

# Schéma de lifting sur graphes et application à l'approximation d'images

M. Hidane, O. Lézoray et A. Elmoataz

Université de Caen Basse-Normandie, ENSICAEN, GREYC UMR CNRS 6072, Caen, France

{moncef.hidane, olivier.lezoray, abderrahim.elmoataz}@unicaen.fr

## Résumé

Nous proposons une transformée multi-échelle s'appliquant à des fonctions scalaires définies sur l'ensemble des sommets d'un graphe non orienté pondéré général. La transformée que nous proposons est basée sur une adaptation du schéma de lifting. Une des difficultés de cette adaptation concerne le partitionnement des sommets du graphe. Nous adoptons une approche gloutonne récemment proposée et l'étendons à une transformée multi-échelle en l'itérant sur des sous-graphes induits. Nous examinons également l'effet de chaque étape sur la base sous-jacente. Nous nous intéressons enfin à l'application de notre algorithme à l'approximation d'images numériques et comparons les erreurs d'approximation non linéaire à  $M$  termes que nous obtenons avec celles obtenues avec d'autres transformées.

## Mots clefs

Graphes pondérés, transformées multi-échelles, approximation, signaux sur graphes, lifting.

## 1 Introduction

Cet article traite de la représentation, à partir de transformées multi-échelles, de fonctions définies sur l'ensemble des sommets d'un graphe pondéré général. Dans la suite, ces fonctions seront appelées des signaux sur graphes [1]. Nous nous intéresserons en particulier au cas des images, modélisées ici comme des signaux sur graphes, mais le propos reste valable pour n'importe quel signal sur graphe.

Dans le contexte du traitement et de l'analyse d'images, le modèle de graphe pondéré permet de tenir compte des similarités généralement présentes dans les images naturelles. Considérer une image comme un signal sur un graphe amène à considérer des méthodes de traitement dites non locales [2, 3].

Dans ce même contexte, les transformées multi-échelles permettent de capturer les corrélations locales généralement présentes dans les signaux et ainsi d'obtenir des représentations hautement compressibles. L'efficacité de ces méthodes motive l'intérêt de l'introduction de telles approches pour le traitement de signaux définis sur des graphes généraux.

Les transformées en ondelettes classiques sont obtenues en calculant, à l'aide d'algorithmes rapides, des produits scalaires avec des translatées et dilatées d'une ondelette ad-

missible [4]. L'adaptation de cette approche à des signaux définis sur des graphes arbitraires pose des problèmes évidents. En particulier, la définition de notions intrinsèques de dilatation et de translation sur un graphe n'est pas immédiate. Ces difficultés ont donné lieu à une série de travaux récents [5, 6, 7, 8].

Nous proposons dans cet article une nouvelle transformée multi-échelle pour des signaux définis sur des graphes pondérés généraux. L'approche que nous adoptons ici est basée sur le schéma de lifting introduit par Sweldens dans [9] et sur la variante proposée récemment dans [8]. Nous étendons l'approche de [8] en itérant le procédé sur des sous-graphes induits et examinons ses performances d'approximation sur des images numériques.

## 2 Schéma de lifting

Le schéma de lifting factorise des transformées en ondelettes orthogonales ou biorthogonales en des étapes élémentaires pouvant être effectuées dans le domaine spatial. Cet algorithme a deux applications. La première est qu'il permet d'accélérer la transformée en ondelettes rapide en factorisant les étapes de filtrage numérique et de sous-échantillonnage. La deuxième application du schéma de lifting est qu'il permet la construction de transformées multi-échelles dites de *seconde génération*, c'est-à-dire des transformées s'appliquant à des signaux définis sur domaines bornés et des surfaces.

Illustrons rapidement le principe pour un signal 1D  $\mathbf{x} = (x_1, \dots, x_N)$ . La première étape du schéma de lifting consiste à *partitionner* les indices de  $\mathbf{x}$  en deux sous-ensembles, que l'on notera  $I$  et  $J$ . En dimension 1,  $I$  peut correspondre aux indices impairs et  $J$  aux indices pairs. Nous noterons  $\mathbf{x}^I = (x_i, i \in I)$ , et  $\mathbf{x}^J = (x_j, j \in J)$ . Une seule étape de lifting consiste en deux opérations : une opération de *prédiction* et une opération de *mise à jour*. Dans la première, on calcule, à partir du vecteur  $\mathbf{x}^J$ , à l'aide d'une application  $P : \mathbb{R}^{|J|} \rightarrow \mathbb{R}^{|I|}$ , une approximation  $\tilde{\mathbf{x}}^I$  de  $\mathbf{x}^I$  :  $\tilde{\mathbf{x}}^I = P(\mathbf{x}^J)$ , puis on évalue l'erreur de prédiction  $(\mathbf{x}^I)^* : (\mathbf{x}^I)^* = \mathbf{x}^I - \tilde{\mathbf{x}}^I$ . Dans la seconde opération, on calcule une approximation basse résolution du signal  $\mathbf{x}$  en additionnant le vecteur  $\mathbf{x}^J$  et une transformation linéaire  $U : \mathbb{R}^{|I|} \rightarrow \mathbb{R}^{|J|}$  de  $(\mathbf{x}^I)^*$ . On obtient alors le vecteur  $(\mathbf{x}^J)^* = \mathbf{x}^J + U((\mathbf{x}^I)^*)$ . Cette deuxième opération vise à préserver les moments d'ordre  $k$  du signal initial  $\mathbf{x}$  dans la version basse résolution  $(\mathbf{x}^J)^*$ . Une seule étape de lif-

