



Représentation multi-échelle hiérarchique de données définies sur des graphes

Moncef Hidane, Olivier Lezoray, Abderrahim Elmoataz, Vinh Thong Ta

► To cite this version:

Moncef Hidane, Olivier Lezoray, Abderrahim Elmoataz, Vinh Thong Ta. Représentation multi-échelle hiérarchique de données définies sur des graphes. GRETSI 2011, 2011, Bordeaux, France. 4 pp. hal-00708978

HAL Id: hal-00708978

<https://hal.science/hal-00708978>

Submitted on 4 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Représentation multi-échelle hiérarchique de données définies sur des graphes

Moncef HIDANE¹, Olivier LÉZORAY¹, Abderrahim ELMOATAZ¹, Vinh-Thong TA²

¹Université de Caen Basse-Normandie, GREYC UMR CNRS 6072
6, Boulevard du Maréchal Juin, 14050 Caen Cedex, France

²LaBRI (Université de Bordeaux-CNRS-IPB)
351, cours de la Libération, 33405 Talence Cedex, France
moncef.hidane@unicaen.fr, olivier.lezoray@unicaen.fr
abderrahim.elmoataz@unicaen.fr, vinh-thong.ta@labri.fr

Résumé – Nous proposons dans cet article une approche permettant d’obtenir une décomposition hiérarchique multi-échelle de données organisées sous forme de graphes. Notre approche est inspirée par les travaux de [7]. Nous détaillons la construction des graphes et fournissons une méthode de sélection des paramètres afin d’obtenir un algorithme automatique.

Abstract – A novel approach for the multiscale hierarchical representation of discrete data on graphs is proposed. Our proposal is inspired by the work of [7]. We address the issues of graph construction and parameter selection in order to have an automatic algorithm.

1 Introduction

Il est maintenant établi que la vision est un phénomène multi-échelle. Une tâche visuelle amène alors à représenter et interpréter une scène de part ses singularités. Par conséquent, il est fondamental de disposer d’un formalisme permettant de décomposer une image en ses composantes pertinentes. Or, les images contiennent généralement deux types d’information : structure et texture. Un formalisme de représentation doit alors pouvoir extraire à différentes échelles ces deux types d’information.

Nous adoptons un point de vue discret, basé sur des graphes, afin de réaliser cet objectif. Ce parti pris est motivé par les développements récents des méthodes de traitements et de régularisation sur graphes [4]. Ceci nous permet d’étendre la décomposition hiérarchique multi-échelle d’images introduite dans [7]. Nous étudions les propriétés de cette nouvelle représentation et montrons comment elle s’applique à la représentation de données discrètes générales dont le support est un graphe.

2 Représentation multi-échelle hiérarchique d’images

La représentation multi-échelle hiérarchique d’images introduite dans [7] est basée sur la minimisation de la variation totale (TV) [6]. Étant donnée une image initiale f définie sur un domaine continu $\Omega \subset \mathbb{R}^2$, le modèle TV- L^2 consiste à minimi-

ser la fonctionnelle suivante :

$$J(u; f, \lambda) = \int_{\Omega} |\nabla u| + \lambda \|f - u\|_{L^2(\Omega)}^2. \quad (1)$$

Depuis les travaux de Meyer [5], la minimisation de (1) est aussi interprétée comme une décomposition structure/texture : l’image initiale f est décomposée en une partie u contenant la géométrie des objets et une partie v contenant les motifs oscillants. La séparation entre les deux parties dépend du paramètre λ , qui peut alors être interprété comme un paramètre d’échelle.

