



**HAL**  
open science

# Décompositions multi-échelles de données définies sur des graphes

Moncef Hidane

► **To cite this version:**

Moncef Hidane. Décompositions multi-échelles de données définies sur des graphes. Traitement des images [eess.IV]. Université de Caen, 2013. Français. NNT: . tel-01081399

**HAL Id: tel-01081399**

**<https://hal.science/tel-01081399>**

Submitted on 7 Nov 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.

# UNIVERSITÉ de CAEN BASSE-NORMANDIE

U.F.R. Sciences

ÉCOLE DOCTORALE SIMEM

## THÈSE

*Présentée par*

**M. Moncef Hidane**

*Et soutenue*

Le 22 Mars 2013

*En vue de l'obtention du*

**DOCTORAT de L'UNIVERSITÉ de CAEN**

Spécialité informatique

*Arrêté du 07 août 2006*

# Décompositions multi-échelles de données définies sur des graphes.

Devant le jury composé de

Laure BLANC-FERAUD	Directrice de recherche CNRS	I3S, Sophia-Antipolis	(rapporteur)
François MALGOUYRES	Professeur	Université Paul Sabatier, Toulouse	(rapporteur)
Antonin CHAMBOLLE	Directeur de recherche CNRS	CMAP, École Polytechnique	(examinateur)
Jalal FADILI	Professeur	ENSICAEN	(examinateur)
Abderrahim ELMOATAZ	Professeur	Université de Caen	(examinateur)
Olivier LÉZORAY	Professeur	Université de Caen	(directeur de thèse)



# Remerciements

En tout premier lieu, j'aimerais remercier très chaleureusement Olivier Lézoray pour ces quelques années passées sous sa direction. Olivier m'a permis par sa patience, ses encouragements et son aide constante d'éviter les mauvaises passes inhérentes à tout travail doctoral.

Je tiens également à remercier Abderrahim Elmoataz qui a accompagné mon travail depuis le début. Ses conseils et ses encouragements m'ont été précieux.

Je voudrais ensuite remercier les membres du jury. Tout d'abord, Laure Blanc-Féraud et François Malgouyres m'ont fait un grand honneur en acceptant de rapporter ma thèse. Je les remercie pour le temps qu'ils ont accordé à la lecture de mon travail. Ils ont, tous deux, contribué à améliorer la qualité du manuscrit final.

Je remercie sincèrement Jalal Fadili d'avoir accepté de présider mon jury. Sa bonne humeur a donné le ton à une soutenance chaleureuse et à une discussion scientifique riche. Je le remercie également pour son aide tout au long de mes années de thèse. C'est à la suite de discussions avec lui que je me suis intéressé à l'optimisation convexe.

Antonin Chambolle a accepté de faire partie de mon jury. Ses travaux m'ont beaucoup accompagné durant mes années de thèse. Je le remercie pour son intérêt ainsi que pour toutes les questions très constructives pendant la soutenance.

Je remercie tous les membres du GREYC, permanents, doctorants et l'équipe administrative pour leur accueil, leur aide et leur amitié. Je voudrais remercier plus particulièrement Matthieu pour son aide durant la rédaction ainsi que Benoît et Xavier pour leur relecture attentive.

Finalement, merci à ma famille et à mes proches pour leur soutien et leur patience. J'aimerais remercier tout particulièrement mon père pour sa relecture attentive du manuscrit.

*Moncef*





# Table des matières

<b>1</b>	<b>Introduction</b>	<b>9</b>
<b>2</b>	<b>Graphes et signaux</b>	<b>15</b>
2.1	Introduction . . . . .	15
2.2	Graphes pondérés et matrices associées . . . . .	16
2.2.1	Premières définitions . . . . .	16
2.2.2	Coloration . . . . .	17
2.2.3	Coupe maximale . . . . .	18
2.2.4	Coupes en optimisation combinatoire . . . . .	18
2.2.5	Matrices associées à un graphe pondéré . . . . .	19
2.3	Construction d'un graphe pondéré à partir d'un ensemble de données . . . . .	22
2.3.1	Construction de la topologie . . . . .	22
2.3.2	Fonctions de similarité . . . . .	26
2.3.3	Construction simultanée de la topologie et des poids . . . . .	27
2.4	Signaux sur graphes . . . . .	28
2.4.1	Espaces $\mathcal{H}(V)$ et $\mathcal{H}(E)$ . . . . .	28
2.4.2	Opérateurs de différence . . . . .	29
2.4.3	Mesures de régularité . . . . .	31
2.5	Les images comme des fonctions sur des graphes . . . . .	33
2.5.1	Principe . . . . .	33
2.5.2	Vecteurs de caractéristiques . . . . .	34
2.5.3	Débruitage d'images par filtres de voisinage . . . . .	37
2.5.4	Lien avec les graphes . . . . .	38
2.6	Conclusion . . . . .	40

<b>3</b>	<b>Approches variationnelles pour le débruitage de signaux sur graphes</b>	<b>41</b>
3.1	Introduction . . . . .	41
3.2	Éléments d’analyse et d’optimisation convexes . . . . .	42
3.2.1	Espace $\Gamma_0(\mathcal{H})$ . . . . .	42
3.2.2	Fonction conjuguée . . . . .	43
3.2.3	Opérateur proximal . . . . .	44
3.2.4	Algorithmes proximaux . . . . .	45
3.3	Le cadre bayésien pour le débruitage d’images . . . . .	46
3.3.1	Débruitage par minimisation d’énergie . . . . .	47
3.3.2	Le cadre bayésien . . . . .	48
3.3.3	A priori classiques en traitement de l’image . . . . .	49
3.4	Approches variationnelles pour le débruitage de signaux sur graphes . . . . .	55
3.4.1	A priori quadratique . . . . .	57
3.4.2	A priori basé sur la variation totale . . . . .	60
3.5	Expériences . . . . .	67
3.5.1	Images . . . . .	67
3.5.2	Maillages . . . . .	68
3.6	Conclusion . . . . .	72
<b>4</b>	<b>Décompositions multi-échelles hiérarchiques de signaux sur graphes</b>	<b>73</b>
4.1	Introduction . . . . .	73
4.2	Décompositions $u + v$ . . . . .	74
4.3	Décompositions multi-échelles . . . . .	80
4.3.1	Définition et motivations . . . . .	80
4.3.2	Décompositions basées sur le principe de la pyramide laplacienne . . . . .	81
4.3.3	L’approche hiérarchique de Tadmor, Nezzar et Vese pour la décomposition d’images . . . . .	82
4.4	Décomposition multi-échelle quadratique . . . . .	84
4.4.1	Principe . . . . .	84
4.4.2	Convergence et choix des paramètres . . . . .	85
4.5	Décomposition basée sur la TV sur graphes . . . . .	87
4.5.1	Principe . . . . .	87

4.5.2	Convergence . . . . .	88
4.5.3	Choix de la suite d'échelle $\lambda_i$ . . . . .	90
4.6	Exemples de décompositions obtenues avec $J_{1,2}^w$ . . . . .	93
4.6.1	Images . . . . .	94
4.6.2	Maillages et nuages de points 3D . . . . .	95
4.7	Applications en rehaussement de détails pour les images et les maillages . . . . .	98
4.7.1	Comparaison avec l'état de l'art . . . . .	99
4.7.2	Rehaussement de contours . . . . .	103
4.7.3	Tone mapping . . . . .	107
4.7.4	Retouche virtuelle de portraits . . . . .	108
4.7.5	Rehaussement de détails sur des maillages . . . . .	110
4.8	Conclusion . . . . .	113
<b>5</b>	<b>Transformées multi-échelles de signaux sur graphes</b>	<b>119</b>
5.1	Introduction . . . . .	119
5.2	Transformées multi-échelles pour les signaux et les images . . .	120
5.2.1	Difficultés d'adaptation pour les signaux sur graphes . .	122
5.3	Transformées multi-échelles définies dans le domaine spectral . .	123
5.3.1	Ondelettes de diffusion . . . . .	123
5.3.2	Ondelettes spectrales . . . . .	125
5.4	Transformée de Haar sur une hiérarchie . . . . .	127
5.4.1	Transformée de Haar sur une hiérarchie . . . . .	127
5.4.2	Construction adaptative d'une hiérarchie à partir d'un graphe pondéré . . . . .	130
5.4.3	Expériences . . . . .	133
5.5	Transformées multi-échelles par lifting sur graphes . . . . .	141
5.5.1	Schéma du lifting pour les signaux 1D . . . . .	141
5.5.2	Lifting sur graphes . . . . .	144
5.5.3	Transformée multi-échelle par suite de liftings élémentaires	146
5.5.4	Expériences . . . . .	151
5.6	Conclusion . . . . .	154
<b>6</b>	<b>Conclusion et perspectives</b>	<b>159</b>
	<b>Liste des figures</b>	<b>163</b>

<b>Liste des algorithmes</b>	<b>169</b>
<b>Liste des publications</b>	<b>169</b>
<b>Références</b>	<b>173</b>

# Chapitre 1

## Introduction

Le développement des techniques d'acquisition, de simulation et de collecte, dans tous les champs de l'activité humaine, engendre un nombre croissant de données. Formellement, un ensemble de données correspond à un ensemble  $V = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , chaque donnée élémentaire  $\mathbf{x}_i$  appartenant à l'espace  $\mathbb{R}^p$ ,  $p \geq 1$ .

L'explosion de ces masses de données s'est accompagnée de la volonté d'une exploitation systématique et maximale. Bien que s'agissant de données de natures différentes, issues de domaines variés, cette exploitation vise le plus souvent à des objectifs communs : débruitage, partitionnement, apprentissage, organisation, catégorisation et plus généralement fouille. Formellement, cette exploitation prend la forme de l'étude de fonctions définies sur l'ensemble discret  $V$ .

Les traitement du signal et de l'image peuvent être vus comme des cas particuliers du cadre précédent. Un signal 1D, à temps discret, correspond à une fonction définie sur l'ensemble de données  $\{t_1, \dots, t_N\}$ , chaque  $t_i$  désignant un instant. De même, une image 2D discrète, correspond à une fonction définie sur l'ensemble de données  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , chaque  $\mathbf{x}_i$  désignant les coordonnées d'un pixel de l'image. Ces deux exemples ont trois caractéristiques importantes. La première concerne le volume :  $N$  est souvent très grand. La deuxième caractéristique concerne la dimension :  $p$  est faible (1 pour les signaux, 2 pour les images). Enfin, la troisième concerne l'échantillonnage : le plus souvent, ce dernier est régulier. Ces trois caractéristiques ont un impact direct sur les techniques employées dans le traitement et l'analyse du signal et de l'image.

Une caractéristique commune aux différents ensembles de données collectées depuis quelques décennies est qu'ils sont de *grand volume* ( $N$  très grand) et de *grande dimension* ( $p$  très grand). Bien que de *dimensionnalité apparente* élevée, plusieurs ensembles de données, obtenus dans divers domaines d'application,

ont en réalité une faible *dimension intrinsèque*. Considérons par exemple une série d'images de tailles  $n \times n$  capturées par une caméra en mouvement. Cet ensemble de données a une dimension apparente égale à  $n^2$  alors que la dimension intrinsèque est égale à un. Cette notion de dimension intrinsèque peut être formalisée à l'aide d'une mesure de probabilité sur une sous-variété  $\mathcal{M}$  de  $\mathbb{R}^p$ . La dimension intrinsèque correspond alors à la dimension de  $\mathcal{M}$ . Ceci a notamment conduit au développement de nombreuses méthodes pour la réduction de dimension.

En pratique, la paramétrisation d'une sous-variété à partir d'un nombre fini d'échantillons s'avère difficile. Au sein de la communauté de l'apprentissage statistique, le modèle de sous-variété est remplacé par un *graphe pondéré* dont les sommets correspondent aux instances  $\mathbf{x}_i$  et les poids à une mesure de similarité entre les instances. On est ainsi amené à étudier des fonctions définies sur des graphes pondérés, c'est-à-dire à prendre en compte la relation d'adjacence en vue de l'étude de ces fonctions. Ce type d'approches a été très populaire, notamment en apprentissage semi-supervisé, en particulier pour la classification.

Au-delà du cadre de l'apprentissage, le modèle de graphe pondéré peut également être considéré pour des données en faible dimension échantillonnées de manière irrégulière. Les approches basées sur des arbres couvrant minimaux ou des triangulations de Delaunay sont maintenant d'un usage fréquent. Même pour des données de faible dimension échantillonnées régulièrement, on rencontre des modélisations sous la forme de graphes pondérés. En analyse de l'image, la modélisation par champs de Markov a permis de traiter un large spectre d'applications, du débruitage à la segmentation. Depuis quelques années, les méthodes dites non locales ont constitué une direction de recherche très importante. Ces méthodes ont pour essence la remarque que les images naturelles sont constituées de beaucoup de petites parties, appelées *patches*, similaires. La modélisation des pixels d'une image comme sommets d'un graphe pondéré permet de tenir compte de cet aspect. L'algorithme des moyennes non locales, qui a en particulier introduit une nouvelle mesure de similarité entre les pixels d'une même image, a beaucoup popularisé ce type d'approches.

Notons enfin que pour beaucoup de types de données, la notion de graphe est intrinsèque. C'est le cas par exemple pour des données collectées sur des réseaux sociaux, biologiques et de communication. Un autre exemple concerne la modélisation 3D à l'aide de maillages polygonaux.

La discussion précédente montre que les graphes pondérés constituent un puissant outil de modélisation pour des types très hétérogènes de données. Le concept de fonction, ou de signal, sur un graphe pondéré apparaît comme une notion unifiante, comme en attestent beaucoup d'articles récents ainsi que la

---

tenue de sessions spéciales sur ce thème dans des conférences nationales et internationales.

Longtemps, les approches basées sur les équations aux dérivées partielles (EDP) et sur l'analyse harmonique ont été incontournables en traitement du signal et de l'image. Au coeur de ces méthodes, se trouvent de nouvelles *représentations* des signaux et des images. La notion d'*échelle* y joue un rôle important : on souhaite obtenir une description d'un même signal à différents niveaux de détail. La théorie espace-échelle (scale space en anglais) permet d'obtenir de telles représentations. Dans sa version discrète et linéaire, ces représentations sont obtenues en appliquant des filtres gaussiens de largeurs différentes. Dans les approches multi-échelles basées sur les ondelettes, une nouvelle représentation est obtenue en changeant de base. Dans tous les cas, ces représentations prennent la forme de *décompositions multi-échelles* : un signal ou une image est représenté comme la somme d'un signal présentant très peu de détails et d'une série de signaux permettant de reconstruire les détails initiaux.

L'approche variationnelle sur graphes basée sur des a priori de régularité, ainsi que l'approche EDP sur graphes ont été développées récemment plus en détail dans [ELB08, BEM09, LTE10, ZS05]. Les approches de type analyse harmonique commencent également à émerger via quelques travaux récents.

## Contributions principales

Nous développons dans ce manuscrit trois approches permettant de construire des décompositions multi-échelles de signaux définis sur des graphes pondérés généraux. Il est important de noter que nos contributions concernent tant les méthodes proposées et leur variété que la diversité des données auxquelles elles s'appliquent.

La première approche est basée sur une approche variationnelle hiérarchique inspirée par Tadmor et al. [TNV04]. Cette approche généralise la décomposition mono-échelle structure-texture, proposée initialement pour les images. Nous l'adaptions à des signaux définis sur des graphes généraux. Cette approche est implémentée à l'aide d'un procédé de régularisation discrète. Nous proposons deux versions basées sur des a priori différents : un a priori quadratique relié à la matrice laplacienne associée au graphe, et un a priori de type variation totale. L'étude de la convergence est effectuée et le choix des paramètres discuté dans chaque cas. Nous détaillons l'application des décompositions obtenues au rehaussement de détails dans les images et les modèles 3D.

La deuxième approche que nous développons est une analyse multirésolution de l'espace des signaux sur un graphe donné. Cette construction, inspirée



par [GNC10], repose sur l'organisation du graphe sous la forme d'une hiérarchie de partitions. Nous développons un algorithme permettant la construction adaptative de telles hiérarchies.

Enfin, dans la troisième approche, nous adaptons l'algorithme du lifting, permettant de factoriser toutes les transformées en ondelettes dans le cadre classique, à des signaux sur graphes. Bien que cette adaptation soit immédiate, son application pose divers problèmes pratiques. Nous optons pour une approche basée sur l'itération d'étapes de liftings élémentaires où un unique coefficient de détail est calculé à chaque fois. Nous proposons enfin une version multicycle de cette transformée.

## **Plan du manuscrit**

La suite de ce manuscrit est organisée de la manière suivante.