ting correspond alors à l'application  $T : \mathbf{x} \in \mathbb{R}^N \rightarrow (\mathbf{x})^*$ . Remarquons à ce stade que  $T$  est bijective, indépendamment des choix des applications  $P$  et  $U$ . En effet, l'inverse de  $T$  s'obtient mécaniquement en inversant l'ordre des étapes et en changeant les signes :  $\mathbf{x}^J = (\mathbf{x}^J)^* - U((\mathbf{x}^I)^*)$ , et  $\mathbf{x}^I = (\mathbf{x}^I)^* + \tilde{\mathbf{x}}^I$ .

L'opérateur  $U$  de mise à jour étant linéaire, si l'on suppose de plus que  $P$  est linéaire, alors  $T$  correspond à un changement de base. Nous examinerons l'effet des opérations de mise à jour et de prédiction sur les signaux de base directement dans le contexte des signaux sur graphes dans la suite.

Nous venons de décrire une unique étape de lifting. Ce procédé peut être itéré en l'appliquant au signal  $(\mathbf{x}^J)^*$ . Notons que des étapes de normalisation peuvent être effectuées au préalable.

La transformée de Haar s'obtient avec le schéma précédent avec, d'une part, un opérateur de prédiction  $P$  correspondant au résultat d'une interpolation polynomiale constante par morceaux :  $\tilde{x}_{2m-1} = x_{2m}$ ,  $x_{2m-1}^* = x_{2m-1} - x_{2m}$ , et, d'autre part, la mise à jour suivante  $x_{2m}^* = x_{2m} + \frac{1}{2}x_{2m-1}^*$ , permettant de préserver la moyenne du signal.

Si l'on augmente l'ordre de l'opérateur de prédiction, c'est-à-dire si l'on passe d'une interpolation constante par morceaux à une interpolation linéaire par morceaux, on obtient la transformée biorthogonale CDF(2,2) de Cohen, Daubechies et Feauveau [4].

Notons que le schéma de lifting facilite la construction de transformées adaptatives en utilisant des opérateurs de prédiction non linéaires. Ce type d'approche a été proposé pour la compression de l'image et de la vidéo, par exemple dans [10, 11, 12].

Notons également que le schéma de lifting s'adapte naturellement à des signaux dont l'échantillonnage est irrégulier. En effet, cet échantillonnage peut être pris en compte dans l'étape de prédiction, si cette dernière est effectuée par exemple par interpolation polynomiale par morceaux. Ce type d'approches a été proposé notamment dans [13] pour la régression non paramétrique avec design aléatoire. Des techniques similaires ont également été proposées dans [14, 15, 16] pour la compression et la collecte distribuées de données enregistrées par des capteurs organisés en réseau.

Nous décrivons dans la sous-section suivante l'adaptation du schéma de lifting à des signaux définis sur graphes.

## 2.1 Lifting sur graphes

**Notations.** Nous commençons par introduire les définitions et notations que nous employons. Un graphe pondéré est un triplet  $G = (V(G), E(G), w)$  où  $V(G)$  est un ensemble de sommets,  $E(G) \subset V(G) \times V(G)$  un ensemble d'arêtes et  $w : V(G) \rightarrow ]0, +\infty[$  est une fonction de pondération exprimant une similarité. Nous nous limitons dans toute la suite de cet article à des graphes non orientés et sans boucles.

Nous noterons, pour  $i, j \in V(G)$ ,  $i \sim j$  si  $(i, j) \in E(G)$ .

Un graphe à  $N$  sommets dont l'ensemble des sommets est fixé peut être caractérisé par sa matrice d'adjacence pondérée  $\mathbf{W} \in \mathbb{R}^{N \times N}$  donnée par  $w_{i,j} = w(i, j)$  si  $(i, j) \in E$  et  $w_{i,j} = 0$  sinon.

