

Suivi de mouvements dans des séquences vidéo : Une approche basée sur la synthèse d'images

Yannick Perret

Saida Bouakaz

Thierry Excoffier

Laboratoire d'Informatique Graphique Image et Modélisation (LIGIM)
UFR Informatique, Université Claude Bernard - Lyon1

43, Bd du 11 Novembre 69622 - Villeurbanne
yperret—bouakaz—exco@ligim.univ-lyon1.fr

Résumé

Notre objectif est la restitution du mouvement d'un objet qui se déplace dans un environnement 3D, à partir d'enregistrements de séquences vidéo. L'objet est supposé "synthétisable", rigide mais peut être articulé. Des caméras calibrées filment le sujet en mouvement et fournissent à chaque instant t un ensemble de vues de l'objet. On dispose ainsi au cours du temps, d'une part des séquences vidéo, une par caméra, et d'autre part d'un modèle géométrique paramétré du sujet d'intérêt, permettant d'en générer des images de synthèse. Ce modèle est une représentation générique qui inclut la description spatiale du sujet ainsi que les couleurs, les textures... Les attributs variables du modèle seront appelés paramètres. Ils comprennent selon les cas la position, l'orientation, les angles des articulations, les mesures anthropomorphiques, etc. On dispose également de la possibilité de générer des images de synthèse du sujet pour une vue et un jeu de paramètres donnés. Dans ce cas, le suivi du mouvement du sujet peut être ramené à la recherche des paramètres qui génèrent les images de synthèse les plus fidèles visuellement aux images réelles. Notre démarche s'appuie sur une notion de mesure d'adéquation entre images, et le résultat désiré correspond aux images et donc aux paramètres qui optimisent cette mesure. Ainsi notre problème se pose en terme d'optimisation dans R^n . Les résultats sont présentés aussi bien sur des scènes synthétiques qui dans notre cas constituent des réalités terrain que sur des cas réels qui montrent en particulier la résistance de notre méthodes aux occultations.

Mots Clef

Suivi de paramètres, séquences vidéo, comparaisons denses, méthode du simplex.

1 Introduction

Le développement de nouveaux domaines de communication et d'applications de l'image, telles que

la vidéo conférence, la réalité virtuelle ou la réalité augmentée pour ne parler que des domaines les plus actuels nécessitent l'exploration et la maîtrise de nouveaux outils pour mieux appréhender l'image. Dans ce contexte l'analyse et la restitution du mouvement d'objets extrait de scènes visuelles réelles est l'une des étapes clé. Dans la littérature la notion de suivi de mouvement peut correspondre, selon les applications visées, à divers niveaux de précision allant de la simple détection de mouvement dans une scène [8] à la "capture" tridimensionnelle du mouvement de l'objet d'intérêt et son transfert à un mannequin, en passant par le suivi de cibles. Ainsi pour ces applications on trouve la "familles" des méthodes basées sur le flux optiques [2] qui ont suscité un grand intérêt dans la communautés de la vision. L'une des approches utilisée s'appuie sur la détection et le suivi de zones d'intérêt dans les images [12], tels que des points particuliers, des contours[1] ou des régions. Ce type d'approche nécessite de connaître les zones d'intérêt attendues, ou du moins leurs caractéristiques dans les images et sont sensibles aux occultations et aux erreurs d'extraction. Une autre technique consiste à mettre en divers endroits du sujet des éléments particuliers, tels que des patchs colorés, des boules ou encore des textures spécifiques. Dans [4] ce sont des courbes, tracées sur des visages, qui permettent de suivre les attitudes faciales d'un sujet. Ces éléments sont associés à des zones caractéristiques du sujet, et servent de guides pour le suivi tout au long des mouvements. Cette technique donne de bons résultats dans un environnement contrôlé, mais elle présente des contraintes quant au choix et la pose pose des problèmes dès lors que le milieu rend l'utilisation de ces éléments difficile. On peut citer aussi les méthodes basées sur le flux optiques [2] qui ont suscité un grand intérêt cette technique est efficace pour le suivi de cible.