Dans [7], les auteurs proposent d’étendre cette décomposition mono-échelle à une décomposition multi-échelle hiérarchique. Étant donnée une suite croissante d’échelles $\lambda_0 < \lambda_1 < \dots < \lambda_n$, la représentation est obtenue par décompositions successives des résidus. Une première décomposition de f est obtenue en utilisant le paramètre λ_0 : on obtient la décomposition $f = u_0 + v_0$ où u_0 est l’argument minimum et v_0 le résidu. Ensuite, v_0 est redécomposé en utilisant (1) avec $\lambda = \lambda_1$. On obtient la décomposition $v_0 = u_1 + v_1$. On obtient finalement la représentation suivante $f = u_0 + \dots + u_n + v_n$ en itérant ce procédé. La suite $(u_i)_{i=0\dots n}$ représente l’image f aux différentes échelles $\lambda_0, \dots, \lambda_n$, tandis que la somme $u_0 + \dots + u_n$ fournit une représentation multi-couche de l’image initiale f . La question de la convergence de la représentation est traitée dans [7]. Nous proposons, dans cet article, de généraliser le principe de la décomposition hiérarchique multi-échelle à des données discrètes quelconques, en nous basant sur les outils de la régularisation sur graphes [4].

3 Régularisation sur graphes

Un graphe pondéré est un triplet $G = (V, E, w)$ où V est un ensemble de noeuds (ou sommets), $E \subset V \times V$ un ensemble d'arêtes, et $w : E \rightarrow \mathbb{R}^+$ une mesure de similarité symétrique. On note $\mathcal{H}(V)$, respectivement $\mathcal{H}(E)$, l'ensemble des fonctions numériques définies sur V , respectivement sur E . Les espaces $\mathcal{H}(V)$ et $\mathcal{H}(E)$ sont munis des produits scalaires usuels $(f, g)_{\mathcal{H}(V)} = \sum_{\alpha \in V} f(\alpha)g(\alpha)$ pour $f, g \in \mathcal{H}(V)$, et $(F, G)_{\mathcal{H}(E)} = \sum_{(\alpha, \beta) \in E} F(\alpha, \beta)G(\alpha, \beta)$ pour $F, G \in \mathcal{H}(E)$. Enfin, si $\alpha \in V$, nous notons $\mathcal{N}(\alpha)$ l'ensemble des noeuds voisins de α . L'opérateur de différence pondéré $d_w : \mathcal{H}(V) \rightarrow \mathcal{H}(E)$ est défini ainsi : pour $f \in \mathcal{H}(V)$, $(\alpha, \beta) \in E$

$$(d_w f)(\alpha, \beta) = \sqrt{w(\alpha, \beta)} (f(\beta) - f(\alpha)). \quad (2)$$

Il permet de mesurer la variation d'une fonction le long d'une arête. L'opérateur d_w permet d'introduire un gradient pondéré ∇_w : pour $f \in \mathcal{H}(V)$ et $\alpha \in V$,

$$(\nabla_w f)(\alpha) = ((d_w f)(\alpha, \beta), \beta \in \mathcal{N}(\alpha))^T. \quad (3)$$

Le choix d'une norme induit alors une mesure de la variation locale. Dans la suite, nous choisissons la norme Euclidienne :

$$\|(\nabla_w f)(\alpha)\| = \sqrt{\sum_{\beta \in \mathcal{N}(\alpha)} w(\alpha, \beta) (f(\alpha) - f(\beta))^2}. \quad (4)$$

L'opérateur $d_w^* : \mathcal{H}(E) \rightarrow \mathcal{H}(V)$, adjoint de d_w , est défini par $(d_w f, F)_{\mathcal{H}(E)} = (f, d_w^* F)_{\mathcal{H}(V)}$, pour tous $f \in \mathcal{H}(V)$, $F \in \mathcal{H}(E)$. L'expression est donnée par

$$(d_w^* F)(\alpha) = \sum_{\beta \in \mathcal{N}(\alpha)} \sqrt{w(\alpha, \beta)} (F(\beta, \alpha) - F(\alpha, \beta)). \quad (5)$$

L'opérateur divergence est défini de manière analogue au cas continu : $\text{div}_w = -d_w^*$. Les opérateurs sur graphes que l'on vient de rappeler permettent de définir une famille de fonctionnelles de régularisation :

$$E_p(u; f, \lambda) = \sum_{\alpha \in V} \|(\nabla u)(\alpha)\|^p + \lambda \|f - u\|^2. \quad (6)$$