### **Chapitre 2**

Nous introduisons dans ce chapitre le vocabulaire et les notions, utiles pour la suite, concernant les graphes pondérés. Les matrices associées sont définies et certaines de leurs propriétés énoncées. Nous nous intéressons ensuite aux approches de construction de graphes pondérés à partir d'un ensemble de données. Les graphes que l'on y obtient sont des graphes de similarité, c'est-à-dire que le poids affecté à une arête reliant deux sommets relève de la similarité accordé à ces sommets. Outre les constructions classiques, nous rappelons également certaines constructions récentes principalement issues de la communauté d'apprentissage. Nous définissons le concept de signal sur graphe et rappelons une famille d'opérateurs permettant de définir des mesures relatives de régularité. La suite du chapitre traite de la modélisation des images numériques comme des signaux sur des graphes. La spécificité de la construction des graphes à partir d'images y est présentée et le lien avec les filtres de voisinage établi.

### **Chapitre 3**

Nous abordons dans ce chapitre des méthodes variationnelles pour le débruitage de signaux définis sur l'ensemble des sommets d'un graphe pondéré général. Nous commençons par rappeler quelques éléments d'analyse et d'optimisation convexes utiles pour la suite, le cadre de l'analyse bayésienne et les a priori classiques en traitement de l'image. Ces a priori sont ensuite transposés sur des graphes, et donnent lieu à deux fonctions d'énergie : l'une,

---

quadratique, basée sur les matrices laplaciennes ; l'autre, non différentiable, de type variation totale (TV). Nous caractérisons la solution du problème basé sur la pénalisation TV sur graphes. Cette caractérisation fait apparaître un équivalent de l'espace  $G$  introduit par Meyer dans [Mey01]. Nous introduisons cet espace et le caractérisons à son tour. Enfin, nous détaillons l'application de trois algorithmes proximaux récents à ce problème. Nous effectuons enfin des tests de débruitage sur des images et des maillages.

## Chapitre 4

Nous introduisons dans ce chapitre deux décompositions multi-échelles de signaux définis sur l'ensemble des sommets d'un graphe pondéré général. Ces décompositions sont basées sur un principe de régularisation itérative, introduit dans [TNV04], et que nous adaptons et généralisons à des signaux sur graphes. Les propriétés de chacune des décompositions introduites sont étudiées. En particulier, la convergence des représentations obtenues est établie et le choix des paramètres d'échelle discuté. Pour la décomposition basée sur l'a priori TV, le choix du premier paramètre peut être caractérisé. Cette caractérisation fait apparaître la norme  $G^w$  définie dans le chapitre précédent. Nous proposons de la calculer à l'aide de l'algorithme primal-dual de Chambolle et Pock [CP11a]. Pour la décomposition basée sur l'a priori TV, nous énonçons également un résultat de *décomposition d'énergie*. Enfin, la dernière section est consacrée aux exemples et applications. Nous mettrons l'accent, en particulier, sur l'application de notre méthode au rehaussement de détails pour les images numériques et les maillages. Pour ces derniers, nous considérons aussi bien le rehaussement de détails *géométriques* que celui de détails *photométriques*.

## Chapitre 5

Plusieurs constructions de transformées multi-échelles sur des graphes ont été proposées récemment. Nous examinerons d'abord deux exemples : la transformée basée sur les ondelettes de diffusion introduite par Coifman et Maggioni dans [CM06] et celle basée sur les ondelettes spectrales introduite par Hammond et al. dans [HVG11]. Nous abordons la construction d'ondelettes de Haar sur une hiérarchie et présentons un algorithme permettant la construction adaptative d'une hiérarchie de partitions à partir d'un graphe pondéré. Des exemples de décompositions viennent illustrer cette approche pour les images et les bases de données d'images. Nous développons ensuite une transformée basée sur le lifting. Nous détaillons l'adaptation de cet algorithme aux signaux sur graphes et discutons les difficultés d'une telle adaptation. La difficulté principale concerne l'étape de partitionnement des sommets. Nous optons

ici pour une approche inspirée par [JNS09], où un unique coefficient de détail est calculé à chaque étape de lifting. Nous détaillons la sélection des sommets, l'effet sur les fonctions de base, ainsi que l'extension de la transformée à plusieurs cycles. Nous calculons enfin, sur des images, les erreurs d'approximation non linéaire à  $M$  termes, obtenues avec différentes configurations de graphes et comparons les résultats avec ceux obtenus à l'aide de trois transformées en ondelettes classiques.

## **Chapitre 6**

Ce dernier chapitre présente plusieurs remarques finales et perspectives concernant les travaux abordés dans ce manuscrit.

# Chapitre 2

## Graphes et signaux

### 2.1 Introduction

Nous introduisons dans ce chapitre le vocabulaire et les notions dont nous ferons usage dans toute la suite de ce manuscrit.

Ce chapitre est organisé de la manière suivante. Dans la section 2.2, nous introduisons la notion de graphe pondéré ainsi que le langage de la théorie des graphes dont nous nous servons. L'étude des propriétés d'un graphe  $G$  peut se faire en examinant les valeurs et vecteurs propres de certaines matrices associées. Il s'agit là de la théorie spectrale des graphes [Chu97, Moh91, Moh97, Spi07]. Les matrices laplaciennes y jouent un rôle important. Nous définissons ces matrices et énonçons certaines propriétés utiles pour la suite du document. Les développements ultérieurs feront usage principalement de la version combinatoire du laplacien. Une autre matrice importante est la matrice de transition associée à une chaîne de Markov dont les états sont les sommets du graphe. La section 2.3 est dédiée aux approches de construction de graphes pondérés à partir d'un ensemble de données. Les graphes que l'on y obtient sont des graphes de similarité, c'est-à-dire que le poids affecté à une arête reliant deux sommets relève de la similarité que l'on accorde à ces sommets. Outre les constructions classiques (graphes des plus proches voisins et  $\epsilon$  graphes), nous rappelons également certaines constructions récentes principalement issues de la communauté d'apprentissage. Dans la section 2.4, nous définissons le concept de signal sur graphe. Il s'agit de fonctions définies sur l'ensemble des sommets d'un graphe pondéré général. L'étude de ces signaux repose sur une famille d'opérateurs discrets que nous donnons ici. Ces opérateurs permettent de définir des mesures relatives de régularité. Nous nous servirons de ces mesures pour proposer des décompositions mono-échelles de signaux sur graphes dans le chapitre 3 et multi-échelles dans le chapitre 4. Enfin, la section

2.5 traite de la modélisation des images numériques comme des signaux sur des graphes. Les spécificités de la construction des graphes à partir d'images y sont présentées. Au-delà de la construction des graphes, cette section illustre un aspect important de la modélisation : l'utilisation des attributs de l'image pour définir le graphe permet de mieux tenir compte des structures. Une conclusion de ce chapitre est présentée dans la section 2.6.

## 2.2 Graphes pondérés et matrices associées

Nous introduisons dans cette section, d'une part, les concepts de théorie des graphes sur lesquelles nous nous appuyerons dans la suite de ce manuscrit, et d'autre part, les matrices associées à des graphes pondérés. Il s'agit des matrices d'*adjacence*, des matrices *laplaciennes*, et de la matrice de *transition* associée à une marche aléatoire sur un graphe.

### 2.2.1 Premières définitions

Soit  $V$  un ensemble fini non vide. On appelle *graphe* sur  $V$  tout couple  $G = (V, E)$  où  $E \subset V \times V$ . Les éléments de  $V$  sont appelés des *sommets* et ceux de  $E$  des *arêtes*. Notons que la définition que nous donnons ici exclut la notion d'arêtes multiples. Dans la suite,  $V(G)$  désignera l'ensemble des sommets d'un graphe  $G$  et  $E(G)$  l'ensemble de ses arêtes.

- On appelle *ordre* de  $G$  le cardinal de  $V(G)$ . Si  $e = (u, v) \in E$ , on dira que
- $e$  est *incidente* à  $u$  et à  $v$  ;
  - $u$  et  $v$  sont les *extrémités* de  $e$  ;
  - $u$  et  $v$  sont *voisins* (ou *adjacents*), et on notera  $u \sim v$ .
  - $e$  est une *boucle* si  $u = v$ .

Un sommet  $u \in V(G)$  est dit *isolé* s'il n'a aucun voisin.  $G$  est dit *non orienté* si  $(u, v) \in E(G)$  si et seulement si  $(v, u) \in E(G)$ .

Les graphes que nous considérons dans toute la suite de ce manuscrit sont supposés d'ordre fini, sans aucun sommet isolé, sans boucles et non orientés, sauf mention contraire.

Si  $u \in V(G)$ , on notera  $d(u)$  le nombre de ses voisins. On dira que  $d(u)$  est le *degré simple* de  $u$ . Le graphe  $G$  est dit  *$k$ -régulier* si  $d(u) = k$  pour tout  $u \in V(G)$ .

Deux sommets  $u, v \in V(G)$  sont dits *connectés* s'il existe un *chemin* les reliant, c'est-à-dire s'il existe  $u_0, \dots, u_n \in V(G)$  tels que

- (i)  $u_0 = u$  ;
- (ii)  $u_n = v$  ;

(iii)  $(\forall i \in \{0, \dots, n-1\}) \quad u_i \sim u_{i+1}$ .

Une partie  $W \subset V(G)$  est dite *connectée* si toute paire de sommets dans  $W$  est connectée par un chemin inclus dans  $W$ . Une *composante connexe* est une partie connectée maximale.  $G$  est *connexe* s'il contient une unique composante connexe. Un exemple de graphe connexe est le *graphe complet*, pour lequel  $(u, v) \in E(G)$  pour tous  $u, v \in V(G)$ .

Soit  $H$  un graphe. On dit que  $H$  est un *sous-graphe* de  $G$  si :

- (i)  $V(H) \subset V(G)$ ;
- (ii)  $E(H) \subset E(G)$ .

On dit d'un sous-graphe  $H$  de  $G$  qu'il est *couvrant* si  $V(H) = V(G)$ .

Un graphe  $G$  muni d'une fonction symétrique  $w : E(G) \rightarrow \mathbb{R}$  s'appelle un *graphe pondéré*. Dans toute la suite de ce document, la fonction de pondération  $w$  est supposée à valeurs dans  $\mathbb{R}_*^+$ . Pour  $(u, v) \in E$ , la quantité  $w(u, v)$  est interprétée comme une mesure de similarité entre  $u$  et  $v$ .

Soit  $u \in V(G)$ . On appelle *degré pondéré* de  $u$  la quantité

$$d^w(u) = \sum_{v \sim u} w(u, v).$$

Si  $H = (V(H), E(H), w_H)$  désigne un graphe pondéré, on dit que  $H$  est un *sous-graphe pondéré* de  $G$  si

- (i)  $(V(H), E(H))$  est un sous-graphe de  $(V, E)$ ;
- (ii)  $w|_{E(H)} = w_H$ .

## 2.2.2 Coloration

Soit  $G$  un graphe et  $k \geq 2$  un entier. Une *k-coloration* de  $G$  est une fonction  $c : V \rightarrow \{1, \dots, k\}$  telle que

$$(\forall e = (u, v) \in E(G)) \quad c(u) \neq c(v).$$

On dit que  $G$  est dit *k-coloriable* s'il admet une *k-coloration*. Le plus petit  $k$  tel que  $G$  soit *k-coloriable* s'appelle le *nombre chromatique* de  $G$  et est noté  $\chi(G)$ . Un graphe 2-coloriable est dit *biparti*.

Tout graphe admet un sous-graphe couvrant biparti. Le problème du *sous-graphe couvrant biparti maximal* consiste à en sélectionner un ayant le plus grand nombre d'arêtes. Si  $G$  est pondéré, on cherche alors un sous-graphe biparti couvrant maximisant le critère suivant :

$$\sum_{\substack{u \sim v \\ u \in A, v \in B}} w(u, v), \tag{2.1}$$

où  $A$  et  $B$  désignent les deux parties du sous-graphe. Ce problème est équivalent à celui de la recherche d'une coupe maximale que nous décrivons maintenant.

### 2.2.3 Coupe maximale

On est souvent amené à partitionner l'ensemble des sommets  $V(G)$  d'un graphe pondéré  $G$  en deux parties disjointes  $A$  et  $B$ . De telles partitions s'appellent des *coupes*. On associe à une coupe  $(A, B)$  la fonction d'utilité suivante :

$$\sum_{\substack{u \sim v \\ u \in A, v \in B}} w(u, v). \quad (2.2)$$

Le problème de la coupe maximale consiste alors à déterminer une coupe maximisant l'utilité 2.2. Ce problème, équivalent au problème du sous-graphe couvrant biparti maximal, est NP-complet. Le lecteur intéressé par les algorithmes d'approximation de ce problème pourra consulter la section 4 [Moh97].

Le problème de la coupe maximale sera rencontré dans le chapitre 5 de ce manuscrit dans le contexte des transformées multi-échelles de signaux sur graphes. Nous y verrons en particulier que l'adaptation de la transformée de lifting [Swe96] à des signaux sur graphes nécessite un critère de partitionnement des sommets.

### 2.2.4 Coupes en optimisation combinatoire

En optimisation combinatoire, la notion de coupe est différente de celle que nous venons de rappeler : elle s'applique à des graphes *orientés* ayant des *noeuds remarquables*. Plus précisément, soit  $G$  un graphe pondéré *orienté*. On suppose que  $V(G)$  contient deux sommets particuliers : un sommet noté  $s$  et appelé *source* ; un sommet noté  $t$  et appelé *puits*. Une *s/t coupe* est définie comme étant une partition de  $V(G)$  en deux ensembles  $\mathcal{S}$  et  $\mathcal{T}$  tels que  $s \in \mathcal{S}$  et  $t \in \mathcal{T}$  (c.f. figure<sup>1</sup> 2.1). On associe à une telle coupe le coût suivant :

$$C(\mathcal{S}, \mathcal{T}) = \sum_{(u,v) \in \mathcal{S} \times \mathcal{T}} w(u, v). \quad (2.3)$$

Le problème de la s/t coupe minimale est celui de la recherche d'une coupe minimisant le coût (2.3). Ce problème peut être résolu en temps polynomial. L'article [BK04] présente, dans le contexte de la vision par ordinateur, une comparaison des performances de plusieurs algorithmes et en propose un nouveau, disponible à l'adresse <http://pub.ist.ac.at/~vnk/software.html>.

Greig et al. [GPS89] ont été les premiers à faire le lien entre la minimisation d'une énergie discrète et la recherche d'une coupe minimale.

1. Cette figure est tirée de l'article [BK04]

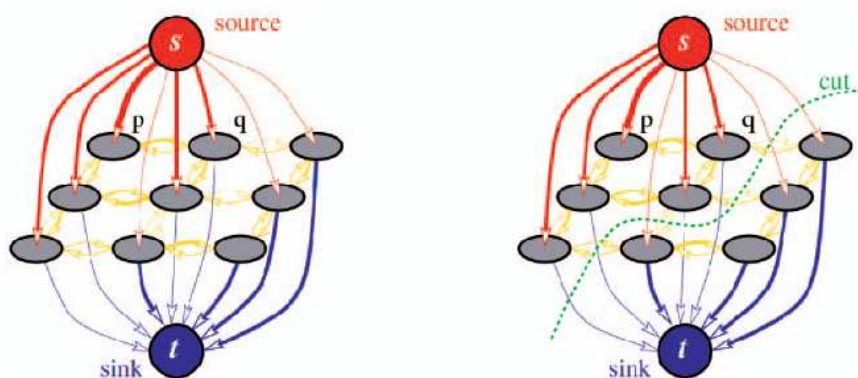


FIGURE 2.1 – s/t coupe d'un graphe pondéré orienté.

L'énergie considérée dans [GPS89], ainsi que dans plusieurs autres travaux ([BVZ01, BK04, KZ04]), est de la forme

$$E(l) = \sum_{p \in \mathcal{P}} D_p(l_p) + \sum_{p, q \in \mathcal{N}} V_{p, q}(l_p, l_q) \quad (2.4)$$

où  $\mathcal{P}$  est l'ensemble des pixels d'une image,  $l = (l_p)_{p \in \mathcal{P}}$  est une *labellisation* de  $\mathcal{P}$ , chaque  $l_p$  étant dans un ensemble fini  $\{1, \dots, Q\}$ , et  $\mathcal{N}$  désignant un système de voisinage. Nous reviendrons sur ce lien dans le chapitre 3.

### 2.2.5 Matrices associées à un graphe pondéré

Dans toute cette section,  $G$  désigne un graphe pondéré d'ordre  $N$ . On pose  $V(G) = \{u_1, \dots, u_N\}$ .