Certains chercheurs se sont orientés vers une collaboration entre analyse et synthèse d'images [13, 5]. Les images de synthèse des objets étudiés permettent de

guider ou de valider la recherche des caractéristiques spatiales des sujets. Dans [3], cette approche est utilisée pour suivre des déformations 2D d'objets. D'une manière générale, ces approches nécessitent de connaître à chaque instant les valeurs des degrés de liberté de l'objet suivi, que nous appelons *paramètres*. Nous décrivons dans cet article une méthode de suivi de paramètres 3D d'objets réels. La section 2 présente brièvement les travaux existants dans ce domaine. La section suivante détaille notre approche. Cette approche nécessite la définition d'une mesure d'adéquation entre images, ainsi que l'optimisation de cette mesure par rapport aux paramètres. Dans cette section est également présenté l'auto-raffinement de modèle ainsi que l'extraction automatique des textures des objets. Divers résultats obtenus sur des séquences vidéo réelles sont présentés et discutés dans la section 4.

2 Notre approche

Nous présentons dans cette section l'approche que nous avons développé pour le suivi de paramètres. Plusieurs flux vidéo sont disponibles, provenant chacun d'une caméra calibré filmant le sujet. Le suivi de paramètres est défini comme étant la recherche des paramètres qui génèrent les vues de synthèse les plus fidèles par rapport aux images vidéo. On dispose pour cela d'un modèle paramétré du sujet suivi. Ce modèle comprend une description géométrique du sujet (tailles, positions, couleurs...) ainsi que la spécification des degrés de liberté tels que la position, l'orientation et les angles des articulations. Ce modèle est utilisé pour générer des images de synthèse du sujet correspondant aux positions des caméras, ceci pour n'importe quelles valeurs des paramètres.

Notre approche se base sur un principe de prédiction / vérification / correction. Un jeu de paramètres initial est proposé, et son adéquation est mesuré en comparant les images de synthèse du sujet avec les vues réelles. Le jeu de paramètres est corrigé en fonction de ce résultat et le processus est ré-itéré jusqu'à convergence (voir Fig. 1).

Ceci correspond à la recherche des valeurs de chaque paramètre pour un jeu de vues réelles donnés, c'est-à-dire pour un instant t particulier. Pour passer d'un instant t à un instant $t+1$, nous utilisons l'hypothèse que les mouvements apparents dans les images sont faibles entre deux instants successifs, et donc que les variations des valeurs des paramètres le sont aussi. Ainsi les valeurs initiales des paramètres à un instant $t+1$ seront celles trouvées par le système à l'instant t . Si de plus on possède un modèle d'évolution des paramètres (même simple comme une interpolation), il est possible de "corriger" ces valeurs initiales afin d'être plus proche des valeurs réelles (Fig. 2).

Le problème de l'obtention de ces valeurs au début des

séquences est abordé dans la section 3.3.

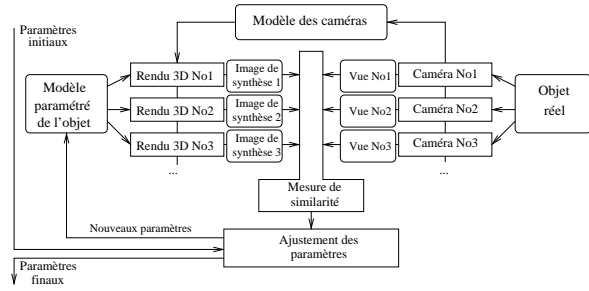


Figure 1: Description de l'approche utilisée

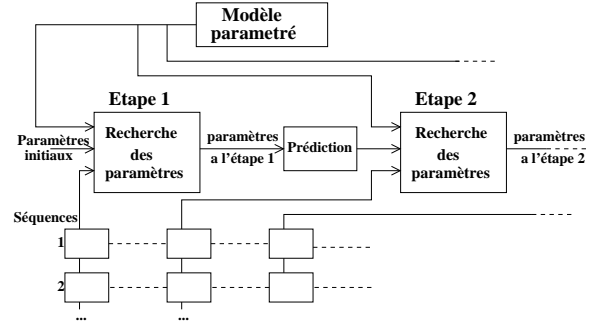


Figure 2: Extension à l'ensemble des séquences, et prédiction

2.1 Mesure d'adéquation

Pour quantifier l'adéquation entre images de synthèse et images réelles nous introduisons une mesure d'adéquation entre images. La notion de similarité dépend des critères utilisés pour la définir. Des critères humain[11] de haut niveau permettent de reconnaître ou de classifier, mais sont moins utiles pour des calculs précis d'écarts. Dans notre cas nous avons des instances similaires d'un même objet à comparer. Il faut cependant tenir compte de plusieurs points :

- le sujet peut être partiellement caché dans les vues réelles par des objets non décrits, et donc non présents dans les images de synthèse ;
- les images de synthèse ne sont pas parfaitement fidèles à la réalité, à cause des erreurs de modélisation, des propriétés des surfaces...