Nous considérons ici uniquement le cas où $p = 1$ qui peut être vu comme une extension du modèle TV-L² aux graphes de topologie arbitraire. La fonctionnelle correspondante est alors strictement convexe mais non différentiable. Les arguments utilisés dans [2] s'adaptent à notre formulation sur graphes. Ils permettent d'avoir la caractérisation suivante du minimum de (6) avec $p = 1$:

$$u^* = f - \pi_{\delta K}(f), \quad (7)$$

$$K = \{v \in \mathcal{H}(V) : \exists p \in \mathcal{H}(E), v = \text{div}_w p, \|p\|_\infty \leq 1\}, \quad (8)$$

$\|p\|_\infty = \max_{\alpha \in V} |p(\alpha, \cdot)|, |p(\cdot, \alpha)| = \sqrt{\sum_{\beta \sim \alpha} p(\alpha, \beta)^2}$, $\delta = \frac{2}{\lambda}$ et $\pi_{\delta K}$ est la projection sur l'ensemble convexe et fermé δK . Avec nos notations, l'algorithme de projection de Chambolle [2] est donné par l'itération suivante :

$$\begin{cases} p^0 = 0, \\ p^{n+1}(\alpha, \beta) = \frac{p^n(\alpha, \beta) + \tau d_w(\text{div}_w p^n - \lambda f)(\alpha, \beta)}{1 + \tau |d_w(\text{div}_w p^n - \lambda f)(\alpha, \cdot)|}. \end{cases} \quad (9)$$

Finalement, la solution est donnée par $u^* = f - \delta \text{div}_w(p^*)$, où p^* est la limite de (9).

4 Représentation multi-échelle hiérarchique sur des graphes de topologies arbitraires

Soit $G = (V, E, w)$ un graphe pondéré, $f \in \mathcal{H}(V)$ une donnée sur G , et $0 < \lambda_0 < \dots < \lambda_n$, $n \geq 1$. Dans la pratique, $\lambda_i = 2^i \lambda_0$. La décomposition mono-échelle résultant de l'application de la régularisation (6) avec $p = 1$ et $\lambda = \lambda_0$ est transformée en une représentation multi-échelle hiérarchique de la même manière que dans [7]. L'algorithme peut être résumé ainsi :

$$\begin{cases} v_{-1} = f, \\ u_{i+1} = \underset{u \in \mathcal{H}(V)}{\text{arginf}} E_1(u; v_i, \lambda_{i+1}), i \geq -1, \\ v_{i+1} = v_i - u_{i+1}, i \geq -1. \end{cases} \quad (10)$$

Nous obtenons alors la décomposition suivante :

$$f = u_0 + \dots + u_n + v_n. \quad (11)$$

Il est intéressant de s'intéresser au comportement asymptotique de la décomposition (11). Comme on peut s'y attendre, si la suite d'échelles $(\lambda_n)_{n \geq 0}$ est croissante alors la suite des résidus $(v_n)_{n \geq 0}$ converge vers 0. La démonstration de ce résultat peut être obtenue directement grâce à la caractérisation (7).

Proposition. Soit $f \in \mathcal{H}(V)$ et $(\lambda_n)_{n \geq 0}$ une suite strictement croissante d'entiers strictement positifs. Soit $f = u_0 + \dots + u_n + v_n$ la décomposition obtenue en appliquant l'algorithme (10) avec la suite $(\lambda_n)_{n \geq 0}$. Alors,

$$\|f - \sum_{i=0}^n u_i\|_{\mathcal{H}(V)} \rightarrow 0, \text{ quand } n \rightarrow \infty. \quad (12)$$

Démonstration. Nous avons

$$\|f - \sum_{i=0}^n u_i\|_{\mathcal{H}(V)} = \|v_n\|_{\mathcal{H}(V)}. \quad (13)$$