#### Matrice d'adjacence

On appelle *matrice d'adjacence pondérée* associée à  $G$  la matrice  $\mathbf{W}(G) \in \mathbb{R}^{N \times N}$  définie par

$$\mathbf{w}_{i, j}(G) = \begin{cases} w(u_i, u_j), & \text{si } u_i \sim u_j, \\ 0, & \text{sinon.} \end{cases} \quad (2.5)$$

La matrice  $\mathbf{W}(G)$  caractérise  $E(G)$ . En effet,  $u_i \sim u_j$  si et seulement si  $\mathbf{w}_{i, j}(G) \neq 0$  (la fonction de poids  $w$  est supposée à valeurs dans  $\mathbb{R}_*^+$ ). Ainsi, un graphe pondéré correspond à la donnée d'un ensemble  $V$  et d'une matrice d'adjacence sur  $V$ .



## Matrices laplaciennes

On appelle *matrice laplacienne* (non normalisée) associée à  $G$  la matrice  $\mathbf{L}(G) \in \mathbb{R}^{N \times N}$  définie par

$$\mathbf{L}_{i,j}(G) = \begin{cases} d^w(u_i), & \text{si } u_i = u_j, \\ -w(u_i, u_j), & \text{si } u_i \sim u_j, \\ 0, & \text{sinon.} \end{cases} \quad (2.6)$$

On appelle *matrice laplacienne normalisée* associée à  $G$  la matrice  $\tilde{\mathbf{L}}(G) \in \mathbb{R}^{N \times N}$  définie par

$$\tilde{\mathbf{L}}_{i,j}(G) = \begin{cases} 1, & \text{si } i = j, \\ -\frac{w(u_i, u_j)}{\sqrt{d^w(u_i)}\sqrt{d^w(u_j)}}, & \text{si } u_i \sim u_j, \\ 0, & \text{sinon.} \end{cases} \quad (2.7)$$

Notons  $\mathbf{D} \in \mathbb{R}^{N \times N}$  la matrice diagonale des degrés :

$$\mathbf{D}_{i,j} = \begin{cases} d^w(u_i), & \text{si } i = j, \\ 0, & \text{sinon.} \end{cases} \quad (2.8)$$

Les matrices laplaciennes s'expriment en fonction de  $\mathbf{D}$  :

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \quad (2.9)$$

et

$$\tilde{\mathbf{L}} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}. \quad (2.10)$$

Les applications linéaires de matrices  $\mathbf{L}$  et  $\tilde{\mathbf{L}}$  sont données par

$$(\mathbf{L}\mathbf{f})_i = \sum_{j=1}^N w_{i,j}(f_i - f_j), \quad (2.11)$$

et

$$(\tilde{\mathbf{L}}\mathbf{f})_i = \frac{1}{\sqrt{d_{i,i}}} \sum_{j=1}^N w_{i,j} \left( \frac{f_i}{\sqrt{d_{i,i}}} - \frac{f_j}{\sqrt{d_{j,j}}} \right). \quad (2.12)$$

Les formes quadratiques associées aux matrices  $\mathbf{L}$  et  $\tilde{\mathbf{L}}$  sont données par

$$\langle \mathbf{L}\mathbf{f}, \mathbf{f} \rangle = \frac{1}{2} \sum_{i,j=1}^N w_{i,j}(f_i - f_j)^2, \quad (2.13)$$

$$\langle \tilde{\mathbf{L}}\mathbf{f}, \mathbf{f} \rangle = \frac{1}{2} \sum_{i,j=1}^N w_{i,j} \left( \frac{f_i}{\sqrt{d_{i,i}}} - \frac{f_j}{\sqrt{d_{j,j}}} \right)^2, \quad (2.14)$$

où  $\langle \cdot, \cdot \rangle$  désigne le produit scalaire canonique sur  $\mathbb{R}^N$ .

Les équations (2.13) et (2.14) montrent que les matrices symétriques  $\mathbf{L}$  et  $\tilde{\mathbf{L}}$  sont semi-définies positives. Elles ont donc  $N$  valeurs propres réelles positives (comptées avec leur multiplicité) chacune.

Remarquons à ce stade que ces valeurs propres ne dépendent pas du choix de la numérotation des sommets. En effet, si  $\sigma$  désigne une permutation de  $V(G)$  et  $\mathbf{L}'$  la matrice laplacienne relativement à la nouvelle numérotation, alors  $\mathbf{L}' = \mathbf{P}^{-1}\mathbf{L}\mathbf{P}$ , où  $\mathbf{P}$  désigne une matrice de permutation. Nous noterons dans la suite  $\alpha_0 \leq \alpha_1 \leq \dots \leq \alpha_{N-1}$  les valeurs propres de  $\mathbf{L}$  et  $\tilde{\alpha}_0 \leq \tilde{\alpha}_1 \leq \dots \leq \tilde{\alpha}_{N-1}$  celles de  $\tilde{\mathbf{L}}$ .

Notons que  $\alpha_0 = \tilde{\alpha}_0 = 0$ . Les multiplicités de  $\alpha_0$  et de  $\tilde{\alpha}_0$  sont égales aux nombres de composantes connexes de  $G$  (Propositions 2 et 4 de [vL07]). Rappelons enfin que du fait de la normalisation, le spectre de  $\tilde{\mathbf{L}}$  est borné (Lemme 1.7 (v) de [Chu97]) :

$$\rho(\tilde{\mathbf{L}}) \leq 2, \quad (2.15)$$

où  $\rho(\tilde{\mathbf{L}})$  désigne le rayon spectral de  $\tilde{\mathbf{L}}$ .

## Marche aléatoire sur un graphe

On peut associer à un graphe pondéré  $G$  une chaîne de Markov à espace d'états dans  $V(G)$ . Rappelons qu'un processus aléatoire  $X = (X_n)_{n \geq 0}$  défini sur un espace de probabilité  $(\Omega, \mathcal{A}, \mathbb{P})$  et à valeurs dans un ensemble fini, ici  $V(G)$ , est une chaîne de Markov de loi initiale  $\boldsymbol{\lambda}$  et de matrice de transition  $\mathbf{P} \in \mathbb{R}^{N \times N}$  si

- (i)  $(\forall i \in \{1, \dots, N\}) \quad \mathbb{P}(X_0 = u_i) = \lambda_i$ ;
- (ii)  $(\forall i_0, \dots, i_{n+1} \in \{1, \dots, N\}) \quad \mathbb{P}(X_{n+1} = u_{i_{n+1}} | X_n = u_{i_n}, \dots, X_0 = u_{i_0}) = \mathbb{P}(X_{n+1} = u_{i_{n+1}} | X_n = u_{i_n}) = \mathbf{P}_{i_n, i_{n+1}}$ .

La construction standard d'une chaîne de Markov sur  $G$  consiste à normaliser les lignes de la matrice d'adjacence  $\mathbf{W}$ . En effet, notons

$$\mathbf{K} = \mathbf{D}^{-1}\mathbf{W}. \quad (2.16)$$

La matrice  $\mathbf{K}$  est alors aléatoire et sert de matrice de transition à une chaîne de Markov dès qu'on se donne une loi initiale.

Remarquons à ce point que le spectre de  $\tilde{\mathbf{L}}$  est relié à celui de  $\mathbf{K}$  :  $\mathbf{f}$  est vecteur propre de  $\tilde{\mathbf{L}}$  associé à la valeur propre  $\lambda$  si et seulement si  $\mathbf{D}^{-1/2}\mathbf{f}$  est vecteur propre de  $\mathbf{K}$  associé à la valeur propre  $1 - \lambda$ . Ainsi, la matrice  $\mathbf{K}$  a elle aussi  $N$  valeurs propres réelles (comptées avec leur multiplicité).

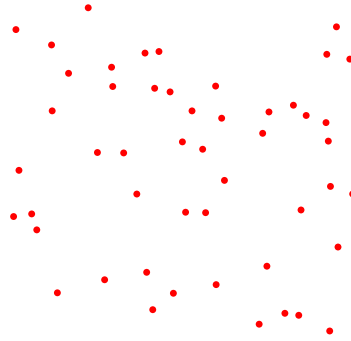


FIGURE 2.2 – 50 points générés aléatoirement dans  $[0, 5] \times [0, 5]$ .

## 2.3 Construction d'un graphe pondéré à partir d'un ensemble de données

Considérons un ensemble  $V = \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$  où  $\mathbf{u}_i \in \mathbb{R}^p$  pour tout  $i \in \{1, \dots, N\}$ . Dans la suite, nous ne ferons aucune hypothèse ni sur le paramètre  $p$ , ni sur l'échantillonnage des points  $\mathbf{u}_i$ . En particulier,  $p$  peut être très grand et les  $\mathbf{u}_i$  échantillonnés de manière non uniforme.

Nous présentons dans cette section quelques procédés de construction de graphes de similarité sur  $V$ . Ces constructions peuvent être séparées en deux catégories. Dans la première, on construit d'abord l'ensemble des arêtes  $E(G)$  à partir d'une distance sur  $\mathbb{R}^p$ . La distance choisie dans la suite est la distance euclidienne. Le choix de la fonction de pondération se fait ensuite. Les algorithmes de la seconde catégorie construisent simultanément la topologie du graphe et les poids de ses arêtes.

Le lecteur intéressé pourra consulter [LG12] pour d'autres constructions non mentionnées ici.

### 2.3.1 Construction de la topologie

#### Graphe complet

Une première construction consiste à considérer le graphe complet sur  $V(G)$ . La figure 2.3 montre un exemple de graphe complet sur l'ensemble des 50 points générés aléatoirement dans  $[0, 5] \times [0, 5]$  de la figure 2.2.

D'un point de vue algorithmique, il est souvent préférable de disposer de graphes dont les matrices d'adjacence sont *creuses*.

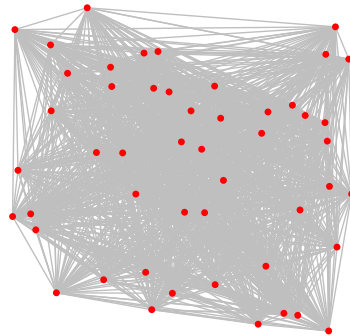


FIGURE 2.3 – Graphe complet sur les points de la figure 2.2.

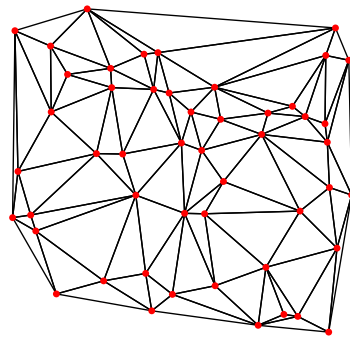


FIGURE 2.4 – Triangulation de Delaunay des points de la figure 2.2.

### Triangulation de Delaunay

La triangulation de Delaunay de  $V$  est le graphe dual du diagramme de Voronoï. La figure 2.4 montre le graphe correspondant à la triangulation de l'ensemble des points de la figure 2.2.

En grande dimension, le nombre d'arêtes de la triangulation de Delaunay devient important. Une approche basée sur un arbre couvrant de poids minimal est alors plus appropriée.

### Arbre couvrant de poids minimal

Soit  $G$  le graphe complet sur  $V$ . On associe à tout arbre couvrant  $G'$  de  $G$  le coût suivant :

$$c(G') = \sum_{(u,v) \in E(G')} w(\|\mathbf{u} - \mathbf{v}\|), \quad (2.17)$$

où  $w$  est une fonction décroissante. Un arbre couvrant est dit minimal (abrégé en MST) s'il minimise le coût (2.17). L'algorithme de Prim [CLRS01] permet

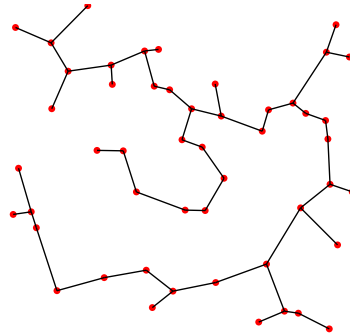
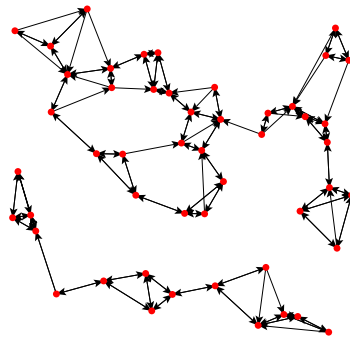


FIGURE 2.5 – Un MST sur les points de la figure 2.2.

FIGURE 2.6 – Graphe des  $k$  plus proches voisins avec  $k = 2$ .

de construire de tels arbres. L'intérêt de l'arbre couvrant de poids minimal est qu'il est connexe et contient  $|V| - 1$  arêtes.

La figure 2.5 montre un MST sur les points de la figure 2.2.

### Graphes des plus proches voisins

Soit  $k \geq 2$  un entier. Les graphes des  $k$  plus proches voisins sont obtenus en connectant chaque sommet à ses  $k$  plus proches voisins selon la distance euclidienne. Les graphes ainsi obtenus sont orientés : un sommet  $\mathbf{u}$  peut appartenir aux  $k$  plus proches voisins de  $\mathbf{v}$  sans que  $\mathbf{v}$  n'appartienne aux  $k$  plus proches voisins de  $\mathbf{u}$ . La figure 2.6 montre un exemple d'un tel graphe.

Afin d'obtenir un graphe non orienté on peut procéder de deux manières. Dans la première, on ne prend pas en compte l'orientation des arêtes. Ceci fournit un graphe dont chaque sommet est connecté à *au moins*  $k$  autres sommets. On parlera alors du *graphe des  $k$  voisins*. Cela correspond à retirer les orientations des arêtes dans la figure 2.6.

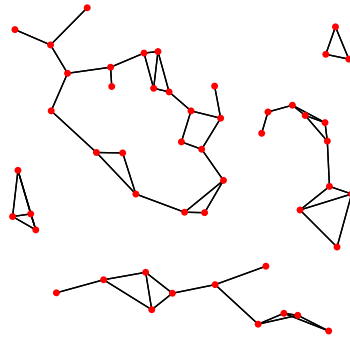


FIGURE 2.7 – Graphe des  $k$  plus proches voisins mutuels avec  $k = 2$ .

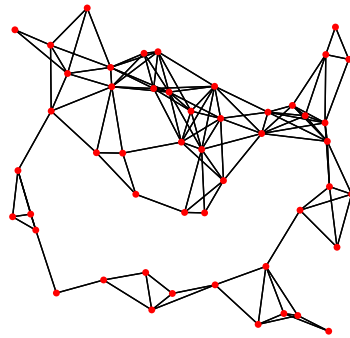


FIGURE 2.8 –  $\epsilon$ -graphe avec  $\epsilon = 1$ .

Dans la deuxième approche, on connecte deux sommets  $\mathbf{u}$  et  $\mathbf{v}$  si et seulement s'ils sont mutuellement voisins, c'est-à-dire si  $\mathbf{u}$  appartient aux  $k$  plus proches voisins de  $\mathbf{v}$  et réciproquement. On parlera alors du *graphe des  $k$  voisins mutuels*. Un exemple de tel graphe est donné dans la figure 2.7.

### $\epsilon$ -graphe

Soit  $\epsilon > 0$  un réel. Le  $\epsilon$ -graphe est obtenu en connectant deux sommets  $\mathbf{u}$  et  $\mathbf{v}$  si  $\|\mathbf{u} - \mathbf{v}\| \leq \epsilon$ . Le graphe que l'on obtient est alors non orienté ; un exemple est donné dans la figure 2.8.

### b-matching

La construction d'un graphe par b-matching à été proposée dans [JWC09] dans le contexte de l'apprentissage semi-supervisé. Cette approche permet en particulier de construire un graphe régulier, contrairement aux graphes des plus

proches voisins et au  $\epsilon$ -graphe. La topologie du graphe peut être encodée par une matrice binaire  $\mathbf{B} \in \mathbb{R}^{N \times N}$  avec la convention :

$$\mathbf{B}_{i,j} = \begin{cases} 1, & \text{si } \mathbf{u}_i \sim \mathbf{u}_j, \\ 0, & \text{sinon.} \end{cases} \quad (2.18)$$

Soit  $\mathbf{D} \in \mathbb{R}^{N \times N}$  la matrice des distances :  $\mathbf{D}_{i,j} = \|\mathbf{u}_i - \mathbf{u}_j\|$ . La matrice  $\mathbf{B}$  choisie est celle qui minimise, sous contraintes, sur l'ensemble des matrices binaires, la norme de Frobenius de  $\mathbf{DB}$  :

$$\hat{\mathbf{B}} = \underset{\mathbf{B} \in \mathbb{R}^{N \times N}}{\operatorname{argmin}} \quad \|\mathbf{DB}\|_F. \quad (2.19)$$

Les contraintes sont les suivantes :

$$\begin{cases} \sum_{j=1}^N \mathbf{B}_{i,j} & = b, \\ \mathbf{B}_{i,i} & = 0, \\ \mathbf{B}_{i,j} & = \mathbf{B}_{j,i}. \end{cases} \quad (2.20)$$

Le graphe dont la topologie est encodée par  $\mathbf{B}$  est ainsi  $b$ -régulier. Nous renvoyons le lecteur vers [HJ07] pour les détails relatifs à la résolution du problème de minimisation précédent.