Considérons une image comme une matrice de pixels, chaque pixel étant référencé par ses coordonnées (x, y) et un vecteur de couleur. Il existe de nombreux espaces de représentation des couleurs (voir [10] pour plus de détails) et chacun à des propriétés particulières. Pour plusieurs raisons décrites plus loin, nous avons choisi d'exprimer les couleurs dans un espace séparant la luminance de la chrominance. Notre choix s'est arrêté sur l'espace $YCbCr$, qui possède les caractéristiques désirés.

La comparaison entre deux images I_1 et I_2 est réalisée en effectuant une comparaison dense (aussi appelé "image registration"). Le principe est de calculer la distance entre les pixels (x, y) des deux images. La valeur moyenne D de ces distances est alors retenue :

$$D_{image}(I_1, I_2) = \frac{1}{H * W} \sum_{x, y} d_{pix}(I_1[x, y], I_2[x, y]) \quad (1)$$

avec H et W respectivement la hauteur et la largeur des images, et d_{pix} une distance entre pixels, décrite plus loin. Cette distance est robuste par rapport aux occlusions, et ne nécessite aucun traitement de haut niveau sur les images.

La distance entre pixels est définie en fonction du fait que les images à comparer ne sont pas identiques. Les erreurs de modélisation, d'attribut de surface ou d'éclairage peuvent générer des variations au niveau des pixels. De plus des objets non modélisés dans la scène peuvent projeter des ombres sur le sujet suivi, sans que celles-ci n'apparaissent dans les images de synthèse.

Pour des éclairages mono-chromatiques, les ombres et les erreurs de positionnement des éclairages génèrent principalement des variations de luminance sur les pixels, laissant la chrominance globalement inchangée. C'est pourquoi nous avons séparé le calcul de la distance entre pixels en deux sous-distances, une dans l'espace des luminances et une autre dans l'espace des chrominances :

$$p_1 \rightarrow Y_1, C_{r1}, C_{b1} \quad (2)$$

$$p_2 \rightarrow Y_2, C_{r2}, C_{b2} \quad (3)$$

$$d_{lum}(p_1, p_2) = \frac{|Y_1 - Y_2|}{2} \quad (4)$$

$$d_{chrom}(p_1, p_2) = \sqrt{(C_{r1} - C_{r2})^2 + (C_{b1} - C_{b2})^2} \quad (5)$$

La distance finale est une combinaison linéaire des ces deux sous distances :

$$d_{pix}(p_1, p_2) = \alpha * d_{chrom}(p_1, p_2) + (1 - \alpha) * d_{lum}(p_1, p_2) \quad (6)$$

avec α contrôlant le poids relatif des termes. Dans notre cas, cet α doit être supérieur à $\frac{1}{2}$ afin d'accorder plus d'importance à l'information la plus stable : la chrominance.

Pour le calcul de l'adéquation d'un jeu de paramètres donné p , les images de synthèse du modèle sont générées pour chaque vue existante, en utilisant ces valeurs des paramètres. La distance entre image D_{image} est appliquée à chacune des paires d'images réelles / de synthèse. Appelons R_i une image réelle et $S_i(p)$ l'image de synthèse du modèle pour les paramètres p , pour une vue donnée i . La fonction d'adéquation D s'écrit alors :

$$D(p) = \sum_i \alpha_i D_{image}(R_i, S_i(p)) + B(p), \sum \alpha_i = 1 \quad (7)$$

où les α_i contrôlent le poids relatif des différentes vues entre elles. Les α_i sont choisis en fonction de la résolution des images, de leur qualité ou de la visibilité du sujet. Le terme $B(p)$ est un terme de pénalisation dont la valeur augmente lorsque les paramètres sont à l'extérieur de leur domaine de définition. Il permet de restreindre l'exploration de l'espace des paramètres aux zones valides.

2.2 Optimisation des paramètres

Dans la section précédente nous avons vu le calcul de l'adéquation d'un jeu de paramètres aux images de référence. Nous avons besoin de trouver le jeu de paramètres qui génère les vues de synthèse les plus proches des images vidéo, c'est-à-dire celui qui minimise la mesure d'adéquation D .