L'espace  $\ell_2(V(G))$  désigne l'ensemble des signaux scalaires sur  $V(G)$ . Les éléments de  $\ell_2(V(G))$  seront écrits en gras. Le vecteur  $\mathbf{1}$  désigne le vecteur dont toutes les composantes sont égales à un.

**Adaptation du schéma de lifting.** Considérons un signal  $\mathbf{f} \in \ell_2(V(G))$ . Afin d'appliquer le schéma de lifting à  $\mathbf{f}$  il est nécessaire de spécifier trois opérateurs : de partitionnement, prédiction et de mise à jour. Nous commençons par préciser le choix des opérateurs de prédiction et de mise à jour. Nous supposons pour l'instant fixé un partitionnement  $(I, J)$  de  $V(G)$ , c'est-à-dire  $V(G) = I \cup J$  et  $I \cap J = \emptyset$ .

En ce qui concerne l'étape de prédiction nous nous limitons à des opérateurs linéaires. En pratique, la matrice d'adjacence pondérée  $\mathbf{W}$  est construite à partir des données en entrée de telle sorte que le poids associé à une arête reliant deux instances représente une mesure de similarité entre elles. Il est ainsi naturel d'utiliser ces poids comme coefficients de prédiction. En procédant de la sorte, l'étape de prédiction revient à multiplier par la matrice  $\mathbf{P}$  suivante :

$$(\forall j \in J) (\forall l \in V(G)) \quad p_{j,l} = \delta_{j,l} = \begin{cases} 1 & \text{si } j = l, \\ 0 & \text{sinon,} \end{cases} \quad (1)$$

et  $(\forall i \in I) (\forall l \in V(G))$

$$p_{i,l} = \begin{cases} \frac{w_{i,l}}{\sum_{j \in J, j \sim i} w_{i,j}} & \text{si } l \in J \text{ et } l \sim i \\ 0 & \text{sinon.} \end{cases} \quad (2)$$

Évaluer l'erreur de prédiction revient alors à multiplier par la matrice  $\mathbf{I} - \mathbf{P}$ .

De la même manière, l'étape de mise à jour peut s'écrire à l'aide d'une matrice  $\mathbf{U} \in \mathbb{R}^{N \times N}$  vérifiant

$$(\forall i \in I) (\forall l \in V(G)) \quad u_{i,l} = \delta_{i,l}, \quad (3)$$

et

$$(\forall j \in J) (\forall l \in V(G)) \quad u_{j,l} \neq 0 \implies l \sim j \text{ et } l \in I. \quad (4)$$

Nous précisons le choix de la matrice  $\mathbf{U}$  dans la section suivante.

Une unique étape de lifting peut alors être représentée à l'aide de la matrice  $\mathbf{T} \in \mathbb{R}^{N \times N}$  suivante :  $\mathbf{T} = (\mathbf{I} + \mathbf{U})(\mathbf{I} - \mathbf{P})$ . Posons  $\mathbf{g} = \mathbf{T}\mathbf{f}$ . Les composantes  $(\mathbf{g}_i)_{i \in I}$  correspondent aux coefficients de détails ou d'ondelettes alors que les composantes  $(\mathbf{g}_j)_{j \in J}$  correspondent à des coefficients d'échelles.

Plus généralement, plusieurs étapes de prédiction et/ou de mise à jour peuvent être effectuées. Une seule étape de lifting correspond alors à la matrice  $\mathbf{T}$  suivante :  $\mathbf{T} = \prod_{m=1}^k (\mathbf{I} + \mathbf{U}_m)(\mathbf{I} - \mathbf{P}_m)$ . Enfin, la transformée peut être itérée si l'on définit une relation d'adjacence sur l'ensemble  $J$ .

**Difficultés.** Nous identifions à ce stade une première difficulté concernant l'application du schéma de lifting à un graphe pondéré général. Cette difficulté concerne le partitionnement de l'ensemble des sommets en deux ensembles. Ce problème comporte deux aspects.

Le premier aspect concerne l'utilisation des arêtes disponibles dans le graphe. En effet, notons que lors d'une seule étape de lifting, seules les arêtes dont les extrémités appartiennent à deux parties différentes interviennent dans la prédiction et la mise à jour. Nous dirons d'une arête  $e = (l, k) \in E(G)$  dont les extrémités appartiennent à une même partie ( $l, k \in I$  ou  $l, k \in J$ ) qu'elle est *conflictuelle*. L'information contenue dans les arêtes conflictuelles n'est donc pas exploitée dans le schéma de lifting. Cet aspect doit être pris en compte lors du partitionnement.