### 2.3.2 Fonctions de similarité

Une fois la topologie de  $G$  fixée, nous pouvons associer un poids  $w(\mathbf{u}, \mathbf{v})$  à chaque arête  $(\mathbf{u}, \mathbf{v})$ . La quantité  $w(\mathbf{u}, \mathbf{v})$  représente alors une mesure de similarité entre  $\mathbf{u}$  et  $\mathbf{v}$ .

Dans la suite de ce manuscrit, nous considérons les deux fonctions de similarité suivantes :

$$(\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^p) \quad w(\mathbf{u}, \mathbf{v}) = e^{-\|\mathbf{u}-\mathbf{v}\|^2/\sigma^2}, \quad (2.21)$$

$$(\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^p) \quad w(\mathbf{u}, \mathbf{v}) = \frac{1}{\|\mathbf{u} - \mathbf{v}\| + \epsilon} \quad (2.22)$$

où  $\sigma$  et  $\epsilon$  sont deux constantes strictement positives fixées.

Le paramètre  $\sigma$  joue un rôle important dans la formule (2.21) puisqu'il contrôle la distribution des similarités que l'on obtient. En choisissant  $\sigma$  petit, on ne considère comme similaires que des points très proches au sens de la distance  $d$ . Plus la valeur de  $\sigma$  est grande plus on autorise des points distants à être considérés comme similaires. Le paramètre  $\sigma$  joue donc le rôle de *seuil*.

Si l'on souhaite accorder la même importance à toutes les arêtes, on choisit simplement  $w(\mathbf{u}, \mathbf{v}) = 1$  pour tout  $(\mathbf{u}, \mathbf{v}) \in E(G)$ .

Nous décrivons maintenant deux approches de construction simultanée de la topologie et des poids d'un graphe.

### 2.3.3 Construction simultanée de la topologie et des poids

#### L'approche de Daitch et al.

Dans [DKS09], les auteurs proposent une formulation variationnelle pour la construction d'un graphe à partir d'un ensemble de données  $V$ . Un graphe étant caractérisé par sa matrice d'adjacence  $\mathbf{W}$ , les auteurs proposent de minimiser sur l'ensemble des matrices  $\mathbf{W}$  symétriques à entrées positives et à diagonale nulle, le critère suivant :

$$f(\mathbf{W}) = \sum_{i=1}^N \left\| d_i \mathbf{u}_i - \sum_{j=1}^N w_{i,j} \mathbf{u}_j \right\|^2, \quad (2.23)$$

où  $d_i = \sum_{j=1}^N w_{i,j}$  désigne le degré du sommet  $i$ . Notons pour  $i \in \{1, \dots, N\}$

$$E_i = \left\| \mathbf{u}_i - \frac{1}{d_i} \sum_{j=1}^N w_{i,j} \mathbf{u}_j \right\|^2. \quad (2.24)$$

Ainsi,  $E_i$  représente l'erreur que l'on commet lorsqu'on *prédit* le vecteur  $\mathbf{u}_i$  à partir de ses voisins  $\mathbf{u}_j$  en utilisant les poids des arêtes comme coefficients de pondération. Remarquons que ce critère a été proposé dans [RS00]. La fonction  $f$  s'écrit alors

$$f(\mathbf{W}) = \sum_{i=1}^N d_i^2 E_i, \quad (2.25)$$

et permet ainsi d'agréger toutes les erreurs en un seul critère. Le critère  $f(\mathbf{W})$  est choisi au lieu de  $\sum_{i=1}^N E_i$  car il est quadratique en  $\mathbf{W}$ . En effet, remarquons d'abord qu'il s'écrit en fonction de la matrice laplacienne  $\mathbf{L}$  sous la forme

$$f(\mathbf{W}) = \|\mathbf{L}\mathbf{U}\|_2^2, \quad (2.26)$$

où  $\mathbf{U}$  est la matrice dont les lignes sont données par les vecteurs  $\mathbf{u}_i^\top$ , et  $\|\cdot\|_2$  désigne la norme de Frobenius. Ensuite, l'application qui associe à toute matrice symétrique  $\mathbf{W} \in \mathbb{R}^{n \times n}$  la matrice  $\mathbf{L}\mathbf{U} \in \mathbb{R}^{n \times d}$  est linéaire. Du point de vue de l'optimisation, le problème de minimisation de  $f(\mathbf{W})$  est posé sur un espace de dimension  $\frac{n(n-1)}{2}$ , ce qui peut en faire un problème de très grande échelle. Le lecteur intéressé pourra trouver des détails sur l'algorithme de minimisation utilisé par les auteurs ainsi que sur les propriétés des graphes obtenus dans [DKS09].



## Le graphe $\ell_1$

La construction du graphe  $\ell_1$  à été proposée dans [CYY+10] et vise à une meilleure adaptativité ainsi qu'à une meilleure robustesse vis-à-vis du bruit qui peut avoir contaminé les éléments de  $V$ . La construction repose sur le concept de décomposition parcimonieuse dans un dictionnaire [Mal09, SMF10, Ela10].

Dans la construction du graphe  $\ell_1$  les vecteurs  $\mathbf{u}_i$  jouent le rôle d'*atomes*. On construit donc un dictionnaire  $\mathcal{D} = [\mathbf{u}_1, \dots, \mathbf{u}_N]$ . Pour  $i \in \{1, \dots, N\}$ , notons  $\mathcal{D}_i$  le dictionnaire obtenu en retirant le vecteur  $\mathbf{u}_i$  :

$$\mathcal{D}_i = [\mathbf{u}_1, \dots, \mathbf{u}_{i-1}, \mathbf{u}_{i+1}, \dots, \mathbf{u}_N].$$

L'approche de [CYY+10] consiste à décomposer chaque vecteur  $\mathbf{u}_i$  dans le dictionnaire  $\mathcal{D}_i$ . La décomposition choisie est celle de norme  $\ell_1$  minimale. L'entrée  $(i, j)$  de la matrice d'adjacence est donnée par

$$\mathbf{W}_{i,j} = \alpha_j^i \tag{2.27}$$

où  $\alpha_j^i$  est la  $j^{\text{ème}}$  composante de  $\mathbf{u}_i$  dans  $\mathcal{D}_i$ . Remarquons que pour effectuer la décomposition précédente il faut que pour tout  $i \in \{1, \dots, N\}$ ,  $\text{rg}(\mathcal{D}_i) \geq p$ . En particulier, cela implique que  $N \geq p + 1$ . Ceci peut être problématique si  $p$  est très grand. Remarquons également que le graphe obtenu est orienté.

## 2.4 Signaux sur graphes

On désigne par *signal sur graphe* toute fonction associant une quantité scalaire ou vectorielle à tout sommet d'un graphe  $G$ . Nous définissons dans cette section des opérateurs différentiels discrets agissant sur de tels signaux. Munis de ces opérateurs, nous définissons des mesures permettant de quantifier la "régularité" relative de signaux sur graphes. Dans toute cette section,  $G$  désigne un graphe pondéré sur  $V = \{u_1, \dots, u_N\}$ .

### 2.4.1 Espaces $\mathcal{H}(V)$ et $\mathcal{H}(E)$

Un signal sur  $G$  est donc une fonction  $f : V(G) \rightarrow \mathbb{R}^m$ . Nous nous limitons dans la suite de cette section au cas scalaire :  $m = 1$ . L'ensemble des signaux sur  $G$  est noté dans la suite  $\mathcal{H}(V)$ . Il est muni du produit scalaire suivant :

$$(\forall f, g \in \mathcal{H}(V)) \quad \langle f, g \rangle_{\mathcal{H}(V)} = \sum_{u \in V} f(u)g(u). \tag{2.28}$$

On est aussi amené à considérer des signaux scalaires définis sur l'ensemble  $E(G)$  des arêtes d'un graphe pondéré  $G$ . Nous noterons cet espace  $\mathcal{H}(E)$  et considérons le produit scalaire suivant :

$$(\forall F, G \in \mathcal{H}(E)) \quad \langle F, G \rangle_{\mathcal{H}(E)} = \sum_{(u,v) \in E(G)}^N F(u,v)G(u,v). \quad (2.29)$$

Pour  $F \in \mathcal{H}(E)$  et  $u \in V$ , on notera  $F(u, \cdot)$  le vecteur  $[F(u,v), v \sim u]^\top$ , et  $\|F(u, \cdot)\|^2 = \sum_{v \sim u} F(u,v)^2$ .

## 2.4.2 Opérateurs de différence

Nous rappelons ici les opérateurs introduits dans [ELB08, ZS05]. Le lecteur intéressé par le calcul discret sur graphes pourra consulter l'ouvrage récent [GP10].

### Opérateurs du premier ordre : gradient et divergence

Une notion de dérivée directionnelle peut être introduite en considérant l'opérateur de différence  $\nabla^w : \mathcal{H}(V) \rightarrow \mathcal{H}(E)$  suivant :

$$(\forall f \in \mathcal{H}(V))(\forall (u,v) \in E(G)) \quad (\nabla^w f)(u,v) = \sqrt{w(u,v)}(f(v) - f(u)). \quad (2.30)$$

L'opérateur  $\nabla^w$  étant linéaire sur des espaces de dimension finie, il admet un adjoint  $(\nabla^w)^* : \mathcal{H}(E) \rightarrow \mathcal{H}(V)$  relativement aux produits scalaires  $\langle \cdot, \cdot \rangle_{\mathcal{H}(V)}$  et  $\langle \cdot, \cdot \rangle_{\mathcal{H}(E)}$ , c'est-à-dire vérifiant :

$$(\forall f \in \mathcal{H}(V))(\forall G \in \mathcal{H}(E)) \quad \langle \nabla^w f, G \rangle_{\mathcal{H}(E)} = \langle f, (\nabla^w)^* G \rangle_{\mathcal{H}(V)}. \quad (2.31)$$

L'expression de  $(\nabla^w)^*$  est obtenue en prenant  $f = \mathbf{1}_{\{u\}}$  dans l'équation précédente. On obtient alors

$$(\forall F \in \mathcal{H}(E)) \quad ((\nabla^w)^* F)(u) = \sum_{v \sim u} \sqrt{w(u,v)}(F(v,u) - F(u,v)). \quad (2.32)$$

L'opérateur divergence  $\text{div}^w : \mathcal{H}(E) \rightarrow \mathcal{H}(V)$  est donné par

$$\text{div}^w = -(\nabla^w)^*. \quad (2.33)$$

## Opérateurs du deuxième ordre : p-laplaciens

La composition des opérateurs  $\nabla^w$  et  $\text{div}^w$  permet d'obtenir un opérateur laplacien sur  $\mathcal{H}(V)$ . Notons  $\Delta^w : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$  l'opérateur défini pour tout  $f \in \mathcal{H}(V)$  par :

$$\Delta^w f = -\frac{1}{2} \text{div}^w(\nabla^w f). \quad (2.34)$$

Son expression est donnée par

$$(\Delta^w f)(u) = \sum_{v \sim u}^N w(u, v)(f(u) - f(v)). \quad (2.35)$$

En comparant avec l'équation (2.11) on voit que la matrice de  $\Delta^w$  dans la base canonique de  $\mathcal{H}(V)$  n'est autre que la matrice laplacienne  $\mathbf{L}$ .

Au-delà des opérateurs linéaires que nous venons de rappeler, l'intérêt du formalisme présenté est qu'il permet d'introduire des opérateurs non linéaires agissant sur des signaux sur graphes. Un exemple de tels opérateurs est donné dans [ELB08] où les auteurs définissent une famille d'opérateurs p-laplaciens  $\Delta^{w,p} : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$  définis pour tout  $f \in \mathcal{H}(V)$  par

$$(\forall p \geq 2) \quad \Delta^{w,p} f = -\frac{1}{2} \text{div}^w(\|\nabla^w f\|^{p-2} \nabla^w f). \quad (2.36)$$

L'expression est donnée par

$$(\Delta^{w,p} f)(u) = \frac{1}{2} \sum_{v \sim u}^N w(u, v) \left( \|\nabla^w f(u, \cdot)\|^{p-2} + \|\nabla^w f(v, \cdot)\|^{p-2} \right) (f(u) - f(v)). \quad (2.37)$$

Pour  $p = 2$ , on retrouve l'opérateur linéaire  $\Delta^w$ . Pour  $p > 2$ , les opérateurs  $\Delta^{w,p}$  sont non linéaires.

## Opérateurs normalisés

Une approche similaire à celle que nous venons de rappeler à été introduite dans [ZS05]. L'opérateur de différence de l'équation (2.30) y est remplacé par une version normalisée donnée par

$$(\forall f \in \mathcal{H}(V))(\forall (u, v) \in E(G)) \quad (\tilde{\nabla}^w f)(u, v) = \sqrt{w(u, v)} \left( \frac{f(v)}{\sqrt{d^w(v)}} - \frac{f(u)}{\sqrt{d^w(u)}} \right). \quad (2.38)$$

L'opérateur  $\tilde{\nabla}^w$  est encore linéaire sur des espaces de dimension finie. Il admet un adjoint  $(\tilde{\nabla}^w)^*$ . L'opérateur  $\tilde{\operatorname{div}}^w = -(\tilde{\nabla}^w)^*$  est donné par :

$$(\forall F \in \mathcal{H}(E)) \quad (\tilde{\operatorname{div}}^w F)(u) = \sum_{v \sim u} \sqrt{\frac{w(u,v)}{d^w(u)}} (F(u,v) - F(v,u)). \quad (2.39)$$

La composition de  $\tilde{\nabla}^w$  et  $\tilde{\operatorname{div}}^w$  donne lieu à un opérateur laplacien dont la matrice dans la base canonique n'est autre que la matrice laplacienne normalisée  $\tilde{\Delta}$  :

$$\tilde{\Delta}^w f = -\frac{1}{2} \tilde{\operatorname{div}}^w (\tilde{\nabla}^w f), \quad (2.40)$$

$$(\tilde{\Delta}^w f)(u) = \frac{1}{\sqrt{d^w(u)}} \sum_{v \sim u} w(u,v) \left( \frac{f(u)}{\sqrt{d^w(u)}} - \frac{f(v)}{\sqrt{d^w(v)}} \right). \quad (2.41)$$

Ici encore, l'intérêt de cette approche est que l'on peut définir une famille d'opérateurs p-laplaciens normalisés non linéaires :

$$(\forall p \geq 2) \quad \tilde{\Delta}^{w,p} f = -\frac{1}{2} \tilde{\operatorname{div}}^w (\|\tilde{\nabla}^w f\|^{p-2} \tilde{\nabla}^w f). \quad (2.42)$$

Nous nous limiterons dans la suite aux opérateurs non normalisés.

### 2.4.3 Mesures de régularité

Mesure basées sur  $\nabla^w$

Soit  $f \in \mathcal{H}(V)$  et  $u \in V$ . La quantité

$$\|(\nabla^w f)(u, \cdot)\| = \left( \sum_{v \sim u} |(\nabla^w f)(u, v)|^2 \right)^{1/2} = \left( \sum_{v \sim u} w(u, v) (f(u) - f(v))^2 \right)^{1/2} \quad (2.43)$$

représente une mesure de la *variation locale* de  $f$  au sommet  $u$ . En agrégeant ces quantités, on peut obtenir une *mesure globale* de la régularité relative de  $f$ . On est donc amenés à considérer la fonctionnelle suivante :

$$(\forall p \geq 1)(\forall f \in \mathcal{H}(V)) \quad J_p^w(f) = \frac{1}{p} \sum_{u \in V} \|(\nabla^w f)(u, \cdot)\|^p. \quad (2.44)$$

En dérivant, pour  $p \geq 2$ , l'expression (2.44) on obtient les opérateurs p-laplaciens non normalisés<sup>1</sup> [ELB08] :

$$\nabla J_p^w = 2\Delta^{w,p}. \quad (2.45)$$

---

1. L'opérateur  $\nabla$  dans le terme de droite dans (2.45) désigne le gradient d'une fonction définie sur un ouvert de  $\mathbb{R}^N$ .

La fonctionnelle  $J_p^w$  pour  $p = 2$  est une forme quadratique de matrice  $\mathbf{L}$ . Elle a été considérée dans divers travaux parmi lesquels [GO07, ELB08].

Pour  $p = 1$ , on obtient une fonctionnelle non différentiable de type *variation totale* sur  $\mathcal{H}(V)$ . Cette fonctionnelle a été considérée dans divers travaux parmi lesquelles [ELB08, GO08, PBC08, BEM09].

Le même principe peut être repris avec les opérateurs normalisés de la section précédente. Cette approche a été proposée pour  $p = 2$  dans [ZS04] et pour  $p \geq 1$  dans [ZS05] dans un cadre d'apprentissage semi-supervisé.

