha)(i(n_1, n_2) - Q_{\Delta/\alpha}(i(n_1, n_2) + d) - d). \end{aligned}$$

On calcule la valeur optimale de α en fonction de σ_B^2.
Cette idée peut être appliquée à W-interp : lorsque $\Delta_\mathcal{T}$ augmente à $N_\mathcal{S}$ constant, la distance entre les distribution de T sous H_0 et H_1 augmente, mais DWR diminue. Soient $\mathcal{T}'$ des paramètres tels que $\Delta_{\mathcal{T}'} > \Delta_\mathcal{T}$. On propose la stratégie d'insertion suivante sous H_1 :

$$\begin{aligned} i_W(n_1, n_2) &= \tilde{i}(n_1, n_2) + (1 - \alpha)(i(n_1, n_2) - \tilde{i}(n_1, n_2)) \\ &= i(n_1, n_2) + \alpha(\tilde{i}(n_1, n_2) - i(n_1, n_2)) \end{aligned}$$

Sous H_1, $R \sim \mathcal{N}(0, (1 + \Delta_{\mathcal{T}'})\sigma_B^2 + (1 - \alpha)^2\sigma_{\epsilon_I}^2)$: la distortion des compensations ajoute des interférences au décodage. Supposons l'influence $\sigma^2_{\epsilon_I(\mathcal{T}')}$ de $\mathcal{T}'$ sur ϵ_I connue. A distortion constante,

$$\alpha = \sqrt{\sigma^2_{\epsilon_I(\mathcal{T})}/\sigma^2_{\epsilon_I(\mathcal{T}')}}$$

η_{th} et le TEB dépendent des variances $(1 + \Delta_{\mathcal{T}'})\sigma_B^2$ et $(1 + \Delta_{\mathcal{T}'})\sigma_B^2 + (1 - \alpha)^2\sigma^2_{\epsilon_I(\mathcal{T}')}$. On peut donc calculer numériquement $\Delta_{\mathcal{T}^*}$ qui minimise ce TEB. En pratique, on peut fournir au décodeur une clé $\mathcal{T}$ ayant une distribution uniforme sur $[-1, 1]$. A l'insertion comme au décodage, il suffira ensuite de pondérer $\mathcal{T}$ par un paramètre a pour utiliser $\Delta_{\mathcal{T}^*}$.
En l'absence de modèle simple, on calcule numériquement $\sigma^2_{\epsilon_I(\mathcal{T}')}$ pour chaque image. Pour modéliser l'influence de $\mathcal{T}$ sur ϵ_I, on propose cependant d'utiliser le modèle de Markov-Gauss suivant : la différence U entre deux pixels de I voisins est supposée gaussienne centrée ($U \sim \mathcal{N}(0, \sigma_U^2)$) [15]. Sous l'hypothèse (abusive) d'indépendance des éléments de U, on peut montrer comme précedemment pour ϵ_B que

$$\sigma^2_{\epsilon_I(\mathcal{T})} = \Delta_\mathcal{T}\sigma_U^2$$

La validité de ce modèle dépend de I. Notamment, il est bien vérifié par l'image Lena lorsque $\Delta_\mathcal{T}$ est faible (cf Fig. 6). Les courbes théoriques des Figs 7 et 8 montrent que selon ce modèle, DC-W-bilin apporte une nette amélioration des performances. Les résultats expérimentaux utilisant $\Delta_{\mathcal{T}^*}$ calculé théoriquement grâce au modèle précédent confirment l'intérêt de DC-W-bilin (cf Fig. 9).

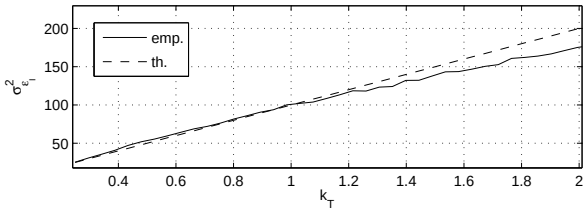


Figure 6 – $\sigma^2_{\epsilon_I}$ *en fonction de* $\Delta_\mathcal{T}$*, Lena*

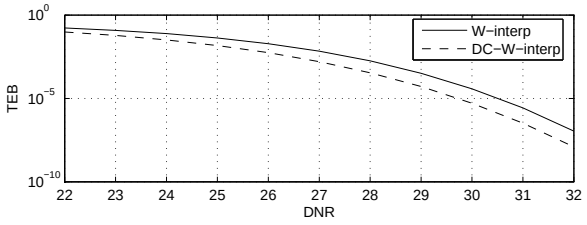


Figure 7 – *Amélioration des performances théoriques par DC-W-bilin, Lena, DWR=38 dB, $L = 256$, $P_\mathcal{S} = 178$*

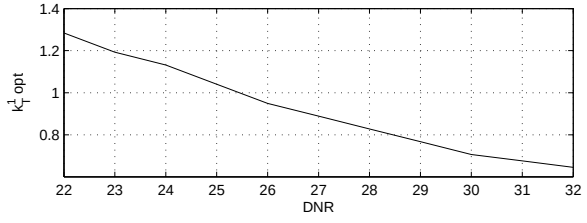


Figure 8 – *Choix optimal de $\Delta_\mathcal{T}$ pour DC-W-bilin, Lena, DWR=38 dB, $L = 256$, $P_\mathcal{S} = 178$*

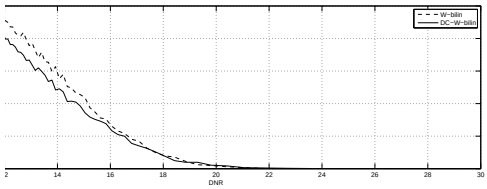


Figure 9 – *Amélioration des performances pratiques par DC-W-bilin, Lena, DWR=38 dB, $L = 256$, $P_\mathcal{S} = 178$*

4.2 Stratégies d'insertion informée

La stratégie d'insertion classique, utilisée également dans la version de base de W-bilin, consiste à maximiser la détection à distortion fixée. La connaissance du détecteur (et de ses performances théoriques) lors de l'insertion permet d'appliquer d'autres stratégies d'insertion, autour des critères de détection, distortion et robustesse [2][13]. La distortion sera mesurée ici par DWR. On choisit de mesurer la détection par la distance de Kullback-Leibler D_{KL}, dans le cas où σ_B^2 est nul. La robustesse est un critère à définir pour chaque technique. Pour DS, il s'agit de la puissance σ_B^2 de bruit qu'un pirate doit ajouter pour fausser le détecteur [2]. Pour W-bilin, le seuil η_{th} dépend déjà de σ_B^2. On préfère donc choisir comme critère de robustesse le TEB, calculé numériquement en fonction des distributions de T sous H_0 et H_1, à σ_B^2 connu.

Maximiser la robustesse à distortion constante. On veut minimiser le TEB à σ_B^2 donné et à distortion fixe. Alors DC-W-bilin constitue déjà une stratégie d'insertion pratique pour ce problème. Par contre, DC-W-bilin nuit à la détection (cf Fig 10 : sans DC, D_{KL} serait infini).

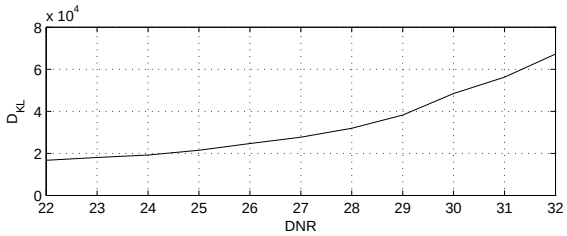


Figure 10 – *D_{KL} pour DC-W-bilin si l'attaque n'a pas lieu*

Minimiser la distortion à détection constante. Cette stratégie est inutile pour W-bilin car une détection parfaite ($D_{\text{KL}} = +\infty$) est possible pour tout DWR en changeant $N_{\mathcal{S}}$ (comme pour DC-W-interp, cette limitation de $\mathcal{S}$ peut se recalculer en réception).

Minimiser la distortion à robustesse constante. A DNR et TEB fixés, une diminution de $N_{\mathcal{S}}$, $\Delta_{\mathcal{T}}$ ou α permet de diminuer la distortion. Plusieurs stratégies sont possibles : diminuer $N_{\mathcal{S}}$, chercher le couple $(\alpha, \Delta_{\mathcal{T}})$ optimal à $N_{\mathcal{S}}$ fixé, ou bien ou effectuer une compensation des distortions sans contrepartie sur $\Delta_{\mathcal{T}}$: si $m(l) = 1$ on insère $i(n_1, n_2) + \alpha(\tilde{i}(n_1, n_2) - i(n_1, n_2))$, avec $\Delta_{\mathcal{T}} = 4/9$ et α variable. Cette technique permet d'améliorer DWR de façon significative, au prix d'une grande perte de performance de décodage (cf Fig. 11). On pourrait également combiner les trois techniques.

5 Etude de W-bilin

5.1 Imperceptibilité

La Fig. 12 montre un exemple de tatouage généré par W-bilin. Grâce à l'utilisation de l'interpolation, les plus grandes déformations sont situées dans les zones de plus

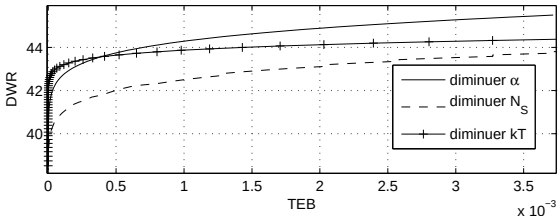


Figure 11 – *Maximisation de DWR en fonction du TEB, Lena, $L = 256$, WNR=-6 dB, $P_{\mathcal{S}} = 178$*

grande activité locale (contours, textures), là où elles sont le moins perceptible.
La qualité perceptuelle est confirmée par des mesures objectives. La Mesure de Similarité Structurelle (SSIM) [11] mesure la dégradation de l'information structurelle dans l'image, de 0 (pas de similarité) à 1 (pas de distortion). Les résultats expérimentaux (cf Tab.1) montrent que selon ce critère, W-bilin offre de meilleurs résultats que la technique DS classique combinée avec le masque de Fonction de Visibilité du Bruit (NVF) [12] ou à une insertion dans le domaine de la DCT avec un masque approprié [1].

DS	0.9827	DS+NVF	0.9897
DS+DCT	0.9897	W-bilin	0.9929

Tableau 1 – *Qualité perceptuelle selon le critère SSIM, DWR=38 dB*

La puissance d'insertion est contrôlée par le DWR. Le tatouage est imperceptible pour DWR$>$ 38 dB. Pour W-bilin,

$$\text{DWR} = \frac{2\sigma_I^2 N}{\sigma_{\epsilon_I}^2 N_{\mathcal{S}}}$$

Figure 12 – *Lena (détail) : originale, tatouée et tatouage, DWR=38 dB*

5.2 Robustesse

W-bilin est particulièrement robuste au débruitage car W est très corrélé à I, donc difficile à estimer (cf Fig. 13). Sa robustesse à la compression JPEG et à l'égalisation d'histogramme est également montrée sur les Figs. 14 et 15. Par contre, W-bilin est sensible aux attaques désynchronisantes telles qu'une rotation. Même en cas de resynchronisation, l'attaque géométrique génère un bruit d'interpolation qui gêne le décodage. Des techniques de resynchronisation exploitant les spécificités de W-bilin sont à l'étude.

5.3 Sécurité

La sécurité de W-bilin repose sur les paramètres $K = \{\mathcal{S}, \mathcal{T}\}$. S'il ne connaît pas les décalages $\mathcal{T}$, un pirate ne peut pas décoder M à partir de I_W. En effet, on peut montrer que pour la distribution de $\mathcal{T}$ utilisée ici, l'emploi de

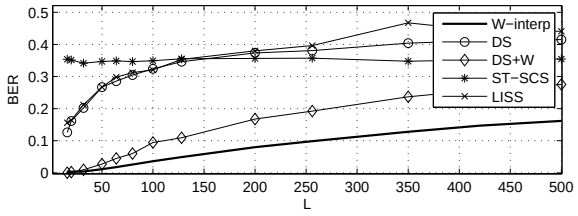


Figure 13 – *Robustesse au débruitage, DWR=38 dB*

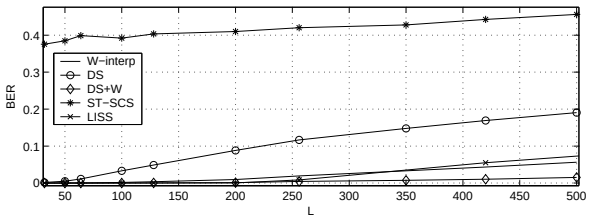


Figure 14 – *Robustesse à l'égalisation d'histogramme, DWR=38 dB*

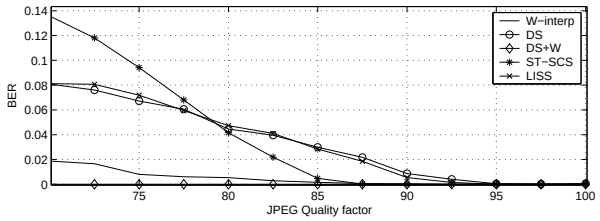


Figure 15 – *Robustesse à la compression JPEG,* $L = 64$*, DWR=38dB*

mauvais paramètres introduit un bruit d'estimation de l'interpolation de variance $\sigma_S^2 = \frac{7}{8}\sigma_{\epsilon_I}^2$. Par contre, s'il a accès à $N_o > 1$ images tatouées avec la même clé, le pirate peut essayer d'estimer K car l'insertion du tatouage a modifié la distribution d'une erreur d'interpolation. Notamment, un algorithme d'Estimation-Maximisation (EM) a été proposé pour estimer simultanément $\mathcal{S}$ et $\mathcal{T}$ [9].

6 Conclusion

W-interp est un algorithme de tatouage utilisant l'interpolation bilinéaire, qui offre des propriétés intéressantes d'imperceptibilité, de sécurité et un rejet des interférences de l'image hôte. W-interp est robuste à des attaques de faible puissance. Il s'agit d'une technique de codage informé, car le tatouage est construit à partir de l'hôte. Afin de tirer profit de la connaissance des performances théoriques du détecteur lors de l'insertion, on a proposé dans cet article d'utiliser des stratégies d'insertion informée. Notamment, la compensation des distortions, similaire à celle des algorithmes quantificatifs, permet d'améliorer significativement les performances de décodage. Elle rejoint la stratégie de maximisation de la robustesse à distortion constante.

Références

[1] J.R. Hernández et F. Pérez-González. Statistical analysis of watermarking schemes for copyright protection of images. *IEEE Proc., Special Issue on Identification and Protection of Multimedia Information*, 87(7) :1142–1166, 1999.

[2] M.L. Miller, I.J. Cox, et J.A. Bloom. Informed embedding : Exploiting image and detector information during watermark insertion. *IEEE Int. Conf. on Image Processing - ICIP*, 3 :1–4, 2000.

[3] H.S. Malvar et D.A.F. Florêncio. Improved spread spectrum : a new modulation technique for robust watermarking. *IEEE Trans. on Signal Processing*, 51(4) :898–905, 2003.

[4] P. Moulin et R. Koetter. Data-hiding codes. *Proc. of the IEEE*, 93(12) :2083–2127, 2005.

[5] J.J. Eggers, R. Bauml, R. Tzschoppe, et B. Girod. Scalar Costa Scheme for Information Embedding. *IEEE Trans. on Signal Processing*, 51(4) :1003–1019, 2003.

[6] P. Thévenaz, T. Blu, et M. Unser. Image interpolation and resampling. Dans I. Bankman, éditeur, *Handbook of Medical Imaging, Processing and Analysis*, chapitre 25, pages 393–420. Acad. Press, San Diego, USA, 2000.

[7] R. Ohbuchi, H. Masuda, et M. Aono. A Shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces. *Proc. of the Comp. Graphics Int. (CGI)*, pages 170–177, 1999.

[8] G. Boato, C. Fontanari, et F. Melgani. Hierarchical deterministic image watermarking via polynomial interpolation. *Proc. of ICIP*, 2005.

[9] V. Martin, M. Chabert, et B. Lacaze. Substitutive watermarking algorithms based on interpolation. *Proc. of EUSIPCO*, 2006.

[10] V. Martin, M. Chabert, et B. Lacaze. A novel watermarking scheme based on interpolation for digital images. *Proc. of ICASSP*, 2006.

[11] Zhou Wang, A.C. Bovik, H.R. Sheikh, et E.P. Simoncelli. Image quality assessment : From error visibility to structural similarity. *IEEE Trans. on Image Proc.*, 13 :600–612, 2004.

[12] S. Voloshynovskiy, A. Herrigel, N. Baumgartner, et T. Pun. A stochastic approach to content adaptive digital image watermarking. *International Workshop on Information Hiding*, pages 212–236, 1999.

[13] J. Delhumeau, T. Furon, N. Hurley, et G. Silvestre. Improved polynomial detectors for side-informed watermarking. *Proc. SPIE*, 2003.

[14] B. Chen et G.W. Wornell. Quantization index modulation : A class of provably good methods for digital watermarking and information embedding. *IEEE Trans. on Information Theory*, pages 1423–1443, 2001.

[15] K. Sullivan, U. Madhow, S. Chandrasekaran, et B. S. Manjunath. Steganalysis of spread spectrum data hiding exploiting cover memory. *Proc. SPIE*, pages 38–46, 2005.

Interaction multimodale multiutilisateurs avec un jeu d'échec sur grand écran

S. Carbini O. Bernier J. E. Viallet

France Télécom Recherche & Développement
Technopole Anticipa, 2 avenue Pierre Marzin,
22307 Lannion Cedex, France.

{sebastien.carbini,olivier.bernier,jeanemmanuel.viallet}@francetelecom.com

Concours Jeune Chercheur : Oui

Résumé

SHIVA (Several-Humans Interface with Vision and Audio) est une interface multiutilisateurs, non intrusive, d'interaction libre par le geste et la parole avec de grands écrans. La tête et les mains de chaque personne sont suivies en temps réel à partir d'une caméra stéréoscopique. A partir de la position 3D de ces parties du corps, le système détermine la direction pointée par chaque utilisateur et les gestes de sélection effectués avec l'autre main sont reconnus. Le geste de pointage est fusionné avec les n-best résultats issus de la reconnaissance de la parole tout en prenant en compte le contexte de l'application. Le système est testé avec un jeu d'échec où deux personnes jouent tour à tour sur un très grand écran mural. Les commandes oro-gestuelles des deux joueurs sont synchronisées et fusionnées en prenant en compte le contexte du jeu. Les commandes sont interprétées et les commandes légales ambiguës, illégales ou impossibles sont représentées de façon à fournir un feedback aux joueurs.

Mots clefs

Suivi multiutilisateurs, détection et suivi de visage et de mains, espace corporel, interface homme-machine non intrusive, multimodale, synchronisation et fusion de modalités, geste de pointage, reconnaissance de parole, jeu d'échec, interprétation de commande ambiguë, feedbacks.

1 Introduction et Travaux antérieurs

Les très grands écrans muraux peuvent être visualisés par plusieurs personnes libres de se déplacer dans une pièce et devraient permettre à plusieurs utilisateurs de travailler ensemble. Mais les utilisateurs sont limités à des interactions via des interfaces de contact et ne peuvent faire appel aux moyens naturels et efficaces de communication, tels que la voix et le geste, utilisés lorsqu'ils collaborent entre eux. SHIVA (Several-Humans Interface with Vision and Audio) est une interface multimodale conçue pour permettre à plusieurs utilisateurs d'interagir librement par le geste et la parole avec de grands écrans et nous présentons ici sa déclinaison pour deux utilisateurs.

Les auteurs de [4] présentent un suivi de personnes multiples, basé sur un filtre à particules, avec une composante visuelle et audio (position, hauteur, et état locuteur de la personne). Dans [13], les auteurs suivent deux personnes dans un environnement extérieur afin d'identifier leur interaction. Le suivi est basé sur l'extraction de blobs et sur une soustraction de fond à partir d'images monoscopiques en niveaux de gris. Dans [2], un système multi-caméra et une fusion bayésienne sont utilisés pour suivre plusieurs personnes dans une pièce. Le principal inconvénient des méthodes précédentes est que chaque personne est considérée comme un seul objet et qu'aucune information sur la position des parties du corps n'est disponible.
Dans [12], les parties du corps de plusieurs personnes sont suivies, en traitant certains cas d'occultations grâce à une technique de suivi multiple de pistes et une fonction de contrainte de cohérence de trajectoire. Mais cette technique intéressante requiert une cadence d'acquisition élevée de façon à vérifier l'hypothèse de mouvements fluides. De plus, le suivi de blobs de couleur chair se fait sans identification (tête ou main ou personne auquel il appartient). Dans [9], après une étape de segmentation basée sur la teinte chair, la tête et les mains sont localisées en s'appuyant sur des heuristiques liées à la morphologie humaine et au contexte applicatif. Le suivi temporel temps réel de plusieurs personnes est réalisé par filtrage de Kalman partiel est robuste aux occultations entre personnes. Mais ainsi que le précise les auteurs, les modèles de la teinte chair sont assez sensibles à l'environnement et la précision obtenue peut diminuer lorsque la couleur des vêtements est proche de la teinte chair ou en présence de bras nus.
Comparé aux travaux antérieurs décrits ci-dessus, SHIVA détecte et suit la tête et les mains de deux utilisateurs, comme dans [12], mais en assignant chacune des parties du corps suivies à l'une ou l'autre personne. Cette interface s'appuie sur les techniques de détection et suivi des parties du corps d'une personne décrits dans [1]. Ainsi que dans [5], une caméra stéréo est utilisée à une résolution 320x240. Le suiveur est robuste à des variations raisonnables de luminosité, aux vêtements de teinte chair et aux fonds complexes. L'ensemble des processus de détection,

de suivi des parties du corps et de détection des pertes est entièrement automatique et fonctionne en temps réel.
Aucune calibration ou adaptation préalable à un utilisateur est nécessaire. Le système conserve le même comportement alors que les utilisateurs se déplacent librement dans la pièce comme dans [15] (tant qu'il n'y a pas d'occultations entre eux). L'axe tête-main est utilisé comme convention de pointage comme dans [7]. La fonction sélection est assurée avec la seconde main ou par la reconnaissance de la parole. Un vocabulaire dédié à l'application permet d'exprimer des commandes plus directes que celles obtenues par une souris gestuelle [5], par exemple pour le contrôle multimodal d'environnement virtuel [11]. Les n-best résultats de la reconnaissance de la parole sont fusionnés avec le geste et le contexte de l'application de façon à autoriser des commandes multimodales compactes et d'obtenir une interface flexible.

2 Détection et suivi des parties du corps

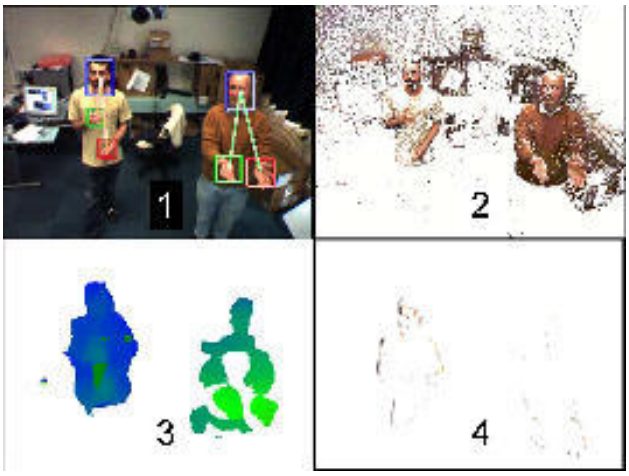


Figure 1 – *(1) Image rectifiée (les cadres représentent les parties du corps). (2) Image teinte chair (25 % des pixels). (3) Image de disparité filtrée. (4) Image du mouvement.*

La détection et le suivi des parties du corps s'appuie sur la détection de teinte chair (figure 1-2), la disparité (figure 1-3) et le mouvement (figure 1-4). La teinte chair est extraite à partir d'un filtre large construit à partir de différent utilisateurs dans différentes conditions d'éclairage. Une disparité fournie par la caméra est filtrée en éliminant les pixels situés à plus de $1.3m$ de la tête de l'une des personnes (après détection du visage). Le mouvement est estimé en soustrayant de l'image courante une image moyenne, adaptée à chaque image de façon à ce qu'une personne immobile s'intègre rapidement au fond. Les algorithmes de détection et de suivi de la tête et des mains sont issus de ceux décrits dans [1].

2.1 Espace corporel et suivi du corps

Après détection par un réseau de neurones, la position 3D de la tête sert de repère pour définir plusieurs zones impliquées dans la détection des mains et l'intentionnalité de l'utilisateur. Des contraintes morphologiques délimitent l'espace de recherche lors de la détection des mains à un sphéroïde centré sur la tête, et de rayon R ($\approx$ 1,3 m). L'espace extérieur à ce sphéroïde est écarté de l'espace de recherche (figure 2-zone 3). De plus, il est raisonnable d'admettre que, lors d'une interaction avec l'écran, l'utilisateur déplace sa main dominante vers l'écran. Ainsi, l'espace de recherche des mains est délimité par une sphère et un plan P, situé à une distance D ($\approx$ 30 cm) devant le plan parallèle à l'écran et passant par la tête. Nous appelons ce volume la zone d'action.

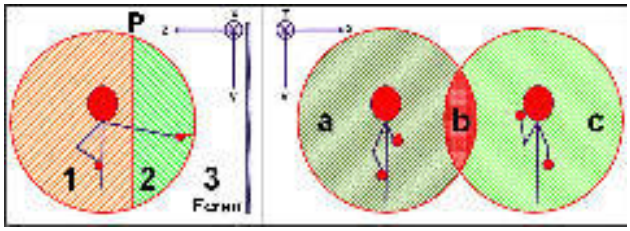


Figure 2 – *Gauche : vue de profil 1 : zone de repos, 2 : zone d'action, 3 : zone hors d'atteinte de la main. Droite : vue de face a : espace privé de l'utilisateur A , b : espace commun à A et B, c : espace privé de l'utilisateur B.*

Chaque utilisateur dispose d'un espace privé (figure 2-espace a et espace c). Si la distance entre les deux utilisateurs est trop faible, les espaces privés s'interpénètrent. L'espace commun (figure 2-espace b) est exclu de la zone de recherche des mains de manière à ne pas affecter une main à la mauvaise personne, lors de la détection.
Pour chacun des deux utilisateurs, la première main détectée est étiquetée main de pointage et la seconde, utilisée pour effectuer des gestes de sélection, est étiquetée main de contrôle. Une main est détectée en tant que zone de teinte chair en mouvement, la plus proche de l'écran, dans la zone d'action. Ainsi, SHIVA, fonctionne aussi bien pour des droitiers que pour des gauchers, sans avoir besoin d'étiquette main gauche ou main droite. Nous faisons en effet l'hypothèse qu'un utilisateur pointera d'abord un objet avec sa main dominante avant d'utiliser son autre main ou la parole pour interagir avec cet objet. La fonction de la main (pointage ou sélection) n'est activée que lorsqu'elle se trouve en zone d'action afin de prendre en compte l'intentionnalité de l'utilisateur et de ne pas déclencher involontairement des sélections incontrôlées. Ces zones et espaces étant référencés par rapport à la tête, ils accompagnent l'utilisateur lors de son déplacement tout en conservant le même comportement. Le suivi s'accommode de fonds complexes (figure 3) et est robuste à la présence de bras nus (figure 3-droite) ou de vêtements de teinte

chair (figure 1-2). Les utilisateurs peuvent être assis ou debout (figure 3-gauche) et se déplacer dans le champ de la caméra. Une fois détectée, une main est suivie même dans l'espace privé de l'autre utilisateur (figures 2, 3-centre et 3-droite), tant que cette main n'est pas automatiquement reconnue comme étant perdue, par exemple lorsqu'elle est masquée par un buste. Si l'une des deux personnes sort du champ de la caméra, une nouvelle personne entrant en scène héritera de l'étiquette de la personne sortie.

Figure 3 – *Quelques exemples de suivi : Les parties du corps suivies sont représentées par des rectangles (bleus pour la tête, rouge pour la première main et vert pour la seconde main). Les lignes blanches indiquent l'association main-tête. De gauche à droite : (1) Un gaucher assis dans le fond et une personne pointant en avant plan.(2) Les parties du corps sont correctement suivies et affectées à la bonne personne même lorsque les espaces privés se recouvrent.(3) Le suivi est robuste aux avant-bras nus et confère aux utilisateurs une grande liberté de déplacement.*

3 Le système multimodal SHIVA

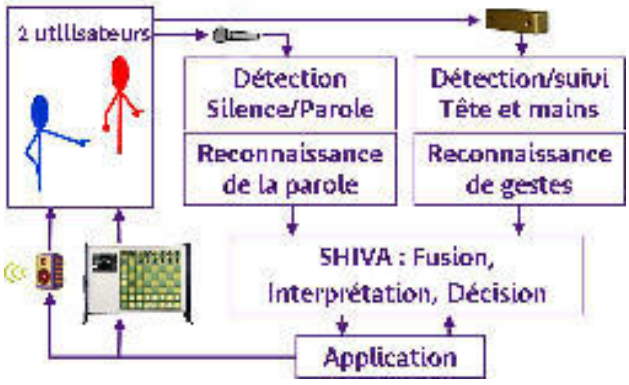


Figure 4 – *Les utilisateurs partagent les dispositifs physiques (écran, caméra, microphone) et les processus de reconnaissance oro-gestuelle de SHIVA.*

Le but du système SHIVA est de permettre aux deux utilisateurs d'effectuer des gestes de pointage et de prononcer des commandes vocales simultanément. Actuellement, SHIVA est testé sur un jeu d'échec, où les utilisateurs interagissent tour à tour au geste et à la voix (figure 5). Le tour de parole permet de n'utiliser qu'un dispositif de prise de son et qu'un processus de reconnaissance de la parole. Pour deux personnes parlant en même temps, on pourrait envisager une solution à base d'antennes acoustiques (en orientant dynamiquement chaque lobe dans la direction connue d'une personne) ou de microphones HF et d'autant de processus de reconnaissance de la parole. Par convention, la première personne qui entre dans le champ de la caméra et est détectée joue les blancs et la seconde les noirs. Un jeu d'échec [16] a été modifié de façon à accepter des commandes générées par le système multimodal SHIVA et à fournir des informations de contexte et des feedbacks adaptés à la nature multimodale des commandes. Une interface homme-machine est une boucle où les utilisateurs sont des entrées (geste et parole dirigés vers les systèmes de reconnaissance) du système et également des sorties, en tant que destinataires des feedbacks audiovisuels (figure 4). Les résultats des modules de reconnaissance de geste et de la parole sont affichés et des animations permettent aux deux joueurs de visualiser sur le grand écran l'interprétation faite par le système de leur commande multimodale, que cette commande soit légale, légale mais ambiguë, illégale voir impossible par exemple en invoquant des pièces qui ne figurent plus sur l'échiquier. Seul le lieu pointé par le joueur dont c'est le tour est représenté sous la forme d'un curseur à l'écran.

Figure 5 – *Gauche : Le joueur à gauche déplace une pièce pendant que l'autre réfléchit. Droite : Le joueur de droite joue et l'autre attend son tour. La caméra et le microphone sont au dessus de l'échiquier.*

3.1 Fusion et déplacement de pièce

Pour interpréter une commande multimodale, l'une des premières étapes consiste à synchroniser les modalités de parole et de geste, en déterminant pour chacune des modalités une référence sémantique commune et à mettre en relation les instants correspondants. Pour une interface gestuelle similaire à Shiva [8], lors d'une interaction avec une carte, 93,7% des gestes sont temporellement alignés avec l'énoncé vocal associé. Chen [3] fait l'hypothèse d'un haut degré de simultanéité entre parole et geste et fusionnent les deux modalités qu'à l'issue des processus de reconnaissance. Stiefelhagen [14] introduit un intervalle de temps de 1 s, avant et après le signal de parole, de façon à prendre en compte l'essentiel de la dynamique gestuelle. Dans notre cas de figure, le processus de pointage est

permanent tant que la main demeure en zone d'action et tous les lieux pointés sont connus en fonction du temps. Le geste de sélection est un événement discret se produisant à l'instant où la main de sélection franchit le plan P. La commande orale est un événement dont le début et la fin sont connus. La fusion geste-parole est donc réalisée dès que le résultat de la reconnaissance de parole est disponible, c'est à dire 240 ms après la fin du signal de parole. Nous faisons également l'hypothèse que geste et parole sont synchrones, c'est à dire que nous considérons les gestes de pointage réalisés pendant l'intervalle de temps de la parole (figure 6-A). Nous étendons cet intervalle d'une durée de 240 ms avant et après la parole, cette durée correspondant au temps nécessaire pour séparer un signal de parole du silence qui l'entoure. Sur un échiquier, si chaque pièce obéit à des règles de déplacement différentes, deux cases suffisent à définir un coup : la position de départ de la pièce concernée et la pièce d'arrivée. Pour déplacer une pièce, un joueur peut pointer et sélectionner successivement les deux cases concernées, fournir l'information uniquement sous la forme d'une commande orale unique du type "pion C2C3" [6] ou encore pointer une case et indiquer oralement la pièce concernée par le déplacement "met la reine". Le premier mode implique deux pointages successifs comme avec une souris, le second fait appel à des commandes surtout connue par des joueurs expérimentés et le dernier associe un seul geste naturel de pointage et une désignation ordinaire de la pièce par son nom. La reconnaissance vocale est sujette à erreurs, notamment sur des énoncés courts, tel "C2C3", erreurs qui peuvent être levées en associant le pointage sur l'une des cases.

La figure 6 présente des gestes de pointage réalisés par un utilisateur effectuant le déplacement d'une pièce, entre deux cases distinctes, sur un échiquier selon trois modalités distinctes. L'amplitude du déplacement est normalisée par la distance entre les deux cases.

La première modalité utilisée pour déplacer une pièce est analogue à la fonction glisser-déposer de la souris ; le pointage est fait avec le geste et la sélection est effectuée à la voix, par l'intermédiaire de commandes vocales "prends"-"lâche". On constate que la parole intervient sur un palier, lorsque l'utilisateur est certain que le curseur est localisé sur la bonne case (figure 6-A). On constate également que le geste de pointage demeure sur le palier, le temps que l'utilisateur perçoive que la commande multimodale a été prise en compte ; à ce moment, les changements de lieu pointés se traduisent par un déplacement de la pièce de l'échiquier visible par l'utilisateur. Le temps nécessaire au déplacement de la pièce est mesuré entre le moment où commence le pointage sur la première case (Début) et le moment où se termine le pointage sur la seconde case (Fin). Sur cet exemple, le temps de déplacement (et donc de pointage) est de 3,52 s.

La seconde modalité (figure 6-B) consiste à synchroniser le geste de pointage avec l'instant où se produit un geste de sélection, réalisé lorsque la seconde main franchit le plan P d'arrière en avant. On constate que ce geste de sélection (indiqué par une verticale) se produit également sur un palier de pointage. La fusion est effectuée en considérant que la case sélectionnée est celle qui correspond au lieu pointé au moment du geste de sélection. La pièce, correspondant à la case sélectionnée, est déplacée puis déposée par l'utilisateur sur une autre case, lorsque la seconde main franchit le plan P d'avant en arrière. Sur cet exemple, le temps de déplacement est de 3,26 s.

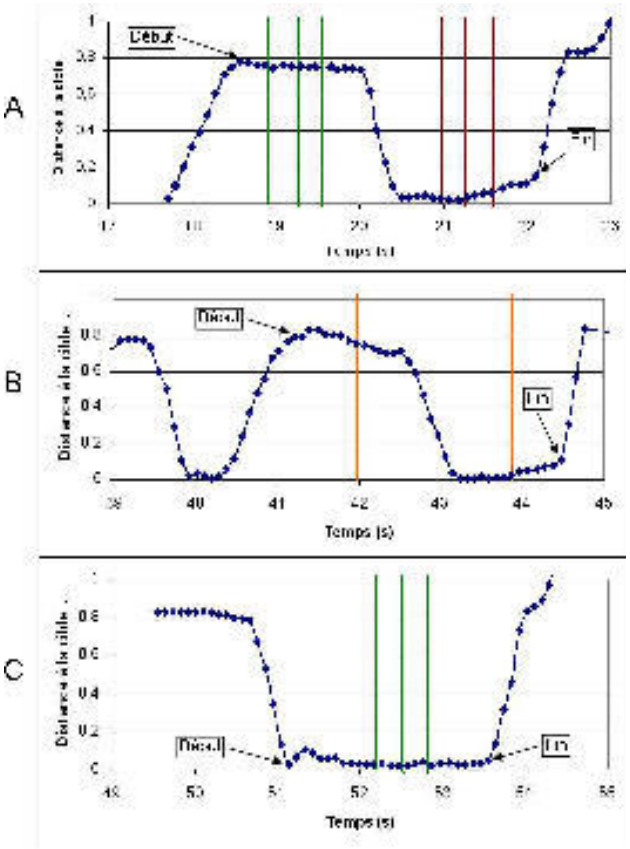


Figure 6 – *Trois façons de déplacer une pièce de l'échiquier. Les temps de déplacement sont calculés entre les instants Début et Fin. (A) La courbe représente les gestes de pointage vers la case de destination et les commandes orales "prends"/"lâche" sont représentées par deux groupes de trois verticales représentant les instants de début et de fin de parole et l'instant où le résultat de la reconnaissance de parole est disponible. (B) : la courbe représente les gestes de pointage et les verticales les instants où se produisent les gestes de sélection et de désélection. (C) Commande multimodale associant gestes de pointage (courbe) et commande orale unique (groupe de trois verticales). Par commodité, les gestes de pointage sont représentés, a posteriori, par la distance entre le lieu courant de pointage et le centre de l'une des deux cases concernées par le déplacement et cette distance est normalisée par la distance entre les deux cases.*

La troisième modalité permettant de déplacer une pièce (figure 6-C) consiste, comme pour la première modalité, à synchroniser le geste de pointage avec l'intervalle de temps correspondant à la commande orale, dont le contenu est suffisant pour effectuer le déplacement sans avoir à préciser par le geste la position de la seconde case. Sur cet exemple, le temps de déplacement est de 2,41 s. Cette modalité, qui n'implique qu'une seule case pointée mais fait appel au contexte de l'application, est réalisée dans un temps plus court que le temps des deux autres modalités qui ne font pas appel au contexte de l'application pour réaliser la fusion oro-gestuelle.

3.2 Fusion et contexte applicatif.

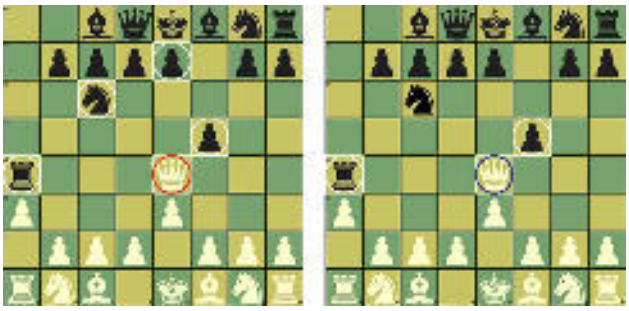


Figure 7 – *Gauche : Quand le joueur qui joue les blancs pointe sa reine blanche (cercle rouge), le contexte correspondant inclut les pièces (cercles blancs) pouvant être prises par la reine : une tour, un cavalier, deux pions. La commande "prend le pion" est ambiguë. Droite : Dans la même configuration, si le joueur qui joue les noirs pointe la reine blanche adverse (cercle bleu), le contexte correspondant inclus ses pièces noires (cercles blancs) qui peuvent prendre la reine blanche : une tour et un pion. La commande orale "prends avec le pion" n'est pas ambiguë.*

En pointant sur une case, un joueur indique l'une des deux cases relatives à un coup. L'information relative à la seconde case peut être spécifiée par la parole. En observant la configuration du jeu, qu'il suppose partagée par le système, le joueur fournit juste l'information qu'il estime suffisante pour compléter l'information manquante. SHIVA doit déterminer quelle est la seconde case concernée, à partir des informations extraites de la reconnaissance de la parole et du contexte du jeu. Le contexte de l'application est principalement lié à la case pointée par le joueur actif selon qu'il joue les blancs ou les noirs (figure 7).
- Lorsque le joueur pointe l'une de ses pièces, le contexte est essentiellement décrit par la liste des pièces adverses prenables (ou les cases qu'elles occupent) par la pièce pointée .
- Quand un joueur pointe une pièce adverse ou une case vide, le contexte est principalement décrit par la liste de ses pièces susceptibles de prendre la pièce adverse pointée ou de se déplacer vers la case vide pointée.
Lorsqu'un joueur prononce le nom d'une pièce du contexte applicatif et si une seule pièce porte ce nom, alors la seconde case est connue et SHIVA effectue le déplacement. Si plusieurs pièces du contexte portent le même nom que le mot prononcé (par exemple les pions de la figure 7-gauche), il y a ambiguïté. SHIVA l'illustre par un déplacement fictif de la reine vers les deux pions (figure 8) avant de revenir à la situation d'origine et d'attendre une nouvelle commande. Quand le joueur ne pointe aucune des cases de l'échiquier, toute l'information nécessaire doit être contenue dans le contexte et dans la parole par exemple sous la forme du nom de deux pièces non ambiguë et d'un verbe tel que "prends". Par exemple l'énoncé oral "reine prends tour" ne fait pas appel au pointage car une seule tour peut être prise par la reine (figure 7-gauche). L'ambiguïté du seul énoncé "reine prends pion" peut être levée en pointant l'un des deux pions.
Le contexte est complété par l'ensemble des coups qui ne respectent pas les règles de déplacement des pièces ou des prises. En effet, rien n'empêche un joueur d'effectuer une commande illégale (par exemple en ne réalisant pas que son roi est en prise). Si le système ne réagit pas à une commande illégale, le joueur peut penser que sa commande est légale mais qu'elle n'a pas été comprise par le système et le joueur réitérera sa commande. En revanche, si le système déplace la pièce du joueur conformément à la commande illégale et replace la pièce à sa position d'origine, le joueur prendra conscience que le système a interprété la commande du joueur mais que le système refuse de la valider.

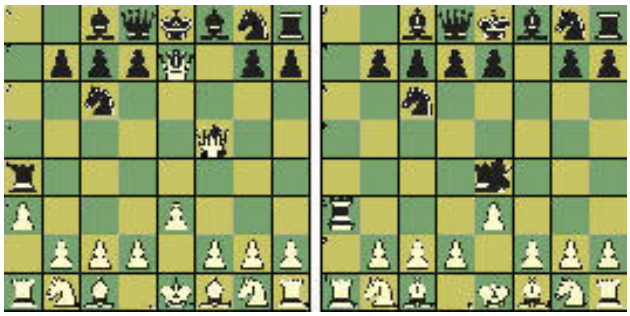


Figure 8 – *Gauche : En pointant la reine blanche, le joueur blanc visualise l'interprétation de la commande orale ambiguë "prends le pion". Droite : Le joueur noir visualise l'ambiguïté de la commande orale "prends" sans pointage. Puis dans les deux cas, la commande est rejetée et la situation d'origine est affichée.*

De la même manière, considérons la situation où un joueur effectue une commande légale, mais que le système l'interprète à tort comme une commande illégale (par exemple en raison d'une erreur de la reconnaissance vocale). En

l'absence d'illustration de la commande interprétée, le joueur ne pourrait déterminer l'origine de la non prise en compte de sa commande. Le système ne peut qu'illustrer la commande qu'il a interprété, à tort ou à raison. Le feedback permet au joueur de comprendre que le système s'est trompé et qu'il a intérêt à réaliser le même but selon une modalité différente.
Enfin le contexte est éventuellement complété par une liste de coups impossibles. En effet, la reconnaissance de la parole peut reconnaître à tort ou à raison une commande vocale impliquant une pièce qui n'existe plus sur l'échiquier. Le système doit savoir qu'une telle pièce n'existe pas, puis matérialiser la commande reconnue en faisant apparaître la pièce inexistante et enfin puis afficher la disposition précédant la commande de façon à ce que l'utilisateur dispose d'un retour lui permettant de comprendre l'interprétation faite par le système de la commande.

4 Conclusion

Nous présentons un système permettant une interaction oro-gestuelle de deux personnes jouant aux échecs sur un grand écran. Les gestes de pointage et de sélection sont obtenus en suivant les parties du corps de deux personnes en quasi temps réel. La précision du suivi confère un pointage précis sur un grand écran. Le système de suivi fonctionne à une cadence de 20 Hz lorsqu'un utilisateur est suivi et à 15 Hz pour deux utilisateurs (Biprocesseur Xeon 3,4 GHz). La faible différence du cout du suivi est encourageante pour le suivi de plus de deux personnes.
Associée à la reconnaissance de la parole, l'interface multimodale oro-gestuelle SHIVA, permet à deux personnes de jouer, tour à tour. La prise en compte du contexte permet de réaliser des commandes plus rapides et plus intuitives mais nécessitent d'adapter le vocabulaire à l'application alors qu'une interface du type souris oro-gestuelle bénéficie d'une interface graphique répandue, ne demande pas de modifications de l'application mais ne permet pas de bénéficier de commandes oro-gestuelles spécifiques.

Références

[1] S. Carbini, L. Delphin-Poulat, L. Perron, O. Bernier, J.E. Viallet, Interaction Multimodale Oro-Gestuelle Personne Libre, Corésa 2005, p. 195-200, Rennes, France, 2005.

[2] T.H. Chang, S. Gong, Tracking multiple people with a multi-camera system, IEEE Workshop on Multi-Object Tracking, p. 19-26, Vancouver, Canada, 2001.

[3] F. Chen, E. Choi, J. Epps, S. Lichman, N. Ruiz, Y. Shi, R. Taib, M. Wu, A study of manual gesture-based selection for the PEMMI multimodal transport management interface, *ICMI (International Conference on Multimodal Interfaces)*, p. 274-281, Trento, Italie, 2005.

[4] N. Checka, K. Wilson, M. Siracusa, T. Darrell, Multiple Person and Speaker Activity Tracking with a Particle Filter, ICASSP, Montréal, Canada, 2004.

[5] D. Demirdjian, T. Darrell, 3-D Articulated Pose Tracking for Untethered Diectic Reference, Proceedings of International Conference on Multimodal Interfaces, p. 267-272, Pittsburgh, Pennsylvanie, 2002.

[6] M. Gabsdil, Combining Acoustic Confidences and Pragmatic Plausibility for Classifying Spoken Chess Move Instructions, Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue, p. 27-30, Cambridge, Massachussets, 2004.

[7] R. Kehl and L. Van Gool, Real-time Pointing Gesture Recognition for an Immersive Environment, IEEE International Conference on Automatic Face and Gesture Recognition, p. 577-582, Séoul, Corée, 2004.

[8] S. Kettebekov, R. Sharma, Understanding Gestures in a Multimodal Human Computer Interaction, International Journal of Artificial Intelligence Tools, vol. 9, n. 2, p. 205-223, 2000.

[9] V. Girondel, L. Bonnaud, A. Caplier, A Human Body Analysis System, EURASIP Journal on Applied Signal Processing, vol. 2006, 2006.

[10] A. Micilotta, R. Bowden, View-based Location and Tracking of Body Parts for Visual Interaction, British Machine Vision Conference, p. 849-858, Kingston, Royaume-Uni, 2004.

[11] K. Moustakas, D. Tzovaras, S. Carbini, O Bernier, J.E. Viallet, S. Raidt, M. Mancas, M. Dimiccoli, E. Yagci, S. Balci, E. Ibanez Leon and M.G. Strintzis, "MASTERPIECE : Experiencing Physical Interaction in VR Applications", IEEE Multimedia, p. 92-100, Volume 13, Issue 3, July-September 2006.

[12] E. Polat, M. Yeasin, R. Sharma, A Tracking Framework for Collaborative Human Computer Interaction, International Conference on Multimodal Interfaces, p. 27-32, Pittsburgh, Pennsylvanie, 2002.

[13] K. Sato, J.K. Aggarwal, Tracking and recognizing two-person interactions in outdoor image sequences, Workshop on Multi-Object Tracking, p. 87-94, Vancouver, Canada, 2001.

[14] R. Stiefelhagen, C. Fuegen, P. Gieselmann, H. Holzapfel, K. Nickel, A. Waibel, Natural Human-Robot Interaction using Speech, Gaze and Gestures, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p 2422-2427, Sendai, Japon, 2004.

[15] Y. Yamamoto, I. Yoda, K. Sakaue, Arm-Pointing Gesture Interface Using Surrounded Stereo Cameras System, ICPR (International Conference on Pattern Recognition), p. 965-970, Cambridge, Royaume-Uni, 2004.

[16] Xboard : http ://www.tim-mann.org/xboard.html.

Prédiction objective des différences de qualité perceptuelles entre un écran CRT et un écran LCD en TVHD

Sylvain Tourancheau Mathieu Carnec Stéphane Péchard Patrick Le Callet Dominique Barba

IRCCyN - Équipe Image Vidéo Communications

École polytechnique de l'université de Nantes
Rue Christian Pauc, BP 50609, 44306 Nantes Cedex 3

{[prenom].[nom]}@univ-nantes.fr

Résumé

Cet article présente la technologie LCD et son impact sur la télévision haute définition. Des tests subjectifs mettant en jeu des observateurs humains sont décrits. Ces tests montrent que la qualité d'image perçue est plus importante sur écran CRT que sur écran LCD. Cette différence de qualité peut s'expliquer par un défaut lié à la technologie LCD : le flou de mouvement. Cet article présente les causes de ce défaut et un modèle mathématique permettant de le calculer. Sur des séquences avec de forts mouvements, un modèle a été construit qui permet de prédire la baisse de qualité due à un écran LCD (par rapport à un écran CRT) pour une séquence donnée, en fonction de la quantité de flou mesurée sur cette séquence.

Mots clefs

LCD, TVHD, flou de mouvement, qualité, tests subjectifs.

1 Introduction

A l'heure où la TVHD s'apprête à être déployée en Europe, de nouvelles technologies d'affichage font leur apparition (LCD et plasma) et la technologie jusqu'ici employée (CRT) tend à disparaître. La technologie LCD semble plus prometteuse étant donné les difficultés rencontrées pour construire des écrans plasma avec un nombre important de pixels. Mais ces écrans LCD souffrent de nombreux défauts d'affichage. En particulier, un flou dû au mouvement gêne l'utilisateur qui perçoit alors les images comme ayant une qualité inférieure à celles qui seraient affichées sur une technologie CRT.

Cet article présente les différences fondamentales entre les écrans CRT et les écrans LCD. Un modèle permettant d'estimer le flou de mouvement est proposé. Des tests subjectifs d'évaluation de qualité sont décrits. Ces tests montrent une différence entre la qualité perçue avec un écran CRT et celle perçue avec un écran LCD. La corrélation entre l'importance du flou de mouvement et cette différence de qualité est mise en évidence.

2 Télévision haute définition et nouvelles technologies d'affichage

2.1 Vers la haute définition

La télévision a toujours souffert d'un manque de profondeur et d'immersion par comparaison au cinéma. Des tests subjectifs datant du début des années 1980 [1, 2] ont montré que la distance optimale d'observation d'images en mouvement est comprise entre trois et quatre fois la hauteur de l'écran ($3H \leqslant D \leqslant 4H$). C'est la distance d'observation moyenne d'une salle de cinéma. L'angle de vue correspondant est d'environ 30 degrés horizontalement par 20 degrés verticalement, ce qui représente 25% du champ visuel. L'immersion et la sensation de profondeur sont donc considérablement accrues.

La résolution de la télévision standard (TVSD), 768×576 en mode entrelacé, est telle que la distance optimale d'observation est $D = 6H$. Plus près de l'écran, la structure des lignes du téléviseur devient perceptible et nuit grandement au confort de visualisation et à la qualité des images affichées. A cette distance, le téléviseur n'occupe qu'une faible proportion du champ visuel.

La TVHD a été developpée de manière à ce que l'utilisateur final puisse éprouver des sensations proches de celles vécues durant une projection cinématographique.Pour jouir pleinement des améliorations apportées par les résolutions haute définition, classiquement 1920×1080 en mode entrelacé (1080i) et 1280×720 en mode progressif (720p), il convient donc de visualiser les diffusions à une distance comprise entre $3H$ et $4H$, H étant la hauteur de l'écran. La taille de l'écran peut donc être considérablement augmentée sans que la distance d'observation ne pose problème.

2.2 Écrans à cristaux liquides (LCD)

L'écran à tube cathodique (CRT pour *Cathodic Ray Tube*), avec plus de quatre-vingts ans d'existence, est une technologie mature et parfaitement adaptée à l'affichage de contenus télévisuels. Cependant, ce type d'écran devient lourd et très encombrant au-delà d'une certaine taille. De nouveaux

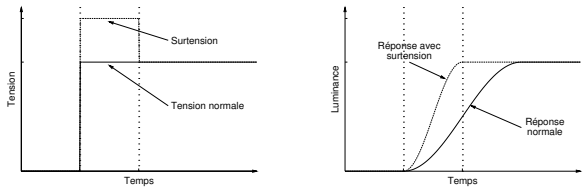
types de téléviseurs ont donc fait peu à peu leur apparition et leurs qualités en terme de taille, d'encombrement, de design, de consommation électrique et de prix devraient leur permettre de supplanter totalement les écrans CRT dans les années à venir. Parmi toutes ces nouvelles technologies, c'est l'écran à cristaux liquides (LCD pour *Liquid Crystal Display*) qui semble promis au plus bel avenir.
La technologie d'affichage par cristaux liquides a énormément évolué au cours des dix dernières années, allant même jusqu'à supplanter la technologie CRT sur le marché des moniteurs de bureau. Les écrans LCD ont atteint une qualité plus que satisfaisante en terme de reproduction de couleur, d'angle de vue et de contraste pour permettre l'affichage d'images fixes ou avec peu de mouvements. Cependant, de nombreux efforts restent à fournir concernant l'affichage de séquences vidéos. A ce titre, la technologie LCD ne peut être considérée comme une technologie pleinement adaptée à la télévision et ses caractéristiques doivent encore être étudiées et modifiées en vue d'égaler un jour la qualité d'affichage d'un écran CRT.
Des tests subjectifs ont été menés courant 2005 sur un panel de 36 observateurs (experts du domaine de la diffusion) dans le but de comparer la qualité d'affichage entre un écran CRT et un écran LCD [3]. Ces tests ont montré que, dans l'ensemble, la majorité des observateurs ont jugé la qualité des images affichées sur LCD inférieure à celle des images affichées sur CRT. De nombreux défauts ont par ailleurs été recensés par les utilisateurs. Le flou de mouvement, en dépit de récentes améliorations, reste le défaut le plus gênant pour les séquences comportant des mouvements assez rapides. C'est pourquoi nous avons choisi d'axer notre étude principalement sur cette dégradation.

3 Le flou de mouvement sur LCD

Sur un écran LCD, la mise à jour de la valeur de chaque pixel se fait progressivement, pixel par pixel, ligne par ligne. A chaque instant, on a donc une image constituée d'une partie de l'image précédente et d'une partie de l'image actuelle, la frontière entre les deux se déplaçant sur l'écran progressivement. Cette frontière sera d'autant plus gênante que le temps de réponse d'un pixel sera long.
En outre, le flou de mouvement (en anglais *motion blur*) est causé par la réponse de l'écran LCD, qui n'est pas de type impulsionnel comme sur les écrans CRT. Cette réponse est une combinaison entre la réaction lente des cristaux liquides composant l'écran, et le comportement de type *sample and hold* de celui-ci. Sur ce type d'affichage, la luminance d'un pixel est constante (une fois la valeur atteinte) durant toute la durée d'affichage d'une image.

3.1 Temps de réponse

Le temps de réponse est une caractéristique importante des écrans LCD. Pendant l'affichage d'une source progressive cadencée à 50 images par seconde, la durée d'affichage de l'image est de 20 ms, il est donc nécessaire d'avoir un temps de réponse faible par rapport à cette durée si l'on veut que l'image soit affichée correctement. Les premiers écrans LCD, dont le temps de réponse gris-à-gris moyen se situait entre 20 ms et 30 ms, ne satisfaisaient pas à cette règle, d'où un effet de rémanence très gênant particulièrement pour l'affichage d'images en mouvement. Pour corriger ce défaut, les dernières générations d'écrans à cristaux liquides sont équipés de systèmes de compensation du temps de réponse, ce qui leur permet d'atteindre des temps de réponse gris-à-gris moyens de l'ordre de 10 ms.
Cette compensation est réalisée en appliquant une surtension (resp. une soustension) à la tension de commande du pixel dans le cas d'une augmentation (resp. une diminution) de la luminosité [4]. La figure 1 illustre ce principe.



(a) Tension de commande. (b) Luminosité associée.

Figure 1 – *Compensation du temps de réponse par application d'une surtension.*

3.2 Affichage par maintien

En dépit de ces améliorations récentes et efficaces, le flou de mouvement demeure un problème important, même sur les écrans dont les temps de réponse sont les plus faibles. En effet, la part la plus importante du flou de mouvement est causée par la nature même de l'affichage du LCD. Sur ce type d'écrans, la lumière émise est maintenue durant toute la durée d'une image, on parle alors d'affichage par maintien (en anglais *hold-type display*). Cet affichage est fondamentalement différent de celui des écrans CRT, constitué d'impulsions, pour lesquels chaque pixel est affiché durant une très faible période (cf. figure 2). La perception du flou de mouvement s'explique alors par le comportement du système visuel humain.

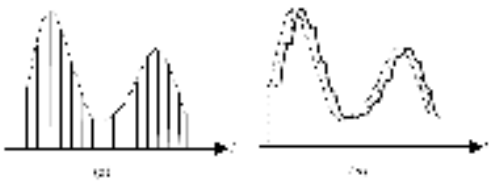


Figure 2 – *Evolution temporelle d'un pixel sur un écran CRT (a) et sur un écran LCD (b).*

Considérons l'affichage d'une séquence sur un écran LCD. Pendant la durée d'affichage d'une image, celle-ci est maintenue à l'écran. Les objets en mouvement contenus

dans la séquence sont donc immobiles durant ce laps de temps. Cependant, l'œil de l'observateur continue à se déplacer à la vitesse de l'objet, anticipant de cette manière son déplacement dans l'image suivante. Les contours de l'objet sont donc intégrés temporellement par l'œil alors même qu'ils se déplacent sur la rétine (si on se place dans un référentiel rétinien) [5]. Ces contours sont donc perçus de manière floue par l'observateur, de la même manière qu'un cliché photographique se révèlera flou si l'appareil a bougé durant le temps d'ouverture de l'obturateur.

3.3 Modélisation et tests psychophysiques

Une modélisation mathématique du flou de mouvement sur écran LCD à été réalisée par Pan *et al* [6]. Ce modèle dépend du type de fonction de recontruction temporelle de l'écran, i.e. de la réponce indicielle d'un pixel. La taille d'un flou engendré par le déplacement d'un contour peut donc être mesurée pour différents types de réponse. Dans le cas d'une réponse de type sinusoïdal, qui constitue une bonne approximation de la réalité, cette taille L s'exprime en pixels, en fonction de la vitesse V du déplacement en pixels par image :

$$L = aV, \qquad \text{avec} \quad a = 1.044. \tag{1}$$

Dans le but de valider ce modèle, nous avons décidé d'effectuer des tests psychophysiques durant lesquels les observateurs devaient mesurer précisément l'extension du flou de mouvement engendré par le déplacement d'une série de barres horizontales ou verticales. Le protocole original de cette expérience a permis d'obtenir des résultats stables avec un intervalle de confiance réduit [7]. Un ajustement linéaire de nos résultats nous a conduit à la relation :

$$L = aV, \qquad \text{avec} \quad a = 1.038. \tag{2}$$

Ces résultats subjectifs sont donc très proches de ceux prédits par le modèle de Pan. Le coefficient de corrélation entre le modèle et nos résultats est de 0.9987.

4 Evaluation subjective de la qualité sur CRT et sur LCD

Afin de pouvoir comparer la qualité perçue sur un écran CRT avec celle perçue sur un écran LCD, des tests psychovisuels de qualité ont été mis en place. Au cours de ces tests, des séquences HD ont été présentées aux observateurs qui doivent en évaluer la qualité. Une première série de tests a été réalisée sur un écran CRT pour l'ensemble des séquences. Puis, une seconde série de tests, pour les mêmes séquences, a été effectuée sur un écran LCD. Nous disposons donc d'une note de qualité (moyenne des notes attribuées par chaque observateur) pour chaque séquence et pour chacun des deux écrans CRT et LCD.

4.1 Conditions

Les tests ont été réalisés dans une pièce spécifique. Les conditions de lumière, les paramètres d'affichage et la distance d'observation ont été mesurés et ajustés suivant les recommandations de l'ITU [8, 9].
Les écrans TVHD utilisés étaient un JVC DT-V 1910CG et un Philips T370HW01. Ces deux écrans peuvent afficher le format 1080i. Le premier est un écran CRT de hauteur d'image H égale à 20,5 cm (ce qui équivaut à une diagonale de 16,5"). Le second est un écran LCD de hauteur 46 cm (diagonale de 37"). La distance d'observation a été fixée à $3H$ soit respectivement 31.5 cm et 138 cm.

4.2 Séquences

Douze séquences ont été utilisées. Quatre d'entre elles proviennent de la chaîne suédoise SVT et ont déjà été utilisées dans une étude sur la qualité HD [10], elles sont disponibles sur Internet. Les huit autres séquences proviennent de la chaîne belge Euro1080 et ont été obtenue par le biais du projet européen ITEA-HD4U dans le cadre duquel est réalisée notre étude. Chaque séquence est constituée de 250 images, pour une durée de 10 secondes.
Chacun des douze contenus a été dégradé par compression H.264 à différents débits, de manière à couvrir complètement la gamme de qualité recherchée. Sept débits de compression différents ont été produits par séquence. Ils ont été définis par des observateurs experts de manière à couvrir une gamme de qualité intéressante pour la TVHD. Le codage H.264 a été réalisé en utilisant le codec de référence [11].
L'évaluation subjective de qualité pour différents débits a été réalisée dans le but de comparer cette qualité avec celle de la télévision standard. Cependant, dans notre étude, seule la baisse de qualité introduite par la différence d'écran nous intéresse. Nous nous sommes donc restreints aux douze séquences de référence (non dégradées).

4.3 Observateurs

Avant la première séance d'évaluation, chaque observateur a passé des tests d'acuité et de détection de daltonisme conformément aux normes de l'ITU [8]. L'acuité est testée grâce aux planches de Monoyer [12]. Les déficiences chromatiques sont détectées à l'aide des tests d'Ishihara [13]. Les candidats ayant au moins une erreur au test d'Ishihara ou moins de 9/10 au test d'acuité sont rejetés.
Pour ces tests, 21 observateurs valides ont été utilisés (un minimum de 15 est recommandé [8]). Ils ont une moyenne d'âge de 25 ans et sont majoritairement étudiants.

4.4 Protocole

L'évaluation nécessitait une construction précise du jugement des observateurs. Des différences de qualité très faibles devaient être détectées, la méthode d'évaluation doit donc forcer à la discrimination. À l'heure actuelle, la méthode la plus performante est la méthode SAMVIQ [14], développée par France Télécom R&D et standardisée par l'Union européenne de radio-télévision (EBU).
SAMVIQ (pour *Subjective assessment methodology for video quality*) est une méthode d'évaluation de stimuli multiples et à échelle de qualité continue. Avec cette

méthode, l'observateur choisit l'ordre dans lequel il visionne les séquences et peut suivre son propre rythme pour l'évaluation, la modification des notes et la répétition des séquences. Le nombre de visionnage pour chaque séquence n'est pas limité. Les observateurs comparent ainsi les séquences (séquences dégradées et séquence de référence cachée) entre elles et avec la référence explicite, ce qui permet une mesure précise de la qualité de chaque séquence. L'échelle de notation est continue, chaque note pouvant prendre une valeur comprise entre 0 et 100.
La consistance des résultats est évaluée après l'ensemble des tests, par l'application d'une méthode de rejet par corrélation de rang normalisée par l'EBU. Les éventuels observateurs incohérents sont ainsi detectés et leurs résultats retirés. Suivant les séquences, entre 15 et 19 personnes ont été gardées par ce critère.

4.5 Résultats

Le tableau 1 présente le MOS (pour *Mean Opinion Score*), c'est-à-dire la note de qualité subjective moyenne fournie par les observateurs, pour chacune des douze séquences de référence et pour les deux types d'écran.
Les deux séries de notes ne sont pas du tout corrélées. La différence entre les notes de qualités subjectives sur CRT et celles sur LCD est notée ΔMOS :

$$\Delta\text{MOS} = \text{MOS CRT} - \text{MOS LCD} \tag{3}$$

Cette différence est presque toujours positive (les notes de qualité subjective obtenues sur CRT sont meilleures que celles obtenues sur LCD), sauf pour deux séquences : GOLF et MOBCAL. A titre d'information, le coefficient de corrélation linéaire et l'écart quadratique moyen entre la série de notes obtenue sur CRT et celle obtenue sur LCD sont respectivement :

$$CC(\text{MOS LCD}, \text{MOS CRT}) = 0.247, \tag{4}$$

$$EQM(\text{MOS LCD}, \text{MOS CRT}) = 8.63. \tag{5}$$

Séquence	MOS CRT	MOS LCD	ΔMOS
MOBCAL	77.56	81.63	-4.07
PARKRUN	86.28	81.32	4.96
SHIELDS	84.68	77.95	6.73
STOCKHOLM	83.56	81.74	1.82
CONCERT	80.33	72.05	8.29
FOOT	83.56	73.05	10.51
MOVIE	87.28	74.91	12.37
VOILE	83.83	73.09	10.74
SHOW	81.15	69.28	11.87
STANDING	75.65	64.96	10.69
CREDITS	82.7	73.76	8.94
GOLF	72.4	77.08	-4.68

Tableau 1 – *Notes de qualité subjective moyennes par séquence et par écran.*

5 Prédiction de la différence de qualité subjective entre CRT et LCD

Afin de prédire la différence ΔMOS entre les notes de qualité subjective obtenues avec l'écran CRT et celles obtenues avec l'écran LCD, l'importance du flou de mouvement dans les séquences utilisées a été mesurée. Une estimation de la qualité subjective sur écran LCD est ensuite réalisée à partir des notes obtenues sur CRT.

5.1 Estimation de mouvement

Le mouvement est estimé en calculant des vecteurs de déplacement par une approche multirésolution. Puisque les vidéos utilisées sont entrelacées, l'estimation de mouvement porte sur chaque trame. Pour cela, les deux trames (trame des lignes paires et trame des lignes impaires) sont extraites de l'image entrelacée. Puis, chaque bloc de taille 16×8 pixels en pleine résolution (correspondant à une taille d'affichage de 16×16 pixels) d'une trame est simultanément comparé aux blocs de même taille des deux trames précédentes et des deux trames suivantes. La comparaison entre deux blocs est du type *block matching*, utilise l'erreur quadratique entre ces deux blocs et ne prend en compte que des trames de même parité. Dans les deux trames précédentes et les deux trames suivantes, la recherche est exhaustive à l'intérieur d'une fenêtre de taille suffisamment grande. Cette taille est déterminée empiriquement pour couvrir le déplacement de l'objet le plus rapide de la séquence vidéo.
Lorsque les vecteurs de mouvement ont été déterminés pour chaque bloc 16×8 pixels de chaque trame, ils sont fusionnés afin d'obtenir un vecteur de mouvement pour chaque bloc 16×16 pixels de l'image entrelacée. Cette estimation de mouvement donne des résultats robustes grâce à la comparaison simultanée d'un bloc avec ses quatre positions potentielles (dans les deux trames précédentes et dans les deux trames suivantes). L'approche multirésolution ne sert qu'à accélérer les calculs.
Enfin, chaque bloc d'un ensemble de cinq images consécutives est classé en trois catégories (contour, texture et zone uniforme) en étudiant les gradients horizontaux, verticaux et diagonaux.

5.2 Calcul du flou moyen

Nous cherchons à obtenir la quantité de flou de mouvement présent dans chacune des douze séquences à notre disposition. Les tests psychophysiques réalisés sur le flou de mouvement [7] avaient montré que la direction du mouvement n'influençait pas la quantité de flou perçu. Par conséquent, nous nous intéressons uniquement au module des vecteurs de mouvement estimés. De plus, seuls les blocs classés contour ou texture et possédant un contraste suffisant sont pris en compte.
Pour chaque séquence, nous disposons donc d'un vecteur

de mouvemement par bloc et par ensemble de cinq images consécutives. Pour un groupe de cinq images, le cumul spatial est effectué par un moyennage pondéré par le nombre de blocs, sur l'ensemble des blocs retenus. Nous obtenons alors un vecteur de mouvement moyen pour cinq images consécutives. Le module de ce vecteur permet d'obtenir la taille du flou de mouvement généré par ce déplacement à l'aide de la relation linéaire donnée précédemment (cf. equation 2), pour chaque ensemble de cinq images successives. Le flou de mouvement moyen est finalement obtenu par une moyenne temporelle sur l'ensemble de la séquence.

5.3 Prédiction du ΔMOS

Comme il a été expliqué précédemment (cf. 2.2), le flou de mouvement n'est pas le seul défaut des écrans LCD. Néanmoins, il apparaît que c'est le défaut le plus gênant lorsqu'il est présent. Nous avons donc choisi, dans un premier temps, de nous intéressez seulement aux séquences qui présentent un mouvement suffisant pour engendrer ce flou. De même, les séquences qui ont été préférées sur LCD ont été écartées car le flou de mouvement ne peut être responsable d'une amélioration de la qualité perçue.

Quatre séquences ont donc été retirées de notre base. Il s'agit des séquences MOVIE, STANDING, GOLF et MOBCAL. Notre modèle de prédiction de la différence de qualité ΔMOS en fonction du flou a donc été construit à partir des huit séquences restantes, pour lesquelles le flou de mouvement est bien présent. La relation entre le flou de mouvement et ΔMOS a été modélisée en trois parties :

– une première partie pour laquelle $\Delta\mathrm{MOS} = 0$ lorsque la taille du flou est inférieure à une certaine valeur ;
– une seconde partie linéaire croissante ensuite ;
– enfin une saturation à partir d'une certaine taille de flou.

La première partie peut s'expliquer du fait que le flou n'est visible qu'à partir d'une certaine taille critique. Empiriquement, cette valeur seuil est d'environ 5 minutes d'angle visuel. La baisse de qualité augmente ensuite proportionnellement à la taille du flou de mouvement à raison d'une pente de 1.67 points (sur l'échelle de qualité qui en contient 100) par minute d'angle visuel. Enfin, la saturation finale est certainement du à des effets contextuels (échelle de notation limitée, présence de séquences très dégradées lors du test, etc.). Elle intervient à partir d'une taille de flou de 11 minutes d'angle visuel, et correspond à un écart de qualité $\Delta\mathrm{MOS} = 10$.

Notre modèle nous permet de prédire la différence de qualité ΔMOS à partir de la valeur du flou de mouvement moyen mesuré sur la séquence. La différence de qualité prédite est notée $\Delta\mathrm{MOS}_p$. Nous pouvons alors estimer la note de qualité subjective que la séquence a obtenue sur LCD à partir de la note de qualité subjective qu'elle a obtenue sur CRT, par la relation :

$$\mathrm{MOS\ LCD}_{est} = \mathrm{MOS\ CRT} - \Delta\mathrm{MOS}_p. \qquad (6)$$

La qualité de cette estimation peut être mesurée par le coefficient de corrélation linéaire et l'écart quadratique moyen entre les notes de qualité obtenues sur LCD et les notes de qualité LCD estimées par notre modèle :

$$CC(\mathrm{MOS\ LCD}, \mathrm{MOS\ LCD}_{est}) = 0.953, \qquad (7)$$

$$EQM(\mathrm{MOS\ LCD}, \mathrm{MOS\ LCD}_{est}) = 1.30. \qquad (8)$$

En ne prenant en compte que les séquences pour lesquelles le flou de mouvement est le défaut principal, nous avons donc construit un modèle permettant de prédire, pour une séquence donnée, la différence de qualité perceptuelle entre une visualisation sur écran CRT et une visualisation sur écran LCD. Cette différence de qualité perceptuelle dépend du flou de mouvement moyen mesuré sur la séquence. La qualité subjective d'une séquence sur un écran LCD peut alors être estimée à partir de son évaluation sur écran CRT. Les notes estimées sont bien corrélées avec les notes réellement obtenues et l'écart quadratique moyen entre les deux séries est assez faible.

5.4 Autres aspects des écrans LCD

La seule prise en compte du flou de mouvement ne suffit pas pour prédire totalement la différence de qualité subjective entre CRT et LCD. En effet, si on cherche à prédire la différence de qualité des quatre séquences exclues en fonction du flou de mouvement moyen mesuré pour ces séquences, les performances de notre modèle sont diminuées. La figure 3 présente la différence de qualité ΔMOS en fonction du flou de mouvement mesuré, pour chacune des douze séquences. Le modèle de prédiction construit est également représenté. Comme prévu, le ΔMOS pour les quatres séquences incriminées ne peut être prédit par un modèle se basant uniquement sur le flou de mouvement. D'autres aspects de l'écran LCD doivent donc être pris en compte pour affiner la prédiction de cette différence de qualité.

Les séquences MOVIE et STANDING, malgré un flou de mouvement moyen très faible, ont été largement préférées sur CRT. D'autres défauts sont donc responsables de cette différence. Dans la séquence STANDING, le fond de la scène est uniforme et très foncé. La différence de gamma entre les deux écrans au niveau des faibles luminances ne joue pas en faveur du LCD, sur lequel le fond apparaît bruité et texturé. La séquence MOVIE est un plan d'ensemble d'un concert à l'intérieur d'une église, et de nombreuses zones d'ombres sont présentes sur les murs et les colonnes texturés du bâtiment. Là encore, ces zones apparaissent beaucoup plus bruitées lors de l'affichage sur LCD. La séquence GOLF, également sans flou de mouvement apparent, est quant à elle mieux notée sur LCD. Cette séquence, filmée en extérieur, est très lumineuse. Compte tenu des différences de dynamique en luminance (0.7-70 cd/m^2 pour le CRT contre 0.7-500 cd/m^2 pour le LCD), elle apparaît beaucoup plus contrastée et lumineuse sur un écran LCD. La séquence MOBCAL, malgré un flou de mouvement d'une taille supérieure à 5 minutes d'angle visuel, est elle aussi mieux notée sur écran LCD. Le contenu de cette séquence est texturé, très contrasté et très coloré.

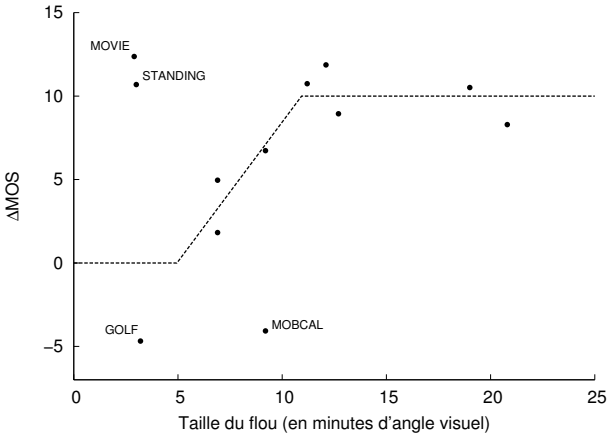


Figure 3 – *Relation entre la taille du flou de mouvement moyen et la différence de qualité pour les douze séquences utilisées.*

Ici, c'est la différence de reproduction de couleur entre les deux écrans qui semble favoriser l'écran LCD.
La prédiction objective de la différence de qualité perceptuelle entre CRT et LCD ne peut se faire en tenant uniquement compte de la quantité de mouvement moyen. Pour huit séquences dont le contenu laissait présager un fort flou de mouvement, notre modèle a permis de prédire la différence de qualité perceptuelle entre CRT et LCD. Cependant, les autres défauts ainsi que les qualité de l'écran LCD devront être mesurés et intégrés à la méthode de manière à pouvoir prédire l'écart de qualité entre les deux écrans pour n'importe quel type de contenu.

6 Conclusion

Le déploiement annoncé de la TVHD en Europe devrait provoquer la disparition des téléviseurs CRT en faveur des nouvelles technologies d'affichage telles que les écrans LCD. Néanmoins, ces derniers sont encore loin d'égaler les performances des écrans CRT, notamment concernant la reproduction des mouvements.
Dans cet article, les causes du flou de mouvement ont été étudiées. Un modèle mathématique, conforté par des tests psychophysiques, a montré qu'il dépendait linéairement de la vitesse de déplacement des objets en mouvement.
Des tests subjectifs d'évaluation de la qualité ont permis de mettre en évidence une différence importante entre la qualité perçue sur un écran CRT et celle perçue sur un écran LCD. Le flou de mouvement moyen d'une séquence donnée a été mesuré à partir des vecteurs de mouvement de cette séquence. Pour les séquences marquées principalement par le flou de mouvement, un modèle permettant de prédire la baisse de qualité sur LCD en fonction de la taille du flou a pu être construit. Cependant, les autres défauts et qualités des écrans LCD devront être pris en compte pour pouvoir mettre en œuvre une prédiction fine de la différence de qualité perceptuelle entre CRT et LCD.
L'influence des écrans LCD, ou tout au moins du flou de mouvement engendré par ces écrans, sur la qualité des images perçues par l'utilisateur final a été mise en évidence par notre méthode.

Références

[1] Takashi Fujio. Future broadcasting and high definition television. Technical report, NHK, June 1982.

[2] Ichiro Yuyama et Tetsuo Mitsuhashi. Fundamental requirements for high-definition television systems. NHK technical monograph, NHK, June 1982.

[3] ITU. Report on results of comparative subjective picture quality assessment test between CRT and LCD. Questions ITU-R 95/6, 102/6, ITU - Radiocommunication Study Groups, August 2005.

[4] Richard I. McCartney. A liquid crystal display response time compensation feature integrated into an LCD panel timing controller. *SID Symposium Digest of Technical Papers*, 34(1) :1350–1353, May 2003.

[5] Taiichiro Kurita. Moving picture quality improvement for hold-type AM-LCDs. *SID Symposium Digest of Technical Papers*, 32(1) :986–989, June 2001.

[6] Hao Pan, Xiao-Fan Feng, et Scott Daly. LCD motion blur modeling and analysis. Dans *IEEE International Conference on Image Processing, 2005. ICIP 2005.*, volume 2, pages 21–24, September 2005.

[7] Stéphane Péchard, Sylvain Tourancheau, Patrick Le Callet, Mathieu Carnec, et Dominique Barba. Towards video quality metrics for HDTV. Dans *Second International Workshop on Video Processing and Quality Metrics, VPQM'06*, Janvier 2006.

[8] ITU-R BT. 500-11. Methodology for the subjective assessment of the quality of television pictures. Rapport technique, ITU, 2004.

[9] ITU-R BT. 710-4. Subjective assessment methods for image quality in high-definition television. Rapport technique, ITU, 1998.

[10] SVT. Overall-quality assessment when targeting Wide XGA flat panel displays. Rapport technique, SVT corporate development technology, 2002. ftp ://ftp.ldv.e-technik.tu-muenchen.de/pub/test_sequences/.

[11] Joint Video Team (JVT). H.264/Advanced Video Coding reference software version 10.1, 2005. http ://iphome.hhi.de/suehring/tml/.

[12] Ferdinand Monoyer. Échelle typographique pour la détermination de l'acuité visuelle. *Académie des Sciences, Comptes rendus*, 80(113), 1875.

[13] Shinobu Ishihara. Tests for colour-blindness. Kanehara Shuppen Company, Ltd., Tokyo, Japan, 1967.

[14] EBU. SAMVIQ – Subjective assessment methodology for video quality. Rapport technique, European Broadcasting Union, 2003.

Protection en temps réel des visages dans une séquence d'images par cryptage partiel et sélectif

J.M. Rodrigues[1] W. Puech[1,2]

[1] Laboratoire LIRMM, UMR CNRS 5506,
Université Montpellier II - France

[2] Centre Universitaire de Formation et de Recherche de Nîmes - France

jose-marconi.rodrigues@lirmm.fr, william.puech@lirmm.fr

Résumé

Dans cet article nous proposons une nouvelle approche de protection de visages pour des environnements surveillés par caméra vidéo. L'objectif est de chiffrer rapidement les visages contenus dans les images d'une séquence. Nous utilisons un algorithme classique pour la détection des visages et nous proposons une nouvelle méthode pour le cryptage partiel et sélectif. Cette approche originale est basée sur un codage à longueur variable et sur le cryptage d'une partie des codes de Huffman du codeur JPEG en utilisant l'algorithme AES en mode de chiffrement par flot. La méthode proposée permet également le déchiffrage partiel d'une région d'intérêt (RI) chiffrée. Cette méthode permet d'obtenir une réduction significative du temps de codage et de décodage en comparaison avec un chiffrement complet des données après compression. Elle fournit également un débit binaire constant en restant conforme aux normes du format du codeur JPEG.

Mots clefs

Cryptage sélectif et partiel, détection de visages, compression d'images, protection.

1 Introduction

La sécurisation des transferts de contenus multimédias peut se faire soit par cryptage total soit par cryptage sélectif ou partiel. Les applications militaires et relatives à la loi exigent un cryptage total. Néanmoins, pour de nombreuses applications, un cryptage sélectif ou partiel est suffisant. Ces approches réduisent le temps de calcul ainsi que les puissances informatiques dans un réseau hétérogène avec des dispositifs de différentes capacités [1]. Le cryptage partiel (CP) d'une image a pour but de ne crypter qu'une partie de l'information spatiale de l'image en s'appliquant uniquement sur des régions d'intérêt alors que le cryptage sélectif (CS) d'une image ne crypte qu'une partie précise des informations de toute l'image (les hautes fréquences par exemple).

Un CS peut être utilisé par exemple pour des images acquises par une caméra de surveillance. Pour des visualisations en temps réel, ces images doivent être rapidement transmises et le cryptage total n'est pas nécessaire. La sécurité d'un CS ou d'un CP est toujours inférieure à celle d'un chiffrage complet, mais le CS ou le CP diminue la quantité de données à chiffrer, et par conséquent le temps de calcul. De plus, un CS ou un CP peut-être intégré à l'intérieur même d'un codeur (JPEG, JPEG2000, H264/AVC) et donc le flux en sortie reste conforme aux normes du codeur. De ce fait, un décodeur classique aura accès à l'information basse résolution. Cependant, un décodeur adapté, muni d'une clef secrète pourra décoder correctement l'information cryptée.

Dans cet article nous proposons une nouvelle approche de cryptage partiel et sélectif (CPS) du codage de Huffman pour des séquences d'images comprimées avec JPEG. Dans notre approche nous utilisons l'algorithme AES (*Advanced Encryption Standard*) [2] en mode de chiffrement par flot OFB (*Output Feedback Block*).

Dans la section 2, nous passons en revue les travaux précédents dans le domaine. Dans la section 3, nous présentons la méthode proposée, l'algorithme de cryptage sélectif, ainsi que le processus de détection des visages. Dans notre expérimentation la caméra est fixe. Du fait d'un non déplacement de la caméra, la détection des visages dans la séquence d'images est fortement facilitée. Enfin, section 4, nous montrons les résultats expérimentaux sur une séquence d'images protégeant le visage de deux personnes.

2 Travaux précédents

De nombreuses méthodes de CS et CP ont été proposées pour des images comprimées par des algorithmes basés sur la transformée en cosinus discrète (DCT). Droogenbroeck et Benedett [3] ont proposé une méthode pour chiffrer une quantité limitée de coefficients AC. Dans leur méthode les coefficients DC ne sont pas chiffrés parce qu'ils sont fortement prévisibles et qu'ils portent une information évidente. Dans cette approche les étapes de compression et de chiffrage sont faites séparément et ceci conduit à un doublement du temps de calcul par rapport à une compression

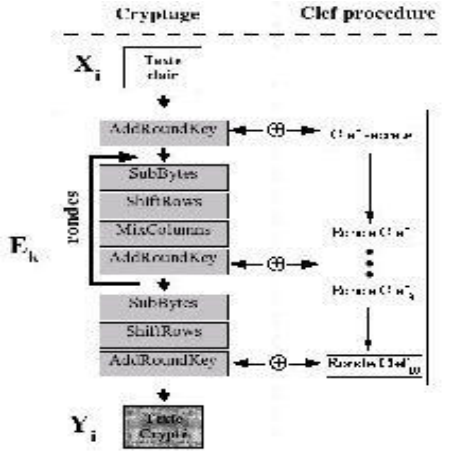
simple. Fisch et al. [4] ont proposé une méthode de cryptage sélectif d'images où les données sont organisées sous une forme de flot de bits graduable. Ces flots binaires sont construits avec des coefficients DC et quelques coefficients AC de chaque bloc de l'image et puis arrangés dans des couches selon leur importance visuelle.
Tang [5] a proposé une technique appelée *permutation zigzag* applicable aux images et aux vidéos basées DCT. Bien que cette méthode offre plus de confidentialité, elle augmente le nombre de bits total. D'autres travaux proposent également des algorithmes de CS ou CP pour des vidéos basées DCT [6, 7, 8]. Par rapport aux méthodes existantes, le fait d'utiliser l'algorithme AES en mode de chiffrement par flot (au lieu d'une approche de chiffrement par bloc) et de l'avoir d'intégré au niveau du codage de Huffman, notre méthode méthode permet de conserver le taux de compression initial du codeur. Récemment, Said [9] a mesuré la robustesse des méthodes de CS. Il a montré que des attaques qui exploitent les informations des bits en clair (non cryptés) permettent un décryptage plus rapide de l'image.