Mais, d'après la caractérisation (7), il existe $p_n \in \mathcal{H}(E)$ tel que $v_n = \text{div}_w(p_n)$ et $\|p_n\|_\infty \leq \frac{2}{\lambda_n}$. L'opérateur div_w étant linéaire, on note $\|\text{div}_w\|$ sa norme subordonnée aux normes $\|\cdot\|_{\mathcal{H}(V)}$ sur $\mathcal{H}(V)$ et $\|\cdot\|_\infty$ sur $\mathcal{H}(E)$. Nous avons alors

$$\|\text{div}_w(p_n)\| \leq \|\text{div}_w\| \|p_n\|_\infty \leq \|\text{div}_w\| \frac{2}{\lambda_n}. \quad (14)$$

Ce qui montre le résultat. $\square$

Le choix du paramètre λ_0 est important. S'il est choisi trop petit, plusieurs niveaux seront requis avant que les détails ne soient reconstruits. Pour des données de grande dimension ($f : V \rightarrow \mathbb{R}^m$, avec m très grand), ou des graphes très denses, un mauvais choix du paramètre λ_0 peut avoir un impact très important sur la complexité de l'algorithme (10). La construction du graphe est aussi une composante importante de notre algorithme. Nous détaillons dans les sections suivantes ces aspects et montrons l'applicabilité de notre formulation à divers type de données organisées ou non organisées.

5 Construction des graphes

Les données définies sur des graphes se répartissent en deux catégories : organisées et non organisées. Pour la première classe, la structure de graphe est connue a priori, tandis que pour la seconde, un graphe de voisinage doit être construit. Les images numériques et les maillages sont des exemples de données organisées, alors que les nuages de points sont un exemple de données non organisées. Même pour des données organisées, il peut être judicieux de considérer un graphe modifié, soit en introduisant une pondération sur les arêtes présentes, soit en considérant une nouvelle topologie. Nous utilisons cette démarche dans le cas des images numériques, ce qui conduit à des traitements non locaux qui ont été popularisés depuis l'algorithme des moyennes non locales [1]. Nous montrons dans les expériences de la section 7 que ces méthodes produisent des résultats meilleurs que ceux obtenus avec des traitements locaux.

Afin de construire des graphes de voisinages, nous considérons dans cet article le graphe des plus proches voisins. Ce graphe associe à chaque sommet ses k plus proches sommets, où k est un entier positif préalablement fixé. Le graphe des plus proches voisins est orienté. Afin de travailler sur un graphe non orienté nous choisissons de connecter chaque noeud à ses plus proches voisins et aux noeuds dont il est un des plus proches voisins. Ceci conduit à un graphe non orienté où chaque noeud est connecté à au moins k autres noeuds. Dans le cas des images, la recherche des plus proches voisins se fait au sein d'une fenêtre de recherche centrée en chacun des pixels. De plus, dans ce cas, le graphe obtenu est couplé avec le graphe grille afin de garantir la connectivité. Le paramètre k , désignant le nombre de plus proches voisins est fixé à 25 dans le cas des images numériques. Une telle valeur permet de garder une complexité raisonnable à notre algorithme. Pour le cas de données non organisées, le paramètre k doit être tel que le graphe obtenu est connecté.

La mesure de distance utilisée pour trouver les plus proches voisins est elle même basée sur les observations dont on dispose. Dans notre cas, la distance entre deux noeuds est calculée comme étant la distance Euclidienne entre leur vecteur d'attributs. Dans le cas des images numériques, la distance Euclidienne entre les vecteur d'attributs (intensité ou couleur) peut être remplacée par la distance Euclidienne entre les attributs des patches centrés en chacun des pixels.