Nous attirons l'attention du lecteur sur le fait que les coordonnées des sommets  $u_i$  de  $G$  n'ont joué aucun rôle dans les opérateurs et les mesures que nous avons introduits. Le point important ici est que nous nous servons exclusivement de la relation d'adjacence pondérée entre les sommets pour étudier des signaux sur graphes.

### Mesures basées sur le spectre des matrices laplaciennes

Les normes que nous venons d'introduire sont basées sur l'opérateur de différence  $\nabla^w$ . Nous définissons ici deux normes basées sur le spectre des matrices laplaciennes  $\mathbf{L}$  et  $\tilde{\mathbf{L}}$ . L'espace  $\mathcal{H}(V)$  est ici identifié avec  $\mathbb{R}^N$  et nous notons ses éléments en gras.

Les matrices laplaciennes introduites dans 2.2.5 étant symétriques réelles, elles sont diagonalisables dans une base orthonormée chacune. Fixons une base orthonormée  $(\mathbf{e}_1, \dots, \mathbf{e}_N)$  constituée de vecteurs propres de  $\mathbf{L}$ . De même, fixons une base orthonormée  $(\tilde{\mathbf{e}}_1, \dots, \tilde{\mathbf{e}}_N)$  constituée de vecteurs propres de  $\tilde{\mathbf{L}}$ . Dans cette section nous voyons les vecteurs propres  $\mathbf{e}_i$  et  $\tilde{\mathbf{e}}_i$  comme des signaux sur  $V(G)$ .

Les bases  $(\mathbf{e}_i)$  et  $(\tilde{\mathbf{e}}_i)$  étant orthonormées, nous avons

$$(\forall \mathbf{f} \in \mathcal{H}(V)) \quad \mathbf{f} = \sum_{i=1}^N \langle \mathbf{f}, \mathbf{e}_i \rangle \mathbf{e}_i, \quad (2.46)$$

et

$$(\forall \mathbf{f} \in \mathcal{H}(V)) \quad \mathbf{f} = \sum_{i=1}^N \langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle \tilde{\mathbf{e}}_i. \quad (2.47)$$

On peut alors définir des a priori de parcimonie [Mal09, SMF10] dans ces bases en considérant les normes  $\ell_1$  des composantes :

$$(\forall \mathbf{f} \in \mathcal{H}(V)) \quad \|\mathbf{f}\|_L = \sum_{i=1}^N |\langle \mathbf{f}, \mathbf{e}_i \rangle|, \quad (2.48)$$

et

$$(\forall \mathbf{f} \in \mathcal{H}(V)) \quad \|\mathbf{f}\|_{\tilde{\mathcal{L}}} = \sum_{i=1}^N |\langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle|. \quad (2.49)$$

Nous nous sommes limités dans toute la suite de ce manuscrit aux mesures basées sur les opérateurs de différence. Les fonctionnelles basées sur ces mesures seront utilisées dans le chapitre suivant dans un cadre variationnel. Dans le chapitre 3 nous les utiliserons pour générer des décompositions multi-échelles de données définies sur des graphes.

## 2.5 Les images comme des fonctions sur des graphes

Dans cette section nous nous intéressons à une modélisation discrète où l'on considère les images comme des fonctions définies sur l'ensemble des sommets d'un graphe pondéré. Nous commençons par présenter les étapes qui permettent d'arriver à un tel modèle. Nous reprenons ensuite, dans le cas spécifique des images, la construction des graphes à partir d'un ensemble de données.

Notons que l'utilisation des graphes en traitement d'images ne se réduit pas à la modélisation. Par exemple, la notion de graphe intervient aussi dans des algorithmes d'optimisation combinatoire permettant de minimiser des énergies discrètes [BVZ01, Cha05, DS06]. Nous verrons une telle application des graphes dans le chapitre suivant. Le lecteur pourra consulter l'ouvrage [LG12] pour un état de l'art récent sur l'utilisation des graphes en traitement et analyse de l'image.

### 2.5.1 Principe

Deux types de modélisations sont souvent rencontrées en traitement et analyse de l'image. Une image étant considérée comme une fonction, deux facteurs interviennent : le domaine de définition et l'ensemble d'arrivée. Pour l'ensemble d'arrivée, nous considérons pour l'instant qu'il s'agit de  $\mathbb{R}$  pour une image en niveaux de gris et de  $\mathbb{R}^3$  pour une image en couleurs, auquel cas on notera  $f^{(l)}$  la  $l^{\text{ème}}$  composante. Notons cependant que la prise en compte de la quantification des images est fondamentale pour l'utilisation d'algorithmes d'optimisation combinatoire pour approcher les minima de fonctions d'énergies. Dans la suite de cette section nous nous intéressons uniquement à l'ensemble de définition.

Une image *continue* est une fonction (ou plus généralement une distribution) définie sur un domaine borné  $\Omega$  de  $\mathbb{R}^2$ . Une image *discrète*  $f$  est une fonction définie sur une *grille*  $\mathcal{Z}$ . Dans la suite nous noterons  $\{x_1, \dots, x_N\}$  les points de  $\mathcal{Z}$  et parlerons de *pixels*. Nous noterons  $f_i$  au lieu de  $f(x_i)$  si  $f$  est une image en niveaux de gris. De même, nous noterons  $f_i^{(l)}$  la  $l^{\text{ème}}$  application composante si  $f$  est une image en couleurs. Souvent,  $\mathcal{Z}$  est vu comme le résultat de la discrétisation d'un domaine continu, ce qui fait le lien entre l'approche discrète et l'approche continue.

Dans cette section nous considérons une image discrète comme un signal sur l'ensemble des sommets d'un graphe de proximité. Considérer une image discrète comme une fonction sur un graphe revient à remplacer la proximité spatiale entre pixels par celle induite par ce graphe. Ainsi, on considère que les intensités (ou les couleurs) ne sont plus définies sur une grille mais sur un ensemble de sommets dont la relation de proximité est donnée séparément par l'ensemble des arêtes. L'identification entre les pixels et les sommets du graphe permet de retrouver l'image.

La démarche qui consiste à considérer une image discrète  $f$  comme une fonction sur l'ensemble des sommets d'un graphe se fait en deux étapes. Dans la première, on associe à tout pixel  $x_i$  un *vecteur de caractéristiques*  $\varphi(x_i) = \mathbf{x}_i \in \mathbb{R}^p$ . L'application  $\varphi$  permet alors de passer de la grille  $\mathcal{Z}$  à l'ensemble  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ . Cette étape, qui peut être adaptative si l'on utilise les valeurs prises par l'image  $f$ , peut être implémentée de différentes manières.

A ce stade, on peut considérer l'image  $f$  non plus comme une fonction sur  $\mathcal{Z}$  mais comme une fonction sur  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ . Afin de traiter cette fonction, on construit un graphe de proximité  $G$  à partir de l'ensemble  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ . On considère alors  $f$  comme une fonction sur  $V(G)$  et on se sert des mesures de régularité définies dans la section précédente.

Nous donnons maintenant quelques exemples de vecteurs de caractéristiques.

### 2.5.2 Vecteurs de caractéristiques

Parmi les vecteurs de caractéristiques que l'on peut associer à un pixel, nous considérons trois variantes. La première est donnée par

$$\varphi_1(x_i) = x_i.$$

Le graphe construit à partir de l'ensemble  $\varphi_1(\mathcal{Z})$  n'est pas adaptatif : les valeurs de l'image  $f$  n'interviennent pas dans la construction. Ce graphe encode simplement une notion de proximité *spatiale* entre les pixels. La figure 2.10

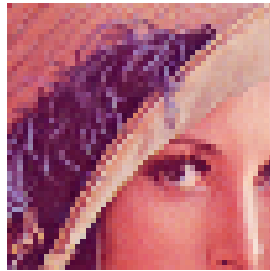


FIGURE 2.9 – Partie de l’image “Lena”.

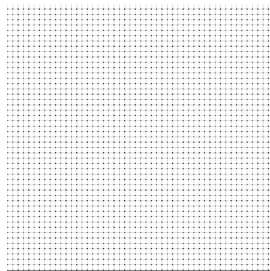


FIGURE 2.10 – Graphe-grille construit de à partir de l’image “Lena” de la figure 2.9.

montre la topologie d’un graphe en 4 voisinage construit à partir de l’image de la figure 2.9.

La deuxième variante, donnant lieu à une construction adaptative, est obtenue en considérant la fonction suivante :

$$\varphi_2(x_i) = (\lambda x_i, f_i),$$

où  $\lambda$  est une constante strictement positive. Si  $\lambda = 0$ , on obtient un graphe qui encode une notion de proximité *photométrique* (ou colorimétrique si l’image est en couleurs) sans aucune référence à la distribution spatiale des pixels. En choisissant  $\lambda > 0$ , on tient compte de la distribution spatiale des pixels. Le paramètre  $\lambda$  permet alors de balancer l’importance des facteurs spatiaux et photométriques. La figure 2.11 montre le graphe des 4 plus proches voisins au sens de la distance euclidienne avec  $\lambda = 0$ . La recherche des plus proches voisins a été effectuée pour chaque pixel au sein d’une fenêtre de taille 11x11. Enfin, la figure 2.12 montre le  $\epsilon$ -graphe obtenu avec le même choix ; la valeur de  $\epsilon$  est de 20.

Une autre construction adaptative, la troisième, est enfin obtenue en considérant des *patches* centrés en chaque pixel. Ce type d’approche a été popularisée en traitement d’images depuis l’introduction par Buades et al. de



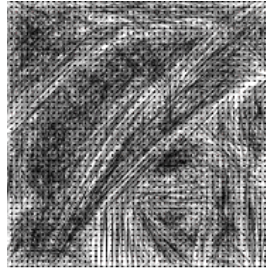


FIGURE 2.11 – Graphe des 4 plus proches voisins au sens de la distance photométrique. La recherche des voisins est limitée à une fenêtre de taille 11x11.



FIGURE 2.12 –  $\epsilon$ -graphe au sens de la distance photométrique. La recherche de voisins est limitée à une fenêtre de taille 11x11.

l'algorithme des moyennes non locales [BCM10]. Avec les notations de cette section, ceci correspond à considérer la fonction suivante :

$$\varphi_3(x_i) = \mathbf{p}_i,$$

où

$$\mathbf{p}_i = \{f_j, j \in N_r(i)\},$$

et  $N_r(i)$  désigne l'ensemble des pixels qui se trouvent à l'intérieur d'une boule  $\ell_\infty$  fermée de centre  $x_i$  et de rayon  $r$ . Si les images que l'on considère sont en couleurs, alors nous modifions la définition des patches de la manière suivante :

$$\mathbf{p}_i = \{f_j^{(l)}, 1 \leq l \leq 3, j \in N_r(i)\}.$$

La figure 2.13 montre la graphe des 4 plus proches voisins au sens de la distance euclidienne avec  $\lambda = 0$ . La recherche des plus proches voisins a été effectuée pour chaque pixel au sein d'une fenêtre de taille 11x11.

Dans la pratique [SWAC12, DFKE07], afin d'atténuer l'instabilité due à la présence de bruit dans les images, les vecteurs  $\varphi_2(x_i)$  et  $\varphi_3(x_i)$  sont calculés à partir d'une version lissée de  $f$ , obtenue par un prétraitement, par exemple par un filtrage gaussien.

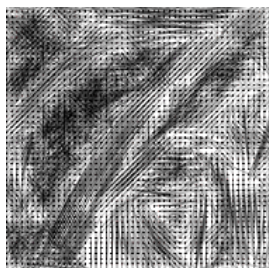


FIGURE 2.13 – Graphe des 4 plus proches voisins au sens de la distance euclidienne entre patches. La recherche des voisins est limité à une fenêtre de taille 11x11, et les patches sont de taille 5x5.

### 2.5.3 Débruitage d’images par filtres de voisinage

On considère le modèle de dégradation suivant :

$$g = f + \epsilon. \quad (2.50)$$

Dans (2.50),  $f$  désigne une image “idéale” non observée et  $\epsilon$  une dégradation appliquée à  $f$  donnant lieu à une image observée  $g$  de plus faible qualité que  $f$ . On se place dans le cadre où  $\epsilon$  est un vecteur aléatoire de moyenne nulle et dont les composantes sont indépendantes identiquement distribuées (i.i.d.). Le but est de construire un estimateur  $\hat{f}$  de  $f$  à partir de l’observation  $g$ . L’approche que l’on adopte est celle du débruitage par filtres de voisinages.

Une estimation linéaire  $\hat{f}$  de  $f$  est obtenue par convolution. Classiquement, l’estimation  $\hat{f}_i$  du pixel  $i$  dépend des distances spatiales entre  $i$  et les autres pixels  $j$ . L’archétype de ce type de méthodes est le filtre gaussien donné par

$$\hat{f}_i = \frac{1}{C_i} \sum_{j=1}^N e^{-\|x_i - x_j\|^2 / 2\sigma^2} f_j, \quad (2.51)$$

où  $C_i$  est un coefficient de normalisation. Le concept de filtre de voisinage a été introduit par Yaroslavsky dans [YE96]. Dans cette approche, les poids intervenant dans le débruitage dépendent de l’image. L’équation générique de débruitage est la suivante :

$$\hat{f}_i = \frac{1}{C_i} \sum_{j=1}^N W_f(x_i, x_j) f_j. \quad (2.52)$$

Dans (2.52),  $W_f$  désigne un noyau, éventuellement dépendant de  $f$ . Le filtre de Yaroslavsky [YE96] s’obtient avec le choix

$$W_f(x_i, x_j) = \mathbf{1}_{\{\|f(x_i) - f(x_j)\| \leq h\}}. \quad (2.53)$$

Le filtre SUSAN [SB97] s'obtient avec le choix

$$W_f(x_i, x_j) = \exp\{-\|f(x_i) - f(x_j)\|^2/2\sigma^2\}. \quad (2.54)$$

Enfin, le filtre bilatéral [TM98] correspond au choix

$$W_f(x_i, x_j) = \exp\{-\|f(x_i) - f(x_j)\|^2/2\sigma_r^2\} \exp\{-\|x_i - x_j\|^2/2\sigma_p^2\}. \quad (2.55)$$

L'adaptativité des filtres que l'on vient de présenter est obtenue en comparant les valeurs ponctuelles des pixels (intensité ou couleur). Le filtre des moyennes non locales introduit par Buades et al. dans [BCM10] propose de comparer des fenêtres centrées en chaque pixel ; ces fenêtres sont appelées des patches. Le filtre des moyennes non locales est obtenu grâce au procédé (2.52) avec le choix :

$$W_f(x_i, x_j) = \exp\{-\|F(x_i) - F(x_j)\|_{2,a}^2/2\sigma^2\}. \quad (2.56)$$

Ici,  $F(x_i)$  désigne un patch centré au pixel  $i$ , c'est-à-dire que  $F(x_i)$  est le vecteur obtenu par concaténation des intensités de  $f$  au sein d'une fenêtre centrée en  $i$ . La norme  $\|\cdot\|_{2,a}$  est la norme  $\ell_2$  pondérée par une gaussienne de variance  $G_a$ . Cette pondération permet de prendre en compte une notion de localité lors de la comparaison de patches.

En pratique, deux points viennent modifier les définitions des filtres que nous venons de rappeler. Le premier concerne la localité des filtres. On restreint en général le voisinage de chaque pixel à une fenêtre de recherche. Ce point est motivé par des raisons algorithmiques et vise à pouvoir effectuer le débruitage en un temps raisonnable. Le deuxième point concerne la robustesse au bruit. A cette fin, les comparaisons entre pixels, qu'elles soient basées sur les intensités ou sur les patches, sont souvent calculées à partir d'une version lissée de l'image observée. Une telle image peut être obtenue simplement par un filtrage gaussien.

Enfin, le procédé que nous venons de rappeler peut être itéré afin d'obtenir une meilleure estimation. Nous abordons cette approche dans la sous-section suivante en faisant le lien avec les graphes.

## 2.5.4 Lien avec les graphes

Si on interprète le noyau  $W_f$  de la sous-section précédente comme la matrice d'adjacence d'un graphe construit sur l'ensemble des pixels, on voit que la formule (2.52) devient

$$\hat{\mathbf{f}} = \mathbf{Kf}, \quad (2.57)$$

où  $\mathbf{K}$  est la matrice de transition de la marche aléatoire sur le graphe, décrite dans (2.2.5). Ainsi, pour débruiter on multiplie par la matrice de transition.