2.1 Compression d'images basée DCT

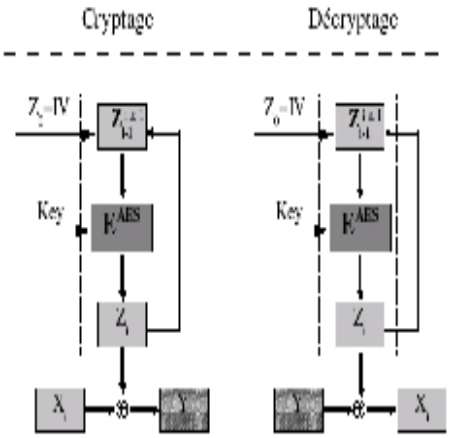
Dans les images comprimés par DCT, le codage de Huffman est fait sur les coefficients quantifiés des blocs de 8×8 pixels, et sont codés par le couple {(HEAD), (AMPLITUDE)}. L'entête HEAD contient les contrôleurs obtenus par les tables de Huffman pour la compression et la décompression. Le paramètre AMPLITUDE est un entier signé correspondant à l'amplitude d'un coefficient AC non nul, ou dans le cas du coefficient DC de la différence entre deux coefficients voisins DC. La structure HEAD varie en fonction du type de coefficient. Pour les ACs, cette structure est composée de (RUNLENGTH, SIZE), alors que pour les DCs elle est composée seulement de la taille SIZE. Pour le codage du coefficient AC, les informations conjointes du RUNLENGTH et AMPLITUDE sont utilisées et appliqués dans les tables standards. La valeur RUNLENGTH correspond au nombre de coefficients AC égaux à zéro précédant une valeur non nulle dans la séquence en zigzag. La taille SIZE est la quantité nécessaire de bits pour représenter la valeur de l'amplitude. Il y a deux codes particuliers correspond à (RUNLENGTH, SIZE) égale à (0, 0) et (15, 0). Ils sont utilisés pour symboliser la fin d'un bloc (EOB) et la longueur d'une plage de zéros. Le symbole EOB est transmis après le dernier coefficient non nul du bloc quantifié. C'est ainsi le chemin le plus efficace pour coder la fin d'une plage de zéros. Le symbole EOB est omis dans le cas où l'élément final du vecteur est non nul. Le symbole ZRL est transmis quand la valeur du RUNLENGTH est plus grande que 15 et représente une longueur de plage de 16 zéros.

2.2 Le chiffrement par AES

L'algorithme AES (*Advanced Encryption Standard*) est le standard pour le cryptage à clef secrète. L'algorithme AES est composé d'un ensemble d'étapes répétées plusieurs fois, appelé ronde. La figure 1.a représente le schéma de chiffrement d'un texte clair X_i. L'algorithme AES peut supporter les modes de chiffrement suivants : CBC *Cipher Block Chaining*, ECB *Electronic Code Book*, CFB *Cipher Feedback*, OFB *Output Feedback* et CTR *Counter Mode*.
Même si l'AES est un algorithme de chiffrement par bloc, les modes OFB, CFB et CTR opèrent comme des chiffrements par flot. Chaque mode a différents avantages et inconvénients. Dans les modes ECB, OFB et CTR par exemple, tout changement dans le bloc du texte clair X_i provoque modification dans le bloc chiffré Y_i, mais les autres blocs chiffrés ne sont pas affectés. Avec les modes CBC ou CFB, si un texte clair du bloc X_i est changé alors le bloc crypté Y_i et tous les blocs chiffrés suivants seront affectés.



(a)



(b)

Figure 1 – *a) Schéma général de l'algorithme AES, b) Cryptage et décryptage en mode OFB.*

Les propriétés des modes CBC et CFB permettent de régler des problèmes d'authentification alors que celles des modes ECB, OFB et CTR permettent de traiter séparément chaque bloc. Enfin, contrairement aux algorithmes de chiffrement par bloc, les algorithmes de chiffrement par flot permettent de faire varier de manière graduable la quantité

de bits à crypter.
Afin de pouvoir traiter séparément les régions de l'image de manière graduable, notre algorithme est donc basé sur le mode OFB, qui est un mode de chiffrement par flot synchrone de l'algorithme AES. La figure 1.b montre le chiffrement en mode OFB où le bloc en clair X_i est chiffré avec la clef secrète k afin de produire le bloc chiffré Y_i pour tout $i \geq 1$:

$$\begin{cases} Z_i &= E_k(Z_{i-1}) \\ Y_i &= X_i \oplus Z_i \end{cases}, \quad (1)$$

où $\oplus$ est le ou exclusif.
Dans la figure 1.b présentant le mode OFB, il est important de noter que la fonction de cryptage $E_k()$ est utilisé pour la phase de cryptage mais également pour la phase de décryptage.

3 Méthode proposée

Soit $E_k(X)$ le cryptage d'un bloc X de n bits en utilisant la clef secrète k avec l'algorithme AES en mode OFB. Dans la description de la méthode, nous supposerons $n = 128$. Soit $D_k(Y)$ le décryptage d'un texte chiffré Y en utilisant la clef secrète k.

3.1 Cryptage sélectif d'une séquence d'images

Le cryptage de la méthode proposée est appliqué conjointement au processus de codage entropique durant la création du vecteur de Huffman du codeur JPEG. La méthode proposée est appliquée au niveau des blocs 8×8 pixels au moment du codage de Huffman des coefficients AC. Les trois étapes sont les suivantes :
– Construction du texte clair X_i à partir des coefficients AC non nuls du flux binaire de Huffman, des plus hautes fréquences vers les basses fréquences,
– Codage de X_i avec l'algorithme AES en mode OFB pour obtenir Y_i,
– Substitution du flux binaire de Huffman par l'information cryptée qui est de même taille.
Il est important de mentionner que ces opérations sont appliquées séparément à chaque bloc quantifié.

Construction du texte clair X. Pour construire le texte clair X_i, nous prenons les coefficients AC non nuls du bloc courant i en accédant au vecteur de Huffman de la fin vers le début afin de créer des paires {HEAD, AMPLITUDE}. De chaque entête HEAD nous obtenons la longueur de l'AMPLITUDE en bit. Ces valeurs sont calculées à partir de l'équation (2). Comme montré dans la vue générale de la méthode proposée (figure 2), seules les AMPLITUDE $(A_n, A_{n-1}...A_1)$ sont prises en compte pour construire le vecteur X_i. La longueur finale du message en clair L_{X_i} dépend à la fois de l'homogénéité ρ du bloc et de la contrainte donnée C :

$$f(\rho) < L_{X_i} \leq C, \quad (2)$$

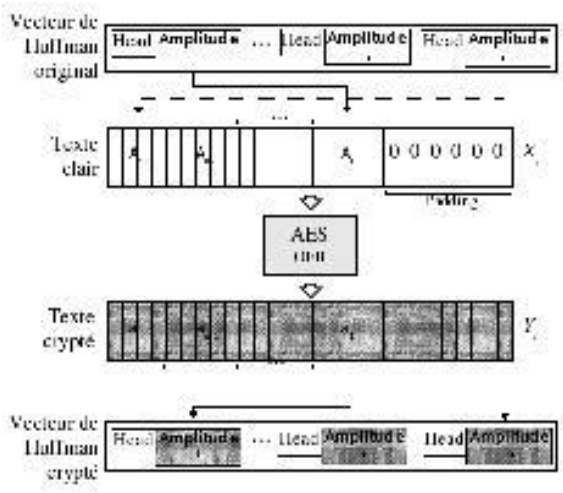


Figure 2 – *Présentation générale de la méthode proposée.*

où $f(\rho) = 0$ pour $\rho \to \infty$ et $C \in \{128, 64, 32, 16, 8\}$ bits. Cette contrainte C spécifie la quantité maximale de bits qui doit être prise en compte dans chaque bloc. La valeur de C doit donc être choisie en fonction de l'homogénéité ρ de l'image mais aussi du pourcentage de bits de l'image que l'on souhaite chiffrer. C'est donc par l'intermédiaire de C que nous graduons l'importance du cryptage. Comme l'homogénéité dépend du contenu de l'image, dans l'équation (2) celle-ci spécifie la quantité minimale de bits. Cela signifie qu'un bloc avec un grand ρ va produire un petit L_{X_i}. Le vecteur de Huffman est donc traité tant que $L_{X_i} \leq C$ et que le coefficient DC n'est pas atteint. Ensuite, nous appliquons la fonction de remplissage (padding) $p(j) = 0$, où $j \in \{L_{X_i} + 1, \ldots, 128\}$, afin de remplir si nécessaire avec des zéros le vecteur X_i.

Chiffrement de X avec AES en mode OFB. Dans l'étape de chiffrement, la clef dynamique Z_{i-1} est utilisée comme entrée pour le cryptage par AES afin d'obtenir une nouvelle clef dynamique Z_i. Pour la première itération, le vecteur d'initialisation IV ($Initialization\ Vector$) est créé à partir de la clef secrète k avec la stratégie suivante : la clef secrète k est utilisée comme une semence pour un générateur de nombres pseudo-aléatoire (GNPA). Ce k est divisé en 16 portions de 8 bits chacun. Le GNPA produit 16 nombres aléatoires qui définissent l'ordre de formation du IV. Ensuite chaque Z_i est additionnée par un ou exclusif avec le texte en clair X_i pour générer le bloc chiffré Y_i.

Substitution du flux binaire de Huffman. L'étape finale est la substitution de l'information initiale par l'information chiffrée dans le vecteur de Huffman. Comme dans la première étape (construction du texte clair X_i), le vecteur de Huffman est lu depuis la fin vers le début mais le vecteur chiffré Y_i est lu du début vers la fin. Connaissant la longueur en bits de chaque AMPLITUDE $(A_n, A_{n-1}...A_1)$, nous commençons par couper ces portions dans Y_i pour remplacer l'AMPLITUDE dans le vec-

teur de Huffman. La quantité totale de bits doit être L_{X_i}. Cette procédure est faite pour chaque bloc. Les blocs homogènes ne sont pas ou peu chiffrés. L'utilisation du mode OFB pour le chiffrement permet une génération de clef dynamique Z_i indépendante. Il est important de noter que l'algorithme de CS utilisé au niveau le vecteur de Huffman n'augmente pas la taille finale du vecteur binaire comprimé. Ceci vient du fait de l'utilisation d'un mode de chiffrement par flot.

3.2 Procédure de décryptage

La procédure de décryptage en mode OFB fonctionne de la manière suivante. Comme pour la phase de cryptage, la clef dynamique Z_{i-1} est utilisée en entrée du cryptage par AES afin d'obtenir une nouvelle clef dynamique Z_i. Dans la phase de décryptage, la différence est que la clef dynamique Z_i est additionnée par un ou exclusif avec le bloc chiffré Y_i afin de régénérer le texte en clair X_i comme illustré figure 1.b. Le vecteur résultat du texte en clair X_i est coupé en parties de la fin vers le début afin de remplacer les AMPLITUDE dans le chiffré de Huffman pour générer le vecteur de Huffman.

3.3 Détection des visages

La méthode de CS présentée section 3.1 est appliquée sur une séquence d'images afin de masquer les visage des personnes. Dans cette section, nous présentons l'étape de détection des visages. Les visages détectés constitueront des régions d'intérêts (RIs) dans lesquelles le CS sera appliqué. Nous obtenons alors un CP (uniquement les RIs) combiné avec un CS (les plus hautes fréquences).
La première étape de l'algorithme consiste en une transformation couleur de l'espace RGB à l'espace couleur YUV. Comme dans le codeur JPEG, nous calculons ensuite la transformée en cosinus discrète pour chaque composante (Y, U et V) par bloc afin de générer les coefficients DC et AC. Nous employons les coefficients DC des composantes U et V pour produire deux imagettes qui sont utilisées pour détecter la couleur peau à partir de l'équation (3). A partir d' une image de $M \times N$ pixels, nous obtenons alors deux imagettes DC_U et DC_V. Un bloc est considéré comme un bloc contenant une partie de visage si la couleur peau est détectée :

$$\sqrt{(DC_U/8 - U_p)^2 + (DC_V/8 - V_p)^2} < S, \quad (3)$$

où U_p et V_p sont les couleurs de peau de référence fournis dans l'espace YUV, S le seuil de détection. Pour un facteur de qualité $FQ = 100\%$ nous avons $DC_U/8$ et $DC_V/8$ qui correspondent aux valeurs moyennes des blocs correspondants.
Le résultat de l'étape de détection est généralement une image binaire bruitée. Afin de filtrer cette image binaire, nous appliquons une fermeture suivie d'une ouverture morphologique [10]. Dans notre approche la caméra est fixe, ceci facilite la détection des RIs dans la séquence d'images. L'image binaire filtrée indique les blocs constituant les RIs qui doivent être chiffrées dans la séquence. Chaque pixel blanc dans l'image binaire correspond à un bloc dans l'image originale. La méthode de CS décrite dans la section 3.1 est alors appliquée sur le vecteur de Huffman de la composante Y.

4 Résultats expérimentaux

Pour nos expériences, nous avons sélectionné quatre images ($\#083, \#123, \#135$ et $\#147$) (figures 3.a) d'une séquence de 156 images de taille 640×480 pixels. Pour la détection des visages nous avons utilisé comme couleurs de référence de la peau $U_p = 120$ et $V_p = 140$ et le seuil $S = 5$. Pour le chiffrement, nous avons employé l'algorithme AES en mode de chiffrement par flot OFB avec une clef de longueur 128 bits. Cependant, notre méthode peut être employée avec d'autres valeurs de longueur pour la clef et pour les blocs. Pour l'utilisation de l'équation (2), nous avons fixé C à 128 bits.
Pour chacune des quatre images sélectionnées présentées figures 3.a, nous sommes passés de l'espace couleur RGB à l'espace YUV. Dans l'espace YUV nous obtenons les imagettes constituées des composantes DC, illustrées figures 4 pour l'image #083. À partir des coefficients DC des composantes U et V nous avons appliqué l'équation (3) pour générer les masques binaires. Nous avons ensuite utilisé l'algorithme d'érosion et dilatation sur les imagettes binaires bruitées afin d'obtenir les figures 3.b. Les pixels blancs représentent des blocs considérés comme représentant des visages. En effet chaque pixel blanc de l'image binaire est un bloc de peaux dans l'image originale. Sur ces blocs nous avons alors appliqué l'algorithme de CS décrit précédemment pour produire des images partiellement et sélectivement chiffrées présentées figures 5.

	Total chiffré			
Image	#. Blocs	Coeffi cients	Bits	% Bloc
#083	79	2547	10112	1.65
#123	113	3042	14464	2.35
#135	159	4478	20352	3.31
#147	196	5396	25088	4.08

Tableau 1 – *Résultats du CS employé dans la séquence d'images.*

Le tableau 1 indiquent les résultats obtenus pour chaque image. Pour l'image #083 nous avons détecté 79 blocs de peaux. Dans cette image 2547 coefficients AC (10112 bits) ont été chiffrés. Le nombre de blocs chiffrés correspond à 1.6 % du nombre total de blocs de l'image originale. Pour l'image #123 nous avons crypté 113 blocs, du fait que les visages sont plus grands que dans l'image #083. Nous avons alors chiffré 3042 coefficients AC, ce qui représente 14464 bits et 2.35 % des blocs chiffrés. Dans la séquence d'images utilisée, la quantité de blocs à chif-

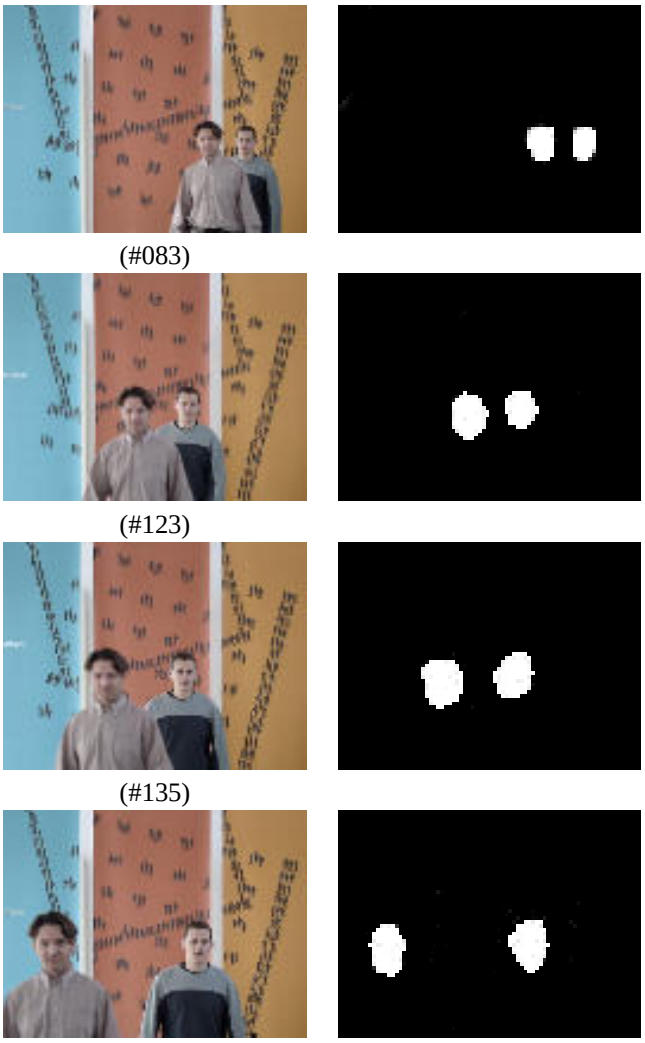


Figure 3 – *a) Séquence des images originales, b) Séquence des images binaires.*

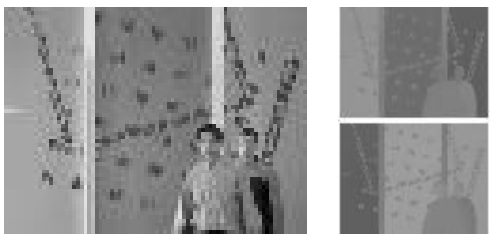


Figure 4 – *Imagettes des coefficients DC de l'image #083 pour les plans YUV.*

Figure 5 – *Séquence d'images cryptée partiellement et sélectivement.*

frer augmente du fait que les deux personnes s'approchent de la caméra. Cependant, comme le montre le tableau 1, la quantité de bits chiffrés est très faible par rapport la taille de l'image. Ceci fait que notre méthode est applicable dans des environnements de faible puissance de calcul comme, par exemple, des vidéos issues de caméras portables. La figure 5 montre les résultats de détection des visages et du cryptage partiel et sélectif dans cette séquence d'images. Afin de montrer plus clairement nos résultats, nous avons agrandit, figures 6, une zone de 216×152 pixels de l'image #123.

Il convient de noter que la sécurité d'un cryptosystème est liée à la capacité de deviner les valeurs des données chiffrées. Par exemple, il est préférable de chiffrer les bits qui sont les plus aléatoires possible. Cependant, la sécurité d'un CS ou d'un CP est toujours plus faible que celle d'un cryptage complet. La raison la plus importante d'accepter ce schéma est la réduction importante du temps de calcul par rapport à un cryptage total. Cependant, en pratique, une attaque est plus difficile sur les coefficients ACs non nuls d'une séquence d'images que sur ses coefficients DC qui sont fortement prévisibles [3, 11].

Figure 6 – *Région de* 216×152 *pixels de l'image #123.*

5 Conclusion

Dans cet article, nous avons proposé un nouveau schéma de cryptage partiel et sélectif pour des séquences d'images codées avec JPEG en utilisant le cryptage AES en mode par flot OFB. Les avantages de notre méthode sont la portabilité, un taux de compression conservé par rapport à une compression standard, une compatibilité avec le codeur JPEG, un cryptage sélectif réglable en quantité et un décryptage partiel par région d'intérêt. En perspectives nous envisageons d'intégrer notre approche dans des séquences vidéos H264/AVC.

Références

[1] X. Liu et A. Eskicioglu. Selective Encryption of Multimedia Content in Distribution Networks :Challenges and New Directions. Dans *IASTED Communications, Internet & Information Technology (CIIT), USA*, November, 2003.

[2] J. Daemen et V. Rijmen. AES Proposal : The Rijndael Block Cipher. Rapport technique, Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, 2002.

[3] M. Van Droogenbroeck et R. Benedett. Techniques for a Selective Encryption of Uncompressed and Compressed Images. Dans *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium*, Sept. 2002.

[4] M. M. Fisch, H. Stgner, et A. Uhl. Layered Encryption Techniques for DCT-Coded Visual Data. Dans *European Signal Processing Conference (EUSIPCO) 2004, Vienna, Austria*, Sep., 2004.

[5] L. Tang. Methods for Encrypting and Decrypting MPEG Video Data Efficiently. Dans *ACM Multimedia*, pages 219–229, 1996.

[6] W. Zeng et S. Lei. Efficient Frequency Domain Video Scrambling for Content Access Control. Dans *ACM Multimedia, Orlando, FL, USA*, pages 285–293, Nov. 1999.

[7] H. Cheng et X. Li. Partial Encryption of Compressed Images and Videos. *IEEE Transactions on Signal Processing*, 48(8) :2439–2451, 2000.

[8] J. Wen, M. Severa, W. Zeng, M. Luttrell, et W. Jin. A Format-Compliant Configurable Encryption Framework for Access Control of Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6) :545–557, 2002.

[9] A. Said. Measuring the Strength of Partial Encryption Scheme. Dans *ICIP 2005, IEEE International Conference in Image Processing, Genova, Italy*, volume 2, pages 1126–1129, 2005.

[10] J. Serra. *Image Analysis and Mathematical Morphology, vol. 2*, volume 2. London : Academic Press, 1988.

[11] T. Lookabaugh. Selective encryption, information theory, and compression. Dans *38th ASILOMAR Conference on Signals, Systems and Computers*, volume 1, pages 373–376, 2004.

Détection entièrement automatique de points de fuite dans des scènes architecturales urbaines

M. Kalantari[a,b,d] F.Jung[a,d] G.Moreau[a,c] JP.Guedon[a,b]

[a] Institut de Recherche sur les Sciences et Techniques de la Ville CNRS FR 2488

[b] équipe Image Vidéo Communications
Institut Recherche Communications Cybernétique de Nantes (IRCCyN) UMR CNRS 6597,

Fédération AtlanSTIC CNRS - FR2819

[c] CNRS CERMA UMR 1563
ENSA Nantes - rue Massenet 44330 NANTES

[d] Ecole Supérieure des Géomètres Topographes
1, boulevard Pythagore - Campus Universitaire - 72000 Le Mans

{mahzad.kalantari,franck.jung}@esgt.cnam.fr ,

guillaume.moreau@ec-nantes.fr,jean-pierre.guedon@polytech.univ-nantes.fr

Résumé

Dans les applications de type 3D urbain, il est apparu comme capital de savoir extraire de façon automatique les points de fuite, par exemple pour l'orientation des images ou pour renforcer les calculs permettant leur assemblage dans une géométrie rigoureuse. Pour cela, il est présenté une nouvelle méthode de détection complètement automatique, robuste et de complexité réduite vis-à-vis des algorithmes classiques, qui tient en deux étapes : une détection de segments par un algorithme de Canny-Deriche modifié, suivie d'une caractérisation de chacun des segments par l'appartenance de leur projection à un cercle spécifique de l'espace original. La bonne performance en termes de robustesse se double d'une réduction de complexité puisque l'ensemble du processus est réalisé en 2D sans aucune aide externe.

Mots clefs

Patrimoine architectural, détection automatique de points de fuite, cercles K.

1 Introduction

De multiples applications techniques au sein de la Ville sont actuellement basées sur de l'imagerie terrestre, comme la conservation du patrimoine historique architectural au cours du temps, et surtout sa gestion au quotidien par les services techniques. En outre, un accès économiquement raisonnable à des techniques de réalité virtuelle, techniques désormais plébiscitées et qui se généralisent très rapidement, requiert un traitement complètement automatisé et si possible temps réel des images acquises. Pour cela, nous nous intéressons à la détection des points de fuite dans des images urbaines [11].

Les points de fuite sont en effet caractéristiques des images d'objets fabriqués par l'homme [9], tout particulièrement les bâtiments, dans lesquels la quasi totalité des lignes visibles sur des images correspondent à des éléments strictement horizontaux ou verticaux. Ces directions spécifiques peuvent donc être employées pour faciliter l'orientation des images, pour autant qu'on sache les retrouver de façon simple dans ces images. Dans la géométrie conique caractéristique de la vision humaine ou de la photographie, ces lignes parallèles se traduisent par des faisceaux de droites qui concourent sur des points de fuite. Classiquement, un bâtiment isolé simple donne lieu à 3 points de fuite, celui lié aux lignes verticales, et un pour chaque groupe de lignes horizontales dans chaque façade visible. Le but de ce papier est la production d'un nouvel algorithme permettant la détection automatique de

tous les points de fuite engendrés par les lignes visibles dans une image 2D.
Nous débutons dans la partie 2 par un état de l'art de la détection des points de fuite qui montre un ordre de complexité élevé des algorithmes mis en jeu et un manque de robustesse à toutes les situations de l'architecture urbaine.
La méthode proposée repose sur deux parties. La partie 3 présente une première détection des contours par une adaptation du filtrage de Canny-Deriche qui associe à chaque segment trouvé un ensemble de paramètres qualifiant son incertitude. La partie 4 utilise cette information en entrée pour établir le classement des segments selon le point de fuite afin de localiser ensuite facilement celui ci. Une validation visuelle des résultats obtenus est finalement présentée et discutée.

2 Etat de l'art

Il existe à ce jour différentes méthodes pour la détection des points de fuite. Une des plus importantes est basée sur l'emploi de la sphère de Gauss, avec une grande diversité de variantes depuis son introduction par Barnard [2] en 1983. L'avantage de cette méthode est de ramener dans un espace fini tous les points de fuite. Cette approche a été complétée en 1984 par Magee et Aggarwal [13] qui accumulent la projection de l'intersection des segments de l'image sur la sphère de Gauss. Cette méthode est très lourde au niveau des calculs mais considérée comme très précise. Elle procède d'un calcul qui intervient dans l'espace 3D de la sphère de Gauss.
En 1998, Lutton et al. [12], proposent une nouvelle approche qui est une adaptation de la transformée de Hough pour la détection des points de fuite. Tuytelaars [16] introduit peu après une méthode interactive basée également sur la transformé de Hough, sous le nom de « Cascade Hough Transform ».
D'autres méthodes n'utilisent pas d'espace fini d'accumulation, et travaillent directement sur l'image. C'est le cas de Quan et Mohr [15] ou Den Heuvel [10] qui ont introduit une méthode de détection basée sur des contraintes géométriques. Brauer et Voss [4] s'intéressent à la détection des points de fuites dans des images où le niveau de bruit est élevé. Récemment, Almansa [1] a développé une nouvelle méthode de détection de points de fuite qui ne nécessite aucune information à priori, mais utilise des modèles probabilistes complexes.
Notre démarche [11] consiste à détecter de façon automatique les points de fuite dans l'espace image sans avoir recours à des espaces finis 3D, en se basant sur une géométrie simple 2D.

3 Détection des contours

Les segments sont extraits en utilisant une approche classique. Le travail est réalisé sur une image de luminance.
Pour commencer, une détection de contours est réalisée par un filtre de Canny-Deriche [6]. Ensuite, une détection des maxima locaux dans la direction du gradient est effectuée. Une localisation subpixellaire de ces maximas est réalisée en utilisant une technique d'interpolation décrite dans [14]. Cette étape est suivie d'un seuillage par hystérésis à deux paramètres ($\mathbf{S_b}$ =5, $\mathbf{S_h}$ =10). Ces seuils sont pris suffisamment bas (à la limite de la distinction visuelle) afin de ne pas constituer des seuils critiques dans cet algorithme. Expérimentalement, aucun problème de détection n'a pu être observé en utilisant ces seuils.
Ces contours sont chaînés. Une approximation polygonale des contours est réalisée. La polygonisation est réalisée en utilisant une fusion itérative basée sur le résidu maximum de la régression orthogonale. On fusionne d'abord les polylignes dont la fusion fournit un résidu maximum minimal. L'algorithme de fusion est stoppé dès que le résidu maximum dépasse un seuil fixé. Dans nos applications et compte-tenu de notre souhait d'utiliser les segments de droites dans des processus d'estimation de points de fuites ainsi que dans un processus d'orientation des images, nous avons choisi un seuil très bas : 0.5 pixels.
Cette étape de polygonation est suivie par une estimation des paramètres de la droite ainsi que de la matrice de variance covariance de ces paramètres. L'algorithme utilisé est décrit dans [5]. Notons qu'aucun de ces paramètres ne semble critique pour la démarche proposée. Il est même possible d'en estimer certains (seuil de polygonisation) en fonction du rapport signal sur bruit de l'image ainsi que du paramètre α du filtre de Canny-Deriche.

4 Classification des segments par groupes de gerbes perspectives.

L'apport de notre contribution à partir de cette image de segments est maintenant présenté. Après avoir rappelé les points particuliers et la géométrie, un résultat de géométrie projective sert à caractériser l'appartenance de chaque point caractéristique d'un segment d'un point de fuite donné à un cercle K dont on calcule ensuite les caractéristiques. Cette démarche se révèle être d'une grande robustesse.

4.1 Caractérisation des segments de l'image

Chaque segment peut être caractérisé de nombreuses façons différentes, par exemple par les coordonnées de ses points d'extrémité. Ici, nous avons choisi de les caractériser d'une part par leur distance à l'origine du système de coordonnées, et d'autre part par l'angle formé par la direction de la droite orthogonale au segment et l'un des axes. Pour des raisons pratiques, l'origine O a été choisie dans le coin supérieur gauche de l'image, l'axe des

x étant horizontal et l'axe des y sur la verticale gauche, le repère ainsi formé n'étant donc pas de sens direct, mais chaque pixel recevant ainsi des coordonnées positives. L'angle θ est compté par rapport à l'axe x (voir figure 1). Nous désignons par H l'intersection entre la droite orthogonale issue de O au segment considéré et ledit segment.

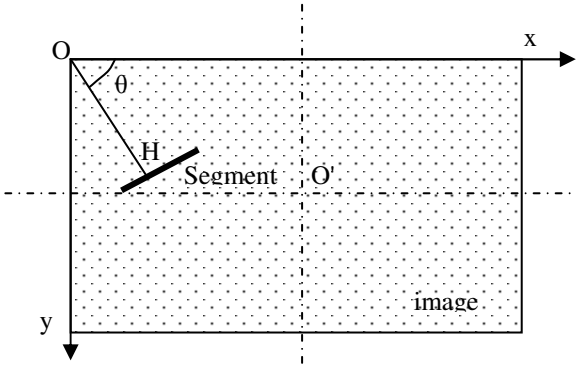


Figure 1 : Référence adoptée pour l'étude

Dans un premier temps, nous avons analysé visuellement l'histogramme formé par l'ensemble des **θ** de tous les segments. Un tel histogramme (cf. Figure 2) montre bien les différents groupes issus des gerbes perspectives correspondant aux différents points de fuite de l'image.

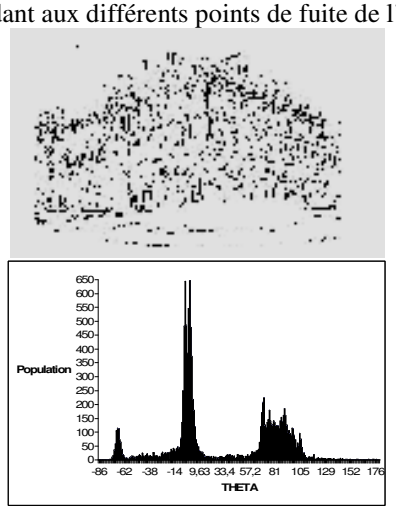


Figure 2 : Une image de segments, et son histogramme des orientations θ.

Néanmoins, si cet histogramme permet d'isoler sans aucune ambiguïté les lignes correspondant aux verticales (cf. Figure 2) pour autant que la photographie ait été acquise avec un axe plus ou moins horizontal, par contre les gerbes perspectives correspondant aux lignes horizontales des bâtiments se mélangent au moins en partie, sans qu'il soit possible de les séparer sur ce seul critère de **θ**.

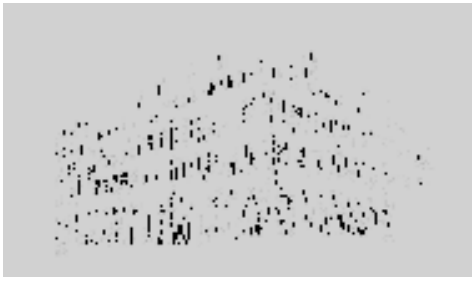


Figure 3 : Extraction des segments verticaux par leur angle θ.

Il suffit pour s'en persuader de prendre l'exemple de lignes horizontales du bâtiment localisées près de l'horizontale issue du centre optique : dans cette zone de l'image, deux gerbes perspectives correspondant à deux façades contiguës d'un même bâtiment sont nécessairement mélangées. Au-delà de cet exemple qui est assez ponctuel (zone très limitée de l'image), on trouve sans difficultés de nombreux groupes de segments parallèles appartenant à deux gerbes perspectives différentes, et qui rendent une discrimination sur le seul **θ** assez largement inopérante.

D'autres critères de sélection ont donc été recherchés. Nous nous sommes intéressés ici à une discrimination basée sur les points **H**. En effet, pour deux segments parallèles appartenant à deux gerbes différentes, les points **H** sont presque toujours très différents. Nous avons donc étudié la géométrie de la figure formée par l'origine **O**, les points **H** de tous les segments, et les différents points de fuite $\mathbf{P_i}$. Il est intéressant de noter que pour une gerbe perspective donnée correspondant à un point $\mathbf{P_i}$, les points **H** de tous les segments de cette gerbe sont théoriquement disposés sur un cercle **K** de diamètre **OP**. En effet, les droites **OH** et **HP** sont orthogonales par construction, donc tous les triangles **OHP** sont rectangles de même hypoténuse **OP**. L'exploitation de cette propriété géométrique avait déjà été suggérée en 1999 par Bräuer & Voss [4], mais sans mise en œuvre.

Nous avons donc étudié la possibilité de regrouper l'ensemble des points **H** de l'image (autant que de segments), directement sous forme de cercles. En effet, il est possible d'envisager ensuite une extraction automatique de tous les cercles formés par ces points **H**. Une caractéristique des cercles **K** est d'être définis par seulement deux paramètres, puisqu'ils passent par l'origine des coordonnées. (cf. Figure 4)

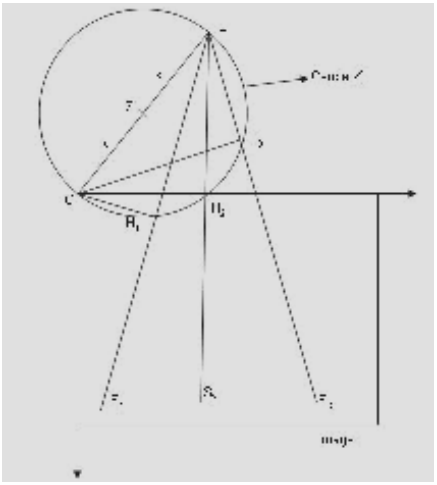


Figure 4 : Configuration des points H, du point de fuite et du cercle K sur l'image.

4.2 Recherche automatique des cercles K.

Le problème est maintenant de définir tous les cercles qui passent par des ensembles significatifs de points **H**. Un cercle est défini par 3 points, et il existe évidemment différentes façons de faire passer un cercle par un ensemble de points, soit en minimisant la distance algébrique entre le cercle et les différents points, soit en minimisant la distance géométrique [8]. Dans notre application et pour la simplicité du modèle nous avons choisi de minimiser la distance algébrique :

$$F(\mathbf{x}) = a\,\mathbf{x}^t\mathbf{x} + \mathbf{b}^t\mathbf{x} + c = 0\,, \qquad (1)$$

*o*ù $a \neq 0$ et $\mathbf{x}$ *et* $\mathbf{b} \in \mathbb{R}^2$.

Pour ajuster un cercle à un nuage de points, il faut calculer a, **b** et c. En insérant les coordonnées des points dans l'équation (1) nous obtenons un système d'équations tel que :

$$B\,\mathbf{u} = 0\,, \qquad (2)$$

où $\mathbf{u} = (a\,;\, b1\,;\, b2\,;\, c)$ et

$$B = \begin{vmatrix} X^2_{11} + X^2_{12}, X^2_{11}, X^2_{12}, 1 \\ \ldots\ldots \\ \ldots\ldots \\ X^2_{n1} + X^2_{n2}, X^2_{n1}, X^2_{n2}, 1 \end{vmatrix}\,. \qquad (3)$$

Pour trouver la solution de l'équation homogène (2) nous imposons la contrainte suivante :

$$\|u\| = 1, \qquad (4)$$

et nous cherchons donc à minimiser le système suivant :

$$\min_{\|u\|=1} \|B\,u\| \quad . \qquad (5)$$

Ce système sera résolu par une décomposition SVD et la solution finale sera celle de la plus petite valeur propre du vecteur propre correspondant.

$$K = (k_1, k_2) = \left(-\frac{b_1}{2a}, -\frac{b_2}{2a} \right) \qquad (6)$$

$$r = \sqrt{\frac{\|b\|^2}{4a^2} - \frac{c}{a}} \qquad (7)$$

où $\mathbf{k_1}$ et $\mathbf{k_2}$ sont les coordonnées du centre du cercle **K**, et **r** son rayon. En outre on est dans un cas de figure où le cercle passe par l'origine O, donc c = 0.

La méthode des moindres carrés est connue pour avoir une robustesse exécrable dès qu'il existe des mauvais points. Nous avons donc adopté la méthode RanSac [7], qui est connue en photogrammétrie et en vision par ordinateur pour sa grande robustesse et sa rapidité de calcul. Il s'agit d'une méthode d'estimation par consensus. On commence par un échantillonnage aléatoire de 3 points qui permet de déterminer un cercle, puis avec les paramètres ainsi calculés, on sélectionne les points dont la distance au cercle est inférieure au seuil prédéfini (t). On cherche à ce stade le meilleur cercle passant par les points ainsi choisis.

En analysant visuellement la représentation graphique des points H extraits, nous retrouvons bien les arcs de cercle évoqués précédemment (cf Figure 5).

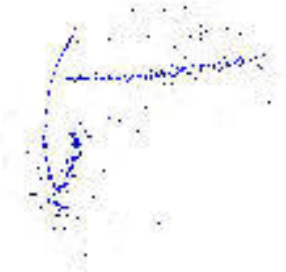


Figure 5 – *Exemple des points H (issus de l'image en Fig.2).*

Pour ce faire, chaque point qui est retenu pour participer à la détermination d'un cercle est retiré de l'ensemble de segments de départ. Le nombre de points minimal participant au consensus est déterminé comme un pourcentage du nombres total de segments détectés, et ce pourcentage est l'un des paramètres importants de réglage de l'algorithme. Par itérations successives, la totalité des cercles est extraite, et si le réglage des seuils est bien adapté, on obtient uniquement les cercles correspondant aux points de fuite (cf. Figure 6).

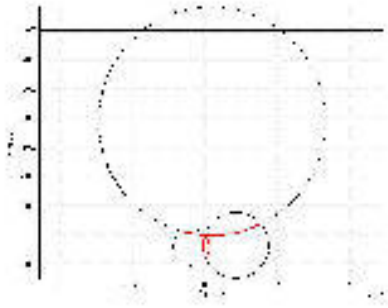


Figure 6 : Cercles extraits à partir des nuages de points H. On note que l'intersection des trois cercles est bien l'origine O, et la zone surlignée en rouge correspond aux points visible dans la figure 5.

4.3 Validation visuelle de l'algorithme.

Une fois que l'on a calculé les paramètres de tous les cercles, pour pouvoir valider l'algorithme, nous traitons le problème en sens inverse, c'est-à-dire que partant de l'ensemble de tous les segments nous sélectionnons et représentons sur l'image d'origine les segments correspondant aux cercles calculés. On peut ainsi optimiser le réglage des paramètres de RanSac, et identifier où prennent naissance des artefacts résiduels. Nous avons ainsi identifié sans surprise que dans les paysages urbains courants, avec des images acquises à hauteur humaine, la partie basse de l'image (en dessous de 1,50 m par exemple) était très riche en segments sans aucun rapport avec les bâtiments (peintures au sol, mobiliers urbains, véhicules en stationnement, …).

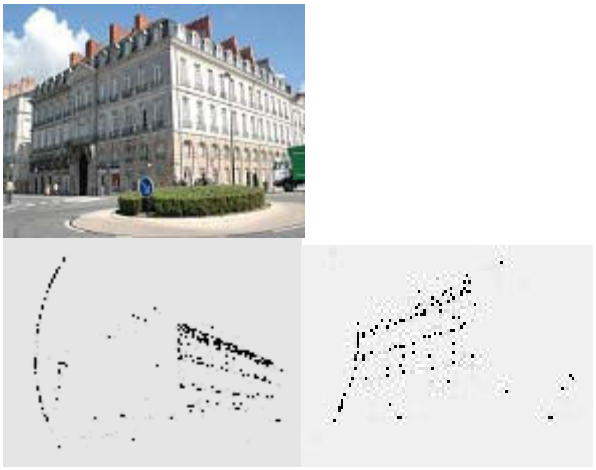


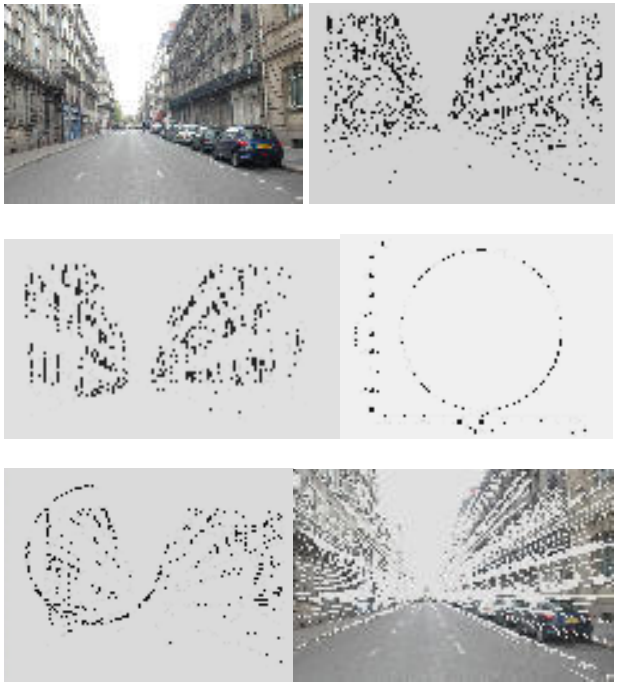
Figure 7- Les cercles K correspondant aux gerbes perspectives de deux façades d'un bâtiment : on notera les nombreux artefacts correspondant aux parties de l'image situées en dessous de l'horizontale.

5 Conclusion

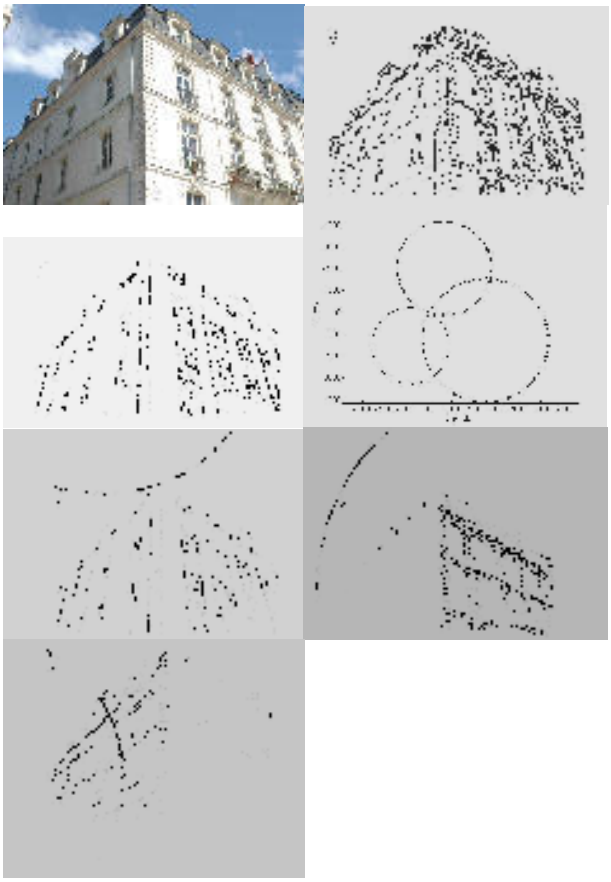
Dans ce papier, nous avons présenté une nouvelle méthode de détection automatique de points de fuite en se basant sur une géométrie simple dans l'espace 2D de l'image. Les résultats montrent l'efficacité de cette méthode de calcul, qui en outre est rapide, ce qui permet d'envisager son utilisation en temps réel sur le terrain.

Annexe

Résultats



Exemple de détection de points de fuite dans une image de rue. De haut en bas : Image brute, segments extraits, segments verticaux seuls, cercles K extraits (le grand correspond aux verticales, avec un point de fuite très loin de l'image), validation visuelle montrant les segments liés à des horizontales et le cercle K correspondant, image d'origine avec superposition les lignes de fuite extraites.



Même série, pour une image différente présentant trois points de fuite. De haut en bas : Image brute, segments extraits, segments verticaux seuls, cercles K extraits (un pour les verticales, deux pour les horizontales), validation visuelle montrant les segments liés à des horizontales et le cercle K correspondant pour les deux façades.

Références

[1] A. Almansa, A. Desolneux, S. Vamech, Vanishing points detection without any a priori information. *IEEE Trans. on PAMI*, 25(4):502–507, 2003.

[2] Barnard S. Interpreting perspective images. *Artificial Intelligence*, vol. 21. 1983.

[3] B. Brillault, B. O'Mahoney,. New method for vanishing point detection. *CVG-IP, Image Understanding*, 54(2):289-300, 1991.

[4] C. Bräuer, Burchardt, Klauss Voss, Robust Vanishing Point Determination in Noisy Images. *Internal report,* Digital Image Processing Group. Institute for Compute Science. University of Jena. Germany. 1999.

[5] R. Deriche, R. Vaillant, O. Faugeras. From Noisy Edges Points to 3D Reconstruction of a Scene : A Robust Approach and Its Uncertainty Analysis . *World Scientific Series in Machine Perception and Artificial Intelligence* , Vol. 2, p. 71-79. 1992.

[6] R. Deriche, Using Canny's criteria to derive an optimal edge detector recursively implemented*, Int. J. Computer Vision*, Vol. 2, p. 15-20, Avril 1987.

[7] M. A Fischler,, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM,* Vol. 24(6):, p.381-395. 1981.

[8] W. Gander, G. H. Golub, and R. Strebel, Fitting of circles and ellipses least squares solution, *Technical Report 217, Institut fur Wissenschaftliches Rechnen,* ETH Zurich, June 1994.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] F.A Heuvel., Vanishing point detection for architectural photogrammetry. *International Archives of Photogrammetry and Remote Sensing* Vol. XXXII part 5, p. 652-659. 1998.

[11] M. Kalantari, F. Jung, Détection entièrement automatique de points de fuite dans des scènes architecturales urbaines, xyz N° 107, juin 2006, pp. 41-46.

[12] E. Lutton, H. Maitre., J. Lopez-Krahe, Contribution to the determination of vanishing points using Hough transform*. IEEE Trans. PAMI*. Vol. 16, N°4, pp. 430-438, Avril 1994.

[13] M. J. Magee, J. K. Aggarwal, Determining vanishing points from perspective images. *CVGIP*, 26(2): pp. 256-267. 1984.

[14] N. Paparoditis Thèse Reconstruction 3D de paysages urbains en imagerie stéréoscopique spatiale haute résolution, Thèse, Université de Nice-Sophia Antipolis, 1998.

[15] L. Quan, R. Mohr, Determining perspective structures using hierarchical Hough transform. *Pattern Recognition Letters* Vol. 9, pp. 279-286. 1989.

[16] T. Tuytelaars, L.Van Gool, M. Proesmans, T.Moons, The cascaded Hough transform as an aid in aerial image interpretation. *Proceedings of ICCV*, p.67-72, 1998.

Analyse des propriétés stochastiques des échos de précipitations par les données bidimentionnelles.

M. Lazri[1], S. Ameur[1] et M. Sehad[1]

[1]Laboratoire d'analyse et de modélisation des phénomènes aléatoires (LAMPA), Département d'électronique, Faculté Génie Electrique et Informatique, Université Mouloud Mammeri, BP no 17 RP 15000 Tizi-Ouzou, Algérie.

m_lazri@yahoo.fr

Résumé :

Ce travail porte sur la modélisation de comportement des précipitations utilisant les données radar météorologique dans les régions de Bordeaux, Sétif et Dakar par l'approche markovienne. Les données que nous avons utilisées pour mettre au point ce modèle, sont des images radar météorologique. L'information radar présentée sur les images, donnée en intensité de précipitation, a été exploitée pour construire des séries d'observations de précipitations permettant d'analyser le comportement stochastique de précipitations. En premier lieu, nous avons élaboré la cartographie des intensités de précipitations dans les régions étudiées. Ensuite, nous avons montré que les échos de précipitations sont bien décrits par les chaînes de Markov de premier ordre à deux états, aussi bien sur mer que sur terre, et quel que soit le climat prévalant dans la zone d'étude. Nous avons montré que la probabilité de succession d'un même état est élevée aussi bien sur terre que sur mer et quelle que soit la région. En deuxième lieu, afin de mieux apprécier l'influence du passé sur le comportement des précipitations, nous avons augmenté la profondeur. A cet effet, nous avons utilisé l'hypothèse des chaînes de Markov de deuxième ordre. Les résultats obtenus par cette deuxième hypothèse donne une meilleure représentation des précipitations. En effet, les probabilités à long terme et les probabilités à priori sont quasiment identiques.

Mots-clés : modèles statistiques et probabilistes ; chaîne de Markov ; modélisation des processus aléatoires ; images radar météorologique ; phénomènes météorologiques ; échos de précipitations.

1. Introduction

L'existence, la connaissance et l'étude des données météorologiques et plus particulièrement celles concernant les précipitations, sont largement utilisées en hydrologie aussi bien pour l'étude et la compréhension des régimes des cours d'eau que pour la prévision des crues.

De nos jours, le radar météorologique est l'instrument le plus utilisé de par le monde pour l'observation météorologique. En effet, il permet une couverture continue des champs de précipitations dans le temps et dans l'espace. Particulièrement, le radar côtier présente l'avantage d'être le seul instrument qui permet d'étudier le comportement des précipitations sur la mer, la côte et le continent et de déceler une quelconque différence si elle existe pour ces trois régions d'étude. Actuellement, les données radars météorologiques, qui permettent de localiser et de suivre des champs de précipitations, sont largement utilisées pour les besoins hydrologiques et la prévention contre les catastrophes naturelles. Il s'agit de détecter et d'évaluer des situations à risques et de fournir des éléments d'aide à la décision. Une bonne description stochastique des précipitations aidera à apporter des informations capitales en hydrologie et améliorera la prévision des crues et des catastrophes naturelles. De ce fait, le modèle markovien est choisi pour analyser et modéliser les précipitations [1].

2. Données :

Les données que nous avons utilisées pour modéliser les précipitations sont des données radar météorologique. Ces données sont des images radar collectées dans les régions de Bordeaux (France), Dakar (Sénégal) et Sétif (Algérie). Les radars côtiers de Bordeaux, de Dakar et de Sétif enregistrent respectivement toutes les cinq minutes, dix minutes et quinze minutes une image sous le format 512x512 pixels. Le paramètre physique représentatif de l'image radar est le facteur de réflectivité exprimé en (dBZ) et qui peut se convertir en taux de précipitations (mm/h) par des formules d'étalonnage adoptées pour chaque type de précipitation.

A titre d'illustration, ce résumé présente l'analyse des précipitations pour la région de Sétif.

3. Cartographies annuelles des intensités de précipitations:

Pour des besoins d'hydrologie et d'agriculture, nous avons élaboré la cartographie des intensités de précipitations pour les trois régions étudiées. La fig.1 donne les intensités de précipitations annuelles cumulatives pou la région de Sétif.

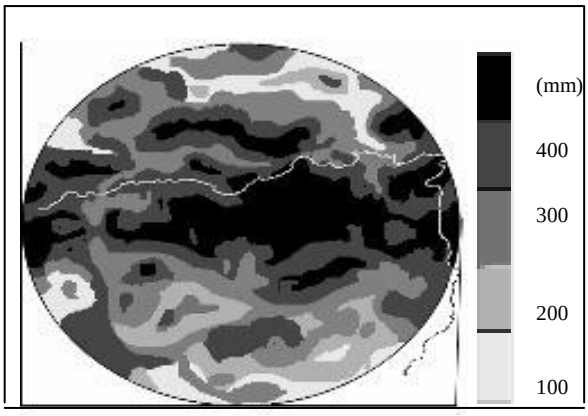


Fig.1. Cartographie des intensités de précipitations cumulatives annuelles pour la région de Sétif.

La fig.1 représente la répartition des intensités de précipitations pour la région de Sétif ; elle montre qu'il pleut plus sur terre que sur mer. De même, l'influence de l'orographie fait que les montagnes du Djurdjura, les monts des Bibans et des Babors sont plus arrosés que les autres régions.

3. Choix des zones d'étude :

Les images radars de la région d'étude, à savoir Sétif sont subdivisées en deux mailles (zones), l'une correspond à la mer et l'autre au continent. Chacune de ces deux zones est un carré de 50 km de côté (voir fig.2). Pour une analyse fine du phénomène, on a considéré des fenêtres de taille 5x5 pixels sur les zones (voir encore fig.2).

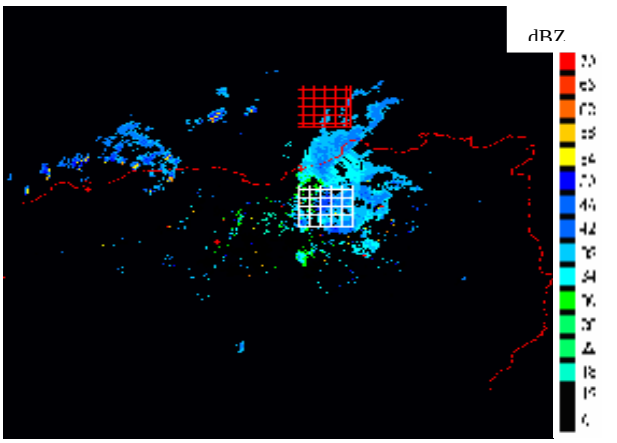


Fig.2. Zones d'étude sur l'image Sétif (rouge : zone de mer ; blanc : zone de terre).

4. Méthode de chaîne de Markov :

La modélisation markovienne d'un système aléatoire consiste d'abord en la construction d'une chaîne de Markov dont l'évolution de la chaîne représente l'évolution du système réel, afin de pouvoir en prédire

le comportement et les performances. Ainsi, l'analyse markovienne d'une série d'observations aléatoire pendant une période fait apparaître l'existence de dépendances stochastiques ; elle permet de constater la probabilité du passage d'un état vers un autre état à l'instant suivant. Autrement dit, tout système aléatoire peut être modélisé et analysé par la méthode des chaînes de Markov.

Dans notre étude, l'évolution des facteurs de réflectivité est présentée par la fig.3 pour les trois régions en terre et en mer. Cette figure donne les valeurs des facteurs de réflectivité moyens en fonction de la période d'observation aussi bien sur terre que sur mer (voir fig.2). Ces courbes sont obtenues pour une fenêtre d'analyse de taille 5x5 pixels. Pour une bonne représentation, on s'est limité à analyse le comportement des précipitations sur cent images. Notons que, ces cent images ont été collectées successivement.

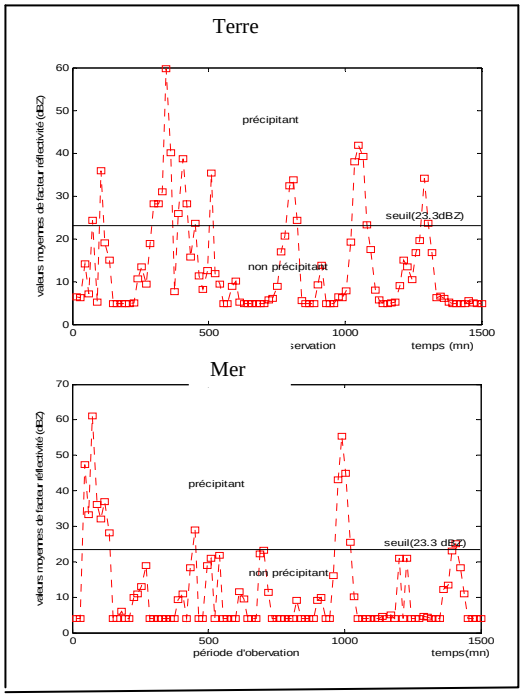


Fig.3. Valeurs moyennes du facteur de réflectivité pour une série de cent images.

5. Construction des séries d'observations :

Pour chaque fenêtre, nous effectuerons l'opération suivante : soit $E(n)$ le processus qui décrit l'état de la fenêtre à l'instant n ; une fenêtre est dite précipitante si E est égale à un et considérée non précipitante si E est égale à zéro. Pour mettre au point ces séries, nous avons utilisé le programme donné par la fig.IV.6. Après avoir choisi les zones nécessaires de mer et de terre, le programme calculera la moyenne de chaque fenêtre. Cette moyenne sera ensuite comparée au seuil de référence S (voir tab.IV.3). L'état E prend la valeur zéro si la moyenne M calculée en (dBZ) est inférieur au seuil S et la valeur un si la moyenne est supérieure ou égale au seuil S.

La table.1 donne la relation $Z-R$ utilisée et le seuil de référence retenu pour le site de Sétif. Le seuil de référence a été choisi de façon à diviser en deux classes l'intensité de précipitations (absence de précipitations ; présence de précipitation). Il est donné en taux de précipitations (mm/h) ainsi qu'en facteur de réflectivité radar (dBZ) correspondant.

Site	Relation $Z-R$	Seuil (S)	
		$Z(dBZ)$	$R(mm/h)$
Sétif	$Z=300R^{1,5}$	$23.3\,dBZ$	$0.8\,mm/h$

Tab.1. Relation Z-R et seuil retenu.

Dans notre étude, le processus $E(n)$ est le processus qui décrit l'état de la fenêtre à un instant n. Nous rappelons que l'unité (ou pas) de temps est de quinze minutes pour Sétif.

Dans toue la suite, nous supposerons que le phénomène étudié (précipitations) satisfait les hypothèses suivantes :

- L'espace d'état est décrit par les états $\{E_0, E_1\}$ pour l'ordre un et les doublets d'états $\{E_0E_0, E_0E_1, E_1E_0, E_1E_1\}$ pour l'ordre deux : il s'agit d'une chaîne de Markov à espace d'états fini.

- L'évolution de phénomène est aléatoire : il s'agit d'un processus stochastique.

- L'évolution future ne dépend que du présent pour l'ordre 1 et du présent et passé immédiat pour l'ordre 2 ; il vérifie la propriété de Markov (absence de mémoire) : il s'agit d'une chaîne de Markov.

- Les évolutions possibles d'un instant à l'instant suivant ne dépendent pas du temps ; le système vérifie la propriété d'homogénéité : il s'agit d'une chaîne de Markov homogène.

6. Résultats:

Nous allons présenter les résultats obtenus à l'aide des hypothèses markoviennes, qui seront scindés en deux parties. La première, exposera ceux obtenus en utilisant l'hypothèse des chaînes de Markov d'ordre 1. Subséquemment, la deuxième partie présentera les résultats de l'application de l'hypothèse des chaînes de Markov d'ordre 2. La programmation s'est faite en environnement MatLab. La démarche adoptée porte sur une fenêtre d'analyse de dimension 5x5 km en zone de terre et en zone de mer, qui est ensuite généralisée pour toute la zone d'étude.

Modèle markovien d'ordre 1 :

On applique dans cette partie les principes de base de l'analyse et la modélisation markovienne. Ainsi, nous avons choisi de commencer à analyser le comportement stochastique des précipitations à l'aide d'une chaîne de Markov d'ordre 1. Les premiers résultats sont comme suit:

Pour chaque fenêtre d'analyse, nous calculons la matrice markovienne d'ordre 1. A titre d'illustration, nous présentons les résultats obtenus pour une fenêtre en zone de terre et une fenêtre en zone de mer (voir tab.2):

Chaîne de Markov d'ordre 1.				
Fenêtres	p_{00}	p_{01}	p_{10}	p_{11}
En terre	88.2%	11.8%	40.9%	59.1%
En mer	90.9%	09.1%	39.2%	60.8%

Tab.2. Matrices de transitions d'ordre un pour une fenêtre en mer et une fenêtre en terre.

Suite à l'application de cette hypothèse, nous constatons que, pour les deux zones et quel que soit l'état initial, la probabilité de retrouver le même état à l'instant suivant est plus forte que celle d'avoir un état contraire.

Pour les deux zones, les probabilités se présentent ainsi

- Si un état est non précipitant, la probabilité pour qu'il soit suivi d'un état précipitant est plus faible en zone de mer qu'en zone de terre.

- Si un état est précipitant, la probabilité d'avoir un état non précipitant à l'instant suivant est plus grande en zone de terre qu'en zone de mer.

- En zone de mer, la probabilité pour qu'un état non précipitant soit suivi d'un état non précipitant est plus forte qu'en zone de terre.

- Par contre, la probabilité d'avoir un état précipitant faisant suite à un état précipitant est moins importante en zone de terre qu'en zone de mer.

La fig.4 illustre les distributions de probabilités des deux états pour le site de Sétif.

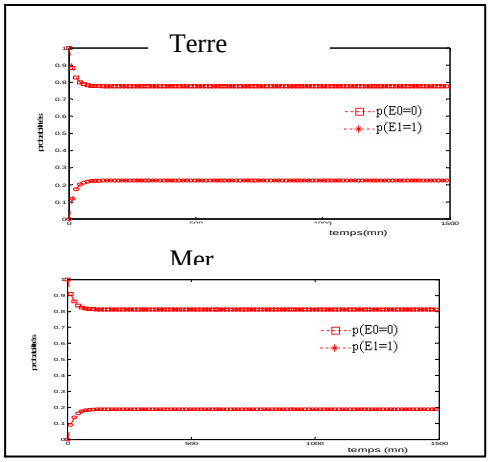


Fig.4. Distributions de probabilités.

Sur la fig.4, nous constatons que, la probabilité d'avoir un état précipitant est faible en zone de mer et en zone de terre. Notons que pour cet état, elle est plus élevée en zone de terre qu'en zone de mer. Par contre, pour le cas non précipitant, les valeurs des probabilités sont assez élevées pour les deux zones mais plus importantes en zone de mer qu'en zone de terre.

	Fenêtre de terre		Fenêtre de mer	
Probabilités	P_0	P_1	P_0	P_1
Probabilités à long terme	77.6%	22.4%	81.2%	18.8%
Probabilités à priori	77.1%	22.9%	83.8%	16.2%

Tab.3. Probabilités à long terme et à priori.

Conclusion de modèle d'ordre 1:

Nous retenons, par l'application de modèle d'ordre 1, que la probabilité de reconduire le même état est forte quelle que soit la zone. Le site présente des probabilités faibles quand on passe d'un état à un état différent. Aussi, les probabilités à long terme et les probabilités à priori sont quasiment identiques aussi bien en terre qu'en mer. Par conséquent, nous pouvons conclure que les précipitations sont bien décrites par le modèle Markovien du premier ordre à deux états sur terre que sur mer. La différence de comportement des précipitations constatée pour les deux zones de chaque site, peut se justifier par le fait que la température varie plus rapidement sur terre que sur mer. Par conséquent, un changement d'état est plus observé sur terre que sur mer. Donc, l'effet de mémoire est plus présent sur mer que sur terre.

Modèle markovien d'ordre 2 :

Lorsque l'on fait dépendre l'instant futur uniquement de l'instant précédent, la profondeur de la chaîne vaut l'unité. Il est légitime de penser que les précipitations à un instant t sont dépendantes au sens probabiliste des précipitations des instants précédents. On pourrait donc imaginer un modèle dont la probabilité d'avoir un état à l'instant t est dépendante non pas uniquement de l'instant précédent, mais des n instants précédents ; ainsi, on parle d'une chaîne de Markov d'ordre n.

Pour mieux prendre en considération le passé, nous allons utiliser dans cette partie l'hypothèse markovienne d'ordre 2. Dans le cas du modèle markovien d'ordre deux, nous nous intéressons aux doublets d'états (E_0, E_0) et (E_1, E_1). Ceci n'est qu'une conséquence de l'approche markovienne du modèle d'ordre 1. Pour les différents sites, les résultats sont comme suit :

Les probabilités de transitions pour l'hypothèse de deuxième ordre, sont résumées dans la table V.4.a :

Chaîne de Markov d'ordre 2.		
fenêtres	La fenêtre de terre	La fenêtre de mer
p_{000}	91.0%	93.5%
p_{001}	09.0%	06.5%
p_{010}	44.4%	20.0%
p_{011}	55.6%	80.0%
p_{100}	66.7%	100%
p_{101}	33.3%	00.0%
p_{110}	38.5%	37.4%
p_{111}	61.5%	62.6%

Tabl.4. Matrices de transitions d'ordre 2 pour une fenêtre en mer et une fenêtre en terre.

Nous constatons que, comme pour l'ordre 1, pour les deux zones et quel que soit l'état initial, la probabilité d'avoir une reconduction d'un état aux instants suivants est élevée.

Pour les deux zones, nous constatons que :

- Si deux états successifs sont non précipitants, la probabilité d'avoir un troisième état non précipitant est forte pour les deux zones. Elle est de 91.0% en zone de terre et de 93.5% en zone de mer. Par contre, la probabilité d'avoir le troisième état précipitant est très faible pour les deux zones. Elle est de 09.0% en zone de terre contre 06.5% en zone de mer.

- De même, la probabilité pour que deux états précipitants successifs soient suivis d'un état précipitant est aussi importante pour les deux zones. Elle est de 61.5% en zone de terre et de 62.6% en zone de mer. En revanche, elle devient faible pour les deux zones quand les états cités ci-dessus sont suivis par un état non précipitant.

- Si une séquence se compose d'un état précipitant suivi d'un état non précipitant, la probabilité d'avoir l'état suivant non précipitant est de 100% en zone de mer. Cette séquence est toujours suivie d'un état non précipitant. Elle vaut 66.7% en zone de terre. La probabilité pour que la séquence soit suivie d'un état précipitant est nulle en zone de mer et faible en zone de terre.

- Et si la séquence se compose d'un état non précipitant suivi d'un état précipitant, la probabilité d'avoir l'état suivant précipitant est très élevée en zone de mer et moins importante en zone de terre. La probabilité pour que cette séquence soit suivie d'un état précipitant est plus faible en zone de mer qu'en zone de terre.

Les courbes suivantes (fig.5) présentent les distributions de probabilités des doublet d'états (E_i, E_j).

Nous constatons à travers les courbes que, la probabilité d'avoir le doublet d'états (E_0, E_0) est plus forte en zone de mer qu'en zone de terre. Dans le cas de doublet d'états (E_1, E_1), les deux zones présentent des probabilités assez faibles.

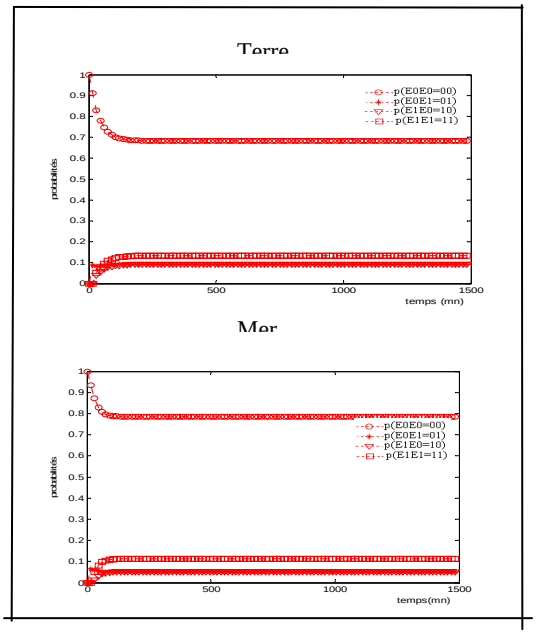


Fig.5. Distributions des probabilités.

	Fenêtre de terre		Fenêtre de mer	
Probabilités	$P_{(0,0)}$	$P_{(1,1)}$	$P_{(0,0)}$	$P_{(1,1)}$
Probabilités à long terme	68.2 %	13.3%	78.4%	10.8%
Probabilités à priori	68.0%	13.6%	70.0%	10.0%

Tab.5. Probabilités à long terme et à priori.

Conclusion de modèle d'ordre 2 :

Nous constatons, suite à l'hypothèse d'ordre 2, que la probabilité d'avoir de mêmes états successifs est élevée, cela confirme les résultats de modèle de premier ordre. En effet, les probabilités à long terme de deuxième ordre sont plus proches aux probabilités à priori que celles de premier ordre aussi bien pour les régions que pour les zones. Par conséquent, le modèle de deuxième ordre décrit mieux la persistance du phénomène. Cette application, donne une meilleure représentation des précipitations. De même, comme pour le modèle de premier ordre, un changement d'état est plus observé sur terre que sur mer et donc, l'effet de mémoire est plus présent sur mer que sur terre.

7. Conclusion générale :

Ce travail avait pour but de modéliser les précipitations dans trois régions où prévalent des climats différents et de rechercher une différence de description des précipitations sur la mer et le continent. Les résultats obtenus montrent que la probabilité d'avoir une situation non précipitante est élevée. Ceci explique le déficit en intensité de précipitations observé lors de cette dernière décennie en Algérie. tous ces résultats montrent que les précipitations sont bien décrites par les chaînes de Markov aussi bien sur terre que sur mer et quel que soit le climat prévalant dans la région .

Par ailleurs, l'exploitation des outils markoviens pour l'analyse et la modélisation du phénomène de précipitations semble être satisfaisante. Nous avons mis en évidence l'intérêt d'avoir recours à l'utilisation des chaînes de Markov en augmentant la profondeur, tout au moins dans le cadre de notre étude. Le passage à un modèle markovien d'ordre supérieur permet de traduire une hypothèse physique intéressante en matière du phénomène de précipitations, car la probabilité de transition vers un état dépendrait de plusieurs états ayant précédé. C'est sur cette réflexion que nous avons opté pour le modèle d'ordre 2. Ce choix nous a permis d'avoir des bons résultats puisque le modèle markovien d'ordre 2 donne une meilleure représentation du phénomène.

Bibliographie :

[1]Arnaud M., 1985. Contribution à l'étude stochastique markovienne des précipitations dans le bassin Adour-Garonne. *Thèse de Docteur*, Toulouse.

[2]Bergaoui Z. 1990. Modélisation stochastique des sécheresses annuelles et pluriannuelles. *Thèse de Doctorat d'État*, Tunis.

[3]Billingsley P., 1960. Statistical methods in Markov chains. *In Stanford meetings of the Institute of Mathematical Statistics*, Chicago, USA, 1960.

[4]Chèze I. et Jourdain S., 2003 : Calcul des quantiles de durées de retour de la température par la méthode gev. *In Calcul des températures à risque*, pages 1–50, Météo France DP/SERV/BEC Toulouse, France.

[5]Chiquet J., 2003 : Estimation des températures journalières à l'aide de techniques markoviennes. *Mémoire de DEA TIS*, Compiègne.

[6]Chrétienne Ph et Faure R., 1974 : *Processus stochastiques, leurs graphes, leurs usages*. France Offset- Aubin, Poitiers.

Modélisation de champs de vecteurs par bases de polynômes : application à l'analyse de la posture d'utilisateurs devant un écran d'ordinateur, via une webcam.

M. Druon[1] B. Tremblais[1] B. Augereau[1]

[1] Laboratoire Signal Image Communications - E.A. 4103
Université de Poitiers

Bt. SP2MI, Bvd M. et P. Curie, BP 30179
86962 Futuroscope Chasseneuil CEDEX, FRANCE

{druon, tremblais, augereau}@sic.univ-poitiers.fr

Résumé

Dans cet article, nous présentons une méthode originale et générale permettant d'approximer des champs de vecteurs et, plus spécialement, des champs de déplacement. Pour cela, nous utilisons une base orthonormée de polynômes multivariables pour exprimer ces champs comme des combinaisons linéaires de ces fonctions spécifiques. Dans un premier temps, nous présentons la partie théorique de notre méthode. Ensuite, nous démontrons la résistance au bruit de notre modèle. Nous terminons l'article en montrant, de façon expérimentale, que notre méthode peut être utilisée pour reconnaître les mouvements simples de la tête d'une personne située devant une webcam.

Mots clefs

Analyse du mouvement, analyse du comportement, gestuelle, polynômes orthogonaux, champs de vecteurs.

1 Introduction

Actuellement, il existe beaucoup de travaux concernant l'estimation du mouvement [1], [2], [3], [4], des comparatifs entre différentes méthodes [5] ou des algorithmes de débruitage [3]. En revanche, peu d'articles traitent de l'analyse du mouvement car c'est un domaine de recherche encore très récent. De plus, ces articles sont généralement axés sur un domaine d'étude spécifique. Par exemple, certains travaux étudient uniquement le comportement du visage ou du corps humain [6], [7], [8]...

La méthode présentée ici propose de caractériser tout type de mouvement comme une combinaison linéaire de polynômes issus d'une base orthonormée [9]. Contrairement aux articles cités précédemment, elle se veut la plus générale possible.

L'article s'organise de la facon suivante : la section 2 détaille la partie théorique de notre méthode. Sa résistance au bruit est démontrée dans la section 3. La section 4 présente, de façon expérimentale, le processus de reconnaissance mis en place. Finalement, nous donnons nos conclusions et perspectives en section 5.

2 Modélisation des champs de vecteurs

Soient $\mathcal{U} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ et $\mathcal{V} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ les fonctions permettant d'obtenir le déplacement du pixel $(x_1, x_2) \in \Omega$ dans un repère cartésien. Un champ de vecteurs peut alors être défini par l'application :

$$\begin{array}{rccc} \mathcal{F} : & \Omega \in \mathbb{R}^2 & \rightarrow & \mathbb{R}^2 \\ & (x_1, x_2) & \mapsto & (\mathcal{U}(x_1, x_2), \mathcal{V}(x_1, x_2)) \end{array}$$

Nous cherchons ici à étudier le mouvement de façon analytique. Pour cela, notre méthode consiste à approximer la fonction $\mathcal{F}$ par des combinaisons linéaires de polynômes.

Dans un premier temps, nous définissons un espace vectoriel de fonctions. Puis nous munissons cet espace d'un produit scalaire pour créer une base orthonormée de polynômes. Finalement, nous définissons, dans cette base, les opérations permettant d'obtenir l'expression analytique recherchée.

2.1 Définition de l'espace vectoriel

Soit $\mathcal{E}_p$ l'espace vectoriel des fonctions de $\Omega \subset \mathbb{R}^2$ dans $\mathbb{R}$ contenant les fonctions $\mathcal{U}$ et $\mathcal{V}$. Soit ϕ l'ensemble des éléments de $\mathcal{E}_p$. ϕ est composé de polynômes bivariables définis comme suit :

$$P_{K,L}(x_1, x_2) = \sum_{k=0}^{K} \sum_{l=0}^{L} a_{k,l} \, (x_1)^k \, (x_2)^l \qquad (1)$$

avec $K \in \mathbb{N}^+$ le degré maximal selon x_1, $L \in \mathbb{N}^+$ le degré maximal selon x_2 et $\{a_{k,l}\}_{l \in [0;L]}^{k \in [0;K]} \in \mathbb{R}^{K*L}$ l'ensemble des coefficients. Le degré du polynôme est alors $K + L$. Par la suite, pour simplifier la lecture de ce document, les

variables x_1 et x_2 sont omises. Par exemple, les polynômes $P_{K,L}(x_1, x_2)$ sont écrits $P_{K,L}$.

2.2 Définition de la base orthonormée

Pour construire une base orthonormée, nous munissons $\mathcal{E}_p$ du produit scalaire entre deux fonctions F_1 et F_2 de ϕ :

$$\langle F_1 \mid F_2 \rangle = \int_a^b \int_a^b F_1 \, F_2 \, \omega(x_1, x_2) \quad dx_1 dx_2 \qquad (2)$$

avec $\omega(x_1, x_2)$ une fonction de poids choisie selon le problème considéré. À partir de ce produit scalaire, il est possible de définir la distance entre ces deux fonctions :

$$\|F_1 - F_2\| = \sqrt{\langle F_1 - F_2 \mid F_1 - F_2 \rangle} \qquad (3)$$

Notre espace vectoriel $\mathcal{E}_p$ muni du produit scalaire définit une base $\mathcal{B}$ composée de polynômes. Pour normaliser cette base, tous ses éléments $\{P_1, P_2, \ldots, P_n\}$ doivent vérifier $\langle P_i \mid P_j \rangle = \delta^{ij}$. Nous cherchons donc à créer un ensemble de polynômes orthonormés. Pour cela, nous utilisons la méthode de Gram-Schmidt qui nous permet d'obtenir une base orthonormée de polynômes.

Dans cet article, nous nous limitons à l'étude des polynômes de Legendre. Ils sont définis par la formule de récurrence suivante :

$$\begin{cases} L_{0,0} = 1 \\ L_{1,0} = x_1 \\ L_{0,1} = x_2 \\ L_{i+1,j} = \dfrac{2i+1}{i+1} \, x_1 \, L_{i,j} \quad - \quad \dfrac{i}{i+1} \, L_{i-1,j} \\ L_{i,j+1} = \dfrac{2j+1}{j+1} \, x_2 \, L_{i,j} \quad - \quad \dfrac{j}{j+1} \, L_{i,j-1} \end{cases} \qquad (4)$$

Par définition, la fonction de poids associée à ces polynômes est $\omega(x_1, x_2) = 1$ et le domaine de définition est $[-1; 1]$. De ce fait, avant chaque opération, les coordonnées $(m, n) \in \mathbb{N}^{+2}$ des pixels d'une image de taille MxN devront être ramenées dans cet intervalle par une transformation affine.

Par la suite, nous appelons *degré D de la base* le degré le plus élevé des polynômes qui la constituent. Par exemple, une base de degré 2 est composée de l'ensemble des polynômes $L_{i,j}$ tels que $i + j \leq D$, soit six polynômes :

$\mathcal{B}$ =

	$(x_2)^0$	$(x_2)^1$	$(x_2)^2$
$(x_1)^0$	$L_{0,0}$	$L_{0,1}$	$L_{0,2}$
$(x_1)^1$	$L_{1,0}$	$L_{1,1}$	–
$(x_1)^2$	$L_{2,0}$	–	–

(5)

2.3 Projection d'un champ de vecteurs sur la base

La projection d'un champ de vecteurs sur une base de degré D est obtenue en calculant le produit scalaire entre les deux fonctions $\mathcal{U}$ et $\mathcal{V}$ associées au champ et chaque polynôme $L_{i,j}$ de la base. Les scalaires obtenus correspondent alors aux coefficients de deux polynômes $P_\mathcal{U}$ et $P_\mathcal{V}$, appelés par la suite *polynômes caractéristiques* :

$$\begin{cases} P_\mathcal{U} = \displaystyle\sum_{i=0}^{D} \sum_{j=0}^{D-i} \alpha_{i,j} \, L_{i,j} \quad \text{avec } \alpha_{i,j} = \langle \mathcal{U} \mid L_{i,j} \rangle \\ P_\mathcal{V} = \displaystyle\sum_{i=0}^{D} \sum_{j=0}^{D-i} \beta_{i,j} \, L_{i,j} \quad \text{avec } \beta_{i,j} = \langle \mathcal{V} \mid L_{i,j} \rangle \end{cases} \qquad (6)$$

Par exemple, étudions l'écoulement d'un fluide incompressible dans une cavité rigide à partir d'une séquence d'images fournie par le Laboratoire d'Études Aérodynamiques de Poitiers. Le champ de vecteurs correspondant, calculé par la méthode P.I.V. (*Particle Image Velocimetry*), est représenté Figure 1. La projection de ce champ sur une base de degré 1 (donc composée des trois polynômes $L_{0,0}$, $L_{0,1}$ et $L_{1,0}$) donne les polynômes caractéristiques suivants :

$$\begin{cases} P_\mathcal{U} = -1.35 \, L_{0,0} + 4.38 \, L_{0,1} + 0.04 \, L_{1,0} \\ P_\mathcal{V} = 1.04 \, L_{0,0} + 0.10 \, L_{0,1} - 0.80 \, L_{1,0} \end{cases} \qquad (7)$$

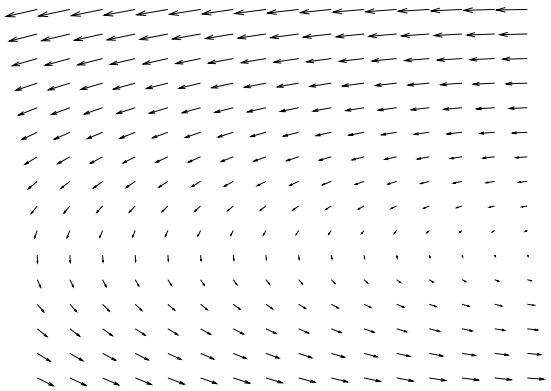


Figure 1 – *Champ de vecteurs, de taille 32x32, associé aux combinaisons linéaires définies en (7).*

2.4 Calcul d'un champ de vecteurs à partir des polynômes caractéristiques

Pour approximer un champ de vecteurs à partir de ses deux polynômes caractéristiques $P_\mathcal{U}$ et $P_\mathcal{V}$, nous déterminons les deux composantes de chaque vecteur en fixant les deux variables de chaque polynôme en fonction de la position du vecteur dans le champ :

$$\forall s, \forall t \in [-1, 1] \begin{cases} \mathcal{U}(s,t) = P_\mathcal{U}(s,t) \\ \mathcal{V}(s,t) = P_\mathcal{V}(s,t) \end{cases} \qquad (8)$$

Par exemple, les combinaisons linéaires suivantes :

$$\begin{cases} P_\mathcal{U} = 3 \, L_{1,0} + L_{2,0} - L_{0,0} - 2 \, L_{0,1} \\ P_\mathcal{V} = 2 \, L_{0,1} + 4 \, L_{2,0} - L_{1,0} \end{cases} \qquad (9)$$

avec $L_{i,j}$ les polynômes de Legendre issus d'une base de degré 2, telle que définie en (5), correspondent au champ de vecteurs présenté Figure 2.

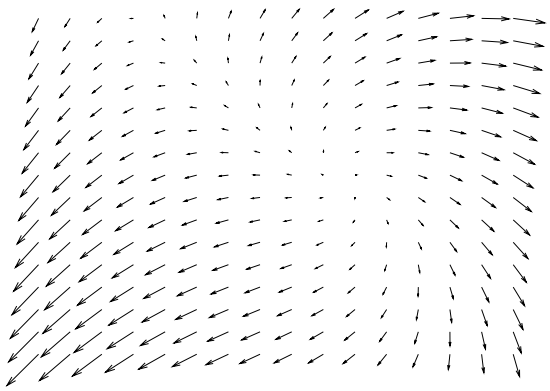


Figure 2 – *Champ de vecteurs, de taille 32x32, associé aux combinaisons linéaires définies en (9).*

2.5 Influence du degré de la base

L'approximation des champs de vecteurs par des combinaisons linéaires de polynômes entraîne un lissage des champs : plus le degré de la base est faible, plus ce lissage est important. Il permet donc de diminuer, de façon naturelle, le bruit. Par contre, un degré trop faible ne permet pas de modéliser des champs de vecteurs (*i.e.* des mouvements) complexes. Il est donc nécessaire d'adapter ce degré en fonction de la complexité des mouvements à analyser et de la quantité de bruit présent dans la séquence.

3 Résistance au bruit

Pour qu'une méthode de reconnaissance de mouvements soit fiable, elle doit être résistante au bruit. Dans cette section, nous présentons deux expériences mettant en évidence la robustesse de notre modèle.

3.1 Expérience 1

Le processus de test, représenté Figure 3, est le suivant : dans une base $\mathcal{B}$, deux polynômes caractéristiques $P_{\mathcal{U}o}$ et $P_{\mathcal{V}o}$ sont générés par combinaisons linéaires des différents polynômes de la base. Le champ de vecteurs $\mathcal{F}_o$ associé à ces deux polynômes est alors calculé. Un bruit Gaussien $\mathcal{G}$ est ajouté à ce champ $\mathcal{F}_o$ pour obtenir un champ bruité $\mathcal{F}_b$. Ce champ $\mathcal{F}_b$ est alors projeté sur la base $\mathcal{B}$ pour obtenir les deux polynômes caractéristiques $P_{\mathcal{U}b}$ et $P_{\mathcal{V}b}$ correspondant à ce champ $\mathcal{F}_b$. Finalement, le champ résultat $\mathcal{F}_r$ est calculé à partir des polynômes $P_{\mathcal{U}b}$ et $P_{\mathcal{V}b}$. La résistance au bruit est alors mesurée en comparant le champ de vecteurs initial $\mathcal{F}_o$ et le champ $\mathcal{F}_r$ calculé dernièrement.

Les tests sont effectués sur des champs de vecteurs de taille 320x240. Les champs initiaux sont générés aléatoirement (les coefficients des polynômes caractéristiques $P_{\mathcal{U}o}$ et $P_{\mathcal{V}o}$ sont tirés selon une distribution uniforme). Le bruit utilisé est un bruit Gaussien de moyenne nulle et dont l'écart-type $\sigma_{\mathcal{G}}$ est déterminé en fonction de la quantité de bruit que l'on souhaite ajouter : $\sigma_{\mathcal{G}} = \sqrt{\text{variance}(\mathcal{F}_o)/\text{RSB}}$.

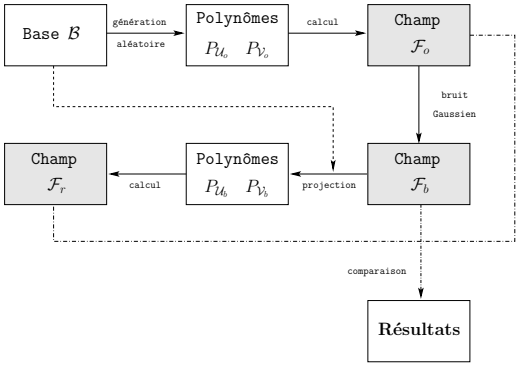


Figure 3 – *Processus de test à la résistance au bruit.*

Ici, nous faisons varier le rapport signal sur bruit (RSB) entre 0.1 et 2.0 par palier de 0.1. La mesure utilisée pour comparer les champs de vecteurs est l'erreur quadratique moyenne (EQM) entre deux champs de vecteurs.