Nous avons adapté une méthode robuste, basé sur le principe du simplex (voir [6] pour plus de détails) à nos besoins. Ce type d'approche présente plusieurs avantages :

- robustesse partielle aux minima locaux ;
- convergence rapide et fiable, même avec un grand nombre de paramètres ;
- absence d'utilisation des dérivés partielles de la fonction à optimiser.

Cette méthode donne de bons résultats et présente une grande robustesse, en particulier aux petites perturbations de la fonction de dissimilarité. Cependant, à l'approche de la solution, le simplex tend à se "contracter" excessivement, ralentissant la convergence vers la solution. Nous résolvons ce problème en introduisant de petites perturbations aléatoires dans les valeurs des paramètres. Ce principe du recuit simulé ou des mutations dans les algorithmes génétiques permet de limiter le ralentissement de la convergence et d'obtenir une meilleure précision.

2.3 Initialisation du système

Comme beaucoup d'applications de suivi, une étape d'initialisation est nécessaire à notre système, ceci afin d'obtenir un jeu de valeurs initiales pour les paramètres. Ces valeurs correspondent à l'état de l'objet suivi au début des séquences vidéo. Ceux-ci peuvent être directement fournis par l'utilisateur.

Cependant, afin de rendre notre système plus autonome, nous avons développé une méthode d'aide à l'initialisation. Les modèles sont définis via un langage descriptif 3D qui comprend des descripteurs géométriques et la spécification des paramètres. Nous avons adjoint à cela des **points d'ajustement**. Ces points 3D sont associés à des parties du modèle. Le système de rendu de synthèse est capable de générer les coordonnées 2D de ces points projetés dans les différentes vues. Ces projections dépendent bien entendu des valeurs des paramètres, ceux-ci contrôlant

l'état spatial du modèle. Nous avons défini une nouvelle mesure de dissimilarité, basé sur des distances entre points 2D. L'utilisateur place tout d'abord sur le modèle des points d'ajustements p_j . Il indique ensuite dans chaque vue réelle i les coordonnées u_{ij} correspondantes aux points d'ajustement. Les valeurs des paramètres minimisant la distance d entre les points u_{ij} et la projection des points d'ajustement du modèle sont retenus comme solution initiale du système :

$$d(v) = \sum_{ij} d(u_{ij}, p'_{ij}(v)) \quad (8)$$

Ici, p'_{ij} correspond à la projection du point d'ajustement p_j dans la vue i , pour un jeu de paramètres v . Toutes les expériences décrites plus loin ont été initialisées grâce à cette technique.

Nous travaillons actuellement sur une seconde approche d'initialisation. Elle est basée sur une exploration stochastique de l'espace des paramètres couplée à l'utilisation de la multi-résolution. Une exploration large des paramètres possibles à faible résolution permet de dégager des solutions potentiellement intéressantes. Ces pré-solutions sont ensuite explorées plus finement à des résolutions plus élevées. Cette approche, ne nécessitant que peu d'interventions, permettrait d'augmenter l'automatisation de notre système.

2.4 Ajustement de modèle et extraction de textures

Réaliser un modèle géométrique d'un objet réel n'est pas forcément une chose aisée. Cela se réalise facilement lorsque l'on possède un plan de l'objet, comme pour de nombreux objets manufacturés, mais dans les autres cas il faut mesurer entièrement l'objet étudié.

Nous avons simplifié le problème en permettant l'utilisation de modèles génériques, plus simples à définir. Dans notre système, les paramètres ne sont pas limités aux positions, orientations et états des articulations. Il est possible de définir comme paramètre n'importe quelle caractéristique interne d'un objet, telles que les tailles, diamètres ou bien couleurs. Nous appelons paramètres externes les paramètres liés à l'état spatial du sujet étudié, et paramètres internes les caractéristiques du modèle générique susceptibles de varier entre deux objets réels différents.

Pour affiner un modèle générique par rapport à un objet donné, les paramètres extérieurs sont tout d'abord calculés par la méthode usuelle. Ces paramètres sont peu précis, car calculés en utilisant un modèle non ajusté. Les paramètres internes sont ensuite calculés de la même façon. Ces deux étapes sont ré-itérées jusqu'à stabilisation des résultats. Ces paramètres internes définissent alors le modèle ajusté de l'objet étudié.