Les différents filtres présentés dans la sous-section précédente peuvent être retrouvés grâce aux constructions de graphes présentées dans 2.5.2. La formule (2.57) peut être réécrite en prenant en compte l'interprétation de  $\mathbf{K}$  comme une matrice de transition :

$$\hat{f}_i = \sum_{j=1}^N \mathbb{P}(X_{n+1} = j | X_n = i) f_j \quad (2.58)$$

$$= \mathbb{E}[f(X_{n+1}) | X_n = i]. \quad (2.59)$$

On peut itérer le processus de débruitage en considérant des puissances de la matrice  $K$ . Ainsi, on pose

$$\begin{cases} \mathbf{f}^0 = \mathbf{f}, \\ \mathbf{f}^{n+1} = \mathbf{K}\mathbf{f}^n. \end{cases} \quad (2.60)$$

Du point de vue de la sous-section précédente, ceci correspondant à garder le même noyau au cours des itérations. On pourra se référer à [SSN09] pour d'autres stratégies d'itérations des filtres de voisinage. La formule (2.59) devient

$$f_i^{n+1} = \mathbb{E}[f(X_n) | X_0 = i]. \quad (2.61)$$

On peut donc faire le lien entre les propriétés asymptotiques des filtres et celles de la marche aléatoire sur le graphe correspondant.

Il existe aussi un lien entre l'opération de débruitage et la mesure de régularité (2.49). Ce lien repose sur l'observation que  $\mathbf{u}$  est un vecteur propre de  $\mathbf{L}$  associé à une valeur propre  $\lambda$  si et seulement si  $\mathbf{D}^{-1/2}\mathbf{u}$  est vecteur propre de  $\mathbf{K}$  associé à la valeur propre  $1 - \lambda$ . Cette équivalence permet de montrer que si  $G$  est un graphe régulier alors l'opération  $\mathbf{K}\mathbf{f}$  fait décroître la mesure (2.49). En effet considérons la décomposition de  $\mathbf{f}$  dans la base  $(\tilde{\mathbf{e}}_i)_{i=1,\dots,N}$  des vecteurs propres de  $\mathbf{L}$  :

$$\mathbf{f} = \sum_{i=1}^N \langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle \tilde{\mathbf{e}}_i. \quad (2.62)$$

Nous avons alors

$$\mathbf{K}\mathbf{f} = \sum_{i=1}^N \langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle \mathbf{K}\tilde{\mathbf{e}}_i \quad (2.63)$$

$$= \sum_{i=1}^N \langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle \mathbf{K}\mathbf{D}^{-1/2}\mathbf{D}^{1/2}\tilde{\mathbf{e}}_i \quad (2.64)$$

$$(2.65)$$

Si  $G$  est régulier, alors  $\mathbf{D}^{-1/2}$  commute avec  $\mathbf{K}$ . On a alors

$$\mathbf{Kf} = \sum_{i=1}^N \langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle \mathbf{D}^{-1/2} \mathbf{K} \mathbf{D}^{1/2} \tilde{\mathbf{e}}_i \quad (2.66)$$

$$= \sum_{i=1}^N (1 - \lambda_i) \langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle \mathbf{D}^{-1/2} \mathbf{D}^{1/2} \tilde{\mathbf{e}}_i \quad (2.67)$$

$$= \sum_{i=1}^N (1 - \lambda_i) \langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle \tilde{\mathbf{e}}_i \quad (2.68)$$

$$(2.69)$$

La norme  $\|\cdot\|_{\mathbf{L}}$  de  $\mathbf{Kf}$  est donc inférieure à celle de  $\mathbf{f}$  :

$$\|\mathbf{Kf}\|_{\mathbf{L}} = \sum_{i=1}^N |(1 - \lambda_i)| |\langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle| \quad (2.70)$$

$$\leq \sum_{i=1}^N |\langle \mathbf{f}, \tilde{\mathbf{e}}_i \rangle| = \|\mathbf{f}\|_{\mathbf{L}}, \quad (2.71)$$

$$(2.72)$$

où l'inégalité précédente repose sur le fait, mentionné dans (2.2.5), que  $0 \leq \lambda_i \leq 2$  pour tout  $i \in \{1, \dots, N\}$ .

## 2.6 Conclusion

Ce chapitre a permis de poser quelques définitions concernant les graphes pondérés et les signaux. Ces définitions seront d'un usage constant dans la suite de ce manuscrit. Nous avons également rappelé comment, pour les signaux sur graphes, des mesures relatives de régularité peuvent être définies à partir d'opérateurs différentiels discrets. La construction d'un graphe pondéré à partir d'un ensemble de données a été détaillée. Enfin, nous avons vu les spécificités de la modélisation, par des graphes pondérés, des images. Cette modélisation a permis d'intégrer dans un même formalisme l'ensemble des filtres de voisinages.

Le chapitre suivant examine comment les mesures de régularité définies ici peuvent être utilisées comme a priori dans un cadre variationnel pour le débruitage et la simplification de signaux sur graphes.

# Chapitre 3

## Approches variationnelles pour le débruitage de signaux sur graphes

### 3.1 Introduction

Nous abordons dans ce chapitre des méthodes variationnelles pour le débruitage de signaux définis sur l'ensemble des sommets d'un graphe pondéré général.

Ce chapitre est organisé de la manière suivante. Nous rappelons tout d'abord dans la section 3.2 quelques éléments d'analyse et d'optimisation convexes en dimension finie. Les outils présentés dans cette section seront utiles pour le développement du présent chapitre et pour le suivant. Dans la section 3.3 nous rappelons le cadre de l'analyse bayésienne et examinons deux a priori classiques en traitement de l'image : l'a priori de Sobolev discret et celui basé sur la variation totale (TV). L'objet de la section 3.4 est de transposer les deux a priori précédents au cas des signaux sur graphes. En vue de l'étape d'optimisation, nous réécrivons ces a priori à l'aide de produits matrice-vecteur. Nous caractérisons la solution du problème basé sur la pénalisation TV sur graphes. Cette caractérisation fait apparaître un équivalent de l'espace  $G$  introduit par Meyer dans [Mey01]. Nous introduisons cet espace et le caractérisons. La fonctionnelle TV sur graphes n'étant pas différentiable, nous détaillons l'application de trois algorithmes proximaux récents à ce problème. Nous montrons également rapidement comment un algorithme combinatoire simple, basé sur les coupures de graphes, permet d'obtenir des solutions approchées du problème quantifié. Dans la section 3.5,

nous effectuons plusieurs tests de débruitage sur des images et des maillages. Une conclusion de ce chapitre est présentée dans la section 3.6.

## 3.2 Éléments d'analyse et d'optimisation convexes

Nous rappelons dans cette section quelques éléments d'analyse et d'optimisation convexes en dimension finie. Notre présentation est brève : le but est d'établir quelques notations et de regrouper des résultats qui seront utiles pour la suite de ce chapitre et pour le suivant. Nous ferons référence à l'ouvrage [BC11] pour les preuves des résultats énoncés ici. Le lecteur intéressé pourra également consulter les ouvrages [HUL01, BL06, Roc96].

Dans toute cette section,  $(\mathcal{H}, \langle \cdot, \cdot \rangle)$  désigne un espace euclidien et  $\|\cdot\|$  la norme induite par  $\langle \cdot, \cdot \rangle$ .

### 3.2.1 Espace $\Gamma_0(\mathcal{H})$

**Définition 3.1.** Soit  $f : \mathcal{H} \rightarrow ]-\infty, +\infty]$ .

1. Le *domaine* de  $f$  est l'ensemble  $\text{dom}(f) = \{x \in \mathcal{H} : f(x) < +\infty\}$ .
2. On dit que  $f$  est *propre* si  $\text{dom}(f) \neq \emptyset$ .
3. L'*épigraphe* de  $f$  est l'ensemble  $\text{epi}(f) = \{(x, a) \in \mathcal{H} \times \mathbb{R} : f(x) \leq a\}$ .
4. On dit que  $f$  est *convexe* si son épigraphe est un ensemble convexe.
5. On dit que  $f$  est *semi-continue inférieurement* (s.c.i.) si son épigraphe est fermé.
6.  $\Gamma_0(\mathcal{H})$  désigne l'ensemble des fonctions de  $\mathcal{H}$  dans  $] -\infty, +\infty]$  propres, s.c.i. et convexes.

**Remarque 3.1.** Notons que si  $f$  est convexe,  $\text{dom}(f)$  est un ensemble convexe. En effet,  $\text{dom}(f) = L(\text{epi}(f))$ , où  $L : (x, a) \in \mathcal{H} \times \mathbb{R} \mapsto x \in \mathcal{H}$  est linéaire. De plus,  $f$  est convexe si et seulement si (Proposition 8.4 [BC11])

$$(\forall x, y \in \text{dom}(f)) (\forall \alpha \in ]0, 1[) \quad f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y). \quad (3.1)$$

Si  $f$  est propre et l'inégalité précédente stricte pour tout  $\alpha \in ]0, 1[$  et tous  $x, y \in \text{dom}(f)$  distincts, alors  $f$  est dite *strictement convexe*.

**Exemple 3.1.** Soit  $C$  une partie de  $\mathcal{H}$ . On appelle *fonction indicatrice* de  $C$  la fonction définie pour tout  $x \in \mathcal{H}$  par

$$\iota_C(x) = \begin{cases} 0, & \text{si } x \in C, \\ +\infty, & \text{sinon.} \end{cases} \quad (3.2)$$

Il est aisé de voir que  $\iota_C \in \Gamma_0(\mathcal{H})$  si et seulement si  $C$  est une partie de  $\mathcal{H}$  convexe, fermée et non vide.

### 3.2.2 Fonction conjuguée

**Définition 3.2.** Soit  $f : \mathcal{H} \rightarrow ]-\infty, +\infty]$ . La fonction conjuguée de  $f$  est la fonction  $f^* : \mathcal{H} \rightarrow ]-\infty, +\infty]$  définie pour tout  $y \in \mathcal{H}$  par

$$f^*(y) = \sup_{x \in \mathcal{H}} (\langle x, y \rangle - f(x)). \quad (3.3)$$

**Exemple 3.2.** Soit  $C$  une partie non vide de  $\mathcal{H}$ .

$$(\iota_C)^*(y) = \sup_{x \in C} \langle x, y \rangle =: \sigma_C(y). \quad (3.4)$$

La fonction  $\sigma_C$  s'appelle *fonction d'appui* de  $C$ .

Le théorème suivant est connu sous le nom de *théorème de Fenchel-Moreau*.

**Théorème 3.1** (Théorème 13.32 [BC11]). Soit  $f \in ]-\infty, +\infty]$  propre. Alors  $f$  est s.c.i. et convexe si et seulement si  $f = f^{**}$ ; auquel cas  $f^*$  est propre.

Le corollaire suivant sera d'un usage fréquent dans la suite.

**Corollaire 3.1** (Corollaire 13.33 [BC11]). Soit  $f \in \Gamma_0(\mathcal{H})$ . Alors,  $f^* \in \Gamma_0(\mathcal{H})$  et  $f^{**} = f$ .

**Exemple 3.3.** La notion de norme duale jouera un rôle important dans la suite. Rappelons que si  $N$  désigne une norme sur  $\mathcal{H}$ , l'application  $N_*$  définie pour tout  $y \in \mathcal{H}$  par

$$N_*(y) = \max_{N(x) \leq 1} \langle x, y \rangle \quad (3.5)$$

est une norme sur  $\mathcal{H}$  dite *norme duale* de  $N$ . De plus,

$$N_{**} = N. \quad (3.6)$$

Nous avons le résultat suivant<sup>1</sup> :

$$(\forall y \in \mathcal{H}) \quad N^*(y) = \iota_{\{N_* \leq 1\}}(y). \quad (3.7)$$

En effet,

$$(\forall y \in \mathcal{H}) \quad N(y) \stackrel{(3.6)}{=} \sup_{N_*(x) \leq 1} \langle x, y \rangle = \sigma_{\{N_* \leq 1\}}(y).$$

L'application  $N_*$  étant une norme, l'ensemble non vide  $C = \{N_* \leq 1\}$  est fermé et convexe. Par suite,  $\iota_C \in \Gamma_0(\mathcal{H})$  et le résultat découle de (3.4) et du corollaire 3.1.

---

1. Le symbole  $*$  en indice désigne la norme duale. Le même symbole placé en exposant désigne la fonction conjuguée.



### 3.2.3 Opérateur proximal

Nous commençons par rappeler la notion de projection sur un convexe fermé.

**Proposition 3.1** (Théorème 3.14 de [BC11]). Soit  $C$  une partie de  $\mathcal{H}$  non vide convexe et fermée, et  $x \in \mathcal{H}$ . L'application  $y \in C \mapsto \|x - y\|^2$  atteint son infimum en un unique point noté  $\text{proj}_C x$  :

$$\text{proj}_C x = \underset{y \in C}{\text{argmin}} \|x - y\|^2. \quad (3.8)$$

De plus,  $\text{proj}_C x$  est l'unique point  $y$  de  $C$  vérifiant

$$(\forall z \in C) \quad \langle z - y, x - y \rangle \leq 0. \quad (3.9)$$

La notion de projection sur un convexe se généralise à l'aide du concept d'opérateur proximal introduit par Moreau dans [Mor62].

**Proposition 3.2** (Définition 12.23 de [BC11]). Soit  $f \in \Gamma_0(\mathcal{H})$  et  $x \in \mathcal{H}$ . La fonction  $y \in \mathcal{H} \mapsto \frac{1}{2}\|x - y\|^2 + f(y)$  atteint son infimum en un unique point noté  $\text{prox}_f x$  :

$$\text{prox}_f x = \underset{y \in \mathcal{H}}{\text{argmin}} f(y) + \frac{1}{2}\|y - x\|^2. \quad (3.10)$$

L'opérateur  $\text{prox}_f : \mathcal{H} \rightarrow \mathcal{H}$  ainsi défini s'appelle *opérateur proximal* de  $f$ .

**Exemple 3.4.** Soit  $C$  une partie de  $\mathcal{H}$  non vide, convexe et fermée. On sait que  $\iota_C \in \Gamma_0(\mathcal{H})$ . On peut donc définir l'opérateur proximal de  $\iota_C$ . En appliquant la définition, on voit que  $\text{prox}_{\iota_C} = \text{proj}_C$ . La notion d'opérateur proximal généralise donc celle d'opérateur de projection.

La proposition suivante est connue sous le nom d'identité de Moreau. Elle a été introduite dans [Mor62] et généralisée dans [CW05].

**Proposition 3.3.** Soit  $f \in \Gamma_0(\mathcal{H})$  et  $z \in \mathcal{H}$ . Alors

$$(\forall \lambda > 0) \quad z = \text{prox}_{\lambda f} z + \lambda \text{prox}_{f^*/\lambda}(z/\lambda). \quad (3.11)$$

L'identité (3.11) conduit à une famille de décompositions mono-échelles d'un signal  $z \in \mathcal{H}$ . Cette famille est paramétrée par  $\lambda$  et  $f$ . L'identité de Moreau permet également le calcul de l'opérateur proximal d'une fonction à partir de celui de sa conjuguée.

### 3.2.4 Algorithmes proximaux

Nous serons amenés dans ce chapitre et le suivant à utiliser des algorithmes proximaux pour la minimisation de fonctionnelles non différentiables. Nous présentons dans cette section trois algorithmes génériques ; il s'agit de l'algorithme explicite-implicite [CW05], de celui de Beck et Teboulle [BT09b, BT09a] et enfin de celui de Chambolle et Pock [CP11a]. Le lecteur intéressé pourra consulter [CP11b] pour une introduction aux principaux algorithmes proximaux utilisés en traitement du signal et de l'image. Notons enfin que nous n'abordons pas ici les algorithmes multi-pas. Une introduction peut être trouvée dans le chapitre 4 de [Wei08].

#### Algorithme explicite-implicite

Soit  $F_1 \in \Gamma_0(\mathcal{H})$  et  $F_2 : \mathcal{H} \rightarrow \mathbb{R}$  convexe, différentiable à gradient  $\beta$ -lipschitzien,  $\beta > 0$ , :

$$(\forall x, y \in \mathcal{H}) \quad \|\nabla F_2(x) - \nabla F_2(y)\| \leq \beta \|x - y\| \quad (3.12)$$

On suppose que le problème suivant

$$\underset{x \in \mathcal{H}}{\text{Minimiser}} \quad F_1(x) + F_2(x) \quad (3.13)$$

admet au moins une solution, c'est-à-dire que  $\underset{\mathcal{H}}{\text{Argmin}} F_1 + F_2 \neq \emptyset$ . Ceci est le cas, par exemple, si  $\lim_{\|x\| \rightarrow +\infty} F_1(x) + F_2(x) = +\infty$ . Cette solution est alors caractérisée par l'équation suivante [CW05] :

$$(\forall \gamma > 0) \quad x = \text{prox}_{\gamma F_1}(x - \gamma \nabla F_2(x)). \quad (3.14)$$

L'algorithme explicite-implicite (forward-backward en anglais) construit une suite  $(x^{(k)})_k$  qui converge vers une solution du problème (3.13) [CW05]. L'algorithme est le suivant :

$$\begin{cases} x^{(0)} \in \mathcal{H}, \\ \gamma^{(n)} \in ]0, 2/\beta[, \\ y^{(n)} = x^{(n)} - \gamma^{(n)} \nabla F_2(x^{(n)}), \\ x^{(n+1)} = \text{prox}_{\gamma^{(n)} F_1} y^{(n)}. \end{cases} \quad (3.15)$$

### Algorithme de Beck et Teboulle (FISTA)

Cet algorithme a été introduit dans [BT09b, BT09a]. Il s'agit d'une modification de l'algorithme explicite-implicite et permet de construire une suite  $(x^{(n)})$  telle que  $(F_1 + F_2)(x^{(n)})$  converge vers  $(F_1 + F_2)(x^*)$ , où  $x^* \in \text{Argmin } F_1 + F_2$ .