La Figure 4 représente l'évolution de l'EQM entre les champs $\mathcal{F}_o$ et $\mathcal{F}_b$ (courbe claire) et les champs $\mathcal{F}_o$ et $\mathcal{F}_r$ (courbe foncée), en fonction du RSB et du degré de la base, variant ici entre 0 et 6.

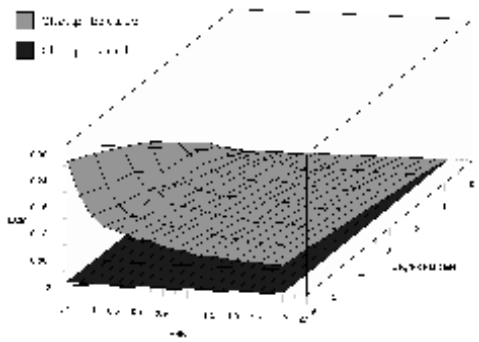


Figure 4 – *Influence du bruit sur le système mis en place.*

Même si le bruit ajouté est important, nous pouvons constater que le champ reconstruit grâce à cette méthode est très proche, au sens de l'EQM, du champ original, et cela quel que soit le degré de la base.

3.2 Expérience 2

Les Figures 5 et 6 montrent deux exemples de reconstruction. Dans les deux cas :

- (a) représente le champ original $\mathcal{F}_o$, généré aléatoirement à partir d'une base de degré 2 (Figure 5) ou de degré 6 (Figure 6) ;
- (b) représente le champ original faiblement bruité $\mathcal{F}_b^-$ ($\sigma_{\mathcal{G}} = 0.01$) ;
- (c) représente le champ original fortement bruité $\mathcal{F}_b^+$ ($\sigma_{\mathcal{G}} = 1.0$) ;
- (d) représente le champ reconstruit $\mathcal{F}_r^-$ à partir du champ faiblement bruité $\mathcal{F}_b^-$;
- (e) représente le champ reconstruit $\mathcal{F}_r^+$ à partir du champ fortement bruité $\mathcal{F}_b^+$.

Pour des raisons esthétiques, les champs présentés ici sont de taille 16x16.

Les tableaux Tableau 1 et Tableau 2 confirment, de façon numérique, les constatations visuelles précédentes. Ils fournissent la valeur minimale, la valeur maximale, la moyenne et l'écart-type des erreurs angulaires (Tableau 1) et des erreurs de norme (Tableau 2) entre :

- $(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_b^-)$: le champ original et le champ faiblement bruité ;
- $(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_b^+)$: le champ original et le champ fortement bruité ;
- $(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_r^-)$: le champ original et le champ reconstruit à partir du champ faiblement bruité ;
- $(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_r^+)$: le champ original et le champ reconstruit à partir du champ fortement bruité.

L'échantillon utilisé pour calculer ces valeurs est composé de 10 champs de taille 320x240 générés aléatoirement à partir d'une base de degré 6.

L'erreur angulaire est calculée de la façon suivante :

$$E_{\text{angle}} = cos^{-1}\left(\frac{\mathcal{F}_o}{||\mathcal{F}_o||} \cdot \frac{\mathcal{F}_r}{||\mathcal{F}_r||}\right) \tag{10}$$

Elle est exprimée en degré. L'erreur de norme est calculée ainsi :

$$E_{\text{norme}} = \frac{||\mathcal{F}_o||}{||\mathcal{F}_r||} \tag{11}$$

Tableau 1 – *Erreur angulaire.*

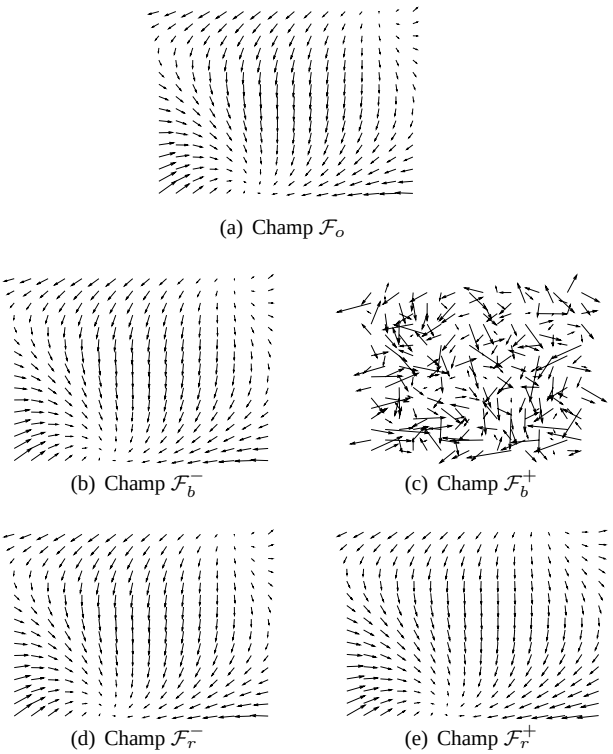
	Min.	Max.	Moyenne	Écart-type
$(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_b^-)$	0.0	96.97	1.07	1.41
$(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_b^+)$	0.0	179.99	66.47	48.71
$(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_r^-)$	0.0	65.84	0.24	0.58
$(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_r^+)$	0.0	153.77	1.88	3.09

Tableau 2 – *Erreur de norme.*

	Min.	Max.	Moyenne	Écart-type
$(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_b^-)$	0.41	3.26	1.01	0.03
$(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_b^+)$	0.01	140.69	0.61	1.10
$(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_r^-)$	0.57	4.38	0.99	0.02
$(\mathcal{F}_o) \leftrightarrow (\mathcal{F}_r^+)$	0.24	40.43	1.01	0.19

Nous pouvons constater, d'après le tableau Tableau 1, que l'erreur angulaire entre un champ original et un champ reconstruit est plus faible que l'erreur angulaire entre un champ original et un champ bruité. En effet, la moyenne et l'écart-type sont plus proches de 0. De même, d'après le tableau Tableau 2, nous pouvons constater que l'erreur de norme entre un champ original et un champ reconstruit est plus faible que l'erreur de norme entre un champ original et un champ bruité car la moyenne est plus proche de 1 et l'écart-type est plus proche de 0.

Ces deux expériences montrent la robustesse au bruit de notre modèle.

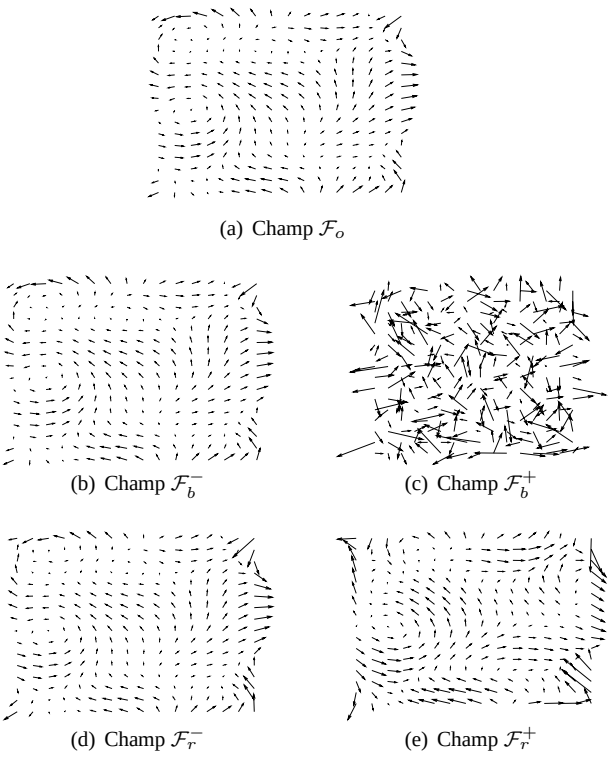


(a) Champ $\mathcal{F}_o$

(b) Champ $\mathcal{F}_b^-$ (c) Champ $\mathcal{F}_b^+$

(d) Champ $\mathcal{F}_r^-$ (e) Champ $\mathcal{F}_r^+$

Figure 5 – *Reconstruction de champs de vecteurs de taille 16x16 en utilisant une base de degré 2.*



(a) Champ $\mathcal{F}_o$

(b) Champ $\mathcal{F}_b^-$ (c) Champ $\mathcal{F}_b^+$

(d) Champ $\mathcal{F}_r^-$ (e) Champ $\mathcal{F}_r^+$

Figure 6 – *Reconstruction de champs de vecteurs de taille 16x16 en utilisant une base de degré 6.*

4 Application

Nous avons vu, au chapitre 2.3, comment définir un champ de vecteurs à l'aide de deux polynômes caractéristiques. Pour étudier le mouvement d'une séquence, c'est à dire un ensemble de champs de vecteurs, nous allons étudier l'évolution dans le temps des coefficients de ces polynômes caractéristiques.

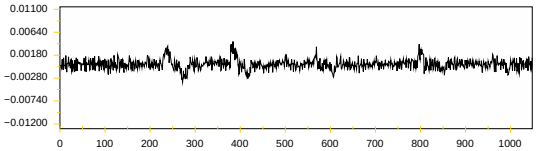
L'application présentée ici a pour but d'analyser le mouvement contenu dans une séquence d'images. Cette vidéo provient du projet SERIBEL[1]. Elle montre le visage d'une personne tournant la tête de gauche à droite et de haut en bas, de façon aléatoire (*i.e.* sans séquence prédéfinie). Elle se compose de 1050 images de taille 320x240, acquises via une webcam. La Figure 7 montre deux images de cette séquence.

(a) Position initiale (b) Position après un mouvement vers la gauche

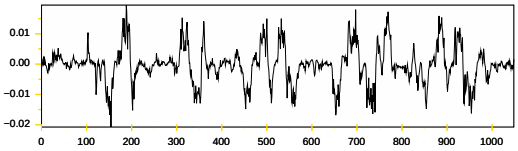
Figure 7 – *Exemple de postures, au cours du temps.*

À partir de la séquence originale, tous les champs de vecteurs sont extraits par une méthode différentielle d'estimation du mouvement apparent que nous avons présenté dans [10]. Cette méthode est fondée sur l'hypothèse de la conservation de la luminance des pixels dans une image [1] et sur l'utilisation d'équations aux dérivées partielles (E.D.P.) de lissage directionnel. Celle-ci utilise un opérateur différentiel, appelé tenseur de structure, permettant de déterminer localement, à partir de données spatio-temporelle, la direction du mouvement apparent. Cette direction est par ailleurs utilisée pour lisser la séquence d'images dans la direction du mouvement. Une évaluation du champ dense et régularisé est ainsi obtenue. Pour chacun de ces champs, les polynômes caractéristiques sont alors calculés par projections sur la base. Finalement, le mouvement est déterminé en étudiant les variations dans le temps des coefficients de ces polynômes.

La base utilisée ici est de degré 2. Nous obtenons donc, pour chaque paire d'images, douze coefficients : six pour le polynôme $P_{\mathcal{U}}$ et six pour le polynôme $P_{\mathcal{V}}$ (cf. Eq. 6). Pour savoir si ces coefficients sont corrélés, nous faisons une analyse en composantes principales (A.C.P.). Elle nous indique ici que, pour $P_{\mathcal{U}}$, près de 83 % de l'information est portée par un seul axe factoriel $\vec{E}(P_{\mathcal{U}})$ et pour $P_{\mathcal{V}}$, plus de 94 % de l'information est également contenue dans un seul axe $\vec{E}(P_{\mathcal{V}})$. Dans notre cas, étudier le mouvement de la séquence revient alors à étudier l'évolution des coefficients de $P_{\mathcal{U}}$ après projection sur $\vec{E}(P_{\mathcal{U}})$ et ceux de $P_{\mathcal{V}}$ après projection sur $\vec{E}(P_{\mathcal{V}})$ (cf. Figure 8).



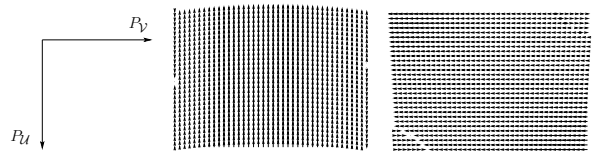
(a) Projection des coefficients de $P_{\mathcal{U}}$ sur $\vec{E}(P_{\mathcal{U}})$



(b) Projection des coefficients de $P_{\mathcal{V}}$ sur $\vec{E}(P_{\mathcal{V}})$

Figure 8 – *Projection des coefficients sur un axe factoriel.*

La signification de ces courbes est obtenue en étudiant les valeurs des deux vecteurs propres $\vec{v_1}(P_{\mathcal{U}})$ et $\vec{v_1}(P_{\mathcal{V}})$ obtenus durant l'A.C.P. Les deux champs de vecteurs $\mathcal{F}_{P_{\mathcal{U}}}$ et $\mathcal{F}_{P_{\mathcal{V}}}$ (cf. Figure 9), calculés en faisant une combinaison linéaire des polynômes P de la base pondérés par les valeurs des deux vecteurs propres $\vec{v_1}(P_{\mathcal{U}})$ et $\vec{v_1}(P_{\mathcal{V}})$, permettent d'interpréter ces résultats.



(a) Repère utilisé (b) Champ $\mathcal{F}_{P_{\mathcal{U}}}$ (c) Champ $\mathcal{F}_{P_{\mathcal{V}}}$

Figure 9 – *Champs de référence permettant d'interpréter les résultats obtenus.*

Nous pouvons constater que la plupart des vecteurs du champ $\mathcal{F}_{P_{\mathcal{U}}}$ est orientée verticalement vers le haut. Une croissance des coefficients se traduit donc par un mouvement de tête vers le haut tandis qu'une décroissance par un mouvement vers le bas. Le même raisonnement peut s'appliquer au champ $\mathcal{F}_{P_{\mathcal{V}}}$: une croissance correspond à un mouvement vers la gauche et une décroissance à un mouvement vers la droite.

Ces courbes représentent les mouvements effectués dans le temps et non la position de la tête par rapport à la position de référence. Comme en cinématique le vecteur vitesse est la dérivée du vecteur position, étudier la position de la tête au cours du temps revient à étudier l'intégrale des deux courbes précédentes. Nous obtenons alors les courbes représentées Figure 10.

[1] Stratégies Expertes de Recherche d'Informations Bibliographiques En Ligne - TCAN CNRS. Projet visant à préciser le potentiel et les limites d'enregistrements vidéo à basse résolution pour l'analyse du comportement d'un sujet effectuant une recherche d'information, en situation de travail sur un ordinateur.

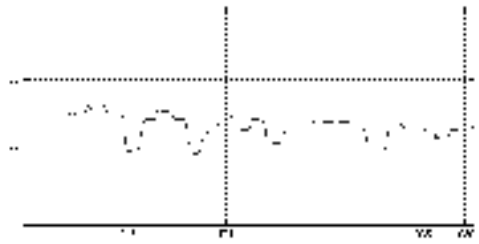
(a) Image n˚1

(b) Image n˚246

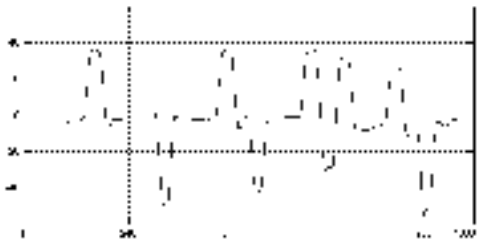
(c) Image n˚474

(d) Image n˚935

(e) Image n˚1030



(f) Position verticale de la tête, au cours du temps



(g) Position horizontale de la tête, au cours du temps

Figure 10 – *Positions verticales et horizontales de la tête au cours du temps. Les lignes verticales en pointillé correspondent aux images (a) - (e).*

Dans le cadre du projet SERIBEL, ces travaux ont été intégrés dans un logiciel d'analyse automatique de postures. Celui-ci a pour but d'accélérer et de simplifier les études menées par des cogniticiens. Une capture d'écran de ce logiciel est présenté Figure 11.

Figure 11 – *Logiciel permettant d'automatiser l'analyse de postures d'un utilisateur devant un écran d'ordinateur.*

5 Conclusion

Ce travail pose les bases d'une méthode de reconnaissance de mouvements. Nous avons proposé une méthode originale et générale de modélisation des champs de vecteurs. Cette méthode est robuste au bruit et peut être utilisée pour étudier des mouvements simples.

Par la suite, nous envisageons, d'une part, de tester la méthode sur des mouvements plus complexes (tels que des rotations, des zooms, des panoramiques...) ou issus de la mécanique des fluides (vortex, cisaillement...), et, d'autre part, d'utiliser d'autres familles de polynômes orthogonaux pour générer la base.

Finalement, nous avons vu, au chapite 2.3, que l'ensemble des coefficients $\alpha_{i,j}$ et $\beta_{i,j}$ forme un modèle associé à un mouvement et relatif à la base de polynômes choisie. Ceci nous permet de mettre en évidence la capacité de notre méthode à réduire la taille des données nécessaires pour exprimer un champ de vecteurs (i.e. un mouvement). Il serait donc intéressant d'étudier la possibilité d'utiliser notre méthode pour faire de la compression video.

Références

[1] B.K.P. Horn et B.G. Schunck. Determining optical flow. *Artificial IntelligenceI*, August 1981.

[2] B.D. Lucas et T. Kanade. An iterative image registration technique with an application to stereo vision. Dans *DARPA81*, pages 121–130, 1981.

[3] H. Scharr et H. Spies. Accurate optical flow in noisy image sequences using flow adapted anisotropic diffusion. 20(6) :537–553, July 2005.

[4] F. Lauze, P. Kornprobst, et E. Mémin. A coarse to fine multiscale approach for linear least squares optical flow estimation. *British Machine Vision Conference*, 2 :767–776, 2004.

[5] J.L. Barron, D.J. Fleet, et S.S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1) :43–77, 1994.

[6] C. Cedras et M. Shah. Motion-based recognition : A survey. *IVC*, 13(2) :129–155, March 1995.

[7] Jessica JunLin Wang et Sameer Singh. Video analysis of human dynamics - a survey.

[8] Christoph Bregler. Learning and recognizing human dynamics in video sequences. Dans *CVPR '97 : Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, Washington, DC, USA, 1997. IEEE Computer Society.

[9] Charles F. Dunkl et Yuan Xu. *Orthogonal polynomials of several variables*. Cambridge University Press, 2001.

[10] B. Augereau, B. Tremblais, et F. Fernandez-Maloigne. Vectorial computation of the optical flow in color image sequences. Dans *13th Color Imaging Conference*, november 2005.

Un Algorithme robuste de Segmentation Vidéo pour des Applications Temps Réels

M. El Hassani, D. Rivasseau
Philips Semiconductors
Caen, France

S. Jehan-Besson, M. Revenu
Laboratoire GREYC
Caen, France

D. Tschumperlé, L. Brun
Laboratoire GREYC
Caen , France

M. Duranton
Recherche Philips
Eindhoven,Pays-Bas

Résumé

Dans ce papier, nous proposons un algorithme de segmentation vidéo temps réel conçu pour être stable temporellement. Notre algorithme est basé sur une segmentation spatiale de l'image en régions homogènes qui sont mises à jour au fil de la séquence en utilisant des informations de mouvement. En ce qui concerne la segmentation spatiale, nous proposons une méthode ascendante de segmentation basée sur des fusions successives. Grâce à un ordre de fusion spécifique et à un seuil adaptatif pour le prédicat, la méthode donne de très bons résultats sur des images naturelles (même pour les régions texturées) avec peu de paramètres à régler. De manière à améliorer la consistance temporelle de la segmentation pour une séquence d'images, nous incorporons une information de mouvement via un masque basé sur la détection des changements d'intensités entre deux images. Ce masque est conçu en utilisant à la fois les différences d'intensités entre image successives et la segmentation de l'image précédente. Notre algorithme tourne en temps réel sur un processeur TriMedia pour des séquences de format CIF (Common Intermediate Format).

Mots clefs

Segmentation basée régions, Fusion de régions, Traitement de Séquences Vidéo, Détection du mouvement, Consistance temporelle, Implémentation matérielle, Temps réel.

1 Introduction

La segmentation de vidéos en régions homogènes est cruciale pour de nombreuses applications. Parmi elles, on peut citer l'estimation de mouvement basée régions et la conversion du 2D au 3D où la segmentation est utilisée pour l'estimation de la profondeur. La tendance actuelle se dirige également de plus en plus vers des traitement adaptés au contenu de l'image (filtrage, rehaussement, compression orientée objet) et donc basés sur une segmentation initiale de la vidéo. Toutes ces applications nécessitent l'obtention d'une segmentation précise et stable temporellement. Par ailleurs, les applications multimédia visées nécessitent un algorithme de traitement des séquences en temps réel.

Les méthodes de segmentation spatiale peuvent être divisées en deux catégories, les méthodes basées régions et les méthodes basées contour. Dans la première catégorie [1], les transitions entre pixels sont calculées et les composantes connexes peuvent alors être extraites. Le principal inconvénient de ces approches est que le calcul du gradient est parfois imprécis et très sensible au bruit. De plus il est alors difficile de tenir compte de propriétés statistiques des régions considérées. La seconde catégorie de méthodes (i.e. basées régions) est donc plus souvent utilisée et c'est dans cette catégorie que se situe notre algorithme. Nous nous intéressons plus particulièrement ici aux méthodes ascendantes qui opèrent par fusions successives [2, 3, 4, 5]. Dans ces méthodes, deux points importants sont à considérer : l'ordre de fusion et le critère de similarité.

Lorsque l'on traite de segmentation vidéo, la dimension temporelle doit être ajoutée et la segmentation en régions doit être stable temporellement. De nombreuses approches ont été testées. Quelques auteurs considèrent le temps comme une dimension supplémentaire et la vidéo devient alors un volume 3D [6]. D'autres approches utilisent une information de mouvement comme la détection des changements d'intensités ou les vecteurs mouvement [7, 8]. Nous n'abordons pas ici le suivi d'objet qui consiste à suivre un objet au cours du temps (voir par exemple les travaux [9]).

Dans ce papier, nous proposons une segmentation spatiale basée sur la fusion de régions. Cette fusion est faite dans un ordre spécifique qui est fonction de la différence d'intensité entre les pixels voisins. Par ailleurs, un critère de similarité basé sur les différences de moyennes entre régions est utilisé pour la fusion. La fusion est arrêtée à l'aide d'un seuil adaptatif justifié par une modélisation statistique de l'image. Nous nous sommes basés sur les travaux de [2] en proposant une modélisation statistique de l'image qui conduit à un seuil plus approprié pour la segmentation en temps réel. Notre méthode donne une segmentation très satisfaisante pour des images naturelles (même pour des images texturées) avec peu de paramètres à régler. De manière à améliorer la consistance temporelle, nous proposons d'ajouter l'information de mouvement. Comme l'estimation du mouvement entre deux images est une opération coûteuse, nous avons choisi de combiner un masque de détection des changements d'illumination avec les informations régions contenues dans la segmentation spatiale de l'image précédente. En faisant des comparaisons au niveau pixellique et au niveau région, on obtient un algorithme très

efficace pour un faible coût calcul.
Notre algorithme tourne en temps réel sur des séquences de format CIF sur le processeur TriMedia. De plus les résultats expérimentaux montrent l'applicabilité de notre méthode sur des séquences d'image que ce soit en terme de qualité de la segmentation obtenue ou de stabilité temporelle.
Ce papier est organisé de la manière suivante. L'algorithme de segmentation spatiale est détaillée dans la section 2. L'introduction d'informations temporelles pour améliorer la consistance est détaillée en section 3. Enfin dans la section 4, nous développons l'implantation de notre algorithme. Les résultats expérimentaux sont finalement donnés section 5.

2 Segmentation spatiale

Considérons une image I de largeur W et de hauteur H, $D = \{1....W\} \times \{1....H\}$ est le domaine de l'image. $I(p,n)$ l'intensité du pixel de position $p = (x,y)^T$ dans l'image n.
La segmentation d'une image en régions consiste à trouver une partition pertinente de l'image en m régions $\{S_1, S_2, .., S_m\}$. L'algorithme proposé part du niveau pixellique et procède par fusions successives pour aboutir à la segmentation finale (méthode ascendante). Un ordre spécifique de fusion est utilisé ainsi qu'un seuil adaptatif pour stopper les fusions. Ces deux étapes sont détaillées ci-après et nous donnerons ensuite l'algorithme complet de segmentation.

2.1 Ordre de fusion

L'ordre de fusion est basé sur les poids des transitions comme dans [2, 10]. L'idée est de fusionner d'abord ce qui est similaire avant de fusionner ce qui est différent. Une transition e est un couple de pixels (p,p') en 4-connexité. La similarité entre pixels est mesurée en calculant la distance entre les intensités des deux pixels. Pour des images couleur, la similarité est calculée de la manière suivante :

$$w(p,p') = \sqrt{\sum_{I \in \{Y,U,V\}} (I(p,n) - I(p',n))^2}. \qquad (1)$$

Nous avons choisi l'espace couleur YUV qui est le format couleur utilisé pour le traitement des vidéos ce qui nous évite ainsi une conversion. De plus, nous avons trouvé que l'espace couleur YUV procure une partition de l'image qui est, subjectivement, de qualité meilleure que celle obtenue avec l'espace RGB. L'espace $L^*a^*b^*$ donne des résultats légèrement meilleurs que l'espace YUV, mais au prix d'une implémentation coûteuse [11].
Les transitions sont ensuite triées dans l'ordre croissant de leurs poids w et les couples de pixels correspondant sont traités dans cet ordre pour la fusion. En ce qui concerne l'implémentation, l'image est seulement parcourue deux fois pour ce tri. Le premier parcours permet de calculer le nombre de transitions de même poids, nombre qui est stocké dans une table. Cette table est ensuite utilisée pour allouer la mémoire pour ces transitions. Le second permet de stocker chaque transition dans la partie de la mémoire correspondante.

2.2 Critère de fusion

Etant données deux régions S_1 et S_2, nous voulons savoir si ces deux régions doivent être fusionnées. En conséquence, un critère de similarité entre régions doit être choisi et évalué. En comparant la valeur de ce critère à un seuil, les régions seront fusionnées ou non. Le choix du seuil est souvent difficile. Dans ce papier, nous utilisons un seuil adaptatif qui dépend de la taille de la région considérée. L'utilisation d'un tel seuil est justifiée par le biais d'inégalités statistiques comme dans [2]. Nous proposons ici une interprétation statistique plus simple de l'image qui nous conduit à un critère plus adapté pour une implémentation temps réel. Nous présentons d'abord ici le prédicat de fusion utilisé, puis sa démonstration.

Prédicat de fusion. La moyenne des intensités de la région S_i est calculé de la manière suivante :

$$\overline{S_i} = \frac{1}{|S_i|} \sum_{k=1}^{k=|S_i|} I_i(p_k)$$

où $I_i(p_k)$ est l'intensité du $k^{ème}$ pixel de la région S_i.
Le prédicat de fusion est alors :

$$P(S_1,S_2) = \begin{cases} vrai & \text{si} \quad (\overline{S_1} - \overline{S_2})^2 \leq Qg^2(\frac{1}{|S_1|} + \frac{1}{|S_2|}) \\ faux & \text{sinon} \end{cases} \qquad (2)$$

avec g le niveau maximum de I ($g = 255$ pour les images de composantes en précision 8 bits). LA notation $|S_i|$ représente la taille de la région S_i. Le paramètre Q permet de régler la finesse de la segmentation. Dans les expériences, nous choisissons $Q = 2$ qui donne de bons résultats pour le format vidéo CIF (taille des images 352×288).

Justification statistique du prédicat. Classiquement, l'image I est considérée comme l'observation d'une image parfaite I^* et les pixels sont alors des observations d'un vecteur de variables aléatoires (v.a) noté $\mathbf{X} = (X_1, .., X_n)^T$. Le prédicat de fusion est basé sur l'inégalité de McDiarmid [12] donnée ci-après :

Theorem 2.1 *(L'inégalité de McDiarmid) Soit* $\mathbf{X} = (X_1, X_2, ..., X_n)$ *une famille de* n *v.a. indépendantes avec* X_k *prenant ses valeurs dans un ensemble* A_k *pour chaque* k*. Supposons que la valeur réelle de la fonction* f *definie sur* $\prod_k A_k$ *satisfasse* $|f(x) - f(x')| \leq c_k$ *si les vecteurs* x *et* x' *diffèrent seulement sur leurs* $k^{ème}$ *coordonnée. Soit* $E(f(\mathbf{X}))$ *l'espérance de* $f(\mathbf{X})$ *alors quelque soit* $\tau \geq 0$*,*

$$Pr(|f(\mathbf{X}) - E(f(\mathbf{X}))| > \tau) \leq 2\exp\left(-2\tau^2 / \sum_k c_k^2\right) \qquad (3)$$

Nous nous intéressons ici à deux régions adjacentes S_1 et S_2. Dans ce cas, nous considérons le vecteur de variables aléatoires suivant :

$$(X_1, ..., X_n) \quad = \quad \left(I_1^*(p_1), ..., I_1^*(p_{|S_1|}), I_2^*(p_1), ..., I_2^*(p_{|S_2|})\right)$$

Avec $I_i(p_j)$ l'intensité du $j^{ème}$ pixel de S_i correspondant à l'observation de la variable aléatoire $I_i^*(p_j)$. Dans ce cas, la taille du vecteur de variables aléatoires est $n = |S_1| + |S_2|$. De manière à appliquer le théorème 2.1, nous choisissons comme fonction $f(\mathbf{x}) = \left(\overline{S_1} - \overline{S_2}\right)$. Pour cette fonction, nous trouvons que $\sum_k c_k^2 = g^2 \left(\frac{1}{|S_1|} + \frac{1}{|S_2|}\right)$.
Par inversion du théorème, nous avons avec une probabilité d'au moins $1 - \delta$ ($0 < \delta \leq 1$) :

$$|(\overline{S_1} - \overline{S_2}) - E\left(\overline{S_1} - \overline{S_2}\right)| \leq g\sqrt{Q\left(\frac{1}{|S_1|} + \frac{1}{|S_2|}\right)}$$

avec $Q = \frac{1}{2} * ln(\frac{2}{\delta})$, et $E(\overline{S})$ l'espérance de $\overline{S}$. Si S_1 et S_2 appartiennent à la même région dans I^*, l'espérance $E\left(\overline{S_1} - \overline{S_2}\right)$ sera nulle et le prédicat suit.

2.3 Algorithme de segmentation

L'algorithme 1 donne le traitement complet pour la segmentation spatiale. L'algorithme de fusion utilise la structure de données "UNION-FIND"[13]. La fonction $UNION$ fusionne deux régions en une seule région et la fonction $FIND$ permet d'identifier la région d'appartenance d'un pixel.

Algorithm 1 La segmentation spatiale

Calcul des poids des transitions et de l'histogramme correspondant
Tri par ordre croissant des transitions en fonction de la valeur de leur poids w.
for $i := 1$ to $2|I|$ **do**
  Lecture de la $i^{ème}$ transition : (p_1, p_2) ;
  $S_1 = FIND(p_1)$;
  $S_2 = FIND(p_2)$;
  if $P(S_1, S_2) = True$ **then**
    $UNION(S_1, S_2)$
  end if
end for

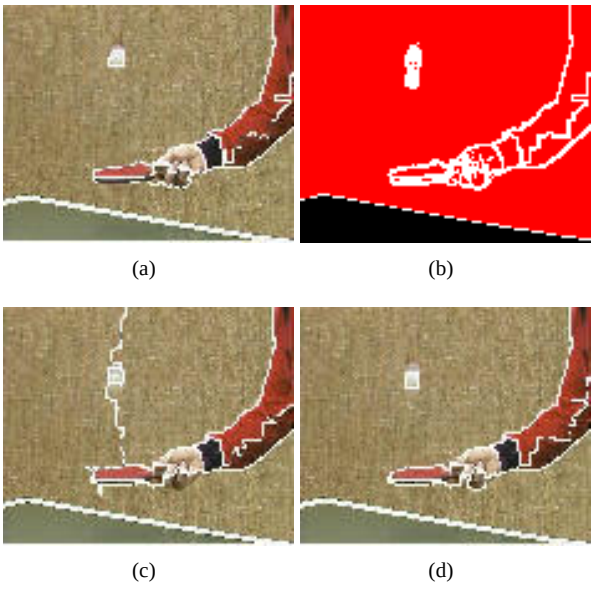
3 Amélioration de la consistance temporelle

La qualité d'une segmentation vidéo dépend non seulement de la bonne séparation entre les régions significatives dans l'image, mais aussi de la consistance temporelle de cette séparation. En effet, si dans deux images successives, une même région est segmentée différemment, ce qui peut être dû au bruit, ou à des situations particulières comme l'occlusion, la deocclusion, alors les résultats de la segmentation seront difficilement exploitables dans des applications comme l'amélioration de l'image ou la conversion 2D/3D. Quelques auteurs [8] utilisent les vecteurs de mouvements pour améliorer la consistance temporelle de la segmentation vidéo. Or l'estimation de mouvement est très coûteuse en temps de calcul et n'est pas toujours fiable. Dans ce papier nous utilisons un masque de changement d'intensités (CDM) combiné avec la segmentation spatiale pour améliorer la cohérence temporelle de la segmentation.
Le CDM est calculé en utilisant la différence d'intensités entre les images et la segmentation de l'image précédente. Premièrement, on détecte les variations d'intensité des pixels en utilisant la différence entre deux images consécutives. Ensuite, nous exploitons la segmentation de l'image précédente afin de classifier les pixels au niveau région. Si on considère l'image courante $I(:, n)$ et l'image précédente $I(:, n-1)$, La différence entre les images FD est donnée par $FD(p) = |I(p, n) - I(p, n-1)|$. Cette différence est classiquement seuillée pour distinguer les variations d'intensité des pixels dues aux objets en mouvement de celles dues au bruit.
Soit L Le résultat de seuillage de l'image FD. Un pixel p, avec $L(p) = 1$ est un pixel classifiée comme changeant, c'est à dire que pour ce pixel, la variation d'intensité entre deux images est supérieure à un seuil. Ensuite on utilise la segmentation précédente pour convertir le CDM du niveau pixel au niveau région, ce qui est plus fiable. Pour chaque région dans la segmentation précédente S_i, on calcule $\tau(S_i)$, qui représente le ratio des pixels changeants dans la région S_i. Nous avons $\tau(S_i) = \frac{N_{i,changeant}}{|S_i|}$, ou $N_{i,changeant}$ est le nombre de pixels changeant dans la région. Les pixels sont classifiés par la suite en trois catégories :

$$CDM(p) = \begin{cases} 0 & si\ (\tau(S_i) \leq tr_2). \\ 1 & si\ (\tau(S_i) > tr_2)\ et\ (L(p) = 1). \\ 2 & si\ (\tau(S_i) > tr_2)\ et\ (L(p) = 0). \end{cases} \quad (4)$$

où tr_2 est une constante positive. Plus cette constante est importante, plus la segmentation sera stable mais on risque alors de ne pas segmenter correctement les zones en mouvement. La valeur du seuil tr_2 est donc choisie faible afin de ne pas manquer une région en mouvement. Dans nos expériences, nous avons pris $tr_2 = 0.01$ (i.e. on considère qu'une région a bougé, si elle contient au moins 1% de pixels changeant, sinon elle est considérée comme statique).
Les pixels appartenant à une région statique sont étiquetés en utilisant $CDM(p) = 0$. Les deux autres étiquettes, $1, 2$, concernent les pixels dans les régions qui ont bougé. Selon la valeur de la différence entre images, appelée ici FD, Le pixel est qualifié comme étant changeant $CDM(p) = 1$ ou non changeant $CDM(p) = 2$. Une telle classification est alors utilisée pour segmenter l'image courante. Premièrement, Les régions statiques sont gardées telles qu'elles étaient segmentés dans l'image précédente. Deuxièmement, on applique un algorithme d'extraction de composantes connexes (CCL)[14] afin d'extraire les com-

posantes connexes des pixels ayant $CDM(p) = 2$. Cette deuxième étape construit des germes à partir de la segmentation précédente. Ces germes lient la segmentation précédente à la segmentation courante tout en améliorant la consistance temporelle. Troisièmement on applique la segmentation spatiale algorithme.1 seulement sur les transitions (p, p') qui connectent les pixels changeant (i.e. $CDM(p) = 1$, $CDM(p') = 1$), et ceux connectant les pixels changeant avec les germes construits dans la deuxième étape (i.e. ($CDM(p) = 1$ et $CDM(p') = 2$) ou ($CDM(p) = 2$ et $CDM(p') = 1$)).
Nous montrons un exemple de segmentation avec et sans consistance temporelle dans Fig.1(d) et 1(c). Dans Fig.1(b), nous représentons les différentes étiquettes (i.e. valeurs du CDM) avec des intensités différentes (noir pour $CDM(p) = 0$, rouge pour $CDM(p) = 2$ et blanc pour $CDM(p) = 1$). Dans la section 5, nous proposons le calcul



(a) (b)

(c) (d)

FIG. 1 – *(a) Segmentation de l'image $n-1$. (b) CDM calculé en utilisant la différence d'illumination entre l'image $n-1$ et l'image n et la segmentation de l'image n. (c) Segmentation de l'image n sans utiliser le CDM. (d) Segmentation de l'image n avec l'utilisation du CDM.*

d'une mesure objective de la consistance temporelle. Les résultats montrent une vraie amélioration de la consistance temporelle pour différentes séquences. En plus de cela, La manière dont nous exploitons le CDM réduit aussi les calculs de l'algorithme. Cette réduction des calculs est due au fait que les transitions dans les régions statiques ne sont pas retraitées, et que les transitions connectant les pixels non changeant sont traitées seulement par un algorithme CCL

4 Implémentation de l'algorithme

La figure 2 montre l'algorithme complet de la segmentation vidéo. A part la boucle de fusion de la segmentation spatiale, les autres fonctions de l'algorithme accèdent aux données dans un ordre connu (par exemple de gauche à droite, de haut en bas). Cette régularité d'accès aux données, limite le nombre de défauts de cache puisque La mémoire cache exploite la localité spatiale et temporelle des données. Contrairement à cela, dans la boucle de fusion, la structure de données UNION-FIND est non prédictible et cause donc un nombre de défauts cache important ce qui ralentit l'exécution de l'algorithme. Afin de limiter le nombre de défauts de cache, nous avons investigué deux optimisations qui sont indépendantes du type de la mémoire cache.

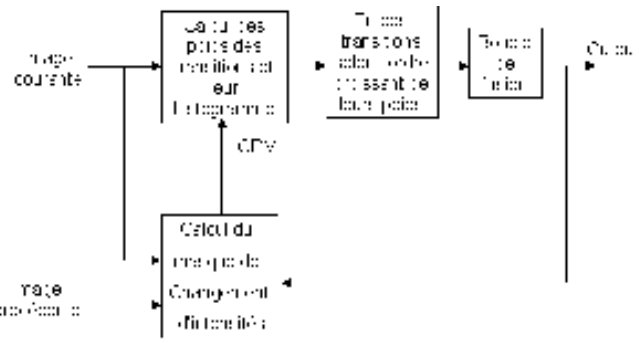


FIG. 2 – *L'algorithme complet de la segmentation vidéo*

La première optimisation concerne la façon dont on étiquette les régions pendant le processus de segmentation. Il y a deux méthodes pour gérer cela. Dans la première, on génère des nouveaux labels qui représentent les régions et donc optimiser l'utilisation des labels. Mais cette première méthode doit utiliser un autre tableau ou sera stocké la correspondance entre les labels générés et les pixels de notre image. Dans cette première méthode nous accédons à deux structures de données qui ont des organisations différentes. Dans la deuxième méthode, chaque pixel dans l'image peut être le représentant d'une région (on peut choisir celui dont l'adresse est la minimale). Nous avons vérifié que la deuxième méthode cause moins de défauts de cache. Donc les données sont stockées dans un tableau A de taille $|I|$, chaque élément $A(p)$ stocke la moyenne des couleurs $(\overline{Y}, \overline{U}, \overline{V})$ et le cardinal $|S|$. La deuxième optimisation Concerne la structure de données UNION-FIND. Elle est encodée séparément du tableau A cité précédemment grâce à un tableau F où $F(p)$ contient le père de p. La solution alternative consisterait à stocker les deux structures dans une seule, mais cela impliquerait le stockage de données inutiles dans le cache lors des opérations FIND.

Nous avons expérimenté ces optimisations sur le processeur TriMedia[15]. La mémoire cache de ce processeur est de taille 128 $KOctet$, 4-associative, avec des blocs de taille 128 $Octet$. L'algorithme de remplacement utilisé est LRU. Nous avons une réduction moyenne de 3 $Mcycles$ du temps d'exécution de l'algorithme. Ces optimisations sont utiles pour plusieurs segmentations qui utilisent la

structure de données UNION-FIND.
Il y a une quantité importante de parallélisme de données (DLP) dans notre algorithme (calcul des valeurs de transitions, calcul du masque de changement entre images). Ceci permet d'accélérer l'algorithme en traitant des données en parallèle si l'architecture cible dispose des ressources suffisantes. Le coeur du TriMedia est une architecture VLIW disposant de plusieurs unités fonctionnelles. Ces unités fonctionnelles peuvent traiter quatre octets en parallèle (mode SIMD). Le parallélisme d'instructions (ILP) est extrait par le compilateur, tandis que le parallélisme de données peut être exploité en utilisant des intrinsèques et le déroulage de boucle. Nous nous sommes servis de ces optimisations pour exploiter le (DLP) qu'offre notre algorithme.

5 Les résultats expérimentaux

Dans cette section, nous présentons des résultats expérimentaux de notre algorithme porté sur le processeur TriMedia et évalué pour des séquences CIF (images de taille 352*288). Le tableau 3 montre les résultats suivants :

1. Mesure de la consistance temporelle :
   Nous avons utilisé ici une mesure classique. Considérons la segmentation de l'image précédente $SEG(n-1)$ et la segmentation de l'image courante $SEG(n)$, nous trouvons la correspondance entre les régions dans $SEG(n-1)$ et celles dans $SEG(n)$. Deux régions ($S_{i,n-1} \in SEG(n-1)$ et $S_{j,n} \in SEG(n)$ correspondent si elles ont un maximum de recouvrement $Overlap(i,j) = |S_{i,n-1} \cap S_{j,n}|$. Nous additionnons le nombre de pixels des zones de recouvrement des régions qui correspondent dans toute l'image. La consistance temporelle est alors le pourcentage de ce nombre par rapport à la taille de l'image.
2. Temps d'execution :
   Nous donnons le nombre de Mega cycles que prend l'exécution de l'algorithme sur TriMedia. Comme nous l'avons déjà noté, l'utilisation du CDM réduit les calculs de l'algorithme. Le cas critique de l'algorithme (temps de calcul le plus long pour une image) correspond à l'exécution de la segmentation spatiale sans CDM. Ce cas critique correspond par exemple au traitement de la première image de la séquence où l'information CDM n'est pas disponible, ou au traitement d'une image après un changement de scène où l'information donnée par le CDM ne réduit pas les calculs. Pour cette raison, nous donnons ici les temps d'exécutions correspondant à ce cas critique. Si on intègre l'algorithme de consistante temporelle, le temps de calcul sera réduit.

Lorsqu'on utilise le CDM, la mesure de consistance temporelle est élevée. Visuellement cela se traduit par une segmentation en régions plus stable d'une image à l'autre. La

Séquence	Akiyo	Paris	Tennis Table	Mobile
Consistance temporelle (Sans CDM)	0.88	0.89	0.73	0.84
Consistance temporelle (Avec CDM)	0.98	0.97	0.92	0.92
Nombre de Meycles par image	15.68	16.59	15.84	16.20

FIG. 3 – *Les mesures expérimentales de notre algorithme*

segmentation reste cependant précise comme le montre la figure 4. A titre de comparaison, nous montrons aussi les résultats de segmentation obtenus avec l'algorithme JSEG [7] qui est maintenant une référence dans le domaine de la segmentation. L'algorithme JSEG procède d'abord à une quantification suivie d'une mesure de similarité à plusieurs échelles, ce qui augmente considéralement sa complexité et le nombre de paramètres à régler. Son approche multiéchelle explique les bons résultats obtenus dans les zones texturées (figure.4(b)). Par contre, notre algorithme donne de meilleurs résultats dans les zones homogènes (figure.4(e)) et il extrait également mieux les petites régions. Ce dernier point est important particulièrement lorsque l'application visée est l'amélioration d'image.
Au niveau de l'éxecution de notre algorithme, le processeur TriMedia utilisé dans notre expérimentation peut fonctionner avec une fréquence égale à $450\ MHz$. Une telle fréquence nous permet de traiter plus de 25 images par seconde, ce qui est suffisant pour une exécution temps réel.

6 Conclusion

La conception d'algorithmes pour le traitement des vidéos nécessite une exécution de ceux-ci en temps réel. Pour cette raison, nous avons choisi d'utiliser des méthodes ayant un temps de calcul raisonnable. Nous avons proposé une méthode de segmentation vidéo consistante temporellement. Notre méthode donne des résultats précis, et d'une grande consistance temporelle. Elle est basée sur une segmentation spatiale statistique et un masque de détection des changements d'illumination. Elle fonctionne en temps réel sur un seul processeur. Des résultats expérimentaux démontrent l'applicabilité de cette méthode. Nos recherches futures concernent la mise en oeuvre de l'algorithme en temps réel sur des séquences HD (Haute Définition), et l'exploitation des résultats de segmentation pour l'amélioration d'image.

Remerciements

Les auteurs remercient P. Meuwissen, O.P. Gangwal et Z. Chamski pour leurs suggestions constructives.

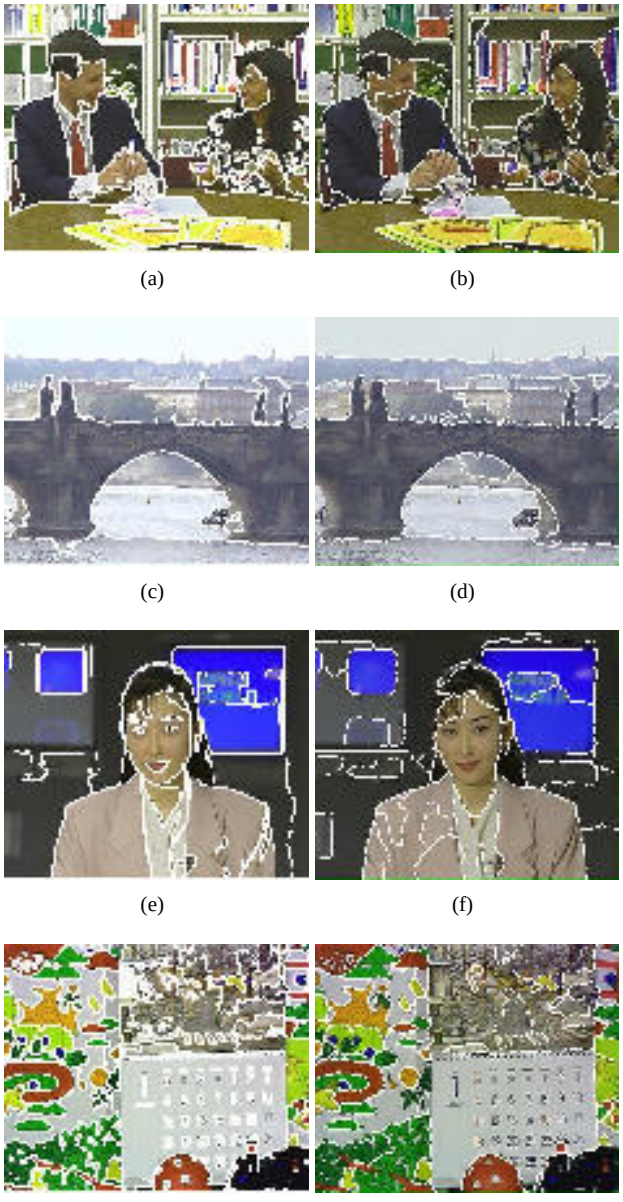


FIG. 4 – *Résultats de segmentation pour les séquences Paris, Bridge, Akiyo et Mobile. Figures a, c, e : Résultats avec notre segmentation. Figures b, d, f : Résultats avec l'algorithme JSEG*

Références

[1] G. Iannizzotto et L. Vita. Fast and accurate edge-based segmentation with no contour smoothing in 2-d real images. *IEEE Transactions on Image Processing*, 9, Issue 7 :1232 – 1237, 2000.

[2] R. Nock et F. Nielsen. Statistical region merging. *IEEE Transactions on PAMI*, 24, 2004.

[3] L. Vincent et P. Soille. Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *IEEE Transactions on PAMI*, 13, Issue 6 :583–598, 1991.

[4] Jianibo Shi et J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on PAMI*, 22, Issue 8 :888 – 905, 2000.

[5] E. Sharon, A. Brandt, et R. Basri. Fast multiscale image segmentation. *IEEE Conference on CVPR*, 1 :70 – 77, 2000.

[6] H.-Y. Wang et K.-K. Ma. Automatic video object segmentation via 3d structure tensor. *ICIP*, 1, 14-17 :153–156, 2003.

[7] Y. Deng et B.S. Manjunath. Unsupervised segmentation of colour-texture regions in images and video. *IEEE Transactions on PAMI*, 23, Issue 8 :800 – 810, 2001.

[8] D. Wang. Unsupervised video segmentation based on watersheds and temporal tracking. *IEEE Transactions on CSVT*, 8, ISSUE 5 :539 – 546, 1998.

[9] F. Moscheni, S. Bhattacharjee, et M. Kunt. Spatio-temporal segmentation based on region merging. *IEEE Transactions on PAMI*, 20, Issue 9, :897 – 915, 1998.

[10] P.F. Felzenszwalb et D.P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, Issue 2 :167–181, 2004.

[11] C.Connolly et T.Fliess. A study of efficiency and accuracy in the transformation from rgb to cielab color space. *IEEE Transactions on Image Processing*, 6(7) :1046 – 1048, 1997.

[12] M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, et B. Reed. Probabilistic methods for algorithmic discrete math. *Springer Verlag*, 20, Issue 9, :1 – 54, 1998.

[13] C. Fiorio et J. Gustedt. Two linear time union-find strategies for image processing. *Theoretical Computer Science*, 154 :165–181, 1996.

[14] K. Wu, E. Otoo, et A. Shoshani. Optimizing connected component labeling algorithms. *Proceedings of SPIE*, 5747 :1965–1976, 2005.

[15] pnx1500 databook. *available at http ://www.tcshelp.com/public_files.html*.

Génération automatique de code distribué à l'aide de RTOS : application au codage d'images LAR

Ghislain ROQUIER Mickaël RAULET Jean-François NEZAN Olivier DÉFORGES

IETR Groupe Image et Télédétection UMR CNRS 6164

INSA de Rennes

20, avenue des buttes de Coësmes, 35043 Rennes, FRANCE

{groquier, mraulet, jnezan, odeforges}@insa-rennes.fr

Résumé

Les futures générations de téléphones mobiles, incluant toujours plus de services multimédia, representent un vrai défi en terme de systèmes temps-réels embarqués de part leurs besoins toujours croissants en fléxibilité et en puissance de calcul. Les architectures multi-composants programmables peuvent alors apporter une solution efficace et évolutive. Le but de nos travaux consiste à développer un processus de développement rapide et automatique spécialement adapté aux architectures multi-composants hétérogènes. Cet article présente un processus de développement basé sur la méthodologie Adéquation Algorithme Architecture (AAA), de la description conjointe d'une application et d'une architecture jusqu'aux exécutifs distribués temps-réel. Nous montrerons ensuite une génération automatique d'exécutifs issus de la méthodologie AAA basée sur l'utilisation de systèmes d'exploitations temps-réel résident (Real-Time Operating System - RTOS). Nous comparons cette approche avec celle sans RTOS en termes de complexité et de performance. Finalement, ce travail est illustré par l'exécution d'une application multimédia basée sur le codec LAR.

Mots clefs

Prototypage rapide, temps-réel, système embarqué, codec LAR

1 Introduction

Les systèmes multimédias modernes requièrent une puissance de calcul toujours plus importante et donc des contraintes d'embarquabilité plus difficiles à satisfaire. D'autre part, leurs temps de développement doivent sans cesse être réduits. Dans ces systèmes, la limitation de la puissance de calcul est souvent palliée par l'utilisation de circuits spécifiques dédiés. Cependant cette solution est difficilement compatible avec un temps de développement court et ne peut pas être mise à jour efficacement. Une alternative peut être apportée par l'utilisation de composants logiciels (DSP, ARM) ou matériels (FPGA) puisqu'ils ont l'avantages d'être programmables et réutilisables. Néanmoins, les aspects parallèles et hétérogènes d'architectures multi-composants laissent apparaître de nouveaux problèmes en termes de distribution et d'ordonnancement des applications sur les différents composants. Une solution de conception appropriée consiste à utiliser une méthodologie de prototypage rapide permettant, à partir d'une description de l'application temps-réel de haut niveau, l'implantation optimisée et automatique sur l'architecture cible. La vocation de la méthodologie Adéquation Algorithm Architecture (AAA) présentée ici est de répondre à ces besoins. Le but de la méthodologie AAA est de générer automatiquement des exécutifs distribués temps-réel à partir des descriptions respectives de l'application et de la cible matérielle. Les exécutifs sont ordonnancés hors-ligne et sont alors particulièrement bien adaptés aux systèmes déterministes, comme par exemple les algorithmes de traitement des images, ainsi qu'aux architectures multi-composants hétérogènes.

L'ordonnancement hors-ligne rend superflu l'utilisation d'un RTOS. En effet, un ordonnancement en-ligne implique plus de données sur le composant. Il est bien adapté lorsque le comportement de l'application ne peut être prédit, comme lors de traitements sur des événements apériodiques [1]. Lorsque le comportement de l'application est parfaitement déterministe, un ordonnancement hors-ligne est suffisant et peut être implémenté par un simple séquenceur de calcul. Un RTOS utilise des ressources trop souvent limitées dans un système embarqué. Néanmoins, une comparaison entre ces deux types d'implantations doit être effectuée pour déterminer l'impact sur la mémoire allouée, l'effet sur les communications entre composants ainsi que sur le comportement temps-réel de l'application. Cet article est organisé comme suit : le 2^{e} chapitre introduit la méthodologie AAA. Les spécifications des exécutifs et l'utilisation de RTOS selon la méthodologie AAA sont décrites dans le 3^{e} chapitre. L'implantation d'une application basée sur le codec vidéo LAR et une discussion sur les résultats sont donnés dans le 4^{e} chapitre. Enfin les conclusions seront détaillées dans le 5^{e} chapitre.

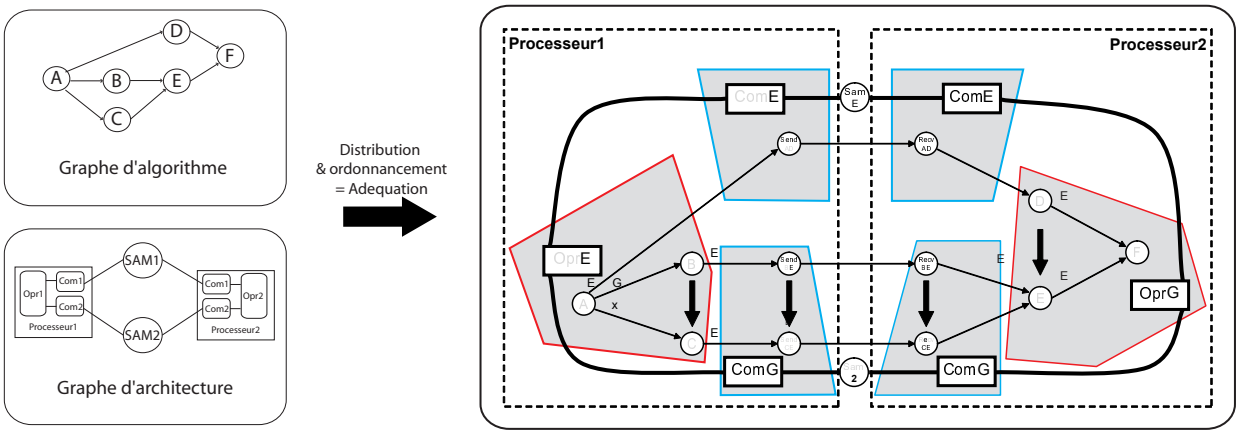


FIG. 1 – *Placement et ordonnancement d'un algorithme sur une architecture*

2 Méthodologie AAA

Le but de la méthodologie AAA consiste à trouver le meilleur placement et ordonnancement d'un algorithme sur une architecture multi-composant La méthodologie AAA trouve son fondement dans la théorie des graphes. L'algorithme et l'architecture sont décrits par deux graphes distincts qui révélent le parallélisme potentiel de l'algorithme et le parallélisme disponible de l'architecture. L'adéquation est une suite de transformations effectuées sur ces deux graphes qui aboutit à une implantation optimisée de l'algorithme au sens de la latence.

2.1 Modèles d'algorithme et d'architecture

L'algorithme de l'application est modélisé par un graphe flot de données (GFD) qui est un hyper-graphe orienté. Chaque sommet et chaque arête représentent respectivement une opération de l'algorithme et un transfert de données entre opérations. Un GFD révèle un ordre partiel pour l'exécution des opérations : deux opérations sans relation de dépendance de donnée peuvent être exécutées dans un ordre arbitraire et plus particulièrement, elles peuvent être exécutées simultanément par deux processeurs distincts. Le GFD permet donc d'exhiber le parallélisme potentiel d'un algorithme [2].

Dans la méthodologie AAA, afin d'être précis dans la description sans être trop complexe au niveau matériel, la machine à états finies est définie comme le composant atomique de l'architecture. Ainsi un processeur ou un circuit dédié peut être vu comme une composition de machines à états finies. Une architecture multi-composant est alors representée par un réseau d'automates finis interconnectés à l'aide de media de communication (bus, mémoires partagés ...). Une architecture peut être représentée par un graphe non-orienté où chaque sommet et chaque arête sont respectivement un processeur et un media de communication. Dans ce modèle, un processeur est composé d'un opérateur et autant de communicateurs que de media connectés. Un opérateur exécute une partie de l'algorithme et un communicateur execute un opération de communication lorsqu'un transfert de données est requis. L'opérateur et les différents communicateurs sont reliés entre eux à travers une mémoire partagée du processeur.

La figure 1 représente un graphe d'architecture composé de deux processeurs connectés via deux media. Chaque processeur est constitué d'un operateur et de deux communicateurs.

2.2 Transformations de graphe

Le graphe d'implantation est obtenu par transformation du graphe d'algorithme et du graphe d'architecture. Cette transformation correspond à la distribution et à l'ordonnancement de l'algorithme. La distribution, également appelée partitionnement, alloue spatialement les différentes opérations du graphe d'algorithme sur les opérateurs du graphe d'architecture. Cela revient à diviser le graphe d'algorithme en plusieurs sous-graphes decrivant les opérations que chaque opérateur doit exécuter. L'ordonnancement utilise les dépendances de données du GFD pour allouer dans l'espace temporel les opérations sur les opérateurs. Cette transformation revient à définir la séquence d'exécution des opérations sur un opérateur comme le montre la figure 1. Les parties grisées représentent les différents opérateurs et communicateurs. Les dépendances de données définissent des précedences : ici, *A* est exécutée avant *B*. Lorsque les opérations n'ont pas de dépendances de données, des précédences sont insérées pour éviter les interblocages. Elles sont représentées par les flèches en gras. Cela implique dans notre exemple que *B* doit être exécutée avant *C*.

Le graphe d'implantation est obtenu par une optimisation simultanée de la distribution et de l'ordonnancement. Un grand nombre d'implantations est envisageable, le problème d'optimisation consiste à déterminer l'implantation la plus efficace en terme de respect des contraintes temps-réel. L'optimisation de la distribution et d'ordonnancement sur une architecture multi-composant est un problème NP-difficile, *i.e.* une recherche exhaustive de toutes les solutions possibles est inconcevable. Une heuris-

tique est donc utilisée pour trouver une approximation de la solution optimale dans un temps raisonnable. Cette heuristique [3] de type gloutonne vise à minimiser la latence de l'algorithme exécuté sur une architecture multi-composant.

2.3 Génération d'exécutif

Une fois le graphe d'implantation optimisée déterminé, un exécutif peu être automatiquement généré pour chaque opérateur. Avant cela, quelques modifications doivent être apportées à ce graphe. Tout d'abord, l'aspect répétitif de l'application ainsi que les synchronisations entre opérateurs doivent être ajoutés dans le graphe d'implantation optimisé. En effet, les applications réactives à implanter sont itératives par nature alors que le GFD ne permet pas de faire ressortir le caractère répétitif d'une application. Des boucles sont alors insérées dans chaque séquence d'opérations et de communications. L'ordonnancement ne fait pas non plus apparaître les synchronisations entre les différents séquenceurs d'un processeur. Des sémaphores sont donc insérés afin de garantir la précedence entre opérations de calcul et de communication sur un même processeur [4]. Une description plus détaillée est donnée dans [2]. Le réseau de Petri de la figure 2 représente les synchronisations entre les différents séquenceurs du premier processeur de notre précedent exemple où *P* et *V* [1] respectivement attend un sémaphore et envoie un sémaphore. Les paramètres d'un sémaphore définissent les opérations à synchroniser, le chemin vers le communicateur (A_1 par exemple), l'état du tampon (plein ou vide) et enfin le nom du media utilisé.

Dans cet exemple, l'opération *A* est exécutée et le résultat de ce calcul stocké dans le tampon mémoire *AD* pour être envoyé sur le communicateur *Com1*, comme défini par l'ordonnancement. Un sémaphore $\{A_1$,*1*,*SAM1*$\}$ est alors envoyé au communicateur de telle sorte que l'opération de communication *send(AD)* de *AD* par *Com1* ne soit pas réalisé avant la fin de l'opération *A*. L'opération *send(AD)* envoie le contenu du tampon vers *processor2*. Un semaphore $\{A_1$,*0*,*SAM1*$\}$ est envoyé vers *OPR1* en fin de transfert pour que celui ci n'accède pas aux données avant qu'elles n'aient été envoyées. Si la séquence de communication est plus rapide, *Com1* devra alors attendre de nouveau que les données *AD* soient calculées par *A* sur *OPR1*. Notons que ces synchronisations sont générées automatiquement dans AAA.

Une fois le graphe d'exécution déterminé, il doit ensuite être transformé en autant de macro-exécutifs que de processeurs présents sur l'architecture. Un exécutif générique est composé d'une liste de macro-instructions qui permettent de spécifier les allocations mémoires, les synchronisations, les séquences de communications ainsi que les séquences de calculs. Ces macro-instructions sont dites génériques, c'est-à-dire qu'elles ne dépendent d'aucun langage spécifique de programmation. Cela permet de rester à un haut niveau d'abstraction sans se soucier de la cible matérielle envisagée. Une ultime transformation, détaillée section 3, permettra la création d'exécutifs dans le langage spécifique à la cible matérielle (C ou assembleur pour DSP et GPP, VHDL pour FPGA).

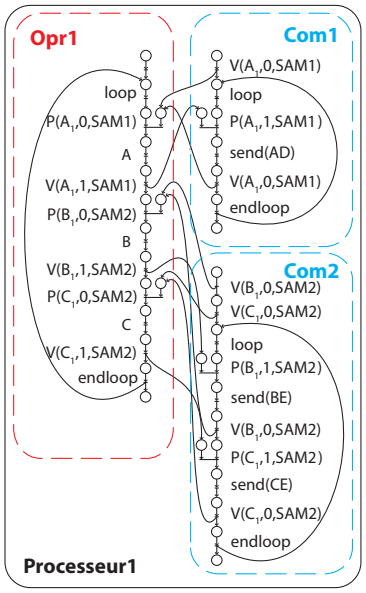


FIG. 2 – *Réseau de Pétri du graphe d'exécution*

2.4 SynDEx

Le logiciel SynDEx[2] est un outil de CAO niveau système librement téléchargeable. Il est principalement développé à l'INRIA Rocquencourt avec notre participation. Cet outil supporte la méthodologie AAA pour le prototypage rapide et l'implantation optimisée d'applications temps-réel distribuées sur architectures multi-composants. SynDEx permet de générer des exécutifs génériques qui respectent la méthodologie AAA.

3 Spécification des exécutifs

Les exécutifs génériques créés selon le méthodologie AAA doivent être transformés afin d'obtenir des exécutifs compilables qui pourront être ensuite chargés sur les éléments de la cible matérielle.

3.1 Traduction

Chaque exécutif générique est traduit en un exécutif compilable à l'aide d'un macro-processeur. Le macro-processeur transforme la liste de macro-instructions en un code source propre à la cible matérielle envisagée. Cela consiste à remplacer chaque macro-instruction par une définition. Chaque définition étant spécifiée à travers des bibliothèques (aussi appelée noyaux) dépendantes du processeur cible ou encore du media de communication reliant deux processeurs. Plusieurs types de bibliothèques

[1] ***P**robeer* et ***V**erhoog* signifient *décrémente* and *incrémente* en Néerlandais

[2] **Syn**chronized **D**istributed **Ex**ecutive

existent, elle permettent de donner des définitions propre à l'architecture comme les allocations mémoire, les synchronisations entre séquences ou encore les transferts de communications. D'autres bibliothèques sont plus proches de l'algorithme et permettent de traduire par exemple les prototypes ou les appels de fonctions. Puisque les macro-exécutifs sont génériques, il existe un grand nombre de traductions possibles amenant à un code source compilable. Cette phase est réservée à l'utilisateur qui doit choisir la traduction la plus appropriée à la cible envisagée. Le logiciel libre GNU-M4 est le macro-processeur que nous utilisons pour la phase de traduction. Il faut remarquer que dans tous les cas (avec ou sans RTOS), l'exécutif temps-réel distribué est statique et défini hors-ligne. Le comportement temps-réel et les synchronisations sans interblocages entre les différentes séquences sont garantis par construction.

3.2 Real-Time Operating Systems dans le contexte AAA

L'approche présentée dans cet article consiste à comparer deux types de traductions. Dans un premier temps, une traduction "classique" dans le sens de AAA est réalisée [5]. Cette traduction consiste à traduire sans l'aide d'un RTOS l'exécutif générique dans le langage de programmation approprié. Dans un deuxième temps, un autre traduction est réalisée mettant en jeux un RTOS. Cette approche consiste lors de la traduction à configurer un RTOS résident capable d'exécuter les différentes séquences et de gérer les synchronisations entre elles. A cette fin, de nouvelles bibliothèques de traduction ont été developpée. La figure 3 represente les deux approches proposées, et ce de la description haut niveau de notre application jusqu'aux basses couches matérielles.

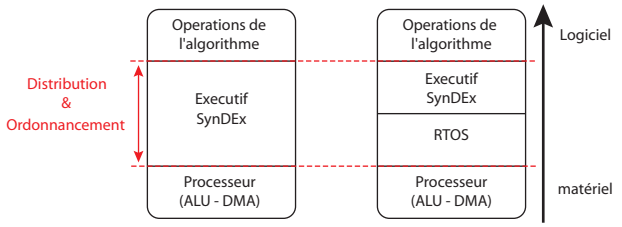


FIG. 3 – *Du logiciel jusqu'au matériel : deux approches*

Un RTOS est un système d'exploitation adapté aux applications temps-réel. Il permet de définir et de contrôler plusieurs tâches au sein d'un même processus. Des mécanismes de communication et de synchronisation entre tâches existent à cet effet. Dans cette approche, il faut considérer les différentes séquences de calcul et communications obtenues par la méthodologie AAA comme différentes tâches distincts du processus. Dans l'approche "classique", les sémaphores sont représentés par des booléens gérés "manuellement" par l'utilisateurs, alors que dans la deuxième approche, les sémaphores sont gérés par le RTOS pour synchroniser les séquenceurs. Il faut remarquer que le RTOS est uniquement utilisé pour la gestion des différentes tâches ainsi que pour le contrôle des synchronisations. Cependant, il faut aussi noter que l'intégration d'un RTOS dans un cible matérielle amène un surcoût. Surcoût en mémoire et en temps d'exécution lié à la gestion de contexte par le RTOS.

4 Travaux réalisés

4.1 Aperçu des RTOS

Un grand nombre de RTOS existent pour différents types de processeurs. Leurs primitives sont souvent spécifiques à une famille particulière de processeurs. *A contrario*, un RTOS plus générique et indépendant d'une famille de processeurs semble mieux approprié pour une implantation rapide sur un matériel varié. Le Linux embarqué semble avoir cet atout. En effet, ce RTOS a l'avantage d'être conforme à la norme POSIX, ce qui rend la programmation de tel RTOS plus aisée et independante de la cible. Pour les DSP de chez TI plusieurs RTOS Linux embarqué existent tels que MediaLinux OS [6], Lightweight OS [6]. Pour le moment, DSP-BIOS, le RTOS propriétaire de TI, est le RTOS que nous utilisons pour les applications que nous developpons. Il faut remarquer que DSP-BIOS n'est utilisable que sur les DSP de TI et que ses primitives sont spécifiques à ce RTOS.

4.2 Plateforme cibles

Plusieurs fournisseurs de matériel tels que Pentek, Sundance ou Vitec MM developpent des architectures multi-composant et quelques unes d'entre elles ont été validées pour supporter la méthodologie AAA. Pour le moment, une plateforme Sundance a été utilisée pour accueillir le RTOS résident. Ces plateformes sont composées d'un PC et d'une carte PCI. Cette carte peut être composée de différents modules interconnectés par différents media. Le module SMT361 est constitué de DSP C6416 bien adaptés pour le traitement des images et le module SMT319 est constitué d'un DSP C6414 connecter à des circuits intégrés permettant la conversion numérique-analogique pour un affichage ou une acquisition de l'image au format PAL. La plateforme utilisée est représentée sur la figure 6 composée de deux modules SMT361 et d'un module SMT319.

4.3 Résultats préliminaires

Afin de déterminer la différence entre les deux approches, une application de communication entre deux DSP a été testé. Les résultats sont donnés sur les figures 4 et 5. Le 1er graphe montre le temps nécessaire selon les deux approches pour exécuter l'application en fonction de la taille des données. Le 2^{e} graphes nous donne le rapport entre les temps d'exécution des deux approches. Le processus utilisant le RTOS est toujours plus lent que celui ne l'utilisant pas. Cependant, plus la taille des données est grande, plus cet impact temporel est réduit. Ainsi, le temps d'exécution de l'application ne semble pas excessivement augmenté par l'utilisation d'un RTOS lorsque les données

sont grandes comme par exemple en traitement des images. En plus de cela, le code source généré est moins grand et plus compréhensible comparé à l'autre méthode. Le debogage durant la phase de vérification fonctionnelle de l'algorithme en est donc facilité.

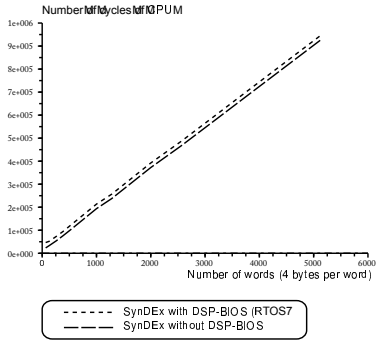


FIG. 4 – *Comparaison des temps d'exécution*

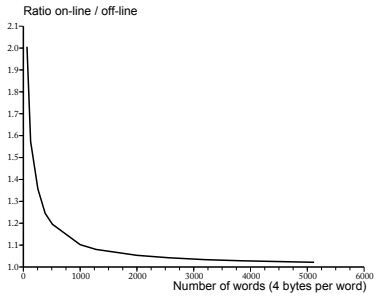


FIG. 5 – *Rapport entre les temps d'exécution des deux approches*

4.4 Implantation du codec LAR

Le LAR[3] est un algorithme de compression et de décompression vidéo developpé dans notre laboratoire [7], bien adapté pour la transmission d'image. Le principe de cette technique est d'adapter la résolution locale (tailles des pixels) selon l'uniformité de la luminance, typiquement une basse résolution (bloc de 16×16 pixels) pour une zone où la luminance est uniforme et au contraire une forte résolution (bloc de 2×2 pixels) pour une zone présentant de forte singularité. C'est un codec vidéo scalable permettant aussi bien un codage en niveaux de gris très bas débit qu'une compression de vidéo couleur sans perte.
L'objectif est de réaliser une implantation de ce codec sur notre architecture multi-composant selon la méthodologie AAA. Le codeur et le décodeur sont respectivement implantés sur le 1er et le 2^{e} DSP de la plateforme. Le GFD du codec LAR est représenté sur la figure 7. Trois différents algorithmes scalables du LAR sont implantés afin de comparer les résultats lorsque la compléxité est croissante [8].

Algorithme 1 : codec vidéo spatial pour la luminance (caractérisé par des blocs 2×2, 4×4 et 8×8).

[3] **L**ocally **A**daptive **R**esolution

algorithme	sans RTOS	avec RTOS
1	18.03 ms	18.05 ms
2	25.35 ms	25.45 ms
3	31.84 ms	32.07 ms

TAB. 1 – *Temps d'exécution des algorithmes LAR*

algorithme	Codeur		Decodeur	
	sans RTOS	RTOS	sans RTOS	RTOS
1	874 kB	928 kB	899 kB	953 kB
2	747 kB	802 kB	658 kB	713 kB
3	642 kB	697 kB	528 kB	583 kB

TAB. 2 – *Mémoire utilisée par le codec*

Algorithme 2 : codec vidéo spatial pour la chrominance ajouté à l'algorithme 1.

Algorithme 3 : codec vidéo spectral pour les blocs 2×2 (caractérisé par l'ajout de l'erreur résiduelle) ajouté à l'algorithme 2.

Les résultats obtenus de l'implantation des différents algorithmes est conforme aux résultats obtenus précédemment. Le temps d'exécution du codec est légèrement plus lent avec l'utilisation d'un RTOS (cf TAB. 1). Le codeur est la partie de l'algorithme la plus lente (typiquement 4 fois plus lent que le décodeur), c'est pourquoi le comportement temps-réel du codec LAR est donné par le temps d'exécution du codeur puisque l'exécution totale est "pipeliné" par les processeurs. L'utilisation du RTOS a aussi un impact sur la mémoire des processeurs (55 kO). Cependant, plus la mémoire est utilisée, plus l'impact devient proportionnellement petit . Pour l'algorithme 3, DSP-BIOS augmente la mémoire utilisée par le codeur et par le décodeur de respectivement 7% et 5% (cf TAB. 2).

5 Conclusion et perspective

Cet article a permis de présenter l'intégration d'un système d'exploitation temps-réel lors de la génération automatique d'exécutifs de la méthodologie AAA. La méthodologie a été introduite afin de présenter la génération automatique d'exécutifs distribués temps-réel. Le developpement de bibliothèques lors de la spécification des exécutifs nous a permis d'utiliser un RTOS et de l'implanter dans l'architecture cible PC-multi-DSP.
Bien que le RTOS ait un impact autant sur la mémoire allouée que sur le temps d'exécution d'un processus, nous avons constaté que ce surcoût devenait faible lorsque la taille des données traitée augmentait. Ainsi, l'utilisation d'un RTOS semble presque aussi efficace que l'approche utilisée auparavant pour le traitement des images où les données sont souvent grandes. D'ailleurs, les exécutifs générés avec des primitives du RTOS est plus simple et permet donc une meilleure compréhension pour l'utilisateur. Ce point est très important pour que notre génération automatique soit toujours utilisable dans le futur. L'implantation du codec vidéo LAR avec l'utilisation d'un

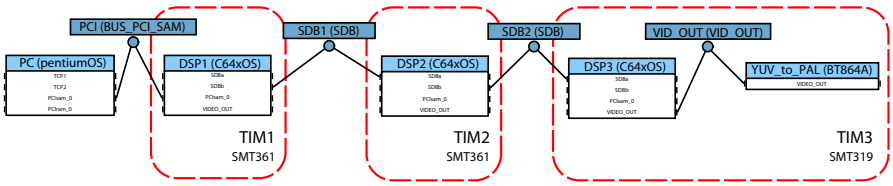


FIG. 6 – *Platforme cible*

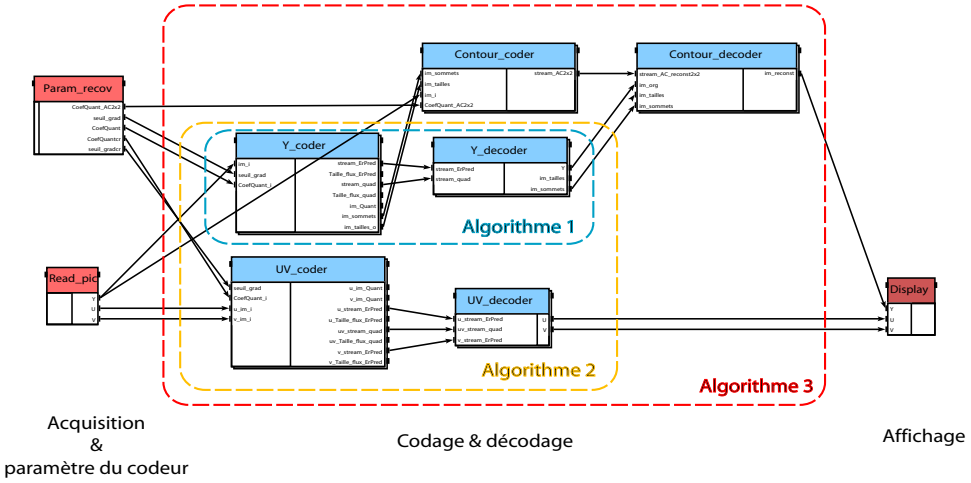


FIG. 7 – *description GFD du codec LAR*

RTOS résident nous a permis de vérifier le comportement temps-réel de l'algorithme nécessaire dans le contexte de systèmes embarqués.

Nous travaillons quant à l'intégration d'un RTOS plus générique, tels que ceux utilisant la norme POSIX, et indépendant du processeur cible. Cela permettra un développement plus rapide de processus multitâche sur des architectures variées à l'aide de la même spécification du RTOS. Ce travail permettra l'implantation de nouveaux algorithmes de traitement d'image en développement dans notre laboratoire tels que le standard MPEG-4 AVC ainsi que le standard MPEG-21 SVC.

Références

[1] F. Balarin and *al*. Scheduling for embedded real-time systems. *IEEE Design and Test of Computers*, 15(1) :71–82, Janvier-Mars 1998.

[2] T. Grandpierre and Y. Sorel. From algorithm and architecture specifications to automatic generation of distributed real-time executives : a seamless flow of graphs transformations. In *First ACM and IEEE International Conference on Formal Methods and Models for Co-Design, Mont Saint-Michel, France*, Juin 2003.

[3] T. Grandpierre, C. Lavarenne, and Y. Sorel. Optimized rapid prototyping for real time embedded heterogeneous multiprocessors. In *proc. of IEEE CODES'99 7th Int. Workshop on Hardware/Software Co-Design, Rome, Italie*, Mai 1999.

[4] Edsger W. Dijkstra. Cooperating sequential processes. In F. Genuys, editor, *Programming Languages : NATO Advanced Study Institute*, pages 43–112. Academic Press, 1968.

[5] M. Raulet, F. Urban, J-F. Nezan, O. Déforges, and C. Moy. Syndex executive kernels for fast developments of applications over heterogeneous architectures. In *EUSIPCO'05, Antalaya, Turquie*, Septembre 2005.

[6] J. Kretzschmar and R.Baumgartl. Lightweight rtai for dsps. In *1st Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT), Palma de Mallorca, Espagne*, Juin 2005.

[7] O. Déforges and J. Ronsin. Region of interest coding for low bit-rate image transmission. *IEEE International Conference on Multimedia and Expos (ICME), New-York, USA*, Août 2000.

[8] M. Raulet, F. Urban, M. Babel, O. Déforges, J-F. Nezan, and Y. Sorel. Automatic coarse-grain partitioning and automatic code generation for heterogeneous architectures. In *IEEE Workshop on Signal Processing Systems (SIPS'03), Seoul, Corée*, Septembre 2003.

Extraction de contours multirésolution pour un codage d'images par bandelettes

G. Jeannic[1] V. Ricordel[1] D. Barba[1]

[1] IRCCyN / Équipe Image et VidéoCommunication

École Polytechnique de l'Université de Nantes
La Chantrerie, rue Christian Pauc, BP 50609, 44306 Nantes Cedex 3

{guillaume.jeannic, vincent.ricordel, dominique.barba}@univ-nantes.fr

Résumé

Depuis quelques années, un certain nombre de transformées orientées a été développé pour améliorer les inconvénients de la transformée en ondelettes bidimensionnelle conventionnelle. Parmi elles, la transformée en bandelettes s'adapte au contenu de l'image pour opérer un filtrage de type ondelettes le long des contours et, exactement, exploiter la régularité dans la direction de ces contours autour d'une bande étroite. Cette transformée requiert une extraction de contours adaptée. Nous proposons donc une méthode d'extraction de contours répondant à ces contraintes. Notre approche produit, à différents niveaux de résolution une carte des contours extraits.

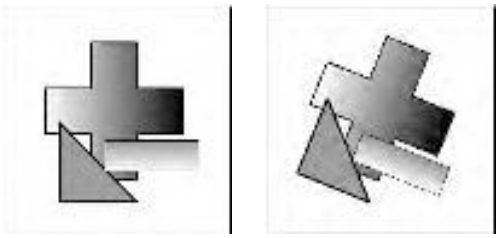
Mots clefs

transformée en ondelettes, transformée en bandelettes, extraction de contours

1 Introduction

La transformée en ondelettes bidimensionnelle utilisée actuellement pour la compression d'image fixe dans la norme JPEG2000 résulte du produit tensoriel de deux transformées en ondelettes unidimensionnelle appliquées suivant les lignes et les colonnes de l'image. Ces ondelettes bidimensionnelles séparables ont prouvé leur capacité à détecter les singularités horizontales, verticales, ou ponctuelles (assimilées à des singularités diagonales). Toutefois, l'analyse de contours qui ne sont pas parfaitement horizontaux, verticaux ou diagonaux conduit à une représentation suboptimale de l'information : les coefficients d'ondelettes représentant de tels contours se retrouvent éparpillés dans les différentes sous-bandes au lieu d'être fortement présents dans une unique sous-bande. Dans un contexte de compression avec perte, la transformée en ondelettes couplée à des techniques de quantification classiques entraîne plus d'artéfacts visuels sur les contours d'orientations quelconques que sur les contours purement horizontaux, verticaux ou diagonaux (voir figure 1).

D'autre part, si la transformée en ondelettes est par exemple à même de détecter une singularité le long d'une colonne, dans le cas d'un contour horizontal, une singularité est détectée pour chaque élément de ce contour. Les ondelettes n'exploitent donc pas complètement la régularité d'un contour pour le représenter.



(a) (b)

Figure 1 – *À taux de compression égal avec JPEG2000, ici d'un ratio* 1 : 31*, on constate moins de défauts visuels au niveau des contours horizontaux, verticaux et diagonaux (a) que sur les contours orientés différemment (b).*

Depuis quelques années, de nouvelles représentations d'images ont été développées de façon à pallier à ces défauts de la transformée en ondelettes tout en conservant ses avantages, à savoir sa multirésolution, sa bonne localisation en espace et en fréquence, ainsi que sa décimation critique. On peut ainsi trouver deux approches différentes dans la littérature : les transformées qui, sur le même modèle que la transformée en ondelettes, prennent en compte plus d'orientations que les simples horizontales et verticales, et d'autres transformées qui elles adaptent la transformée en ondelettes conventionnelle au contenu de l'image. Parmi ces transformées adaptatives, la transformée en bandelettes requiert une information sur le contenu de l'image, à savoir ses contours. Un schéma d'extraction adapté sera présenté ainsi que les résultats obtenus.

1.1 Transformées à bases fixes

Parmi les transformées où la base de projection est fixe, la transformée en "ridgelettes" [1] décompose une image en contours rectilignes traversant toute l'image. L'idée est de pouvoir représenter un tel long contour avec un unique coefficient en rapport avec son amplitude et situé dans une sous-bande caractérisant son épaisseur. En pratique une

telle décomposition est obtenue par transformée en ondelettes unidimensionnelle sur différentes projections de Radon de l'image. À part dans des conditions très restrictives [2], cette transformation introduit de la redondance, ce qui n'est pas souhaité dans un contexte de codage de source. D'autre part, les contours dans une image sont rarement rectilignes et aussi longs que les dimensions même de l'image. Pour résoudre le second point, la transformation est appliquée par blocs avec les mêmes problématiques d'effets de blocs que la transformée en cosinus discrète. L'idée est reprise par la transformée en "curvelets" [3] qui introduit une analyse multi-échelle en appliquant une transformée en "ridgelets" par blocs après une décomposition en sous-bandes. Ces deux transformées nécessitent l'implémentation d'un opérateur de rotation ce qui n'est pas simple dans le domaine discret. À cette fin a été développée la transformation "contourlets" [4] à base de filtres non séparables. Elle propose une décomposition multi-échelle non directionnelle qui s'apparente de la décomposition en curvelettes. Outre la redondance introduite par cette transformée, la forte taille des filtres employés introduit des artéfacts visuels.