A partir du modèle et de ses paramètres ajustés, il est possible de connaître la projection dans chaque vue du

sujet suivi. Ceci est utilisé pour extraire dans chaque vue les textures visibles des surface de l'objet réel. Ces textures partielles issues des différentes vues sont ensuite combinées[7] en fonction d'un degré de confiance associé aux pixels extraits. Ceci permet d'améliorer l'aspect visuel des modèles générés en utilisant directement leur apparence telle qu'elle apparaît dans les images vidéo.

Un exemple détaillé de cette technique est décrit à la section 4.

3 Applications et discussions

Nous présentons dans cette section quelques résultats obtenus sur diverses séquences vidéo. Un premier ensemble d'exemples, réalisés sur des séquences vidéo de synthèse, est présenté dans [9]. L'utilisation de séquences de synthèse offre l'avantage de connaître les valeurs réelles des paramètres, permettant ainsi de mesurer les écarts entre les résultats trouvés et les valeurs théoriques. Le second jeu d'exemple, présenté ici, se compose de séquences vidéo en couleur d'une résolution de 320. Dans chaque cas l'objet suivi a été filmé par deux caméras, de type "webcam". Comme les valeurs réelles des paramètres ne sont pas connues pour des objets réels, il n'est pas possible de mesurer directement la validité des résultats. Pour valider nos tests, nous utilisons les paramètres trouvés par notre système pour générer des vues de synthèse du sujet suivi. Chaque vue correspond à une caméra réelle, et il est alors possible de comparer visuellement l'adéquation par rapport à l'objet de référence. Dans les exemples qui suivent, nous avons superposé images de synthèse et images réelles pour permettre une vérification aisée des résultats.

Exemple simple

Dans cette séquence, une boîte de "Scotch" se déplace sur une table, filmée image par image par deux caméras. La figure 3 montre quelques extraits de ces séquences pour l'une des caméras. Nous utilisons un modèle géométrique texturé généré par notre méthode, dont l'exemple complet se trouve plus loin. La figure 4 montre les résultats obtenus. Une version en fil-de-fer de la boîte est superposée aux images originales en utilisant les paramètres spatiaux trouvés.

Suivi de livre

Dans cet exemple, deux caméras filment un livre en mouvement, avec une main passant devant celui-ci vers la fin de la séquence (voir la figure 5). Le modèle texturé du livre a été obtenu par la méthode habituelle. Toujours sur la figure 5 on montre la superposition du modèle du livre en fil-de-fer sur les images réelles. On constate que notre système suit le livre dans ses déplacements, y compris pendant l'occlusion.

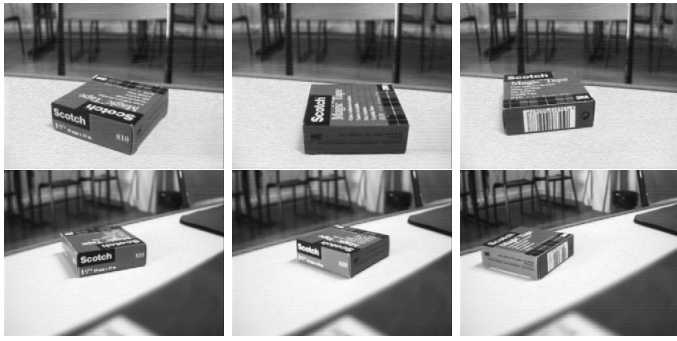


Figure 3: Extraits des deux vues au début, au milieu et à la fin de la séquence

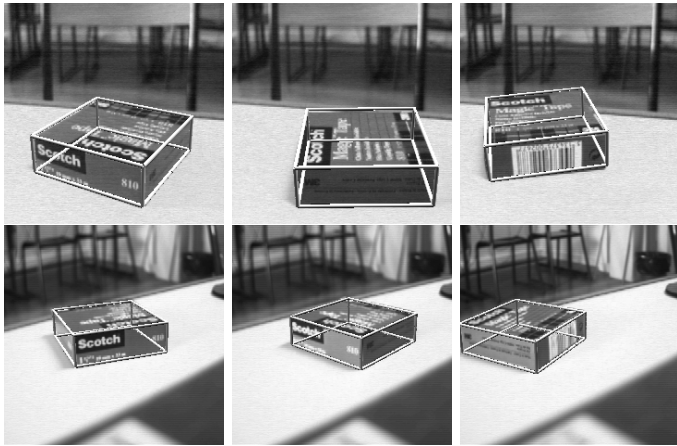


Figure 4: Superposition de la boîte de synthèse sur les images réelles.