Un critère naturel peut être obtenu en sommant les poids de toutes ces arêtes ; on est alors amené à minimiser ce critère. Il s'agit là du problème du sous-graphe couvrant biparti maximal, qui est équivalent au problème de la coupe maximale. Ce dernier problème est NP-dur. Dans [17], les auteurs utilisent l'algorithme randomisé proposé dans [18] pour calculer une solution approchée. Les auteurs de [19] utilisent l'algorithme proposé dans [20]. Les travaux [19] portent sur la compression de la vidéo.

Le deuxième aspect concerne la compressibilité des coefficients de détail. Ce point est clairement lié au précédent. Cependant, il est difficile d'énoncer un critère global à optimiser.

Au vu de ces difficultés, nous avons opté pour une approche gloutonne, inspirée par [8], consistant en une suite d'étapes de liftings élémentaires. Cette approche a aussi l'avantage de clairement montrer l'effet des opérations de prédiction et de mise à jour sur les signaux de bases. Nous détaillons cette approche dans la section suivante.

### 3 Liftings élémentaires

L'approche proposée dans [8] est résumée<sup>1</sup> dans l'algorithme 1.

Nous appellerons une itération de la boucle principale de l'algorithme 1 une étape de lifting élémentaire. La  $r^{\text{ième}}$  étape de lifting consiste à sélectionner un sommet  $i_r$  et à calculer un unique coefficient de détail  $d_r$  ainsi que des coefficients d'échelle  $c_{r,k}$ . Le coefficient  $d_r$  correspond à l'erreur de prédiction. L'interprétation d'une étape de lifting comme un changement de base permet de clarifier le choix des coefficients  $\beta_{r,k}$  dans (5). Nous nous intéressons d'abord, dans la sous-section suivante, à la procédure de sélection des sommets.

#### 3.1 Sélection des sommets

Nous proposons ici une procédure permettant la sélection des sommets de détail. Cette procédure consiste simplement à sélectionner, à chaque étape, le sommet de degré maximal parmi les sommets d'échelles restant :  $i_r =$

1. Nous adoptons une numérotation ascendante, contrairement à celle utilisée dans [8].

- 1: Entrée : Un graphe pondéré  $G$  sur  $\{1, N\}$  et un signal  $\mathbf{f} \in \ell_2(V(G))$ .
- 2: Initialisation :  $\mathcal{S}_0 := V(G)$ ,  $\mathcal{D}_0 := \emptyset$ ,  $c_{0,i} := f_i$ ,  $\forall i \in \mathcal{S}_0$ .
- 3: **Pour tout**  $r \in \{1, m\}$
- 4: Choisir un sommet  $i_r \in \mathcal{S}_{r-1}$ .
- 5:  $\mathcal{S}_r := \mathcal{S}_{r-1} \setminus \{i_r\}$
- 6:  $\mathcal{D}_r := \mathcal{D}_{r-1} \cup \{i_r\}$
- 7: Prédiction :  $d_r := c_{r-1,i_r} - \sum_{\substack{k \sim i_r \\ k \in \mathcal{S}_r}} \alpha_{r,k} c_{r-1,k}$ .
- 8: Mise à jour :
$$(\forall k \in \mathcal{S}_r) \quad c_{r,k} = \begin{cases} c_{r-1,k} + \beta_{r,k} d_r, & \text{si } k \sim i_r \\ c_{r-1,k}, & \text{sinon.} \end{cases} \quad (5)$$
- 9: **Fin**

Algorithme 1 – Transformée multi-échelles par suite de liftings élémentaires

$\arg \max_{i \in \mathcal{S}_{r-1}} \sum_{j \in \mathcal{S}_{r-1}} w_{i,j}$ . Intuitivement, l'erreur de prédiction doit décroître selon la centralité du noeud. Ici, la mesure de centralité adoptée est le degré pondéré. Ce choix favorise ainsi la compressibilité de la transformée globale.

#### 3.2 Prédiction

Comme mentionné dans la section précédente, on se limite à des opérateurs linéaires dont les coefficients sont reliés aux poids des arêtes du graphe :  $(\forall k \in \mathcal{S}_r, k \sim i_r) \quad \alpha_{r,k} = \frac{w_{i_r,k}}{\sum_{l \in \mathcal{S}_r} w_{i_r,l}}$ .

Ce choix est motivé par le fait que les graphes que nous considérons ici sont des graphes de similarité. La normalisation vise à obtenir des coefficients de détail nuls pour des signaux localement constants.

Remarquons que bien que l'opérateur de prédiction soit linéaire si l'on considère la topologie et les poids des arêtes du graphe fixés, il peut être globalement non linéaire si l'on autorise le graphe à dépendre du signal que l'on analyse.