1.2 Transformées à bases adaptatives

La transformée en ondelettes bidimensionnelle peut être implémentée selon un schéma de "lifting". La technique est appliquée en utilisant des échantillons de l'image le long des lignes (respectivement des colonnes). La seconde approche que l'on retrouve dans la littérature utilise cette technique mais en utilisant des échantillons qui ne sont pas tous situés sur une même ligne (respectivement une même colonne). Cela revient à appliquer une transformée en ondelettes le long d'une direction particulière. Le choix d'une telle direction est délicat. Dans la transformée en "curved wavelet" [5], une transformée en ondelettes est appliquée selon cinq orientations différentes pour chaque bloc d'une image. Une direction est ensuite retenue pour chaque bloc, celle minimisant l'énergie contenue dans la sous-bande des hautes fréquences. Le choix de la direction est alors implicite et peut correspondre à l'orientation globale des contours de tout un bloc. Dans la transformée en bandelettes [6], le choix de la direction est la conséquence d'une extraction de contours dans l'image. Pour chaque une bande autour d'un contour donné est réarrangée par translation selon les lignes ou les colonnes. Une transformée en ondelettes unidimensionnelles est ensuite appliquée sur cette bande réarrangée. Notre bande est donc filtrée selon deux directions : la direction du contour qui est caractérisée par une certaine régularité, et la direction orthogonale au contour redressé qui est caractérisée par une forte singularité. Cette régularité dans la direction du contour est ensuite exploitée par application d'une transformée en ondelettes unidimensionnelle sur les lignes (respectivement colonnes) des détails horizontaux respectivement verticaux) pour un contour réaligné sur l'horizontale respectivement la verticale, voir aussi la figure 2.

Ces deux transformées induisent un supplément de codage que ce soit pour indiquer le choix d'orientation pour chaque bloc de l'image, ou pour représenter les contours dans l'image.

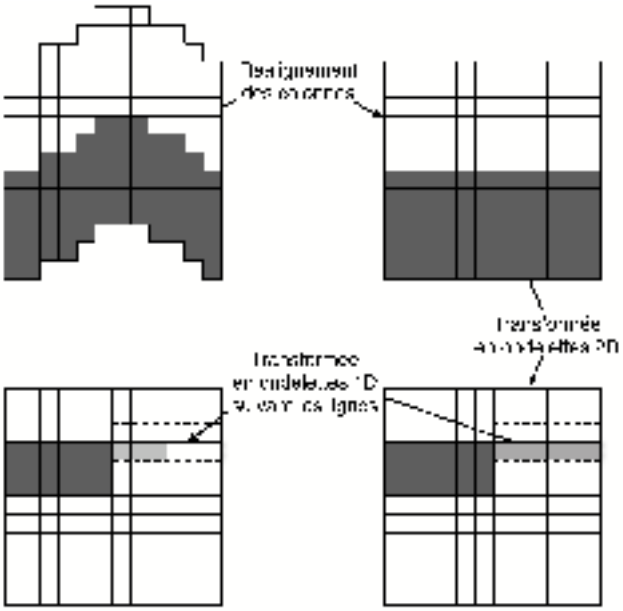


Figure 2 – *Exemple de décomposition en bandelettes d'un contour réaligné sur l'horizontale.*

2 Une extraction de contour adaptée aux bandelettes

Notre attention se porte sur les transformations adaptatives qui n'introduisent pas de redondance mais un supplément de codage et dont l'implémentation pour se faire selon un schéma de lifting comme la transformée en ondelettes bidimensionnelle conventionnelle. L'adaptation de la transformée en "curved wavelet" au contour de l'image dépend de la taille des blocs utilisés. La transformée en bandelettes permet une adaptation plus flexible pour filtrer des contours quelconques, mais ne permet toutefois pas de traiter des zones texturées orientées. On peut dès lors imaginer filtrer les bandes autour des contours forts d'une images par la transformée en bandelettes et utiliser la transformée en ondelettes courbées sur les autres zones, celles appartenant au "fond", qui peuvent contenir des textures orientées.

La transformée en bandelettes requiert l'extraction des contours de l'image. Ces contours guident le filtrage de l'image, ils doivent donc être transmis au décodeur. Pour ce faire Le Pennec [6], les représente en des courbes splines. Dans un contexte de codage, il est plus intéressant d'extraire des contours longs, plutôt que des contours morcelés. Il serait simple d'éliminer les contours de longueurs insuffisantes par un post-traitement après extraction, mais il est encore plus intéressant d'arriver à représenter plusieurs segments de contours en les reliant en un seul et

unique contour, et donc une seule et unique spline.
Le Pennec propose dans sa thèse une extraction *ad hoc* de contour basée sur une extraction de points de contours forts à différents niveaux de résolution. Les points sont reliés entre eux selon des contraintes basées entre autre sur l'orientation du gradient en ces points, de telle sorte qu'une continuité existe suivant les orientations. Dans le cas où un point a plusieurs successeurs possibles, la liaison entre ces points est rompue. Ceci afin de ne pas créer de coins et ainsi éviter la superposition de bandes.
Notre schéma d'extraction de contours fait aussi intervenir une analyse multi-échelle afin de pouvoir extraire des contours plus longs et être moins sensible au bruit. Notre approche permet en plus de produire pour chaque niveau de résolution une carte complète des contours extraits. Une transformation en bandelettes peut être donc réalisée à chacun des niveaux. Enfin nous souhaitons gérer la détection des noeuds et des coins dans l'image afin de pouvoir les traiter de façon adaptée et non comme le serait le fond de l'image.

3 Extraction de contours multirésolution

Le schéma que nous proposons conbine de trois mécanismes (voir figure 3) : la construction d'une pyramide multi-échelle de l'image [7], la prolongation de contours par abaissement progressif du seuil du gradient, et l'affinement en position de contours obtenus à résolution inférieure par la méthode des contours actifs. Ce schéma multirésolution permet de trouver le compromis entre contours suffisamment longs et suffisamment précis. L'affinement en position de contours permet de conserver les structures détectées à basse résolution. Des contours qui auraient été détectés de façon morcelés à cause de chutes locales du gradient ou à cause du bruit, conservent grâce à cette étape leur unité. La prolongation de contours par abaissement progressif du seuil du gradient permet à une résolution donnée de faire apparaître de nouveaux germes ou de prolonger des contours déjà détectés à plus basse résolution en s'appuyant sur les points de contours forts.

3.1 Prolongation de contours à une résolution donnée

Un simple détecteur des extrema locaux du gradient produit un ensemble de points correspondant aux contours de l'image mais aussi aux textures et au bruit. Nous voulons donc retenir et relier entre eux, uniquement les points appartenant aux contours forts de l'image tout en restant robuste au bruit (voir figure 4).
Notre approche est itérative et vise à chaque boucle, à prolonger les extrémités des contours déjà formés ou à faire émerger de nouveaux germes. Exactement pour une itération donnée, le seuil du gradient est abaissé faisant apparaître de nouveaux points. Aux extrémités des contours déjà formés, apparaissent d'éventuels points successeurs qui sont reliés s'ils minimisent une fonction de coût. Cette

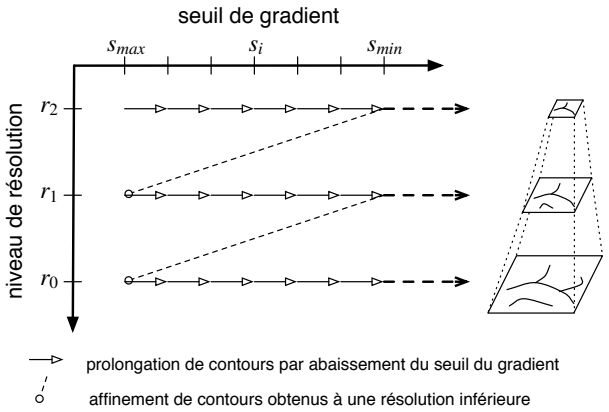


Figure 3 – *Schéma d'extraction multirésolution des contours.*

étape d'agrégation de points au contour est poursuivi tant que des successeurs sont trouvés. Lorsque l'on ne peut plus prolonger de contour, le seuil du niveau de gradient est abaissé et on réitère le processus.
Si l'on considère un contour à prolonger d'extrémité P_e, la fonction de coût à minimiser :

$$\alpha E_{continuité} + \beta E_{orientation} + \gamma E_{image} \qquad (1)$$

fait intervenir trois termes pondérés par des paramètres α, β et γ :

$E_{continuité}$ **:** une contrainte de continuité qui tend à prolonger la chaîne dans la direction donnée par P_e et son prédecesseur ;

$E_{orientation}$ **:** une contrainte lié à la mesure de l'orientation du gradient au point P_e ;

E_{image} **:** l'opposé de la norme du gradient au point candidat pour privilégier les points de contours forts.

On recherche parmi les trois points 8-connexe qui font varier l'orientation de la chaîne de 0° ou $\pm 45^\circ$ celui qui minimise la fonction de coût. Si ce dernier à un niveau de gradient suffisant, il prolonge le contour d'un élément.
Un contour prolongé peut rencontrer une autre chaîne pour former des coins ou des noeuds. S'il rencontre l'extrémité d'une autre chaîne, on réalise une fusion si un angle droit ne se forme pas entre ces deux structures, sinon on forme un coin. Si le contour rencontre un point interne d'un autre contour, on aura alors formation d'un noeud.
Lorsque l'on a atteint un seuil minimal du gradient, la prolongation s'arrête. On a donc alors extrait un ensemble de contours à une résolution donnée. Ces structures extraites peuvent maintenant être transposées à la résolution supérieure pour être prolongées. À la résolution supérieure, de nouvelles structures et détails peuvent aussi émerger.

(a)

(b)

Figure 4 – *Une image de la séquence* `Foreman` *(a) et les extrema locaux du gradient de norme supérieure à* $0,05\%$ *de la norme maximale théorique (b). Attention, ici aucun contour n'est formé, les points sont à relier.*

3.2 Affinement d'un contour à la résolution directement supérieure

Les éléments 8-connexes d'un contour sont projetés à la résolution juste supérieure. Ces points, qui se retrouvent distants de 2 ou $2\sqrt{2}$ pixels, sont considérés comme l'initialisation d'un contour actif dont l'énergie est modélisée par :

$$\alpha' E_{élasticité} + \beta' E_{courbure} + \gamma' E_{image} \qquad (2)$$

où :

$E_{élasticité}$ **:** représente l'élasticité entre les éléments de contours. Ce terme est minimal quand ces points restent à une distance constante les uns des autres ;

$E_{courbure}$ **:** représente la courbure du contour, cette énergie est minimale pour une courbure nulle ;

E_{image} **:** est égale à l'opposé de la norme du gradient afin d'attirer les éléments du contours actif vers les points de forte norme du gradient.

Cette énergie est minimisée par un algorithme glouton [8]. Les points du contour actif sont déplacés à tour de rôle pour minimiser la fonction de coût. Lorsqu'un minimum local est atteint ou qu'un certain nombre d'itérations est dépassé, les éléments du contour sont reliés entre eux par l'algorithme de Bresenham.

Nous attirons l'attention sur le fait qu'un noeud qui se forme à une résolution donnée, ne correspond pas nécessairement à un noeud à la résolution supérieure. La norme du gradient décroit à proximité des noeuds, par conséquent la position des noeuds en est moins précise. Pour cette raison, les extrémité de contours de type noeud ne sont pas projetés à la résolution supérieure. Si deux contours forment un contours à une résolution donnée, ils seront projetés à la resolution supérieure sans être connecté, mais pourront l'être par la suite par prolongation de ces contours.

4 Résultats

La prolongation des contours à pleine résolution ne parvient pas toujours à relier des structures qui appartiennent vraisemblablement au même contour. Dans certains cas une chute locale du gradient ne permet pas un lien entre ces structures, dans d'autres cas un coin se forme à la connexion entre deux structures qui empêche leur fusion. En procédant par exemple à une extraction de contours multirésolution sur deux niveaux de résolution, on permet une meilleure fusion entre les contours (voir figure 5).

Le nombre de contours ainsi détectés reste très important. Cette sur-détection n'est pas souhaitable dans un contexte de codage par bandelettes. Un post-traitement pour éliminer les contours de petite taille ne permettraient pas de supprimer systématiquement des contours proches. Par contre, une extraction de contours initiale à basse résolution, puis affiner ces contours de résolution en résolution sans faire apparaître de nouvelles structures à partir d'un certain niveau, permet d'obtenir des contours se situant à une certaine distance les uns des autres (voir figure 6).

5 Conclusion

Notre approche d'extraction de contours multirésolution permet une description à plusieurs niveaux de résolution de l'image. Dans le contexte d'une application de codage par transformée en bandelettes, il est nécessaire d'extraire des contours présentant une structure suffisamment longue afin de réduire leur coût de codage. L'extraction de contours à basse résolution, suivi d'un affinemment des structures extraites de résolution en résolution jusqu'à la pleine résolution permet de répondre à cette contrainte et d'obtenir des contours suffisamment éloignés.

Références

[1] Emmanuel J. Candès et David L. Donoho. Ridgelets : a key to higher-dimensional intermittency ? *Philosophical Transactions Royal Society London A*, (357) :2495–2509, 1999.

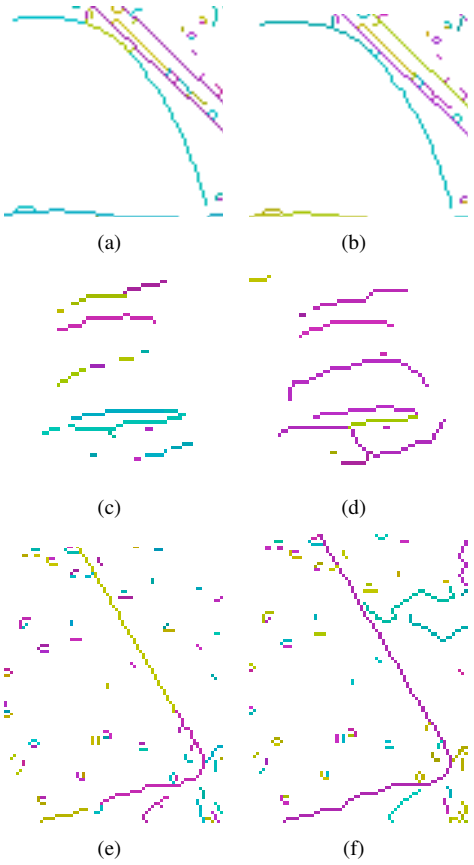


Figure 5 – *Des exemples de contours extraits directement à pleine résolution (a, c, e), et avec à deux niveaux de résolution (b, d, f).*

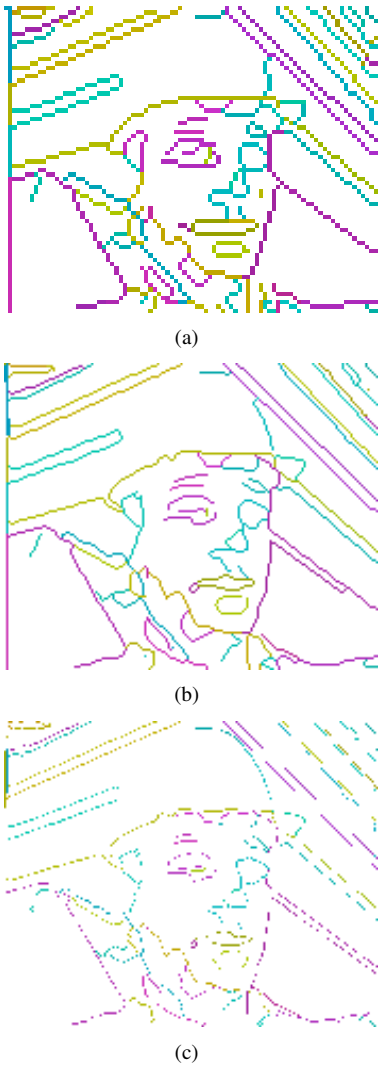


Figure 6 – *Extraction de contours sur le deuxième niveau d'une décomposition multirésolution (a) puis affinage des structures extraites à la résolution de niveau un (b), puis à la pleine résolution (c).*

[2] Minh N. Do et Martin Vetterli. The finite ridgelet transform for image representation. *IEEE Transactions On Image Processing*, 12(1), janvier 2003.

[3] Jean-Luc Starck, Emmanuel J. Candès, et David L. Donoho. The curvelet transform for image denoising. *IEEE Transactions On Image Processing*, 11 :670–684, novembre 2000.

[4] Minh N. Do et Martin Vetterli. The contourlet transform : An efficient directional multiresolution image representation. *IEEE Transactions On Image Processing*, 14(12) :2091–2106, december 2005.

[5] Demin Wang, Liang Zhang, et André Vincent. Curved wavelet transform and overlapped extension for image coding. Dans *ICIP 2004*, Singapour, 2004.

[6] Erwan Le Pennec. *Bandelettes et représentation géométrique des images*. Thèse de doctorat, École Polytechnique, decembre 2002.

[7] Peter J. Burt et Edward H. Adelson. The laplacian pyramid as a compact image code. Dans *IEEE Transactions On Communications*, volume 31, pages 532–540, 1983.

[8] Jean-Jacques Rousselle. *Les contours actifs, une méthode de segmentation. Application à l'imagerie médicale*. Thèse de doctorat, Université François Rabelais de Tours, juillet 2003.

Augmentation de la résolution temporelle du banc de filtres du codage MPEG AAC à l'aide de transformées orthogonales

Ewen Camberlein Pierrick Philippe

France Télécom R&D
4, rue du Clos Courtel
35 512 Cesson Sévigné Cedex

{ewen.camberlein,pierrick.philippe}@orange-ft.com

Résumé

Le système de référence pour le codage des signaux de musique est aujourd'hui la norme MPEG AAC [1]. Un codage par transformée est utilisé et deux tailles de transformée sont préconisées (1024 ou 128) suivant la nécessité d'avoir une bonne résolution temporelle ou fréquentielle pour coder un signal particulier. Ce changement de taille requiert l'utilisation de fenêtres de transition, qui concentrent peu l'énergie du signal. Ces fenêtres de transition imposent de plus un délai et une complexité de codage supplémentaire. La technique présentée dans cet article se base sur la combinaison des coefficients transformés par l'utilisation de transformées orthogonales afin d'améliorer à la volée la résolution temporelle sans fenêtre de transition. Dans ce contexte, l'étude présente un critère d'évaluation d'une transformée orthogonale donnée, par l'études des valeurs de résolutions temporelle et fréquentielle du banc de filtres considéré.

Mots clefs

Codage audio numérique, MDCT, Matrices orthogonales, résolution temporelle, Heisenberg.

1 Introduction

Dans une application de codage des signaux de musique, les coefficients temporels du signal sont traités et transmis au décodeur par blocs de N échantillons. Dans l'encodeur, ils sont d'abord représentés comme une combinaison linéaire des fonctions de base de la transformée et subissent ensuite une quantification suivie d'un codage entropique avant d'être transmis.
Un problème typique du codage par transformée est que le bruit de quantification introduit par la quantification des coefficients est réparti au décodage sur l'ensemble du bloc des N échantillons. Ceci est particulièrement audible pour des signaux transitoires, pour lesquels des artefacts gênants comme les effets de "pré-écho" apparaissent.
Une des solutions préconisées [1] pour répondre à ce problème consiste à changer de taille de transformée en diminuant celle-ci afin d'améliorer la résolution temporelle lorsque des signaux transitoires sont détectés. Cela implique un délai et une complexité d'encodage supplémentaire car il faut utiliser des fenêtres de transitions asymétriques, comme présenté Figure 1. Ces fenêtres de transition sont nécessaires pour conserver la propriété de reconstruction parfaite de la transformée. De plus les fenêtres de transition utilisées ne sont pas idéales en termes de résolution temps fréquence et induisent une perte en efficacité de codage.

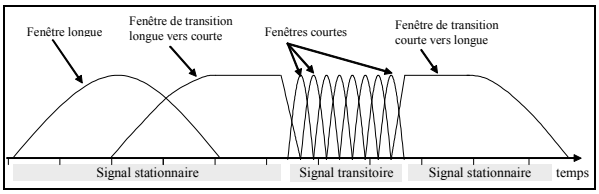


Figure 1 : *Exemple de succession de différentes fenêtres utilisées dans la norme MPEG AAC.*

Dans ce document nous présentons différentes méthodes pour effectuer la combinaison linéaire des fonctions de base de la transformée, afin d'obtenir des coefficients dans le domaine transformé ayant une meilleure localisation temporelle. Cette augmentation de la résolution temporelle sera ainsi obtenue sans recours à des fenêtres de transition.

L'opération effectuée consiste à combiner un certain nombre de coefficients successifs en sortie de la MDCT. Les coefficients transformés par une MDCT sont obtenus pour une fenêtre donnée $h[n]$ et un signal temporel $x[n]$ par :

$$X[k] = \sum_{n=0}^{2N-1} x[n] * h[n] * \cos\left[\frac{\pi}{N}\left(n + \frac{N+1}{2}\right)\left(k + \frac{1}{2}\right)\right] \qquad (1)$$

On transforme M coefficients X[k], exprimés sous forme d'un vecteur $X_k = {}^T\left[X[k],\ldots,X[k+M-1]\right]$, à l'aide d'une

matrice carrée A_M de taille MxM et on obtient le vecteur $X'_k = {}^T\left[X'[k],...,X'[k+M-1]\right]$:

$$X'_k = A_M X_k \quad (2)$$

L'application de cette transformée A_M permet d'améliorer la résolution temporelle de la MDCT et de construire des transformées non uniformes adaptées au signal à coder à un instant donné. Différents scénarios d'utilisations de ces transformées blocs et les découpes temps fréquences en résultant sont présentés Figure 2 pour une MDCT de taille N=16.

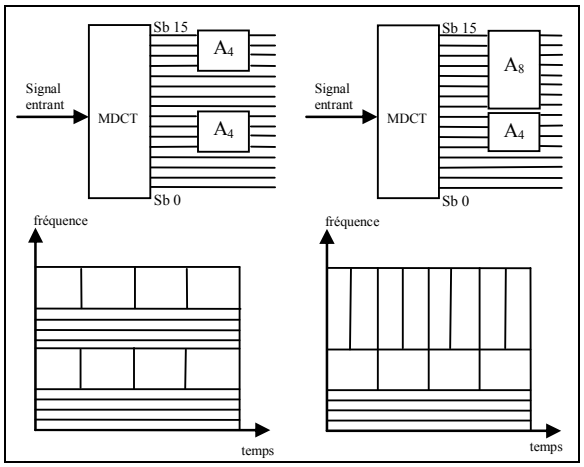


Figure 2 : *Exemples de découpes temps/ fréquence.*

Nous nous restreignons ici à l'étude des découpes uniformes, construites grâce à cette technique. Nous proposons une mesure de performance de ces transformées basé sur leur résolution temporelle et fréquentielle. Nous concluons sur les performances atteignables par ce type de structures en comparaison à celle utilisée par l'AAC.

2 Etat de l'art

Une méthode proposée par Mau [2] consiste à effectuer des opérations de somme et de différence appliquées sur deux coefficients adjacents. La transformée utilisée, équivalente à une transformée de Hadamard de taille 2 s'écrit alors :

$$A_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Nous pouvons voir Figure 3 et Figure 4 les modifications apportées aux réponses fréquentielles et impulsionnelles de la transformée initiale et après application de cette transformée de Hadamard appliquée sur les coefficients 17 et 18 d'une MDCT de taille 32.

En combinant deux coefficients, deux nouvelles sous bandes sont obtenues ayant la même localisation fréquentielles mais une localisation temporelle différente.

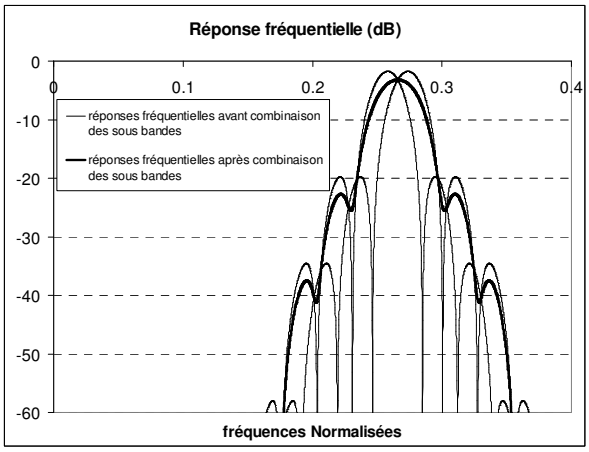


Figure 3 : *Réponses fréquentielles avant et après combinaison des coefficients.*

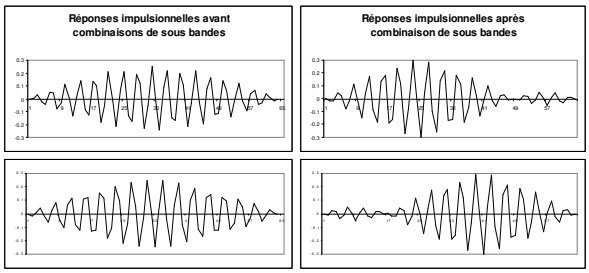


Figure 4 : *Réponses impulsionnelles avant et après combinaison des coefficients.*

Mau propose également l'utilisation de matrices de Hadamard de taille 4 appliquée sur 4 coefficients adjacents.

Malvar propose l'utilisation d'une autre transformée de taille 4 [3] afin d'augmenter la résolution temporelle en haute fréquences dans le cas d'un codage de signal de paroles.

Dans [4] Niamut propose l'utilisation de matrices de Hadamard de tailles quelconques. Il montre que l'utilisation de telles matrices assure que le module des réponses fréquentielles des fonctions de base combinées soit égal à la somme des modules des réponses fréquentielles des sous bandes avant combinaison.

Cette contrainte forte ne nous semble pas fondamentale pour une application de codage : nous étendons ici l'étude de ces transformées au cas des matrices orthogonales.

L'orthogonalité garantie la conservation de l'énergie des coefficients par la transformée. Cette propriété permet que la variance moyenne de l'erreur de quantification introduites sur les coefficients temporels soit égale à la variance moyenne de l'erreur introduite sur les coefficients transformés. Cela permet une évaluation simple de la qualité de codage lors de l'étape d'allocation de bits et donc assure une quantification simple des coefficients transformés [5].

3 Critères d'évaluation de l'intérêt d'une transformée donnée

La localisation temporelle et fréquentielle des coefficients du signal est importante en codage audio. Ceci permet de mettre en forme le bruit de quantification en tenant compte du masquage fréquentiel pour les signaux stationnaires et du masquage temporel pour les signaux transitoires.

Nous évaluons donc ici la transformée résultante de ces deux opérations (MDCT + transformées orthogonales) en termes de résolution temporelle σ_t^2 et fréquentielle σ_f^2. Ces résolutions sont obtenues à partir des réponses impulsionnelles $h_{sb}[n]$ et fréquentielles $H_{sb}[k]$. Les formules, dérivées de [6], utilisées pour calculer ces résolutions (pour une fréquence d'échantillonnage Fe) pour chaque sous bande sb, sont les suivantes :

$$\sigma_{t,sb}^2 = E_{t,sb}^{-1} \sum_{n=0}^{2N-1} \left(\frac{n}{Fe} - \mu_{t,sb} \right)^2 h_{sb}^2[n] \tag{3}$$

$$\mu_{t,sb} = E_{t,sb}^{-1} \sum_{n=0}^{2N-1} \frac{n}{Fe} h_{sb}^2[n] \tag{4}$$

$$E_{t,sb} = \sum_{n=0}^{2N-1} h_{sb}^2[n] \tag{5}$$

$$\sigma_{f,sb}^2 = \begin{cases} E_{f,sb}^{-1} \sum_{k=-FFTSize/2}^{FFTSize/2} \left(\frac{kFe}{FFTSize} - \mu_{f,sb} \right)^2 |H_{sb}[k]|^2 & si\ sb = 0 \\ E_{f,sb}^{-1} \sum_{k=-FFTSize/2}^{FFTSize/2} \left(\left| \frac{kFe}{FFTSize} \right| - \mu_{f,sb} \right)^2 |H_{sb}[k]|^2 & si\ 0 < sb < N-1 \\ E_{f,sb}^{-1} \sum_{k=0}^{FFTSize} \left(\frac{kFe}{FFTSize} - \mu_{f,sb} \right)^2 |H_{sb}[k]|^2 & si\ sb = N-1 \end{cases} \tag{6}$$

$$\mu_{f,sb} = \begin{cases} E_{f,sb}^{-1} \sum_{k=-FFTSize/2}^{FFTSize/2} \frac{kFe}{FFTSize} |H_{sb}[k]|^2 & si\ sb = 0 \\ E_{f,sb}^{-1} \sum_{k=-FFTSize/2}^{FFTSize/2} \left| \frac{kFe}{FFTSize} \right| |H_{sb}[k]|^2 & si\ 0 < sb < N-1 \\ E_{f,sb}^{-1} \sum_{k=0}^{FFTSize} \frac{kFe}{FFTSize} |H_{sb}[k]|^2 & si\ sb = N-1 \end{cases} \tag{7}$$

$$E_{f,sb} = \begin{cases} \sum_{k=-FFTSize/2}^{FFTSize/2} |H_{sb}[k]|^2 & si\ 0 \le sb < N-1 \\ \sum_{k=0}^{FFTSize} |H_{sb}[k]|^2 & si\ sb = N-1 \end{cases} \tag{8}$$

Pour mesurer l'intérêt d'une transformée bloc donnée nous calculons la moyenne des résolutions temporelles et fréquentielles :

$$\sigma_f^2 = \frac{1}{N} \sum_{sb=0}^{N-1} \sigma_{f,sb}^2 \tag{9}$$

$$\sigma_t^2 = \frac{1}{N} \sum_{sb=0}^{N-1} \sigma_{t,sb}^2 \tag{10}$$

Le principe d'incertitude d'Heisenberg nous permet d'évaluer si le compromis résolution temporel/ résolution fréquentielle est proche de la borne théorique :

$$\sigma_t^2 \sigma_f^2 \ge \frac{1}{(4\pi)^2} \tag{11}$$

4 Résultats

Les transformées blocs optimales A* présentées Figure 6 résultent de l'optimisation du paramètre de résolution temporelle moyenne σ_t^2 sur le sous ensemble des transformées blocs orthogonales constitué par l'ensemble des matrices de rotations {A}.

La matrice A* est donc recherchée tel que :

$$A^* = \arg\min_{\{A\}} \left(\sigma_t^2 \right) \tag{12}$$

En utilisant le fait qu'une matrice de rotation de dimension M*M peut être définie par seulement $M/2 (M-1)$ angles, cette optimisation devient possible en un temps raisonnable.

Nous avons testé les différents types de fenêtres utilisés par l'AAC (Figure 5). Il s'agit des fenêtres de Kaiser

Bessel Dérivées (KBD) et sinusoïdales de taille 1024 [1]. Les formes de ces fenêtres permettent de concentrer différemment l'énergie temporelle du signal, et ont une influence directe sur les résolutions temporelles obtenues.

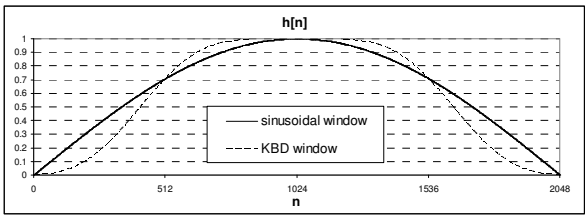


Figure 5 : *Différents types de fenêtres utilisés dans la norme MPEG AAC.*

Sur la Figure 6 sont représentées un certain nombre de transformées caractérisées par leurs résolutions temporelles et fréquentielles moyennes (pour une fréquence d'échantillonnage de 48 kHz).

A titre de références, nous avons tracé les caractéristiques de Heisenberg et celle obtenue par la MDCT seule. Sont également présentés les résultats obtenus après application des matrices orthogonales optimales, obtenues après optimisation sur les fenêtres définies par la norme AAC.

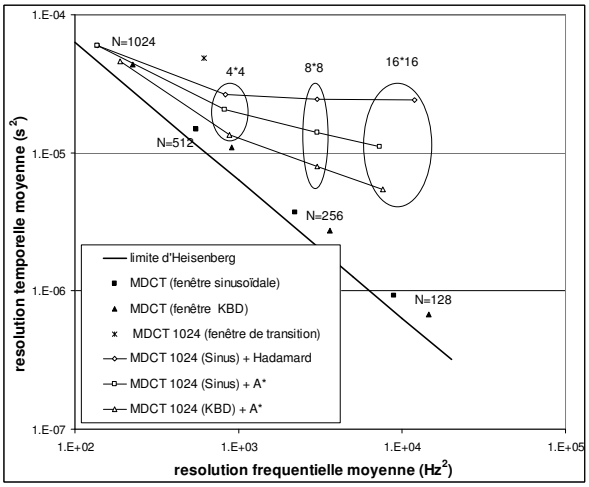


Figure 6 : *Comparaison des concentrations énergétiques obtenues par différentes transformées.*

L'utilisation de transformées orthogonales en sortie d'une MDCT utilisant une fenêtre de Kaiser Bessel dérivée (KBD) permet une meilleure efficacité par rapport à une configuration utilisant des fenêtres sinusoïdales.

Il apparaît clairement sur cette figure que l'utilisation de transformées de Hadamard ne permet pas d'obtenir les meilleurs résultats. Les matrices optimales obtenues par notre algorithme offrent de meilleures concentrations temps fréquence.

En revanche, on observe que l'application de transformées orthogonales ne permet pas de retrouver une concentration énergétique aussi bonne que celle de la MDCT : on s'éloigne de la limite d'Heisenberg à mesure que la taille de la transformée orthogonale croît.

Ces résultats montrent également qu'il semble difficile de retrouver une résolution temporelle équivalente à celle d'une MDCT de taille 128 à partir d'une MDCT de taille 1024 sur laquelle est appliquée une matrice orthogonale.

A titre d'illustration, nous présentons Figure 7 la concentration temporelle obtenue par les fonctions de bases avant et après transformée orthogonale, dans le cadre d'une transition. Ces résultats sont présentés dans le cadre des fenêtres sinusoïdales et de Kaiser Bessel dérivées. On observe la meilleure concentration d'énergie temporelle obtenue grâce aux fenêtres KBD.

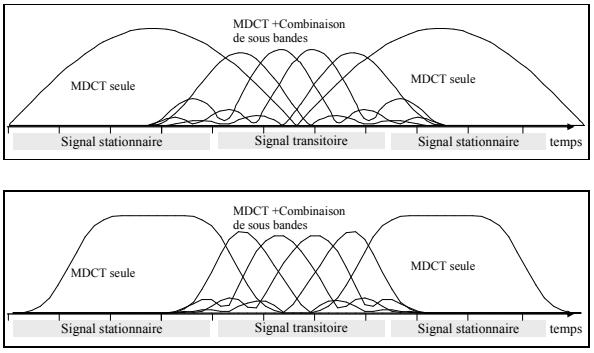


Figure 7 : Exemple de fenêtres temporelles obtenues par l'utilisation d'une transformée A_4^*.

Les matrices A_4^* utilisées Figure 7 sont les suivantes :

$$A^*_{4,Sinus} = \begin{bmatrix} 0.378621 & 0.597329 & 0.597190 & 0.378429 \\ -0.597205 & -0.378430 & 0.378656 & 0.597291 \\ 0.597254 & -0.378561 & -0.378489 & 0.597265 \\ -0.378524 & 0.597227 & -0.597269 & 0.378562 \end{bmatrix}$$

$$A^*_{4,KBD} = \begin{bmatrix} 0.386597 & 0.592050 & 0.592072 & 0.386614 \\ -0.592050 & -0.386643 & 0.386618 & 0.592040 \\ 0.592102 & -0.386662 & -0.386548 & 0.592020 \\ -0.386569 & 0.592010 & -0.592079 & 0.386692 \end{bmatrix}$$

5 Conclusion

Nous avons défini une méthode permettant de sélectionner une transformée orthogonale afin d'augmenter la résolution temporelle d'une MDCT de taille donnée. Cette technique permet l'utilisation de différents compromis résolution temporelle/ résolution fréquentielle qui n'étaient pas envisageables avec les transformées utilisées dans la norme MPEG AAC, et ceci sans utilisation de fenêtres de transition. Par contre obtenir une résolution temporelle équivalente à celle d'une MDCT 128 semble difficile.

La suite de cette étude consistera à définir un sous ensemble des transformées présentées ainsi qu'un critère permettant de décider comment construire un banc de filtre non uniforme s'adaptant à un signal donné. La quantité d'information à transmettre au décodeur pour la construction de la transformée inverse sera également prise en considération dans cette étude. L'insertion de cette technique au sein d'un schéma de codage nous permettra enfin d'évaluer les performances envisageables par l'utilisation de ces transformées.

Références

[1] MPEG-2 Advanced Audio Coding, Norme internationale *AAC.*, ISO/IEC 13818-7, Edition 2003.

[2] J. Mau. et al. Time-varying orthogonal filter banks without transient filters. *ICASSP,* vol. 2, pages 1328 – 1331. mai 1995

[3] H. S. Malvar. Enhancing the performance of subbands audio coders for speech signals. Dans *Proc. Int. Symp. Circuits and Systems'98* : 90-101, juin 1998.

[4] O.A. Niamut et R. Heusdens. Subband merging in cosine-modulated filter banks. *IEEE Signal processing letter*. vol. 10,num. 4, avril 2003.

[5] N.S. Jayant et Peter Noll, *Digital Coding of waveforms*, pages 517- 525, Prentice-Hall, 1984.

[6] C. Taswell. Empirical tests for the evaluation of multirate filter bank parameters. Rapport technique. Computational Toolsmiths. Stanford, février 1998.

Etude de maillage rectangulaire déformable pour l'estimation de mouvement dans des séquences vidéo

Vianney Muñoz-Jimenez[+*], Anissa Zergaïnoh[+*]

[*]L2TI, Institut Galilée, Université Paris 13
99, Avenue Jean Baptiste Clément, 93 430 Villetaneuse, France

[+]LSS/CNRS, Supelec
Plateau de Moulon, 91 192 Gif sur Yvette, France

Email : {munoz, anissa.zergainoh}@lss.supelec.fr

Résumé

L'estimation de mouvement adoptée par les standards vidéo actuels est basée sur l'algorithme d'appariement de blocs. Cet algorithme ne détermine que les mouvements de translation des objets dans la scène. A bas débit, ces mouvements se traduisent par des effets visuels d'artefacts sur les images prédites. Pour mieux décrire les différents types de mouvement d'objets dans la scène, nous étudions l'estimation de mouvement à partir de la déformation des mailles rectangulaires des grilles associées aux images de référence.

Mots clefs

Interpolation, spline, maillage rectangulaire, déformation, estimation de mouvement.

1 Introduction

Une séquence vidéo est caractérisée par un volume de données important qu'elle doit véhiculer à travers un réseau de transmission ou occuper sur un support de stockage. Elle est représentée par un ensemble d'images consécutives fortement corrélées. Celles-ci défilent les unes après les autres, à une cadence temporelle régulière et prédéfinie, pour donner une illusion de mouvement. Afin de diminuer le coût de codage, les redondances temporelles inter-images sont généralement exploitées. En effet, les images de la séquence vidéo ne sont pas codées de manière indépendante. A ce jour, l'estimation et la compensation de mouvement sont des techniques largement développées et utilisées par les standards vidéo dans le but de réduire les redondances temporelles ([1], [2], [3], [4], [5], [6]). Généralement, la prédiction d'une image courante de la séquence vidéo est obtenue à partir d'une ou de plusieurs images préalablement reconstruites. Cette prédiction inclut une opération de compensation de mouvement de l'image courante. Le mouvement apparent 2D dans une séquence vidéo peut avoir plusieurs origines : le mouvement d'objets 3D dans la scène, les mouvements de la caméra ou les variations d'illuminations. Dans les normes actuelles de vidéo, le champ de mouvement des objets dans la scène est très souvent représenté à partir d'un modèle de translation basé sur une approche d'appariement de blocs de l'image courante avec l'image de référence (ou vis versa). Chaque bloc de l'image courante est obtenu par translation d'un bloc de l'image de référence qui semble se rapprocher au mieux (selon une mesure prédéfinie) du bloc de l'image de référence. Le principal intérêt de cette approche est sa simplicité. Néanmoins les blocs compensés ne forment pas toujours une partition complète de l'image de référence. Il peut apparaître dans l'image prédite des zones qui sont soient découvertes ou recouvertes puisque aucune contrainte n'est imposée aux blocs adjacents dans l'image courante. Afin de remédier aux inconvénients de cette approche, de nouvelles méthodes basées sur la construction d'un maillage déformable ont été développées ([7]).

Des maillages uniformes à base de mailles rectangulaires, triangulaires et hexagonales ont été proposés ([8], [9], [10], [11], [12], [13]). Des méthodes proposent d'utiliser un seul modèle optimal de déformation adapté à toute la séquence vidéo considérée ([14], [15]). Certaines méthodes combinent également l'appariement de bloc avec le maillage uniforme. Tandis que d'autres méthodes emploient des décompositions hiérarchiques (sous-bandes, quad-tree) pour réduire le temps de calcul nécessaire à l'estimation de mouvement ([16], [17], [18], [19], [20], [21], [22], [23]). Certains algorithmes de maillage sont adaptés au contenu de l'image. Bien que performants, certains algorithmes souffrent de complexité de calculs.

Dans cet article nous étudions et comparons les performances de l'estimation de mouvement dans une séquence vidéo à partir de la déformation des mailles rectangulaires des grilles associées aux images. Nous présentons trois types découpages.

Cet article est organisé comme suit. La section 2 introduit le concept général de la représentation des images animées par maillage rectangulaire. La section 3 concerne l'estimation de mouvement locale obtenue à partir de l'optimisation des déplacements des nœuds des différentes mailles. La section 4 analyse les résultats expérimentaux obtenus.

2 Présentation de l'approche basée maillage

Une séquence vidéo est représentée par un ensemble d'images successives fortement corrélées. Afin d'obtenir de bonnes performances de codage, les standards vidéo actuels classifient les images en trois catégories selon la pertinence de l'information contenue dans les images : (i) les images de type I (codées en Intra), (ii) les images prédites de type P (codées en Inter) et (iii) les images bidirectionnelles de type B (codées en Inter). Dans le cadre de notre premier travail, nous ciblons les applications de type vidéoconférence (cadre simple). Dans cet article, nous utilisons une méthode d'estimation de mouvement basée maillage de type « forward ». Introduisons, tout d'abord, dans le paragraphe ci-dessous la construction de la grille de référence nécessaire pour l'estimation de mouvement.

2.1 Construction de la grille de référence

Considérons la $k-1$ ième image de type I dans la séquence vidéo. Notons I_{k-1} l'intensité de cette image de référence. Une grille de référence, notée G_{k-1}, est associée à l'image I_{k-1}. Cette grille G_{k-1} est construite progressivement à partir d'un ensemble de mailles rectangulaires, notées $\{\mathcal{M}_i^{k-1},\ i \in N\}$.
Une maille rectangulaire $\mathcal{M}_i^{k-1}$ est définie à partir de 4 nœuds choisis sur l'image à mailler (voir Figure 3 (a)). Les mailles rectangulaires sont juxtaposées les unes à coté des autres afin de former une partition continue de la grille de référence.
Dans cet article nous comparons les performances de trois types de maillage rectangulaire de la grille de référence G_{k-1} : (i) le maillage uniforme (ii) le maillage semi-régulier et adapté (iii) le maillage hiérarchique et adapté. Les deux derniers types de maillage sont basés sur le gradient de l'image de référence ∇I_{k-1}. Cette stratégie est motivée par le fait que le mouvement des objets dans une scène peut être déduit par uniquement par le suivi du déplacement des contours de ces objets dans l'image suivante. Nous verrons, ci-dessous, que la grille de référence G_{k-1} est construite itérativement. Commençons par présenter le maillage hiérarchique et adapté.

Maillage hiérarchique adapté
Dans le cadre de maillage hiérarchique, nous commençons par un pavage rectangulaire uniforme et grossier de l'image de référence I_{k-1}. Ce découpage s'apparente au découpage quad-tree. La différence réside dans le fait que la grille est initialement découpée en maille rectangulaire de taille identique. Le pavage initial est ensuite affiné en fonction du gradient appliqué à l'image de référence ∇I_{k-1}. Chaque maille est à son tour subdivisée en maille rectangulaire de façon à ce que les contours soient isolés dans des mailles de taille plus petites. Un exemple de pavage est donné par la Figure 1, où les pointillés correspondent aux contours de l'image de référence. Les nœuds des mailles sont marqués par des points.

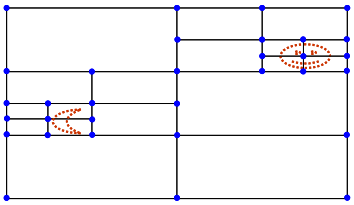


Figure 1 – *Maillage adapté hiérarchique*

Maillage semi régulier adapté
Dans ce type de maillage, nous procédons tout d'abord à un découpage horizontal (grossier) de l'image de référence I_{k-1}. Celui-ci est ensuite affiné horizontalement selon le gradient de l'image de référence. Pour l'exemple de la Figure 2, sept découpages horizontaux ont été appliqués sur l'image de gradient. Le processus est ensuite répété verticalement. Cinq découpages horizontaux sont réalisés pour construire le maillage de l'exemple de la Figure 2.

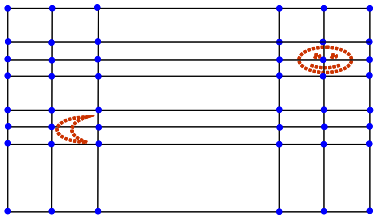


Figure 2 – *Maillage adapté et semi régulier*

Quelque soit le type de maillage adopté, les mailles forment une partition continue de l'image de référence.

3 Estimation de mouvement par maillage rectangulaire déformable

L'estimation de mouvement présentée dans cet article est basée sur le principe de la déformation de la grille maillée G_{k-1} construite au paragraphe précédent.

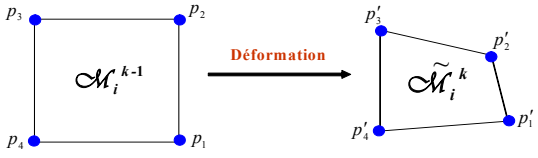


Figure 3 – *Mailles :* (a) *rectangulaire,* (b) *déformée*

Cette déformation se traduit par le déplacement des nœuds de chaque maille tout en préservant la continuité des mailles associées à l'image courante I_k. La Figure 3 présente une maille rectangulaire $\mathcal{M}_i^{k-1}$ décrite par ses 4 nœuds p_1, p_2, p_3, p_4. Cette maille est déformée en une maille quadrilatère, notée $\tilde{\mathcal{M}}_i^{k}$, décrite par ses 4 nœuds p'_1, p'_2, p'_3, p'_4. Les déplacements optimaux des nœuds p_1, p_2, p_3, p_4 sont choisis de façon à minimiser l'erreur quadratique moyenne d'appariement des mailles. Nous

commençons par introduire sommairement la méthode de reconstruction basée sur la fonction spline bilinéaire.

3.1 Fonction spline bilinéaire

Ce paragraphe s'intéresse à la prédiction des pixels à l'intérieur d'une maille rectangulaire quelconque $\subset\!\mathcal{M}_l^{k-1} \in G_{k-1}$ décrite par ses 4 positions nodales notées p_1, p_2, p_3, p_4 (voir Figure 3). Pour des commodités d'explications, notons les coordonnées de ces nœuds comme suit :

$$p_1 = (s_{i+1}, t_{i+1}),\ p_2 = (s_i, t_{i+1}),\ p_3 = (s_i, t_i)$$

et $p_4 = (s_{i+1}, t_i)$ avec $s_i < s_{i+1}$ et $t_i < t_{i+1}$.

De plus, les intensités des pixels en ces nœuds sont connues. Soit $f(s,t)$ la fonction d'interpolation spline bilinéaire définie sur le domaine rectangulaire suivant :

$$[s_i, s_{i+1}] \otimes [t_i, t_{i+1}]$$

où $\otimes$ représente le produit tensoriel.

La fonction spline bilinéaire $f(s,t)$ est donnée par l'expression suivante ([24]) :

$$f(s,t) = \sum_{l=0}^{1} \sum_{m=0}^{1} a_{l,m} B^1_{l,[s_i,s_{i+1}]}(s) B^1_{m,[t_i,t_{i+1}]}(t)$$

où l'ensemble des coefficients $\{a_{l,m}\}$ représentent les coefficients de la surface spline.

L'ensemble des éléments

$$\left\{ B^1_{l,[s_i,s_{i+1}]}(s) B^1_{m,[t_i,t_{i+1}]}(t) \right\}$$

constitue la base de l'espace spline. La l ième fonction B-spline non-uniforme $B^1_{l,[s_i,s_{i+1}]}(s)$ correspond à un polynôme de degré un. Donnons ci-dessous les expressions respectives des fonctions B-splines de degré un :

$$B^1_{0,[s_i,s_{i+1}]}(s) = (s - s_i)/(s_{i+1} - s_i)\ ;$$

$$B^1_{1,[s_i,s_{i+1}]}(s) = (s_{i+1} - s)/(s_{i+1} - s_i)\ ;$$

$$B^1_{0,[t_i,t_{i+1}]}(t) = (t - t_i)/(t_{i+1} - t_i)\ ;$$

$$B^1_{1,[t_i,t_{i+1}]}(t) = (t_{i+1} - t)/(t_{i+1} - t_i)\ ;$$

avec $s_i \le s \le s_{i+1}$ et $t_i \le t \le t_{i+1}$

Le calcul des coefficients de la surface spline est déterminé facilement ([25]) :

$$a_{0,0} = I_{k-1}(s_{i+1}, t_{i+1}),\ a_{1,0} = I_{k-1}(s_i, t_i),\ a_{0,1} = I_{k-1}(s_i, t_{i+1})$$

et $a_{1,1} = I_{k-1}(s_{i+1}, t_i)$

3.2 Prédiction de l'image courante à partir de la grille déformée

Ce paragraphe concerne la prédiction de l'image courante I_k à partir des nœuds appartenant à la grille déformée G_k. Supposons que les positions des noeuds d'une maille déformée soient connues. Les coordonnées de ces nœuds sont notées comme suit :

$$p'_1 = (x_i, y_i),\ p'_2 = (x_{i+1}, y_{i+1}),\ p'_3 = (x_{i+2}, y_{i+2}) \text{ et}$$

$$p'_4 = (x_{i+3}, y_{i+3}).$$

Rappelons que la méthode de reconstruction décrite ci-dessus est valide dans le cas d'une maille rectangulaire puisque les éléments de la base spline sont construits sur un domaine rectangulaire de type $[s_i, s_{i+1}] \otimes [t_i, t_{i+1}]$. L'approche utilisée dans cet article s'inspire des résultats développés dans le domaine des éléments finis ([26], [27]). Les déplacements des nœuds d'une maille rectangulaire $\subset\!\mathcal{M}_l^{k-1}$ sont modélisés par une fonction de déformation (warping) notée $w(s,t)$.

Dans cet article, la fonction de déformation est choisie comme étant une fonction d'interpolation bilinéaire ([28], [29]):

$$\boldsymbol{w}(s,t) = \begin{bmatrix} \xi(s,t) \\ \eta(s,t) \end{bmatrix} = \sum_{i=0}^{1} \sum_{j=0}^{1} \underline{\boldsymbol{c}}_{i,j} B^1_{i,[s_i,s_{i+1}]}(s) B^1_{j,[t_i,t_{i+1}]}(t)$$

où les coefficients $\underline{\boldsymbol{c}}_{i,j}$ sont donnés par :

$$\underline{\boldsymbol{c}}_{0,0} = [x_i, y_i]^t,\ \underline{\boldsymbol{c}}_{0,1} = [x_{i+1}, y_{i+1}]^t,\ \underline{\boldsymbol{c}}_{1,0} = [x_{i+2}, y_{i+2}]^t,$$

$$\underline{\boldsymbol{c}}_{1,1} = [x_{i+3}, y_{i+3}]^t.$$

La fonction de déformation inverse $w^{-1}(x,y)$ permet de transformer la maille déformée (quadrilatère) en une maille rectangulaire (fonction bijective). De ce fait, la prédiction des pixels dans la maille déformée, est ramenée au calcul de la prédiction des pixels dans le référentiel de la maille rectangulaire par le biais de la fonction d'interpolation spline décrite précédemment. La Figure 4 résume les différentes étapes à suivre pour prédire l'image courante.

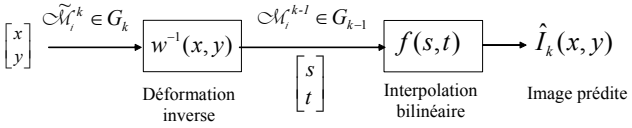


Figure 4 – *Prédiction de l'image courante à partir de la grille déformée*

3.3 Déformation de la grille de référence

Nous nous s'intéressons à la construction de la grille maillée de l'image courante G_k. La grille de l'image de référence G_{k-1} est superposée sur le gradient de l'image courante ∇I_k. La grille déformée G_k est construite progressivement. Les nœuds des mailles de la grille G_{k-1} sont déplacés sur le gradient de l'image ∇I_k de façon à minimiser l'erreur quadratique moyenne de prédiction. Nous accordons une importance particulière à cette phase d'optimisation puisque c'est l'erreur de prédiction (information résiduelle) qui est en général codée pour la compensation de mouvement.

L'erreur quadratique moyenne de prédiction des pixels dans la maille $\subset\!\mathcal{M}_l^k$ est calculée entre l'image courante prédite et l'image courante originale :

$$\varepsilon_p(\subset\!\widetilde{\mathcal{M}}_l^k) = \frac{1}{n(\subset\!\widetilde{\mathcal{M}}_l^k)} \sum_i (I_k(x_i, y_i) - \hat{I}_k(x_i, y_i))^2$$

où $\hat{I}_k(x_i, y_i)$ correspond au pixel prédit de coordonnée (x_i, y_i) et $n(\subset\!\widetilde{\mathcal{M}}_l^k)$ représente le nombre de pixels prédits dans la maille déformée $\subset\!\widetilde{\mathcal{M}}_l^k$.

On remarque dans ce contexte particulier que l'erreur de quadratique moyenne de prédiction est égale à l'erreur

quadratique moyenne d'appariement, notée $\varepsilon_A(\widetilde{\mathcal{M}}_l^k)$, puisque en effet :

$$\varepsilon_A(\widetilde{\mathcal{M}}_l^k) = \frac{1}{n(\widetilde{\mathcal{M}}_l^k)} \sum_{i\in} (I_{k-1}(x_i - \Delta x_i, y_i - \Delta y_i) - I_k(x_i, y_i))^2$$

où $\hat{I}_k(x_i, y_i) = I_{k-1}(x_i - \Delta x_i, y_i - \Delta y_i)$ et $(\Delta x_i, \Delta y_i)$ représente le déplacement d'un noeud par rapport à sa position initiale (x_i, y_i).

Pour réduire le temps de calcul nécessaire à la recherche optimale du déplacement d'un nœud, nous proposons de vérifier un certain nombre de points avant de prédire toutes les valeurs des pixels dans chacune des mailles $\{\widetilde{\mathcal{M}}_i^k\}$ affectées par le déplacement du nœud. La démarche consiste à vérifier si le nœud que nous envisageons de déplacer : (i) appartient à un contour de l'image courante ($\nabla I_k(p_j') \neq 0$); (ii) n'engendre pas de discontinuité dans la structure de la grille G_k, et (iii) préserve la structure quadrilatère (pas de forme dégénérée). La dernière vérification se traduit par le calcul du Jacobien donné ci-dessous. Celui-ci doit être positif pour tous les points de la maille considérée ([28]):

$$J(s,t) = \begin{vmatrix} \frac{\partial \xi(s,t)}{\partial s} & \frac{\partial \eta(s,t)}{\partial s} \\ \frac{\partial \xi(s,t)}{\partial t} & \frac{\partial \eta(s,t)}{\partial t} \end{vmatrix} > 0$$

4 Analyse des résultats expérimentaux

Dans cette section, nous présentons les résultats obtenus sur la séquence vidéo de test « *Miss America* » au format QCIF (4:2:0). La résolution spatiale de chaque image est de 144×176 et la résolution temporelle est de 12 images par seconde. Nous présentons les résultats d'estimation de mouvement obtenus sur les 11 premières images de la séquence vidéo numérotées de 0 à 10.

La première image « 0 » est choisie comme image de référence (de type I). Celle-ci est représentée par les trois composantes; I_0 (luminance), Cb_0 (chrominance bleue) et Cr_0 (chrominance rouge). La phase initiale de la méthode d'estimation de mouvement consiste à construite la grille maillée de référence G_0 associée à l'image d'intensité I_0. Rappelons que le maillage de référence est construit à partir des informations liées au gradient de l'image ∇I_0. Trois types de découpages sont analysés et comparés dans cette section.

Commençons par présenter le maillage hiérarchique adapté. L'image de gradient ∇I_0 est tout d'abord subdivisée en plusieurs mailles de taille identique. Dans cet exemple, nous avons fixé la taille initiale de chaque maille à 49×49. Certaines de ces mailles initiales, en accordance avec l'image de gradient ∇I_0, peuvent être ensuite subdivisées en mailles rectangulaires. Dans cet exemple, la taille de la plus petite maille est égale à 4×4. La Figure 5 présente le découpage superposé sur le gradient de l'image I_0. Il serait judicieux que les tailles initiale et finale des mailles puissent être ajustées par le codeur en fonction par exemple du débit imposé et de la qualité désirée de la séquence vidéo décodée. Cet aspect d'optimisation n'est pas considéré dans cet article et fera l'objet d'investigations futures.

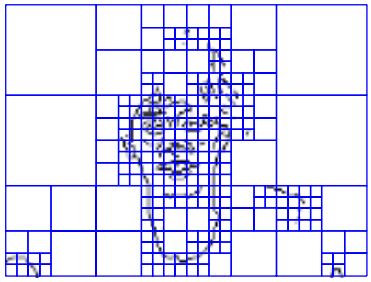


Figure 5 – *Maillage rectangulaire de la grille de référence*

La Figure 6 présente la déformation de la grille de référence G_0. Dans cet exemple, la grille G_0 est composée de 779 nœuds. Pour la phase d'estimation de mouvement décrite au paragraphe 3, parmi les 779 nœuds un premier trie des nœuds montre que seuls 524 nœuds sont autorisés à être éventuellement déplacé si les conditions (i) et (iii) sont vérifiées. Pour les composantes de chrominance (Cb_6, Cr_6), il suffit de redimensionner la grille déformée G_6 par un facteur 4 puisque le format utilisé correspond à 4:2:0.

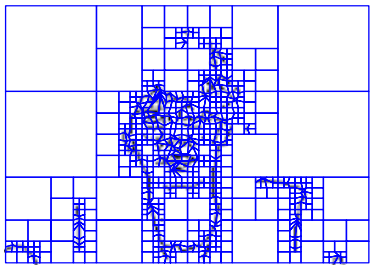


Figure 6 – *Maillage hiérarchique G_6 associé à l'image courante I_6*

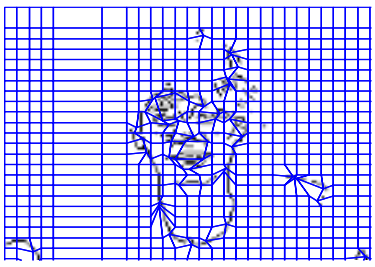


Figure 7 – *Maillage semi régulier G_6 associé à l'image courante I_6*

Afin de comparer les performances des trois types de maillage, nous avons ajusté le nombre de nœuds de chaque grille de référence de façon à travailler dans des conditions identiques. Pour un maillage rectangulaire uniforme, la taille de chaque maille est fixée à 7×7 (voir

Figure 8). Dans ce cas la grille régulière de référence G_0 est composée de 775 nœuds. La Figure 7 présente la déformation de la grille G_6 dans le cas d'un maillage semi régulier adapté composé de 784 nœuds.

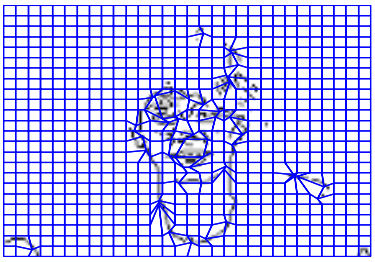


Figure 8 – *Maillage uniforme G_6 associé à l'image courante I_6*

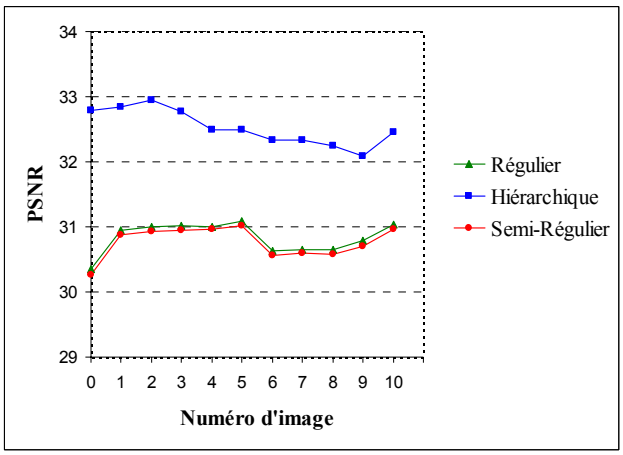


Figure 9 – *PSNR en fonction du numéro de l'image prédite dans la séquence vidéo ($I_0 - I_{10}$)*

La Figure 9 compare les erreurs quadratiques moyennes de prédiction (PSNR) des images d'intensité engendrées par les 3 types de déformation de maillage : uniforme, semi régulier et hiérarchique. Ces courbes montrent que nous obtenons de meilleurs résultats pour le maillage hiérarchique adapté. De plus, le temps nécessaire pour la phase de recherche des positions optimales est moins important comparé aux deux autres types maillages. Dans cet exemple particulier, parmi les 775 nœuds de la grille régulière, seuls 667 nœuds sont susceptibles d'être déplacé (contre 524 nœuds pour le maillage hiérarchique).

5 Conclusion

Dans cet article, nous nous sommes intéressés à l'estimation de mouvement à partir de maillage déformable adapté au gradient de l'image dans le but d'améliorer l'approche classique basée sur l'appariement de blocs. Notre méthode d'estimation nécessite, dans une première phase, la construction de la grille de référence associée à une image sélectionnée (de type Intra) dans la séquence vidéo. Le déplacement des nœuds de cette grille permet de modéliser le mouvement des objets dans l'image courante. La recherche de la position optimale du nœud à déplacer est dictée par la position qui engendre la plus petite erreur quadratique moyenne de prédiction dans les mailles. Les résultats de simulations ont montré que le maillage hiérarchique est le mieux adapté à l'estimation de mouvement. De plus, ce type de maillage nécessite beaucoup moins de calcul lors de la phase de recherche de la position optimale des nœuds.
Cette étude préliminaire sera complétée dans des travaux de recherche futurs. Nous envisageons de traiter un ensemble de points importants. Nous analyserons l'influence du degré de la fonction spline sur l'erreur de prédiction puisque cette erreur sera codée ultérieurement pour réaliser la compensation de mouvement. Nous étudierons également les problèmes de codage induits par le maillage. L'ajustement des paramètres, tels que le seuil de détection de contours, la taille des mailles initiale et finale, en accordance avec le débit offert sera considéré. Le but final de ce travail serait de proposer une chaîne complète de codage où l'estimation et la compensation de mouvement seront basées sur le principe de maillage déformable.

Références

[1] ISO/IEC 13818. *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information*. 2000.

[2] ITU-T, Recommendation H.263 (version 2). *Video Coding for low Bitrate Communications*. February 1998.

[3] ISO/IEC 14496-2. *Coding of Audio-Visual Objects – Part 2: Visual*. 2001.

[4] F. Pereira and T. Ebrahimi. *The MPEG-4 Book*. IMSC Press, 2002.

[5] ISO/IEC 14496-10 and ITU-T Rec. H.264. *Advanced Video Coding*. 2003.

[6] Iain E. G. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*., Ed. John Wiley & Sons, 2003.

[7] Haibo Li, A. Lundmark, R. Forchheimer. Image sequence coding at very low bit rates: a review. *IEEE Transactions on image processing* 3(5) : 589–609, September 1994.

[8] J. R. Jain, A. K. Jain. Displacement measure and its application in interframe image coding. *IEEE Transactions on communication*, 29(12) : 1799–1806, December 1981.

[9] M. H. Chan, Y. B. Yu, A.G. Constantinides. Variable size block matching motion compensation with applications to video coding. *Communications, Speech and Vision, IEE Proceedings,* 137(4) : 205–212, August 1990.

[10] H. Brusewitz. Motion compensation with triangles. *Proc. 3rd Inter. Conf on 64 kbit Coding of moving*

Video, Rotterdam, Netherlands, September. 1990.

[11] Y. Nakaya, H. Harashima. An iterative motion estimation method using triangular patches for motion compensation. *Proc. SPIE Visual Comm. and Image* 546–557, Boston, MA, November. 1991.

[12] G. J. Sullivan, R. L. Baker. Motion compensation for video compression using grid interpolation. *ICASSP'*91, pages 2713–2716, Toronto, Canada, May 1991.

[13] Y. Nakaya, H. Harashima. Motion compensation based on spatial tansformation. *IEEE Transactions on circuits and systems for video technology*, 4(3) : 339–356, June 1994.

[14] Aria Nosratinia, M. T. Orchard. Optimal warping prediction for video coding. *ICASSP'*96, pages 1986–1989, Atlanta, May 1996.

[15] Aria Nosratinia. New Kernels for Fast Mesh-Based Motion Estimation. *IEEE Transactions on circuits and systems for video technology,* 11(1) : 40–51, January 2001.

[16] C. L. Huang, C. Y. Hsu. A new motion compensation method for image sequence coding using hierarchical grid interpolation. *IEEE Transactions on circuits and systems for video technology*, 4(1) : 13–52, February 1994.

[17] Y. Wang, O. Lee. Active mesh-A feature seeking and tracking image sequence representation scheme. *IEEE Transactions on image processing*, 3(5) : 610–624, September 1994.

[18] Gary J. Sullivan, Richard L. Baker. Efficient Quadtree Coding of Images and Video. *IEEE Transactions on image processing*, 3(3) :327–331, May 1994.

[19] Haibo Li, Astrid Lundmark, Robert Forchheimer. Image Sequence Coding at Very Low Bitrates: A Review. *IEEE Transactions on image processing*, 3(5) : 589-609, September 1994.

[20] C. Toklu, A. T. Erdem, M. I. Sezan, A. M. Tekalp. Tracking motion and intensity variations using hierarchical 2D mesh modelling for synthetic object transfiguration. *Graph. Models Image Process*, 58(6) : 553–573, November 1996.

[21] Y. Altunbask and A. M. Tekalp. Closed-form connectivity-Preserving solutions for motion compensation using 2-D meshes. *IEEE Transactions on image processing*, 6(9) : 1255–1269, September 1997.

[22] P. V. Beek, A. Tekalp, N. Zhang, I. Celasum, M. Xia. Hierarchical 2D-mesh representation tracking and compression for object video. *IEEE Transactions on image processing*, 9(2) : 353–369, Mars 1999.

[23] Ghassan Al-Regib, Yucel Altunbasak, Russell M. Mersereau. Hierarchical Motion Estimation With Content-Based Meshes. *IEEE Transactions on circuits ad systems for video technology,* 13(10) : 1000–1005, October 2003.

[24] C. De Boor. *A practical guide to splines*, revised edition, New York, Springer-verlag, 2001.

[25] N. Chihab, A. Zergaïnoh, P. Duhamel, J-P. Astruc. The influence of the non-uniform spline basis on the approximation signal. *EUSIPCO*'04, Vienna, Austria, September 2004.

[26] E. B. Becker, G. F. Gatey, J. T. Oden. *Finite Elements, An introduction*. Englewood Cliffs, NJ Prentice-Hall, 1982.

[27] K. J. Bathe. *Finite Element procedures in engineering analysis*. Englewood Cliffs, NJ Prentice-Hall, 1982.

[28] Y. Wang and O. Lee. Use of two-dimensional deformable mesh structures for video coding, part I-The synthesis problem: Mesh-based function approximation and mapping. *IEEE Transactions on circuits and systems for video technology*, 6(6) : 636–646, December 1996.

[29] Y. Wang, O. Lee, A. Vetro. Use of two-dimensional deformable mesh structures for video coding, part II-The analysis problem and a region based coder employing an active mesh representation. *IEEE Transactions on circuits and systems for Video Technology*, 6(6) : 647–659, December 1996.

LAR VIDEO : CODAGE VIDEO SANS PERTE A SCALABILITE SEMANTIQUE

Samir AMIR[1] Erwan FLÉCHER[1] Marie BABEL[2] Olivier DÉFORGES[2]

UMR CNRS 6164 IETR Groupe Image et Télédétection

INSA de Rennes
20, avenue des buttes de Coësmes, 35043 Rennes, FRANCE

[1]{samir, eflecher}@ens.insa-rennes.fr
[2]{mbabel, odeforge}@insa-rennes.fr

Concours Jeune Chercheur : Oui

Résumé

Baptisé "LAR Video", le schéma proposé dans cet article décrit un nouvel algorithme de compression vidéo sans perte avec scalabilité sémantique. Propre au codage vidéo, une étape d'estimation de mouvement est effectuée afin de produire une image d'erreur résiduelle. L'idée ici est d'appliquer sur cette erreur, une décomposition pyramidale issue d'une méthode de codage scalable d'images fixes, le LAR-APP. Résultantes d'une prédiction inter/intra niveau, les erreurs d'évaluation sont transmises progressivement. Le décodeur peut ainsi reconstruire les images de la séquence de façon scalable par niveau de résolution spatiale. Enfin au vu des résultats obtenus, nous pouvons affirmer que le schéma proposé offre en plus de la scalablité, des performances intéressantes en compression.

Mots clefs

Partitionnement quadtree, scalabilité sémantique, codage vidéo sans perte multirésolution.

1 Introduction

Le codage vidéo est utilisé dans de nombreuses applications qui nécessitent le développement d'outils de codage efficaces et rapides afin de compresser au maximum le flux vidéo tout en conservant une qualité visuelle optimale. Ces applications doivent pouvoir faire face aux caractéristiques hétérogènes et variantes dans le temps des réseaux de transmission. Pour résoudre ce problème, il est intéressant de disposer d'un flux vidéo scalable [1]. La scalabilité désigne l'aptitude d'un algorithme de compression à représenter une source hiérarchiquement sur plusieurs couches complémentaires [2]. Tout d'abord, une couche de base permet la reconstruction des informations émises avec une qualité minimum. D'autres niveaux de raffinement se succèdent ensuite. L'application des traitements non réversibles pour le codage vidéo avec pertes est nécessaire à l'obtention des forts taux de compression. En contre partie, ils peuvent détruire une partie essentielle de l'information ou faire apparaître des artefacts conduisant à une interprétation erronée de l'image. Ces dégradations ne sont pas tolérables pour certaines applications dédiées, notamment la télémédecine et la production cinématographique. Par exemple, une échographie en l'absence de compression, nécessite un temp de transfert inconcevable sur les réseaux de communication classiques. A ce jour, rares sont les solutions qui mutualisent à la fois compression sans perte efficace et option de scalabilité. Inspiré par cette constatation, un nouveau schéma d'un codage vidéo scalable basé sur une approche pyramidale de compression d'image fixe (LAR-APP) a été élaboré.
La méthode LAR (Locally Adaptive Resolution) fondée sur une représentation de l'image à taille de bloc variable définit une technique efficace de compression sans perte [3]. Son principe de codage a inspiré trois approches pyramidales de compression d'images fixes, le LAR-APP, l'Interleaved S+P et le RWHT+P [4, 5, 6]. Après avoir évoqué les particularités de ces trois méthodes (§2), nous décrivons le principe et les performances d'un schéma de codage vidéo sans perte qui se base sur le codage scalable de l'erreur de prédiction par le LAR-APP (§3). Enfin les résultats obtenus par le même schéma mais dans le cas de la non réversibilité sont présentés dans paragraphe 4.

2 Méthode LAR sans perte pour image fixe

2.1 Schéma général

Considérant l'image à compresser comme la superpostion d'une information globale et de la texture locale, il est alors possible de décider, à partir d'un même schéma d'encodage, de fournir une image compressée à bas débit (information globale), ou de l'enrichir en texture (ajout de l'image d'erreur). Le principe de l'encodage en deux couches est la base de la méthode LAR et offre naturellement au moins deux niveaux de progressivité (Fig. 1).
La première étape du codage LAR d'une image consiste

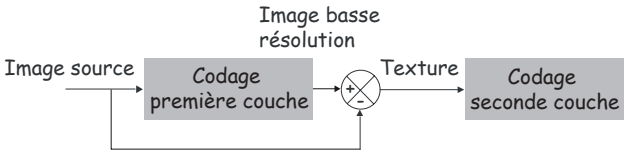


Figure 1 – *Schéma général du codeur LAR : deux couches de codage*

en l'obtention d'une image basse résolution (en termes de qualité visuelle). Cette phase est subordonnée à la création d'un partitionnement quadtree de l'image originale où la taille de chaque bloc est définie après évaluation de l'activité locale. Il en résulte une image de blocs de taille variable affectés de la luminance moyenne de son contenu. Ce pavage offre à la fois un lissage des zones homogènes mais aussi un bon rendu visuel des contours. Par la suite, nous définirons la partition quadtree par $QP^{[N_{max}..N_{min}]}$, où N_{max} et N_{min} représentent la taille maximale et minimale autorisée, des blocs carrés. En l'absence de seconde couche, ce codeur vise clairement les forts taux de compression. L'image reconstruite bas débit est visuellement acceptable en raison de l'adaptation au contenu, du support de l'image des blocs par le quadtree. Pour des débits plus élevés, la première phase est suivie d'une couche de raffinement qui, si aucune quantification n'est réalisée, permet une compression sans perte de l'information. Ainsi, la propriété de codage par scalabilité sémantique est rendue possible par la transmission progressive de l'information conditionnée par un partitionnement basé contenu. Soucieux d'enrichir ce principe de codage, trois appproches scalables reposant sur la méthode LAR ont ensuite vu le jour.

2.2 Approches pyramidales

Le LAR-APP [4], l'Interleaved S+P [5] et le RWHT+P [6] sont trois méthodes de codage d'images fixes développées au sein du laboratoire. Il s'agit d'algorithmes unifiés de compression avec ou sans perte combinant prédiction dans un contexte enrichi et scalabilité par niveau de résolutions et de détails. Ces trois méthodes reprennent la décomposition en deux couches du LAR et y apportent en plus la multirésolution. Une décomposition pyramidale est effectuée selon deux descentes successives, contraintes par une partition quadtree préalablement calculée. La première descente opère un raffinement uniquement sur les contours et modélise les zones homogènes par de grands blocs. Le principe de la division conditionelle est illustré par la figure 2 pour des tailles de blocs allant de 2×2 à 8×8.
Prenant en considération le partitionnement régulier quadtree du LAR, le codage scalable résultant de la première pyramide est renforcé par la scalabilité sémantique. En effet, à un niveau donné, la résolution adaptée au contexte de l'image accroît la qualité (sur critère visuelle) de l'imagette reconstruite. Si la première passe réalise la reconstruction de l'image basse résolution (LAR) par progression sémantique, la seconde permet de récupérer la texture locale. Ainsi à chaque niveau de la pyramide, les blocs ignorés par la première descente sont décomposés par la couche de raffinement.

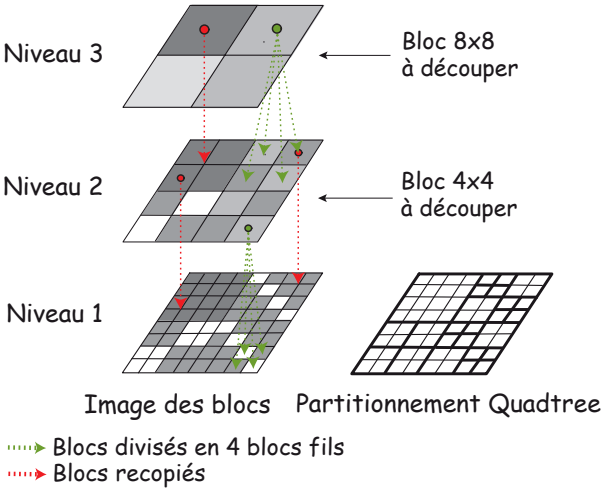


Figure 2 – *Décomposition de la pyramide de l'image des blocs conditionnellement au pavage quadtree* $QP^{[8..2]}$

La modélisation de contexte est devenue un facteur clé d'efficacité des algorithmes de compression sans perte [7]. En effet, l'entropie brute d'une image peut être réduite lorsque des classes de symboles, propres à différentes lois de probabilité, peuvent être isolées. L'observation du comportement de nos algorithmes vis-à-vis de l'entropie engendrée a permis la séparation des lois de probabilité en deux niveaux : la taille de blocs et le niveau de décomposition dans chacune des deux passes. De ce fait, grâce à la nature même du schéma de codage, une modélisation de contexte implicite est réalisée.
A ce stade, il est raisonnable d'estimer que nos trois approches pyramidales reposent sur une même base algorithmique opérant selon deux descentes successives. Si la première décrit une pyramide d'imagettes LAR dont la qualité évolue conformément au principe de scalabilité sémantique, la seconde offre un enrichissement progressif en texture. Bénéfique d'un point de vue entropique, la modélisation de contexte implicite est un autre aspect largement exploité par ces trois méthodes.

2.3 Méthodes de décomposition

Après avoir brièvement présenté les similitudes de ces trois approches, il est maintenant intéressant d'en évaluer les particularités. Elles sont inhérentes au processus de décomposition d'un bloc en quatres fils dans le contexte d'un codage réversible. La prédiction MICD scalable du LAR-APP repose pour sa part sur une succession de passes visant à tirer profit des informations contextuelles présentes aussi bien dans le domaine causale (prédiction intra-niveau) que dans l'image sous-échantillonnée (prédiction inter-niveau). Contrairement au

LAR-APP, l'Interleaved S+P et le RWHT+P ne se restreignent pas au domaine spatial mais oeuvrent dans des domaines transformés plus appropriés à l'obtention de taux de compression élevés. Dans son principe, l'Interleaved S+P est associé à une implantation particulière de la transformée en S. Plus précisément, son espace de prédiction résulte de l'application de la transformée en S 1D sur deux vecteurs constitués chacun des deux pixels diagonalement adjacent d'un bloc 2×2 donné. L'originalité de cet algorithme tient dans le fait qu'il est possible de prédire les coefficients de la transformée au moyen de deux pyramides entrelacées. La troisième et dernière méthode se nomme RWHT+P, elle met en oeuvre une version réversible de la transformée Walsh Hadamard appliquée à des blocs 2×2. L'Interleaved S+P et le RWHT+P sont deux méthodes pyramidales extrêmement efficaces de compression sans perte, surpassant largement le LAR-APP et les méthodes de l'état de l'art telles que CALIC [8] et S+P [7]. Malgré cela, c'est le LAR-APP qui a été implanté dans un premier temps dans notre codeur vidéo sans perte en raison de sa simplicité de mise en oeuvre. L'une de nos perspectives proches vise à le substituer par l'Interleaved S+P et le RWHT+P afin d'accroître les performances de codage.

3 Principe du codage vidéo LAR sans perte

L'estimation et la compensation de mouvement sont deux étapes majeures permettant de tirer profit de la redondance temporelle naturellement présente dans une séquence vidéo. Prédire l'image courante par une référence compensée en mouvement est le principe régissant généralemenent le codage video. La première étape consiste à diviser l'image en blocs réguliers (MPEG-4) ou selon un découpage irrégulier (H.264 [9]). Au moyen d'un critère de distortion, l'estimateur de mouvement recherche pour chaque bloc, le meilleur appariement dans une image de référence. La position relative entre les deux blocs est décrite par un vecteur de mouvement qui est envoyé au décodeur. A ce stade, l'erreur résiduelle résultante de la différence entre l'image compensée et l'image à coder doit être efficacement transmise. Pour y parvenir, une transformée est appliquée sur cette erreur afin de profiter de la redondance spatiale présente. Classiquement les moyens utilisés par les standards permettent une forte compression mais n'offrent pas de solution performante au codage sans perte.
La construction d'un schéma de codage vidéo par l'approche pyramidale prédictive (§2) vise donc à répondre à deux attentes fondamentales. La première consiste à proposer une méthode nouvelle scalable par niveau en résolution et en qualité. Le second objectif est de proposer à l'utilisateur une solution simple et unifiée de compression avec ou sans perte. Le LAR-APP appliqué à l'erreur résiduelle en est une réponse : un seul algorithme est utilisé à des fins de codage réversible ou non (Fig. 3).

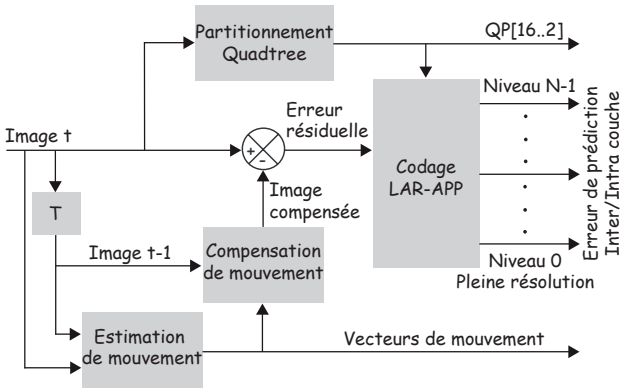


Figure 3 – *Schéma général du codeur vidéo LAR sans perte avec compression de l'erreur résiduelle sur N niveaux par le LAR-APP*

3.1 Estimation de mouvement

La mise en correspondance de blocs est une solution simple et efficace d'estimation de mouvement. Ce procédé approxime le mouvement apparent entre deux images par un modèle translationnel propre à chaque bloc issu du découpage de l'image courante. Une recherche exhaustive du meilleur appariement donne la solution optimale (au sens d'un critère de ressemblance) mais s'avére demandeure d'une grande puissance de calcul. De la famille des techniques de recherche par zone, l'estimateur de mouvement EPZS (Enhanced Predictive Zonal Search) [10] surmonte ce problème en restreignant l'espace de recherche à partir d'une connaissance à priori du déplacement. La prédiction de déplacement d'un bloc donné est issue du mouvement évalué dans son voisinage causal mais aussi dans une image de référence. Même si le codage de l'erreur résiduelle n'a pas encore été détaillé, il peut être intéressant d'analyser l'influence de la taille des blocs sur l'entropie de cette erreur. Le graphe ci-dessous (Fig. 4) illustre l'ascendance d'une grille régulière de blocs 16×16, 8×8 et 4×4 sur le codage de l'erreur par l'approche pyramidale LAR-APP. Les entropies mentionnées résultent du codage sans perte des images de la séquence Foreman.
Engendrée par une prédiction plus fine, la diminution de l'entropie de l'erreur s'accompagne naturellement d'une augmentation du coût des vecteurs de mouvement. Dans un contexte de codage à bas et moyen débit, des solutions [11] comparables à celle adoptée par H.264 [9] s'avèrent appropriées. Elles reposent généralement sur un partitionnement optimal de l'image au sens du mouvement en considérant simultanément le coût des vecteurs de mouvement et celui de l'erreur résiduelle. Or appliqué au codage réversible, il est évident que le gain apporté par ces solutions serait masqué par la dynamique entropique de l'erreur. Ainsi, la solution retenue dans notre codeur est d'estimer le mouvement sur une grille régulière de taille de blocs 8×8.

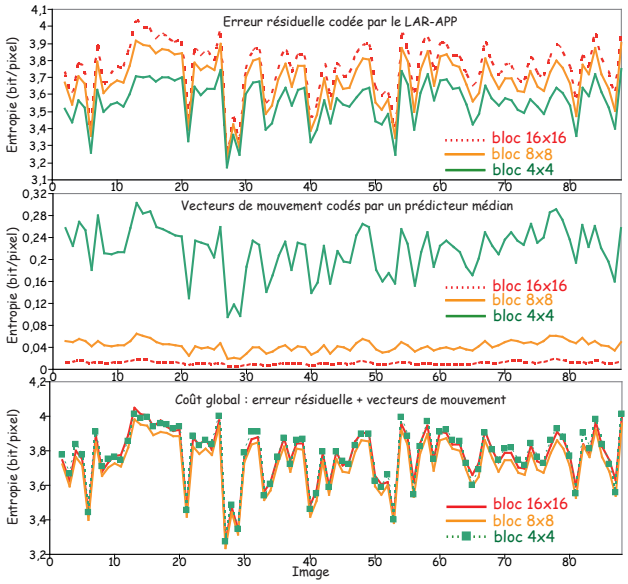


Figure 4 – *Influence de la taille du support du modèle de mouvement sur l'erreur résiduelle codée par le LAR-APP sans perte (séquence Foreman)*

3.2 Codage de l'erreur résiduelle

Afin de tirer profit de la scalabilité sémantique et de la modélisation de contexte, la décomposition (sur critère spatial) est ici appliquée sur des images directement extraites de la séquence. Le partitionnement de l'image ainsi que la décomposition en flux scalable de l'erreur résiduelle par le LAR-APP peuvent être modélisés par un schéma bloc (Fig. 5). Conformément à la méthode décrite dans le paragraphe 2, la compression sans perte de l'erreur résiduelle consiste à établir un codage pyramidal total. Chaque bloc du pavage LAR est décomposé jusqu'à l'obtention de la pleine résolution, selon le principe de décomposition pyramidale et de prédiction dans un contexte enrichi. Le codage par scalabilité sémantique impose que l'information liée aux petits blocs soit dissociée de celle issue du reste de l'image, de telle sorte que la première passe décrive exactement les contours forts. Ainsi le schéma général du LAR-APP définit à chaque étage de la pyramide, un niveau de décomposition minimal relatif à la taille des blocs. Associé à une partition $QP^{[16..2]}$, le schéma (Fig. 5) met en lumière ce principe dans le cas où pour un niveau n de la pyramide, la taille minimale de décomposition autorisée est 2^n. Par conséquent, le niveau 0 de la pyramide est entièrement traité lors de la seconde descente. En raison de la similarité du processus d'évaluation au niveau du récepteur, seules les erreurs de prédictions inter/intra couches transitent conformément au principe de l'emboîtement de flux.