Notons que la suite des itérés  $(x^{(n)})$  n'est plus assurée de converger vers une solution de (3.13). L'intérêt ici est que la vitesse de convergence de  $(F_1 + F_2)(x^{(n)})$  est optimale au sens de [Nes03]. L'algorithme de Beck et Teboulle est le suivant :

$$\begin{cases} x^{(0)} \in \mathcal{H}, z^{(0)} = x^{(0)}, t^{(0)} = 1, \\ y^{(n)} = z^{(n)} - \beta^{-1} \nabla F_2(z^{(n)}), \\ x^{(n+1)} = \text{prox}_{\beta^{-1} F_1} y^{(n)}, \\ t^{(n+1)} = \frac{1 + \sqrt{4(t^{(n)})^2 + 1}}{2}, \\ \lambda^{(n)} = 1 + \frac{t^{(n)} - 1}{t^{(n+1)}}, \\ z^{(n+1)} = x^{(n)} + \lambda^{(n+1)}(x^{(n+1)} - x^{(n)}). \end{cases} \quad (3.16)$$

### Algorithme primal-dual de Chambolle et Pock

Le problème type considéré dans [CP11a] est le suivant :

$$\underset{x \in \mathcal{X}}{\text{Minimiser}} F(Kx) + G(x). \quad (3.17)$$

Ici,  $\mathcal{X}$  et  $\mathcal{Y}$  désignent deux espaces vectoriels de dimension finie,  $F \in \Gamma_0(\mathcal{Y})$ ,  $G \in \Gamma_0(\mathcal{X})$  et  $K : \mathcal{X} \rightarrow \mathcal{Y}$  est un opérateur *linéaire*. On suppose ici que le problème type (3.17) admet une solution. Les auteurs proposent l'algorithme suivant :

$$\begin{cases} y^{n+1} = \text{prox}_{\sigma F^*}(y^n + \sigma K \bar{x}^n), \\ x^{n+1} = \text{prox}_{\tau G}(x^n - \tau K^* y^{n+1}), \\ \bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n), \end{cases} \quad (3.18)$$

où  $K^*$  désigne l'opérateur adjoint de  $K$ . La convergence de l'algorithme (3.18) est garantie si  $\theta = 1$  et  $0 < \tau \sigma L^2 < 1$  où  $L = \|K\| = \max_{\|x\| \leq 1} \|Kx\|$  désigne la norme subordonnée de  $K$ . Nous renvoyons le lecteur intéressé à [CP11a] pour une étude de la vitesse de convergence de (3.18).

## 3.3 Le cadre bayésien pour le débruitage d'images

Nous rappelons dans cette section quelques approches variationnelles développées pour le débruitage d'images. Il s'agit là de méthodes qui visent

à estimer une image originale à partir d'une version bruitée. Nous rappelons, en particulier, comment de telles méthodes peuvent être obtenues à partir d'une analyse bayésienne.

Cette section ne contient ni état de l'art complet, ni comparaisons des performances des différentes approches. Le but ici est de rappeler certains modèles utilisés en traitement de l'image dont nous généraliserons l'usage aux signaux sur graphes dans la section 3.4. Le lecteur intéressé par un état de l'art des méthodes de débruitage d'images pourra consulter les articles de synthèse [BCM10, Mil11] ainsi que les références qui y sont fournies.

Nous considérons dans ce qui suit les images digitales scalaires de taille  $m \times n$  comme des vecteurs dans  $\mathcal{X} = \mathbb{R}^N$ ,  $N = m \times n$ , obtenus en concaténant les colonnes des matrices images. L'espace  $\mathcal{X}$  est muni du produit scalaire usuel noté dans la suite  $\langle \cdot, \cdot \rangle_{\mathcal{X}}$  et  $\|\cdot\|_{\mathcal{X}}$  désigne la norme associée. Ainsi, on notera  $f_i$  l'intensité en un pixel  $i$ . On s'autorisera également à noter  $f_{l,k}$  afin de faire apparaître les coordonnées de la grille sous-jacente. L'espace  $\mathcal{Y} = \mathbb{R}^N \times \mathbb{R}^N$  est muni du produit scalaire suivant :

$$(\forall \mathbf{p} = (\mathbf{p}^1, \mathbf{p}^2), \mathbf{q} = (\mathbf{q}^1, \mathbf{q}^2) \in \mathcal{Y}) \quad \langle \mathbf{p}, \mathbf{q} \rangle_{\mathcal{Y}} = \sum_{i=1}^N (p_i^1 q_i^1 + p_i^2 q_i^2),$$

et  $\|\cdot\|_{\mathcal{Y}}$  désigne la norme associée.

### 3.3.1 Débruitage par minimisation d'énergie

Considérons deux images  $\mathbf{f}, \mathbf{u}_0 \in \mathcal{X}$ . Nous supposons que ces deux images sont reliées par le modèle suivant :

$$\mathbf{f} = \mathbf{u}_0 + \boldsymbol{\eta}, \tag{3.19}$$

où  $\boldsymbol{\eta}$  désigne une *perturbation* aléatoire affectant le signal "parfait"  $\mathbf{u}_0$  et donnant lieu à l'observation bruitée  $\mathbf{f}$ . Nous précisons la nature de cette perturbation dans la sous-section suivante. Le but ici est de construire un estimateur  $\hat{\mathbf{u}}$  de  $\mathbf{u}_0$ .

Dans une approche variationnelle, l'estimateur  $\hat{\mathbf{u}}$  minimise une *fonction d'énergie*  $E$  de la forme

$$E(\mathbf{u}) = \lambda R(\mathbf{u}) + D(\mathbf{u}, \mathbf{f}). \tag{3.20}$$

Le terme  $R(\mathbf{u})$ , appelé *fonction de pénalisation* ou *terme de régularité* sert à favoriser une classe de signaux ayant des propriétés prédéfinies. Le terme  $D(\mathbf{u}, \mathbf{f})$ , appelé *terme d'attache aux données*, sert quant à lui à favoriser la

proximité entre les signaux candidats  $\mathbf{u}$  et le signal observé  $\mathbf{f}$ . Enfin,  $\lambda > 0$  est un paramètre, dit *de régularisation*, qui permet de balancer l'importance des deux termes précédents.

Les méthodes de minimisation d'énergie sont à la croisée de plusieurs méthodologies. En particulier, elles peuvent être reliées aux approches d'estimation bayésienne [GG84, GG86, Gem90, Bes93], à la régularisation de problèmes inverses mal posés (chapitre 1 de [Sch10]), et enfin aux méthodes variationnelles continues [AK06]. Le lecteur intéressé pourra consulter le chapitre 5 de [Sch10] ainsi que les références qui y sont données pour plus de détails.

Dans ce qui suit, nous adoptons un point de vue bayésien. Notre présentation est inspirée de [GG86] auquel nous renvoyons le lecteur pour plus de détails.

### 3.3.2 Le cadre bayésien

Nous précisons tout d'abord la nature de la dégradation  $\boldsymbol{\eta}$ . Nous supposons que  $\boldsymbol{\eta}$  est une *réalisation d'une variable aléatoire* à valeurs dans  $\mathcal{X}$ . On se restreindra ici au modèle le plus simple : celui du bruit blanc gaussien.

L'approche bayésienne pour le débruitage comporte cinq étapes. La première consiste à formuler une préférence ou un *a priori* quant aux signaux que l'on souhaite privilégier. Dans le cadre bayésien, cette préférence prend la forme d'un modèle probabiliste : on se donne une densité de probabilité  $\Pi$  sur  $\mathcal{X}$ , appelée *densité a priori*. Dans la pratique, cette densité a une forme *gibbsienne*, c'est-à-dire

$$(\forall \mathbf{u} \in \mathcal{X}) \quad \Pi(\mathbf{u}) = \frac{1}{z} e^{-\beta R(\mathbf{u})}, \quad (3.21)$$

où  $z$  est une constante de normalisation. Le terme  $\beta R(\mathbf{u})$  s'appelle alors *énergie de l'a priori*  $\Pi$ .

Dans la deuxième étape, on précise le modèle de dégradation. Il s'agit d'un modèle probabiliste reliant le signal observé  $\mathbf{f}$  au signal inconnu  $\mathbf{u}_0$ . Formellement, ceci revient à se donner la loi conditionnelle de  $\mathbf{f}$  sachant  $\mathbf{u}_0$ . Dans le cas d'un bruit blanc gaussien cette loi est donnée par :

$$(\forall \mathbf{u} \in \mathcal{X}) \quad \Pi(\mathbf{f}|\mathbf{u}) = \left( \frac{1}{2\pi\sigma^2} \right)^{N/2} \exp \left\{ -\frac{1}{2\sigma^2} \|\mathbf{f} - \mathbf{u}\|^2 \right\}. \quad (3.22)$$

L'objet de la troisième étape est de *mettre à jour* la densité a priori  $\Pi$  au vu des observations que l'on a obtenues. Formellement, il s'agit de calculer la

densité de la loi conditionnelle de  $\mathbf{u}$  sachant  $\mathbf{f}$ , appelée *densité a posteriori*. La formule de Bayes donne alors

$$(\forall \mathbf{u} \in \mathcal{X}) \quad \Pi(\mathbf{u}|\mathbf{f}) = \frac{\Pi(\mathbf{f}|\mathbf{u})\Pi(\mathbf{u})}{\Pi(\mathbf{f})}. \quad (3.23)$$

La densité de probabilité  $\Pi(\mathbf{u}|\mathbf{f})$  combine maintenant l'a priori  $\Pi$  et le modèle de bruit.

La quatrième étape est une étape d'estimation ponctuelle qui consiste à sélectionner *une seule image*  $\hat{\mathbf{u}}$  à partir de la densité a posteriori  $\Pi(\mathbf{u}|\mathbf{f})$ . Nous considérons ici l'estimateur du maximum a posteriori (MAP) qui consiste à sélectionner  $\hat{\mathbf{u}}$  parmi les maximiseurs de  $\Pi(\mathbf{u}|\mathbf{f})$  :

$$\hat{\mathbf{u}} \in \underset{\mathbf{u} \in \mathcal{X}}{\text{Argmax}} \Pi(\mathbf{u}|\mathbf{f}) = \underset{\mathbf{u} \in \mathcal{X}}{\text{Argmax}} \Pi(\mathbf{f}|\mathbf{u})\Pi(\mathbf{u}) = \underset{\mathbf{u} \in \mathcal{X}}{\text{Argmin}} -\ln(\Pi(\mathbf{f}|\mathbf{u})\Pi(\mathbf{u})). \quad (3.24)$$

En remplaçant par les expressions (3.21) et (3.22) on arrive à

$$\hat{\mathbf{u}} \in \underset{\mathbf{u} \in \mathcal{X}}{\text{Argmin}} \lambda R(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|^2, \quad (3.25)$$

où  $\lambda = \beta\sigma^2$ . Remarquons que si  $R \in \Gamma_0(\mathcal{X})$ , le problème (3.25) admet comme une unique solution  $\text{prox}_{\lambda R}(\mathbf{f})$ .

Enfin, la cinquième étape est une étape d'optimisation du critère (3.25), généralement de manière approchée, en construisant une suite minimisante.

Remarquons que si l'on fixe le modèle de bruit et le choix de l'estimateur ponctuel, seul le choix de la densité a priori subsiste.

Nous examinons dans la sous-section suivante deux exemples classiques d'a priori utilisés en traitement de l'image. Nous généraliserons plus tard ces deux a priori aux signaux sur graphes en utilisant les opérateurs de la section 2.4.

### 3.3.3 A priori classiques en traitement de l'image

Les a priori classiquement utilisés en traitement de l'image peuvent être regroupés en deux familles. La première comporte les a priori basés sur la régularité de l'image sous-jacente. Ces a priori pénalisent différentes fonctions dépendant d'une norme du gradient. Les deux exemples classiques sont l'a priori de Sobolev [TA] et celui de la variation totale [ROF92]. Dans le cadre discret de ce manuscrit nous adoptons la discrétisation suivante de l'opérateur gradient :  $\nabla : \mathcal{X} \rightarrow \mathcal{Y}$  :

$$(\nabla \mathbf{u})_{l,k}^1 = \begin{cases} \mathbf{u}_{l,k+1} - \mathbf{u}_{l,k}, & \text{si } 1 \leq k \leq m-1, \\ 0, & \text{si } k = m, \end{cases} \quad (3.26)$$

$$(\nabla \mathbf{u})_{l,k}^2 = \begin{cases} \mathbf{u}_{l+1,k} - \mathbf{u}_{l,k}, & \text{si } 1 \leq l \leq n-1, \\ 0, & \text{si } l = n. \end{cases} \quad (3.27)$$

La seconde famille d'a priori concerne ceux favorisant la *parcimonie* des images dans diverses bases ou repères obliques (frames en anglais) [Mal09]. Les exemples classiques sont la base de Fourier et diverses bases d'ondelettes. Nous nous limiterons dans ce chapitre à la première famille d'a priori.

Nous ne détaillons pas dans cette section les algorithmes d'optimisation utilisés. Certains algorithmes seront détaillés dans la section 3.4 dans le cadre des signaux sur graphes.

### A priori de Sobolev

L'énergie de l'a priori de Sobolev est donnée par  $\beta J_2$ , où  $\beta > 0$  et

$$J_2(\mathbf{u}) = \sum_{l=1}^n \sum_{k=1}^m \|(\nabla \mathbf{u})_{l,k}\|^2. \quad (3.28)$$

Cet a priori a été introduit par Tikhonov dans [TA] dans le cadre de la régularisation de problèmes inverses mal posés. Remarquons à ce point que l'énergie  $J_2$  est une forme quadratique reliée à une version discrète de l'opérateur laplacien. En effet,

$$J_2(\mathbf{u}) = \langle \nabla \mathbf{u}, \nabla \mathbf{u} \rangle_{\mathcal{Y}} = \langle \mathbf{u}, \nabla^* \nabla \mathbf{u} \rangle_{\mathcal{Y}}, \quad (3.29)$$

où  $\nabla^*$  désigne l'opérateur adjoint de  $\nabla$  relativement aux produits scalaires  $\langle \cdot, \cdot \rangle_{\mathcal{X}}$  et  $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$ . En posant  $\text{div} = -\nabla^*$  et  $\Delta = \text{div} \circ \nabla$  on obtient alors

$$J_2(\mathbf{u}) = -\langle \Delta \mathbf{u}, \mathbf{u} \rangle_{\mathcal{X}}. \quad (3.30)$$

En adoptant l'a priori  $J_2$ , l'estimateur  $\hat{\mathbf{u}}$  est obtenu comme solution du problème suivant :

$$\underset{\mathbf{u} \in \mathcal{X}}{\text{Minimiser}} \quad \lambda J_2(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathcal{X}}^2, \quad (3.31)$$

où  $\lambda = \beta \sigma^2$ . Résoudre (3.31) revient donc à minimiser la forme quadratique suivante :

$$Q_2(\mathbf{u}) = \frac{1}{2} \langle \mathbf{A} \mathbf{u}, \mathbf{u} \rangle_{\mathcal{X}} - \langle \mathbf{u}, \mathbf{f} \rangle_{\mathcal{X}}, \quad (3.32)$$

où

$$\mathbf{A} = \mathbf{I} - 2\lambda \Delta. \quad (3.33)$$

Il est aisé de vérifier que l'opérateur  $\mathbf{A}$  est défini positif. Par conséquent, le problème (3.31) admet une unique solution  $\hat{\mathbf{u}}$ , elle-même solution du système linéaire

$$\mathbf{A}\mathbf{u} = \mathbf{f}. \quad (3.34)$$

Remarquons que le système linéaire (3.34) correspond à celui obtenu par une discrétisation de l'équation aux dérivées partielles

$$u - \lambda\Delta u = f, \quad (3.35)$$

avec conditions aux bords de Neumann [AK06].

Dans la pratique, la solution  $\hat{\mathbf{u}}$  du problème (3.31) ne permet pas d'obtenir un compromis satisfaisant entre la réduction de bruit et la préservation des contours des objets présents dans l'image originale. Nous illustrons rapidement le comportement de la solution de (3.31).