#### 3.3 Changement de base

La transformée que nous venons de présenter est linéaire et bijective. Elle correspond donc à un changement de base. Soit  $T_r : \mathbb{R}^N \rightarrow \mathbb{R}^N$  la transformée correspondant à l'application de  $r$  étapes de liftings élémentaires :

$$(\forall \mathbf{f} \in \mathbb{R}^N) \quad (T_r \mathbf{f})_l = \begin{cases} c_{r,l} & \text{si } l \in \mathcal{S}_r, \\ d_l & \text{si } l \in \mathcal{D}_r. \end{cases} \quad (6)$$

Soit  $(\phi_{0,i})_{1 \leq i \leq N}$  la base canonique de  $\mathbb{R}^N$  ( $\phi_{0,i}(j) = \delta_{i,j}$ ). À chaque étape  $r \in \{1, \dots, m\}$ , les coordonnées de  $T_r \mathbf{f}$  dans la base canonique correspondent aux coordonnées de  $\mathbf{f}$  dans une nouvelles base multi-échelle notée  $MSB_r := \{\phi_{r,i}, \psi_{r,j}\}_{i \in \mathcal{D}_r, j \in \mathcal{D}_r}$  :

$$\mathbf{f} = \sum_{i \in \mathcal{S}_r} c_{r,i} \phi_{r,i} + \sum_{i \in \mathcal{D}_r} d_i \psi_{r,i}. \quad (7)$$

En raison de la linéarité de  $T_r$ , la base  $MSB_r$  est donnée simplement en appliquant  $T_r^{-1}$  à  $(\phi_{0,i})_{1 \leq i \leq N}$  :

$$\begin{cases} (\forall i \in \mathcal{S}_r) & \phi_{r,i} = T_r^{-1}(\phi_{0,i}), \\ (\forall j \in \mathcal{D}_r) & \psi_{r,j} = T_r^{-1}(\phi_{0,i}). \end{cases} \quad (8)$$

L'expression précédente permet de relier les éléments de deux base successives. Cette relation est la suivante :

$$(\forall i \in \mathcal{S}_r) \quad \phi_{r,i} = \begin{cases} \phi_{r-1,i} + \alpha_{r,i} \phi_{r-1,i_r} & \text{si } i \sim i_r \\ \phi_{r-1,i} & \text{sinon.} \end{cases} \quad (9)$$

et

$$(\forall i \in \mathcal{D}_r) \quad \psi_{r,i} = \begin{cases} \phi_{r-1,i_r} - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} \phi_{r,i} & \text{si } i = i_r, \\ \psi_{r-1,i} & \text{sinon.} \end{cases} \quad (10)$$

### 3.4 Mise à jour

L'équation (10) permet de clarifier le rôle de l'étape de mise à jour. En effet, notons  $I_{r,i} = \langle \phi_{r,i}, \mathbf{1} \rangle$ ,  $\langle \cdot, \cdot \rangle$  désignant le produit scalaire usuel sur  $\mathbb{R}^N$ . Notons que  $I_{0,i} = 1$  pour tout  $i \in \{1, \dots, N\}$ . Comme pour les transformées en ondelettes usuelles, nous cherchons ici à construire un signal dont la moyenne est nulle. Cette condition est équivalente, dans notre cadre, à  $I_{r-1,i_r} = \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} I_{r,i}$

pour tout  $r \in \{1, \dots, m\}$ . Il s'agit là d'un système sous-déterminé et nous optons, comme dans [8], pour la solution de norme euclidienne minimale, correspondant à  $\beta_{r,i} = \frac{I_{r,i} I_{r-1,i_r}}{\sum_{\substack{j \sim i_r \\ j \in \mathcal{S}_r}} I_{r,j}^2}$ .

### 3.5 Itération sur des sous-graphes induits

Le nombre d'étapes de liftings élémentaires est spécifié dans l'algorithme 1 par le paramètre  $m$ . Ce paramètre dépend de la topologie du graphe. En effet, chaque étape de lifting invalide toutes les arêtes incidentes au sommet de détail sélectionné. Ainsi, selon la topologie du graphe considéré, il se peut que l'itération d'un nombre  $k < N - 1$  d'étapes de liftings élémentaires invalide l'ensemble des arêtes du graphe. Dans ce cas, nous dirons que nous avons effectué un cycle de la transformée.