Au décodeur, le processus de reconstruction scalable des images opère en deux temps (Fig. 6). A un niveau donné de la pyramide, l'erreur résiduelle est tout d'abord décodée puis sommée à une image compensée de même dimension. Pour cela une pyramide d'images compensées est élaborée sur le même principe que celle de l'erreur résiduelle. Naturellement, le décodage par couche scalable nécessite une étape de prédiction/compensation de mouvement progressive. Même si pour l'instant cette phase n'est pas encore réalisée limitant ainsi l'entièrement progressivité de notre schéma de codage, des applications peuvent lui être destinées. Dans le cadre d'une compression réversible, la pyramide se doit d'être complète afin de retrouver après l'opération de décodage, une image originale compressée sans perte. Ainsi l'utilisation des imagettes décodées comme moyen de prévisualisation des images de la séquence à différents niveaux de résolution est une des fonctionalités offerte par notre décodeur.

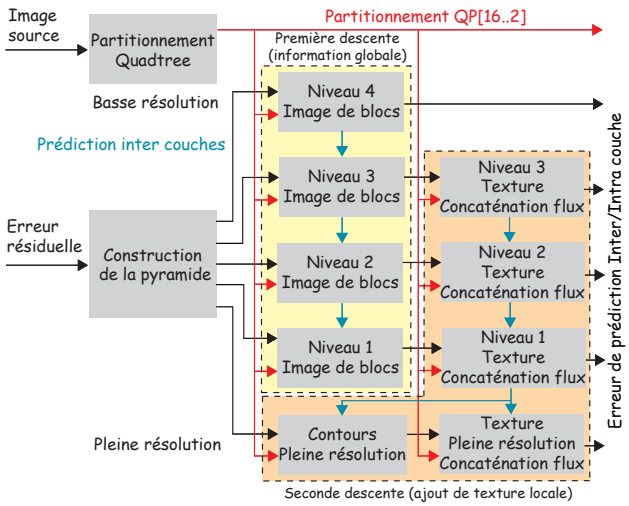


Figure 5 – *Codage scalable de l'erreur résiduelle par l'approche pyramidale prédictive LAR-APP*

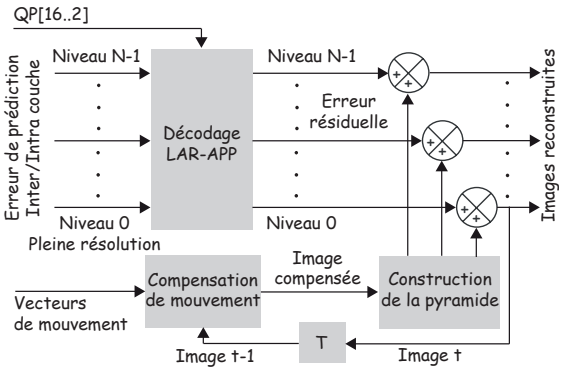
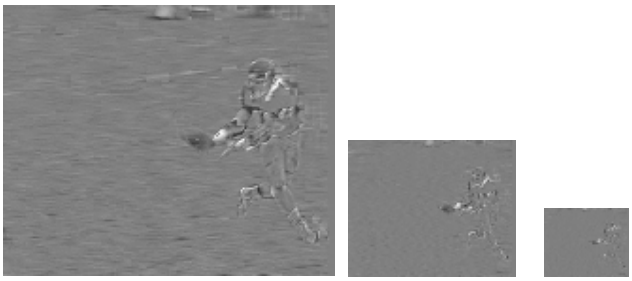


Figure 6 – *Schéma général du décodeur vidéo LAR sans perte avec décompression de l'erreur résiduelle sur N niveaux par le LAR-APP*

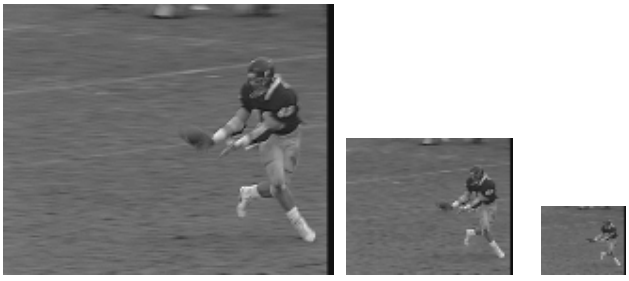
Les figures 7 et 8 présentent les imagettes sucessivement obtenues lors du décodage d'une image de la séquence Football par l'approche pyramidale prédictive. Elles décrivent respectivement la pyramide de l'erreur résiduelle, ainsi que les images reconstruites progressive-

ment selon le processus précédemment décrit.



(a) (b) (c)

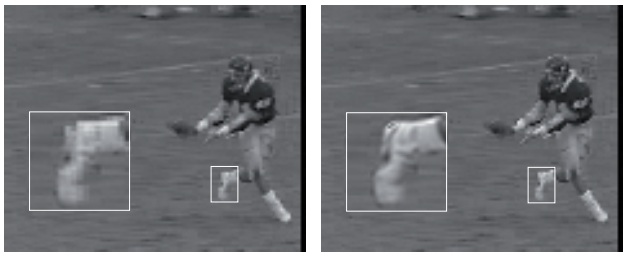
Figure 7 – *Images d'erreur obtenues lors de la décomposition pyramidale. (a) Pleine résolution, sans perte à 3.80 bpp (b) niveau 1 à 1.01 bpp (c) niveau 2 à 0.37 bpp*



(a) (b) (c)

Figure 8 – *Reconstruction progressive de l'image source jusqu'au sans perte à partir de la pyramide de l'erreur résiduelle. (a) Pleine résolution, sans perte (b) niveau 1 (c) niveau 2*

La figure 9 montre le résultat de la reconstruction de l'image source à des niveaux intermédiaires en qualité. Comme expliqué précédemment, elle résulte de l'addition de l'image compensée pleine échelle et dans le cas présent d'une erreur intermédiaire. Dans le premier cas, l'erreur est obtenue par interpolation de l'image issue de la première passe (image LAR du niveau 1). Dans le second, la pleine résolution de l'erreur est atteinte en décomposant uniquement les contours (blocs 2×2).
Après avoir présenté notre codeur vidéo ainsi que certaines images intermédiaires, il est maintenant intéressant d'en évaluer l'efficacité. Le tableau 1 synthétise les résultats obtenus lors du codage de diverses séquences tests par le LAR-APP sans perte. Les performances de l'approche pyramidale sur des images directement extraites de la séquence (APP Intra) et celles résultantes de l'erreur de compensation de mouvement (APP Inter) y sont comparées selon un critère entropique. Au vu des résultats, il est aisé de conclure que le codage en mode Inter surpassent l'APP Intra. Les entropies obtenues ici laissent présager du gain réalisé par l'implantation future de l'Interleaved S+P et du RWHT+P.



(a) (b)

Figure 9 – *Reconstruction intermédiaire au moyen de l'erreur issue de la première passe (a) erreur première passe + interpolation (b) erreur première passe + décomposition des contours*

	Entropie (bpp) - Format CIF (352×288)			
Séquence	APP Intra	Erreur résiduelle	APP Inter	Vecteurs
Foreman	4.30	**3.66**	3.69	0.042
Mother	3.60	2.91	**2.85**	0.035
Mobile	5.86	5.17	**5.04**	0.030
Football	4.09	4.33	**3.81**	0.081
Tempete	5.35	4.68	**4.51**	0.033
CoastGuard	5.36	4.66	**4.40**	0.025

Tableau 1 – *Performances en mode Intra et Inter de l'approche LAR-APP sans perte sur différentes séquences tests*

La figure 10 présente une description plus fine des résultats obtenus précédemment pour les séquences Football et MotherandDaughter. Elle illustre à la fois l'évolution temporelle de l'entropie brute de l'erreur résiduelle et celle engendrée par l'approche pyramidale sans perte. Tout en offrant une scalabilité par niveau, le LAR-APP permet une compression efficace.

4 Codage vidéo LAR avec pertes

De part sa nature non réversible, l'opération de quantification est naturellement bannie de notre schéma de codage sans perte. Par contre lorsque des dégradations sont tolérées, cet outil s'avère efficace afin d'obtenir des taux de compression importants. Classiquement les techniques de compression fondées sur l'optimisation débit/distorsion tentent de trouver le meilleur compromis entre coût de codage et détérioration de l'image d'un point de vue PSNR. Or des expérimentations ont montrées que le système visuel s'avère beaucoup moins sensible à des variations de luminance dans des zones de type frontière que dans des zones uniformes. Rappelons que notre partitionnement quadtree est piloté par l'activité locale, de grands blocs représentent des régions uniformes alors que des blocs de petits taille modèlisent les contours. Ainsi une quantification conditionnée par la surface des blocs, est associée à notre schéma de codage vidéo. En accord avec la sensibilité du système visuel, la quantification associée aux petits blocs est importante alors que celle attribuée aux zones uni-

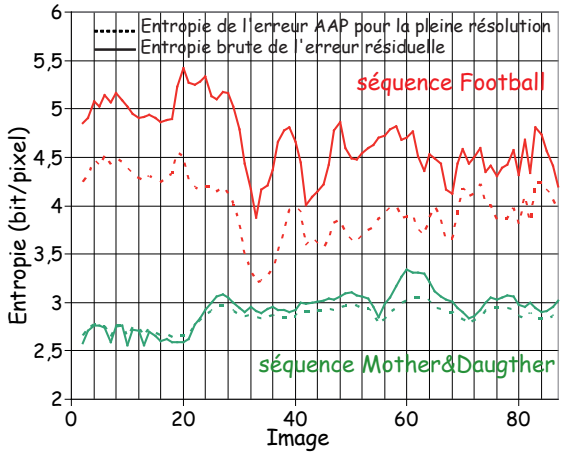
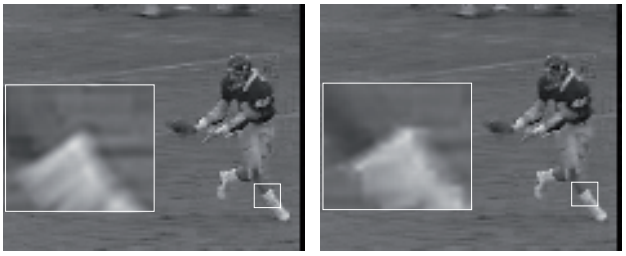


Figure 10 – *Entropie résultante du codage par le LAR-APP sans perte comparée à l'entropie brute de l'erreur sur les séquences Football et MotherandDaughter*

formes est faible.
Le codage de l'erreur résiduelle par la première passe du LAR-APP associée à une quantification adaptative, offre une compression accrue tout en maîtrisant les dégradations engendrées. La figure 11 illustre ce principe sur une image pleine résolution (pr) de la séquence Football. Les pas de quantifications adoptés sont définis par $Q^T = [q_{16}, q_8, q_4, q_2, q_{pr}]^T = [2, 4, 8, 16, 32]^T$



(a) (b)

Figure 11 – *(a) Image reconstruite à 37.12 dB par une erreur non quantifiée à 0.85 bpp. (b) Image reconstruite à 35.88 dB par une erreur quantifiée à 0.10 bpp*

5 Conclusion

Le schéma de codage vidéo scalable présenté est une solution efficace de compression avec ou sans perte. Nous avons montré sur plusieurs exemples que les performances du schéma proposé, s'avèrent encourageantes pour la suite de nos travaux. L'objectif à court terme est de substituer le LAR-APP par l'Interleaved S+P et le RWHT+P, qui comme nous l'avons dit, reposent sur le même principe mais accroîent l'efficacité de compression. Appliquer un codage par une approche pyramidale sur l'erreur résiduelle n'est pas suffisant pour doter notre schéma d'une entière progressivité. La deuxième perspective est donc de rendre la méthode intégralement scalable en incluant entre autre, un codage hiérarchique des vecteurs de mouvement.

Références

[1] G. Marquant. *Représentation par maillage adaptatif déformable pour la manipulation et communication d'objets vidéo*. Thèse de doctorat, Université de Rennes 1, Rennes, Décembre 2000.

[2] N. Cammas. *Codage vidéo scalable par maillages et ondelettes 2D*. Thèse de doctorat, Université de Rennes 1, Rennes, Novembre 2004.

[3] O. Déforges et J. Ronsin. Region of Interest Coding for Low Bit-Rate Image Transmission. Dans *Proc. International Conference on Multimedia and Expo ICME'2000*, volume 1, pages 107 – 110, 30 July-2 Aug 2000.

[4] M. Babel, O. Déforges, et J. Ronsin. Lossless and Lossy Minimal Redundancy Pyramidal Decomposition for Scalable Image Compression Technique. Dans *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'03*, volume III, pages 249–252, Hong Kong, April 6-10 2003. Conference cancelled - Invited paper in ICME 2003.

[5] M. Babel, O. Déforges, et J. Ronsin. Interleaved S+P Pyramidal Decomposition with Refined Prediction Model. Dans *IEEE International Conference on Image Processing, ICIP'05*, volume 2, pages 750–753, Genova,Italy, September 2005.

[6] O. Déforges, M. Babel, et J. Motsch. The RWHT+P for an improved lossless multiresolution coding. Dans *EUropean SIgnal Processing COnference, EUSIPCO'06*, To be published, September 2006.

[7] A. Said et W. A. Pearlman. An image multiresolution representation for lossless and lossy image compression. *IEEE Trans. on Image Processing*, 5 :1303–1310, September 1996.

[8] X. Wu, N. Memon, et K. Sayood. A Context-based, Adaptive, Lossless/Nearly-Lossless Coding Scheme for Continuous-Tone Images (CALIC), 1995.

[9] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, et T. Wedi. Video Coding with H.264/AVC : Tools, Performance, and Complexity. *IEEE Circuits and Systems Magazine*, 4(1) :7–28, First Quarter 2004.

[10] A. M. Tourapis. Enhanced predictive zonal search for single and multiple frame motion estimation. Dans C.-C. J. Kuo, éditeur, *Proc. SPIE Vol. 4671, p. 1069-1079, Visual Communications and Image Processing 2002, C.-C. Jay Kuo ; Ed.*, pages 1069–1079, Janvier 2002.

[11] J. Zhang, M.O. Ahmad, et M.N.S. Swamy. Quadtree Structured Region-Wise Motion Compensation for Video Compression. *IEEE Transactions On Circuits And Systems For Video Technology*, 9(5) :808–822, August 1999.

Classification IN/OUT/OFF d'un intervenant dans un document audiovisuel

J. Philippeau J. Pinquier P. Joly

IRIT équipe Structuration Analyse MOdélisaion de la Vidéo et de l'Audio
Université Paul Sabatier
UMR 5505 CNRS - INPT, 118 Route de Narbonne, 31062 Toulouse Cedex 9
{philippe, pinquier, joly}@irit.fr

Résumé

Ce papier s'inscrit dans le cadre de l'indexation de documents audiovisuels. Il traite de la définition d'un nouveau descripteur : l'intervenant. Nos travaux ont porté sur la caractérisation de sa localisation, c'est-à-dire sa recherche dans une séquence audiovisuelle et sa classification en 3 catégories : IN, OUT ou OFF. A partir de l'étude de différents outils d'analyse des modes audio et vidéo, nous définissons un jeu de descripteurs qu'il est possible de renseigner automatiquement, potentiellement influents pour décider de la classe de la localisation de l'intervenant. Cette décision est effectuée à l'aide d'une modélisation des transitions d'une classe à une autre.

Mots clefs

intervenant, indexation audiovisuelle, descripteur multimodal, classification IN OUT OFF, modélisation multimédia.

1 Introduction

De nombreux travaux ont été entrepris dans le domaine de la caractérisation automatique de contenus audiovisuels grâce à des descripteurs à la fois audio et vidéo, mais les orientations choisies l'ont été dans le but d'améliorer par la vidéo les performances de systèmes se basant exclusivement sur l'audio (en Reconnaissance Automatique de la Parole [1] par exemple) ou inversement [2].

Nous avons mené notre étude dans le but de concevoir un nouveau descripteur pertinent pour caractériser un document audiovisuel en vue de son indexation. Considérons un intervenant, c'est-à-dire tout individu qui intervient par la parole et est localisable par celle-ci, dans une séquence audiovisuelle. Notre préoccupation est de savoir si, à un instant donné, sans connaissance *a priori* sur le type de document traité, un intervenant est visible ou s'il ne l'est pas. Jusqu'à présent, les travaux s'apparentant le plus à la classification d'intervenants considéraient le problème ainsi : un locuteur est de classe IN lorsqu'une personne est détectée à l'écran pendant la locution, sinon il est OUT. Toutefois, cette classification un peu arbitraire ne prenait pas en considération l'activité visible de parole à part entière : la personne détectée à l'écran n'est pas forcement celle qui parle.

Notre objectif est de préciser cette classification en considérant cet aspect visible de la locution et en définissant de nouvelles classes d'intervenants :

- la personne qui parle est visible, elle est de classe IN,
- la personne qui parle n'est pas visible, mais elle a déjà été filmée ou le sera durant son élocution, elle est de classe OUT.
- la personne qui parle n'est jamais visible pendant toute son intervention, elle est de classe OFF.

La pertinence du choix de l'intervenant comme contenu descriptif dans un document audiovisuel prend sens dès que l'on considère l'apport conjoint des modalités audio et vidéo.

Après avoir exhibé les descripteurs vidéo et audio que nous avons choisis pour caractériser un intervenant, nous présenterons les expériences menées sur ceux-ci et montrerons la manière dont nous les avons conjointement utilisés pour créer un descripteur audiovisuel à part entière. Nous ferons enfin une comparaison entre notre proposition et la classification traditionnelle IN/OUT.

2 Contexte applicatif

2.1 Choix du corpus

Dans un soucis de confort et de généricité, nous avons choisi d'étudier des séquences répertoriées pour la campagne d'évaluation TRECVID2004 [3]. Nous avons également porté notre attention sur une émission du jeu télévisuel français « Pyramide ». La résolution de ces vidéos (352*264 px à 29.97 fps) est jugée suffisante pour les traitements à opérer. De plus, la qualité relativement mauvaise des images due à l'encodage mpeg gage de la généricité de nos travaux sur la qualité des documents. Pour finir, la parole y est omniprésente et n'est pas interrompue, nous permettant une analyse mono-locuteur du signal de parole.

2.2 Segment audiovisuel

Nous définissons un segment audiovisuel comme une séquence pendant laquelle une classe d'intervenant reste stable. Un segment sera donc délimité par les frontières suivantes : un changement de locuteur, un changement de plan, une combinaison des deux ou un long silence. Afin de réaliser ces segmentations, nous nous sommes appuyés sur les travaux de [4] pour trouver les zones de parole, et sur ceux de [5] pour détecter les changements de plans. Les taux de reconnaissance (ou « accuracy ») listés dans ce papier ont été obtenus sur des segments extraits à la fois de TRECVID2004 et de Pyramide.

3 Point de vue vidéo

3.1 Détection du visage

De nombreux travaux ont été menés sur la détection automatique de visages (cités dans [6]) et s'appuient sur des techniques variées : sur des caractéristiques « bas-niveau » (comme la couleur, la forme ou la texture), sur la détection de caractéristiques faciales (comme les yeux, le nez ou la bouche), ou encore grâce à des approches statistiques. C'est un détecteur provenant de cette dernière catégorie que nous avons utilisé : le détecteur de visages de Violat et Jones [7]. L'analyse se fait donc image par image sur la totalité de la durée du segment vidéo considéré, pendant que de la parole est détectée.

Nous décidons qu'il y a effectivement présence d'un visage lorsqu'il a été détecté dans au moins 7 images sur une fenêtre temporelle de 11 images. Ces deux paramètres sont des valeurs optimales utilisées lors du processus de détection de personnes basé sur le détecteur de Viola et Jones et utilisé dans [6].

Pour savoir si un visage est le même d'une image à l'autre, nous avons construit une fenêtre de recherche autour de chaque visage détecté. Si 2 visages localisés dans la même fenêtre ont des dimensions suffisamment proches (plus ou moins 10%), nous considérons qu'il s'agit de la même personne.

Ce détecteur « oubliant » régulièrement un voir plusieurs visages sur la totalité du segment, nous avons du générer visuellement les visages manquants. Nous avons choisi de générer un visage V_0 non détecté par interpolation linéaire des coordonnées des deux visages temporellement les plus proches de V_0, à savoir V_1 (détecté avant V_0) et V_2 (détecté après V_0). Cette méthode nous a donné de bons résultats (non quantifiables mais visuellement corrects). En ce qui concerne les fausses détections, elles sont relativement rares et partiellement évincées grâce à l'algorithme du calcul du score d'activité explicité section 3.2.

La présence d'un visage apparaissant pendant tout un segment dans le champ de la caméra constitue notre premier descripteur fiable. Grâce à celui-ci, nous obtenons un taux de précision de détection des intervenants IN de 90,2%.

3.2 Analyse de l'activité labiale

Nous nous sommes ensuite penchés sur le problème de la localisation des lèvres pour pouvoir quantifier l'activité de celles-ci. Les divers travaux existant s'appuient sur des dispositifs intrusifs et/ou sur des images propres (bonne définition et fréquence élevée) dans des conditions de laboratoire (prises de vue frontale avec illumination constante) [8]. Ces méthodes étant impossibles à mettre en œuvre dans le cadre de nos travaux, nous nous sommes donc restreints à les localiser approximativement, c'est-à-dire dans le tiers bas du visage, entre les 2 et 4 cinquièmes de la largeur du visage (cf. Figure 1).
Outre le fait que cette localisation soit extrêmement facile et rapide à mettre en œuvre, elle permet de toujours cerner les lèvres, que le visage soit de face ou de profil.

Figure 1 – *Exemple de résultats de détections du visage, du corps et des lèvres.*

Pour quantifier l'activité labiale, nous avons procédé par paires d'images pour ensuite obtenir un résultat global. Nous avons donc considéré deux images successives I1 et I2 contenant le visage d'une même personne. Après localisation des lèvres, représentées par les régions L(I1) et L(I2), nous avons construit une fenêtre autour de L(I1) et déplacé L(I2) dans cette zone de recherche. L'appariement ainsi que la valeur représentant la différence de pixels entre L(I1) et L(I2) ont tous deux été obtenus en minimisant l'Erreur Quadratique Moyenne (EQM), normalisée par la taille de L(I2), sur le canal de luminance de l'espace HLS. La moyenne des EQM ainsi calculées sur l'ensemble du segment vidéo considéré nous donne une valeur quantitative de l'activité labiale d'un personnage. Nous l'avons appelé Taux d'activité Labial (TAL).

Nous avons ensuite considéré que l'activité de parole concernait une zone plus large que celle des lèvres, car une personne bouge généralement corps et visage lorsqu'elle parle. Ainsi, pour le :

– Taux d'Activité du Visage (TAV), nous avons opéré l'appariement en privilégiant la ressemblance au niveau des lèvres, ceci en appliquant un masque de poids sur le visage, donnant une valeur deux fois plus élevée aux pixels de la région des lèvres que sur le reste du visage,

– Taux d'Activité du Corps (TAC), nous avons considéré un rectangle de largeur deux fois la taille du visage et de même hauteur que le visage, positionné sous celui-ci.

De ce fait, une hiérarchie entre les descripteurs s'organise d'elle même, plaçant dans l'ordre décroissant d'importance le TAL, le TAV, puis le TAC. Pour comparer l'activité labiale de deux personnages *i* et *j* filmés dans une même séquence ou dans deux séquences successives (afin de savoir lequel des deux est susceptible d'être le locuteur), nous avons défini un score d'activité basé sur une somme pondérée de ces trois taux.

Le taux de précision d'une décision basée sur ce score appliqué à l'identification du locuteur entre deux personnages au sein d'un même segment ou sur 2 segments consécutifs est de 95,7%.

4 Point de vue audio

4.1 Soustraction cepstrale

La soustraction cepstrale est couramment utilisée pour débruiter le signal de parole du bruit de la source d'enregistrement (micro, canal téléphonique...) [9]. C'est ce bruit du canal qui nous intéresse.

Nous décrivons ici le processus usuel d'analyse cepstrale ([10]) : Pour chaque trame de signal de parole, une préaccentuation des aigus et un fenêtrage de Hamming sont effectués. Les énergies sont calculées dans 24 filtres après application du module de la Transformée de Fourier. On répartit alors ces canaux selon l'échelle Mel pour tenir compte de la perception humaine. L'analyse de chaque trame donne un vecteur d'observations de 26 paramètres, comprenant l'énergie du signal et sa dérivée, ainsi que 12 coefficients cepstraux (ou MFCC) et les dérivées respectives.

Pour assurer une relative indépendance vis-à-vis du canal de transmission, on soustrait habituellement à chaque coefficient cepstral la moyenne des 12 MFCC. Nous avons décidé d'exploiter les informations contenues dans l'évolution des MFCC entre classes vocales au vu des résultats obtenus durant l'étude de la soustraction cepstrale.

4.2 Réflexions sur le comportement des descripteurs

Il est tout d'abord nécessaire d'énumérer les différentes configurations possibles de transitions entre classes et le comportement attendu des descripteurs dans chaque cas, comme l'illustre la figure 2 :

1. *Les descripteurs doivent caractériser une stabilité de l'environnement audio dans le cas d'une transition due à un changement de plan si le même intervenant continue de parler* (**groupe A**). Le cas particulier de la transition entre 2 intervenants en voix IN sur un changement de locuteur sans changement de plan est aussi à prendre en compte dans ce groupe ci.
2. *Ils doivent aussi témoigner d'un changement d'environnement audio dans le cas de transition entre locuteurs sensés évoluer dans des cadres d'enregistrements acoustiques différents* (**groupe B**).
3. Il est à noter que *des transitions particulières ne se produisent jamais* (**groupe C**). Il s'agit d'un passage de voix OFF à voix IN ou OUT (et réciproquement) sans changement de locuteur. Cela insinuerait que l'intervenant en voix OFF a été ou sera dans le champ de la camera, ce qui contredit sa définition.
4. En ce qui concerne le reste des cas envisageables (**groupe D**), *nous ne pouvons pas tirer d'informations en considérant uniquement ces descripteurs*. En effet, une transition entre deux locuteurs OFF, par exemple, peut se passer dans le même environnement sonore (deux commentateurs sportifs par exemple) ou non (une voix OFF introduit un reporter sans liaison vidéo avec celui-ci).

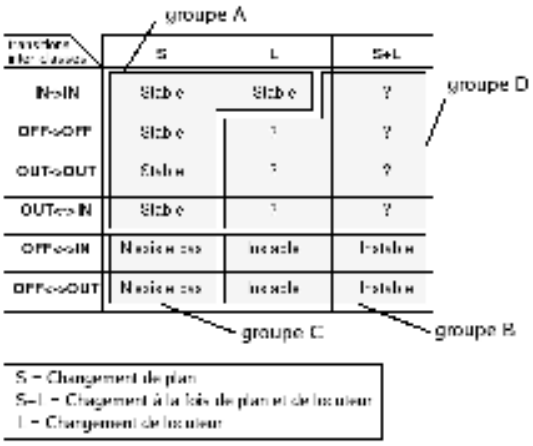


Figure 2 – *Comportement souhaité des MFCC utilisés pour caractériser les transitions entre classes d'intervenants.*

4.3 Expériences et résultats

Après avoir vainement tenté de caractériser dans quel environnement sonore évoluait le locuteur, nous nous sommes penchés sur la caractérisation des changements d'environnements entre deux segments adjacents s^1 et s^2.

Soit un segment s^k d'une seconde échantillonné à 16kHz, l'analyse cepstrale étant calculée sur des fenêtres de 256 points avec un recouvrement sur la moitié. Nous obtenons 125 vecteurs $y_i = (y_{i,1} \cdots y_{i,12})$ de dimension 12 (autant que de MFCC), avec $i \in \{0, \cdots, 125\}$ l'indice de vecteur. Si nous effectuons l'analyse sur les 2 dernières secondes du segment s^1 et sur les 2 premières du segment s^2, cela nous donne respectivement deux collections de vecteurs $(y_1 \ldots y_{250})$ et $(y_{251} \ldots y_{500})$ (cf. Figure 3).

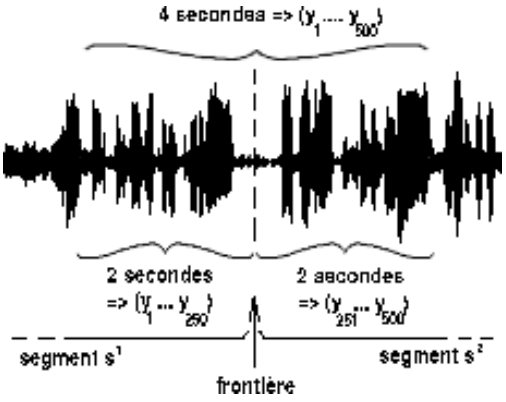


Figure 3 – *Récupération de la collection de vecteurs sur s^1 et s^2.*

Si nous voulons caractériser un changement de comportement des MFCC à la frontière des segments s^1 et s^2, nous allons supposer que les :
– $(y_1 \dots y_{250})$ suivent une loi Normale $N(M^1, \Sigma^1)$,
– $(y_{251} \dots y_{500})$ suivent une loi Normale $N(M^2, \Sigma^2)$,
– $(y_1 \dots y_{500})$ suivent une loi Normale $N(M^3, \Sigma^3)$.

Si nous considérons que toutes les composantes des vecteurs sont indépendantes [11], il est possible de faire les deux hypothèses suivantes :

– hypothèse (h_1) : *il y a un changement d'environnement sonore entre s^1 et s^2*. Ceci se traduit par :

$$P(y_1 \dots y_{500}/h_1) = P(y_{1\dots}y_{250}/N(M^1, \Sigma^1)) \cdot P(y_{251\dots}y_{500}/N(M^2, \Sigma^2)) \quad (1)$$

– hypothèse (h_2) : *l'environnement sonore de s^1 est le même que celui de s^2*. Ceci s'exprime ainsi :

$$P(y_1 \dots y_{500}/h_2) = \prod_{i=1}^{500} P(y_i/N(M^3, \Sigma^3)) \quad (2)$$

Le test d'hypothèses est basé sur le rapport de vraisemblance :

$$\Delta(s^1, s^2) = \frac{P(y_1 \cdots y_{500}/h_2)}{P(y_1 \cdots y_{500}/h_1)} \quad (3)$$

En fixant un seuil θ à la forme logarithmique de ce test, il est possible alors de prendre une décision en faveur d'une des deux hypothèses (h1) ou (h2).

Nous avons remarqué qu'un seuil expérimental de $-68.5 * 10^{-3}$ fonctionne pour 92.8% des cas étudiés.

5 Mise en œuvre conjointe des descripteurs

5.1 Présentation des résultats

Nous avons décidé d'utiliser les descripteurs audio et vidéo suivants :

– **Presence**$_t \in \{yes, no\}$: présence ou absence de personnage durant le segment t (section 3.1).
– $\Phi_{t,t+1}$: score d'activité comparé entre le deux personnages ayant le TAL le plus élevé, présents dans deux segments consécutifs t et t+1 (section 3.2).
– $\Delta_{t,t+1} \in \{yes, no\}$: stabilité ou instabilité de l'environnement acoustique au passage du segment t au segment t+1 (section 4.3).
– **Transition** $\in$ {S,L,S+L} : frontières audio et/ou vidéo. S pour une détection de changement de plan, L pour une détection de changement de locuteur et S+L pour une combinaison de ces deux détections (Figure 2, section 4.3).

Nous avons choisi de créer un automate à 4 états : **IN**, **OUT**, **OFF**, ainsi qu'un état de **doute** qui nous sert d'état initial ainsi que d'échappatoire lorsque les informations dont nous disposons ne sont pas suffisantes pour attribuer une classe à un intervenant (Figure 4). Nous avons considéré qu'un état restait stationnaire sur toute la durée du segment analysé, et défini les transitions de l'automate comme les possibilités à explorer à chaque nouvelle prise de décision, c'est-à-dire comme les conditions acoustiques et visuelles inhérentes à nos descripteur pour chaque nouvel état.

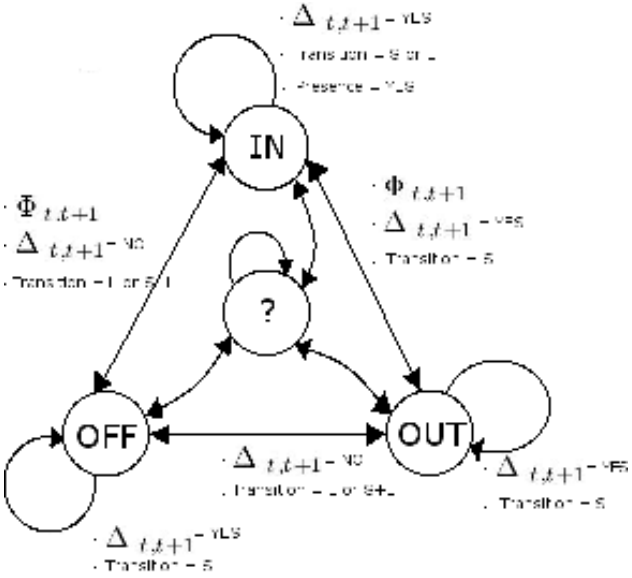


Figure 4 – *Présentation de l'automate à 4 états : IN, OUT, OFF et ? (doute).*

Par exemple, une transition de l'état IN à l'état OUT doit être considérée :
– si personne n'est détecté au segment $t+1$ ou si le personnage détecté en $t+1$ a un score d'activité plus faible que celui détecté en t,
– si l'environnement acoustique reste stable,
– et si un changement de plan a été détecté.

Comme il n'existe aucune vérité terrain prenant en compte une classification IN/OUT/OFF, nous avons développé notre propre corpus d'évaluation d'une durée totale de 21 minutes. Voici une présentation des résultats obtenus par notre automate :

- si nous considérons **DOUTE** comme une classification correcte, nous obtenons un taux de précision de 87,1%,
- si nous considérons **DOUTE** comme une erreur de classification, nous obtenons un taux de précision de 55,8%,
- si nous faisons abstraction du doute, c'est-à-dire si nous considérons uniquement les segments qui ne sont pas classés **DOUTE**, nous obtenons un taux de précision de 82,6%.
- l'automate rentre en état **DOUTE** dans 24,2% des cas.

Afin de pouvoir apprécier les performances de ce classifieur, nous allons comparer les résultats obtenus avec le type de classification qui existait avant notre étude : *un locuteur est de classe IN lorsqu'une personne est détectée à l'écran pendant la locution, sinon il est OUT*. Pour pouvoir faire la comparaison nous appellerons cette classification **old**. Nous nommerons la notre **new** et nous fusionnerons nos classes OUT et OFF en une classe **OUT** unique (les erreurs entre OUT et OFF ne sont plus considérées).

Le tableau 1 donne le pourcentage de segments correctement classifiés (de manière automatique) si nous ne tenons pas compte des segments classés **DOUTE** pour l'évaluation. Le tableau 2 présente les résultats si nous les prenons en compte. Dans les deux cas, une amélioration notable est obtenue grâce à l'utilisation de notre classifieur.

Tableau 1 – *Résultats obtenus en ne tenant pas compte du doute.*

	Doute = Void
old	55%
new	88.4%

Tableau 2 – *Résultats obtenus en tenant compte du doute.*

	Doute = False	Doute = True
old	48%	48%
new	70.5%	91.4%

5.2 Explications

La figure 5 présente une comparaison entre 3 séquences audiovisuelles extraites de notre corpus et les résultats obtenus par notre classifieur :

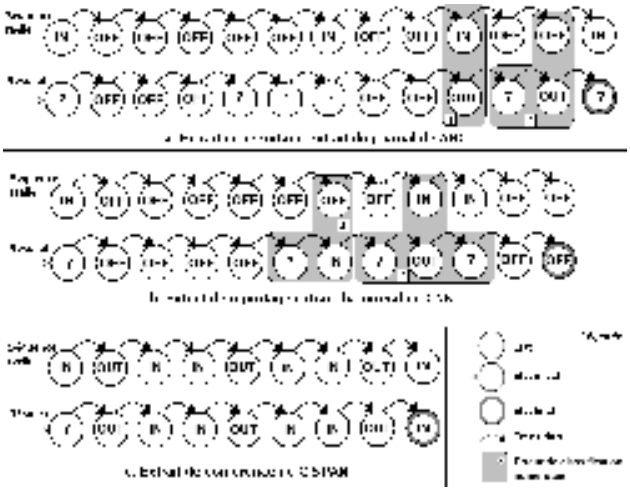


Figure 5 – *Schémas illustrant la comparaison entre des séquences audiovisuelles et les résultats obtenus par notre classifieur.*

– **analyse de la figure 5c :**

Ces valeurs proviennent d'un extrait (5 minutes 12 secondes) d'une conférence. Les segmentations sont des changements de plans : la caméra passe du protagoniste à des personnes du public.
Nous obtenons dans ce cas d'excellents résultats car nous nous trouvons dans une configuration claire, sans ambiguïté sonore ou visuelle :
- l'environnement sonore reste stable car le son émane du même micro par une même personne durant toute sa locution,
- le personnage n'est jamais interrompu et le flux de parole est constant,
- le protagoniste ainsi que les spectateurs sont filmés en plans américains, sont assis et ne bougent pas trop.

– **analyse des figures 5a et 5b :**

Ces valeurs sont issues d'extraits (1 minute 37 secondes pour la 5a et 1 minute 5 secondes pour la 5b) de reportages. La voix OFF d'un reporter couvre le document, qui est une succession de changements de plans, cédant ponctuellement la parole à des intervenants filmés qui sont interviewés au micro.

Les erreurs de classification sont dues :

- à la non-détection visuelle du locuteur, à cause de la trop faible qualité vidéo du document : un intervenant IN est pris pour un intervenant OUT (**erreur 1** figure 5a),

- à la prise de décision aléatoire de sortie de l'état **DOUTE** lorsque plusieurs possibilités sont offertes (**erreur 2** figure 5a). Une manière très simple de remédier à ce problème est d'attribuer des poids aux transitions suivant

leur fréquence d'apparition dans tel ou tel type de document (et rajouter ainsi de l'information *à priori*),

- à une mauvaise interprétation de la stabilité de l'environnement sonore : la voix OFF post-synchronisée du reporter enregistrée en régie, se fond à certains moments avec l'ambiance sonore propre à l'environnement visuel filmé. Le canal de la voix OFF est alors perturbé par un bruit acoustique différent de celui propre à l'enregistrement (**erreur 3** figure 5b),

- au manque de mémoire de l'automate (**erreur 4** figure 5b) : le prototype que nous avons construit ne traite que 2 états à la fois, ne gardant pas de trace du chemin déjà effectué ou à venir. Une manière de résorber ce dernier problème serait d'opérer une classification en deux passes.

6 CONCLUSION

Nous avons présenté des descripteurs vidéos qui nous ont permis de d'étudier l'activité de parole des intervenants d'un segment sur l'autre, de déterminer lequel des personnages parle au sein d'un même segment, et enfin de pallier une partie des déficiences du détecteur de visages de Viola et Jones lors de son utilisation pour du suivi de visage. Nous pensons que la fiabilité de ces descripteurs peut devenir meilleure si des mouvements de caméra, tels que le zoom ou le travelling, sont pris en compte.

Nous avons également montré que la variation des MFCC aux frontières des transitions entre classes sonores constitue un descripteur fiable lorsqu'il s'agit de caractériser un changement ou une stabilité entre deux environnements sonores. Malgré les mauvais résultats que nous avons eu lors de nos premières analyses concernant l'évolution des MFCC, analyses faites dans le but de caractériser un environnement acoustique plutôt que des changements entre eux, nous pensons toujours qu'il y a suffisamment d'information dans les coefficients cepstraux pour exhiber ce genre d'information.

Enfin, la mise en commun de ces informations au sein d'un automate nous a permis de créer un descripteur audiovisuel pertinent pour obtenir une classification IN, OUT ou OFF inédite d'un intervenant dans un document. Nous pouvons envisager, afin de perfectionner ce classifieur, de *rajouter de la mémoire à l'automate* en lui permettant de pratiquer une classification en deux passes. Nous pouvons également *adapter le classifieur au contexte* en attribuant des poids à chaque transition entre états selon leur fréquence d'apparition, suivant le type de document traité. De plus, *pondérer l'état* ***DOUTE*** de manière plus ou moins forte permettrait de pouvoir jouer entre précision (en augmentant son poids) et fiabilité (en diminuant son poids).

Nous espérons que ce type de classification pourra permettre, de part la nature bimodale de ces classes, de simplifier l'identification du locuteur en liant les medias audio et vidéo au sein même de la modélisation du problème.

Références

[1] G. POTAMIANOS, C. NETI, J. LUETTIN, et I. MATTHEWS. Audio-visual automatic speech recognition : An overview. Dans G. BAILLY, E. VATIKIOTIS-BATESON, et P. PERRIER, éditeurs, *Issues in Visual and Audio-Visual Speech Processing*. MIT Press, 2004.

[2] E. KIJAK. *Structuration multimodale des videos de sports par modeles stochastiques*. Thèse de doctorat, Universite de Rennes 1, Décembre 2003.

[3] Wessel Kraaij, Alan Smeaton, Paul Over, et Joaquim Arlandis. Trecvid 2004 - an introduction. Dans *Proceedings of the TRECVID 2004 Workshop*, pages 1–13, Gaithersburg, Maryland, USA, Novembre 2004.

[4] Julien Pinquier, Jean-Luc Rouas, et Régine André-Obrecht. Fusion de paramètres pour une classification automatique parole/musique robuste . Dans *Technique et science informatiques (TSI) : Fusion numérique/symbolique* , volume 22, pages 831–852. Hermès, 8, quai du marche neuf, F-75004 Paris, 2003.

[5] G. JAFFRE, P. JOLY, et S. HAIDAR. The SAMOVA Shot Boundary Detection for TRECVID Evaluation 2004. Dans *TRECVID 2004 Workshop, Gaithersburg, Maryland USA*, pages 179–183. NIST, 15-16 novembre 2004.

[6] G. JAFFRE et P. JOLY. Costume : A new feature for automatic video content indexing. Dans *RIAO 2004*, pages 314–325, Avignon, France, april 2004.

[7] P. VIOLA et M. JONES. Rapid object detection using a boosted cascade of simple features. Dans *IEEE CVPR*, 2001.

[8] G. POTAMIANOS, H.P. GRAF, et E. COSATTO. An image transform approch for hmm based automatic lipreading. Dans *Proceedings of the Internationnal Conference on Image Processing*, volume 3, pages 173–177, Chicago, 1998.

[9] C. MOKBEL, D. JOUVET, et MONNE J.. Blind equalization using adaptitive filtering for improving speech recognition over telephone. Dans *European Conference on Speech Communication and Technology*, pages 817–820, Madrid, Spain, 1995.

[10] Calliope. *La parole et son traitement automatique*. Masson, Paris, France, 1989.

[11] QIAN-JIE F. TIANHAO, L. Analyze perceptual adaptation to spectrally-shifted vowels with gmm technique. Dans *10th Annual Fred S. Grodins Graduate Research Symposium*, pages 120–121. USC School of Engineering, 04 2006.

Labellisation du Comportement de Descripteurs Locaux pour la Détection de Copies Vidéo

J. Law-To[1 2] V. Gouet-Brunet[2] O. Buisson[1] N. Boujemaa [2]

[1] INA Direction de la Recherche et Expérimentation, Bry Sur Marne
[2] INRIA Rocquencourt Equipe IMEDIA, Rocquencourt

{jlawto@ina.fr, valerie.gouet@inria.fr, obuisson@ina.fr, nozha.boujemaa@inria.fr}

Résumé

Ce papier présente une approche efficace d'indexation et de recherche dans de grandes bases de vidéos. Cette indexation automatique exploite un ensemble de descripteurs locaux et leurs trajectoires à travers la séquence vidéo. Cette méthode permet d'une part de réduire la redondance temporelle intrinsèquement liée à la vidéo et d'ajouter d'autre part un contexte de comportement à ces descripteurs. Ainsi, en partant d'une description bas-niveau du signal, notre approche permet d'aboutir à une représentation de plus haut niveau, associant une tendance de comportement aux descripteurs locaux. La description obtenue est d'une part plus compacte, non redondante et d'autre part peut être rendue spécifique à la vidéo en fonction de l'application de recherche désirée. Une application cruciale dans la gestion de patrimoines numériques est la traçabilité du catalogue vidéo et nous proposons dans cet article un système de détection de copie par le contenu et son évalution. L'évaluation montre une nette amélioration des performances face à une technique état de l'art tout en présentant une meilleure flexibilité. Elle est de plus temps réel sur une base vidéo importante (plusieurs centaines d'heures).

Mots clefs

Indexation vidéo par le contenu, détection de copies vidéos, descripteurs locaux, trajectoires.

1 Introduction

La croissance des contenus audiovisuels, et en particulier vidéo nécessite la création d'outils de recherche par similarité ou copies. La traçabilité des contenus audiovisuels est une nécessité pour les professionnels des archives et les détenteurs de droits vidéo. La détection de copie par le contenu (en anglais *Content Based Copy Detection, CBCD)* est une alternative au tatouage d'images pour tracer un fond d'archives vidéo. Les méthodes de recherche par le contenu et en particulier sur la détection de copies dans les fonds vidéo consistent généralement à extraire des éléments caractéristiques de la vidéo appelées signatures et à les comparer à une base. Plusieurs approches existent dans la littérature : dans [1, 2], les auteurs utilisent des signatures temporelles alors que dans [3], les auteurs comparent des méthodes basées sur des descriptions globales au niveau image (couleurs) et temporelles (mouvement, distribution de l'intensité). Ces descriptions globales (temporelles ou spatiales) ont l'avantage de caractériser les séquences de manière légère (1 vecteur par trame) mais sont peu robustes et peu discriminantes. En effet, la notion de copie dépasse la réplique exacte et inclu tout nouveau montage issus d'une vidéo. Les modifications sont de types divers (changement de la luminance, insertion d'éléments, décalage de l'image, remontage, etc). Ainsi la rechercher de copie apparait comme un sous ensemble du vaste dommaine de la recherche par similarité. La figure 1 illustre cette idée : elle montre l'exemple de deux vidéos beaucoup plus similaires en terme de contenu qu'une vidéo et sa copie.

Similaires mais non copies (les cravates sont différentes)

Copies (l'une est faite à partir de l'autre)
Source video : *Gala du Midem.* G. Ulmer 1970 (c) Ina

Figure 1 – *Copie / similarité.*

Ces contraintes nous ont orientés vers une *description locale* de la vidéo et donc vers les points d'intérêt. L'utilisation de signatures basées sur ceux ci a prouvé son efficacité

pour retrouver des images [4] ou des vidéos [5].
Le concept que nous proposons est basé sur l'estimation et la caractérisation de trajectoires de points d'intérêt le long de la séquence vidéo. Il présente deux avantages : tout d'abord, la redondance temporelle des descriptions locales, intrinsèquement liée à la vidéo, est éliminée avec une perte d'information réduite, comme cela a déjà été fait notamment dans [6, 7]. Dans un deuxième temps, l'analyse de ces trajectoires fait ressortir des tendances de comportements locaux et permet donc d'enrichir chaque descripteur local en lui ajoutant une information sur le comportement spatial et temporel du point. Cette description permet d'assigner des labels de comportements aux descripteurs locaux. L'objectif est d'obtenir une description de la vidéo par le contenu plus *riche*, plus *compacte* tout en restant *générique*. De tels labels, associés à des tendances de comportement des points, peuvent être interprétés comme un contexte cinématique associé aux descripteurs locaux. La notion de contexte associé à une description locale a été récemment proposé pour les images fixes, en ajoutant une information semi-globale autour du point [8] ou caractérisant les relations spatiales entre voisins [9].

2 Description bas-niveau par descripteurs locaux

Cette section présente la description bas niveau des séquences vidéos que nous avons choisies. Elle se fait en deux étapes : l'extraction de la description locale sur chaque trame et le suivi de cette description le long de la séquence vidéo. Les techniques présentées ici sont classiques et ne représentent pas la contribution majeure de notre travail.

Descripteurs locaux. Les points d'intérêt ont d'abord été développés pour la Vision par Ordinateur puis pour la recherche d'images par le contenu. De nombreuses approches de détection de points et de description locale ont été proposées. Le lecteur peut trouver une évaluation des méthodes les plus connues dans [10]. Les points d'intérêt ont été étendus au niveau spatio-temporel [11].
Les points d'intérêt sont pertinents pour une recherche précise et locale dans l'image comme des détails ou des objets. Associés à un vote spécifique, ils sont robustes aux occultations, aux décalages et à certaines transformations géométriques et par conséquent sont pertinents pour la détection de copies. Pour évaluer notre algorithme, nous avons utilisé le détecteur de Harris [12] associé à une description locale classique (jet local) sur quatre positions autour du point détecté $\vec{s_i} = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial^2 I}{\partial x \partial y}, \frac{\partial^2 I}{\partial x^2}, \frac{\partial^2 I}{\partial y^2}\right)$. Nous obtenons donc une signature par points de dimension 20. Cette description est ensuite normée pour être invariante à un changement affine de luminance. Cet espace de description est designé S_{Harris}. Nous n'avons pas utilisé la populaire approche SIFT [13], car elle implique un espace de signature trop important (128 dimensions par points) ce qui appliqué à la vidéo devient très vite problématique (une heure de vidéo représente $25 * 3600$ image aboutissant à 3×10^6 descripteurs). Nous n'utilisons pas ce type de points car ils ne permettent pas de décrire certaines informations de contexte temporel pourtant pertinentes, comme par exemple les points de décor qui décrivent une information non traitée par les points d'intérêt spatio-temporels.

Construction des trajectoires. Les trajectoires sont ensuite construites sur le même principe que le classique algorithme KLT [14] : on apparie les descriptions locales de trame en trame. La différence est que nous effectuons cette mise en correspondance sur les 15 trames précédentes et suivantes, de manière à être robuste à d'éventuelles ruptures de trajectoires, non négligeables dans les vidéos où la qualité d'image est plutôt faible. A noter que cette méthode de suivi est générique et pourrait être appliquée à tout autre type de descripteurs.

3 Vers les labels de comportement

Cette section décrit la manière d'aboutir à une description de plus haut niveau, comportant donc plus de sémantique que la simple description bas-niveau précédente.

3.1 Description du signal sur une trajectoire

A chaque trajectoire de points, on souhaite associer une description du signal que l'on qualifie de description bas niveau. Pour cette description, nous calculons la moyenne de chaque composante des descriptions locales des points de la trajectoire. La description obtenue est notée $\vec{S}_{mean}$. Lors de la construction des trajectoires, l'appariement étant fait de proche en proche, la valeur du descripteur local peut théoriquement dériver largement. Pour vérifier la pertinence de la description choisie, plusieurs valeurs de $\vec{S}_{mean}$ ont été analysées sur une séquence vidéo de 1 heure : pour chaque trajectoire, nous avons calculé la distance de chacun des points avec la valeur de $\vec{S}_{mean}$ correspondant. Dans 95 % des cas, cette distance est plus petite que le seuil utilisé dans l'étape d'appariement de la construction des trajectoires. Cette expérience confirme que $\vec{S}_{mean}$ caractérise bien le signal le long d'une trajectoire. Une approche similaire est décrite dans [7] : les auteurs observent que sur la trajectoire construite à partir d'un changement d'angle de vue progressif, la description SIFT a une variation quadratique ; les auteurs prennent également la moyenne de la description comme description finale. Dans la suite du papier, nous appelons cet espace de description S_{Signal} ; il est de même nature que S_{Harris} (même dimension, même type de distribution).

3.2 Description cinématique des trajectoires

Une description de plus haut niveau peut être obtenue en associant un contexte spatio-temporel au descripteur $\vec{S}_{mean}$. Ce contexte est obtenu en récupérant les propriétés des trajectoires, qui sont de nature spatiale et temporelle, donnant une information cinématique sur le comportement du point d'intérêt le long de sa trajectoire. Nous considérons les propriétés suivantes, calculées pour chaque trajec-

toire et stockées durant la partie d'indexation hors-ligne :
– Time code de début et de fin : $[tc_{in}, tc_{out}]$;
– Variation spatiale : $[x^{min}, x^{max}], [y^{min}, y^{max}]$.
Cette espace de description est noté S_{Traj} par la suite. L'association de S_{Signal} et S_{Traj} permet d'enrichir la description de la vidéo, qui reste générique.

3.3 Definition des labels

A partir des propriétés définies plus haut, il est possible de déterminer des tendances de comportements. Considérons par exemple les points ayant les caractéristiques suivantes :
– En mouvement / immobile ;
– Persistant / rare (persistance = 1) ;
– Mouvement rapide / lent ;
– Mouvement horizontal / vertical.
Cette liste ne représente que quelques exemples de labels que l'on peut attribuer. En classant les descripteurs locaux en fonction de leur comportement, il devient donc possible d'étiqueter chaque descripteur de S_{Signal}. Pour l'instant, les labels et catégories de comportement sont obtenus par simples seuillages globaux.
Cette annotation constitue une description de *haut niveau* car elle implique une interprétation de la vidéo : le choix d'un label plutôt qu'un autre est *spécifique* au contenu de la vidéo. Le potentiel des labels obtenus est multiple : dans un premier temps, il vont servir à sélectionner des sous-ensembles de trajectoires pertinents afin de réduire de manière efficace l'espace de description et dans un deuxième temps, ils vont être exploités spécifiquement dans une fonction de vote pour améliorer la recherche.
Dans ce travail, nous nous attachons à la détection de copie et les labels que nous allons considérer sont : les points immobiles et persistants qui définissent un label *Décor* et les points persistants et en mouvement qui définissent le label *Mouvement*. Les points de décor apportent de la *robustesse* à la description tandis que les points en mouvement sont propres à la vidéo et donc très *discriminants*. Dans la section suivante, nous détaillons comment ces différents espaces de description venant d'être présentés vont être exploités dans un algorithme de recherche de copie vidéo.

4 Algorithme de recherche

Cette section présente la méthode de recherche de vidéos à partir de l'indexation décrite précédemment. Le cas particulier de la détection de copie est développé ici.

4.1 Une technique asymétrique

Contrairement à la plupart des méthodes de recherche aussi bien images que vidéos, nous n'effectuons pas les mêmes opérations sur les vidéos requêtes (VR) que sur les vidéos sources (VS). Le calcul des trajectoires n'est pas appliqué à la VR pour les raisons suivantes : la première est d'ordre pratique, en effet l'indexation hors ligne des VS nécessite un long temps de calcul (voir section 5.3 pour des ordres de grandeur) alors qu'un système de détection de copies pour assurer la traçabilité d'archives audiovisuelles doit être au minimum temps réel (les VR étant constituées du flux de toutes les chaines de TV). Ce constat pénalise d'ailleurs la plupart des méthodes utilisant des descriptions par tubes spatio-temporels (voir [15] par exemple) et si de plus on souhaite être efficace sur des volumes conséquents, on ne peut se permettre les couts de calculs de l'indexation hors ligne sur les VR. Une deuxième raison plus fondamentale est que l'on veut être robuste au remontage, à l'utilisation d'extraits et dans ces cas,les trajectoires peuvent être tronquées. Les VR sont donc échantillonées dans un espace de description similaire à S_{Harris} selon 2 paramètres :
– la periode p du choix de d'image extraite du flux ;
– le nombre n de points de Harris choisis.
Pour l'instant p et n sont constants et fixés par un opérateur mais on peut imaginer par la suite un choix dynamique de ces valeurs. L'avantage de la méthode asymétrique est qu'elle est rapide mais surtout, elle permet un choix en ligne de la qualité et de la granularité temporelle des détections. Le principal challenge de cette méthode est que l'on a d'un cotés des points de Harris (VR) dans l'espace de description S_{Harris} et de l'autres des trajectoires (VS) avec les espaces S_{Signal} et S_{Traj}. La figure 2 l'illustre bien : sur l'image de gauche, les croix représentent les requêtes tandis que les propriétés des trajectoires associées à la vidéo source sont représentées sur l'image de droite.

Vidéo Requête
Les + representent les sous-requêtes.
(Points d'interêt)

Vidéo Source
Les boites representent S_{Traj}
$[x^{min}, x^{max}], [y^{min}, y^{max}]$

Figure 2 – *Illustration des espaces de descriptions utilisés pour la méthode asymétrique.*

4.2 Recherche spécifique pour la détection de copies

Nous détaillons ici les différentes étapes de notre algorithme de recherche basé sur la méthode de description présentée. Dans ce travail, l'algorithme est dédié à la détection de copies, mais il peut s'appliquer à d'autres applications de recherche par contenu dans les vidéos, en choisissant les labels appropriés.

Recherche bas-niveau des plus proches voisins. La vidéo requête a donc été echantillonée en K sous-requêtes ayant chacune une description dans S_{Harris}, un time code tc_k et une position (x_k, y_k) ($k \in [1, K]$). La première étape de la recherche permet de ramener les plus proches voisins dans un rayon donné de l'espace de description S_{Signal}.

Chaque voisin ramené est une trajectoire et possède donc en plus de sa description dans S_{Signal}, une description dans S_{Traj} ainsi qu'un ou plusieurs labels de comportement. Cette étape utilise un algorithme de recherche approximative probabiliste non détaillé ici (voir [5]). A partir de ces voisins, une recherche spécifique en fonction de la robustesse nécessaire et de la similarité désirée sera effectuée. Les choix faits pour la détection de copies sont ici présentés.

Choix des labels. Pour la détection de copies, nous considérons les deux types de comportements des points introduits dans la section 3.3 pour leur pertinence : les points labellisés *Décor* et les points labellisés *Mouvement*. Ce choix a tout d'abord pour conséquence de réduire le volume de l'espace de description des points ; dans un deuxième temps, ces deux types de points vont permettre un recalage des sous-requêtes dans les trajectoires, comme décrit ci-après.

Recalage spatio-temporel. Un vote par comptage simple des candidats ramenés précédemment n'est pas assez discriminant pour la détection de copie. Le vote que nous avons développé va tenter de recaler l'ensemble des sous requêtes sur les voisins ramenés qui sont des trajectoires. Le recalage spatio-temporel consiste à évaluer le décalage temporel et spatial entre la vidéo dans la base de données et le flux requête. Pour cela, on utilise la description de l'espace S_{Traj}. Les détails de ce recalage ne sont pas explicités ici car ce n'est pas l'objet du papier mais le principe est de faire un premier recalage par image et par label de comportement puis de fusionner ce recalage par image pour enfin propager dans le temps le décalage estimé. Chaque recalage se fait par comptage du nombre de requêtes compatibles à un décalage donné. Ce vote donne une grande robustesse au système : robustesse au décalage,à l'insertion de cadre, au remontage (en utilisant des extraits). La détection de copie par le contenu utilisant ce vote spécifique est évaluée dans la section suivante.

5 Evaluation pour le CBCD

Cette section présente notre méthode d'évaluation sur des cas simulés et des cas réels, et donne des indices de performance comparés à une méthode de l'état de l'art.

5.1 Cadre de l'évaluation

Base vidéo. Toutes les expériences sont réalisées sur une base de données vidéo réelles de 320 heures : 300 heures aléatoirement choisies dans les archives de l'INA [1] et 20 heures correspondant aux vidéos nécessaires à la dernière expérimentation (voir 5.4). Ces vidéos encodées en *MPEG-1* (25 im/s) présentent toute sorte de contenus (journaux TV, sports, émissions de variétés etc...) de différentes époques (émissions couleurs ou noir et blanc).

Définition des transformations. Afin de tester la robustesse du système, nous avons défini un certain nombre de transformations potentielles voulues (translations, insertions, recadrage, modification du gamma) ou non voulues (bruits, dégradations colorimétriques) comme on peut le voir sur les images (a) et (b) de la figure 3. La robustesse du système sera testée en prenant des vidéos de la base et en les transformant artificiellement.

[1] Institut Nationnal de l'Audiovisuel

(a) Reportage TV *1993, France 3*

(b) *Chronique Bretonne 1970* (c) Ina

(c) Gauche : *Les duos de l'impossible* 2005, Droite : *Vient de Paraitre.* J. Guyon 1965 (c) Ina.

(d) Gauche : *Les duos de l'impossible* 2005, Droite : *Système deux.* C. Fayard 1975 (c) Ina.

Figure 3 – *Exemples de détection de copies. A gauche, les Vidéos Requêtes (vidéos avec des transformations aléatoires ou émission tv). A droite : vidéo de la base (VS).*

Technique de référence. Afin d'évaluer notre technique de détection de copie, nous avons besoin d'une référence. Nous avons choisi de comparer notre méthode à celle décrite dans [5]. Les auteurs utilisent aussi des descripteurs locaux et obtiennent de très bons résultats, même sur des grandes bases de vidéos (10 000 heures). Il y a cependant deux différences fondamentales : nous avons indexé toutes les images sans limitation du nombre de descripteurs à priori, tandis que la technique de référence indexe uniquement 20 points sur des images clefs. La deuxième diffé-

rence est que l'on ajoute un contexte de comportement des points à notre description. Les techniques utilisant des descripteurs globaux ne nous semblent pas assez robustes à de grandes transformations comme les insertions. Par exemple dans [3], les performances sont moindre surtout pour des extraits courts alors que la base de donnée est très petite. Les auteurs de [5] nous ont confié leur code pour effectuer cette comparaison dans les mêmes conditions de tests.

5.2 Evaluation sur un jeu de test

Notre méthode est tout d'abord testée sur deux jeux de test : *Bench1min* et *Bench30* construits comme représenté sur la figure 4. Nous avons selectionnés aléatoirement 40 extraits vidéo de notre base puis nous les avons transformés artificiellement en utilisant des paramètres aléatoires. Ces segments ont une durée de 1 minute pour *Bench1min* et une durée aléatoire comprise entre 5 images et 30 secondes pour *Bench30*. Ces segments sont ensuite insérés au hasard dans un flux de 7 heures de flux vidéos de différentes chaines de TV.

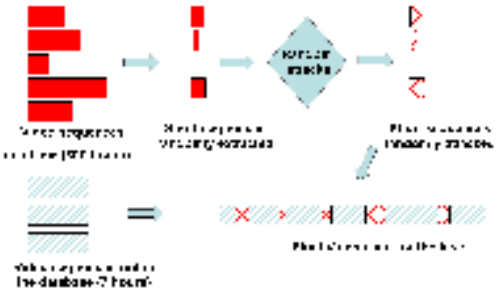


Figure 4 – *Construction du jeu de test.*

Ces expériences permettent d'évaluer le système dans une situation "réelle" simulée : des segments vidéos attaqués sont inclus dans un flux important de vidéos. Le but étant de détecter ces segments et de les retrouver le plus préciément possible. Nous avons utilisé deux jeux de paramètres requêtes :

– $p = 30$ and $n = 20$ pour avoir le même nombre de sous-requêtes que la référence,
– $p = 15$ and $n = 50$ pour tester l'amélioration possible.

Les figures 5 et 6 présentent les courbes précision/rappel pour les deux jeux de tests. Le second test est plus difficile car il met en jeu des segments très courts (moins de 1s pour certains). Plusieurs remarques peuvent être faites :

– Sur le test *Bench1min*, les techniques sont très performantes avec un rappel supérieur à 90% pour une précision de 95%. En augmentant le nombre de requêtes, on retrouve toutes les séquences (100% comparé à 97% pour la référence).
– Pour le test *bench30*, la chute du rappel apparait à une précision de 52% pour la référence tandis qu'elle apparait à 64% pour notre technique avec le même nombre de requêtes.

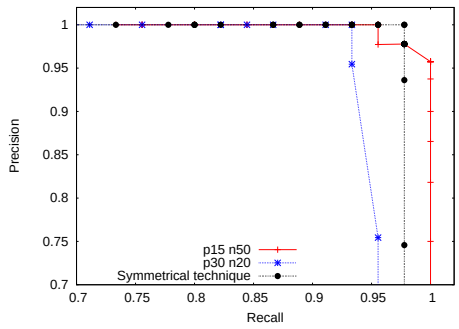


Figure 5 – *Precision/rappel pour Bench1min.*

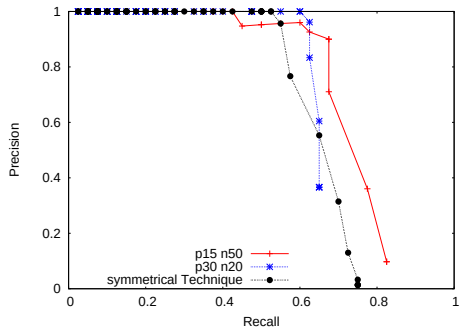


Figure 6 – *Precision/rappel pour bench30.*

– L'augmentation du nombre de requêtes permet d'augmenter le rappel mais cause parfois une baisse de la précision du fait de l'apparition de certaines fausses alarmes : la précision chute plus tot (44% du rappel).

En conclusion pour ces 2 tests, le rappel pour une précision acceptable (supérieure à 90%), est meilleur pour notre technique et d'autant meilleur que les segments à retrouver sont courts (100% comparé à 97% et 71% comparé à 55%) ce qui est particulièrement intéressant pour l'utilisation d'images d'archives dans des reportages par exemple.

5.3 Temps de calculs

Le fait de travailler sur de gros volumes de vidéos pose le problème des temps de calculs. Le système final doit être temps réel car dans la pratique les vidéos requêtes sont infinies (flux TV 24h/24). Nous donnons ici les valeurs mesurées sur l'expérimentation précédente en utilisant la fonction *time* de linux (table 1). L'indexation hors ligne est au final 1.5 fois plus lente que le temps réels sur un PC standard (Pentium IV, 2.5 GHz, 1 Go RAM). C'est le calcul des points de Harris qui prend la majorité du temps CPU mais nous n'avons pour le moment pas optimisé ce code. L'avantage de notre méthode est que ce calcul n'est effectué qu'*une seule fois* et qu'à partir de cette description, nous pouvons extraire très rapidement les descripteurs que l'on souhaite utiliser en fonction de l'application. La recherche en ligne est très rapide : 6 fois le temps réel alors

que la partie la plus gourmande est la partie image (extraction de points de Harris requêtes).

Indexation hors ligne	320 heures de vidéos	
Calcul de S_{Traj} et S_{Signal}	460 hours	0.7 T.R.
Construire l'espace	5 min	3600 T.R.
Détection en ligne	7 heures de requêtes	
Calcul des requêtes	45 min	9 T.R.
Recherche et vote	22 min	19 T.R.
Total	67 min	6 T.R.

Tableau 1 – *Temps de calcul : Temps mesuré et comparé au temps réel (T.R.).*

5.4 Un cas réel difficile

De nombreuses émissions TV utilisent des images d'archives et certaines effectuent en post-production de très fortes transformations de l'image comme l'insertion de personnes dans une vidéo d'archive associée à une forte translation (voir image (c) de la figure 3) ou même l'élimination du décors (voir image (d) de la figure 3). Ces transformations sont des cas extrèmes ; elles vont nous permettre d'illustrer la force des descriptions locales pour la détection de copies. En effectuant un test sur 3 heures d'émissions en requête et notre base de 320 heures de vidéos, nous avons comparé les résultats à la technique de référence (voir la table 2).

Segments retrouvés avec la technique de référence	43
Segments retrouvés par notre technique	82
Temps de recherche avec la technique de référence	7min53s
Temps de recherche par notre technique	10min44s

Tableau 2 – *Résultats de détection de copie sur un cas réel.*

Comme précédemment, notre technique montre son avantage sur les segments courts (la taille moyenne des segments supplémentaires détectés est 4.3 s) et au final la quantité de vidéo retrouvée en plus est importante : 2min 51s, ce qui correspond à un gain de 36 %, non négligeable dans un but de traçabilité du patrimoine audiovisuel.

6 Conclusion et perspectives

Ce papier présente une méthode robuste et efficace d'indexation et de recherche par le contenu de vidéos. Il décrit deux contributions complémentaires : la première est une description du comportement de descripteurs locaux dans une vidéo en leur attribuant un contexte. La deuxième est l'utilisation de ce contexte pour effectuer une recherche spécifique de vidéo illustrée dans ce papier pour la détection de copies. L'utilisation de labels de comportement permet de rendre l'espace de description plus compact tout en améliorant la robustesse et la discriminance du système. L'évaluation de cette méthode sur un jeu de test difficile et sur un cas réel extrème montre l'efficacité de la méthode et l'amélioration des résultats face à des systèmes état de l'art tout en étant temps réel. Les travaux futurs seront axés sur l'utilisation de descripteurs locaux complémentaires utilisant la généricité de nos algorithmes afin d'améliorer encore les performances. Une autre perspective est l'amélioration de la méthode de définition des labels de comportement en utilisant des méthodes de classification non supervisées. Enfin, les applications de ce type d'indexation sont multiples et le développement de détections de type d'émission par similarité est envisagé.

Références

[1] P. Indyk, G. Iyengar, et N. Shivakumar. Finding pirated video sequences on the internet. Technical report, Stanford University, 1999.

[2] X-S. Hua, X. Chen, et H-J. Zhang. Robust video signature based on ordinal measure. Dans *ICIP*, 2004.

[3] A. Hampapur et R. Bolle. Comparison of sequence matching techniques for video copy detection. Dans *Conference on Storage and Retrieval for Media Databases*, pages 194–201, 2002.

[4] S-.A. Berrani, L. Amsaleg, et P. Gros. Robust content-based image searches for copyright protection. Dans *ACM Intl. Workshop on Multimedia Databases*, pages 70–77, 2003.

[5] A. Joly, C. Frelicot, et O. Buisson. Feature statistical retrieval applied to content-based copy identification. Dans *ICIP*, 2004.

[6] J. Sivic et A. Zisserman. Video Google : A text retrieval approach to object matching in videos. Dans *ICCV*, volume 2, pages 1470–1477, Octobre 2003.

[7] M. Grabner et H. Bischof. Extracting object representations from local feature trajectories. Dans *1st Cognitive Vision Workshop*, 2005.

[8] E. N. Mortensen, H. Deng, et L. Shapiro. A sift descriptor with global context. Dans *CVPR*, 2005.

[9] J. Amores, N. Sebe, et P. Radeva. Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors. Dans *CVPR*, 2005.

[10] Krystian Mikolajczyk et Cordelia Schmid. A performance evaluation of local descriptors. *ICPR*, 2003.

[11] I. Laptev et T. Lindeberg. Space-time interest points. Dans *ICCV*, 2003.

[12] C. Harris et M. Stevens. A combined corner and edge detector. Dans *4th Alvey Vision Conference*, pages 153–158, 1988.

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. Dans *IJCV*, 2004.

[14] C. Tomasi et T. Kanade. Detection and tracking of point features. Technical report CMU-CS-91-132, Avril 1991.

[15] D. DeMenthon et D. Doermann. Video retrieval using spatio-temporal descriptors. Dans *ACM international conference on Multimedia*, 2003.

Prédiction de séries temporelles : application à la structuration de flux audiovisuels

Jean-Philippe Poli[1,2] Jean Carrive[1]

[1] Equipe Description des Contenus Audiovisuels

Direction de la Recherche et de l'Expérimentation
Institut National de l'Audiovisuel, 4 avenue de l'Europe, 94366 Bry-sur-Marne cedex - France

{jppoli,jcarrive}@ina.fr

[2] LSIS - UMR CNRS 6168

Universite Paul Cézanne (Aix-Marseille 3)
Domaine Universitaire de Saint-Jérôme
Avenue Escadrille Normandie-Niemen, 13397 MARSEILLE CEDEX 20 - France

{jean-philippe.poli}@lsis.org

Résumé

Les séries temporelles sont abondamment employées dans des domaines très variés. De nombreux chercheurs se sont intéressés à caractériser et à prédire des séries temporelles régulières, c'est-à-dire dont les valeurs sont mesurées à intervalles temporels réguliers. Nous proposons de considérer la grille de programme d'une chaîne de télévision comme une série temporelle dont les dates sont les heures de début d'un programme et les valeurs en sont le genre. Une prédiction d'une telle série temporelle nous permet à la fois de connaître la succession des genres d'émissions mais aussi d'estimer l'heure de début d'un programme afin de rechercher localement dans le signal la rupture avec l'émission précédente. Nous présentons dans cet article un moyen de prédire une telle série temporelle compte tenu d'une difficulté jamais rencontrée dans la littérature : la prédiction à la fois de la date et de la valeur de la prochaine observation de la série temporelle.

Mots clefs

Série temporelle, prédiction, grille de programmes, arbre de régression, modèle de Markov, structuration automatique de flux audiovisuel.

1 Introduction

La plupart des chaînes de télévision diffusent à présent leurs programmes en continu. La conservation de leurs diffusions, présentant autant d'intérêts sociologiques qu'éducatifs ou même ludiques, n'est plus un problème sur le plan technique grâce aux coûts abordables de leur captation et de leur sauvegarde. La difficulté réelle, à laquelle est confronté l'Institut National de l'Audiovisuel, réside dans la mise à diposition de ces archives dont la vitesse de croissance ne fait qu'augmenter avec la multiplication des chaînes et des moyens de diffusion. En effet, les flux télévisuels nécessitent d'être décrits afin de permettre une recherche ultérieure. Afin d'accomplir une telle tâche, il est souhaitable de décomposer ces flux en plus petites unités. L'unité de la programmation d'une chaîne de télévision étant l'émission, nous nous proposons de décomposer automatiquement un flux télévisuel en émissions afin que chacune d'elles soit documentée.

Si les chercheurs en indexation automatique se penchent activement sur des problèmes tels que la reconnaissance des types d'émissions, la structuration et le résumé automatiques de vidéos[1], ils ne se sont que peu intéressés, à notre connaissance, à la structuration de flux. Pourtant, les grilles prévisionnelles publiées chaque semaines dans les magazines ou par voie électronique (EPG) représentent une source importante d'information sur le flux et les programmes qui le composent. En effet, les programmes mentionnés dans ces grilles prévisonnelles sont typés, résumés et la plupart du temps annotés. Idéalement, la structuration d'un flux consisterait à l'aligner sur la grille prévisionnelle. Malheureusement, une grille prévisionnelle est incomplète puisque que les émissions courtes (météo, traffic routier, magazines courts sur la vie quotidienne...) n'y figurent pas. Notre approche consiste à augmenter statistiquement la complétude des grilles de programmes prévisionnelles. Cependant, dans ce papier, nous nous intéressons à un modèle de prédiction du contenu d'un flux, indépendament de sa grille prévisionnelle. Pour cela, nous considérons la grille de programmes d'une chaîne comme une série temporelle dont chaque temps serait la date de diffusion (i.e. le jour

de la semaine et l'heure de diffusion) et chaque valeur serait un type de programmes. Compte tenu du fait que les programmes ont des durées différentes, l'espacement temporel entre deux valeurs ne sera pas régulier. La difficulté de l'approche consiste donc à la prédiction d'une série temporelle irrégulière pour laquelle il faut prédire à la fois les dates et les valeurs.
Nous nous proposons d'estimer un intervalle de durées pour chaque émission, à l'aide d'un arbre de décision, afin de prédire la date de la prochaine valeur, et d'utiliser un modèle Markovien contextualisé afin de prédire les valeurs. Nous exposerons à la fin les résultats que nous avons obtenus.

2 Séries temporelles

Définition 1 *(Série temporelle) On appelle série temporelle la suite d'observations (o_t, $t \in \mathbb{T}$) d'une variable O à différentes dates t.*

Remarque 1 *Dans cet article, nous qualifions de* régulières *les séries temporelles dont les différentes dates sont régulièrement espacées, et d'*irrégulières *celles qui n'ont pas un taux d'échantillonage constant.*

Les séries temporelles régulières sont extrêmement étudiées en économétrie[2], en astronomie[3] et dans bien d'autres domaines[4]. Beaucoup d'outils permettent de caractériser leurs évolutions et de prédire des valeurs en fonction de l'historique de la série temporelle. Ces différentes communautés ne se sont que très peu intéressées aux séries temporelles irrégulières. Pourtant, la prise en compte de telles séries permettrait d'étudier des données dont la collecte est évènementielle. [5] propose des modèles statistiques pour étudier des séries temporelles irrégulières à valeurs continues en se préoccupant de leur caractère stationnaire. Puisque l'hypothèse de travail de l'auteur est la possibilité de décomposer la série en une fonction déterministe et un bruit aléatoire, ces modèles ne peuvent pas être utilisés dans notre travail. En effet, nos séries temporelles vont représenter des grilles de programmes de chaînes de télévision. Les valeurs seront donc symboliques puisqu'elles représenteront les différents types d'émissions. La difficulté de la prédiction de telles séries réside dans le fait qu'il faut prédire d'une part la date de la prochaine observation, et d'autre part sa valeur.
Compte-tenu de la difficulté de la prédiction de la date d'un programme, nous faisons l'hypothèse que la date d'initialisation du système t_0 coïncide avec le début d'une émission. Les autres dates t_{i+1} de la série temporelle seront calculées à partir de t_i en y ajoutant la durée de l'émission. Nous présentons dans les sections suivantes la modélisation d'une série temporelle irrégulière sans perdre de vue que notre but est l'application à la structuration des flux.

3 Prédiction des valeurs par un modèle de Markov

Dans cette section, nous nous intéressons à la modélisation d'une série temporelle par un modèle de Markov intervenant dans la prédiction des valeurs.

3.1 Modèles semi-Markovien contextuels

Lorsqu'il s'agit de prédire les valeurs d'une série temporelle, il est nécessaire de choisir un modèle statistique qui permettra de calculer une valeur en fonction de l'historique des observations. Certains auteurs se sont intéressés aux réseaux de neurones[6], aux SVM (machines à vecteurs supports)[7] ou encore aux modèles de Markov[8]. Notre choix s'est porté sur ces derniers qui sont très utilisés pour représenter des processus stochastiques séquentiels, malgré leur inadaptation à la prise en compte de durées. De plus, nous allons utiliser ce modèle non pas pour estimer la probabilité d'une séquence d'observations mais plutôt pour obtenir de lui les séquences les plus probables que nous considérons comme des prédictions. Nous avons dû introduire une extension des modèles de Markov afin de limiter les possibilités de transition d'un état à l'autre et éviter d'avoir trop de séquences d'observations possibles mais improbables. Pour cela, nous avons contextualisé les transitions entre états.

Définition 2 *(Contexte) Un contexte θ est un ensemble de variables $x_1, ..., x_n$ à valeurs respectivement dans les domaines discrets ou continus $\{D_1, ..., D_n\}$. Une instance θ_i de ce contexte correspond à l'affectation d'une valeur à chacune de ses variables :*

$$\forall\, i \in \{1, ..., n\}, x_i = v_i \text{ avec } v_i \in D_i.$$

Dans cet article, nous appellerons *contexte* l'instance d'un contexte sans perte de généralité.

Exemple 1 *Dans notre cas, un contexte θ représentera le contexte de diffusion, avec une variable $Heure$ qui représentera l'heure de diffusion par un entier entre 0 et 86399, et une variable Jour qui représentera le jour de la semaine par un entier entre 0 et 6.*

$$\theta = \{Heure, Jour\}$$
$$et\ D_{Heure} = \{0, ..., 86399\}, D_{Jour} = \{0, ..., 7\}.$$

Définition 3 *(Fonction d'évolution) Soit Θ l'ensemble de toutes les instances possibles d'un contexte θ. On appelle fonction d'évolution la fonction F définie par :*

$$F : \Theta \times D_{p_1} \times ... \times D_{p_m} \rightarrow \Theta$$
$$\theta_i, p_1, ..., p_n \rightarrow \theta_{i+1}$$

où D_{p_i} est le domaine de valeurs du paramètre extérieur p_i.

Exemple 2 *En reprenant l'exemple 1, nous pouvons définir F comme étant la fonction qui à θ_i, qui représente le début de l'émission en cours, ajoute sa durée pour obtenir θ_{i+1}. Soit $\mathbb{D}$ l'ensemble des durées possibles, et L une durée particulière. Alors dans notre exemple, F peut être définie par :*

$$F : \Theta \times \mathbb{D} \rightarrow \Theta$$

$$\left\{ \begin{array}{l} Heure = h \\ Jour = j \end{array} \right\} \rightarrow \left\{ \begin{array}{l} Heure = (L+H) \mod 86400 \\ Jour = \left(j + \lfloor \frac{L+H}{86400} \rfloor\right) \mod 7 \end{array} \right\}$$

Définition 4 *(Modèles semi-Markoviens contextuels) Un modèle semi-Markovien contextuel est totalement défini par le n-uplet $< S, \Sigma, \Theta, F, \pi_\theta, A_\theta, B_\theta >$, où :*

- *S est un espace d'états de cardinalité n et s_i représente le i^e état de la séquence d'états,*
- *Σ est un alphabet de cardinalité m et ϵ_j représente le j^e symbole de l'observation,*
- *Θ est l'ensemble des instances du contexte θ,*
- *F, est une fonction d'évolution des instances du contexte θ,*
- *π_θ est un vecteur stochastique paramétré dont la i^e coordonnée correspond à la probabilité que la séquence d'état débute par l'état i :*

$$\forall\, \theta \in \Theta, \sum_{i=1}^{n} \pi_i(\theta) = 1.$$

π_i est une fonction de θ qui représente la distribution initiale dans le contexte θ :

$$\forall i \in \{1, ..., n\}, \pi_i(\theta_1) = P(s_1 = i|\theta_1),$$

- *A est une matrice stochastique n×n où a_{ij} représente la probabilité que l'état i soit suivit par l'état j dans la séquence d'état et où chaque a_{ij} est une fonction de θ :*

$$\forall \theta \in \Theta, \forall i \in \{1, ..., n\}, \sum_{j=1}^{n} a_{ij}(\theta) = 1.$$

$$\forall k, t \in \mathbb{N}, \forall i, j \in \{1, ..., n\},$$

$$a_{ij}(\theta_k) = P(s_{t+1} = j|s_t, \theta_k),$$

- *B est une matrice stochastique n×m où b_{ik} représente la probabilité d'observer le symbole k lorsque l'on est sur l'état i :*

$$\forall \theta \in \Theta, \forall i \in \{1, ..., n\}, \sum_{k=1}^{m} b_{ik}(\theta) = 1.$$

$$\forall k, t \in \mathbb{N}, \forall i \in \{1, ..., n\}, \forall j \in \{1, ..., m\}$$

$$b_{ij}(\theta_k) = P(\epsilon_{t+1} = j|s_t, \theta_k).$$

Afin de conserver la facilité d'expression d'une probabilité d'avoir une certaine séquence d'observations, il est nécessaire d'exprimer l'hypothèse de Markov contextuellement. Soit T la longueur d'une séquence d'observations :

$$P(s_t|s_{t-1}, \theta_1, ..., \theta_t, ..., \theta_T) = P(s_t|s_{t-1}, \theta_t)$$

$$P(s_t, \epsilon_t|s_1, ..., s_{t-1}, \epsilon_1, ..., \epsilon_{t-1}, \theta_1, ..., \theta_t) = P(s_t, \epsilon_t|s_t, \theta_t)$$

Autrement dit, les calculs des probabilités de transition et d'émission ne se font que par rapport au contexte courant et à l'état sur lequel on se trouve. Aucun historique des états ou des contextes n'intervient dans les calculs.
Avec ce modèle, nous allons représenter une grille de programmes en associant chaque état à un type d'émission. Mais au lieu de considérer un ensemble de symboles, nous allons utiliser un ensemble continu de valeurs entières afin de représenter les durées des programmes. Nous verrons dans la section suivante comment obtenir les probabilités des durées des émissions.
Il faut à présent entraîner le modèle afin d'en estimer les paramètres.

3.2 Estimation des paramètres du modèle

Intuitivement, il est aisé de voir que la prise en compte d'un contexte et de durées dans le modèle de Markov va augmenter le coût de l'apprentissage. Dans notre application à la structuration des flux et avec le contexte et la fonction d'évolution que nous avons choisis, l'estimation des paramètres du modèle semi-Markovien contextuel va donc dépendre fortement de la prédiction des dates de la série temporelle. En effet la fonction d'évolution F nécessite de connaître la durée entre deux évènements, c'est-à-dire dans notre cas la durée de l'émission. Nous proposons dans la section suivante une méthode de prédiction des durées.
En supposant le problème des durées résolu, l'estimation des paramètres de notre modèle revient à l'estimation des paramètres d'une chaîne de Markov, en calculant pour un contexte donné les fréquences d'occurences pour chaque type, puis chaque transition.
L'algorithme parcourt l'ensemble des exemples contenus dans la base. Pour chaque exemple, l'heure de début est estimée en fonction de l'heure de début réelle et de la durée du programme précédent. Et pour toutes les durées possibles de l'exemple en cours, on calcule les probabilités de transition.
Pour donner un ordre de grandeur du coût de l'apprentissage, nous pouvons chiffrer à 604800 le nombre de contextes possibles dans Θ et à 38 le nombre de genres d'émissions que nous considérons.
A ce stade, les valeurs de la matrice B du modèle semi-Markovien contextuel n'ont pas besoin d'être évaluées. L'outil de régression des durées que nous allons utiliser devra être en mesure de nous fournir la probabilité de ces durées.

4 Prédiction des dates et estimation des probabilités

Afin que notre représentation des grilles de programmes par notre extension des modèles markoviens soit valide, il est nécessaire pour un type d'émission donné de pouvoir estimer la probabilité d'une durée prédite. Plus généralement, pour pouvoir prédire une série temporelle irrégulière, nous avons besoin de prédire la durée entre deux évènements déclenchant un relevé des données.

4.1 Arbre de régression pour la prédiction des durées

Nous avons vu que la difficulté de prédire une série temporelle irrégulière dont la collecte des valeurs repose sur le déclenchement d'un évènement est un problème difficile qui demande la prévision de l'intervalle entre deux évènements. Dans le cas de la représentation des grilles de programmes, l'évènement est le début d'une nouvelle émission et il faut donc être en mesure de prédire la durée d'une émission. Contrairement à ce que nous pensions au début des travaux, même si deux émissions appartiennent à la même collection, leurs durées peuvent être très différentes. Par exemple, le magazine *Tout le monde en parle* sur France 2 peut durer entre une heure et demi et trois heures.

Il existe différentes méthodes de régression mais nous avons opté pour une méthode symbolique tirée des arbres de décision. Ces derniers sont des classifieurs dont l'entraînement supervisé se fait sur des classes symboliques. Quinlan dans [9] propose de remplacer ces classes symboliques par des classes continues. On parle alors d'arbres de régression. Comme dans la plupart des méthodes de régression, on suppose qu'il existe une seule variable de sortie, celle à approcher, et plusieurs variables d'entrées. L'avantage des arbres de décision ou de régression est que l'ensemble des variables d'entrées peut comporter aussi bien des variables continues que des variables symboliques.