Objet articulé

Dans cette séquence un objet composé de deux pièces articulées entre elles est filmé par deux caméras. L'objet se déplace, et l'angle entre les deux parties se modifie (voir la première ligne de la figure 6). La deuxième ligne de cette figure montre la superposition du modèle (en blanc) sur les images originales, en utilisant les paramètres trouvés.

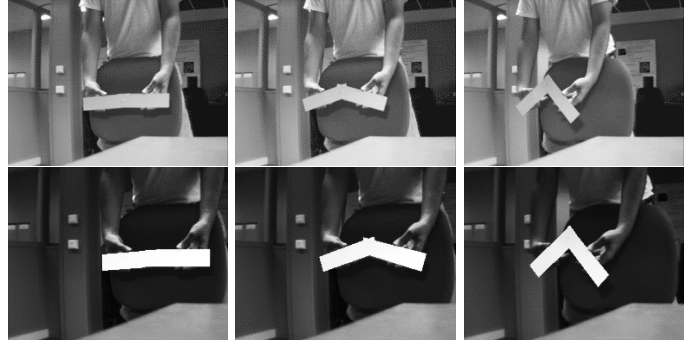


Figure 6: Première ligne: Images de l'objet articulé pour l'une des caméras. Deuxième ligne: superposition du modèle trouvé aux images réelles.

Ajustement de modèles et extraction de textures

Un point important abordé auparavant est la possibilité de définir librement les paramètres. En particulier il est possible de spécifier des caractéristiques internes des objets, telles que les dimensions ou les couleurs. Ceci permet en particulier d'utiliser comme paramètres des caractéristiques internes du modèle afin d'affiner celui-ci par rapport aux images réelles. La figure 3 montre la séquence de la boîte de "Scotch". Pour créer son modèle, nous commençons tout d'abord par fabriquer un modèle *a priori*, à partir de ses dimensions approximatives. Nous recherchons un jeu de paramètres initial par la méthode vue plus haut. Ensuite les traitements suivants sont appliqués :

- affinage de la position/orientation du modèle par le programme ;
- affinage des caractéristiques internes par le programme.

Ces deux étapes sont itérées jusqu'à stabilisation des résultats. Ceci nous donne le modèle géométrique ajusté, ainsi que les paramètres initiaux au début de la séquence (voir Fig.7).

Ensuite l'extraction automatique des textures collecte les parties visibles de l'objet dans les différentes vues. Ces éléments sont combinés pour former les textures de l'objet (voir Fig. 9).

Finalement, ce modèle ajusté et texturé est utilisé



Figure 5: Colonnes 1 et 3 : Extraits de la séquence pour l'une des caméras. Colonnes 2 et 4 : superposition du résultat sur les images originales.

pour effectuer le suivi de la boîte dans les séquences.

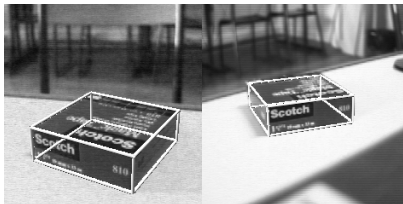


Figure 7: Superposition de la boîte trouvée sur les images originales. Le rendu "fil-de-fer" permet une meilleure visibilité.

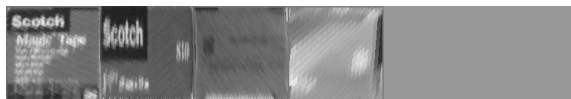


Figure 8: Textures extraites par le programme à partir des vues réelles, pour chaque face. Les dernières sont unies (originellement vertes) car elles correspondent à des faces non visibles dans les images. La couleur par défaut est alors validée.

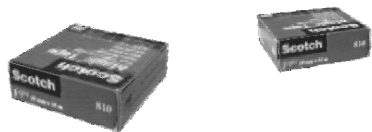


Figure 9: La boîte de synthèse, rendue en utilisant les textures trouvées.

4 Conclusion

Dans cet article nous avons présenté une approche pour le suivi de paramètres d'objets dans des séquences vidéo. Cette approche utilise une collaboration entre analyse et synthèse d'images, ainsi que des comparaisons denses entre images, permettant une forte résistance aux occlusions. De plus la définition des paramètres est souple et permet de spécifier des caractéristiques internes des modèles. Ceci est utilisé pour ajuster automatiquement un modèle générique, facilitant la création d'un modèle fidèle à l'objet étudié. L'extraction automatique des textures est un autre point important. La projection du modèle utilisée dans les différentes vues sert à extraire les textures des objets réels. Ceci permet d'améliorer l'apparence des modèles, ou bien pour étudier les surfaces des objets.