Afin de pouvoir procéder à de nouveaux cycles de la transformée, il est nécessaire de définir un nouveau graphe. Notons  $m_i$  le nombre de liftings élémentaires à la fin de chaque cycle  $i$  ( $m_1 = m$ ). Ainsi, à la fin du cycle  $i$ , nous avons  $V(G) = \mathcal{S}_{m_i} \cup \mathcal{D}_{m_i}$ . Un nouveau graphe doit être défini sur  $\mathcal{S}_{m_i}$ . Nous proposons d'inférer la topologie ainsi que les poids des arêtes de ces graphes à partir de la matrice d'adjacence  $\mathbf{W}$ . Ainsi, deux sommets  $i, j \in \mathcal{S}_{m_i}$  sont connectés si et seulement si  $\mathbf{W}_{i,j}^{i+1} \neq 0$ . Observons que si  $\mathbf{W}$  est binaire,  $\mathbf{W}_{i,j}^n$  donne le nombre de chemins de longueur  $n$  reliant  $i$  et  $j$ .

### 3.6 Expériences

Nous illustrons dans cette sous-section le comportement de la transformée de lifting que nous venons de décrire. Nous

limitons aux cas des images numériques, bien que l'approche décrite plus haut s'applique à des signaux plus généraux. Nous commençons tout d'abord par expliquer les constructions de graphes que nous employons. L'ensemble des sommets de tous les graphes que nous considérons correspond à l'ensemble des pixels. Nous considérons trois relations d'adjacence : 4-voisinage, 8-voisinage et plus proches voisins. La distance intervenant dans le choix des plus proches voisins est la distance photométrique. Cette même distance peut être transformée en similarité en utilisant un noyau gaussien,  $w_{i,j} = e^{-d_{i,j}^2/\sigma^2}$ , et peut servir alors à pondérer les arêtes obtenues.

L'illustration que nous donnons est l'erreur d'approximation non linéaire à  $M$  termes : la transformée de lifting est appliquée aux images 128x128 "Lena" et "Barbara". Ensuite, seuls les  $M$  plus grands coefficients de détail (en valeur absolue) sont gardés : les autres coefficients sont remplacés par zéro. Enfin la transformée inverse est appliquée et l'erreur d'approximation calculée. Nous avons opté pour le psnr entre l'image originale et celle reconstruite à partir des  $M$  plus grands coefficients. La formule du psnr est donnée par :  $psnr(f, g) = 10 \log_{10}(\frac{255^2}{EQM(f,g)})$ , où EQM désigne l'erreur quadratique moyenne. Toutes les transformées engendrent un unique coefficient d'échelle : tous les autres coefficients correspondent à des coefficients de détail.

La figure 1 montre l'erreur d'approximation de l'image "Barbara" pour différentes valeurs de  $M$ . Sur cette figure nous montrons, d'une part, l'erreur d'approximation de la transformée de lifting pour différentes constructions de graphes, d'autre part, l'erreur d'approximation obtenue avec les transformées orthogonales de Haar, Daubechies 4, Daubechies 8, et biorthogonale de CDF(2,2). Les différentes configurations de graphes que nous avons considérées sont : le graphe de 4 voisinage non pondéré, le graphe de 8 voisinage pondéré avec des poids gaussiens ( $\sigma = 15$ ) et le graphe des 25 au sein d'une fenêtre 21x21 avec des poids gaussiens ( $\sigma = 10$ ).

Les transformées basées sur les deux dernières configurations fournissent des erreurs de prédiction bien plus faibles que celles des transformées classiques non adaptatives. Notons cependant que pour effectuer la transformée inverse il est nécessaire de disposer, d'une part, du graphe sous-jacent, d'autre part, des coefficients de lifting. Les performances montrées dans la figure 1 ne peuvent donc pas être exploitées directement pour effectuer la compression des images en question. Le point important ici est qu'il est possible de construire des représentations très compressibles en appliquant la transformée de lifting que nous décrivons.

Les figures 2 et 3 montrent les approximations à 1024 et 256 termes, respectivement, de l'image "Barbara" pour différents graphes et pour différentes transformées. Ces images confirment les performances de la figure 1. Enfin, les mêmes tests ont été effectués sur l'image "Lena". Les résultats, reportés dans les figures 4, 5 et 6, sont similaires.

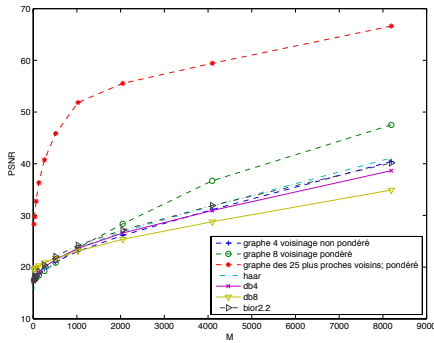


Figure 1 – Erreur d'approximation non linéaire à  $M$  termes pour l'image "Barbara". Transformées considérées : lifting sur graphe (4 et 8 voisinage, plus proches voisins), Haar, Daubechies 4, Daubechies 8, CDF(2,2).