Enfin, nous introduisons une mesure de similarité sur les arêtes du graphe en choisissant une fonction décroissante et en l'appliquant aux distances préalablement calculées. Dans cet article, nous utilisons les noyaux suivants :

$$1) w(\alpha, \beta) = \frac{1}{1+d(\alpha, \beta)}$$

$$2) w(\alpha, \beta) = e^{\frac{-d^2(\alpha, \beta)}{\sigma^2}}$$

où $\sigma > 0$ est fixé et $d(\alpha, \beta)$ désigne la distance entre les noeuds α et β . Pour le noyau exponentiel, le paramètre σ est généralement difficile à régler. Nous proposons de l'estimer localement en chaque noeud. À cette fin, nous fixons la taille d'une fenêtre centrée en chaque pixel, et estimons la variance du signal

au sein de cette fenêtre. Si α et β désignent deux noeuds du graphe, σ_α et σ_β les variances empiriques estimées au sein de deux fenêtres de même taille centrées en α et β , alors la similarité entre les noeuds α et β est donnée par : $e^{\frac{-d^2(\alpha, \beta)}{\sigma_\alpha \sigma_\beta}}$.

6 Paramètres de régularisation

Nous avons détaillé dans la section précédente la construction de graphes pondérés que nous utilisons. Afin d'avoir un algorithme de décomposition sans paramètres, il reste à considérer le choix de la suite d'échelles λ_i . Dans les expériences que nous proposons dans la section suivante, nous avons choisi une progression géométrique : $\lambda_{i+1} = 2\lambda_i$. Ce choix est classique en traitement du signal. Ceci nous amène donc au choix du paramètre d'échelle initial λ_0 . Dans le cas du débruitage dans un modèle additif Gaussien, ce paramètre est lié au multiplicateur de Lagrange obtenu en considérant la version contrainte de la minimisation de (6). La paramètre λ_0 peut être calculé comme dans [3] en estimant la variance du signal. Cependant, dans notre cas le paramètre λ_0 ne sert pas à débruiter l'image, mais à obtenir une première décomposition des données que nous observons. À cette fin, nous proposons de le choisir comme multiple de $\frac{1}{\sigma}$ où σ est une estimation de l'écart-type. Dans les expériences, nous avons choisi $\lambda_0 = \frac{1}{2\sigma}$.

7 Expériences

Nous montrons brièvement l'application de notre approche avec trois exemples : la représentation hiérarchique d'une image digitale, celle d'un maillage 3D, et enfin celle d'un nuage de points. Dans le premier cas (figure 1), nous montrons comment l'intégration d'informations non locales permet une meilleure représentation de l'image initiale. La première ligne correspond à l'algorithme obtenu en utilisant un graphe grille non pondéré. La seconde ligne correspond à notre approche avec un graphe des 25-plus proches voisins sélectionnés dans une fenêtre de taille 7x7, et dont les arêtes ont été pondérées par une fonction Gaussienne comme expliqué dans les sections précédentes. Nous remarquons dans ce dernier cas que la décomposition obtenue au premier niveau contient uniquement les contours de l'image initiale. De plus, la séparation des échelles est plus nette, chaque échelle permet de reconstruire un niveau de texture donné. Les textures les plus fines sont reconstruites de manière homogène au dernier niveau. Enfin, avec notre approche, le bruit présent dans l'image initiale n'a pas été réintégré au dernier niveau. Dans la figure 2, la structure du graphe est donnée directement par le maillage. Les données à régulariser sont les coordonnées dans l'espace des sommets du maillage. Cet exemple montre l'applicabilité de notre approche à des données discrètes organisées. Enfin, la figure 3, montre la représentation d'une hélice toroïdale bruitée.