Un arbre est construit récursivement par partition de l'ensemble d'apprentissage. Au début, tous les exemples sont dans un même ensemble. Le système va tenter de partitionner cet ensemble en fonction des valeurs des variables d'entrée. En fonction d'un certain critère, un test est choisi et le système réitère le processus sur chacune des partitions obtenues. Dans le cas d'un arbre de décision, lorsque les données d'une partition ont atteint l'enthropie désirée, le noeud est transformé en noeud terminal, appelé feuille. Tout nouveau cas présenté à l'arbre et arrivant sur cette feuille sera attribué à la classe dominante dans la partition associée à la feuille. Dans le cas d'un arbre de régression, il s'agira de la moyenne et de l'écart-type des individus présents dans la partition. Parfois même, une fonction est rattachée à la feuille pour régresser linéairement localement la variable d'entrée.

Afin d'adapter cet apprentissage symbolique à nos besoins, nous avons remplacé l'écart-type par la valeur minimum et maximum des durées ; cela nous permet d'obtenir une fenêtre temporelle dans laquelle on peut s'attendre à l'évènement suivant. En effet, si notre fenêtre temporelle devait être formée par la moyenne plus ou moins l'écart-type, elle ne couvrirait pas tous les cas de l'ensemble d'apprentissage. De ce fait, le critère proposé par Quinlan dans [9] a été remplacé par :

$$\begin{aligned}\Delta Error = \quad & \max(T) - \min(T) \\ & - \textstyle\sum_i \frac{|T_i|}{|T|}(\max(T_i) - \min(T_i))\end{aligned}$$

où T représente l'ensemble à partitionner et T_i une de ses partitions.

Les tests que nous avons effectués montrent que la variable d'entrée la plus importante est le genre de programme. Nous forçons donc l'apprentissage à commencer par cet attribut. Les autres tests effectués sur les données concernent l'heure de début, le jour de la semaine, ainsi que le genre du programme précédent. Nous nous sommes en effet aperçus que ce dernier était un bon moyen de réduire les problèmes causés par les retards de diffusion. Par exemple, considérons la suite consistituée d'un magazine court, d'une météo et d'un magazine long. Si le premier magazine commence trop tôt, il faut éviter que le deuxième soit considéré comme un magazine court au lieu d'un long.

Il arrive parfois que les arbres de décision ou de régression souffrent d'un sur-apprentissage. Dans ce cas là, les classes ou les valeurs de l'ensemble d'apprentissage sont parfaitement prédites, mais le taux d'erreur sur de nouveaux exemples est élevé. La plupart du temps, l'arbre est élagué, ce qui nécessite un ensemble de test. Comme nous n'avons pas suffisamment d'exemples, compte tenu de la diversité des durées, nous avons préféré introduire deux paramètres ρ et λ. ρ est la distance minimale entre la durée maximum et la durée minimum en-dessous de laquelle la partition de l'ensemble de données doit s'arrêter. Le paramètre λ, lui, impose une distance minimum entre la durée minimale et la durée maximale. Il est très important de faire la distinction entre les deux paramètres. Les sous-ensembles sont partitionnés tant que l'amplitude des valeurs est supérieur à ρ. Mais il se peut qu'un test crée un sous-ensemble avec une durée minimale très proche de la durée maximale. Alors on force la distance entre les deux à être égale à λ.

4.2 Estimation des probabilités par une gaussienne asymétrique

Nous sommes à présent en mesure de prédire des durées avec un arbre qui nous fournit une durée moyenne, une durée minimale et une durée maximale.

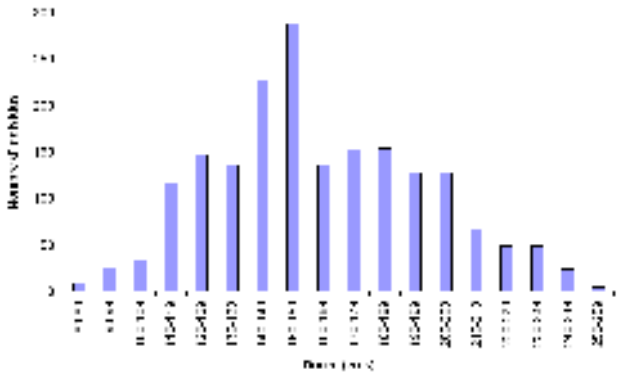


Figure 1 – *Distribution des valeurs pour une feuille de l'arbre représentant la météo.*

La figure 1 nous montre un exemple de la répartition des durées dans l'ensemble correspondant à la feuille représentant les bulletins météos. Nous faisons donc l'hypothèse que les distributions peuvent être approchées par une gaussienne asymétrique comme celle proposée dans [10]. La probabilité d'une durée $d \in \mathbb{D}$ peut être donnée par :

$$A(d, \mu, \sigma^2, r) = \frac{2}{\sqrt{2\pi}} \frac{1}{\sigma(r+1)} \begin{cases} e^{-\frac{(d-\mu)^2}{2\sigma^2}} \text{si d} > \mu \\ e^{-\frac{(d-\mu)^2}{2r^2\sigma^2}} \text{sinon} \end{cases} .$$

Dans cette expression, μ représente la valeur moyenne des durées retournée par la feuille de l'arbre de régression, σ est égale à la valeur absolue de la différence entre la valeur moyenne et la valeur maximale est r est donné par la relation :

$$r = \left| \frac{\mu - \text{valeur minimale}}{\mu - \text{valeur maximale}} \right| .$$

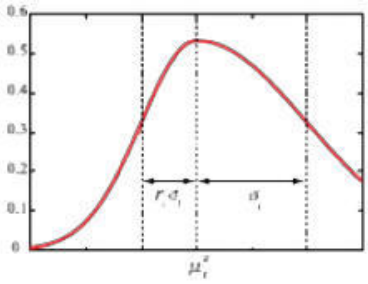


Figure 2 – *La distribution gaussienne asymétrique selon [10].*

5 Expérimentations

Nous avons procédé à un apprentissage de l'arbre de décision, puis du modèle markovien en utilisant les grilles de programmes de 2003 et 2004 de France 2, dont l'Institut Nationnal de l'Audiovisuel est en possession. La base de données n'a pas été filtrée. Par exemple, nous n'avons pas distingué les vacances scolaires ou l'été de la période normale. Nous n'avons retiré de la base que des problèmes techniques, en laissant volontaire du bruit comme par exemple les jeux Olympiques d'été. Des résultats meilleurs pourraient sans doute être obtenus après filtrage de la base d'apprentissage et de la base de tests.

5.1 Test de l'arbre de régression

Nous avons procédé à un test sur l'ensemble des durées de 2005. Il est possible qu'un individu présenté à l'arbre de décision n'aboutisse pas à une feuille. Dans ce cas là, l'individu est considéré comme inconnu et lui sont affectés la durée moyenne, minimale et maximale du dernier noeud qu'il a franchi. Nous considérons comme erreur uniquement les individus dont la durée réelle n'est pas comprise entre les durées minimales et maximales prédites.
De plus, il est souhaitable que les durées soient encadrées le plus finement possible, ce qui peut être obtenu en donnant des valeurs optimales aux paramètres ρ et λ.

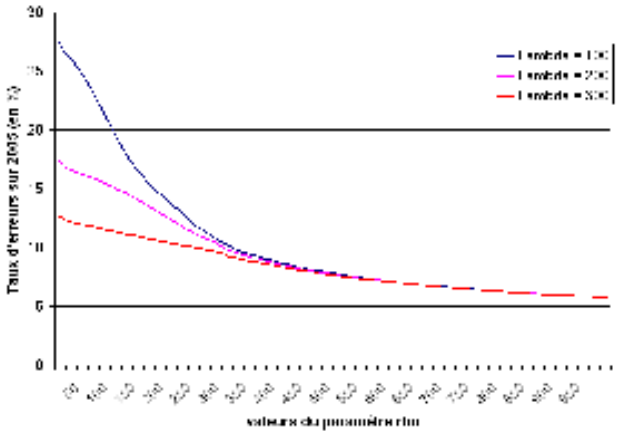


Figure 3 – *Impact du paramètre ρ sur le taux d'erreurs de prédiction pour des arbres entraînés avec différentes valeurs de λ.*

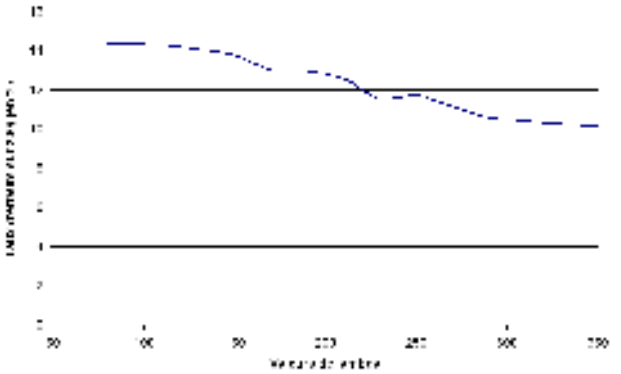


Figure 4 – *Impact du paramètre λ sur le taux d'erreur avec $\rho = 200$.*

Les figures 3 et 4 confirment que pour baisser le taux d'erreurs de prédiction sur 2005, il faut accepter d'encadrer ces durées moins finement. Avec $\rho = \lambda = 300$, il est possible d'obtenir 90,5% de bonnes prédictions avec des encadrements des durées d'envergure 200 secondes en moyenne.

5.2 Expressivité du modèle de Markov

En utilisant un arbre de régression avec les paramètres $\rho = \lambda = 300$ pour l'entraînement du modèle semi-Markovien contextuel, nous l'avons mis à l'épreuve sur le mois de Mars 2005. L'année 2005 est en effet une mauvaise année pour tester un modèle statistique car elle a connu plein d'évènements (mort du pape, référendum pour la constitution) qui ont bouleversé particulièrement les chaînes nationnales, dont fait partie France 2. Il était nécessaire de tester le modèle sans tenir compte de la prédiction des durées. Nous avons pour cela présenté une à une les émissions du mois de Mars en mettant à jour le contexte avec la durée réelle de l'émission. 83% des journées sont *observables* par le modèle. Les journées qui ont été comptées comme des erreurs contenaient toujours une émission déplacée dans la journée ou un communiqué de la chaîne (seuls les évènements exceptionnels ont été retirés de la base de tests). A chaque transition se fait d'un état vers au plus 6 états contre 38 états au plus avec un modèle non contextuel.

6 Conlusion

Nous avons présenté dans cet article un moyen de prédire des séries temporelles irrégulières dans le but de pouvoir prédire des grilles de programmes de chaînes de télévision. Nous avons souligné la difficulté de la tâche qui nécessite non seulement la prédiction des valeurs, mais aussi des dates.
Les résultats obtenus avec des méthodes statistiques et des méthodes de régression symbolique simples sont fortement encourageant même si tout apprentissage a ses limites.
Nous poursuivons nos travaux en essayant de prendre en compte cette fois-ci les grilles prévisionnelles. Nous essayons de ramener notre problème à une prédiction de séries temporelles irrégulières compte tenu d'une série temporelle *tuteur* qui est une observation partielle et imprécise de celle-ci.

Références

[1] Cees G.M. Snoek et Marcel Worring. Multimodal video indexing : A review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1) :5–35, 2005.

[2] C. Lee Giles, Steve Lawrence, et Ah Chung Tsoi. Noisy time series prediction using a recurrent neural network and grammatical inference. *Machine Learning*, 44(1/2) :161–183, July/August 2001.

[3] Eric Perlman et Akshay Java. Predictive Mining of Time Series Data in Astronomy. *Astronomical Data Analysis Software and Systems XII ASP Conference Series*, 295 :431–434, January 2003.

[4] Bo Xu et Ouri Wolfson. Time-series prediction with applications to traffic and moving objects databases. Dans *MobiDe '03 : Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access*, pages 56–60, New York, NY, USA, 2003. ACM Press.

[5] Emre Erdogan, Sheng Ma, Alina Beygelzimer, et Irina Rish. Statistical models for unequally spaced time series. Dans *Proceedings of SIAM Data Mining 2005*, 2005.

[6] C. Lee Giles, Steve Lawrence, et Ah Chung Tsoi. Noisy time series prediction using a recurrent neural network and grammatical inference. *Machine Learning*, 44(1/2) :161–183, July/August 2001.

[7] S. Mukherjee, E. Osuna, et F. Girosi. Nonlinear prediction of chaotic time series using support vector machines. Dans J. Principe, L. Giles, N. Morgan, et E. Wilson, éditeurs, *IEEE Workshop on Neural Networks for Signal Processing VII*, page 511. IEEE Press, 1997.

[8] Gerhard Dangelmayr, Sabino Gadaleta, Douglas Hundley, et Michael J. Kirby. Time series prediction by estimating markov probabilities through topology preserving maps. Dans Bruno Bosacchi, David B. Fogel, et James C. Bezdek, éditeurs, *Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation II*, volume 3812, pages 86–93. SPIE, 1999.

[9] J. R. Quinlan. Learning with continuous classes. Dans *Proceedings of 5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, 1992.

[10] T. Kato, S. Omachi, et H. Aso. Asymmetric gaussian and its application to pattern recognition. Dans *Lecture Notes in Computer Science (Joint IAPR International Workshops SSPR 2002 and SPR 2002)*, volume 2396, pages 405–413, 2002.

Joint informed embedding and spread spectrum video watermarking

Sorin Duță[1], Mihai Mitrea[1,2], Françoise Prêteux[1]

[1] ARTEMIS Department, GET/INT
9, Rue Charles Fourier, 91011 Evry Cedex
[2] Faculty of Electronics and Telecommunications, POLITEHNICA University of Bucharest, Romania

{sorin.duta,mihai.mitrea,francoise.preteux}@int-evry.fr

Abstract

In the Internet era, any piece of digital/digitalised art (be it image, video, audio, 3D ...) can be anytime and anywhere replicated with a simple click, thus frustrating the artists/producers from a large part of their economic benefits. Imposing itself as a viable solution to the copyright enhancement problem, watermarking represents a research field which exploded in the last decade.
The present paper reports on an original video watermarking method for very low rate video based on: (1) the synergy between spread spectrum and informed embedding approaches (patent pending) and (2) an accurate statistical modelling of some real-life attacks. The detection is oblivious (it does not require the unmarked object). Firm results concerning transparency (no visible differences between the marked and the unmarked video) and robustness (with respect to both mundane transforms, as the compressions, and malicious attacks, as the StirMark attack) are obtained. The data payload is increased more than 10 times with respect to the state-of-the-art. Beyond traditional watermarking, our method can be applied for emerging enriched multimedia applications: interactive television, video on demand, scalable enriched content streaming, and adaptive indexing.

Key words

Low rate video watermarking, robustness, transparency, informed embedding, spread spectrum.

1 Introduction

When considering the Information Society in general and the Internet in particular, the art producers find themselves in a quite awkward position. On the one hand, a digital dimension is added as a completing element giving art a whole new perspective. On the other hand, this very dimension opens the door to author spoliation: any piece of digital/digitalised art (be it image, video, audio, 3D …) can be anytime and anywhere replicated with a simple click. For instance [1], in 2004, the DVD piracy summed up to a total amount of $ 512 billion. When expressing this phenomenon in terms of markup [1], it turns out to be more "profitable" than cocaine traffic, Fig. 1.
Watermarking is the potential solution to such a problem. It provides a mean to persistently associate copyright information with virtually any digital representation. Despite its very short history, we already dispose of a sound theoretical background [2-5] and of various derived technical solutions.
The present paper reports on an original watermarking method aimed to protect the property rights connected to video distribution in mobile networks. Section 2 presents the watermarking main definitions. Section 3 describes a new watermarking scheme (patent pending) also relying on an accurate statistical analysis of some real-life attacks (*i.e.* rotation, linear filtering, StirMark) described in Section 4. Section 5 experimentally supports our method while Section 6 concludes the paper and opens the perspectives for our future work.

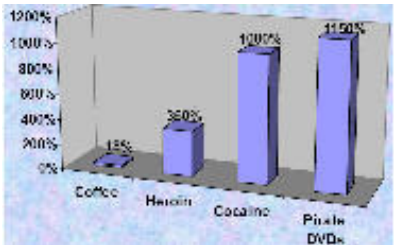


Figure 1 - *A comparison between four of the most profitable black market products*

2 Watermarking definitions

In its largest acceptation, *watermarking* stands for the practice of imperceptibly modifying an original piece of media in order to embed a message.
This embedded message is referred to as *mark* or *watermark*. Generally, it conveys copyright information and should be generated starting from some secret information referred to as *key*. According to the targeted application, the size (in bits) of the copyright information (*the data payload*) may vary.
When the embedded message does not alter the visual quality of the considered object, the watermarking procedure features *transparency*.
The *robustness* refers to the ability of the watermark to survive signal processing operations. Two classes of such operations should be considered. The first class contains the common transformations applied to the video sequence, *e.g.* compression, change of file format, *etc*. The second class is represented by *the attacks*. These are

malicious transforms designed to make the watermark detection unsuccessful while preserving a good visual quality for the considered video.
When the unmarked video is not required during the detection procedure, the method is *oblivious.*
The probability of false alarm expresses the probability of taking an unmarked object for a marked one. Its upper limit is application dependent.
By summarising these four requirements, it can be noticed that they are contradictory, *e.g.* the better the transparency the weaker the robustness. Hence, for each and every method, a trade-off among them should be reached.
The communication theory represents a generic framework for the watermarking applications, Fig. 2. The mark is generated starting from the message to be embedded and from the key. At the detection side, the message should be recovered. Hence, it represents a sample from the information source. Every factor that makes it difficult for the message to be recovered (the original video itself and the transformations applied to the marked video) is considered as noise.

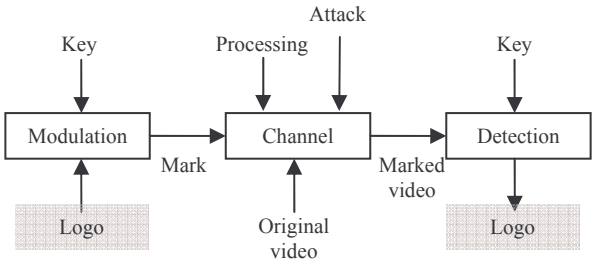


Figure 2 - *Watermarking application as a noisy channel*

From the owner point of view, the mark is corrupted by three noise sources: the original video, the distortions and the attacks. However, despite this theoretical model, a human observer (the person who buys the video) is interested in that video. From his/her point of view, the mark is responsible for the artefacts induced in the marked video. In order to keep these artefacts as un-disturbing as possible, the mark should have a very low power. Hence, a watermarking technique should allow the reliable detection of a very low power signal. Under the communication framework, the Spread Spectrum (SS) techniques have already offered good solutions to such problems. In its largest acceptation, an SS technique represents a communication technique in which the information is transmitted into a very large band. Consequently, an SS watermarking technique is a method in which the mark is spread over the original video, while occupying a much larger band than strictly necessary. In practice, the SS watermarking techniques feature good robustness but a quite small data payload.
The informed embedding (IE) is a different approach which exploits the knowledge about the original video in order to optimise the embedding procedure, *i.e.* it takes advantage on the fact that the main noise component is known at the embedder. From the theoretical point of view [6, 7], such a noise should not alter the channel capacity (*i.e.* the maximal amount of information which can be *theoretically* transmitted through the channel). Hence, the informed embedding approach is *a priori* very promising, at least from the theoretical point of view. In practice, they allow an impressive data payload but, generally, a quite poor robustness.
The watermarking procedure we designed (Section 3) synergistically combines the SS and IE principles, in order to reach the trade-off between data payload and robustness.

3 Method presentation

In order to pass from some theoretical concepts to a real life application, this paper adapts and extends the principles in [8] and also takes into account the results of the statistical investigation in Section 4.

☛ Mark modulation

The M bits corresponding to the logo to be inserted are encoded according to the SS principles, by means of a modified trellis code, [8], [9].
The trellis has K states and 2 arcs exiting each state (each transition codes one bit). Each arc is labelled with an N length vector which components are real numbers.
These labels are computed starting from the key, *i.e.* they are known only by the true object owner.
The mark thus obtained is a vector denoted by g, with $M \times N$ real number components.

☛ Video representation

The mark is inserted into a vector of salient characteristics of the video sequence, obtained as follows.
Be there a colour video sequence consisting in L frames. Each frame is represented in the HSV (hue-saturation-value) space [10]; the V component is normalised to [0,1] interval.
For each frame in the video sequence, the 2D-DWT is applied to the V component, at an N_r resolution level. The DWT coefficients corresponding to the LH and HL lowest frequency sub-bands, Fig. 3, are recorded into a vector. The coefficient hierarchy is built up by sorting in a decreasing order the vector previously obtained and the largest $M \times N$ rank values alongside with their original locations are recorded into two vectors, denoted by c_0 and λ, respectively.

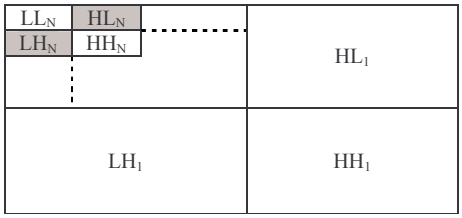


Figure 3 - *The chosen sub-bands*

☛ ***The detection***

We aim at establishing whether a suspicious video sequence was marked or not, *i.e.* finding out whether it contains the M bit logo or not.

We first extract from the video the vector susceptible to convey the mark. In this respect, the DWT is applied as above. Then, the coefficients corresponding to the λ locations are recorded, thus obtaining a $\hat{c}_W$ vector with $M \times N$ real components.

This vector is the input of a Viterbi decoder [9]. The decoder is pair designed with the trellis encoder. The cost involved in the Viterbi algorithm is the (un-normalised) correlation coefficient between the input sequence and the transition labels. This cost is to be maximised. High performances are obtained for uncorrelated labels.

☛ ***Informed embedding***

Designed by adapting the principles in [8], the embedding procedure aims at finding a c_W vector which is as close as possible to the c_0 vector and for which the Viterbi decoder produces the same output as for the g vector.

This c_W vector is iteratively computed, Fig. 4.

In the first iteration, c_W is initialised with c_0. Further on, a vector denoted by b is computed by applying the Viterbi decoder to $c_W + n$, and by trellis encoding the resulting bits. Here, n is a vector of $M \times N$ length, whose components are sampled from a noise source modelling the channel perturbations. Section 4 presents an accurate statistical investigation on the noise behaviour.

The c_W vector is now modified according to the following formula:

$$c_W \leftarrow c_W + \alpha \cdot (g - b) \big/ |g - b| .$$

The α scalar value is computed as follows:

$$\alpha = R_t - R(g, b, c_W) ,$$

where $R(g, b, c_W) = c_W \cdot (g - b) \big/ |g - b|$ and R_t is a scalar. The dot product between the c_W and the $(g - b)$ vectors is the un-normalised correlation coefficient.

The loop of b computation and c_W modification is repeated until the condition $R(g, b, c_W) \geq R_t$ is reached several times successively (*e.g.* 100 times – $N_j = 100$). If the equality between the g and the b vectors is reached before the $R(g, b, c_W) \geq R_t$ condition is achieved, then the b vector is computed without modifying c_W. If such a situation is encountered many times successively (*e.g.* 100 times – $N_i = 100$) then we consider that the g mark was successfully embedded into the c_W vector: regardless the added noise, the decoder can recover the message.

The c_W vector thus computed replaces the c_0 salient vector in the original video.

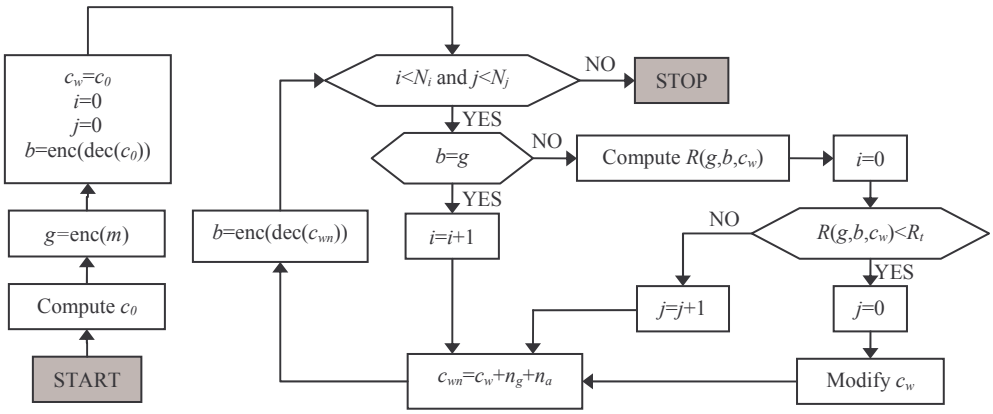


Figure 4 - *The informed embedding algorithm*

4 Noise statistical investigation

Concerning the effects of the video processing techniques and attacks, no reliable statistical investigation procedure has yet been advanced. However, in the literature, the popular AWGN (additive white Gaussian noise) model is assumed [2-5]. In order to find out whether this general assumption does hold in the particular case of our method (*i.e.* when embedding the mark into the DWT coefficient hierarchy) we reconsidered a statistical approach which already proved its efficacy in watermarking [11, 12].

The algorithm is structured in two parts. Steps 1 - 6 start from the attacked and original videos (2D random process - r.p.) and defines the noise as a 1D r.p. Steps 7 - 12 represent the statistical investigation on this r.p. Be there the same colour video sequence consisting in L frames represented in the HSV space, with the V component normalised to [0,1] interval.

Noise_modelling algorithm

For each frame in the video sequence:

Step 1: Apply the 2D-DWT to the V component, at an N_r resolution level.

Step 2: Record in a vector the DWT coefficients corresponding to the LH and HL lowest frequency sub-bands, see Fig. 3.

Step 3: Build up the coefficient hierarchy by sorting in a decreasing order the vector obtained in the previous step; record the largest N rank values alongside with their original locations; the resulting vectors are denoted by c_0 and λ, respectively.

Step 4: Apply the chosen attack to the considered frame.

Step 5: Resume the Steps 1 and 2 on the attacked video sequence; record the coefficients corresponding to the λ positions, thus obtaining the c_a vector.

Step 6: Compute the difference between the two vectors: $n = c_a - c_0$.

For each rank in the hierarchy and for all frames (i.e. for each component in all n vectors):

Step 7: Record in an x vector the values taken by a chosen r rank, $r \in [1, N]$, along all the L frames. In other words, the x vector is obtained by concatenating the r^{th} component from each of the L noise vectors.

Step 8: Periodically sample the x vector. When the D sampling period is large enough, by shifting the sampling origin, a partition into D classes with L/D independent elements is obtained.

Step 9: Verify whether all classes in the partition obey to the Gaussian law. In this respect, a Chi-square test [13] is run on each class. The ratio of the number of tests which are not passed to the D value can be considered as a measure of the overall Gaussian behaviour.

Step 10: Verify whether the D sampling period is large enough so as to afford the independence among the data in a partition class. Therefore, we run a Ro test on correlation [13] (for Gaussian data, the correlation and the independence are equivalent). As in the previous step, the relative number of tests which were not passed (*i.e.* the ratio of the tests which were not passed to the D value) is computed.

Step 11: A homogeneity investigation on the data in the same partition class is carried out by combining the Fisher F test and the Student T test [14].

Step 12: A homogeneity investigation among the D partition classes is performed by combing the same F and T tests.

5 Experimental results

5.1 The statistical investigation on attacks

The corresponding experiments have been carried out on 10 video sequences of about 25 minutes each (a number of $L = 35000$ frames in each sequence). These sequences were coded at a very low rates (64kbit/s, 192x160 pixel frames), as imposed by the mobile networks constraints.

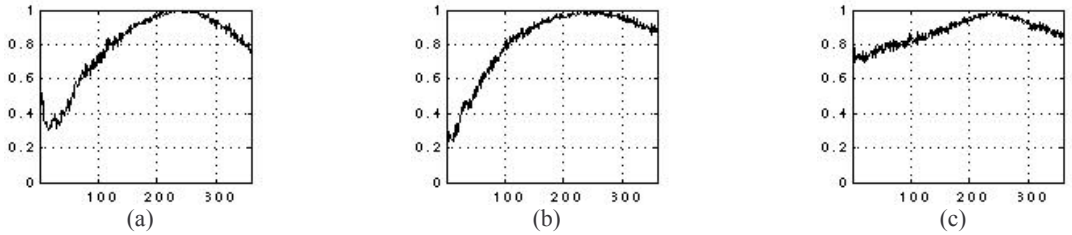
The (9,7) bi-orthogonal 2D-DWT [15] was applied at an $N_r = 3$ resolution level. The first $N = 360$ ranks of the hierarchy in each frame were considered. All the statistical tests were applied at an $\alpha = 0.05$ significance level.

The results obtained when applying the Step 9 are synoptically represented in Figs. 5, for three types of attacks: a rotation of 2 degrees (Fig. 5.a), a Gaussian linear filtering (Fig. 5.b), and the StirMark [16] attack (Fig. 5.c). The abscissa corresponds to the considered rank (from 1 to $N = 360$) while the ordinate depicts the ratio of the Chi-square tests which were not passed to the D sampling period (in these plots, $D = 550$ frames). Figs. 5 prove that for each type of attack and for each investigated rank, the Gaussian assumption is refuted: very large values are encountered for the rejecting ratio.

Figs. 6 are drawn for the same attacks and correspond to the Step 10. They *a posteriori* validate the $D = 550$ sampling period: each and every time, the ratio of refuted tests is about α. Note that the Ro tests are not properly run: they are meaningful only when the observation data are Gaussian distributed. However, as they are so nicely passed, we considered them as an additional (yet not theoretically sound) support for the sampling period.

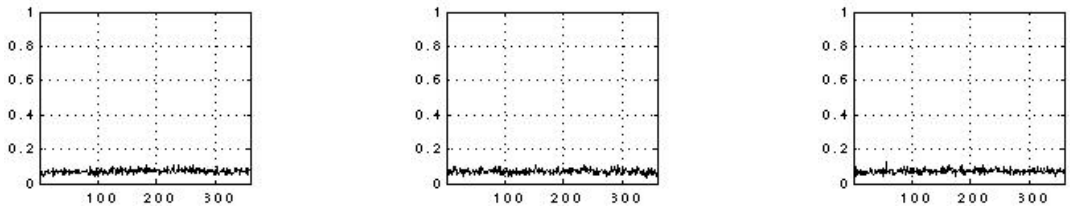
As the Chi-square tests were not passed, the Steps 11 and 12 become meaningless.

When inspecting Figs. 5 & 6, it becomes obvious that the Gaussian assumption does not hold for this application. Consequently, in order to obtain good watermarking performances, we considered in our method (Fig. 4) the very particular way in which some attacks act. When we tested the robustness against the attacks by computing the vector as a sum of the vector and of two noise sources: , corresponding to a Gaussian noise, and , corresponding to the StirMark attack. Note that the plots represented in Figs. 5&6 were obtained for a particular video sequence; however, the experiments resumed on the other 9 sequences led to the same general behaviour.



(a) (b) (c)

Figure 5 - *The Gaussian investigation on the noise r.p.: the ratio of Chi-square tests which are not passed to the D sampling period (ordinate) vs. the investigated rank (abscissa)*



(a) (b) (c)

Figure 6 - *The sampling period validation: the ratio of Ro tests which are not passed to the D sampling period vs. the investigated rank*

5.2 Robustness & transparency in practice

The watermarking experiments were run on 20 video sequences, each of them having 1000 frames (40 sec). These sequences are coded at 64 kbit/s and have 192×160 pixel frames.

The 2D-DWT is applied at a $N_r = 3$ resolution level.

The original message to be inserted is represented on $M = 1000$ bits and corresponds to the binary SFR logo (just for illustration see Fig. 9.a). Each bit from this message is trellis encoded by a $N = 360$ real number label. These numbers are extracted from a random generator obeying a Gaussian distribution of $\mu = 0$ mean and $\sigma = 0.005$ standard deviation.

The R_t parameter involved in the embedding scheme (Fig. 4) was set to $R_t = 2$.

In order to subjectively evaluate the transparency, 25 human observers of different ages were involved in our experiments: 5 researchers deeply involved in the image/video processing, 5 researchers working in fields not connected with video processing, 5 persons with various educational backgrounds (foreign languages, history, law), 6 students, 1 film director, 1 film producer and 2 painters. They agreed that the method features fidelity. In order to also offer an objective measure of the transparency, the UIQI (Universal Image Quality Index [17]) was computed for each frame in the video sequence: their minimal, maximal and mean values are 0.9798, 0.9994, and 0.9981 respectively (a UIQI of 1 corresponds to identical images). Frames from original and marked *Advertising* sequences are represented in Figs. 7 and 8.

The method also features very good robustness. First, we check up the resistance against the mundane video processing: change of file format (from mpg to avi), linear and non-linear filtering (Gaussian, Laplace, median), small rotations (each frame was randomly rotated up to 2 degrees), noise addition, spatial and temporal cropping (up to 25% of frames have been randomly dropped). Each and every time, the visual logo has been successfully recovered. Secondly, the StirMark attack was individually applied to each frame in the sequence: although the commercial value of the video sequence was completely destroyed during this attack, the logo was still recovered. Fig. 9 illustrates the robustness. The logos recovered after the file format changing, Laplace filtering and the StirMark attack are represented in Fig. 9 a, b, and c, respectively.

(a) (b) (c)

Figure 7 - *Original frames sampled from the Advertising sequence*

(a) (b) (c)

Figure 8 - *Transparency for video watermarking: marked frames sampled from the marked Advertising sequence, and corresponding to the originals in Fig. 7*

(a) (b) (c)

Figure 9 - *Robustness for video watermarking: the SFR logo recovered after the file format changing (a), Laplace filtering (b) and the StirMark attack (c). Note that the logo (a) is practically identical to the original logo*

6 Conclusion

The paper presents a video watermarking method (patent pending [18]) which synergistically combines the spread spectrum and informed embedding principles in order to reach the trade off between data payload and robustness: we inserted a binary logo into a very low rate video sequence of 40s and we recovered it after the StirMark attack. This means increasing data payload by more than 10 times as compared to the state-of-the-art.

The method is based on the original informed embedding scheme represented in Fig. 4 and on the attack statistical analysis described in Section 4.

Fig. 4 is generic enough so as to be applied to other media types; very good results were already obtained for 3D objects and for audio (speech, music) signals.

Section 4 proves that the popular Gaussian law cannot model all the attacks, at least not when inserting the mark in the DWT hierarchy. Consequently, we considered in Fig. 4 both a Gaussian and a StirMark noise generator. Moreover, Fig. 4 allows the user to add some extra noise generator, modelling emerging/future designed attacks.

To conclude with, beyond traditional watermarking (copyright protection), this method has the potential to be extended for emerging applications, such as: interactive television, video on demand, scalable enriched content streaming, and adaptive indexing.

Acknowledgement

This study was partly supported by the French SFR mobile services provider (Vodafone Group).

References

[1] International Intellectual Property Alliance. 2006 Special 301 Report on Global Copyright Protection and Enforcement. http://www.iipa.com/special301_TOCs/2006_SPEC301_TOC.html

[2] I. Cox, M. Miller, J. Bloom. *Digital Watermarking*. Morgan Kaufmann Publishers, 2002.

[3] M. Arnold, M. Schmucker, S. Wolthusen. *Techniques and Applications of Digital Watermarking and Content Protection*. Artech House, 2003.

[4] F. Davoine, S. Pateux. *Tatouage de documents audiovisuels numériques*. Lavoisier, 2004.

[5] S. Katenbeisser, F. Petitcolas. *Information Hiding – Techniques for Steganography and Digital Watermarking*. Artech House, 2000.

[6] C.E. Shannon. Channels with Side Information at the Transmitter. *IBM Journal*: pp. 289-293, Oct. 1958.

[7] M. Costa. Writing on dirty paper. *IEEE Transactions on Information Theory*, Vol. IT-29, pp. 439-441, 1983.

[8] M. Miller, G. Doerr, I. Cox. Applying informed coding and embedding to design a robust high-capacity watermark. In *IEEE Trans. on Image Processing*, Vol. 13, No. 6, pp. 792-807, 2004.

[9] S. Lin, D. J. Costello Jr. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[10] The MPEG-7 International Standard, Text of ISO/IEC International Standard 15938-3 – Information Technology – Multimedia Description Interface, Part 3 Visual, Geneva, Switzerland, September 2001.

[11] M. Mitrea, F. Preteux, A. Vlad, C. Fetita. The 2D-DCT Coefficient Statistical Behaviour: A Comparative Analysis on Different Types of Image Sequences. *JOAM*, Vol.6, No.1, pp. 95-102, 2004.

[12] M. Mitrea, F. Prêteux, M. Petrescu. Very Low Bitrate Video: A Statistical Analysis in the DCT Domain. *LNCS*, Vol. 3893, pp. 99-106, 2006.

[13] R.E. Walpole and R.H. Myres. *Probability and Statistics for Engineers and Scientists*. MacMillan Publishing, 1989.

[14] B.R. Frieden. *Probability, Statistical Optics and Data Testing*. Springer-Verlag, N.Y., 1983.

[15] A. Chalderbank, I. Daubechies, W. Sweldens, B. Yeo. Wavelet transforms that map integers to integers. *Appl. Comput. Harmon. Anal.*, Vol.5, No.3, pp. 332-369, 1998.

[16] F. Petitcolas, R. Anderson, and M. Kuhn. Attacks on copyright marking systems. *LNCS*, Vol. 1525, 1998.

[17] Z. Wang, A. Bovik. A Universal Image Quality Index. *IEEE Signal Processing Letters*, Vol. 9, No. 3, pp. 81-84, 2002.

[18] M. Mitrea, F. Prêteux, J. Nunez. *Procédé de Tatouage d'une Séquence Video*. French Patent Request No. 05 54132, deposed on December 29th, 2005, in the name of SFR and GET/INT.

Algorithme de tatouage basé sur le Prolongement Analytique de la Transformée de Fourier Mellin.

O. GUEMIR S. MHIRI F. GHORBEL

Groupe de Recherche en Images et Formes de Tunisie

Laboratoire CRISTAL
Ecole Nationale des Sciences de l'Informatique

{Faouzi.ghorbel, slim.mhiri}@ensi.rnu.tn
omar.guemir@cristal.rnu.tn

Résumé

Dans ce travail, nous présentons une nouvelle approximation du Prolongement Analytique de la Transformée de Fourier Mellin(PATFM). Ensuite, nous proposons un algorithme de tatouage basé sur le PATFM. cet algorithme est robuste vis-à-vis des transformations géométriques (essentiellement les rotations et les changements d'échelles) et des attaques désirant supprimer le marqueur de l'image tatouée. Enfin, une validation de la méthode d'approximation proposée sera effectuée à travers l'application de tatouage et comparée avec d'autre méthodes de la littérature.

Mots clefs

Prolongement Analytique de la Transformée de Fourier Mellin, Tatouage, Transformation géométrique, Nouvelle approximation PATFM.

1 Introduction

Le tatouage est exploité dans divers médias tels que le texte, le son, l'image et le vidéo. Il a ses contraintes et ses particularités pour chaque média. Dans notre étude nous traitons la protection de droit d'exploitation d'image contre les attaques. Hartung[1] classifie ces dernières en quatre catégories. La première ne modifie pas la géométrie de l'image tels que la compression *JPEG* ou le filtrage de l'image. La seconde regroupe les transformations géométriques tels que la rotation, le changement d'échelle. Elle perturbe la synchronisation du détecteur du marqueur. La troisième catégorie d'attaque consiste à introduire un deuxième marqueur par le pirate pour bruiter et bloquer l'identification de celui introduit par le propriétaire. La dernière est désormais la plus sophistiquée. Elle consiste à détecter le marqueur et le supprimer. Dans ce travail nous nous intersessions plus aux deuxième et quatrième catégories.

Dans la littérature, plusieurs travaux traitent l'amélioration de la robustesse contre les transformations géométriques. Une première approche utilise les points d'intérêts dans une image [2, 3]. L'avantage de cette méthode est sa robustesse aux distorsions locales. Mais elle présente une faiblesse au niveau du temps de traitement pour un nombre important de points. Une deuxième approche utilise un domaine d'extraction invariant par les distorsions géométriques[4]. O'Ruanaidh [5] et Lin[6] proposent l'insertion du marqueur dans le domaine de la transformée de Fourier Mellin. Cette transformée peut diverger. Pour résoudre ce problème, nous avons recours au prolongement analytique de la transformée de Fourier Mellin proposée par Ghorbel [7]. Le prolongement analytique résout ce problème au voisinage de zéro et permet ainsi une approximation numérique.

Dans un premier lieu, nous présentons la transformée de Fourier-Mellin ainsi que son prolongement analytique. Nous présentons en second lieu ses différentes approximations. Ensuite, une nouvelle approximation du PATFM est donné. Finalement, nous exposons la solution de tatouage proposée ainsi que les résultats des testes.

2 Prolongement Analytique de la Transformée de Fourier Mellin

2.1 Transformée de Fourier Mellin

La transformée de Fourier-Mellin standard se présente comme l'association des fonctions circulaires harmoniques et de la transformée de Mellin radiale. La représentation $\{r^{-iv}e^{-ikt}\}$ du groupe G est le produit des représentations individuelles des groupes des rotations et des homothéties vectorielles. Ce qui donne à la transformée un certain nombre de propriétés qui la rendent adaptée pour l'analyse des objets à niveaux de gris soumis à l'action des rotations et des dilatations. La TFM standard d'une fonction f est définie en coordonnées polaires, lorsqu'elle existe, est donnée par :

$$M_f(k,v) = \frac{1}{2\pi}\int_0^{+\infty}\int_0^{2\pi} f(r,\theta)\, e^{-ik\theta} r^{-iv} d\theta \frac{dr}{r} \quad (1)$$

L'espace des paramètres (k, v) de la TFM définit le groupe dual de $\mathcal{G}$: $\hat{\mathcal{G}} = \mathbf{Z} \times \mathbf{R}$.

La transformée de Fourier sur $\mathcal{G}$ existe et est appelée transformée de Fourier-Mellin inverse de f

$$f(r,\theta) = \int_{-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} M_f(k,v)\, e^{ik\theta} r^{iv} dv \qquad (2)$$

2.2 Prolongement Analytique de la Transformée de Fourier Mellin

La transformée de Fourier-Mellin d'un objet existe si sa représentation f est intégrable sur $\mathcal{R}_+ \times \mathcal{S}$, ce qui est traduit par :

$$\int_0^{+\infty} \int_0^{2\pi} \left| f(r,\theta)\, d\theta \frac{dr}{r} \right| < \infty \qquad (3)$$

Les fonctions représentantes des images à niveau de gris posent un problème au voisinage de zéro, où la fonction n'est pas intégrable dans la majorité des cas. Dans la littérature, deux solutions à ce problème ont été présentées. Une première [8] consiste à annuler un disque suffisamment petit au voisinage de zéro, ce qui induit généralement à une perte considérable d'information. Une deuxième solution introduite par Ghorbel [7] consiste à modifier la fonction f en introduisant le terme r^σ pour rendre la fonction intégrable au voisinage de zéro.

$$f_\sigma(r,\theta) = r^\sigma f(r,\theta) \qquad (4)$$

La transformée de cette nouvelle fonction f_σ est appelée prolongement analytique de fourier Mellin de f et elle s'écrit :

$$M_{f_\sigma}(k,v) = \frac{1}{2\pi} \int_0^{+\infty} \int_0^{2\pi} f(r,\theta)\, e^{-ik\theta} r^{\sigma - iv} d\theta \frac{dr}{r} \qquad (5)$$

La transformée inverse de la TFM de f_σ existe. Ainsi l'inverse du PATFM de f existe et s'écrit :

$$f(r,\theta) = r^{-\sigma} \int_{-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} M_{f_\sigma}(k,v)\, e^{ik\theta} r^{iv} dv \qquad (6)$$

3 Approximation de la PATFM

Les travaux de Derrode et Ghorbel[9, 10] ont abouti à l' approximation numérique de la PATFM par trois méthodes différentes. Ces approximations diffèrent par :
– La manière avec laquelle l'image est ré-échantillonnée.
– La méthode numérique utilisée.

3.1 Approximation directe par ré-échantillonnage polaire

La Méthode directe consiste à ré-échantillonner l'image $f(x,y)$ en coordonnées polaires puis à calculer une approximation directe de l'intégrale.
Le ré-échantillonnage est réalisé à l'aide d'une grille polaire formée de N cercles concentriques à M rayons. Les pas de ré-échantillonnage sont respectivement :

$$\Delta\rho = \frac{R}{N} \text{ et } \Delta\theta = \frac{2\pi}{M}.$$

Les fonctions harmoniques circulaires de l'image sont estimées par la transformée de Fourier discrète de chaque cercle :

$$\mathcal{F}_f(r,k) = \sum_{m=0}^{M-1} f(r,\theta_m) e^{-ik\theta_m} \qquad (7)$$

L'intégration sur les rayons est calculée en remplaçant l'intégrale de Mellin par une somme de Riemann. Ainsi, l'approximation de PATFM directe s'écrit :

$$\mathcal{M}_{f_\sigma}(v,k) = \frac{\Delta\rho\Delta\theta}{2\pi} \sum_{n=1}^{N} \mathcal{F}_f(\rho_n,k)(\rho_n)^{\sigma - iv - 1} \qquad (8)$$

avec $k \in [-K,K] et v \in [-V,V]$.

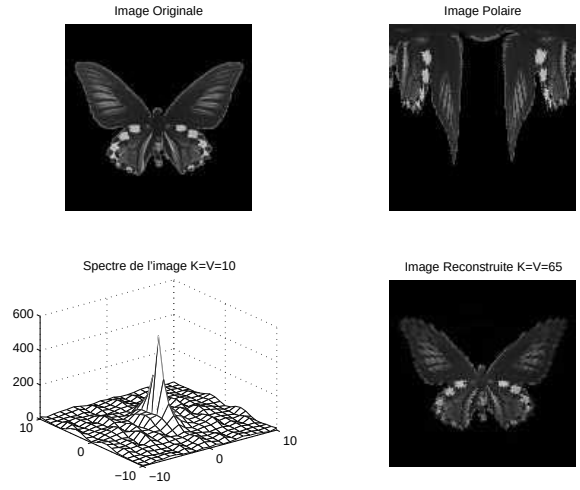


Figure 1 – *Approximation du PATFM Directe.*

La reconstruction de l'image $\hat{f}(p,q)$,à partir de sa représentation discrète du PATFM directe, est réalisée comme suit : Dans un premier lieu, les harmoniques circulaires sont approximées par une transformée de Fourier discrète inverse :

$$\mathcal{G}_\sigma(v,\theta_m) = \sum_{k=-K}^{K} \mathcal{M}_{f_\sigma}(v,k) e^{ik\theta_m} \qquad (9)$$

Dans un deuxième lieu, une approximation de l'intégrale de Mellin inverse est éffectuée[1]. Ainsi la transformée de Fourier-Mellin Analytique inverse en coordonnées polaire s'écrit :

$$\hat{f}(\rho_n,\theta_m) = \rho_n^\sigma \sum_{v=-V}^{V} \mathcal{G}_\sigma(v,\theta_m)(\rho_n)^{iv} \qquad (10)$$

La dernière étape de l'algorithme consiste à reconstruire l'image en coordonnées cartésiennes, en utilisant la grille définie précédemment.

[1] l'intégrale est approximée par une somme de Reimann

3.2 Approximation rapide par échantillonnage log-polaire

L'approximation rapide (PATFM-F) consiste à ré-échantillonner l'image $f(x,y)$ en coordonnées log-polaires puis estimer le PATFM par une transformée de Fourier. Cet algorithme est couramment utilisé pour l'estimation de TFM. Elle se base sur le fait que la TFM peut être écrite sous forme de la transformée de Fourier en effectuant le changement de variable $t = ln(r)$. Ainsi le PATFM-F s'écrit :

$$M_{f_\sigma}(k,v) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\int_{0}^{2\pi} f_\sigma\left(e^t,\theta\right) e^{-ik\theta} e^{(\sigma-iv)t} d\theta dt \tag{11}$$

De même la transformée inverse s'écrit :

$$f\left(e^t,\theta\right) = e^t\sigma \int_{-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty} M_{f_\sigma}(k,v)\, e^{ik\theta} e^{ivt} dv \tag{12}$$

Similaire à l'algorithme précèdent, le ré-échantillonnage est réalisé à l'aide d'une grille log-polaire formée de N cercles à M rayons. Les pas d'échantillonnage sont respectivement :

$$\Delta\rho = \tfrac{\ln(R_{max})-\ln(R_{min})}{N} \text{ et } \Delta\theta = \tfrac{2\pi}{M}.$$

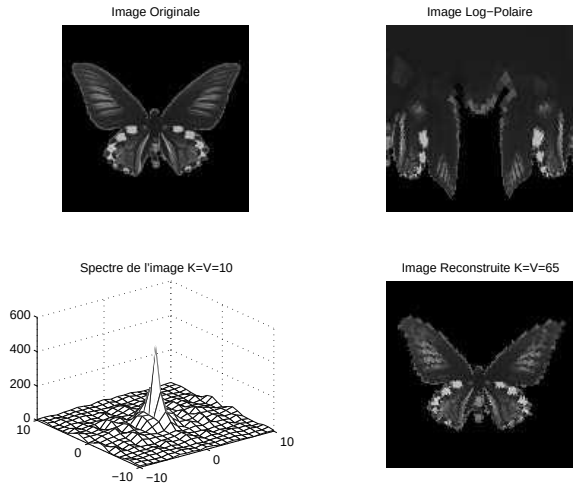


Figure 2 – *Approximation du PATFM Rapide.*

Ainsi, l'estimation du PATFM-F revient à l'estimation de la transformée de Fourier et s'écrit donc :

$$\mathcal{M}_{f_\sigma}(k,v) = \frac{\Delta\rho\Delta\theta}{2\pi}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f(e^{\rho_n},\theta_m) e^{\sigma\rho_n} e^{-(iv\rho_n+ik\theta_m)} \tag{13}$$

D'après l'équation 12, la PATFM-F inverse se calcule par une transformée de Fourier discrète bidirectionnelle. Ainsi elle s'écrit en coordonnées log-ploaires :

$$\hat{f}(e^{\rho_n,\theta_m}) = e^{-\sigma\rho_n}\sum_{v=-V}^{V}\sum_{m=-K}^{K}\mathcal{M}_{f_\sigma}(k,v) e^{(iv\rho_n+ik\theta_m)} \tag{14}$$

Similaire à l'algorithme de PATFM-D inverse, la derniére étape consiste à reconstruire l'image à partir de sa présentation en coordonnées log-plaires.

3.3 Approximation cartésienne

La troisième approximation, PATFM Cartésienne, ne nécessite pas un ré-échantillonnage préalable. Elle est obtenue en réalisant le changement de variable d'intégration suivant :

$$d\theta dr = \left|\begin{array}{cc}\frac{\partial r}{\partial x} & \frac{\partial\theta}{\partial x}\\ \frac{\partial r}{\partial y} & \frac{\partial\theta}{\partial y}\end{array}\right| dxdy = \frac{1}{(x^2+y^2)^{\frac{1}{2}}} dxdy \tag{15}$$

Le PATFM-C et son inverse sont donc donnés par les expressions respectives :

$$M_{f_\sigma}(k,v) = \frac{1}{2\pi}\int\int_{\mathbf{R}^2} f(x,y)\frac{(x^2+y^2)^{\frac{k+\sigma-2-iv}{2}}}{(x+iy)^k} dxdy \tag{16}$$

$$f(x,y) = \int_{-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty} M_{f_\sigma}(k,v)\frac{(x+iy)^k}{(x^2+y^2)^{\frac{k+\sigma-iv}{2}}} dv \tag{17}$$

L'approximation de la transformée et son inverse sont obtenues en remplaçant les intégrales par des sommes[2]. Anisi la TFMA-C et son inverse s'écrivent :

$$M_{f_\sigma}(k,v) = \frac{1}{2\pi}\sum_{y=Y_{min}}^{Y_{max}}\sum_{x=X_{min}}^{x_{max}} f(x,y)\frac{(x^2+y^2)^{\frac{k+\sigma-2-iv}{2}}}{(x+iy)^k} \tag{18}$$

$$f(x,y) = \sum_{v=-V}^{V}\sum_{m=-K}^{K} M_{f_\sigma}(k,v)\frac{(x+iy)^k}{(x^2+y^2)^{\frac{k+\sigma-iv}{2}}} dv \tag{19}$$

[2]Les intégrales sont approximés par la méthode de trapèze

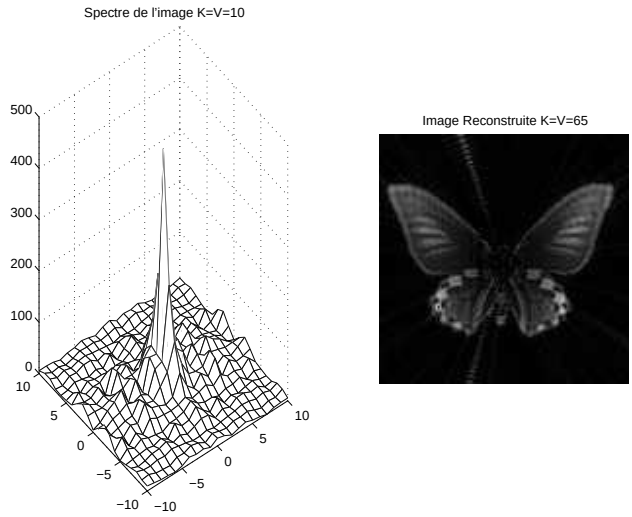


Figure 3 – *Spectre du PATFM Cartésienne.*

Les coordonnées sont centrées par le barycentre de l'image et X_{min}, X_{max}, Y_{min} et Y_{max} représentent le plus petit rectangle englobant l'objet.

4 Nouvelle Approximation du PATFM

Dans ce paragraphe, nous présentons une nouvelle approximation de PATFM. Elle exploite l'intégrabilité du PATFM au voisinage de zéro grâce au terme r^{σ} introduit par Ghorbel[7]. En exploitant le fait que les fonctions harmoniques circulaires d'une image peuvent être estimées par la transformée de Fourier discrète, l'équation (5) s'écrit alors :

$$M_{f_\sigma}(k,v) = \frac{\Delta\theta}{2\pi}\int_0^{+\infty} \mathcal{F}_f(r,k) r^{\sigma-iv}\frac{dr}{r} \qquad (20)$$

Nous écrivons l'intégrale(20) comme somme des intégrales sur des intervalles de la forme $[\rho_n, \rho_{n+1}]$:

$$M_{f_\sigma}(k,v) = \frac{\Delta\theta}{2\pi}\sum_{n=0}^{+\infty}\int_{\rho_n}^{\rho_{n+1}} \mathcal{F}_f(r,k) r^{\sigma-iv}\frac{dr}{r} \qquad (21)$$

Comme les fonctions harmoniques circulaires sont calculées à une ρ_n donnée, elles sont par conséquent constantes sur les intervalles de type $[\rho_n, \rho_{n+1}]$. L'équation (21) s'écrit alors :

$$M_{f_\sigma}(k,v) = \frac{\Delta\theta}{2\pi}\sum_{n=0}^{+\infty}\mathcal{F}_f(\rho_n,k)\int_{\rho_n}^{\rho_{n+1}} r^{\sigma-iv}\frac{dr}{r} \qquad (22)$$

La fonction $r^{\sigma-iv}$ est intégrable sur R_+^*, en particulier sur les intervalles de type $[\rho_n, \rho_{n+1}]$. L'idée est de calculer cette intégrale au lieu de l'approximer. L'expression du PATFM s'écrit donc :

$$M_{f_\sigma}(k,v) = \frac{\Delta\theta}{2\pi(\sigma-iv)}\sum_{n=0}^{+\infty}\mathcal{F}_f(\rho_n,k)[\rho_{n+1}^{\sigma-iv} - \rho_n^{\sigma-iv}] \qquad (23)$$

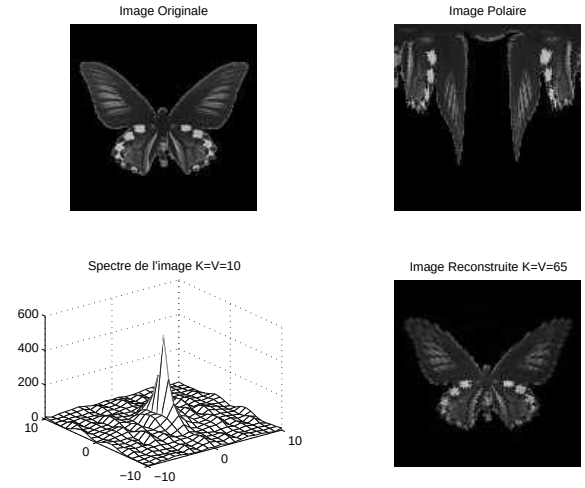


Figure 4 – *Nouvelle approximation du PATFM.*

Pour la reconstruction nous utilisons la même expression que celle donnée dans l'approximation directe.

5 Algorithme de tatouage basé sur Le PATFM

Cette section est consacrée à l'algorithme de tatouage. Nous y présentons son schéma. Nous montrons par suite la robustesse de cette solution vis-à-vis des transformations géométriques. Les résultats expérimentaux de la solution sont ainsi donnés.

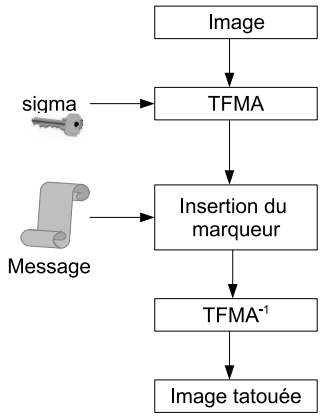


Figure 5 – *Schéma d'insertion du message*

5.1 Schéma de tatouage.

Dans notre solution, nous nous proposons d'insérer le marqueur dans le domaine de Fourier-Mellin. La figure 5

illustre le schéma de tatouage que nous proposons. La génération de l'image tatouée passe par trois étapes essentielles. Dans un premier temps, le PATFM de l'image est calculé en choisissant un clef bien précis (σ). Dans un second lieu, nous insérons le marqueur dans le spectre de l'image. Finalement, nous calculons l'image tatouée via la transformée inverse. L'extraction du marqueur de l'image tatouée, illustrée dans la figure 6, ne nécessite pas uniquement la connaissance préalable de l'endroit préalable d'insertion de ce premier mais aussi du paramètre σ pour calculer la PATFM qui convient au domaine d'insertion. Ce qui attribue à la solution une robustesse contre la quatrième catégorie d'attaque.

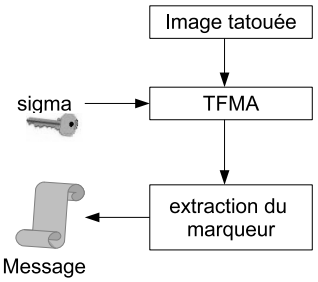


Figure 6 – *Schéma d'extraction du message*

5.2 Robustesse par translation.

Le calcul de la PATFM dans les quatre approximations présentées est fait par rapport au centre de gravité de l'image. Ce ci dit, toute translation T de l'image change les coordonnées de ses pixels par rapport au repère de l'écran. Par contre, elles restent invariantes par rapport au centre de gravité vu qu'il n'y a aucun changement de la structure de l'image. Pour garantir la robustesse, le calcul de centre de gravité est fait sur des points d'intérêts de l'image. Ce prétraitement permet ainsi de minimiser l'effet de la translation sur le calcul de la PATFM et garantie à la solution une robustesse par rapport la la translation.

5.3 Robustesse par Rotation et changement d'échelle.

Soient $f(r,\theta)$ la représentation de l'image dans un repère polaire, $R(\phi)$ la rotation planaire, $h(\alpha)$ une homothétie de facteur alpha et g l'image résultante après application de R et h.
$g(r,\theta) = f(\alpha r, \theta + \phi)$

$$M_{g_\sigma}(k,v) = \frac{1}{2\pi}\int_0^{+\infty}\int_0^{2\pi} f_\sigma(\alpha r, \theta+\phi)\, e^{-ik\theta} r^{-iv} d\theta \frac{dr}{r} \tag{24}$$

$$M_{g_\sigma}(k,v) = \alpha^\sigma e^{-ik\phi} M_{f_\sigma}(k,v) \tag{25}$$

$$|M_{g_\sigma}(k,v)| = \alpha^\sigma |M_{f_\sigma}(k,v)| \tag{26}$$

L'équation (26) garde le facteur d'échelle. Pour éliminer ce facteur d'échelle nous divisons tous les harmoniques par l'élément $M(0,0)$ qui est un réel non nul [7, 9]. En insérant ainsi l'information dans la norme du spectre nous garantissons son invariance par rapport à la rotation et au changement d'échelle.

5.4 Résultats

Dans cette section, nous présentons les résultats expérimentaux de l'algorithme par les différentes approximations du PATFM. La figure 7 est obtenue par $K = V = 65$ et $\sigma = 0.5$. Les figures (8, 9, 10) sont obtenues pour $K = V = 128$, $\sigma = 0.5$. Le message caché est un cercle. Le résultat de la nouvelle implémentation du PATFM donne un meilleur résultat au niveau de l'extraction du message et l'invisibilité de l'information sur l'image tatouée.

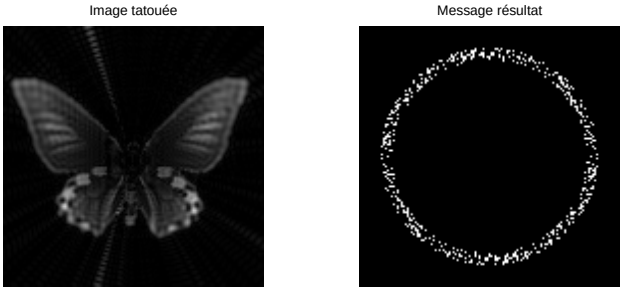


Figure 7 – *Image tatouée et le message extrait après détection par la méthode cartésienne .*

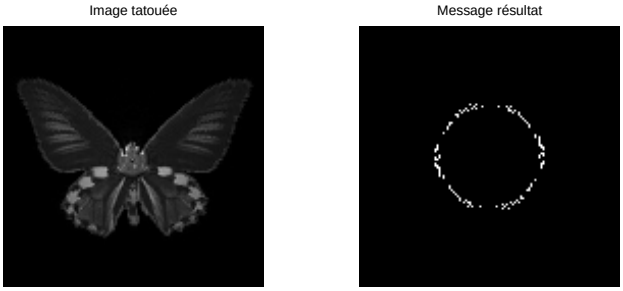


Figure 8 – *Image tatouée et le message extrait après détection par la méthode directe .*

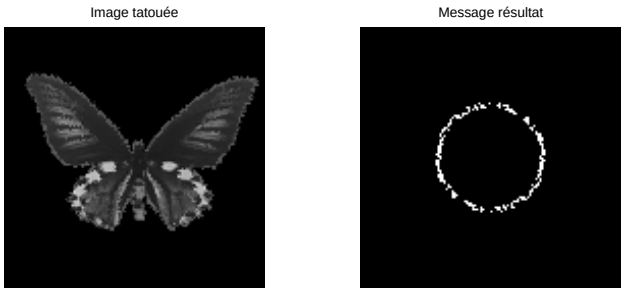


Figure 9 – *Image tatouée et le message extrait après détection par la méthode rapide .*

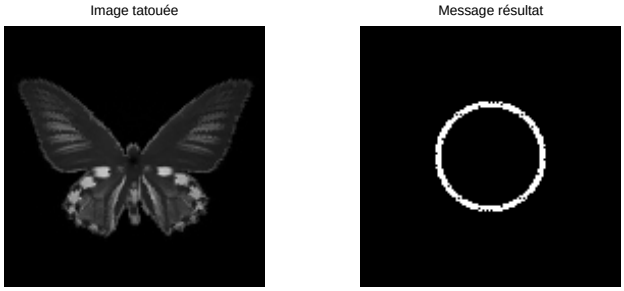


Figure 10 – *Image tatouée et le message extrait après détection par la nouvelle méthode.*

Nous avons fait subir à l'image tatouée un changement d'échelle égale à 2 et une rotation de 30 degrés. La figure (11) représente l'image résultante après cette attaque et le message extrait.

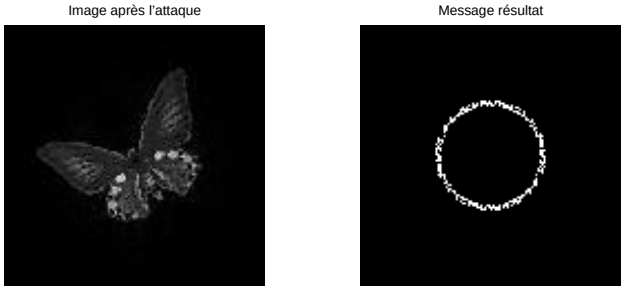


Figure 11 – *Image tatouée avec* $\theta = 30h = 2.0$ *et le message extrait après détection par la nouvelle méthode.*

6 Conclusion

Dans cette contribution, nous avons introduit une nouvelle approximation du PATFM et une solution de tatouage robuste contre les transformations géométriques. L'exploitation de la paramètre σ introduite pour la convergence de la TFM a permis de renforcer la sécurisation de l'information insérée. Il ne suffit donc pas de connaître l'emplacement de l'insertion dans le domaine fréquentiel mais il faut aussi avoir le paramètre σ utilisé.

Références

[1] J. K. Su F. Hartung et B. Girod. Spread spectrum watermarking : Malicious attacks and counterattacks. Dans *In Proc. SPIE Security and Watermarking of Multimedia Contents 99*, San Jose, CA, Janvier 1999.

[2] J.-M. Chassery P. Bas. Tatouage d'images résistant aux transformations géométriques. Dans *In Dix-septième colloque GRETSI 99*, pages 271–274, Vannes, Septembre 1999.

[3] Chassery J. M. P. Bas et B. Macq. Geometrically invariant watermarking using feature points. *IEEE Trans. on Image Proc*, 11(9), Septembre 2002.

[4] C. Delpha M. Ossonce et P. Duhamel. Rotation and scale insensitive image watermarking. Dans *In Proc. of the Intl Conf. on Image Processing*, 2004.

[5] J. J. K. O'Ruanaidh et T. Pun. Rotation, scale and translation invariant digital image watermarking. Dans *In Proc. IEEE ICIP*, pages 536–539, Octobre 1997.

[6] Jeffrey A. Bloom Ingemar J. Cox Matt L. Miller Ching-Yung Lin, Min Wu et Yui Man Lui. Rotation, scale, and translation resilient watermarking for images. *IEEE Trans. on Image Proc*, 10(5), 2001.

[7] JL. de Bougrenet de la Tocnaye et F. Ghorbel. Scale-rotation invariant pattern recognition applied to image data compression. *Pattern recognition letters*, 8(1) :55–58, Juillet 1988.

[8] Z. Kiss PE. Zwicke. A new implementation of the mellin transform and its application to radar classification. *IEEE Trans. on PAMI*, 5(2) :191–199, Mars 1983.

[9] S. DERRODE et F. GHORBEL. Robust and efficient fourier-mellin transform approximations for invariant grey-level image description and reconstruction. *Computer Vision and Image Understanding*, 83(1) :57–78, Juillet 2001.

[10] S. DERRODE et F. GHORBEL. Shape analysis and symmetry detection in gray-level objects using the analytical fourier-mellin representation. *Signal Processing*, 84(1) :25–39, Janvier 2004.

Tatouage informé pour le codage distribué

Cagatay Dikici, Khalid Idrissi, Atilla Baskurt

INSA de Lyon, Laboratoire d'InfoRmatique en Images et Systèmes d'information,
LIRIS, UMR 5205 CNRS, France

Concours jeune chercheur : Oui

Résumé

Nous partons de la dualité entre le tatouage et le codage distribué pour proposer un schéma qui exploite ces deux techniques dans un même système. Dans celui-ci, nous proposons de faire simultanément de l'insertion de message et de la compression en faisant appel au codage distribué. Pour la compression, nous utilisons des méthodes de code de correction d'erreurs comme LDPC, alors que pour l'insertion il sera fait appel à une méthode de quantification simple afin de ne pas augmenter la complexité du codeur. Des résultats expérimentaux pour des données de synthèse sont fournis, puis le système proposé est comparé à d'autres systèmes existants.

Mots clefs

Codage distribué, tatouage, LDPC

1 Introduction

La dualité entre le codage canal et le codage source est connue depuis quelques décennies [1] et les techniques cherchant à s'approcher de la capacité limite sont développées dans les deux cas. Pour le codage source, ou tout simplement la compression, les approches qui exploitent la redondance de la source, telle que le codage arithmétique [2] peuvent atteindre des taux proches des limites données par la théorie de l'information. De même pour le codage canal, les codeurs convolutionnels et les décodeurs itératifs ont permis [3] s'approcher des limites
Le codage source avec une information parallèle, désignée par Side Information (SI), disponible au décodeur a été étudié par Slepian Wolf [4] and Wyner Ziv[5], respectivement pour les cas sans et avec pertes. Les applications qui utilisent ce schéma datent de quelques années, et ont cherché à déporter la complexité vers le décodeur, afin de compresser à moindre coût en terme de puissance de calcul. De nombreuses approches pour le codage de sources distribuées (DSC) sont proposées telles que le codage par block [6], les turbo codes [7] ou les codes LDPC [8]
De la même manière, le codage canal avec une SI disponible au codeur a été étudié par [9] et [10]. La similarité entre ce schéma et le tatouage aveugle, dans lequel un message est transmis à travers un canal et où le signal support est disponible uniquement au codeur a été initialement abordée par [15]**.** Par la suite, des travaux ont concerné la dualité entre le codage source avec SI (SCSI) et le codage canal avec SI (CCSI) [11][12][13] puis de nombreux systèmes de tatouage informé exploitant cette dualité ont été proposés [14][15][16].

Dans ce papier, on se propose d'utiliser cette dualité pour réaliser un système de tatouage informé couplé avec du codage source distribué. Un message caché M sera alors inséré dans un signal hôte X puis compressé, sachant qu'un signal Y, corrélé avec X est disponible au niveau du décodeur.

Ce papier est organisé de la façon suivante: dans la Section 2 nous formalisons le problème du tatouage informé, puis la théorie du codage de sources distribuées est présentée en Section 3. Après une brève introduction sur la technique de quantification structurée basée sur les codes LDPC en Section 4, les détails de la méthode proposée sont présentés en Section.5. Enfin, la dernière section est consacrée à la discussion des premiers résultats obtenus, ainsi qu'à la comparaison avec des méthodes existantes.

2 Tatouage Informé

Le problème du tatouage aveugle peut être abordé comme du codage canal avec une SI au codeur tel que présenté en Figure 1. Le codeur a accès au signal de tatouage M, ainsi qu'au signal support X dans lequel l'information de tatouage va être insérée. Une contrainte de distorsion entre X et le signal tatoué W est fixée telle que $E[(X-W)^2] \le D_1$, avec $W = X + e$, et l'erreur e est dépendante de X et de M.Ensuite, le signal tatoué W peut être sujet à une distorsion à l'issue d'une attaque Z.
La capacité que l'on peut atteindre [9] pour un système de tatouage avec une probabilité d'erreur $P_e^n = \Pr\left\{\hat{M}(Y^n, X^n) \neq M\right\}$ est :

$$C_{10} = \max_{p(u,w|x)} [I(U;Y) - I(U;X)] \quad (1)$$

où U est une variable auxiliaire, où la maximisation porte sur toutes les fonctions de densité de probabilités conditionnelles $p(u,w|x)$ et où $I(U;Y)$ représente l'information mutuelle entre U et Y. Un taux R peut être atteint s'il existe une séquence parmi les $(2^{nR},n)$ codes avec $P_e \to 0$ [13].

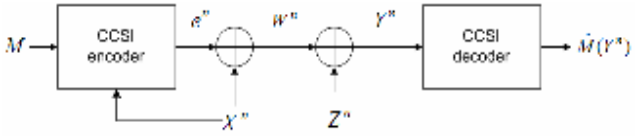


Figure 1. Codage Canal avec une information parallèle au codeur

3 Codage Distribué

Le codage de sources distribuées (DSC) peut être considéré comme un problème de Débit/Distorsion avec une SI disponible au décodeur tel que représenté Figure 2. La notation dans [13] est telle que l'indice 01 dans $RDSI_{01}$ indique la disponibilité de la SI au décodeur et non au codeur. Considérons $\{(X_k, Y_k)\}$ une séquence i.i.d. $\approx p(x,y)$ des variables aléatoires X et Y. X_k est codé sur un block de longueur n dans un flux binaire avec un débit utilisant une séquence parmi les $(2^{nR},n)$ codes avec $i: X^n \to \{1,2,..,2^{nR}\}$ et $\hat{X}^n : \{1,2,..,2^{nR}\} \to \hat{X}^n$. Le signal d'entrée X doit être codé et transmis au récepteur où l'on dispose de Y, une observation bruitée du signal d'entrée X et de $\hat{X}$ une estimation de X avec un critère de fidélité D_2 tel que $E[(\hat{X} - X)^2] \le D_2$. Le débit minimum de codage [5] pour un critère de fidélité donné D_2 est:

$$R_{01}(D_2) = \min_{\hat{X}=f(U;Y),p(u|x)} [I(U;X) - I(U;Y)] \quad (2)$$

Où la minimisation porte sur toutes les fonctions de densité de probabilités conditionnelles $p(u|x)$ et la fonction $f(U;Y)$ telle que $E(X - \hat{X})^2 \le D_2$. U étant une variable auxiliaire pour l'ensemble des mots-codes représentant X et $I(U;X)$ l'information mutuelle entre U et X.

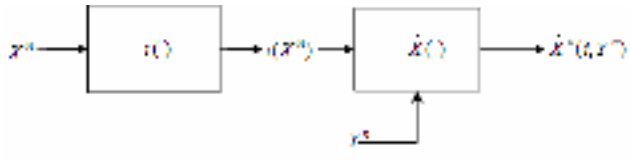


Figure 2.Codage Source avec une information parallèle en décodeur

4 Codage LDPC

Les limites théoriques présentées dans les Sect. 2 et 3 peuvent être atteintes dans le cas de codes à longueur infinie, en utilisant une technique de quantification aléatoire. Cependant, il ne semble pas réaliste d'utiliser cette approche, étant donné sa complexité et la longueur des blocs excessive nécessaire. En revanche, un codage canal adéquat, peut permettre de se rapprocher de ces limites.

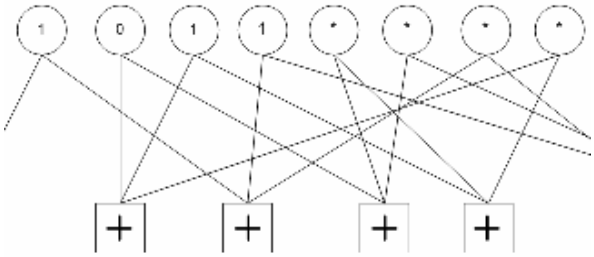


Figure 3.Le codage LDPC

Les codes LDPC initialement proposes par [17] et revus par [18] font partie des codes blocs ou les bits de contrôle sont intégrés au signal d'information de manière à détecter et corriger les erreurs introduites lors de la transmission. Il est fait appel à une matrice de parité H composée d'un faible nombre de 1. Les codes LDPC sont dits réguliers ou irréguliers selon que le nombre de bits de contrôle rajoutés est fixe ou non. Dans notre cas nous utiliserons le code régulier. La Figure 3 illustre un exemple de codage LDPC : Les cercles représentent les bits du message, tandis que les carrés représentent des nœuds de contrôle. Pour le codage, l'addition modulo 2 des bits reliés à chaque nœud doit être nulle. Une partie du bloc à coder est donnée en Figure 3, le but étant de trouver les bits de contrôle notés $*$. Ainsi pour le premier nœud, le bit de contrôle doit être à 1 alors que pour le deuxième, il doit être à 0. Ceci permet de trouve un mot-code unique.

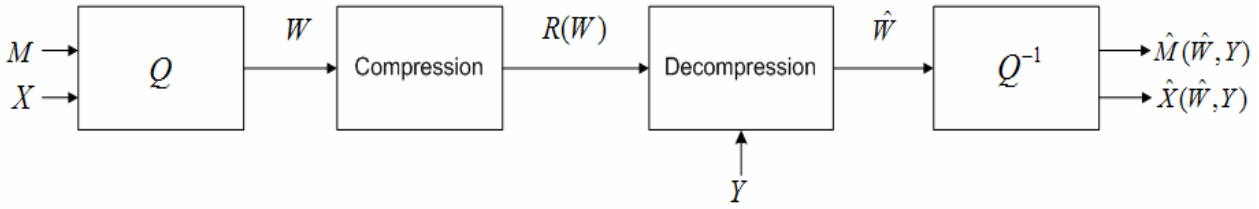


Figure 4. Le système proposé

Le décodeur basé sur un algorithme MAP est optimal pour décoder les codes LDPC. Mais, étant donné la longueur des blocs, il ne semble pas réaliste d'implémenter ce genre d'algorithme en raison de la complexité induite. Cependant, une bonne estimation $\hat{X}$ peut être obtenue en un algorithme de vraisemblance [3] comme dans le décodage LDPC classique [18].

5 Système proposé

Nous proposons un schéma hybride utilisant le codage canal au codeur et le principe de débit/distorsion avec SI au décodeur (Figure 4). Précisément, notre système permet de d'insérer un message M dans un signal hôte X avec une certaine distorsion D_1. Ensuite le signal tatoué W est compressé et transmis avec un critère de fidélité D_2, le récepteur disposant de Y, une observation bruitée de X.

Y est obtenue en simulant un canal binaire symétrique entre le signal d'entrée X et l'information parallèle Y. Le récepteur decode alors le signal reçu, en exploitant l'information Y avec un critère de fidélité D_2 tel que $E[(\hat{W}-W)^2] \le D_2$ et fournit une estimation du message inséré $\hat{M}$ avec une probabilité d'erreur $P_e(\hat{M})$.

Mathématiquement, le but est de résoudre le problème :

$$\min_{E[(X-W)^2 \le D_1, E[(\hat{W}-W)^2 \le D_2]} P_e(\hat{M}) \qquad (3)$$

où $P_e(\hat{M})$ représente la probabilité de l'erreur de décodage $\Pr\{\hat{M}(\hat{W},Y) \neq M\}$, et W, X, Y sont des variables aléatoires i.i.d. $\approx p(w,x,y)$.
De plus, la contrainte de distorsion $E[(\hat{W}-W)^2] \le D_2$ amène à une fonction de débit minimal:

$$R(D_2) = \min_{p(u|w,x)\,p(\hat{w}|u,y)} [I(U;X,W) - I(U;Y)] \qquad (4)$$

Ce problème hybride peut être présenté comme une forme de tatouage semi-aveugle, le récepteur n'ayant pas accès au signal d'entrée X pour extraire le message $\hat{M}$ depuis le message tatoué W, mais seulement à Y, observation bruitée de X.

6 Expérimentation

A ce stade, nous nous sommes intéressés à l'aspect théorique de la dualité codage canal avec SI et codage source avec SI. Dans cette section, nous présentons le système réel proposé, et qui implémente simultanément, l'insertion de message et la compression.

6.1 Les signaux d'entrée

Pour la simulation, des données de synthèse sont générées sous forme de vecteurs binaires i.i.d X et Y, respectivement disponibles au codeur et au décodeur, ainsi que pour le message M à insérer. L'information X est créée à partir d'une source de Bernoulli pseudo aléatoire ½ de longueur de block adaptée pour avoir $H(X) = 1$ bit/symbole.

L'information parallèle est donnée par $Y = X \oplus N$ où le niveau de corrélation N entre X et Y est un vecteur pseudo aléatoire de Bernoulli(p) de même longueur que Y et $\oplus$ est l'opérateur d'addition modulo 2. La variable $p : 0 \le p \le 1$ contrôle le niveau de corrélation tel que $H(X|Y) = H(p) = p \times \log_2(p) + (1-p) \times \log_2(1-p)$

6.2 Le Tatouage

Dans le cas d'un tatouage informé, où l'on insère M dans X, une quantification basée sur la construction de cosets est utilisée. L'algorithme se comporte de la manière suivante : 3 bits d'information sont partitionnés en 4 cosets tels qu'une distance de hamming de 3 existe entre chaque couple de cosets. L'élément du coset à utiliser sera choisi en fonction de la valeur des bits de M qui sont à insérer (ici par groupe de 2 bits, donc $Coset00 = \{000,111\}, Coset01 = \{001,110\}$, $Coset10 = \{010,101\}, Coset11 = \{011,100\}$). Une fois le dictionnaire créé, 2 bits de M et R bits de X sont considérés, formant un bloc de 2+ R bits. On prend les 3 bits de poids faible de ce dernier pour réaliser

l'insertion. Ces 3 bits sont quantifiés par $W : W(X, M) = \arg \min_{Z \in CosetM} \|Z - X\|$ où W a au plus 1 bit de différence avec X. La distance de hamming est choisie. Ce processus d'insertion de 2 bit dans un bloc de R bits est poursuivi jusqu'à épuisement des données de M. Un exemple est présenté en Figure 5 : si le message 10 doit être inséré dans les 3 bits de X de valeur 100, le minimum de distance de hamming entre 100 et les éléments du $Coset10$ est obtenu pour $W = 101$, ce qui sera retenu comme sortie. Au niveau du décodeur, l'extraction du tatouage est simple, car la connaissance du dictionnaire permet de retrouver le message inséré

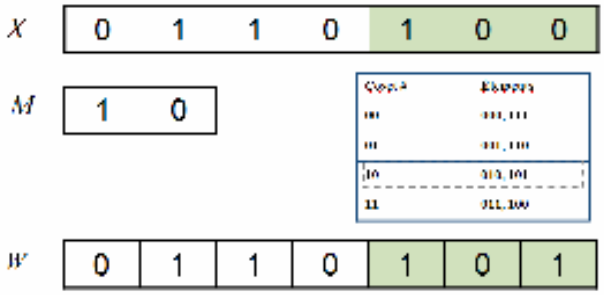


Figure 5. Un exemple d'insertion de tatouage

6.3 DSC

Cette partie décrit la compression du signal tatoué W utilisant la technique du codage de sources distribuées, sachant que l'information parallèle Y est connue uniquement au décodeur. Classiquement, les codes LDPC rajoutent de la redondance au signal d'entrée. Après le processus de codage, l'ensemble peut être décomposé en 2 parties, la partie systématique S_w contenant le signal d'origine et la partie de contrôle P_w contenant les cosets. En fait, après le codage du signal d'entrée, seule l'information de contrôle est transmise. Le Récepteur a accès à Y et aux bits de contrôle du signal tatoué P_w. Le but de décodeur est d'extraire les erreurs entre Y et W. Pour cela un algorithme de propagation de vraisemblance modifié, proche du codage LDPC standard [18].

Tout d'abord, les taux de vraisemblance des bits systématiques sont initialisés en adéquation avec le niveau de corrélation N entre X et Y. Ensuite, les taux de vraisemblance des bits de contrôle sont choisis sachant que la probabilité d'erreur en réception de ces bits est faible ε. De plus, la mise à jour des noeuds de contrôle est modifiée de manière à corriger les erreurs dans les bits systématiques, sachant que les bits de contrôle sont corrects avec une probabilité élevée. Enfin, la connaissance du dictionnaire des cosets et l'estimation de $\hat{W}$ à l'aide du décodage LDPC rend l'extraction du message $\hat{M}$ triviale. Les niveaux de distorsion obtenus sur $\hat{W}$ et $\hat{M}$ sont donnés dans les résultats.

7 Simulations

Dans l'expérimentation réalisée, 100 blocs de 4000 bits chacun, représentant le signal d'entrée X sont générés. L'information parallèle Y est créée comme cela a été décrit dans la Sec.6.1 avec une entropie conditionnelle $H(X|Y)$ entre 0 et $0{,}5$. Un premier test a consisté à évaluer les performances de la compression sans tatouage. Le signal d'entrée X est compressé à l'aide de LDPC avec un taux de 1/2 tel que décrit dans la Sec.6.3, puis les 2000 bits de contrôle de chaque bloc sont transmis. Au niveau du récepteur, le décodage est réalisé par LDPC à l'aide de l'information parallèle Y, en se limitant à 50 itérations.

Le système proposé a été comparé à d'autres systèmes existants basés sur le turbo code ou le LDPC régulier et irrégulier avec différentes longueurs de blocs [8][19] [20]. Par la suite, et toujours avec le même schéma, un message M est inséré avec un taux de $1/200$ pour différentes valeurs de $H(X|Y)$. Dans ce test, 2 bits de tatouage sont insérés dans chaque bloc de 400 bits X (cf Sec.6.2).

La Figure 6 représente la probabilité d'erreur entre le signal X et son estimation au décodeur $\hat{X}$ en fonction de l'entropie conditionnelle $H(X|Y)$, autrement dit, en fonction de la distorsion entre X et l'information parallèle Y. Le réseau de courbe est relatif à différentes méthodes de codage et à différentes longueurs de blocs. La limite théorique de Slepian Wolf pour un codage ½ est $H(X|Y) = 0{,}5$. La courbe relative à notre système (LDPC régulier avec une longueur de bloc de 4000 bits) atteint une erreur de 10^{-6} pour $H(X|Y) = 0{,}36$. La comparaison avec les autres méthodes montre que le turbo code [19] permet d'avoir le même résultat pour $H(X|Y) = 0{,}35$ et que le codage régulier LDPC avec une longueur de block de 10^4 [20] donne de meilleures performances que le système proposé. En revanche il nécessite une longueur de blocs 3 fois supérieure. Comme on peut le voir sur la courbe, le codage LDPC irrégulier avec des blocs de longueur 10^4 et 10^5 [8]

atteint le même niveau d'erreur pour des valeurs de $H(X|Y)$ respectivement de 0,42 et 0,45.