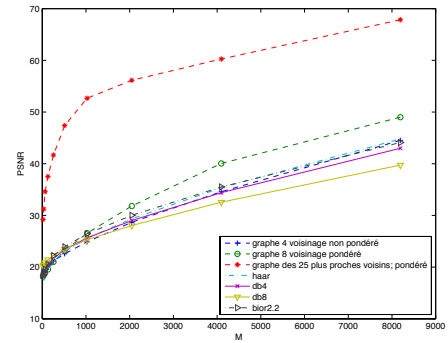


Figure 4 – Erreur d'approximation non linéaire à  $M$  termes pour l'image "Lena". Transformées considérées : lifting sur graphe (4 et 8 voisinage, plus proches voisins), Haar, Daubechies 4, Daubechies 8, CDF(2,2).

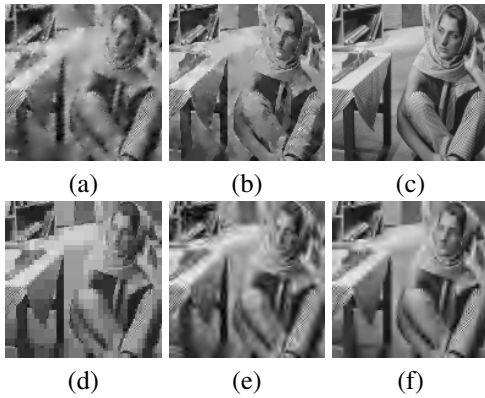


Figure 2 – Reconstructions de l'image "Barbara" à partir de 1024 coefficients. (a) lifting sur graphe 4 voisinage non pondéré. (b) lifting sur graphe 8 voisinage pondéré. (c) lifting sur graphe des 25 plus proches voisins; poids gaussiens. (d) Haar. (e) Daubechies 4. (f) : CDF(2,2).



Figure 5 – Reconstructions de l'image "Lena" à partir de 1024 coefficients. (a) lifting sur graphe 4 voisinage non pondéré. (b) lifting sur graphe 8 voisinage pondéré. (c) lifting sur graphe des 25 plus proches voisins; poids gaussiens. (d) Haar. (e) Daubechies 4. (f) CDF(2,2).

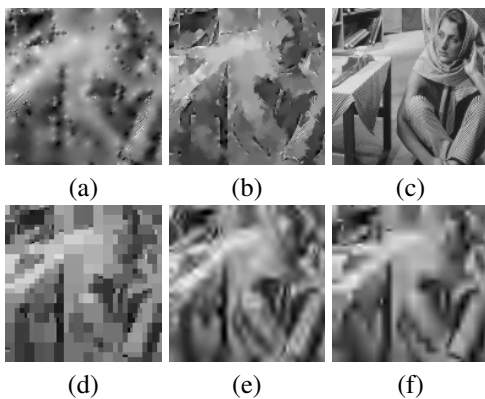


Figure 3 – Reconstructions de l'image "Barbara" à partir de 256 coefficients. (a) lifting sur graphe 4 voisinage non pondéré. (b) lifting sur graphe 8 voisinage pondéré. (c) lifting sur graphe des 25 plus proches voisins; poids gaussiens. (d) Haar. (e) Daubechies 4. (f) CDF(2,2).

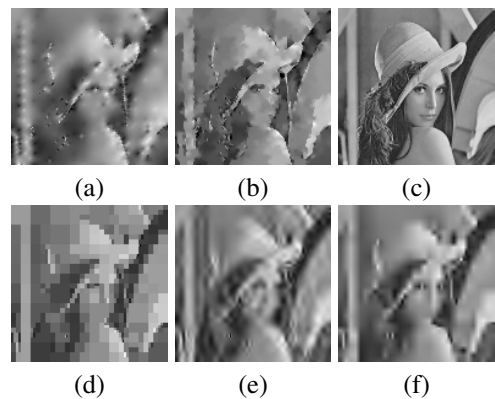


Figure 6 – Reconstructions de l'image "Lena" à partir de 256 coefficients. (a) lifting sur graphe 4 voisinage non pondéré. (b) lifting sur graphe 8 voisinage pondéré. (c) lifting sur sur graphe des 25 plus proches voisins; poids gaussiens. (d) Haar. (e) Daubechies 4. (f) CDF(2,2).