Nous testons actuellement notre méthode avec des modèles plus complexes. L'un d'eux consiste en du suivi de visage, dont un modèle auto-texturé est

présenté à la figure 11. Nous préparons également des essais de suivi de main. Une main (Fig. 10) correspond à une trentaine de paramètres (angles et position), ainsi qu'à une cinquantaine de paramètres internes (tailles, diamètres...).



Figure 10: Image réelle d'une main et le modèle ajusté correspondant.

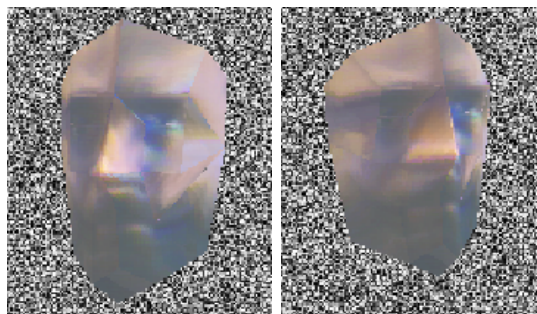


Figure 11: Deux vues d'un modèle ajusté et auto-texturé d'un visage.

A l'heure actuelle, notre système ne permet pas de faire du temps réel. Par exemple la séquence de la boîte **Scotch** prend 40 secondes par images sur un PC de capacité moyenne (PIII-500MHz). Pour améliorer cela, nous travaillons sur une approche hiérarchique. Cette hiérarchie concernent aussi bien la multi-résolution que la dépendances des paramètres. En termes de multi-résolution, les calculs commencent par les images à basse résolution, donnant des résultats peu précis mais rapides. La résolution des images est progressivement augmentée à l'approche de la solution, permettant de raffiner les calculs.

Concerant la corrélations des paramètres, le but est de déterminer les groupes de paramètres indépendants pour améliorer les traitements, ainsi que pour déterminer et exploiter les hiérarchies entre paramètres par rapport à la fonction de similarité.

Une autre possibilité est la parallélisation. Plusieurs parties de notre système telles que la synthèse ou la comparaison entre images qui sont couteuses en temps, se prêtent bien à la parallélisation.

References

- [1] R. Basri and D. Weinshall. Distance metric between 3d models and 2d images for recognition and classification. *IEEE Transactions on Pattern*

Analysis and Machine Intelligence 18(4), pages 465–470, 1996.

- [2] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *University of Pennsylvania technical report MS-CIS-97-23 (to appear in IJCV 2000)*.
- [3] J.Isodoro and S.Sclaroff. Active voodoo dolls: A vision based input device for nonrigid control. *Proc. Computer Animation*, June 1998.
- [4] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [5] M.Yasushi and S.Yoshiaki. Tracking moving object in 3-d space based on optical flow and edges. In IEEE Computer Society, editor, *Proc. International Conference on Pattern Recognition 14(2)*, pages 1439–1441, August 1998.
- [6] J.A. Nedler and R. Mead. A simplex method for function minimisation. *The Computer Journal* 7, pages 308–313, July 1965.
- [7] E. Ofek, E. Shilat, A. Rappoport, and M. Werman. Multiresolution textures from image sequences. *IEEE Computer Graphics and Applications 17(2)*, pages 18–29, March 1997.
- [8] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.
- [9] Y. Perret, T. Excoffier, and S. Bouakaz. Parameters matching of objects in video sequences. In *Proceedings of SPIE, Three-Dimensional Image Capture and Applications*, 3958, January 2000.
- [10] William K. Pratt. *Digital Image Processing*, chapter 2 and 3, page 21 to 89. Wiley Interscience, 2nd edition, 1991.
- [11] S. Santini and R. Jain. Similarity matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996.
- [12] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. *International Conference on Computer Vision, Bombay, India*, 1988.
- [13] S.Yonemoto, N.Tsuruta, and R.Taniguchi. Tracking of 3d multi-part objects using multiple view-point time-varying sequences. In IEEE Computer Society, editor, *Proc. International Conference on Pattern Recognition 14(1)*, pages 490–494, August 1998.