A noter que l'augmentation de la longueur des blocs entraîne un accroissement de la complexité et du temps de décodage.

La courbe en haut à gauche est relative au système complet, incluant tatouage et compression. Il faut bien voir que nous avons définit un critère de similarité entre le signal d'entrée X et le signal tatoué W, or, étant donné que c'est W qui est utilisé pour le décodage, l'erreur entre X et son estimation au décodeur $\hat{X}$ ne peut pas être inférieure à l'erreur de $W - X$. Ainsi, le système complet permet d'avoir une probabilité d'erreur de 5×10^{-3} pour $H(X|Y) = 0{,}34$. Le message M quant à lui, peut être retrouvé avec une probabilité d'erreur de 10^{-6} pour $H(X|Y) = 0{,}36$. D'autres tests seront effectués afin de comparer les méthodes entre elles, en utilisant les mêmes longueurs de blocs.

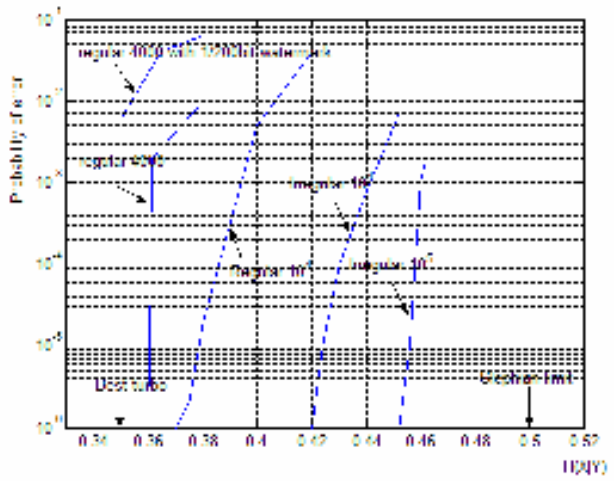


Figure 6 – la comparaison du system codage distribué

8 Conclusions

Un système hybride pour le tatouage et la compression, se basant sur la technique du codage distribué a été proposé. Les récents travaux sur la dualité entre le codage canal et la notion de débit/distorsion exploitant l'existence d'une information parallèle a été employé.

Le schéma de la Figure 4 permet de nombreuses possibilités d'usage. Les entrées et la nature du message M à insérer pouvant être choisis en fonction du problème à traiter. Le système proposé a été comparé à des systèmes existants utilisant diverses méthodes de codage. D'autres approches peuvent être envisages (tatouage à base de treillis, ou de quantification LDPC) pour améliorer les performances. Ce système peut également être facilement adapté à la vidéo, en utilisant la corrélation entre les images successives.

Références

[1] C. E. Shannon, Coding theorems for a discrete source with a fidelity criterion , IRE Nat. Conv. Rec., vol. Part 4, pp. 142-163, 1959.

[2] Elias, P. (1975) Universal codeword sets and representations of the integers. IEEE Trans. Info. Theory 21 (2): 194-203.

[3] Berrou, C., and Glavieux, A., Near optimum error correctingcoding and decoding: Turbo-codes. IEEE Trans. On Communications 44: 1261-1271,1996.

[4] J. D. Slepian and J. K. Wolf, Noiseless coding of correlated information sources , IEEE Transactions on Information Theory, vol. IT-19, pp. 471-480, July 1973.

[5] A. D. Wyner and J. Ziv, The Rate-Distortion Function for Source Coding with Side Information at the Decoder , IEEE Transactions on Information Theory, vol. IT-22, no. 1, pp. 110, Jan. 1976.

[6] S.S. Pradhan, K. Ramchandran, Distributed Source Coding Using Syndromes (DISCUS). *IEEE Transactions on Information Theory*, vol. 49, no. 3, March 2003. IEEE, USA.

[7] B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero , Distributed video coding, *Proceedings of the IEEE, Special Issue on Video Coding and Delivery*, vol. 93, no. 1, pp. 71-83, Jan. 2005.

[8] Z. Xiong, A. Liveris, and S. Cheng, Distributed source coding for sensor networks, IEEE Signal Processing Magazine, vol. 21, pp. 80-94, September 2004

[9] S. Gel'fand and M. Pinsker, Coding for channel with random parameters , Problems of Control and Information Theory, vol. 9, pp. 19-31, 1980.

[10] M. Costa, Writing on dirty paper, IEEE Trans. on Information Theory, vol. 29, pp. 439-441, May 1983.

[11] S. S. Pradhan, J. Chou and K. Ramchandran, Duality between source coding and channel coding and its extension to the side information case. IEEE Transactions on Information Theory, vol. 49, vo. 5, May 2003. IEEE, USA.

[12] J. K. Su, J. J. Eggers and B. Girod, Illustration of the Duality Between Channel Coding and Rate Distortion with Side Information, Actes de la *34th Asilomar Conf. on Signals, Systems, and Computers*. Oct. 29-Nov. 1, 2000, Asilomar, CA, USA.

[13] T. M. Cover and M. Chiang, Duality between channel capacity and rate distortion with two-sided state information, IEEE Trans. of Inform. Theory, vol. 48, no. 6, pp. 1629 - 1638, June 2002.

[14] Chappelier V., C. Guillemot and S. Marinkovic, Turbo Trellis Coded Quantization, Actes de la *Intl. symp. on turbo codes*, September, 2003.
[15] Miller M. L., G. J. Dorr and I. J. Cox., Applying informed coding and informed embedding to design a robust, high capacity watermark," IEEE Trans. on Image Processing, 3(6): 792807, 2004.
[16] Eggers J., R. Buml, R. Tzschoppe and B. Girod, Scalar costa scheme for information embedding, IEEE Trans. Signal Processing,2002.
[17] R. G. Gallager, Low density parity check codes, Ph.D. dissertation, MIT, Cambridge, MA, 1963.
[18] MacKay, D. J. C. and R.M. Neal, Near Shannon limit performance of low density parity check codes, Electronics Letters, vol. 33, pp. 457-458, 1996.
[19] A. Aaron and B. Girod, Compression with side information using turbo codes, Proc. DCC'02, Snowbird, UT, April 2002.
[20] A. Liveris, Z. Xiong and C. Georghiades, Compression of binary sources with side information at the decoder using LDPC codes, IEEE Communications Letters, vol. 6, pp. 440-442, October 2002.

Auto-similarité de formes pour la discrimination des styles d'écriture des manuscrits médiévaux

Ikram Moalla [(1)(2)], Franck LeBourgois [(1)], Hubert Emptoz [(1)], Adel M. Alimi [(2)]

[(1)] Laboratoire d'InfoRmatique en Image et Systèmes d'information
[(2)] REsearch Group on Intelligent Machines
Email: ikram.moalla@ieee.org

Résumé

Ce article présente notre contribution à la discrimination des écritures des manuscrits médiévaux pour la Paléographie. Nous cherchons à retrouver les classifications construites par les paléographes sur la généalogie des écritures et de leurs évolutions durant le moyen âge. Ce travail devrait contribuer à confirmer objectivement les travaux des paléographes et tester les possibilités de l'analyse des images dans la discrimination des écritures médiévales. Nous avons choisi de caractériser statistiquement les formes des écritures sans segmenter l'image ni sa structure physique. Nous utiliserons principalement la notion de cooccurrence comme mesure d'auto-similarité. Les premiers résultats obtenus semblent confirmer les classifications données par les experts.

Mots clefs

Documents anciens, paléographie, reconnaissance de styles, autosimilarité, matrice de cooccurrence.

1 Introduction

L'Analyse d'Images de Documents est un domaine de recherche particulier qui se situe entre l'analyse des images, la reconnaissance des formes et les sciences humaines, et en particulier la science de l'histoire des textes. Cette discipline connaît actuellement une expansion avec l'avènement de la numérisation des fonds anciens du patrimoine notamment dans les bibliothèques et les archives nationales, départementales, municipales etc. Dans le domaine de la recherche sur les textes anciens, la philologie (science qui s'intéresse au problème de datation, de localisation et d'édition de texte), étudie autant la manière dont les textes sont écrits ou imprimés que leurs contenus. L'analyse du style personnel de l'écriture permet de différencier les différents scripteurs d'un manuscrit ou d'authentifier un document alors que le style général de l'écriture et de la mise en page permet des applications innovantes en paléographie, une des sciences sur laquelle repose la philologie.

L'analyse des styles d'écritures apporte des informations complémentaires aux contenus des textes que l'on considère comme étant des méta-données. La manière dont le texte est représenté constitue une information introduite de façon consciente ou inconsciente par l'auteur ou le scripteur qui peut permettre par exemple, de dater, d'authentifier ou d'indexer un document.

La présentation d'un document imprimé se manifeste par sa structure physique et la typographie des caractères (polices, taille, déclinaison, fonte) alors que la présentation d'un manuscrit ancien recèle d'autres niveaux d'interprétation comme le style personnel d'écriture du scripteur, la calligraphie utilisée et la mise en page du document. Ces derniers peuvent être représentatifs d'une époque et d'un lieu et servir à la datation et la transcription.

L'objet de ce travail consiste à apporter une première contribution méthodologique et applicative à l'analyse automatique des mises en forme des documents, au service de la recherche en histoire des textes. Nous nous intéresserons plus précisément aux manuscrits anciens latins du Moyen Age, période qui précède la Renaissance et l'avènement de l'imprimerie. La définition du style est multiple et complexe. Nous nous concentrerons sur une approche visuelle et perceptive du style des écritures, celle que l'on pourra qualifier et étudier avec des outils d'analyse d'images. La principale difficulté consiste à discerner le style d'une écriture manuscrite qui soit relié à la période historique et/ou une localisation géographique indépendamment du style personnel du scripteur.

2 Définitions et généralités

- La philologie classique : elle a pour objectif d'étudier les textes et les langues anciennes, leurs grammaires, l'histoire et la phonétique des mots pour l'enseignement et la compréhension des textes anciens. La philologie se base principalement sur le contenu des textes et non sur leur forme. Cette science concerne aussi bien les manuscrits que les imprimés.

- La paléographie : c'est l'étude de l'écriture et de la forme des caractères, de l'évolution des manières d'écrire. La science paléographique est une discipline complémentaire de la philologie pour les documents manuscrits car elle étudie les écritures manuscrites anciennes et leurs évolutions alors que la philologie classique étudie le contenu des textes, des langues et de leurs évolutions.
Les objectifs de la science paléographique sont principalement l'enseignement du déchiffrement correct des écritures anciennes et l'étude de l'histoire de l'évolution de l'écriture.

- **Les écritures latines et leurs évolutions :** depuis la fin du I^{er} siècle avant J.-C, les écritures se sont transformées selon les usages mais le fonctionnement est resté le même. Du VIII^{e} au XII^{e} siècle, la caroline règne sur l'Occident. Elle évolue vers des formes anguleuses

pour donner naissance à l'écriture gothique. Le passage d'une écriture à l'autre ne s'est pas toujours effectué de façon radicale mais par évolution lente et progressive. Ce qui explique qu'il est difficile d'identifier catégoriquement une écriture donnée. Par exemple on observe des textes en écriture *caroline* qui contiennent déjà des attributs de l'écriture *gothique*. Le paléographe doit alors quantifier précisément la part de mélange des familles d'écritures. C'est le cas par exemple de la classe d'écriture *prégothique* qui est intermédiaire entre l'écriture caroline et l'écriture gothique (Figure1) ou encore l'écriture *Hybrida* entre la *cursiva* et la *textualis* (Figure2).

Figure1 - *Evolution progressive de la caroline à la prégothique puis à la gothique.*

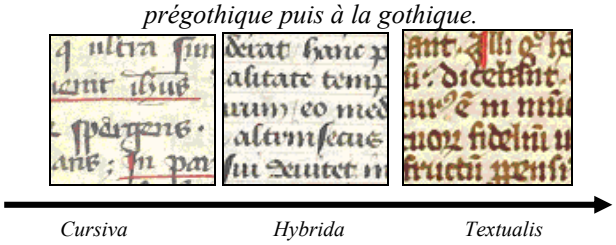


Figure2 - *Evolution progressive de la Gothique cursiva à* la *gothique hybrida* puis à la *gothique textualis*.

La diversification des familles d'écritures en Europe s'accélère jusqu'à la Renaissance et voit se développer des sous-familles d'écritures à l'intérieur de chaque grande famille de gothiques. Ainsi on peut distinguer plusieurs sous-familles de *gothique cursiva* représentées dans la Figure3 qui traduisent la précision de l'execution de cette écriture. De même, la Figure4 montre plusieurs sous-familles de *gothique textualis*.

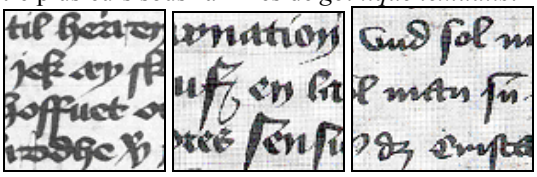


Figure3 - *Exemples de sous- familles de styles cursiva entre le VIII*ème *et le XVI*ème *siècle [1]*

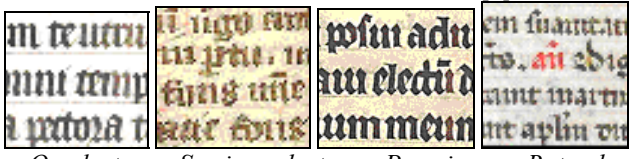


Figure4 - *Exemples de sous-familles de styles textualis entre le VIII*ème *et le XVI6*ème *siècle*

La Figure5 montre la variabilité des écritures à l'intérieur d'une même sous-famille comme la classe *gothique textualis rotunda*. Elle illustre la difficulté en terme d'analyse d'images à définir des descripteurs de formes pour trouver une homogénéité entre les différents échantillons d'une même écriture. La paléographie est, de ce fait, une science complexe.

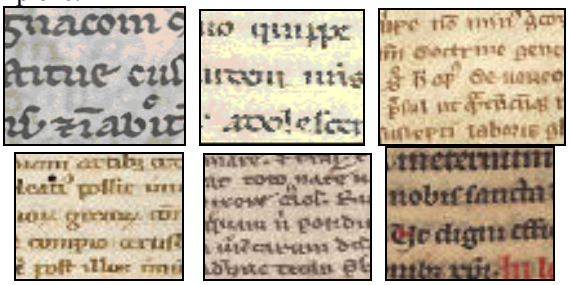


Figure5 - *Exemples d'images de textes représentant la variation intra-classe du style gothique textualis rotunda [BGB]*

De plus, il existe une diversité de classifications en familles et sous-familles d'écritures qui dépendent des paléographes. Un style d'écriture peut être étiqueté différemment par deux paléographes différents. La paléographie est une science subjective puisqu'elle n'a pas de règles précises pour trouver le style exacte d'une écriture. De plus, en examinant un style, le paléographe peut créer une confusion entre ses connaissances *a priori* ainsi que ses connaissances paléographiques sur les styles.

Trouver un système pour assister les paléographe est donc loin d'être un travail simple.

3 Etat de l'art

Les travaux sur la caractérisation des écritures ont été réalisés pour des applications différentes de celle de la paléographie comme la vérification et l'authentification de scripteurs, la pré-classification des écritures en terme de lisibilité pour une meilleure reconnaissance dans le tri automatique du courrier et des chèques. Toutes ces études sont connexes à notre problématique mais ces contributions ne sont pas toutes directement ré-exploitables pour l'étude paléographique. La rose des directions binaire a été utilisée par [2] pour identifier les formes différentes d'écritures en vue de leur reconnaissance. L'analyse fractale mesure les autosimilarités présentes dans une image, c'est une bonne mesure du style d'un scripteur qui peut servir à classer les écritures suivant leur lisibilité [3]. L'indice fractal est aussi susceptible de caractériser les différents alphabets dans les textes imprimés. [4] caractérise différents styles de texte par des mesures de complexité des formes, de lisibilité et de compacité indépendamment de l'alphabet utilisé. Enfin nous signalons d'autres travaux susceptibles d'être réutilisés pour la reconnaissance des écritures médiévales comme la reconnaissance des scripts (des mots dans un alphabet particulier) dans les documents multilingues. Ces travaux utilisent la similarité de

graphèmes [5], la texture [6], ou l'analyse de profil de projection [7] *etc*.

Du point de vue analytique, nous distinguons deux approches complémentaires pour le traitement des styles d'écritures:

- **Approche locale :** elle consiste à reproduire le travail des paléographes, en cherchant à établir des similarités visuelles entre des écritures à partir de lettres très particulières caractéristiques d'une écriture (exemples : 'r','s','e','a'). En effet certaines lettres spécifiques sont utilisées par les paléographes comme des marqueurs porteurs d'information nécessaire à la reconnaissance d'une écriture. Ces lettres doivent être prises au milieu des mots car leurs graphies changent suivant le scripteur quand elles sont situées en début ou à la fin des mots [8] [9].
- **Approche globale :** on ne cherche pas à reproduire le travail des paléographes, mais à utiliser une méthode plus appropriée à l'analyse automatique d'images. Elle consiste à analyser statistiquement l'image entière d'un manuscrit, sans segmenter l'image ni segmenter les lignes de texte, les mots ou les caractères et à trouver des descripteurs de formes capables de distinguer les différentes écritures.

Le SPI pour *System for Paleographic Inspections* [8], constitue la seule tentative pour la réalisation d'un système d'assistance automatique en paléographie [9] en utilisant une approche locale. Elle consiste à isoler manuellement les caractères représentatifs d'une écriture et à les comparer à des caractères de référence contenus dans une base paléographique étiquetée manuellement. La comparaison utilise la distance tangente et la règle des *k-plus proches voisins*. Les conditions expérimentales ne permettent pas d'évaluer le système avec objectivité. De plus, aucun détail n'a été donné concernant les styles utilisés. Et le manque de résultats chiffrés nous empêche de juger la qualité de ce travail.

4 Méthode proposée

Notre objectif est le développement de méthodologies et d'outils d'analyse d'images pour assister les historiens dans la classification et la datation des manuscrits anciens latins à partir de la reconnaissance des écritures. En effet, chaque époque de l'histoire a été marquée par un ou plusieurs types d'écritures. Ainsi, la connaissance de l'écriture d'un document permet de connaître sa date et/ou son origine géographique. Notre domaine d'étude couvre les écritures anciennes latines du VIII^{ème} siècle jusqu'à la Renaissance au XVI^{ème}. L'étude des écritures latines antérieures au VIII^{ème} siècle comme l'*onciale* ou l'écriture *cursiva* n'a pas un réel intérêt pour les paléographes. En revanche l'assistance à l'expertise des écritures médiévales est très utile à partir du XII^{ème} siècle. Il s'agit de différencier les grandes familles d'écritures comme le montre la répartition de la Figure6.

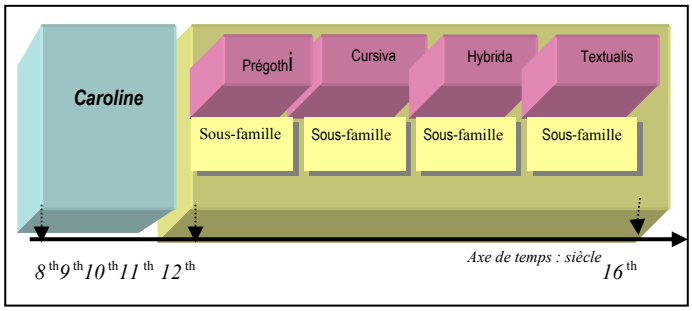


Figure 6 - *Répartition des différentes familles et sous-familles de styles latin entre le 8ème et le 16ème siècle*

Dans un premier temps, notre travail s'est focalisé sur l'extraction de caractéristiques suffisamment discriminantes pour pouvoir différencier le plus grand nombre d'écritures latines possibles. Cette étude a permis d'étudier la faisabilité d'un système d'analyse automatique des images à l'usage des paléographes.
Dans un deuxième temps nous avons affiné l'étude en tentant progressivement de discriminer les deux grandes familles de styles d'écritures latines : *caroline*, *gothique* puis les sous familles de *gothiques* : *cursiva*, *hybrida* et *textualis* et enfin les sous-familles telles que la *rotunda*, la *quadrata,* la *semi-quadrata* et la *prescissa* pour la *textualis*; la *formata*, la *libraria* et la *currens* pour la *gothique cursiva*. Cette analyse a pour objectif à la fois d'augmenter la précision de la discrimination entre les écritures et d'étudier plus en détail les confusions possibles entre les écritures proches.

4 .1 Des conditions difficiles

Le développement d'un système d'assistance à l'expertise des manuscrits anciens est une tâche rendue difficile par de nombreux facteurs la complexité des formes d'écritures (Figure2, 3, 4), la grande variabilité des écritures d'une même classe (Figure5), l'existence d'écritures hybrides issues de mélanges de plusieurs écritures (Figure1, 2), la faible qualité de conservation des manuscrits, le vieillissement des supports et des encres (Figure7), l'enchevêtrement des lignes et des mots (Figure8), la présence de notes dans la marge et/ou entre les lignes (Figure9) et la grande variabilité de la qualité des images de différentes origines : certaines images en couleurs proviennent d'une numérisation de qualité, d'autres images de mauvaise qualité sont issues de la numérisation de livres ou de microfilms, en niveaux de gris. La plupart des images présentent des dégradations dues à une trop forte compression (JPEG). Enfin nos échantillons ont été numérisés avec des résolutions toutes différentes (Figure10).

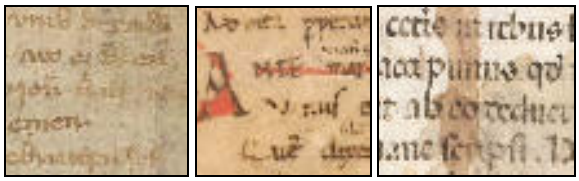


Figure7 - *Vieillissement de l'encre [BGB]*

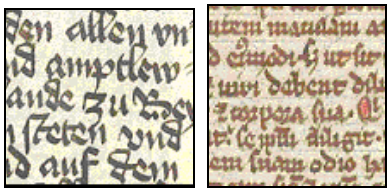


Figure8 - *Enchevêtrement des lignes [BGB]*

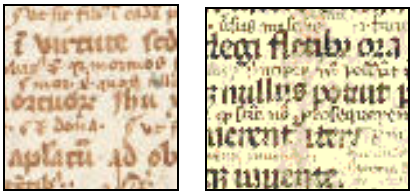


Figure9 - *Ecriture à la marge et/ou entre les lignes [BGB]*

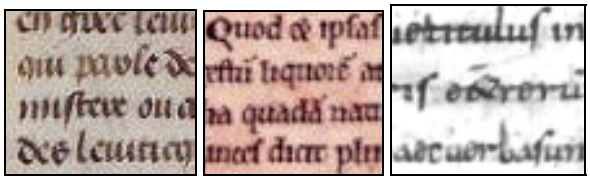


Figure10 - *Dégradations dues à une mauvaise qualité de prise d'image*

Dans ce contexte difficile, nous allons analyser directement l'image en niveaux de gris sans filtrage préalable, sans restauration et sans correction géométrique. Ce choix nous prive d'une grande partie des travaux réutilisables et en particulier tous ceux basés sur la segmentation.

4 .2 Notre approche

Nous avons donc choisi *l'approche globale* du fait des conditions difficiles décrites précédemment. Nous avons cherché des descripteurs de formes capables de distinguer les différentes écritures. Ces mesures globales devraient être indépendantes du *contenu du texte*, du *style personnel du scripteur*, de la *langue utilisée*, des *lettres employées* et de leurs fréquences. L'analyse globale s'affranchit des formes fantaisistes des caractères de début et de fin de mots, ainsi que de l'éventuelle présence d'ornements incrustés dans le texte. Ces avantages sont très précieux pour pouvoir analyser une grande quantité d'images de manuscrits de qualité très variable ainsi que les manuscrits dégradés.
Sans travaux antérieurs dans le domaine de l'analyse globale des écritures manuscrites médiévales, nous avons expertisé des descripteurs qui vérifient un certain nombre de conditions. Pour vérifier que l'on caractérise l'écriture et non le contenu du texte lui-même ni le style personnel du scripteur, l'inclinaison du texte, sa mise en page ou sa taille, les descripteurs de formes doivent être robustes, donc doivent pouvoir se calculer sans segmentation préalable des images et doivent résister au *changement de luminosité* et de contraste et au changement de résolution. Les descripteurs doivent être également *invariants au scripteur*, aux *contenus des te*xtes, à *la taille de l'échantillon* de texte, au *changement d'échelle*, au *changement de ratio* et à la *rotation*.

4 .3 Application de la cooccurrence sur les écritures médiévales

La cooccurrence a été largement utilisée comme moyen de caractériser une texture en analyse d'images. Les images de documents présentent aussi des textures par la répétition des motifs réguliers des caractères, des mots et des lignes de texte. Cependant nous ne voulons pas mesurer la mise en page ni décrire la gestion des espaces (densité des traits, interlignes…). Nous cherchons plutôt à caractériser les écritures. Nous allons utiliser la cooccurrence de façon à ne mesurer que les variations des formes elles-mêmes et non les variations des formes entre elles. Pour cela nous devons effectuer de très faibles déplacements et nous assurer que l'on ne compare pas verticalement une ligne de texte avec les lignes adjacentes ou recouvrir horizontalement une lettre avec les lettres voisines. Par conséquent nous avons calculé les coccurrences sur des images qui ont été normalisées manuellement pour qu'elles présentent toutes un corps de texte de 30 pixels de hauteur et nous avons limité les déplacements à moins de la moitié de la taille du corps des lignes de texte. La normalisation de l'échelle des images par rapport au corps des textes est aussi nécessaire pour ne pas influencer la comparaison des observations sur des tailles de texte trop différentes (contrairement aux paléographes qui travaillent sur des images à l'échelle 1:1).
La cooccurrence se généralise aux images en niveaux de gris, et donne des matrices de taille Ng×Ng avec Ng le nombre de niveaux de gris de l'image pour chaque coordonnées trigonométrique (ρ,θ).

$$Co\phi dx=\rho\cos\theta, dy=\rho\sin\theta)=\frac{1}{N}\left[\sum_{x,y} I(x,y)=i \cap I(x+dx,y+dy)=j\right]_{i,j}=\frac{1}{N}\left[M_{i,j}\right]_{(i,j)=0..Ng-1}$$

(Avec I(x,y): image d'origine, I(x+dx, y+dy) : image translatée et $M_{i,j}$: matrice de cooccurrence)

Nous avons choisi d'utiliser initialement un maximum d'information et de prendre un pas très fin pour les valeurs de ρ et de θ. Nous avons utilisé 16 directions (θ∈[0..15]) et 15 déplacements possibles (ρ∈[1..15]) soit 16x15 matrices au maximum. Les valeurs des pixels ont été ramenées de 256 à 16 valeurs différentes. Une subdivision plus fine des valeurs de gris n'apporte pas d'information complémentaire pour des images de manuscrits qui sont constituées essentiellement de traits. Chaque écriture décrit une signature différente suivant les valeurs de ρ et θ (Figure11).

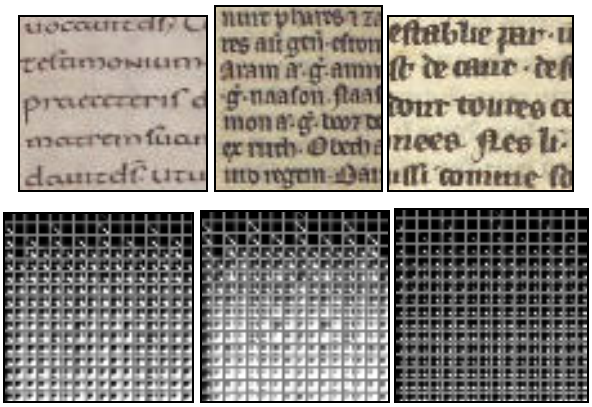


Figure 11 - *Matrices de cooccurrences relatives à quelques exemples de planches de styles différents*

4 .4 Analyse des descripteurs

Nous analysons les données des ***n*** observations décrites par ***p*** variables avec ***p*** égal au nombre de matrices de cooccurrence non nulles suivant ρ et θ, multiplié par un nombre de descripteurs issus des travaux de Haralick [10] que nous présentons ci-dessous. Soit :

$$P(i,j) = \frac{M(i,j)}{\sum_{i=0}^{i<Ng}\sum_{j=0}^{j<Ng} M(i,j)}$$

$f1$	*Second Moment Angulaire ou énergie*	$f_1=\sum_{i=0}^{i<Ng}\sum_{j=0}^{j<Ng} P(i,j)^2$	Cette caractéristiques mesure l'homogénéité, et détecte le degré de dispersion d'une texture.
$f2$	Moment de la Différence de l'élément	$f_2=\sum_{k=0}^{k<Ng} k^2\times P_{x-y}(k)$	Cet indice mesure le degré de contraste ou de variation locale présent dans une image.
$f3$	Correlation	$f_3=\sum_{i=1}^{Ng}\sum_{j=1}^{Ng}\frac{(i-\mu_x)\times(j-\mu_y)P(i,j)}{\sigma_x\times\sigma_y}$	Cet indice décrit les corrélations entre les lignes et les colonnes de la matrice de cooccurrence et mesure de ce fait les dépendances linéaires entre les niveaux de gris dans une image.
$f4$	Variance	$f_4=\sum_{i=1}^{Ng}\sum_{j=1}^{Ng}(i-j)^2P(i,j)$	Une forte valeur caractérise une texture fine.
$f5$	Moment de la Différence Inverse	$f_5=\sum_{i=1}^{Ng}\sum_{j=1}^{Ng}\frac{P(i,j)}{1+\lvert i-j\rvert^2}$	Une forte valeur indique que les éléments texturaux sont de grande taille.
$f6$	Moyenne des sommes	$f_6=\sum_{k=0}^{k<2Ng-1} k\times P_{x+y}(k)$	*Moyenne des projections* P_{x+y}
$f7$	Variance des sommes	$f_7=\sum_{k=0}^{k<2Ng-1}(k-f_6)^2\times P_{x+y}(k)$	*Variance des projections* P_{x+y}
$f8$	Entropie des sommes	$f_8=-\sum_{k=0}^{k<2Ng-1} P_{x+y}(k)\times\log(P_{x+y}(k))$	*Entropie de* P_{x+y}
$f9$	Entropie	$f_9=-\sum_{i=0}^{i<Ng}\sum_{j=0}^{j<Ng} P(i,j)\times\log(P(i,j))$	C'est un indicateur de désordre.
$f10$	Variance des Différences	$f_{10}=\sum_{k=0}^{k<Ng}(k-m_{x-y})^2\times P_{x-y}(k)$	*Variance de* P_{x-y} *avec* $m_{x-y}=\sum_{k=0}^{k<Ng} k\times P_{x-y}(k)$
$f11$	*Entropie des Différences*	$f_{11}=\sum_{k=0}^{k<Ng} P_{x-y}(k)\times Log(P_{x-y}(k))$	*Entropie de* P_{x-y}
$f12$	*Entropy Measure*	$f_{12}=\frac{HXY-HXY1}{\max\{HX,HY\}}$ $HXY=f_9$	$HX=-\sum_{i=0}^{i<Ng} Px(i)\times Log(Px(i))$ $HY=-\sum_{j=0}^{j<Ng} Py(j)\times Log(Py(j))$ $HXY1=-\sum_{i=0}^{i<Ng}\sum_{j=0}^{j<Ng} P(i,j)\times Log(Px(i)\times Py(i))$

Avec $Px(i)=\sum_{j=0}^{j<Ng} P(i,j)$, $Py(j)=\sum_{i=0}^{i<Ng} P(i,j)$, $P_{x+y}(k)=\sum_{i<Ng}\sum_{j<Ng}^{i+j=k} P(i,j)$, $P_{x-y}(k)=\sum_{i<Ng}\sum_{j<Ng}^{\lvert i-j\rvert=k} P(i,j)$ sont respectivement les projections selon les axes horizontal, vertical et oblique de la matrice normalisée *P*.

Nous obtenons donc ***n*** points dans $\boldsymbol{IR^p}$ avec ***p***=216×12, ***n*** étant le nombre d'images d'écritures observées. L'espace des caractéristiques est bien trop grand par rapport au nombre d'observations ***n*** pour un classifieur. Nous pensons qu'il existe parmi les ***p=2592*** variables, un nombre limité de facteurs qui peuvent faire apparaître les classes d'écritures. Un travail manuel de sélection des caractéristiques serait trop long et fastidieux. Il est donc nécessaire de réduire le nombre de descripteurs par une analyse statistique de la variance. Cette analyse nous a permis de trouver les descripteurs corrélés et donner un nombre réduit de facteurs qui sont des combinaisons linéaires des ***p*** variables d'origine. L'analyse des données est encore l'occasion de mener une analyse canonique de la proximité des classes puis comparer les résultats avec ceux des experts.

5 Analyse des résultats

L'analyse discriminante (AD) a permis de trouver les vecteurs permettant de projeter toutes les observations définies dans un espace à ***p*** dimensions, sur un plan qui discrimine une grande majorité de classes (Figure12).

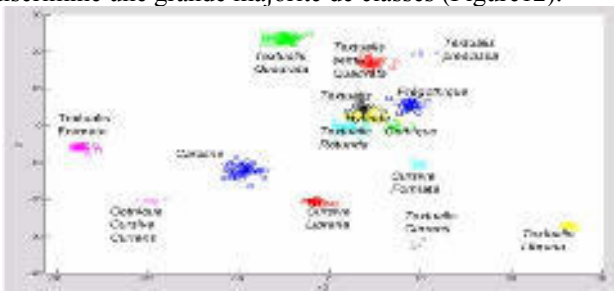


Figure 12 - *Résultats de l'AD pour les 15 classes*

Le fait d'obtenir une majorité de classes séparées signifie qu'il existe des combinaisons linéaires de descripteurs qui peuvent résoudre notre problème de discrimination des écritures médiévales. Nous avons obtenu une bonne dispersion des classes : *1.Caroline*, *3.Cursiva Libraria*, *4.Cursiva Formata*, *5.Cursiva Currens*, *8. Textualis prescissa*, *9. Textualis Quadrata*, *10. Textualis Semi-Quadrata*, *12. Textualis Formata*, *13. Textualis Liraria* et *14. Textualis Currens*. La matrice de confusion a donné des taux de discrimination assez satisfaisants (de 48% pour la classe *12. Textualis Formata* à 100% pour la classe *5. Cursiva Formata*) pour les types d'écritures relatifs à ces classes. Les exceptions concernent les classes *2. Gothique* et *7. Textualis* non considérées comme de vraies familles ainsi que la *8. Textualis Prescissa* et la *14. Textualis Currens* qui ne sont pas significatives vu leurs nombres très réduits de représentants.

Quant aux *2. Gothique*, *6. Hybrida*, *7. Textualis*, *11. Textualis Rotunda* et *15. Prégothique*, elles sont les moins bien séparées par l'AD et cela a entraîné des taux de

confusions assez importants entre ces classes. Nos résultats montrent qu'il existe de véritables classes paléographiques au sens de la reconnaissance des formes. Certaines classes sont nettement séparées et forment de véritables familles bien identifiées. Ce sont les familles des écritures les plus détaillées (*8. Textualis Prescissa, 9. Textualis Quadrata, 10. Textualis Semi-Quadrata, 12. Textualis Formata, 13. Textualis Libraria, 14. Textualis Currens*) et les familles *1. Caroline, 3. Gothique cursiva* ; elles sont nettement séparées des autres classes. Les quatre classes confuses qui sont la *2.Gothique*, la *7. Textualis*, la *15. Prégothique* et la *6. Hybrida* ne constituent pas de véritables classes d'écritures homogènes au sens de l'analyse d'images. Nous pensons (sous réserve que ces résultats soient validés par les experts) que les classes *2. Gothique* et *7. Textualis* contiennent des écritures non suffisamment renseignées par les paléographes et qu'il est donc normal que ces classes génériques soient confuses avec les sous-familles respectives. Enfin nous supposons que les écritures prégothiques sont des écritures transitoires entre les écritures *carolines* et *gothiques*. En omettant les classes confuses les plus problématiques qui sont la *2. Gothique*, la *7. Textualis*, la *15. Prégothique* et la *6. Hybrida,* nous obtenons 11 classes correctement séparées.

Tableau - Matrice de confusion obtenue par analyse discriminante sur 11 classes en utilisant les 12 caractéristiques de Haralick (f1à f12)

	1	2	3	4	5	6	7	8	9	10	11	Pourcent
1.Caroline	58	1	<1	1	<1	<1	1>	1	<1	1>	1>	91%
2.Caroline Mineure	1>	15	1	1>	<1	<1	1>	1>	<1	1	1>	83%
3.Caroline Romaine	1>	1	1	1>	<1	1	1>	1>	<1	1>	1>	71%
4.Caroline Claire	1>	1	<1	5	<1	<1	1>	1>	<1	1>	1>	83%
5.Gothique Fracture	1>	<1	<1	1>	7	<1	1>	2	<1	1>	1>	88%
6.Gothique Quadrata	1>	<1	<1	1>	<1	19	1	1	2	1>	1>	83%
7.Gothique Semi Quadrata	1>	<1	<1	1>	<1	1	75	9	2	1>	1>	83%
8.Textualis Prescissa	1	1	0	0	0	5	3	64	1	0	0	88%
9.Textualis Quadrata	0	1	0	1	0	2	0	1	10	2	0	76%
10.Textualis Semi-Formata	0	2	1	1	0	0	0	0	1	14	1	76%
11.Textualis Libraria	0	0	0	0	0	0	0	0	0	2	2	50%
Total	60	22	7	11	2	61	80	85	22	21	5	83%

Le taux moyen de discrimination est passé de 59% à 81%. Il pourrait s'améliorer si nous disposions d'effectifs plus équilibrés et d'une meilleure représentation des classes *8. Textualis Prescissa* et *14. Textualis Currens*. Les 2 classes de faibles effectifs à savoir *8. Textualis Prescissa* et *14. Textualis Currens* ne peuvent pas être analysées du point de vue statistique, ce qui explique leurs faibles taux de discrimination. En conclusion, il existe bel et bien des classes d'écritures qui paraissent compatibles avec l'expertise paléographique (selon la Bibliothèque de Grande Bretagne [BGB]) et la cooccurrence constitue une bonne mesure pour différencier les différentes écritures.

6 Conclusion et perspectives

Après avoir établi le cadre de notre travail par rapport à la science de la paléographie, nous avons posé le problème de la classification des types d'écritures. Nous avons défini une approche globale comparant directement le type d'écriture à partir de zones de texte quelconques dans des documents. Nous avons choisi de travailler avec les indices statistiques de Haralick pour décrire nos matrices de cooccurrence afin d'avoir un nombre plus réduit de descripteurs par image.

Après une décorrélation des données par une analyse factorielle, nous avons constaté que nos descripteurs d'images basés sur des mesures statistiques de cooccurrence permettent de retrouver approximativement les classes d'écritures définies par la Bibliothèque de Grande Bretagne [BGB]. Nous reprenons actuellement les tests avec une autre classification des types d'écritures présentée dans [1] pour la comparer avec la classification anglo-saxonne. Les résultats de l'analyse discriminante ont été concluants et nous ont permis d'avoir dans plusieurs cas des séparations correctes des classes d'écritures. Nous avons obtenu un taux de 81% de discrimination global lorsque nous avons éliminé les quatre classes posant des problèmes de sous-représentation statistique ou bien d'absence de précision. Le passage d'une famille à une autre n'étant jamais brusque et certaines écritures peuvent présenter un mélange de caractéristiques des écritures qui ont contribué à leurs formations, nous devons remplacer pour ces écritures l'analyse discriminante par une analyse qui mesure le taux de mélange avec les autres classes bien définies.

Références

[1] Derolez A., "The Palaeography of Gothic Manuscript Books", from the Twelfth to the Early Sixteenth Century", Cambridge Studies in Palaeography and Codicology, Cambridge University Press, 2003.

[2] Crettez J. P., "A set of handwriting families : style recognition", International conference on Document Analysis and Recognition, Vol 1, pp 489, 1995.

[3] Vincent N., Boulétreau V., Sabourin R., Emptoz H., "How to use fractal dimensions to qualify writings and writers, Fractals", World Scientific, Vol 8, n°1, pp.85-97, 2000.

[4] Eglin V., "Contributions à la structuration fonctionnelle des documents imprimés. Exploitation de la dynamique du regard dans le repérage de l'information", Thèse de Doctorat, INSA de Lyon, 1998.

[5] Moalla I., Alimi A.M. Ben Hamadou A., "Extraction of Arabic text from multilingual documents", IEEE International Conference on Systems, Man and Cybernetics, Tunisie, 2002.

[6] Tan T. N., "Rotation Invariant Texture Features and Their Use in Automatic Script Identification", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 7, pp. 751-756, 1998.

[7] Wood S. L, Yao X., Krishnamurthi K., Dang L., "Language Identification for Printed Text Independent of Segmentation", Proc. IEEE ICIP, pp. 428-431, 1995.

[8] Aiolli., Simi M., Sona D., Sperduti A., Starita A., Zaccagnini G. SPI: a System for Palaeographic Inspections. AIIA Notizie, vol. 4, pp 34-38, 1999.

[9] Ciula A., "Digital palaeography: using the digital representation of medieval script to support palaeographic analysis", Digital Medievalist 1.1, 2005.

[10] Haralick R. M., Shanmugam K. and Its'Hak Dinstein. "Textural Features for Image Classification", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 3, pp. 610-621, 1973.

[BV] http://www.villevalenciennes.fr/bib/fondsvirtuels/microfilms/accueil.asp#item

[BGB] http://prodigi.bl.uk/illcat/searchMSNo.asp.

Contribution à la Reconnaissance Automatique des Documents d'Entreprises

Djamel GACEB *Frank LEBOURGEOIS* *Véronique EGLIN* Hubert EMPTOZ
LIRIS UMR 5205CNRS, INSA de Lyon 69621 Villeurbanne Cedex

djamel.gaceb1@insa-lyon.fr flebourg@rfv.insa-lyon.fr veronique.eglin@insa-lyon.fr hubert.emptoz@liris.cnrs.fr

Résumé

Le traitement automatique de documents et courrier d'entreprises est un domaine exigeant en terme de performances et de vitesse. Les systèmes actuels utilisent des architectures modulaires dans lesquelles chaque étape du processus de reconnaissance est indépendante. Pour augmenter les performances, il est nécessaire de réintroduire une coopération entre les différents modules. Dans ce cadre, nous proposons une approche hybride de localisation des zones de textes et de binarisation des images. Ce couplage a permis à la fois de gagner en temps de calcul en évitant de traiter l'arrière plan de l'image et d'obtenir une meilleure segmentation en caractères pour l'OCR. Nous présenterons les résultats obtenus à partir de l'implémentation de notre nouvelle approche sur une ligne industrielle qui traite quotidiennement plusieurs tonnes de courrier et documents internes de grandes entreprises.

Mots clefs

Localisation de textes, segmentation des images, courrier d'entreprises.

1 Introduction

Le domaine du traitement automatique du courrier d'entreprises possède en générale plusieurs contraintes :

- Très grande variété de documents (texte manuscrit ou imprimé, qualité, couleur et texture de papier différentes)
- Contraintes de temps réel (temps de traitement limité)
- Adaptation au mode de capture par système de caméra linéaire (on devra développer les outils d'analyse d'image à la particularité de cette prise d'image pour optimiser les temps de calcul)
- Une obligation de résultats (Le système doit être le plus performant possible pour éviter les coûteuses interventions manuelles).

On retrouve aussi d'autres contraintes particulièrement liées à l'application industrielle qui nous concerne:

- Les images à traiter sont réparties en catégories correspondantes aux familles de courriers des clients d'entreprise : Courrier interne manuscrit (CIM), Courrier interne dactylographique (CID), formulaire (FRM), planus (PL), carte bleue (CB), listing A3(LA3), listing A4(LA4), NPAI, chèque circulant (CHC). Ces images sont très différentes du point de vue de leur taille, de leur orientation, des couleurs du fond et du texte, de la position de texte dans l'image, de la taille des caractères et des types d'écritures (imprimés, imprimés matriciels, manuscrits…). Les documents sont traités par lots ou bien arrivent en vrac.
- Temps de traitement limité, pour l'acquisition de l'image, sa binarisation, la localisation des zones de textes.
- La résolution actuelle de la caméra CCD utilisée est d'environ 200dpi (10 pouces/2048 pixels) et ne peut prendre qu'une seule image par document.
- Les documents non reconnus sont immédiatement traités manuellement. L'échec de reconnaissance s'explique généralement par un dysfonctionnement des étages de prétraitements et en particulier des étages de segmentation et de localisation [1][2].

2 Comparaison des méthodes existantes

2.1 Architectures logicielles linéaires et approches coopératives

Les limites atteintes par les systèmes de vision actuels sont dues à l'organisation linéaire du traitement de l'information. Le taux de rejet et le taux d'erreur des systèmes industriels sont élevés à cause de l'indépendance des processus engagés dans la reconnaissance.

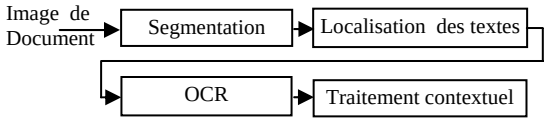


Figure 1 – *Architecture linéaire de systèmes de vision*

Cette séparation des processus est adaptée à la répartition des tâches sur plusieurs ordinateurs connectés, mais l'échec d'une seule étape du processus conduit irrémédiablement le système à rejeter ou bien à commettre une erreur d'interprétation. Certains travaux font déjà référence à des architectures plus avancées. [3] propose un système multi-agents pour l'échange des données et la collaboration entre les différents modules d'acquisition et de reconnaissance. [4] décrit une architecture

collaborative des différents modules pour reconnaître les adresses et les codes postaux. [5] décrit une approche probabiliste pour combiner la localisation, la segmentation et la reconnaissance. Enfin, [6] décrit une segmentation en mots dirigée par une étape de reconnaissance.
C'est de dans ce contexte que s'inscrivent nos travaux pour réduire les taux de rejet et les erreurs du système de vision existant en introduisant une meilleure coopération entre les différentes étapes de la reconnaissance tout en restant dans les limites d'un processus de temps réel. Nous allons donc étudier une architecture non linéaire du processus de reconnaissance en introduisant des bouclages d'informations possibles entre les différents étages (classification des documents, localisation des zones d'intérêts, localisation des zones de texte, segmentation, OCR, reconnaissance de la structure du document et classification du type de document…). Parmi les couplages possibles, nous proposons dans cet article de commencer par une coopération entre la segmentation et la localisation des zones textuelles. Cette coopération devrait nous permettre à la fois d'économiser le temps de traitements et d'améliorer la qualité de la segmentation.

2.2 Comparaison des méthodes de binarisation des documents

La numérisation des documents et courriers avec une caméra CCD, donne des images en niveaux de gris. La réduction de la quantité d'informations à analyser pour l'OCR, nécessite souvent une étape préliminaire de binarisation. La binarisation est le passage irréversible d'une image en niveaux de gris qui permet une classification entre le fond (image du support papier) en blanc et la forme (traits, graphique, caractères) en noir. Le mauvais choix du seuil, peut détruire une grande part d'information contenue dans l'image de l'enveloppe. En effet, une bonne binarisation doit être capable de conserver à la fois tous les caractères et les gravures sans récupérer trop de bruit.

On peut trouver dans la littérature de très nombreux travaux concernant la binarisation de documents. Les plus simples et les plus rapides utilisent l'histogramme de l'image comme les célèbres méthode d'Otsu [7], de Fisher [8] ou d'entropie [9] pour déterminer un seuil qui leur sera appliqué. Ces méthodes globales ont l'avantage d'être extrêmement rapide mais la variation d'éclairage sur le document fait chuter la qualité de la binarisation (Figure2).

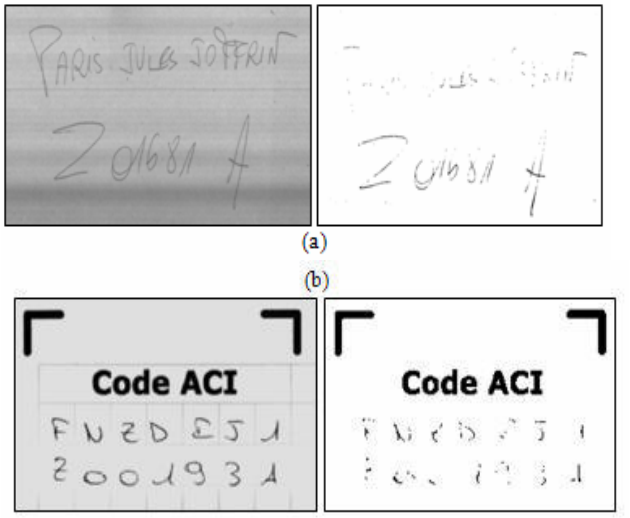


Figure 2 - *Binarisation, (a) par la méthode de Fisher, (b) par la méthode d'Otsu.*

D'autres, telles que les méthodes introduites par Niblack [10], Sauvola[11], Wolf [12] utilisent une approche locale aux pixels pour déterminer une valeur de seuil, pour chaque pixel de l'image, en analysant son voisinage. Leur adaptation locale aux changements de contraste explique l'efficacité de ces méthodes sur les images de manuscrits ou encore sur les documents qui utilisent des couleurs d'encre différentes. Cette approche permet d'obtenir un résultat faiblement dépendant des variations de luminosité sur la page (Figure 3).

Figure 3 - *Binarisation adaptative par la méthode de Sauvola, (a) Image CHC, fenêtre 7×7, (b) Image CIM, fenêtre 9×9, (c) Image 'Insert' fenêtre 15×15.*

Malgré leur efficacité, les méthodes locales possèdent les inconvénients suivants :

- temps de calculs prohibitifs en fonction la taille de la fenêtre d'analyse

- sur-segmentation des défauts et de la texture de papier sur l'arrière plan de l'image
- Traitement difficile des documents dont la taille des caractères est très variable, la fenêtre d'analyse étant fixe durant tout le traitement.

Le tableau suivant contient les temps moyens de traitement, calculés sur une base de 9341 images.

Méthodes de binarisation Type de documents	Niblack	Sauvola	Wolf	Fisher
CHC	4,44	4,47	4,38	0,32
NPAI	3,75	2,28	2,30	0,29
CB	4,34	3,46	4,30	1,74
LA3	4,38	4,46	4,32	0,31
LA4	2,42	2,43	2,42	0,23
FRM	3,59	3,50	4,33	0,25
PL	4,53	4,53	4,45	1,69
CIM	4,50	4,61	4,39	0,36
CID	1,30	1,28	1,29	0,15

Tableau 1 - Vitesses d'exécution des méthodes classiques de binarisation (en secondes).

Après cette analyse, on peut conclure qu'aucune des méthodes classiques ne remplit toutes les conditions imposées (efficacité et rapidité).

2.3 La localisation des zones de texte

On regroupe les travaux sur la localisation des blocs adresses en plusieurs classes :
- Les méthodes basées sur la multi-résolution
- Les méthodes agrégatives par filtrage
- Les méthodes ascendantes basées sur l'adjacence des composantes connexes
- Les méthodes basées sur la segmentation d'images
- Les méthodes basées sur l'apprentissage

Les contraintes de temps réel et la grande variabilité des tailles des caractères et d'espaces entre les mots ont amené plusieurs chercheurs à utiliser la multi-résolution. La localisation du bloc adresse par multi-résolution ne nécessite pas une binarisation préalable de l'image et s'appuie sur une construction pyramidale permettant de faire apparaître à un niveau de résolution approprié la structure d'un bloc de lignes [13]. L'approche pyramidale permet aussi une analyse de type descendante pour construire un arbre d'inclusion de composantes connexes segmentées aux différents niveaux de résolution [14]. D'autres travaux utilisent les méthodes agrégatives classiques de type RLSA [15] qui sont rapides car elles ne nécessitent pas un calcul coûteux de capture des connexités. Cependant ces méthodes sont sensibles à l'inclinaison des documents et nécessitent une bonne orientation et un parfait alignement des lignes de texte. Ces approches agrégatives, ne sont pas nouvelles au regard des premiers travaux sur la localisation des textes dans les images par G. Nagy pour extraire la structure physique du courrier et des formulaires au service des grandes entreprises [Nagy 68]. Les ordinateurs de l'époque n'ayant pas la puissance de calcul nécessaire pour des algorithmes évolués, il eut l'idée d'utiliser la défocalisation progressive de l'optique de la caméra pour rendre l'image progressivement floue dans laquelle les caractères deviennent des «taches» qui s'agglomèrent progressivement entre elles pour désigner les mots, les lignes et les blocs de texte.

Les travaux sur la localisation des blocs adresses basés sur le regroupement des composantes connexes sont nombreux et ne sont pas adaptés aux contraintes de temps réel. En effet la capture de toutes les connexités et la binarisation « aveugle » de toute l'image est trop coûteuse en temps de calculs [18]. De plus ces méthodes nécessitent une classification complexe des connexités en fonction de leur alignement et un rejet des connexités qui ne correspondent pas à des éléments textuels [16][17]. Enfin ces méthodes nécessitent une binarisation préalable de l'image.

Les méthodes basées sur la segmentation de l'image avec des méthodes classiques comme le Split&merge [19] permettent de localiser rapidement les régions non uniformes de l'image susceptibles de contenir du texte. D'autres méthodes de segmentation utilisent aussi les informations sur la texture avec des filtres de Gabor [21][20] ou les ondelettes [23]. Ces méthodes localisent à la fois les zones pertinentes de l'image sans capturer les connexités, mais elles différencient les zones de texte des éléments non textuels à partir de leurs textures. Cependant ces approches intéressantes sont néanmoins très coûteuses en temps de calculs.

Les systèmes de localisation par apprentissage [22] [24][25] nous paraissent difficiles à mettre en œuvre devant la grande variété des documents que nous avons a traiter. De plus certains chercheurs admettent que les systèmes à apprentissage sont moins performants que les systèmes dont les règles ont été ajustés manuellement au problème posé [26].

3 Notre proposition

3.1 Couplage binarisation/localisation

La séparation entre l'étape de la binarisation et celle de localisation des textes, augmente à la fois le temps de calcul et conduit à une sur-segmentations du bruit et de la texture de papier sur des zones vides de l'image. Nous avons pu optimiser notre méthode de binarisation en appliquant les calculs de seuils adaptés uniquement à

proximité des zones de texte. Pour cela nous détectons très rapidement des zones de textes afin d'y appliquer une méthode de segmentation locale de type Sauvola. Nous évitons ainsi de binariser les zones vides qui représentent la plus grande partie de l'image. Cette approche nous permettra aussi de corriger le défaut de sur-segmentation des méthodes adaptatives sur les zones non textuelles de l'image.

3.2 Application à la localisation des zones textuelles

La localisation doit s'effectuer directement sur l'image en niveaux de gris issue de la caméra. La méthode développée doit aussi réduire le plus possible le nombre de fausses détections et ajuster les zones au voisinage du texte. Nous avons utilisé une méthode robuste qui permet de localiser rapidement toutes les zones textuelles dans une scène naturelle sans éclairage particulier ni contrainte lors de la prise d'image. Ce procédé consiste à agglomérer certaines périodicités caractéristiques des lignes de texte qui proviennent des variations lumineuses sur des contours des caractères ou générées par les alternances entre les traits ou entre les caractères. Ces périodicités sont calculées à partir de séquences de pixels à gradients élevés. Pour éviter de filtrer ces points et d'introduire de nouveaux seuils, on effectue localement, dans un voisinage V en chaque point (x_0,y_0), une simple sommation des normes des gradients normalisée par le nombre N de pixels du voisinage $V(x_0,y_0)$*(1).*

$$G(x_0, y_0) = \frac{1}{N} \sum_{(x,y) \in V(x_0, y_0)} \frac{\partial f(x,y)}{\partial \vec{v}} \qquad (1)$$

Ce filtre de « gradients cumulés », initialement développé pour la localisation de textes dans les images vidéos [27], a été utilisé pour la localisation des titres dans les vidéos non contraintes comme les archives télévisuelles [12] et la segmentation de l'imprimé composite couleur [28]. Cette méthode des « gradients cumulés » possède plusieurs inconvénients pour notre application. Nous proposons de l'adapter à notre environnement :

Le filtre suppose que la direction de l'enveloppe est a priori connue. En effet, les dérivées sont calculées dans la direction supposée du texte et sommés dans cette même direction. Pour rendre le filtrage insensible à la rotation de l'image du document, nous allons calculer les dérivées horizontales et verticales et les sommer dans les deux directions (2). Nous utiliserons une approximation grossière mais rapide pour le calcul des dérivées (3). Le coût du calcul de la sommation en chaque point de l'image dans un voisinage V est trop élevé pour notre application. Nous allons réduire ce coût de calcul en effectuant la sommation par blocs en multi-résolution. Nous divisons l'image en blocs rectangulaires de taille dx×dy puis nous calculons dans chaque bloc la somme des gradients verticaux et horizontaux (Figure 4).

$$J(x_0, y_0) = \frac{1}{dx\ dy} \sum_{i=1}^{dy} \sum_{j=1}^{dx} \left| \frac{\partial I(x_0 + j, y_0 + i)}{\partial x} \right| + \left| \frac{\partial I(x_0 + j, y_0 + i)}{\partial y} \right| \qquad (2)$$

$$\frac{\partial I}{\partial x}(u,v) = I(u-2,v) - I(u+2,v)$$
$$\frac{\partial I}{\partial y}(u,v) = I(u,v-2) - I(u,v+2) \qquad (3)$$

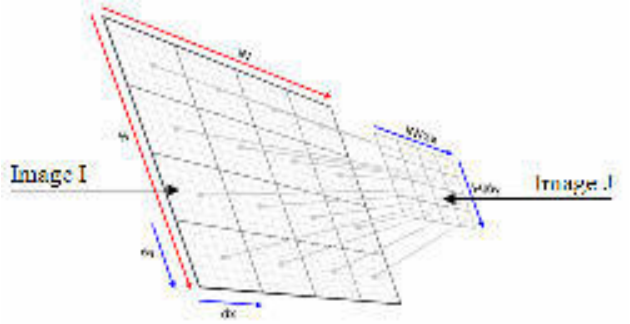


Figure 4 - *Réduction de la taille d'image avec traitement sur le voisinage*

Figure 5 - *Etapes de la binarisation hybride*

Nous obtenons une image J de taille réduite (Figure4, Figure5) dans laquelle les zones claires représentent les zones textuelles dans l'image originale. Cette sommation par bloc ne donne pas les mêmes résultats qu'une sommation en chaque pixel. Nous devons effectuer un prétraitement morphologique sur l'image J pour obtenir un filtrage équivalent à celui de l'algorithme original. Sur cette image J, nous appliquons consécutivement d1 fois l'opérateur dilatation, e1 fois érosions, d2 fois dilatations et e2 fois érosions (4).

$$K = Ee2(Dd2(Ee1(Dd1(J)))) \qquad (4)$$

Le but d'appliquer ces transformations morphologiques est d'une part, de re-densifier le texte et donc de l'agglomérer en blocs, et d'une autre part de prendre une marge suffisante au tour de trait afin d'inclure

l'information pertinente de l'arrière plan (la texture et la couleur) pour un meilleure seuillage. Les paramètres d1, e1, d2, e2 du masque ainsi que la taille de la fenêtre dx×dy ont été fixés pour le moment arbitrairement. On peut constater qu'une augmentation de dx et dy mène à une détection grossière du texte et plus rapide alors que l'augmentation de d1 et e1 détecte mieux les zones de textes agglomérées entre elles. Donc une étude de la stabilité du résultat sur les différents types d'images peut aboutir à un compromis satisfaisant. (e1=d1=2, pour détecter les zones de textes et e1=d1=1, pour détecter les mots). Le surcoût de calcul des opérations morphologiques est négligeable puisqu'il est effectué sur l'image réduite.

3.3 Méthode utilisée pour la binarisation

Nous avons choisi d'utiliser la méthode de Sauvola d'une part pour sa rapidité (table 1) et d'autre part pour ses performances (la méthode Wolf est spécifique aux images vidéo et ne convient pas pour notre application). Le temps économisé nous a permis d'utiliser une grande taille de fenêtre 21x21 pour l'application de l'algorithme de Sauvola ce qui permet d'obtenir de très bons résultas sur les documents imprimés ou manuscrits avec des tailles de caractères très variables.

4 Résultats

Les temps de traitement écoulés sont très proches de ceux écoulés par une binarisation globale et beaucoup moins importants qu'avec les techniques de binarisations adaptatives classiques. Nous avons pu améliorer également les résultats de reconnaissance (TAB2), ce qui a permis à la société d'avoir des résultats de lecture nettement supérieurs à ceux qu'elle avait avant l'utilisation de notre méthode. Tout en sachant que la société utilisait la méthode de binarisation fournie avec le module de l'OCR commercial, les résultats de la reconnaissance sont une moyenne de six jours sur six mois successifs sachant que la société traite moyennement 29225 courriers par jour.

Documents model	Méthode hybride (localisation/ Segmentation)	Amelioration de l' OCR
CCH	0,56	+2%
NPAI	1,19	+26%
BC	1,42	+21
LA3	0,51	+11%
LA4	0,27	+11%
FMR	0,68	+30
PLN	1,12	+20%
HIM	1,64	+76%
TIM	0,23	+16%

Tableau 2 - Vitesses d'exécution de notre algorithme (en secondes) et amélioration de la reconnaissance.

Parallèlement, nous avons pu obtenir de meilleurs résultats sur les enveloppes manuscrites, car c'est sur ce type de document qu'on trouve le plus de variations locales :

- Variation de la taille des caractères : suivant le style d'écriture des gens.
- Variation de l'épaisseur du trait : suivant le stylo, crayon ou fluorescent utilisé.
- Variation de la couleur du trait : suivant la couleur du stylo utilisé.
- Variation du fond due aux différents papiers utilisés pour les enveloppes internes (papier craft, pochette plastique).

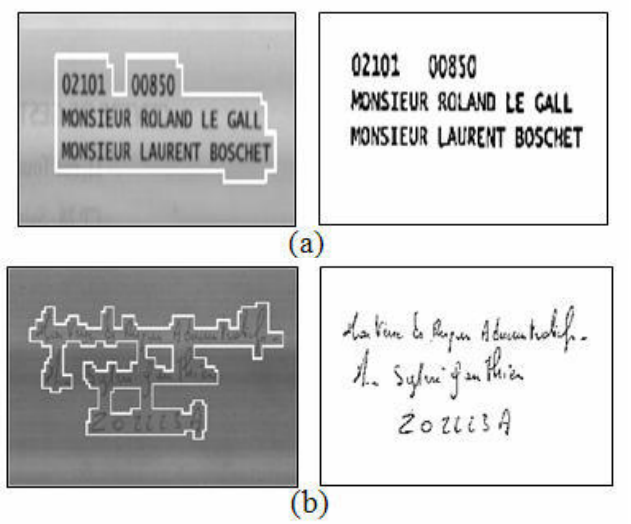


Figure 6 – *Efficacité de notre méthode sur : (a) le texte imprimé, (b) le texte manuscrit (même de faible contraste).*

5 Conclusion et perspectives

Notre méthode nous a permis à la fois de réduire le temps de calcul et d'augmenter la qualité de la binarisation (par une meilleure séparation fond/écriture). Les temps de traitement écoulés sont très proches de ceux écoulés par une binarisation globale et beaucoup moins importants qu'avec des techniques de binarisation adaptatives classiques. Nous avons pu améliorer également les résultats de reconnaissance par l'OCR ce qui a permis à la société d'avoir des résultats de lecture nettement supérieurs à ceux qu'elle avait avant l'utilisation de notre méthode. On peut également, étendre notre combinaison des différentes étapes de reconnaissance pour assurer une

meilleure coopération et interaction entre tous les modules de système de tri.
Ce travail est adopté par la société CESA (www.cesa.fr).

Références

[1] N. Gorski and al, A new A2iA bankcheck recognition system, Handwriting Analysis and Recognition, IEEE Third European Workshop, 1998, pp.1-6.

[2] N. Gorski and al, A2IA check reader, ICDAR'99, pp. 523-526.

[3] U. Miletzki, Documents on the Move, DA&IR-Driven Mail Piece Processing Today and Tomorrow, DAS' 96, pp. 547-563.

[4] S.Srihari, E. Kuebert, Integration of hand-written address interpretation Technology into the United States Postal Service Remote Computer Reader System, ICDAR 97, V.2, pp. 892-896.

[5] Y. Lu and al., An implementation of postal numerals segmentation and recognition system for Chinese business letters, ICDAR99, pp. 725-728.

[6] J. Zhou and al, A feedback-based approach for segmenting handwritten legal amounts on bank cheques, in proc. of ICDAR'01, pp. 887-891.

[7] N. Otsu, A threshold selection method from grey-level histogram, IEEE trans system, man and cybernetics, vol 9, 1979, pp. 62-66.

[8] J. Fisher, S. Hinds, K. D'Amato, A Rule-Based System for Document Image Segmentation, in proc. of the 10th Int'l Conf. Pattern Recognition, 1990, pp. 567-572.

[9] A. Abutaleb, Automatic thresholding of grey-level pictures using two-dimensional entropy, computer vision graphics Image processing, 1985, pp. 22-32.

[10] [Nib86] W. Niblack, An Introduction to Digital Image Processing, Englewood Cliffs, N.J.:Prentice Hall, pp. 115–116, 1986.

[11] J. Sauvola, and al. Adaptive Document Binarization, ICDAR'97, vol 1, pp. 147–152, 1997.

[12] C. Wolf C, J.M. Jolion, F. Chassaing, Text Localization, Enhancement and Binarization in Multimedia Documents, ICPR, 2002, pp. 1037-1040.

[13] O. Deforges, C.Viard-Gaudin, D.Barba, Gray-level Document Image Analysis, 2nd French-Korean Workshop, Man-Machine Handwritten Communication, CNRS Ile de France, 1996, pp. 139-149.

[14] C.Viard-gaudin, D. Barba, Localisation du bloc adresse par une approche multi-résolution, ICDAR91, pp.954-962.

[15] Wahl F, Wong K., Casey G., Block segmentation and text extraction in mixed text/image documents, Computer graphics and image processing, 1982, pp.375-390.

[16] J.C. Oriot, d. Barba, J. Salome, Adress Block Locating Method Based On Transition Analysis Approach: Design And evaluation on flats objects, ICDAR 91, pp.665-673.

[17] J.C. Oriot, D. Barba, M. Gilloux, Localisation du bloc adresse sur les objets postaux par une méthode de segmentation ascendante : évaluation et optimisation, Traitement du Signal, 1995.

[18] B. Yu, A. K. Jain and M. Mohiuddin, Address Block Location on Complex Mail Pieces, in proc. of ICDAR'97, V.2, pp. 897-901.

[19] M. Wolf, H. Niemann, W. Schmidt, Fast Address Block Location on Handwritten and Machine Printed Mail-piece Images, ICDAR 97,V.2, pp. 753-757.

[20] O. Deforges, D. Barba, A fast multiresolution text-line and non text line structures extraction and discrimination scheme for document image analysis, in proc. ICPR 94, pp. 134-137.

[21] A. K. Jain, Y. Chen. Address block location using color and texture analysis, Computer Vision, Graphics and image processing : image understanding, 1994, pp.179-190.

[22] C. Jarousse, C. Viard-Gaudin, Localisation du code postal par réseau de neurones sur bloc adresse manuscrit non contraint, CIFED'98, pp. 72-81.

[23] D. Menoti and al., Segmentation of postal envelopes for address block location : an approach based on feature selection in wavelet space, ICDAR 03, pp. 699-703.

[24] H. Walischewski, Learning regions of interest in postal automation, ICDAR'99, pp. 317-320.

[25] U. Miletzki and al., Continuous learning systems postal address readers with built-in learning capability, ICDAR'99, pp. 329-332.

[26] K. Nitz, An Image-based mail facing and orientation system for enhanced postal automation, ICDAR '03, pp. 694-698.

[27] LeBourgeois F., Robust multifont OCR system from gray level images, fourth ICDAR, International Conference on Document Analysis and Recognition, Ulm, 1997, p. 1-5.

[28] F. LeBourgeoisF. , H. Emptoz H., Document Analysis in Gray Level and Typography Extraction Using Character Pattern Redundancies, Int. Conf. On Doc. Analysis and Recognition ICDAR'99, 1999, India, pp.177-180.

EXTRACTION DES CARACTERISTIQUES PICTURALES DE BOIS GRAVES POUR UNE RECHERCHE DE FORMES

Victor Chen-yuan, Su Ruan

SIC-CReSTIC, groupe Image
IUT-TROYES
9, rue du Québec
10026 Troyes – France
{Prénom.Nom}@univ-reims.fr

Concours Jeune Chercheur: Non

Résumé

Dans cet article, nous proposons de segmenter des images de tampons de bois gravés anciens à des fins d'illustrations de documents patrimoniaux ou de productions de nouveaux tampons à l'aide de l'analyse multi-échelle. L'enjeu de cet archivage est de fournir aussi bien aux utilisateurs professionnels qu'au grand public les contenus des images de tampons sous forme de logos. Notre contribution est de mettre en œuvre une plate-forme qui permette, en partant d'une image de tampon, d'obtenir l'impression qu'il aurait produite sur papier. Nous montrons dans cette communication que la plupart des méthodes classiques de détection de contours échouent sur ces types d'images mêmes si a priori les contenus semblent composés de traits. Les aplats constitués d'empreintes laissées par des outils sont difficilement éliminés et le dimensionnement des contours est rarement pris en compte. Afin de réduire ces fausses descriptions, nous procédons à l'utilisation de traitements basés sur une analyse multi-échelle afin de différencier au mieux les variétés de structures. Les résultats d'extraction obtenus sont présentés et sont perceptuellement satisfaisants tout en montrant une bonne impression de l'image sans avoir à effectuer trop de retouches.

Mots Clefs

Segmentation, détection de contours, décomposition multi-échelle,

1 Introduction

L'archivage d'images numériques est devenu cette dernière décennie un axe de recherche très actif en tant que données visuelles avec l'émergence d'Internet. Avec ce moyen de communication, il devient possible de mettre des documents anciens rares et fragiles, en particulier des tampons de bois gravés, à la disposition du public sur cédéroms ou sur les réseaux ; la communication de ces documents était jusqu'alors interdite en raison de leur fragilité et de leur unicité. Afin de faciliter l'accès au patrimoine archivistique, le projet que nous menons a pour but, outre l'archivage des images originales, la production indexée des images bitonales de leur homologue pour la conservation et la diffusion des tampons de bois gravés. Cette activité a été crée pour répondre à des besoins des consortiums comprenant des organismes en charge du patrimoine, des institutions de recherche et des utilisateurs grand public pour une illustration de documents ou pour une fabrication de nouveaux tampons. La tâche importante à réaliser concerne la segmentation des images qui doit fournir un rendu similaire à l'impression qu'il aurait produit sur papier. Au vu des échantillons d'images (figure 1) qui semblent a priori constituer essentiellement d'images de traits, nous avons initié nos procédures de traitement à l'aide des opérateurs classiques de détection de contours. Malheureusement, ces approches n'ont pas permis de réaliser de segmentation attendue. Il semble que ces méthodes standard de traitement ne sont pas très adaptées aux contenues structurelles des images de tampons. Elles échouent sur le dimensionnement des contours et sur la suppression difficile des empreintes laissées par les outils. En fait, la procédure de segmentation doit, outre l'extraction des contours, requérir une analyse apte à discriminer les aspects relatifs aux textures qui peuvent représenter le fond et les objets picturaux de l'image. Pour y remédier, nous orientons nos travaux vers des représentations multi-échelle. Cette théorie de l'espace-échelle développée par Lindberg [1] est fondée sur l'extraction des régions d'intérêt par la recherche des extrema d'une fonction lissée. L'association d'une région entière à un simple extremum est attirante par la simplicité de représentation. Nous proposons dans ce travail de caractériser les images de tampons en mettant en œuvre l'approche par espace-échelle Gaussien [2]. Le modèle présente non seulement de bonnes propriétés sur le plan de la généralité mais aussi de la bonne fiabilité et de robustesse [3]. L'implantation du filtre gaussien est très pratique, il est stable et séparable. Il peut être implanté récursivement si on cherche un gain de temps de calculs. La suite de l'article est articulé comme suit. Nous introduisons dans la section 2 un rappel du cadre

multi-échelle utilisé pour la caractérisation des images de bois gravés. Dans la section 3, nous décrivons l'opérateur DoG (Differential of Gaussian) utilisé initialement pour la caractérisation des points saillants stables [4] et argumentons de ce choix dans le cadre de cette application. Nous présentons dans la section 4 les différents résultats obtenus et nous concluons l'article par de brèves discussions sur l'automatisation et la qualité de la segmentation des images de bois gravés.

Figure 1 - Présentation d'un échantillon d'image de bois gravé ancien

2 Analyse multi-échelle

L'idée principale de l'approche multi-échelle est de mettre en évidence les différentes tailles de structures de l'image par l'application d'un banc de filtres lissants passe bas. Les filtres de faible suppriment les plus fins tandis que les filtres de plus grand rayon conservent les grandes structures tout en éliminant les structures plus petites. La description de cette procédure peut être illustrée sur la figure 2. Dans la théorie espace-échelle, l'analyse s'appuie de façon implicite sur un noyau constituée d'une fonction Gaussienne de paramètre σ qui sert de facteur d'échelle. L'expression de la fonction Gaussienne bidimensionnelle est définie par :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{\frac{-(x^2+y^2)}{2\sigma^2}} \qquad (1)$$

Les atouts de la fonction Gaussienne sont multiples. Elle permet d'obtenir une description invariante de l'image et de constituer aussi un vecteur de caractérisations d'un point donné de l'image moyennant uniquement des dérivées consécutives de la fonction [7]. L'association de ce vecteur aux différentes composantes produites est exploitée dans de nombreux travaux à des fins d'extraction de points d'intérêts stables et répétables en vue d'appariement ou d'indexation d'images [8][9].

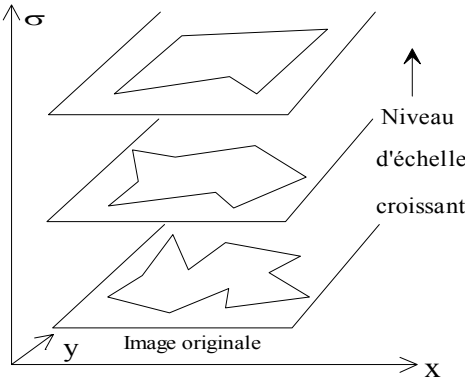


Figure 2 - Illustration de la représentation multi-échelle d'une image

3 Les méthodes de décomposition

Dans cette section, nous introduisons la méthode de décomposition utilisée pour segmenter les images de tampons. Appelons L(x,y,σ) la fonction de lissage résultant du produit de convolution entre l'image I(x,y) et la fonction Gaussienne G(x,y,σ), soit:

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y) \qquad (2)$$

où $\otimes$ représente l'opérateur de convolution.

Habituellement, les caractéristiques locales sont calculées à partir des dérivées Gaussiennes moyennant une échelle σ appropriée. A partir de cette approche, un certain nombre de descripteurs multi-échelle ont été développés pour des applicatifs tels que l'appariement ou l'indexation des images. Ici, nous nous intéressons à la caractérisation de tampons de bois gravés en exploitant le potentiel de filtrage et de détection de contours pour la décomposition multi-échelle. A ce jour, les résultats de segmentation sont jugés subjectivement par une analyse visuelle de l'expert. En vue de s'affranchir du contrôle humain et rendre la binarisation automatique, nous nous recourons à quelques opérateurs optimaux de détection de contours qui nous informent sur l'efficacité de segmentation. Pour cela, nous testons deux opérateurs de contours optimaux choisis suivant une implémentation simple et un coût de traitement équivalent à notre approche.

3.1 Opérateur de Canny

Afin de juger des performances de chacune des approches, nous introduisons l'opérateur de Canny [10] dont l'approche originale a été développée pour permettre la bonne compréhension des conditions d'une bonne détection de contours moyennant les trois critères : bonne détection, bonne localisation et réponse unique. En outre, pour mettre en œuvre son opérateur

en 2D, Canny a montré que la dérivée première d'une fonction Gaussienne est une bonne approximation de son filtre avec des dégradations minimes telle que:

$$h(x,\sigma) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

Le passage à un espace à deux dimensions se fait simplement en raison de la séparabilité de la gaussienne, on a :

$$h(x,y,\sigma) = h(x,\sigma).h(y,\sigma) \quad (4)$$

3. 2 Opérateur de Shen-Castan

L'opérateur de Shen [11] fait partie de la classe de détecteurs dont la réponse impulsionnelle est discontinue en zéro dans le domaine spatial continu. Ce détecteur de contours proposé a la forme d'une fonction exponentielle symétrique. Le filtre de lissage obtenu par Shen s'écrit :

$$h(x) = ce^{-\alpha|x|} \quad (5)$$

et le filtre de dérivation correspondant :

$$h'(x) = \begin{cases} de^{-\alpha\alpha} & \text{pour } x \geq 0 \\ de^{\alpha x} & \text{pour } x < 0 \end{cases} \quad (6)$$

$$\text{avec } c = \frac{1-e^{-\alpha}}{1+e^{-\alpha}}$$

$$\text{et } d = 1-e^{-\alpha}$$

où c et d représentent le facteur de normalisation du filtre de lissage et de dérivation, α est le paramètre d'étalement des filtres.

3. 3 Opérateur Laplacien de la gaussienne

Pour décrire les descripteurs multi-échelle d'une image, le représentant le plus connu d'entre eux est certainement l'opérateur LoG (Laplacian of Gaussian) en raison de sa performance dans la caractérisation des blobs [3]. En général, on utilise le LoG normalisé afin de rendre possible la comparaison des fiabilités de blobs issus d'échelles différentes:

$$\nabla^2_{norm} LoG(x,y,\sigma) = \sigma^2 \left(L_{xx}(x,y,\sigma) + L_{yy}(x,y,\sigma)\right) \quad (7)$$

où $L_{uv}(.)$ représente le résultat du filtrage issu des dérivées spatiales partielles d'ordre 2 de la gaussienne suivant la direction u et v respectivement.

3. 4 Opérateur Différence de la Gaussienne

Initialement introduit par Marr et hildreth [12] pour modéliser les systèmes visuels biologiques pour la segmentation d'images, l'opérateur DoG (Differential of Gaussian) connaît un regain d'intérêt en matière de caractérisation d'images pour la simplicité et la souplesse de paramétrages. A la place de l'opérateur LoG(x,y,σ), nous optons pour le modèle de décomposition l'opérateur DoG(x,y,σ) qui présente des propriétés plus intéressantes. Les raisons de cette préférence sont: 1) l'existence des techniques rapides pour générer des fonctions Gaussiennes modélisant une structure pyramidale; 2) l'opérateur DoG(x,y,σ) est quasi-invariant aux changements d'échelles évitant la normalisation des résultats par σ^2; 3) la répétabilité du DoG(x,y,σ) dans le cadre de la détection de coins [5] et la fiabilité de la localisation de ces points d'intérêt [6]. Nous proposons donc une alternative intéressante basée sur l'opérateur DoG(x,y,σ) pour extraire outre les contours les informations basées sur les caractéristiques des régions. L'opérateur proposé est obtenu en différenciant simplement deux fonctions gaussiennes séparées par un facteur multiplicateur k:

$$\begin{aligned} DOG(x,y,k\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) \otimes I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \quad (8)$$

Nous voyons que le résultat de l'équation (8) est primordial dans la description des caractéristiques qui se composent uniquement de la soustraction de deux fonctions lissées à des échelles différentes. Ce modèle évite ainsi d'introduire la constante de normalisation comme dans l'utilisation du Laplacien $\sigma^2\nabla^2 LoG(x,y,\sigma)$ étudié par Lindberg [1] afin de s'assurer de l'invariance des caractéristiques à différentes échelles. Nous illustrons sur la figure 3a et 3b une représentation en 2D de l'opérateur LoG et de l'opérateur DoG utilisés pour la détection des contours qui se caractérisent par la recherche du passage par zéro pour le premier et la pyramide des erreurs pour le deuxième. Nous observons que les deux graphes sont quasi-similaires à une constante près donnée par [13]:

$$DoG(x,y,\sigma) = 0.4875\sigma^2 LoG(x,y,\sigma) \quad (9)$$

Malgré leur similitude, la différence principale entre les deux opérateurs réside dans la pratique. Outre les qualités citées précédemment, le DoG permet une scrutation rapide, les objets suivis sont plus rapidement appréhendés. Ainsi, on trouve le DoG comme moteur de recherche dans la procédure de mean-shift [13] connue pour traquer les blobs contenus dans une image.

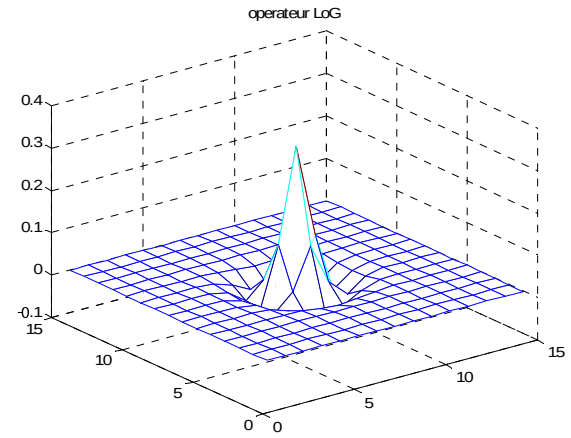


Figure 3a - représentation en 2D de l'opérateur LoG

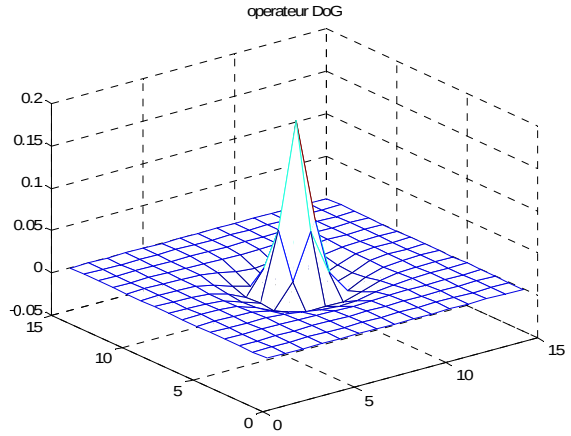


Figure 3b - Représentation en 2D de l'opérateur DoG

4 Expérimentations

Dans ce paragraphe, nous présentons des résultats expérimentaux montrant l'attrait de l'utilisation de la fonction Gaussienne pour la caractérisation et la segmentation des images de tampons de bois gravés. Les expérimentations sont initialement menées sur la base des opérateurs différentiels classiques (gradient, Sobel). Sur l'application visée ensuite, dans le but de s'affranchir de l'analyse visuelle de l'expert et réaliser une binarisation automatique éventuelle, nous avons recouru à quelques filtres optimaux de détection de contours (Canny, Shen) pour une évaluation prédictive des résultats extraits de la carte de saillance ou de contours. L'objectif ici est d'obtenir une évaluation de la qualité des traits de silhouettes et des frontières moyennant ces dispositifs de calculs. Le mode de sélection est-il en adéquation avec le rendu du traitement donné par l'application de la décomposition multi-échelle. Dans son implantation, le filtre de Canny utilisé est constitué du noyau issu de la dérivée première de la Gaussienne, le filtre de Shen est implémenté sous forme de noyau et non pas de séquences récursives, l'opérateur Laplacien LoG(x,y,σ) est réalisé à partir de la dérivée seconde de la Gaussienne. Nous comparons enfin les résultats obtenus à ceux fournis par le DoG(x,y,σ). Le protocole de segmentation utilisé est très classique : 1) Si la caractérisation est opérée suivant deux directions privilégiées en ligne et en colonne, la binarisation est effectuée selon la technique de recherche des maxima locaux et la construction des contours par hystérésis ; 2) Lorsque la caractérisation s'opère sans direction particulière, l'extraction des contours est réalisée à partir d'un seuillage global. Nous illustrons sur la figure 4 un banc de résultats réalisés par les différents opérateurs. Du point de vue de l'analyse de scènes, les résultats obtenus (Figure 4a) par l'application des opérateurs classiques sont inappropriés, ils ne respectent pas les formes picturales du tampon. Les opérateurs utilisés apparaissent mal adaptés aux tâches de sélection des caractéristiques dominantes (évolution des traits, analyse de texture du fond, discrimination des objets). Nous observons aussi que les opérateurs à double composante directionnelle ne conviennent pas aux types d'images traités. Les traits sont altérés ou entrecoupés. Cette inadéquation de structures pose alors de problème dans la représentation ou la restitution d'objets cartographiques. Les méthodes d'inspirations analytiques sont également mal adaptées à l'interprétation des images. Les images segmentées (Figure 4b) par des filtres optimaux montrent au premier regard une uniformité de contours et un difficile problème de stylisation de rendu. Leur conjonction n'a pas non plus permis d'aboutir à une évaluation de la qualité des caractéristiques extraites. Nous présentons sur la figure 4c les résultats provenant de l'analyse gaussienne multi-échelle. A ce jour, les meilleurs résultats de segmentation sont fournis par les deux approches avec une restitution des caractéristiques plus fine pour le DoG. Les ornements de la tête sont visiblement mieux ressortis avec cet opérateur. Pour cette image caractéristique, la valeur de résolution σ permettant de réaliser une cartographie de contours convenable avec une perte minimale se situe entre 1 et 2. L'extraction des traits caractéristiques est réalisée à l'aide d'un seuillage global. Une expérimentation plus systématique est en préparation notamment l'utilisation de l'approche sur d'autres séries de tampons pour juger de sa stabilité et de sa reproductibilité.