## 4 Conclusion

Nous avons proposé dans ce travail une adaptation du schéma de lifting à des signaux définis sur l'ensemble des sommets d'un graphe pondéré général. Nous avons suivi une approche récemment proposée dans [8] et l'avons étendue en considérant l'itération de cette transformée à des sous-graphes induits. Nous avons également examiné avec soin l'effet de chaque étape de l'algorithme proposé sur la construction de la base sous-jacente. Enfin, nous avons examiné l'utilisation de notre transformée pour l'approximation d'images numériques.

La transformée que nous avons proposée permet d'obtenir des approximations très fidèles dans la mesure où la structure de l'image est codée par le graphe (topologie et poids) sous-jacent construit à partir de l'image. Une première direction d'investigation concerne la recherche d'un compromis entre la parcimonie de la matrice d'adjacence  $W$  et les propriétés d'approximation de la transformée. Enfin, une autre direction de recherche concerne l'utilisation de notre transformée pour la résolution de problèmes inverses faisant intervenir des signaux sur graphes.

## Références

- [1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, et P. Vandergheynst. The emerging field of signal processing on graphs : Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Sig. Process. Mag.*, 30(3) :83–98, 2013.
- [2] A. Elmoataz, O. Lézoray, et S. Boughleux. Nonlocal discrete regularization on weighted graphs : A framework for image and manifold processing. *IEEE T. Image Process.*, 17(7) :1047–1060, jul 2008.
- [3] G. Gilboa et S. Osher. Nonlocal operators with applications to image processing. *Multiscale Model Sim.*, 7(3) :1005–1028, 2008.
- [4] S. Mallat. *A Wavelet Tour of Signal Processing, the Sparse way*. Academic Press, 2009.
- [5] R.R. Coifman et M. Maggioni. Diffusion wavelets. *Applied Comput. Harmon. A.*, 21(1) :53–94, 2006.
- [6] D.K. Hammond, P. Vandergheynst, et R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied Comput. Harmon. A.*, 30(2) :129–150, 2011.
- [7] S. K. Narang et A. Ortega. Perfect reconstruction two-channel wavelet filter banks for graph structured data. *IEEE T. Sig. Process.*, 60(6) :2786–2799, 2012.
- [8] M. Jansen, G.P. Nason, et B.W. Silverman. Multiscale methods for data on graphs and irregular multidimensional situations. *J. Roy. Stat. Soc. B.*, 71(1) :97–125, 2009.
- [9] W. Sweldens. The lifting scheme : A construction of second generation wavelets. *SIAM J. on Mathematical Analysis*, 29(2) :511–546, 1998.
- [10] R.L. Claypoole, G.M. Davis, W. Sweldens, et R.G. Baraniuk. Nonlinear wavelet transforms for image coding via lifting. *Image Processing, IEEE Transactions on*, 12(12) :1449–1459, 2003.
- [11] G. Piella et H.J.A.M. Heijmans. Adaptive lifting schemes with perfect reconstruction. *Signal Processing, IEEE Transactions on*, 50(7) :1620–1630, 2002.
- [12] M. Kaaniche, B. Pesquet-Popescu, A. Benazza-Benyahia, J.C. Pesquet, et al. Adaptive lifting scheme with sparse criteria for image coding. *EURASIP Journal on Advances in Sig. Proc.*, 2012 :22, 2012.
- [13] V. Delouille, J. Simoens, et R. von Sachs. Smooth design-adapted wavelets for nonparametric stochastic regression. *Journal of the American Statistical Association*, 99(467) :643–658, 2004.
- [14] R. Wagner, H. Choi, R. Baraniuk, et V. Delouille. Distributed wavelet transform for irregular sensor network grids. Dans *SSP, 2005 IEEE/SP 13th Workshop on*, pages 1196–1201. IEEE, 2005.
- [15] G. Shen et A. Ortega. Optimized distributed 2d transforms for irregularly sampled sensor network grids using wavelet lifting. Dans *ICASSP 2008*, pages 2513–2516. IEEE, 2008.
- [16] R. Baraniuk, A. Cohen, et R. Wagner. Approximation and compression of scattered data by meshless multiscale decompositions. *Applied and Comput. H. Analysis*, 25(2) :133–147, 2008.
- [17] S.K. Narang et A. Ortega. Lifting based wavelet transforms on graphs. Dans *APSIPA ASC 2009*, pages 441–444, 2009.
- [18] S. Fitzpatrick et L. Meertens. An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs. *Stochastic Algorithms : Foundations and Applications*, pages 345–347, 2001.
- [19] E. Martinez-Enriquez, E. Diaz-de Maria, et A. Ortega. Video encoder based on lifting transforms on graphs. Dans *ICIP 2011*, pages 3509–3512. IEEE, 2011.
- [20] C.P. Hsu. Minimum-via topological routing. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 2(4) :235–246, 1983.