La première décomposition induit alors une simplification

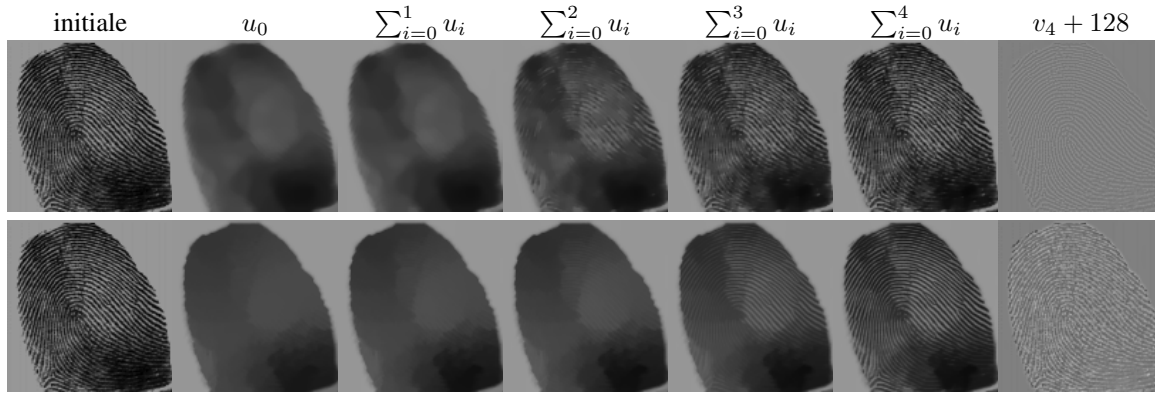


FIGURE 1 – Représentation hiérarchique d’une image. La première colonne désigne l’image originale. Les colonnes successives désignent le niveau de décomposition. La première ligne correspond à la représentation de [7]. La seconde ligne est obtenue avec un graphe des 25 plus proches voisins. Voir le texte pour plus de détails.

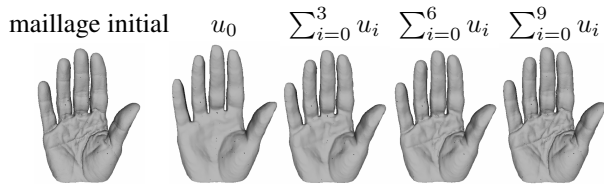


FIGURE 2 – Représentation hiérarchique d’un maillage triangulaire.

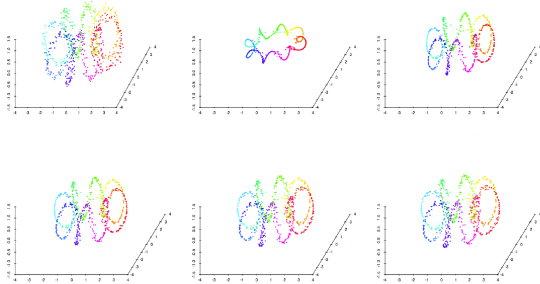


FIGURE 3 – Décomposition d’une hélice toroidale bruitée. $\lambda_0 = 0.1$. De haut en bas, et de gauche à droite : nuage de points initial, niveaux 1, 2, 3, 4 de la décomposition hiérarchique.

très forte de la structure de l’hélice. La version débruitée de l’hélice apparaît alors suite aux décompositions successives. Cet exemple démontre l’applicabilité de notre approche à des données non organisées.

8 Conclusion

Nous avons proposé dans cet article une approche permettant la représentation multi-échelle de données organisées sous forme de graphes. Notre approche est basée sur les outils de la régularisation discrète. Nous avons adapté au cas des graphes un algorithme efficace permettant d’obtenir la solution de la

régularisation. Enfin, nous avons abordé le problème du choix des paramètres et la construction de graphes pondérés à partir des données.

Références

- [1] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4 :490–530, 2005.
- [2] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2), 2004.
- [3] T. F. Chan, S. Osher, and J. Shen. The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10 :231 – 241, 2001.
- [4] A. Elmoataz, O. Lézoray, and S. Boughleux. Nonlocal discrete regularization on weighted graphs : A framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7), 2008.
- [5] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*. University Lecture Series. AMS, 2001.
- [6] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4) :259–268, 1992.
- [7] E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using hierarchical (BV, L2) decompositions. *Multiscale Modeling and Simulation*, 2(4) :554–579, 2004.