5 Conclusion

Nous avons présenté une approche de segmentation de tampons de bois gravés en s'appuyant sur une technique de décomposition multi-échelle de la Gaussienne. Par l'intermédiaire de cette approche, nous obtenons actuellement une bonne impression de l'image sans avoir à effectuer trop de retouches. Les résultats rencontrés pour une échelle σ compris entre 1 et 2 sont peceptuellement satisfaisants. Dans le cadre de cette étude, nous nous sommes également

confrontés sur la position des traits à la surface des objets. Ainsi, la représentation de certains traits par les opérateurs de contours est incohérente à la perception visuelle, ce sont des traits laissés par les outils et sont considérés comme étant l'image de fond. Globalement, les résultats donnés par l'analyse multi-échelle sur la segmentation d'images de tampons médiévaux sont encourageants. Certes, des points durs subsistent, l'automatisation de la segmentation est somme toute rédhibitoire en raison de la diversité de matériaux utilisés et de la variété de tampons à traiter. La qualité de segmentation est également très tributaire des phénomènes liés aux problèmes d'encrage et aux accidents subits par le bois au cours de son existence. L'intervention de l'expert s'avère nécessaire pour juger de la bonne binarisation du tampon. Des améliorations ultérieures peuvent être envisagées en mettant en place une procédure de sélection pertinente de crêtes et de vallées à base de système d'attention visuelle moyennant des modèles de représentations de connaissances adaptées.

Références

[1] T. Lindberg, *Scale-space theory in computer vision,* Kluwer Academic Publishers, Pays-bas, 1994.

[2] T. Lindberg, Feature detection with automatic scale selection, *International Journal of Computer Vision*, vol. 30, n°2, pages 79-116, 1998.

[3] R. Megret, J.M. Jolion, Suivi de blobs de niveaux de gris pour la représentation du contenu dynamique d'un vidéo, *Rapport de Recherche RR-2001-05, RFV, INSA de Lyon*, Septembre 2001.

[4] D.G. Lowe, Object recognition from local scale invariant features, *International Conference on Computer Vision (ICCV)*, pages 1150-1157, 1999.

[5] K. Mikolajczyk, C. Schmid, Indexing based on scale invariant interest point, *International Conference on Computer Vision*, pages 525-531, 2001.

[6] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, vol. 60, n°2, pages 91-110, 2004.

[7] J.J. Koederink, A.J. Van Doorm, Representation of local geometry in the visual system, *Biological Cybernetics*, 55, pages 367-375, 1987.

[8] C. Schmid, R. Mohr, Local greyvalue invariants for image retrieval, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, n°5, pages 530-535, 1997.

[9] Y. Dufournaud, C. Schmid, R. Horaud, Appariement d'images à des échelles différentes, *Actes du 12ème Congrés Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, France, vol. 2, pages 327-336, 2000.

[10] J. Canny, A computational approach to edge detection, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 8, n°6, pages 679-698, 1986.

[11] J. Shen, S. Castan, An optimal linear operator for edge detection, *IEEE Conference Vision Pattern Recognition*, pages 109-114, 1986.

[12] D. Marr, E. Hildreth, Theory of edge detection, *Proc. Roy. Soc. London*., vol. B 207, pages 187-217, 1980.

[13] R. T. Collins, Mean-shift blob tracking through scale space, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, pages 234-240, 2003.

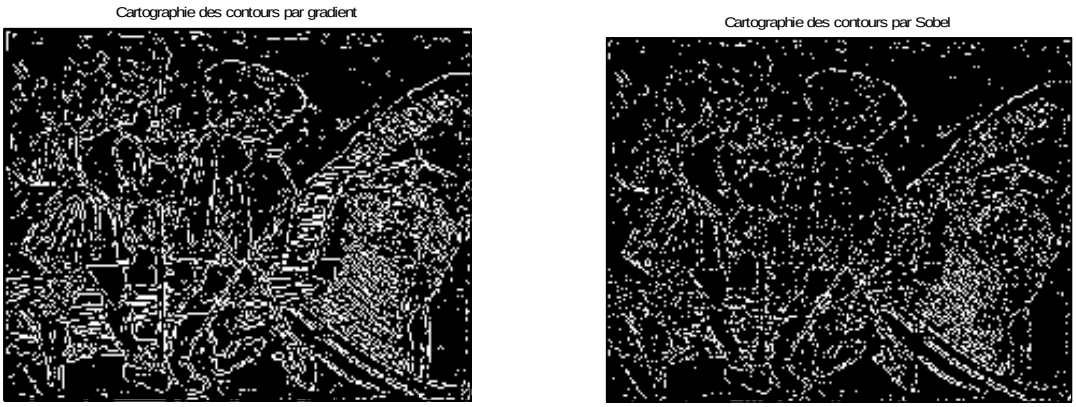


Figure 4a - Cartographie des contours par des opérateurs de contours classiques

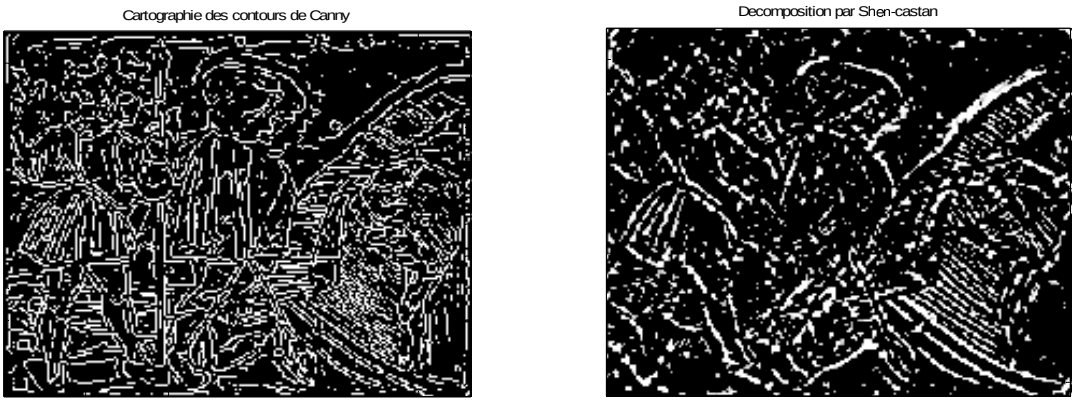


Figure 4b - Cartographie des contours par des opérateurs optimaux des contours

Figure 4c - Cartographie des contours par l'analyse gaussienne multi-échelle

Reconnaissance biométrique sans contact de la main intégrant des informations de forme et de texture

Julien Doublet[1], Marinette Revenu[2], Olivier Lepetit[1]
1: France Telecom, 42 rue des coutures, B.P. 6243, 14066 CAEN Cedex 4, France
2: GREYC – ENSICAEN, 6 Bd Maréchal Juin, 14050 CAEN Cedex, France
E-mail: julien.doublet@orange-ft.com

Résumé

La reconnaissance biométrique de la main a été développée avec succès pour l'authentification ou l'identification biométrique. Dans ce papier, nous proposons une méthode originale de reconnaissance biométrique combinant des informations de couleur, de texture et de forme. Tout d'abord, la segmentation intègre les composantes couleurs de la peau et un modèle de forme. Ensuite, le processus d'authentification fusionne par convolution les caractéristiques géométriques des doigts et la texture de la paume analysée. Les résultats expérimentaux montrent le bien fondé de cette approche avec un taux d'erreur d'authentification inférieur à 2% pour une population de 16 individus.

Mots clefs

Reconnaissance de la main, biométrie, processus de fusion, détection de la peau, modèle de forme actif.

1 Introduction

La biométrie joue un rôle de plus en plus important dans les systèmes d'authentification et d'identification. Les processus de reconnaissance biométrique permettent la reconnaissance d'individus en se basant sur les caractéristiques physiques ou comportementales. Différentes technologies ont été développées telles que: les empreintes digitales, l'iris, le visage, la voix, la signature et la main. Cette dernière méthode s'appuie sur une étude de la forme de la main et de la texture de la paume. Elle présente de nombreux avantages par rapport aux autres technologies. Premièrement, le système de capture est moins coûteux que celui pour la reconnaissance d'iris, les caractéristiques de la main sont plus nombreuses que celles des empreintes digitales et elles peuvent être déterminées avec des images à faible résolution. De plus, ce système est bien accepté par les utilisateurs et la main laisse peu de traces contrairement à un système basé sur l'empreinte digitale. Ce type de système est ainsi prôné par la Commission Nationale de l'Informatique et des Libertés.
L'architecture des systèmes biométriques est organisée en 4 étapes: l'acquisition des données, le traitement du signal pour l'amélioration des données et la segmentation de la modalité analysée, l'extraction des caractéristiques et la comparaison avec une ou plusieurs références. Dans notre système, l'étape 2 correspond à la détection de la main dans une image complexe. Les méthodes de Skin Blob Tracking, de contours actifs, de Mean Shift ou de condensation [1] sont couramment utilisées dans les systèmes d'interactions homme-machine. Ces processus présentent cependant deux contraintes majeures incompatibles avec une reconnaissance biométrique. Ils fournissent une détection peu précise de chaque doigt de la main et ils nécessitent une séquence d'images. Pour augmenter la qualité de la segmentation nous proposons une méthode combinant les informations de couleur de peau et de forme de main.
La troisième étape consiste en une caractérisation de la forme et de la paume de la main. De nombreux travaux ont été réalisés ces dernières années. La forme de la main peut être décrite par la largeur de la paume et la forme des doigts. Chaque doigt est défini classiquement par sa longueur, un ensemble de largeurs et la courbure de son extrémité. Initialement, la paume de la main a été caractérisée par ses lignes [2] en utilisant une méthode proche de la reconnaissance des empreintes digitales. D'autres systèmes utilisent l'information globale de la paume de la main [3-5] et ils présentent de meilleures performances en reconnaissance. Pour aboutir à des systèmes plus fiables, il est possible de fusionner les décisions obtenues à partir de la paume et de la forme de la main [6].
Dans ce travail, nous proposons une méthode de reconnaissance de la main sans contact. La section 2 introduit le processus de détection de la main. La section 3 décrit l'extraction des caractéristiques de la forme et de la paume de la main ainsi que la méthode de fusion de ces caractéristiques. Les sections 4 et 5 présentent les résultats expérimentaux et donnent les conclusions, respectivement.

2 Segmentation de la main dans un fond complexe

Dans un système de reconnaissance biométrique, un processus de segmentation précis et rapide doit être élaboré. Dans cette section, nous décrivons rapidement notre méthode de détection de la main basée sur une

combinaison d'une modélisation de la couleur de la peau et d'un modèle de forme [7].

2.1 Modélisation de la couleur de la peau

Contrairement aux modèles basés sur les classifieurs de Bayes ou les modèles de mélanges de Gaussiennes [8], la teinte de peau est modélisée par apprentissage supervisé. Pour obtenir un bon compromis entre la vitesse d'exécution et la précision de la détection, nous utilisons un réseau de neurones (RN). Les entrées du RN sont composées par trois neurones, une pour chaque composante couleur des pixels dans le domaine RGB. La sortie du RN est la probabilité qu'un pixel soit un pixel de peau. L'entrainement du réseau à partir d'une base de données de pixels de peau et de fond dans le domaine RGB permet la modélisation de la couleur de peau. Parallèlement, une analyse en composantes principales [9] sur les pixels de peau de la base établit un domaine couleur spécifique à la peau.

2.2 Segmentation de la peau

Dans la phase de détection, le RN calcule la probabilité que chaque pixel soit un pixel de peau. Ce processus construit la carte des probabilités (Fig. 1). Pour obtenir une segmentation proche du temps réel, un processus multi-résolution construit la carte de probabilités. Un seuillage et un renforcement du contraste sont appliqués sur cette carte pour obtenir une image binarisée.

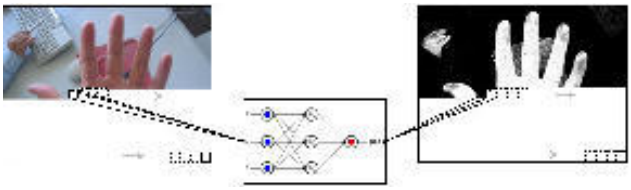
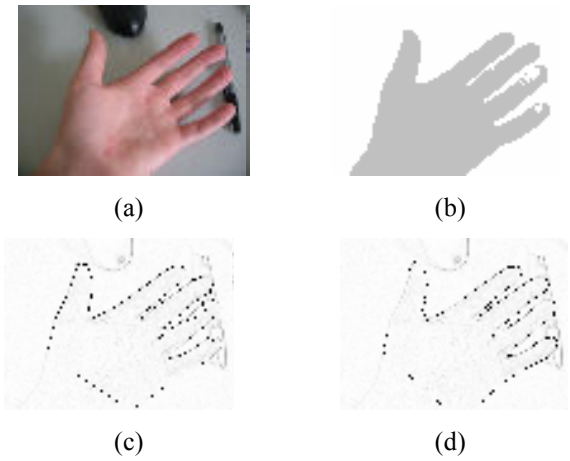


Figure 1 - *Calcul de la carte des probabilités*

2.3 Segmentation de la main par modèle de forme

La segmentation par la couleur de la peau ne peut pas effectuer de façon robuste la tâche de détection de la main. Un modèle spécifique de forme active [10] est défini pour résoudre ce problème. Les deux difficultés majeures dans ces modèles actifs de forme sont l'initialisation du contour qui doit être proche de la forme à rechercher et la convergence du modèle dans la phase de détection.
Classiquement, la forme à détecter est définie par une série de points: les landmarks. Dans la phase d'apprentissage de la forme, la forme moyenne et les variations du contour sont calculées par analyse en composantes principales sur une base de mains annotées par ces landmarks. Dans la phase de segmentation, le contour est initialisé par les points caractéristiques de la main: les cinq points au bout des doigts et les quatre points situés dans la vallée entre deux doigts adjacents. Ces points sont calculés à partir de la carte de l'image binarisée par analyse du contour. Deux autres points sont automatiquement ajoutés près du poignet à partir de ces points. Les autres landmarks définissant plus précisément la forme de la main sont disposés entre ceux-ci. Ainsi le modèle X est définie par les 11 points initiaux et les N points intermédiaires disposés entre ceux-ci. On obtient X par $X = (X[0], \ldots, X[11 + N \times 5 \times 2 - 1])$ où $X[i]$ est le ième landmark. Après la phase d'initialisation, le modèle est déformé. Pour maitriser le problème de la divergence du modèle qui ne suit pas les contours réels de la main, un poids est appliqué aux déformations pour limiter les contraintes de formes [7]. Pour que le terme gradient ne fasse intervenir que les contours de la main et limite ainsi les possibilités de divergence de la forme, il est déterminé dans l'espace de couleur de peau par l'algorithme de Di Zenzo [11]. Ce gradient est ensuite pondéré par le coefficient des pixels de la carte de probabilités. Les expérimentations montrent qu'un bon compromis entre le temps d'exécution et la précision de la détection est obtenu en fixant N à 12. Le processus de détection complet est illustré à la Figure 2.



(a) (b)

(c) (d)

Figure 2 - *Processus complet: (a) image originale (b) segmentation par la peau (c) contour initial (d) contour final*

3 Extraction des caractéristiques

Cette section explicite les caractéristiques de la paume et de la forme de la main ainsi que la méthode de fusion de ces informations pour une reconnaissance biométrique performante. Dans la section 3.1, l'extraction de la paume et la spécification de la texture par un filtre de Gabor sont décrites. Les caractéristiques de la forme de la main sont définies à partir de ses contours. Finalement, la fusion de toutes les caractéristiques par un processus de convolution est détaillée dans la section 3.2.

3.1 Calcul des caractéristiques de la forme et de la paume

Après avoir détecté la main, il est nécessaire de localiser la paume indépendamment de la distance entre la main et le capteur. La détermination de la paume s'appuie sur la dimension de la main et la méthode de localisation de paumes décrite dans [3]. Dans cet article, deux valeurs sont fixées: la distance entre les points *O1* et *O2* et la taille de la paume ||*A1A2*|| (Fig. 3). Ces valeurs, constantes dans les systèmes de reconnaissance classiques, sont ici définies suivant la taille de la main. Elles sont déterminées à partir de la largeur de la main calculée par la distance Euclidienne entre les points *X[L1]* et *X[L2]* où *L1* et *L2* sont des indices fixés après expérimentations à 30 et 125. Ainsi, ||*O1O2*|| et ||*A1A2*|| sont définis par:

$$||O1O2|| = \alpha \, ||X[L1]X[L2]|| \qquad [1]$$

$$||A1A2|| = \beta \, ||X[L1]X[L2]|| \qquad [2]$$

Où α et β sont les coefficients de dimensions choisis à 1/10 et 2/3, respectivement. Ensuite la paume est redimensionnée à une taille fixe M*M où M est fixé à 100.

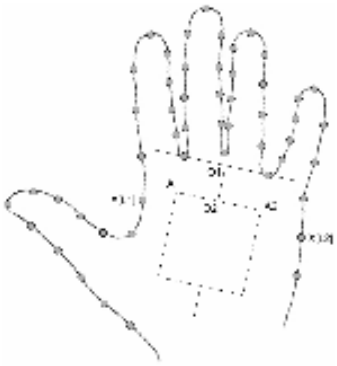


Figure 3 - *Extraction de la paume avec N=4*

La paume extraite contient des lignes principales qui peuvent être déterminées par une méthode spécifique d'extraction [2]. Ces lignes ne sont pas propres à chaque individu, il est donc nécessaire d'utiliser les lignes secondaires de la main. Ces lignes plus fines ne peuvent pas être extraites de la paume avec des images à faible résolution, ainsi une caractérisation globale de la paume est préférable.

Différentes méthodes permettent d'obtenir les caractéristiques de la paume: l'analyse en ondelettes, la transformée de Fourier, l'Analyse en Composantes Principales, le filtre de Gabor… Grace à ses bonnes performances en reconnaissance de l'iris et de la paume et à ses qualités propres: localisation précise en temps/fréquence et robustesse aux variations de contraste et de luminosité, nous avons utilisé un filtre de Gabor. Différentes implémentations de ce filtre existent. Dans [3], un filtre de Gabor 2D dans le domaine complexe est utilisé. Pour limiter le temps de calcul et la taille des caractéristiques, le filtre dans le domaine réel décrit dans [12] est employé:

$$G(x,y) = \exp\left[-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right] \times \cos\left(2\pi\frac{x'}{\lambda}+\varphi\right) \qquad [3]$$

$$x' = (x-\xi)\cos\theta - (y-\eta)\sin\theta$$

$$y' = (x-\xi)\sin\theta - (y-\eta)\cos\theta$$

Où le couple *(ξ, η)* définit le centre de la fonction, θ contrôle l'orientation de la fonction, σ est la déviation standard de l'enveloppe gaussienne, γ est le ratio de l'aspect visuel fixé à 0.5, λ est la période de l'onde et φ est la phase. σ est défini par le rapport constant σ/λ=0.56 [12]. Pour plus de robustesse à la luminosité, le filtre est centré au point (0,0) en utilisant pour un filtre de taille (2k+1)² la formule:

$$\Omega(x,y) = G(x,y) - \frac{\sum_{i=-k}^{k}\sum_{j=-k}^{k} G(i,j)}{(2k+1)^2} \qquad [4]$$

Ainsi les caractéristiques de la paume de la main sont obtenues par le résultat de la convolution de l'image de la paume avec ce filtre de Gabor robuste par:

$$C = I * \Omega \qquad [5]$$

Où * est l'opérateur de convolution. Pour compléter la représentation, les caractéristiques de la forme de la main sont extraites. Il s'agit des largeurs et des longueurs des doigts. Les longueurs et les largeurs sont approximées par la distance entre les points du modèle de main et sont définies par:

$$L[i] = d\big(m(X[H_{2i}], X[H_{2i+2}]), X[H_{2i+1}]\big)$$

$$l[i][j] = d\big(X[H_{2i+1} - j], X[H_{2i+1} + j]\big) \qquad [6]$$

Où *d(a, b)* est la distance Euclidienne entre les points a et b, *m(a, b)* retourne le milieu du segment ||ab||, *L[i]* est la longueur du ième doigt, *l[i][j]* est la jème largeur du ième doigt et $H_i = i(N+1)$ est l'indice du ième point caractéristique de la main (Fig. 4).

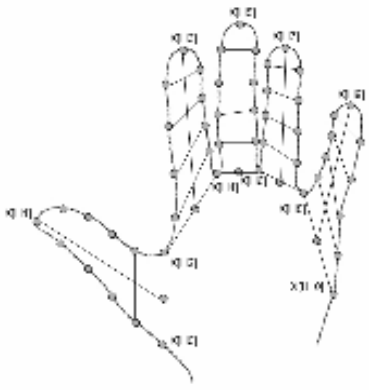


Figure 4 - *Extraction des données de formes avec N=4*

3.2 Fusion de la texture et de la forme

Dans les systèmes biométriques, trois méthodes de combinaison de données sont utilisées. La combinaison peut être effectuée à la représentation en regroupant les caractéristiques extraites, au niveau de la comparaison ou au niveau des décisions. Notre fusion est basée sur la convolution de la texture de la paume avec les caractéristiques de la forme de la main afin d'ajouter un facteur géométrique à la texture. Le résultat de cette convolution est binarisé pour limiter la taille des caractéristiques et les temps de calcul dans la phase de comparaison. Le processus complet est défini par:

$$S(x,y) = b(C(x,y) * H) \qquad [7]$$

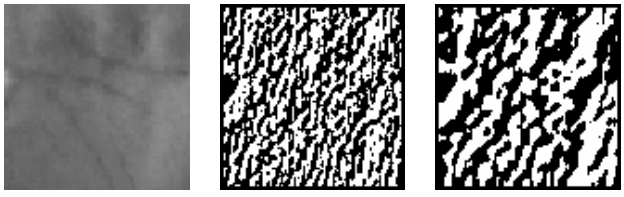
Où b(x) = 0 si x<0 et b(x) =1 sinon, * est l'opérateur de convolution et H est un filtre de taille 5x5 correspondant aux caractéristiques de formes. Il peut être explicité par:

$$H(x,y) = H'(x,y) \times \frac{1}{\sum_{i=0}^{4}\sum_{j=0}^{4} H'[i][j]} \qquad [8]$$

Où H'=

L[0]	l[0][1]	l[0][2]	l[0][3]	l[0][4]
L[1]	l[1][1]	l[1][2]	l[1][3]	l[1][4]
L[2]	l[2][1]	l[2][2]	l[2][3]	l[2][4]
L[3]	l[3][1]	l[3][2]	l[3][3]	l[3][4]
L[4]	l[4][1]	l[4][2]	l[4][3]	l[4][4]

Le coefficient de normalisation de H permet d'être robuste à la distance entre la main et le système d'acquisition. La fusion par convolution des données caractéristiques augmente l'unicité de la main. Elle permet en effet de distinguer des personnes ayant des formes de main ou des paumes très similaires. L'extraction de la texture seule et la combinaison de la texture et de la géométrie de la main sur une paume sont illustrées à la Figure 5.



(a) (b) (c)

Figure 5 – *Extraction des caractéristiques (a) paume extraite, (b) texture de la paume, (c) convolution de la texture et de la forme*

4 Expérimentations

Dans cette section, nos résultats expérimentaux sont indiqués pour valider cette approche convolutionnelle. Premièrement, la méthode de comparaison des mains est indiquée. Ensuite, la base de données de mains et l'évaluation des performances du système sont décrites.

4.1 Comparaison de mains

Pour la comparaison, les caractéristiques de la main *S* sont assimilées à une matrice. Une méthode classique de comparaison de matrices binaires est ainsi appliquée: la distance de Hamming normalisée. Cette distance est une comparaison pixel par pixel et elle donne une réponse normalisée entre 0 et 1, 0 étant la correspondance parfaite. Elle est définie pour deux caractéristiques de mains *X* et *Y* par:

$$D_0(X,Y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} X(i,j) \oplus Y(i,j)}{M^2} \qquad [9]$$

Où $\oplus$ est l'opérateur ou-exclusif. La segmentation de la main ainsi que la détection de la paume n'étant pas parfaites, une tolérance en translation et en rotation est appliquée au processus de comparaison. Cette comparaison souple s'exprime pour deux caractéristiques de mains *X* et *Y* par:

$$D_r(X,Y) = \min_{|s|<S,|t|<T,|a|<A} D_O(R(T(X,s,t),a),Y) \qquad [10]$$

Où *T(X, s, t)* est la translation de l'image *X* horizontalement par *s* et verticalement par *t* et *R(X, a)* est la rotation de l'image X par un angle a. Les constantes *S* et *T* sont fixés à 2 pixels et *A* est limité à 2° afin de limiter le coût de calcul.

4.2 Évaluation des performances

Une base de données spécifique est élaborée pour valider notre approche. Toutes les images de la base ont été acquises avec une webcam Philips ToUcam Pro 740K à la résolution 640x480 pixels. La base contient 160 images, certaines avec des mains possédant des bagues, provenant de 16 personnes. 10 images sont acquises pour chaque individu de la base.

Pour obtenir les coefficients optimaux et le meilleur filtre représentant la forme, un ensemble de paramètres est utilisé pour valider notre approche. Premièrement, un banc de filtre est créé pour déterminer le meilleur score de reconnaissance pour la paume seule. Chaque paume de la base de mains est comparée avec toutes les autres afin de déterminer ce score. Le taux de reconnaissance est le rapport entre le nombre de comparaisons incorrectes et le nombre total de comparaisons. Une comparaison est incorrecte si un utilisateur est accepté à tort ou si un utilisateur est rejeté à tort. Deux taux définissent ces erreurs: le taux de faux acceptés (FAR) défini par le rapport entre le nombre de personnes authentifiées à tort et le nombre de comparaisons et le taux de faux rejetés (FRR) défini par le rapport entre le nombre de personnes rejetées à tort et le nombre de comparaisons.

Les paramètres de tests pour le filtre de Gabor sont inspirés de [12], l'orientation du filtre est testée suivant huit valeurs θ=22.5°, θ=45°,…, θ=180°, trois valeurs sont utilisées pour la fréquence spatiale λ=5.47, λ=8.20 et λ=10.93, le couple (ξ, η) et la phase φ sont fixés à (0,0) et 0 respectivement, tandis que la déviation standard σ est déterminée suivant le rapport σ/λ=0.56. Le meilleur taux de comparaison présenté par la courbe de la Figure 7 est obtenu avec les paramètres θ=157.5°, λ=10.93 et σ=6.12. Avec ces coefficients, l'EER (Equal Error Rate) correspondant à FAR=FRR indique une erreur de 2.25%. Le filtre de Gabor complexe présenté en [3] avec les paramètres optimaux montrent des résultats similaires sur notre base de données. Pour diminuer les erreurs de reconnaissance, nous avons présenté une nouvelle méthode de fusion des caractéristiques de forme et de texture de la paume. Après expérimentations, le filtre optimal définissant la forme est seulement composé des largeurs des doigts. Ainsi, le filtre optimal H est défini par:

$$H(x, y) = I(x, y+1) \times \frac{1}{\sum_{i=0}^{4} \sum_{j=0}^{4} I[i][j+1]} \qquad [11]$$

Les performances du système de reconnaissance globale sont bien augmentées entre la reconnaissance par la paume seule et la reconnaissance globale de la main (Fig. 6). En effet, l'EER est égal à 1.85% et un système plus sécurisé peut être défini avec FAR=10^{-3}% et FRR=2.2%.

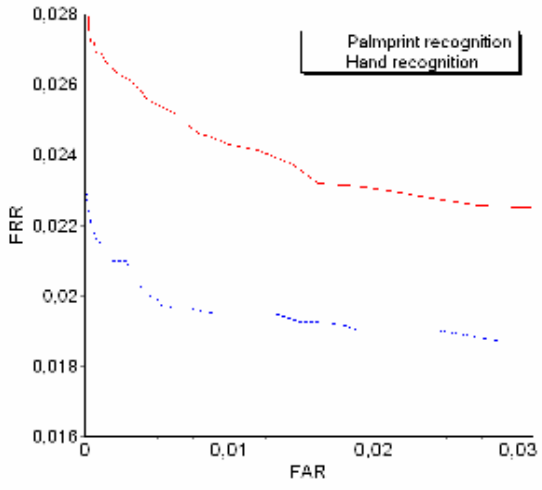


Figure 6 - *Reconnaissance globale*

Le processus de reconnaissance complet (segmentation, extraction des caractéristiques et comparaison) est effectué en moins de 1 seconde sur un Pentium M à 1.60GHz.

5 Conclusion

Dans ce papier, nous avons présenté une nouvelle méthode de reconnaissance biométrique de la main pour un système sans contact. Tout d'abord, la segmentation de la main est expliquée. Elle est effectuée grâce à une intégration des composantes couleurs de la peau et un modèle de forme de main. Ensuite, le processus d'authentification par fusion est décrit. Il s'appuie sur une combinaison par convolution des données géométriques de la main et de la texture de la paume. Les caractéristiques de la paume sont déterminées par un filtre de Gabor 2D dans le domaine réel permettant une représentation compacte. Les données du contour de la main sont extraites du modèle de forme ayant permis la segmentation. Le processus complet est validé après expérimentations. Il présente un taux d'erreur de reconnaissance de 1.85% pour un temps d'exécution inférieur à 1 seconde.

Afin de gérer la rotation dans l'espace de la main, une méthode invariante aux perspectives et aux cisaillements devra être recherchée. De plus, la base de tests devra être diversifiée et complétée pour confirmer nos approches. Ces deux pistes font actuellement l'objet de notre travail.

6 Références

[1] Y. Zhu. Hand Detection in an Active Vision System. Thèse, 2003.

[2] N. Duta, A. K. Jain, K. V. Mardia. Matching of palmprints. Pattern Recognition Letters: 23(4), 477-485, Février 2002.

[3] W. Kong, D. Zhang, W. Li. Palmprint feature extraction using 2-D Gabor filters. Pattern Recognition: 36(10), 2339-2347, Octobre 2003.

[4] G. Lu, D. Zhang, K. Wang. Palmprint recognition using eigenpalms features. Pattern Recognition Letters: 29(9-10), 1463-1467, Juin 2003.

[5] X. Q. Wu, K. Q. Wang, D. Zhang. Wavelet Based Palmprint Recognition. Conference on Machine Learning and Cybernetics, Beijing, Novembre 2002.

[6] A. Kumar, C. M. Wong, C. Shen, A. K. Jain. Personal Verification using palmprint and hand geometry biometric. Proc. Intl. Conf. Audio Video based Biometric Personal Authentication, Washington D. C., Mars 1999.

[7] J. Doublet, O. Lepetit, M. Revenu. Hand detection for contact less biometrics identification. Intl. Conf. Cognitive System with Interactive Sensors, Paris, Mars 2006.

[8] L. Lucchese, S. Mitra. Color image segmentation: A state of the art survey. Proc. of the Indian National Science Academy (INSA-A), New Delhi, India: 67(2), 207-221, Mars 2001.

[9] L. I. Smith. A tutorial on Principal Components Analysis. 2002.

[10] T.F. Cootes, C.J. Taylor. Statistical models of appearance for computer vision. Technical report, University of Manchester, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, Manchester M13 9PT, United Kingdom: 1999.

[11] S. Di Zenzo. A note on the gradient of a multi-image. Computer Vision, Graphics and Image Processing: 33(1), 1986.

[12] P. Kruizinga, N. Petkov. Nonlinear Operator for Oriented Texture. IEEE Trans. on Image Processing: 8(10), 1395-1407, 1999.

Estimation du suivi du ventricule gauche du cœur par Modèle d'Etat Harmonique

EVINA EKOMBO P. Lionel
Laboratoire LISQ*
evinalio@yahoo.fr

OUMSIS Mohammed
Laboratoire LISQ*
oumsis@fsdmfes.ac.ma

MEKNASSI Mohammed
Laboratoire LISQ*
m_meknassi@yahoo.fr

*Faculté de Sciences Dhar-el-Mahraz, Fès
Concours jeune chercheur : Oui

Résumé

Nous avons développé une nouvelle méthode de suivi de contour fermé qui se base sur un modèle d'état harmonique (MEH), afin de réaliser le suivi du ventricule gauche (VG) du cœur durant tout le cycle cardiaque. Cette méthode nous fournit les trajectoires des points du contour du VG, information nécessaire à l'analyse du mouvement cardiaque. Le vecteur d'état généré par le MEH permet une modélisation robuste et correcte de contour fermé. Nous nous appuyons sur ce vecteur d'état et nous l'utilisons comme descripteur local d'une région du VG. Le suivi du mouvement est réalisé par une recherche de vecteurs d'état similaires durant tout le cycle cardiaque. L'application en simulation de notre méthode donne des résultats satisfaisants. Sur les données réelles extraites de séquences ciné-IRM, les trajectoires calculées, des points du contour du VG tout au long du cycle cardiaque, nous permettent d'avoir une différence clairement visible entre un cœur sain et un cœur pathologique.

Mots clefs

Modèle d'état harmonique, Suivi de mouvement, Mesure de distance, Imagerie cardiaque, images Ciné-IRM.

1 Introduction

Les maladies cardiovasculaires, importante cause de mortalité dans le monde entier, sont provoquées majoritairement par une perturbation de la fonction contractile du cœur qui est réalisée par le myocarde, et en particulier le ventricule gauche (VG). Le suivi du mouvement ventriculaire gauche permet donc le dépistage d'un grand nombre de maladies cardio-vasculaires. L'imagerie médicale fournit un support incontournable pour le diagnostic cardiaque. Malheureusement, son interprétation directe est rendue difficile à cause de la faible résolution et la forte présence de bruits. L'emploi des techniques de modélisation spatio-temporelle est une excellente alternative qui donne beaucoup de paramètres pour l'analyse. Son utilisation respecte trois étapes : segmentation du contour du VG à partir des images cardiaques et modélisation (2D/3D) de la forme du VG, suivi temporel du mouvement du VG et enfin l'analyse des paramètres calculés lors des deux étapes précédentes pour l'émission du diagnostic. Dans ce travail, nous nous intéressons à la deuxième étape. Le suivi du mouvement du VG est une phase clé dans le processus d'analyse de séquences cardiaques. Il apporte des informations intéressantes sur l'évolution temporelle du VG.

Actuellement, le suivi pèche encore par son manque de précision ou son coût de calcul élevé lorsque le résultat est satisfaisant. L'objectif de cet article est de présenter une nouvelle méthode de suivi qui se veut à la fois précise et adaptée au suivi de formes périodiques comme le ventricule gauche. Elle s'appuie sur un Modèle d'Etat Harmonique (MEH) [1]. Ce modèle offre une bonne modélisation d'un contour fermé (une évolution périodique) et dispose d'un vecteur d'état robuste qui exploite le filtre de Kalman pour son estimation.

Dans [1], nous avons proposé une modélisation du mouvement cardiaque (le mouvement ventriculaire gauche) sous forme d'un modèle d'état harmonique et linéaire. Ce modèle de mouvement combine trois caractéristiques essentielles pour un mouvement ventriculaire à savoir, l'accès à la dynamique cardiaque sur l'ensemble du cycle, une robustesse certaine aux bruits et l'interprétation physique directe de paramètres fonctionnels du VG. Le modèle est linéaire, périodique et traduit un modèle dynamique correspondant à la décomposition en série de Fourier du mouvement cardiaque. Utilisé comme modèle d'état dans un filtre de Kalman, ce modèle offre l'avantage de fournir une estimation robuste aux bruits, des paramètres du mouvement comme la vitesse et l'accélération qui sont des composantes du vecteur d'état du modèle. Jusqu'à présent, le modèle d'Etat Harmonique (MEH) a été exploité dans une dimension temporelle pour modéliser le mouvement du VG. La périodicité de la forme du VG (surface fermée) nous permet également une transposition du modèle dans une dimension spatiale à un instant donnée introduisant ainsi des contraintes de forme et de lissage via la décomposition harmonique. Cette double caractéristique révèle l'intérêt potentiel d'un tel modèle pour le suivi 2D/3D de la paroi du VG dans une séquence d'images. Dans ce travail, nous proposons une application du modèle MEH dans une dimension spatiale afin de suivre les déformations locales des régions du VG.

Ce papier est scindé en quatre parties. La première partie est plus théorique et présente le modèle MEH, précédé par un état de l'art, et s'achève sur notre méthode de suivi. Puis en deuxième partie, nous présentons la validation de notre méthode dans une simulation, qui fait ressortir les éléments caractéristiques de la méthode. La

troisième partie est consacrée à la mise en œuvre sur des séquences de données réelles de patients. En fin nous achevons ce document avec une conclusion et des perspectives.

2 Description de la méthode

2.1 Etat de l'art

De nombreux travaux ont été menés afin de fournir des outils d'aide au diagnostic. Ces travaux vont de la segmentation du cœur dans des séquences d'images à son analyse, en passant par la modélisation de la forme et le suivi du mouvement du cœur. Les recherches sur le suivi se divisent en deux groupes suivant le type de mouvement approché : rigide et non-rigide. Le mouvement du VG est non-rigide. Les contours actifs [2] assurent ce type de suivi et les travaux portant sur ce sujet foisonnent. Mais, ils sont sensibles à l'information contenue et ne tiennent pas compte des détails liés à la forme du VG, par exemple la périodicité du VG. Les méthodes de recalage fournissent des résultats intéressants, mais cela nécessite le plus souvent des modèles temporels assez complexes [3,4]. L'Iterative Closest Point (ICP) présentée par Besl et Mc Kay [5] est une méthode très utilisée pour le suivi. Des versions améliorées ne cessent d'être faites, elles portent notamment sur la mesure d'appariement (distance, flot optique, …) [6,7,8] ou encore la fonction de transformation (rigide ou non-rigide) [8,9]. Il existe aussi des travaux dont la mesure de distances a retenu notre attention de par leurs résultats, il s'agit plus précisément des travaux de Wang [10], Papademetris [11] et Geiger [12].

Dans ce document, nous proposons une nouvelle méthode d'estimation du mouvement non-rigide qui vienne combler les manques recensés dans les méthodes mentionnées. Notre méthode prend en considération la périodicité du contour du VG et l'exploite par le biais du MEH. Il est à noter que le MEH [1] a déjà servi pour la modélisation temporelle du mouvement du VG (mouvement périodique). Ce modèle assure également la réduction de l'impact du bruit des mesures en réalisant un filtrage des données par le filtre de Kalman.

2.2 Modèle d'état harmonique pour un contour fermé

Dans chaque plan de coupe du coeur, le VG est modélisé par un contour fermé. Celui-ci peut être décrit par une fonction continue et périodique $\rho(\theta)$ obtenue par développement du contour autour de son centre de gravité. Cette fonction est parfaitement caractérisée par un nombre fini d'échantillons :

$$\rho_l = \rho\left[(\theta_0 + l\Delta\theta)\right] \text{ avec } l=1,...,N \quad (1)$$

l est le numéro de l'échantillon par rapport à θ, N le nombre des échantillons et $\Delta\theta = 2\pi/N$ le pas d'échantillonnage en angle θ. La fonction $\rho(\theta)$ présente une périodicité par rapport à θ (θ varie de *0* à 2π). Si on considère la décomposition en série de Fourier, cette fonction peut s'écrire sous la forme suivante :

$$\rho(\theta) = \overline{\rho} + a_1 \sin(\omega\theta + \varphi_1) + ... + a_n \sin(n\omega\theta + \varphi_n) \quad (2)$$

Le terme $\overline{\rho}$ est la valeur moyenne, ω la pulsation et n le nombre d'harmoniques. Les coefficients a_i et φ_i représentent respectivement les amplitudes et les phases des différents harmoniques.

Dans [1], nous avons montré qu'une telle évolution peut être modélisée par un modèle d'état harmonique d'ordre *n*. Ce modèle a été exploité dans une dimension temporelle pour modéliser le mouvement du VG (mouvement périodique). La périodicité de la forme du VG (surface fermée) nous permet également une transposition du modèle dans une dimension spatiale, introduisant ainsi donc des contraintes de forme et de lissage via la décomposition harmonique. Cette double caractéristique révèle l'intérêt potentiel d'un tel modèle pour le suivi 2D et 3D de la paroi du VG. Dans ce travail, nous proposons de modéliser la surface 2D du VG (contour fermé) par un MEH [1]. Ce modèle est un modèle dynamique, linéaire et permet d'associer une décomposition harmonique en série de Fourier (2) à un modèle dynamique linéaire (4). Utilisé comme MEH dans un filtre de Kalman, ce modèle offre l'avantage de fournir une estimation robuste aux bruits ainsi que des paramètres du vecteur d'état du modèle (3). Ce vecteur est composé de *2n+1* éléments :

$$R(\theta) = (\overline{\rho}, \rho(\theta), ..., \rho^{(2n-1)}(\theta))^T \quad (3)$$

$\rho^{(j)}(\theta)$ représente la dérivée d'ordre j par rapport à θ.

L'équation d'état du modèle dynamique est sous la forme suivante :

$$R(\theta + \Delta\theta) = F^{\theta} R(\theta) + \zeta_p(\theta) \quad (4)$$

La matrice F^{θ} est une matrice de transition calculée en fonction du pas d'échantillonnage $\Delta\theta$ [1], et $\zeta_p(\theta)$ est un bruit gaussien à moyenne nulle.

Le vecteur d'état $R(\theta)$ est un descripteur 2D local du contour du VG au niveau d'un point de contrôle. Pour une position θ, le contour du VG peut être restitué par une série de multiplications récursives du vecteur d'état $R(\theta)$ et de la matrice de transition F^{θ}.

Ce modèle permet de générer autant de points que l'on souhaite sur le contour et de leur associer un vecteur d'état qui les caractérisent.

2.3 La méthode de suivi

L'objectif de notre méthode est de pourvoir suivre les points du contour du VG tout au long du cycle cardiaque, et d'être capable de reconstruire leur trajectoire.
Comme le vecteur d'état du MEH est un descripteur local, le suivi du mouvement est effectué au niveau de chaque point par une recherche de vecteurs d'états similaires. Etant donné un point de contrôle p du contour du VG, C_t à l'instant t, de vecteur d'état R_{pt}, nous déterminons lequel des points du contour C_{t+1} à l'instant t+1, après mouvement, possède un vecteur d'état très proche de R_{pt}.
Afin de réaliser ceci, nous faisons appel à plusieurs méthodes de mesure de distance entre vecteurs. Après une étude des mesures existantes (Manhattan, Hausdorff, …) [13] et de multiples simulations, nous retenons deux mesures pour leurs excellents résultats : la distance euclidienne et la mesure de corrélation.

$$D_{Eucl}(X,Y)=\sqrt{\sum_{i=0}^{n-1}(x_i-y_j)^2} \quad (5)$$

avec X, Y des vecteurs de taille n

$$Corr(X,Y)=\frac{Cov(X,Y)}{Cov(X,X)*Cov(Y,Y)} \quad (6)$$

avec $Cov(X,Y)=E(XY)-E(X)*E(Y)$
où $E(X)$ est la moyenne du vecteur X [14]

L'algorithme correspondant à notre méthode de suivi est présenté sur la figure 1.

L'ouverture angulaire de la fenêtre de recherche, qui définit sa taille, influe sur les résultats de la méthode. Des tests de simulations nous ont permis de lui attribuer la valeur de *20°*.
Mesure de proximité : Pour élire un point q_j de la fenêtre de recherche comme étant le plus proche de p_i, la valeur de la mesure entre Rp_i et Rq_j doit dans le cas de :

a. la distance euclidienne, minimiser la distance entre les deux vecteurs ;
b. la corrélation, maximiser la vraisemblance, soit donc avoir la mesure la plus proche de 1 en valeur absolue.

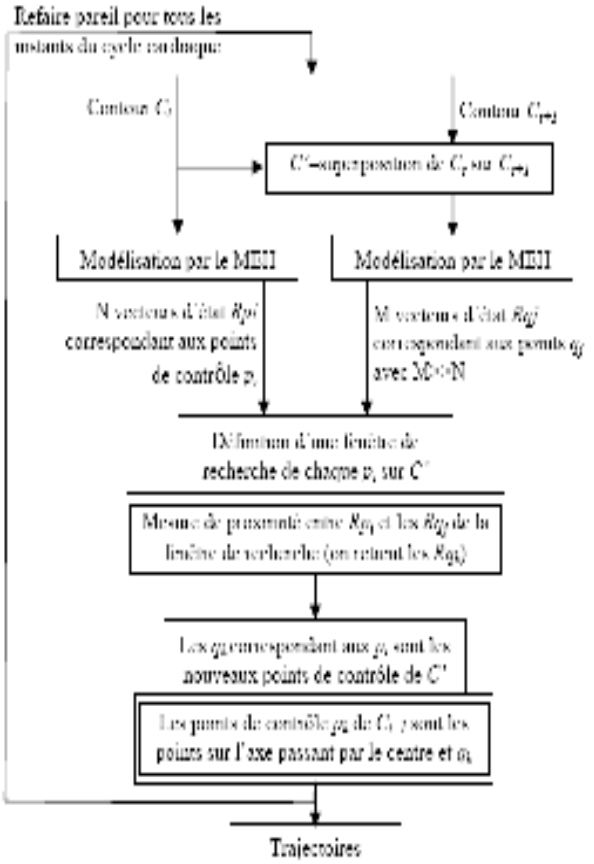


Figure 1 – *Algorithme de suivi*

3 Validation en simulation

La simulation se fait à partir d'un contour de VG extrait d'une image ciné-IRM. Sur ce contour, nous appliquons une succession de mouvements faibles (rotation, translation, dilatation/contraction) ou une combinaison de ces mouvements. Le choix de la mesure de distance est déterminé lors de cette simulation. Un autre élément déterminant dans notre méthode est le choix de l'ordre du modèle, nous trouverons le meilleur ordre pour assurer un suivi correct.

Le contour utilisé pour la simulation comporte *24* points. Lors de la modélisation, nous générons *200* points et sélectionnons *20* d'entre eux comme points de contrôles afin de faire leur suivi. Les mouvements appliqués, de faible amplitude, sont de 5 pixels pour chaque point du contour. La translation se fait suivant les deux axes.

3.1 Mesure de distance

D'après la section **2.3**, les mesures de distance donnant de bons résultats en accord avec notre méthode sont la distance euclidienne et la mesure de corrélation. Les figures 2 et 3 donnent un aperçu des correspondances

obtenues après applications d'un mouvement de translation. La comparaison de ces figures permet de retenir la mesure de corrélation comme étant la mesure assurant le meilleur suivi.

3.2 Ordre du modèle

La transformée de Fourier Discrète comparée au MEH de même ordre fourni une plus grande erreur de modélisation. Dans [1], l'ordre 3 a été retenu comme étant celui qui offre un bon compromis pour la modélisation. Cependant, cet ordre 3 ne renferme pas toutes les informations sur la forme modélisée. La variation de l'ordre du modèle a été faite durant les tests de simulations, les résultats sont mentionnés sur les figures 2 et 3. La lecture de ces figures nous permet de retenir le modèle d'ordre 5, indépendamment du type de mesures utilisées. C'est en parfaitement harmonie avec la théorie, car le fait d'augmenter l'ordre permet d'avoir plus d'harmoniques, donc plus d'informations sur la forme locale du contour.
Il est à noter que le calcul du modèle MEH pour un ordre supérieur à 7 prend un long temps de calcul dû aux grandes valeurs à manipuler.

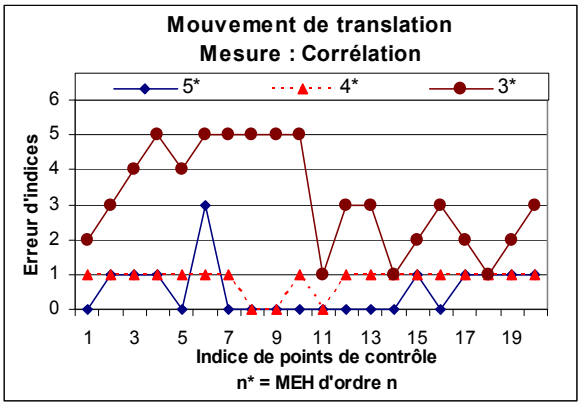


Figure 2 – *Mesure de corrélation pour divers ordre du modèle MEH*

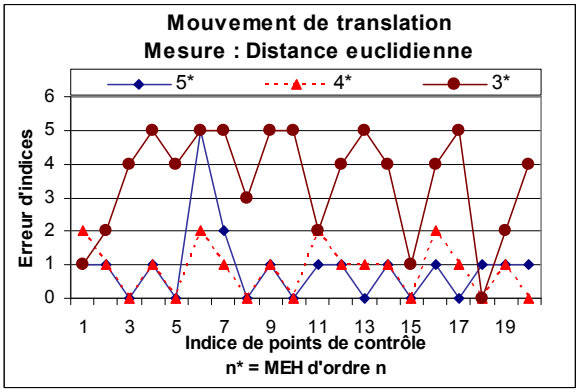


Figure 3 - *Mesure de distance euclidienne pour divers ordre du modèle MEH*

3.3 Mouvement de simulation

Dans un premier temps, nous observons le comportement du suivi après application de mouvements indépendants de rotation, de translation et de contraction (figure 4).

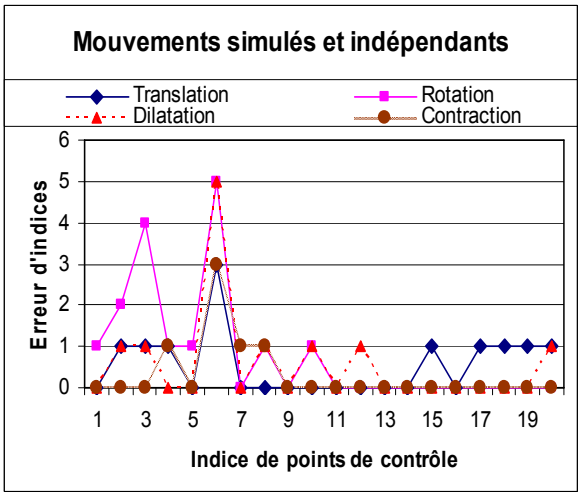


Figure 4 – *Mouvement simulé et indépendant de translation, de rotation, de dilatation et de contraction.*

Puis nous réalisons un mouvement similaire à celui fait par le VG lors du relâchement, à savoir un mouvement combiné de translation, rotation et dilatation. Le résultat du suivi se trouve sur la figure 5 et la figure 6.

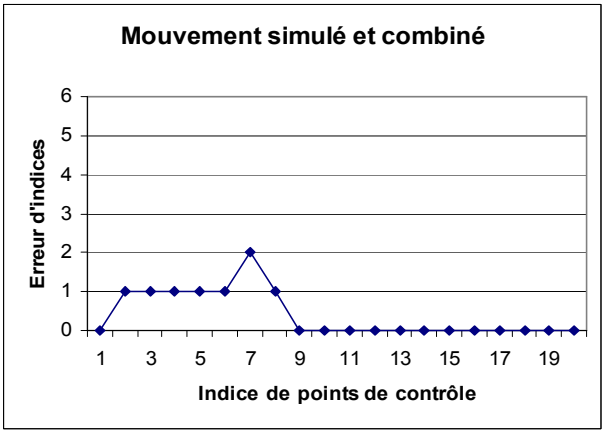


Figure 5 – *Mouvement simulé et combiné (translation, rotation et dilatation)*

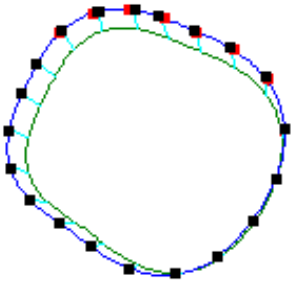


Figure 6 – *Trajectoires des points après un mouvement simulé du VG*

Comme on peut le remarquer sur les figures 5 et 6, les correspondances sont encourageantes. Nous passons au suivi du mouvement réel de VG à des données extraites des examens radiologiques.

4 Résultats sur des données réelles

La validation de la méthode de suivi se poursuit sur des données extraites de séquences d'images ciné-IRM acquise chez des patients sains et malades. Les séquences d'images sont de type 2D suivant la coupe du VG. La détermination des trajectoires des points, appartenant aux contours extraits, devrait nous permettre d'émettre un avis sur l'état des cœurs analysés (sain ou pathologique).

4.1 Extraction de contours

Pour les différentes images de la séquence, nous délimitons les parois internes et externes du VG par des contours.
Ces contours constituent les points d'entrée du MEH. Le choix des points de contrôle, dont on observera les trajectoires, est fait automatiquement.

La figure 7. montre le cœur dans deux états qu'il occupe durant le cycle cardiaque.

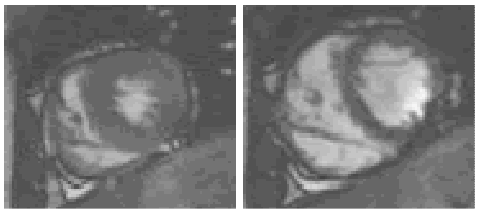


Figure 7. - *Image ciné-IRM du cœur pendant la systole, à gauche, et la diastole à droite.*

4.2 Séquences d'images ciné-IRM

Nous appliquons la méthode de suivi sur deux séquences d'images issues des examens de deux patients différents.
Les séquences d'images dont nous disposons contiennent 16 images successives qui couvrent entièrement le cycle cardiaque. Nous suivons *20* points de contrôle sur le cycle.

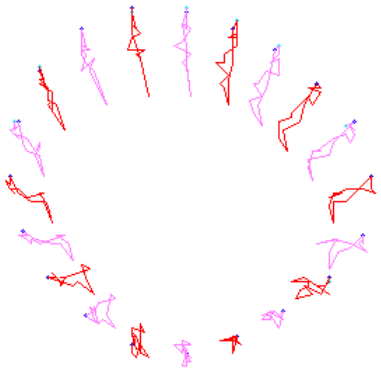


Figure 8. – *Trajectoires estimées des points de contrôle de la paroi interne du VG*

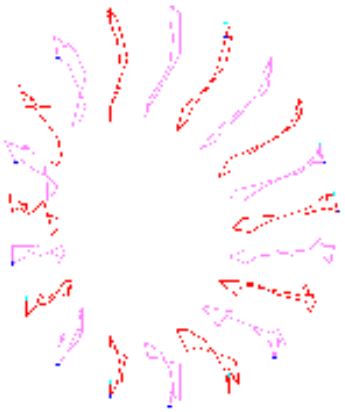


Figure 9. – *Trajectoires estimées des points de contrôle de la paroi interne du VG*

4.3 Analyse des résultats

L'analyse et la comparaison des deux cœurs, ayant servis pour notre test, est directement faisable à la lecture des figures précédentes.
La figure 8. montre des trajectoires très écrasées dans la partie gauche inférieure (septale et postérieure), bien au delà du fait de la proximité du ventricule droit. Certaines trajectoires sont complètement écrasées.
La figure 9. avec des trajectoires amples et allongées, correspond le mieux au mouvement réalisé par le VG d'un cœur sain.
Le VG du cœur de la figure 8. peut alors être considéré comme étant pathologique.

5 Conclusion et perspectives

Nous avons vu que les méthodes de modélisation et de suivi sont nombreuses dans la littérature. Il importe pour une méthode de suivi du VG d'avoir une technique qui réalise un bon suivi, tout en tenant compte des spécificités qu'il possède et des possibles informations à conserver. Nous nous sommes appuyés sur le MEH pour notre

méthode de suivi, car il offre une bonne modélisation sur contour fermé. Notre méthode exploite la modélisation spatiale du modèle, qui n'avait pas encore été faite, ainsi que sa résistance aux bruits. Les correspondances fournies dans le cadre des simulations sont quasiment exactes, et les résultats sur des données réelles sont très encourageants. Notre méthode permet donc de faire facilement et de manière instantanée la distinction entre un cœur pathologique et un cœur sain par la reconstruction des trajectoires.

Le nombre de séquences utilisés lors de la simulation était limité, mais les résultats obtenus nous amènent à passer à une étape de validation sur une plus grande échelle. Cette étape de validation, nous permettra également de faire la comparaison entre notre méthode et les meilleures techniques existantes comme entre autre la méthode HARP [15].

Nous comptons poursuivre ce travail sur deux principaux volets : tout d'abord, améliorer le tracé des trajectoires, d'où une meilleure précision, par l'application d'un filtre ; ensuite, reproduire le même procédé sur des enveloppes de plus grandes dimensions.

Références

[1] M. Oumsis, A. D. Sdigui, B. Neyran et I.E. Magnin. Modélisation et suivi par modèle d'état harmonique du mouvement ventriculaire gauche du cœur en Imagerie par Résonance Magnétique. Dans *Traitement du Signal 2000* – volume 17 – n 5/6 – pages 501-516.

[2] Kass, M., Witkin, A., et Terzopoulos, D. Snakes : Active Contour Models. *International Journal of Computer Vision, volume 1* pages 321-331, 1988.

[3] J. Declerck, J. Feldmar, and N. Ayache, Defininion of a 4D continuous planispheric transformation for the tracking and the analysis of left-ventricle motion, *Med. Image Anal.*, vol. 2, no. 2, pp. 197–213, 1998.

[4] J. Huang, D. Abendschein, V. Davila-Roman, and A. Amini, "Spatio-temporal tracking of myocardial deformations with a 4-D B-spline model from tagged MRI," *IEEE Trans. Med. Imag*., vol. 18, no. 10, pp. 957–972, Oct. 1999.

[5] P. J. Besl and N. D. McKay, A method for registration of 3-D shapes, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(2):239-256, February 1992.

[6] J. Declerck, J. Feldmar, N. Ayache, Definition of a 4D continuous polar transformation for the tracking and the analysis of LV motion*, INRIA*. N° 3039, November 1996.

[7] S. Benayoun, N. Ayache and I. Cohen, An adaptive model for 2D and 3D dense non rigid motion computation, *Technical report 2297, INRIA*, May 1994.

[8] M. Sühling, M. Arigovindan, Myocardial Motion Analysis from B-Mode Echocar-diograms*, IEEE Transactions on image processing*, VOL., N° 4, April 2005.

[9] M. J. Ledesma-Carbayo, J. Kybic, M. Desco, A. Santos and M. Unser, Cardiac Motion Analysis from Ultrasound Sequences Using Non-rigid Registration. *MICCAI 2001*, p. 889-896, 2001.

[10] Y. Wang, B. S. Peterson and L. H. Staib, Shape-based 3D surface correspondence using geodesics and local geometry. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, p. 644-651, 2000.

[11] X. Papademetris, J. Sunisas, Dione and Duncan, Estimation of 3D left ventricular deformation from Echocardiography, *Medical Image Analysis In Press*, March 2001.

[12] Geiger, Gupta, Costa, and Vlontzos, Dynamic Programming for Detecting, Tracking, and Matching Deformable Contours, *IEEE Transactions PAMI* 17(3)294-302, 1995.

[13] Dengsheng Zhang, Guojun Lu, Evaluation of similarity measurement for image retrieval, *Neural Networks and Signal Processing*, pages 928-931 vol 2, 2003.

[14] Murray R. Spiegel, Statistique, cours et problèmes, *Série Schaum, McGraw-Hill, Paris* 1993.

[15] N.F. Osman and J.L. Prince, "Direct calculation of 2D components of myocardial strain using sinusoidal MR tagging", *to appear in Proc. of the SPIE Medical Imaging: Image Processing Conference*, *San Diego, California*, 1998.

Détermination du nombre de classes par le principe du maximum d'entropie pour des classes en chevauchement

A. LACHKAR[1], O. AMMOR[2], N. RAIS[3]
[1]E.S.T. M, Université Moulay Ismail, Meknès Maroc. E-mail: abdelmonaime_lachkar@yahoo.fr
[2]Laboratoire LMCS FSTF USMBA , Fès, Maroc, E-mail : w_ammor@yahoo.fr.
[3]Laboratoire ISQ. FSDM, USMBA, Fès, Maroc. E-mail : raissn@gmail.com

Résumé

Nous présentons un nouvel indice pour la détermination du nombre de classes basé sur le Principe du Maximum d'Entropie (V_{MEP}). La procédure est complètement automatique. Les performances de V_{MEP} sont illustrées à travers des exemples simulés et réels. Cet indice montre une grande robustesse, et une supériorité par rapport à d'autres méthodes existantes et récentes, particulièrement dans le cas du chevauchement spatial.

Mots clefs

Classification non supervisée, Principe du Maximum d'Entropie, chevauchement, nombre de clusters.

1 Introduction

La classification est une notion qui intervient fréquemment dans la vie courante. En effet, il est souhaitable de regrouper les éléments d'un ensemble hétérogène, en un nombre restreint de classes les plus homogènes possibles. Son application a joué un rôle très important pour résoudre plusieurs problèmes en reconnaissance des formes, imagerie, segmentation d'images couleur, data mining,…et dans différents domaines comme la médecine, la psychologie, la biologie, etc.

Nous parlons de classification non supervisée, ou regroupement, lorsqu'on ne dispose d'aucune information a priori sur les variables à traiter ; et de classification supervisée autrement. Le travail développé dans cette recherche s'inscrit dans le cadre des techniques de classification non supervisée, qui s'apparente à la recherche des groupes homogènes au sein d'un mélange multidimensionnel où le nombre de groupes est inconnu. Les résultats de classification obtenus dépendent fortement du nombre de classes fixé. Il est donc primordial de choisir le nombre exact de classes pour espérer avoir une bonne qualité de classification. Ceci n'est pas toujours simple, surtout en présence de chevauchement.

Plusieurs approches ont été proposées sur ce sujet pour différentes applications [1]-[7] Cependant, pour les mêmes données, on peut obtenir des résultats différents selon le nombre de classes k fixé par l'utilisateur. Pour des classes bien séparées, les algorithmes de classification retrouvent généralement le même nombre de clusters.

Le problème se pose dans le cas de chevauchement de classes : rares sont les algorithmes qui arrivent à détecter le nombre réel de classes, et ils deviennent invalides pour un degré de chevauchement relativement fort.

Le processus d'évaluation des résultats des algorithmes de classification est appelé indice de validité des clusters. Trois critères sont en général utilisés [8]: Externe, Interne et Relatif. Les deux premiers sont basés sur des méthodes statistiques et demandent beaucoup de temps de calcul [9]. Comme il est mentionné par Maria et al [10], les techniques basées sur le Critère Relatif citées dans la littérature [11]-[16], fonctionnent correctement dans le cas de classes compactes et sans chevauchement. Cependant, plusieurs applications présentent différents degrés de chevauchement, et l'application de ces algorithmes reste limitée.

Dans cet article, nous présentons une nouvelle méthode de détermination du nombre optimal de classes d'un mélange multidimensionnel basée sur le principe du maximum d'entropie.

Dans la prochaine section, nous présentons quelques critères de validité les plus utilisés, ainsi que leurs limites et inconvénients. La section 3 détaillera notre nouvel indice de validité noté V_{MEP}. Les résultats expérimentaux sur des exemples réels et artificiels sont présentés dans la section 4, montrant l'efficacité et la robustesse de notre nouvel indice, particulièrement dans le cas du chevauchement spatial entre classes. On finira par la conclusion dans la section 5.

2 Indices de validité basés sur les critères relatifs

Les algorithmes de classification floue (Fuzzy C-means FCM) ont été largement utilisés pour obtenir les k-partitions floues. Cet algorithme suppose la fixation a priori du nombre de classes k par l'utilisateur, ce qui n'est pas toujours possible. Différentes partitions sont ainsi obtenues pour différentes valeurs de k. Une méthodologie d'évaluation est requise pour déterminer le nombre optimal de clusters k^*. C'est ce qu'on appellera indice de validité des clusters (cluster validity index).

Le processus pour le calcul de l'indice de validation des clusters est résumé par les étapes suivantes:

Etape 1 : Initialiser les paramètres des FCM excepté le nombre de clusters k.
Etape 2 : Appliquer l'algorithme FCM pour différentes valeurs de k avec k=2,3,…,cmax. (cmax est fixé par l'utilisateur).
Etape 3 : Calculer l'indice de validité pour chaque partition obtenue à l'étape 2.
Etape 4 : Choisir le nombre optimal k^* de clusters.
Plusieurs indices de validité de clusters sont proposés dans la littérature. Bezdek a défini deux indices: le Coefficient de partition (V_{PC}) [17] et l' Entropie de Partition (V_{PE}) [18]. Ils sont sensibles au bruit et à la variation de l'exposant m. D'autres indices V_{FS} et V_{XB} sont proposés respectivement par Fukayama et Sugeno [19] et Xie-Beni [20]; V_{FS} est sensible aux valeurs élevées et basses de m, V_{XB} donne de bonnes réponses sur un large choix pour c=2,…,10 et $1<m\leq 7$. Cependant, il décroît rapidement avec l'augmentation du nombre de clusters. Kwon et al. [21] ont apporté une amélioration à cet indice. Maria Halkidi et al. [15] ont défini V_{S_Dbw} basé sur les propriétés de compacité et de séparation de l'ensemble des données. Cet indice donne de bons résultats en cas de classes compactes et bien séparées, notamment quand il n'y a pas de chevauchement. Do-Jong Kim [22] a proposé un nouvel indice V_{SV}, en se basant sur la sommation des deux fonctions sous-partitionnement et sur-partitionnement. Cet indice s'est avéré plus performant que les autres cités auparavant.
Plus récemment, un nouvel indice de validité V_{OS} proposé par Dae-Won Kim et al en 2004 [23], exploite une mesure de séparation et une mesure de chevauchement entre clusters. Il est défini comme le rapport entre le degré de chevauchement et de séparation. La mesure du degré de chevauchement entre les clusters est obtenue en calculant le degré de chevauchement inter clusters. La mesure de séparation est obtenue en calculant la distance entre les clusters. D'après les auteurs [23], l'indice V_{OS} est plus performant que plusieurs autres indices. Cependant, il reste incapable de déterminer le nombre réel de clusters dans l'exemple des Iris [23], où il y a un réel chevauchement.

3 Nouvel indice de validité proposé V_{MEP}

3.1 Principe du maximum d'entropie

Considérons un ensemble de données avec k clusters $c_1 \ldots c_j \ldots c_k$, et leurs centres respectifs $g_1 \ldots g_j \ldots g_k$. On définit les probabilités P_{ij} comme le lien entre le point i de sa classe c_j (j obtenu préalablement par l'algorithme de FCM) et son centre g_j. Les points i qui n'appartiennent pas à la classe c_j, ne possèdent aucun lien avec g_j ; c'est-à-dire $P_{ij}=0$.

On a : $$\sum_{i\in c_j} P_{ij} = 1 \; pour j = 1.....k \qquad (1)$$

Pour toutes les classes, on obtient :

$$\sum_{j=1}^{k}\sum_{i\in c_j} P_{ij} = k \text{ , et par suite } \sum_{j=1}^{k}\sum_{i\in c_j}\left(\frac{P_{ij}}{k}\right) = 1$$

On définit une entropie qui mesure l'information apportée par toutes les classes par :

$$S = -\sum_{j=1}^{k}\sum_{i\in c_j}\left(\frac{P_{ij}}{k}\right)\ln\left(\frac{P_{ij}}{k}\right) \qquad (2)$$

$$S = -\frac{1}{k}\sum_{j=1}^{k}\sum_{i\in c_j} P_{ij}\,\ln(P_{ij}) + \ln(k) \qquad (3)$$

$$S = \frac{1}{k}\sum_{j=1}^{k} S_j + \ln(k) \qquad (4)$$

Avec : $$S_j = -\sum_{i\in c_j} P_{ij}\,\ln(P_{ij}) \qquad (5)$$

S_j est l'entropie correspondant à la classe j. Le nombre optimal de classes k^* sera celui pour lequel l'entropie S est maximale.

3.2 Calcul des coefficients P_{ij}

Pour chaque classe c_j, nous favorisons les points i les plus proches de son centre g_j en introduisant une contrainte additionnelle qu'on cherchera à minimiser, définie par :

$$W = \sum_{j=1}^{k}\sum_{i\in c_j} P_{ij}\left\|x_i - g_j\right\|^2 \qquad (6)$$

où $\|\ \|^2$ est la distance euclidienne.
Nous cherchons ainsi à avoir une concentration la plus élevée possible autour du centre g_j de chaque classe c_j. Maximiser S et minimiser W revient à minimiser l'expression suivante :

$$T = W - S \qquad (7)$$

$$T = \frac{1}{k}\sum_{j=1}^{k}\sum_{i\in c_j} P_{ij}\,\ln\left(P_{ij}\right) - \ln(k) + \sum_{j=1}^{k}\sum_{i\in c_j} P_{ij}\left\|x_i - g_j\right\|^2 \qquad (8)$$

sous contrainte $\sum_{i\in c_j} P_{ij} = 1$; pour j=1..k

Le lagrangien de l'optimisation de la formule (8) sous les k contraintes est donné par :

$$L = \frac{1}{k}\sum_{j=1}^{k}\sum_{i\in c_j} P_{ij}\,\ln(P_{ij}) - \ln(k) + \sum_{j=1}^{k}\sum_{i\in c_j} P_{ij}\left\|x_i - g_j\right\|^2 + \sum_{j=1}^{k}\alpha_j\left(\sum_{i\in c_j} P_{ij} - 1\right) \qquad (9)$$

Où α_j est le multiplicateur de Lagrange associé à la j^{ème} contrainte. L'annulation de la dérivée de L par rapport à P_{ij} donne :

$$\frac{1}{k}\ln(P_{ij}) + \frac{1}{k} + \left\|x_i - g_j\right\|^2 + \alpha_j = 0 \qquad (10)$$

Les expressions des P_{ij}, pour i ∈ c_j et j = 1...k, sont déduites à partir de l'équation (10) par :

$$P_{ij} = \exp-\left(1 + k\alpha_j\right)\exp\left[-k\left\|x_i - g_j\right\|^2\right] \qquad (11)$$

Notons $Z_j = \exp\left(1 + k\alpha_j\right)$. Nous obtenons donc :

$$P_{ij} = Z_j^{-1}\exp\left[-k\left\|x_i - g_j\right\|^2\right] \qquad (12)$$

Tenant compte de la contrainte (1), Z_j est le coefficient de normalisation. En remplaçant P_{ij} par sa valeur dans (12), nous obtenons :

$$Z_j = \sum_{i \in c_j} \exp\left[-k\left\|x_i - g_j\right\|^2\right] \qquad (13)$$

Et par suite, à partir de (12), les coefficients P_{ij} sont donnés par :

$$P_{ij} = \frac{\exp\left[-k\left\|x_i - g_j\right\|^2\right]}{\sum_{i \in c_j}\exp\left[-k\left\|x_i - g_j\right\|^2\right]} \qquad (14)$$

3.3 Définition du nouvel indice de validité proposé : V_{MEP}

Finalement, notre indice V_{MEP} est défini comme une entropie par :

$$V_{MEP} = S = \frac{1}{k}\sum_{j=1}^{k} S_j + \ln(k)$$

Où S_j est défini par (7) qui utilise les Pij définis dans l'équation (14). Le nombre optimal k^* de clusters sera celui pour lequel la valeur de V_{MEP} est maximale.

4 Résultats expérimentaux

L'indice V_{SV} proposé par Do-Jong Kim et al [22] a été comparé dans plusieurs publications aux indices V_{PC}, V_{PE}, V_{FS}, V_{XB}, V_K et V_{SV} a montré une grande performance par rapport à tous les autres cités. Cet indice a été aussi utilisé avec succès dans un travail antécédent de l'un des auteurs [24] pour trouver le nombre optimal de clusters utilisant le modèle de mélange des gaussiennes (Gaussian Mixture Mode : GMM), et l'algorithme EM pour le processus de groupement, permettant d'extraire la forme des régions dans les images de textiles couleurs.

Par conséquent, nous comparerons notre nouvel indice V_{MEP} uniquement à V_{SV} sur des exemples de données artificiels et réels. Partant des boules polonaises [25] générées selon des distributions normales dont les paramètres sont rapportés dans la Table-1, nous avons générés 16 bases de données (BDi ; i=1..16) avec des degrés de chevauchement croissants entre les deux clusters 2 et 3 : pour la base de données BD1, les deux clusters 2 et 3 sont complètement distincts ; et pour BD16, ils sont pratiquement confondus. Etant donné le manque de place dans cet article, nous rapportons uniquement les figures au passage décisif. Ainsi, la figure-1 présente les 7 graphiques correspondant à BD1, BD2, BD5, BD6, BD7, BD13 et BD14. Dans BD1, on distingue 4 clusters compacts et bien séparés alignés sur la diagonale.

Nombre cluster	Nombre points	Moyennes	Covariances
Cluster 1	1000	(-4 ; -4)	$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$
Cluster 2	1000	(0 ; 0)	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Cluster 3	1000	(4 ; 4)	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Cluster 4	1000	(8 ; 8)	$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$

Table 1 : *Paramètres utilisés pour générer BD1*.

Les BD2, BD5, BD6, BD7, BD13 et BD14 sont obtenues à partir de BD1 en déplaçant le cluster 2 ayant pour centre (0,0) (Table-1), vers le cluster 3 de centre (4, 4). Les centres respectifs du cluster 2 dans ces bases de données sont (1, 1), (1.7, 1.7), (1.8, 1.8), (2.0 ; 2.0), (3.6, 3.6), (3.7, 3.7).

Nous appliquons alors les indices V_{SV} et V_{MEP} sur l'ensemble de ces données.

Ces deux indices donnent le même résultat optimal, à savoir 4 clusters, tant qu'il s'agit d'un chevauchement léger. C'est le cas pour les trois premiers graphiques de la figure-1, dont les coordonnées des centres respectifs du cluster 2 sont (0 ; 0), (1 ; 1) et (1.7, 1.7).

A partir de BD6, dont le centre du cluster 2 est (1.8; 1.8), qui est relativement plus proche du cluster 3, c'est-à-dire présentant un chevauchement légèrement plus élevé, V_{sv} n'arrive plus à détecter le nombre de clusters corrects 4. Ceci est confirmé pour toutes les autres bases de données avec un chevauchement encore plus fort, notamment BDi, i=7…16.

Tandis que pour V_{MEP}, la limite de la bonne détection du nombre correct de clusters continue jusqu'à un très fort degré de chevauchement BDi, i=7…13 et illustré par les deux graphiques décisifs correspondant à BD7 et BD13 de la figure 1.

Pour BD14, BD15 et BD16, le nombre optimal de clusters déterminé par V_{MEP} est 3 comme illustré par le graphique

correspondant à BD14 de la figure-1. En effet, les deux clusters 2 et 3 sont pratiquement confondus.

La performance de V_{MEP} est montrée aussi par l'application aux données réelles Iris [26]. L'ensemble des données formé de 150 points répartis en 3 clusters de 50 points, nommés respectivement : Setosa, Versicolor, et Verginica. La plupart des récents indices cités auparavant n'arrivent pas à détecter le nombre réel de clusters des IRIS. Plus récemment, en 2004, Dae-Won Kim et al [23] a proposé un autre indice V_{OS} qui utilise le concept du degré de chevauchement et séparation. Cependant, il reste incapable de détecter le nombre réel de clusters en cas d'un chevauchement important, et comme mentionné par les auteurs dans [23], dans le cas des Iris, le nombre optimal de clusters qu'il détecte est 2, ce qui est un résultat faux. Dans la figure 2, nous présentons les résultats trouvés en utilisant V_{SV} et V_{MEP} sur les Iris. Les deux indices déterminent le nombre optimal correct de clusters qui est 3. Ici, V_{SV} fonctionne bien car il y a un faible degré de chevauchement.

Les bases de données générées artificiellement avec des degrés de chevauchement croissants (BDi, i=1…16), ainsi que les données réelles des Iris, nous ont permis de mettre en évidence les limites de performances des deux indices V_{SV} et V_{MEP}. La supériorité de V_{MEP} à V_{SV}, et par conséquent à tous les autres indices cités auparavant, est ainsi bien établie.

5 Conclusion

Dans ce papier, nous avons proposé un nouvel indice pour l'évaluation de la qualité des résultats d'un algorithme de partitionnement. L'indice proposé, noté V_{MEP}, est basé sur le principe du maximum d'entropie, et ne nécessite aucun paramètre. Le nombre optimal de clusters correspond au nombre k^* pour lequel l'indice V_{MEP} est maximal. La performance de notre nouvel indice est établie sur des exemples artificiels et réels. V_{MEP} peut détecter le nombre optimal correct de clusters même avec un grand degré de chevauchement. Il peut être très utile dans les applications réelles en médecine, biologie, imagerie médicale, etc. où c'est important de connaître le nombre réel de clusters.

Les résultats trouvés montrent la supériorité de notre indice V_{MEP} sur les autres.

Notons que, comme tous les autres indices, V_{MEP} dépend aussi des résultats obtenus par l'algorithme FCM. Si celui-ci converge vers un minimum local, l'évaluation des indices de validités est inutile.

Nous finirons par signaler un autre avantage de notre nouvel indice V_{MEP} : il ne dépendant d'aucun paramètre produit par l'algorithme de classification utilisé ; de ce fait, il reste indépendant de l'algorithme de classification. Ceci nous donne la liberté de choisir celui qui semble le plus adapté pour l'application considérée ; comme l'algorithme Gustafson-Kessel (GK) adapté pour les clusters de formes ellipsoïdales, ou encore l'algorithme EM. Ce sera l'objet d'un futur travail.

Références

[1] K. Jain, M. N. Murty and P. J. Flynn: Data clustering: a review, ACM Computing Surveys, Vol. 31, No. 3, pp. 264-323, 1999.

[2] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice Hall, New Jersey, 1988.

[3] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.

[4] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum, 198.

[5] J. Hartigan, Clustering Algorithms. New York: Wiley, 1975.

[6] J. Tou and R. Gonzalez, Pattern Recognition Principles. Reading, MA: Addison-Wesley, 1974.

[7] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, Fuzzy Cluster Analysis-Methods for Classification, Data Analysis and Image Recognition. John Wiley & Sons, LTD, 1999.

[8] S. Theodoridis and K. Koutroubas: Pattern Recognition, Academic Press, 1999

[9] Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis Cluster Validity Methods : Part I

[10] Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis Clustering Validity Checking Methods: Part II

[11] J. C. Dunn: Well Separated Clusters and Optimal Fuzzy Partitions, Journal of Cybernetica, Vol. 4, pp. 95-104, 1974

[12] D. L. Davies and D. W. Bouldin: Cluster Separation Measure, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1, No. 2, pp. 95-104, 1979

[13] Subhash Sharma: Applied multivariate techniques, John Wiley & Sons, Inc., 1996

[14] M. Halkidi, Y. Batistakis and M. Vazirgiannis: On Clustering Validation Techniques, Journal of Intelligent Information Systems, Vol. 17, No. 2-3, pp. 107-145, 2001

[15] Maria Halkidi and Michalis Vazirgiannis: Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set, Proc. of ICDM 2001, pp. 187-194, 2001

[16] M. Halkidi and M. Vazirgiannis and Y. Batistakis: Quality Scheme Assessment in the Clustering Process, Proc. of the 4 th European Conference on Principles of Data Mining and Knowledge Discovery, pp. 265-276, 2000

[17] Bezdek,J.C.,1974.Numerical taxonomy with fuzzy sets. J. Math. Biology 1,57 –71.

[18] Bezdek,J.C.,1974.Cluster validity with fuzzy sets J. Cybernit.3,58 –72.

[19] Y.Fukuyama, M.Sugeno, A new methodof choosing the number of clusters for the fuzzy c-means method,in: Proceedings of the Fifth Fuzzy Systems Symposium,1989,pp. 247 –250.

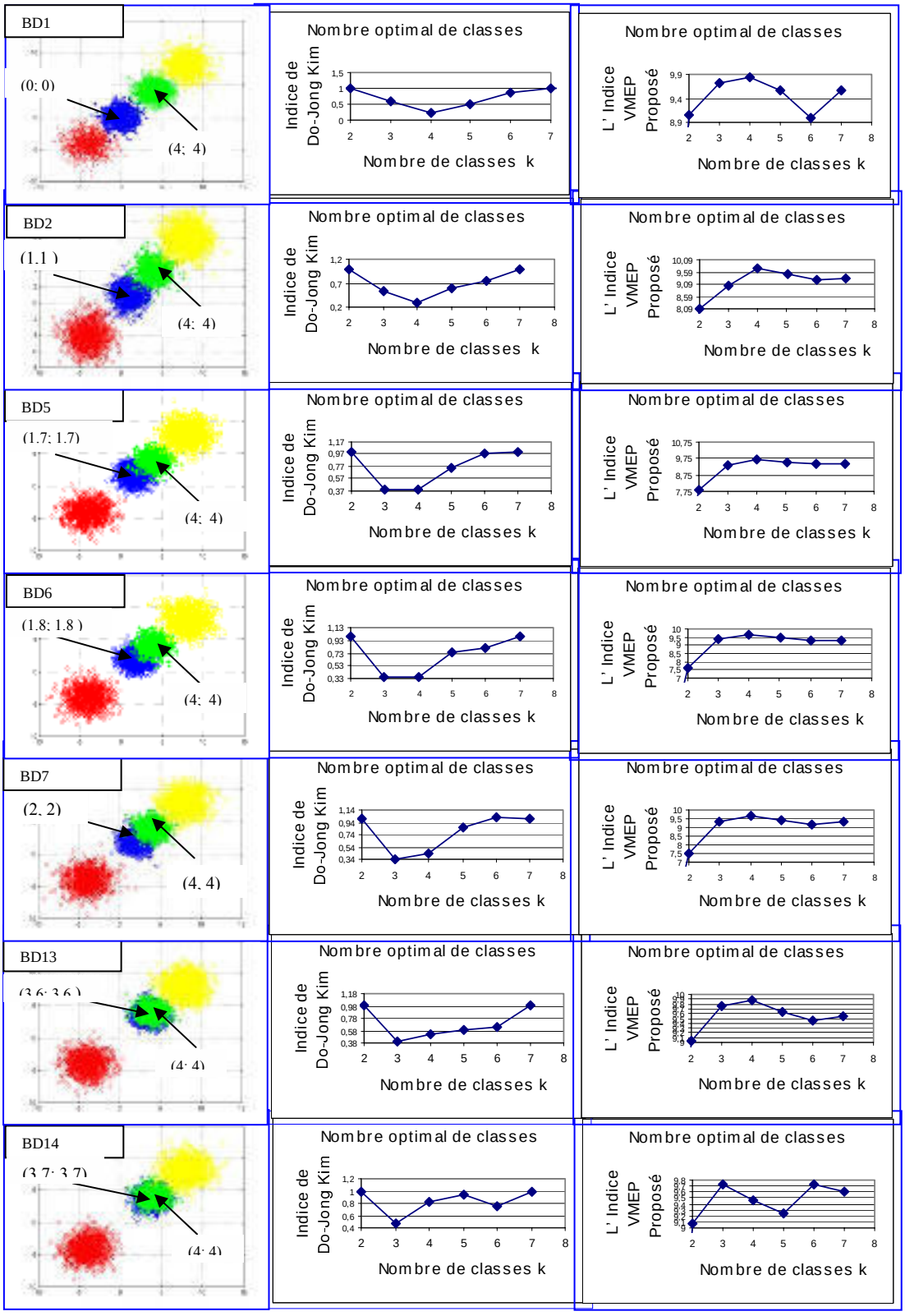


Figure1 : *Indice de Do-Jong Kim's V_{SV} (valeur minimale) et l'indice proposé V_{MEP} (valeur maximale), affichés respectivement pour BD1, BD2, BD5, BD6, BD7, BD13, et BD14.*

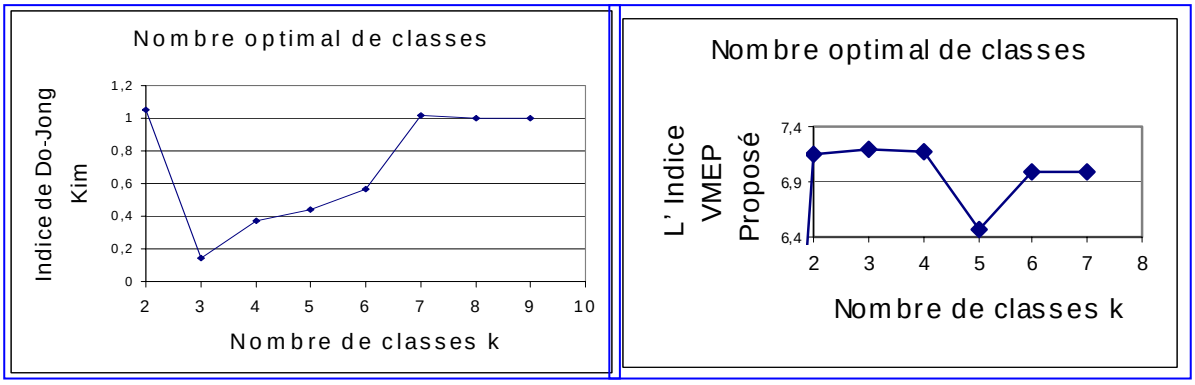


Figure2 *: Indice de Do-Jong Kim's V_{SV} (valeur minimale) et l'indice proposé V_{MEP} (valeur maximal) affichés respectivement pour les données Iris.*

[20] X.L.Xie,G.Beni, A validity measure for fuzzy clustering, IEEE Trans.Pattern Anal.Mach.Intell.13(8)(1991)841–847.

[21] S.H.Kwon, Cluster validity index for fuzzy clustering, Electron.Lett.34(22) (1998) 2176–2177

[22] D.J.Kim, Y.W.Park, and D.J.Park, A novel validity index for determination of the optimal number of clusters,IEICE Trans. Inform.Syst.D-E84(2)(2001)281 –285.

[23] Dae-Won Kim a ,Kwang H.Lee ,and Doheon Lee On cluster validity index for estimation of the optimal number of fuzzy clusters. Pattern Recognition. Vol 37. pp.2009 –2025. (2004)

[24] A. Lachkar, R. Benslimane, L. D'Orazio, E. Martuscelli,. A system for textile design patterns retrieval part 1: Design patterns extraction by adaptive and efficient colour image segmentation method. To appear in The Journal of the Textile Institute. Ref.: Ms. No. 10.1533.joti.2005.124R1

[25] Cembrzynski, T. Banc d'essai sur "les boules polonaises", des trois criteres de decision utilises dans la procedure de classification MNDOPT pour choisir un nombre de classes. RR-0784 Rapport de recherche de l'INRIA.

[26] Anderson E. The IRISes of the Gaspe peninsula. Bull Am IRIS Soc 1935;59:2– 5.

Organisé par

Avec le soutien de