La première colonne de la figure 3.1 représente les images de test que nous avons sélectionnées. La deuxième colonne représente les versions artificiellement bruitées de ces images. Nous avons considéré un bruit blanc gaussien d'écart-type  $\sigma = 20$ .

La première colonne de la figure 3.2 représente les estimateurs  $\hat{\mathbf{u}}$  obtenus avec (3.31). Nous avons systématiquement optimisé la valeur du paramètre  $\lambda$  : les images que nous montrons sont les meilleures, en terme de PSNR<sup>1</sup>.

### A priori basés sur la variation totale

L'utilisation de la semi-norme de la variation totale (TV en abrégé) a été initiée en traitement de l'image dans [ROF92]. Dans le cadre discret de ce manuscrit, nous considérons la semi-norme suivante

$$J_1(\mathbf{u}) = \sum_{i,j=1}^n \|(\nabla\mathbf{u})_{i,j}\|, \quad (3.36)$$

où  $\nabla$  est l'opérateur de différence défini dans (3.26)-(3.27).

Remarquons à ce point que d'autres versions discrètes de la semi-norme TV peuvent être considérées. De telles variantes peuvent être obtenues soit en modifiant la discrétisation (3.26)-(3.27) du gradient, soit en modifiant la norme intervenant dans (3.36). Un exemple de la première approche peut être trouvé dans [CLL11] où les auteurs proposent une version basée sur un schéma aux différences finies décentré. Le lecteur intéressé pourra trouver dans [WL11] une étude de l'erreur entre la solution du problème continu introduit dans [ROF92]

---

1. Peak Signal-to-Noise Ratio.



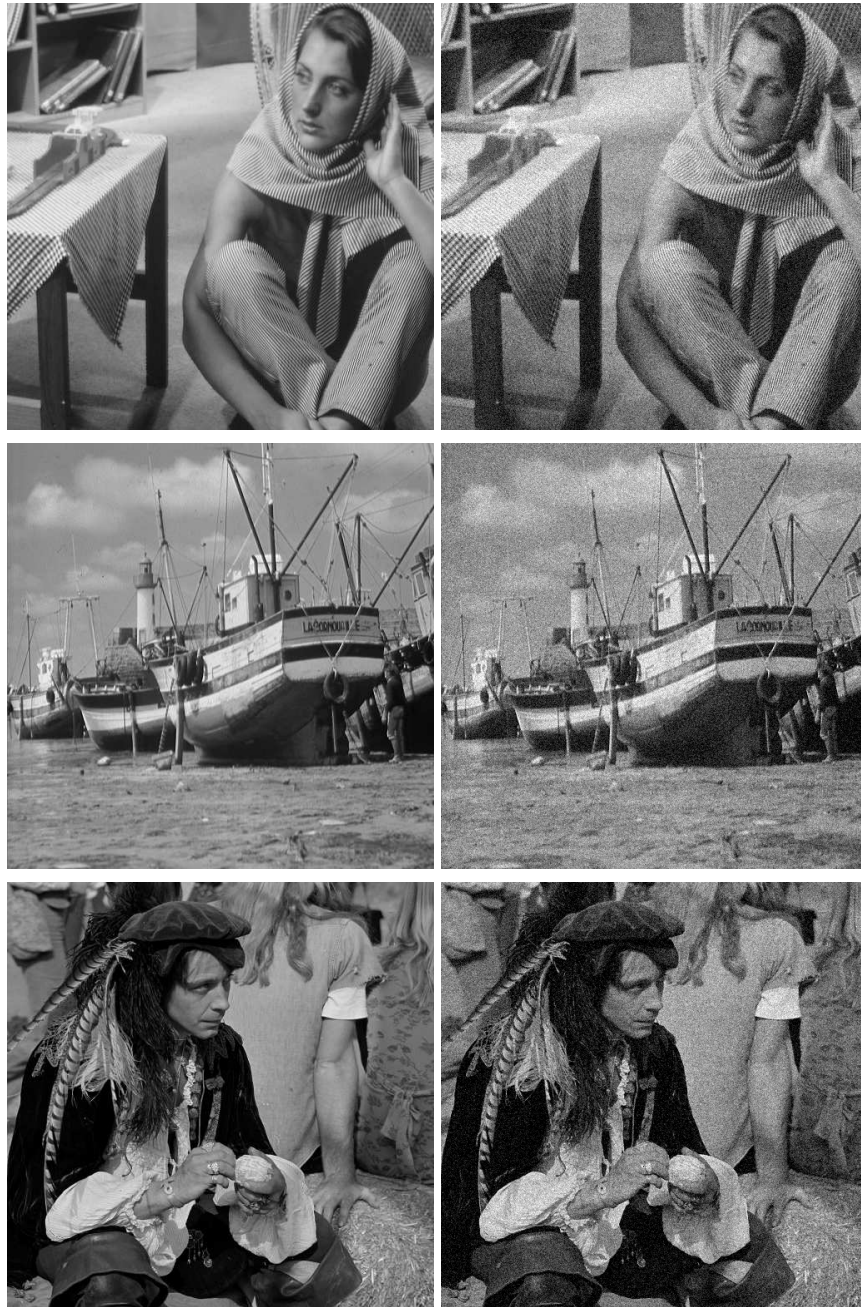
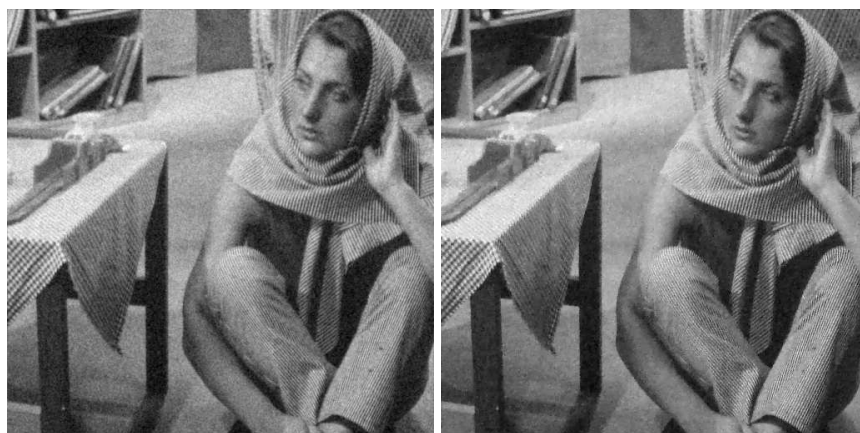
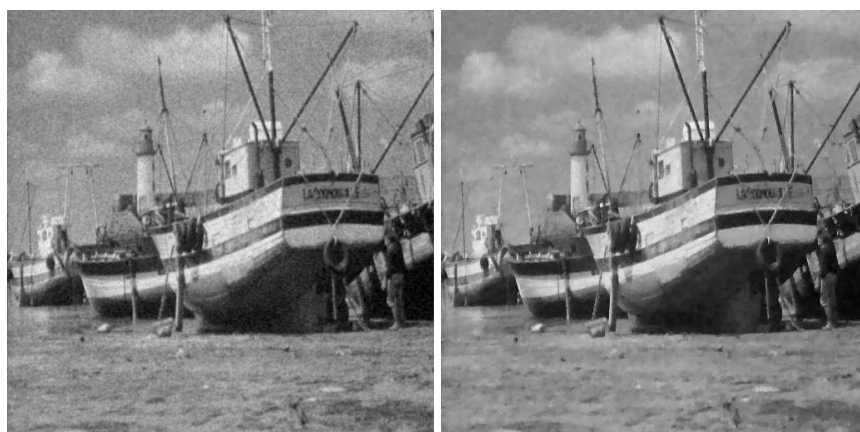


FIGURE 3.1 – Images de test et leur version bruitée avec un bruit blanc gaussien d'écart-type  $\sigma = 20$ .



PSNR = 26.05

PSNR = 26.90



PSNR = 28.04

PSNR = 29.16



PSNR = 27.13

PSNR = 28.24

FIGURE 3.2 – Images débruitées par minimisation d'énergie. La colonne de gauche présente les résultats obtenus avec l'a priori  $J_2$ . Celle de droite présente les résultats obtenus avec une pénalisation TV. Les images montrées sont obtenues pour des valeurs de  $\lambda$  optimales en terme de PSNR.

et les solutions correspondant à divers schémas aux différences finies. L'exemple standard obtenu avec la deuxième approche est la TV anisotrope obtenue en remplaçant la norme  $\ell_2$  dans (3.36) par une norme  $\ell_1$ . Cette approche a été considérée dans [Cha05, BT09a]. La semi-norme TV anisotrope a été récemment considérée dans [CVGO10] pour le débruitage d'images de codes-barres. Le lecteur intéressé pourra consulter [CCC+10] pour une introduction aux divers aspects de la TV ainsi qu'à son utilisation en traitement de l'image.

Posons, pour  $\mathbf{u} \in \mathcal{X}$  et  $\lambda > 0$

$$E_1(\mathbf{u}) = \lambda J_1(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathcal{X}}^2. \quad (3.37)$$

Adopter la TV comme a priori amène à considérer le problème suivant (appelé ROF dans la suite) :

$$\underset{\mathbf{u} \in \mathcal{X}}{\text{Minimiser}} \quad E_1(\mathbf{u}). \quad (3.38)$$

Notons que  $\lambda J_1 \in \Gamma_0(\mathcal{X})$ . Par conséquent, le problème (3.38) admet une unique solution et il s'agit de  $\text{prox}_{\lambda J_1}$ .

L'énergie  $E_1$  n'est pas différentiable. Dans [Cha04], Chambolle fournit une caractérisation de la solution  $\hat{\mathbf{u}}$  du problème (3.38) ainsi qu'un algorithme pour le calcul de cette solution. Comme remarqué dans [CW05], la caractérisation donnée dans [Cha04] peut être obtenue en appliquant la décomposition de Moreau de la manière suivante :

$$\begin{aligned} f &= \text{prox}_{\lambda J_1}(\mathbf{f}) + \lambda \text{prox}_{J_1^*/\lambda}(\mathbf{f}/\lambda), \\ &= \hat{\mathbf{u}} + \lambda \text{prox}_{J_1^*/\lambda}(\mathbf{f}/\lambda), \end{aligned} \quad (3.39)$$

où  $J_1^*$  désigne la fonction conjuguée de  $J_1$ . L'expression de  $J_1^*$  est donnée dans [Cha04] par

$$J_1^* = \iota_K, \quad (3.40)$$

où

$$K = \{\text{div} \mathbf{p}, \|p_{i,j}\| \leq 1, \forall i, j \in \{1, \dots, n\}\}, \quad (3.41)$$

et

$$(\forall i, j \in \{1, \dots, n\}) \quad \|p_{i,j}\| = \sqrt{(p_{i,j}^1)^2 + (p_{i,j}^2)^2}. \quad (3.42)$$

La colonne de droite de la figure 3.2 montre les résultats obtenus avec cette approche. Ici aussi, la valeur de  $\lambda$  a été optimisée<sup>1</sup> dans la page Nos résultats sont obtenus avec la fonction **tvdenoise** fournie avec le package **tvreg** disponible à l'adresse <http://www.getreuer.info/home/tvreg..> On peut constater que les résultats obtenus sont meilleurs qu'avec un a priori de Sobolev.

---

1. d

### 3.4 Approches variationnelles pour le débruitage de signaux sur graphes

Nous généralisons à des signaux définis sur l'ensemble des sommets d'un graphe pondéré d'ordre fini la méthodologie rappelée dans la section précédente. L'application de cette extension au débruitage d'images (vues comme des signaux sur graphes), de maillages (avec et sans attributs de couleur) sera considérée dans la section suivante.

Afin de pouvoir adopter ici une approche bayésienne, nous avons besoin de définir des a priori pour les signaux définis sur un graphe donné. Les a priori que nous considérons ici ne dépendent que de la topologie de  $G$  et de la fonction de poids  $w$ . Ainsi, on considère fixée dans toute la suite de cette section une matrice d'adjacence pondérée  $\mathbf{W} \in \mathbb{R}^{N \times N}$  représentant un graphe connexe sur  $\{1, \dots, N\}$ . Si le graphe n'est pas connexe, on se ramène à l'hypothèse de connexité en considérant chaque composante connexe séparément.

En vue de l'étape d'optimisation qui va suivre, nous considérons les éléments de  $\mathcal{H}(V)$  comme des vecteurs de l'espace  $\mathcal{X} = \mathbb{R}^N$  muni du produits scalaire

$$\langle \mathbf{f}, \mathbf{g} \rangle_{\mathcal{X}} = \sum_{i=1}^N f_i g_i,$$

et de la norme associée  $\|\cdot\|_{\mathcal{X}} = \sqrt{\langle \cdot, \cdot \rangle_{\mathcal{X}}}$ .

Nous modifions légèrement les notations des opérateurs  $\nabla^w$  et  $\text{div}^w$  du chapitre précédent et adoptons celles que nous avons introduites dans [HLE12]. Posons  $\mathcal{Y} = \mathbb{R}^{N \times N}$  muni du produit scalaire

$$\langle \mathbf{F}, \mathbf{G} \rangle_{\mathcal{Y}} = \sum_{i,j=1}^N f_{i,j} g_{i,j},$$

et de la norme associée  $\|\cdot\|_{\mathcal{Y}} = \sqrt{\langle \cdot, \cdot \rangle_{\mathcal{Y}}}$ . Nous désignerons par  $\mathbf{F}_{i,\cdot}$  la  $i^{\text{ème}}$  ligne de la matrice  $\mathbf{F}$  et par  $\mathbf{F}_{\cdot,j}$  la  $j^{\text{ème}}$  colonne. Sur  $\mathcal{Y}$  nous considérons aussi la famille de normes mixtes suivante

$$(\forall p, q \geq 1) \quad \|\mathbf{F}\|_{p,q} = \left( \sum_{i=1}^N (\|\mathbf{F}_{i,\cdot}\|_q)^p \right)^{1/p}, \quad (3.43)$$

si  $p < \infty$ , et

$$(\forall q \geq 1) \quad \|\mathbf{F}\|_{\infty,q} = \max_{i \in \{1, \dots, N\}} \|\mathbf{F}_{i,\cdot}\|_q. \quad (3.44)$$

La norme mixte  $\ell_{p,q}$  de  $\mathbf{F}$  est donc simplement la norme  $\ell_p$  du vecteur dont les composantes sont les normes  $\ell_q$  des vecteurs lignes de  $\mathbf{F}$ . La norme duale de

$\|\cdot\|_{p,q}$  est donnée par  $\|\cdot\|_{p^*,q^*}$  où  $p^*$  et  $q^*$  désignent les exposants conjugués de  $p$  et  $q$ , respectivement.

Nous réécrivons aussi l'opérateur de différence  $\nabla^w$  et son adjoint défini dans le chapitre précédent sous forme matricielle :  $\nabla^w : \mathcal{X} \rightarrow \mathcal{Y}$

$$(\nabla^w \mathbf{f})_{i,j} = \sqrt{w_{i,j}}(f_j - f_i). \quad (3.45)$$

L'opérateur adjoint  $\nabla^{w,*} : \mathcal{Y} \rightarrow \mathcal{X}$  vérifiant

$$\langle \nabla^w \mathbf{f}, \mathbf{G} \rangle_{\mathcal{Y}} = \langle \mathbf{f}, \nabla^{w,*} \mathbf{G} \rangle_{\mathcal{X}}, \quad (3.46)$$

pour tous  $(\mathbf{f}, \mathbf{G}) \in \mathcal{X} \times \mathcal{Y}$  est donné par

$$(\nabla^{w,*} \mathbf{G})_i = \sum_{j=1}^N \sqrt{w_{i,j}}(g_{j,i} - g_{i,j}). \quad (3.47)$$

L'opérateur divergence  $\operatorname{div}^w : \mathcal{Y} \rightarrow \mathcal{X}$  est égal à  $-\nabla^{w,*}$  :

$$(\operatorname{div}^w \mathbf{G})_i = \sum_{j=1}^N \sqrt{w_{i,j}}(g_{i,j} - g_{j,i}). \quad (3.48)$$

L'opérateur  $\Delta^w : \mathcal{X} \rightarrow \mathcal{X}$ , défini par  $\Delta^w = -\frac{1}{2}\operatorname{div}^w \circ \nabla^w$ , correspond au laplacien non normalisé :

$$(\Delta^w \mathbf{f})_i = (\mathbf{L}\mathbf{f})_i = \sum_{j=1}^N w_{i,j}(f_i - f_j). \quad (3.49)$$

Enfin, les mesures de régularité du chapitre précédent se réécrivent sous la forme plus générale

$$\begin{aligned} J_{p,q}^w(\mathbf{f}) &= \frac{1}{p} (\|\nabla^w \mathbf{f}\|_{p,q})^p \\ &= \frac{1}{p} \sum_{i=1}^N \left( \sum_{j=1}^N w_{i,j}^{q/2} |f_i - f_j|^q \right)^{p/q}. \end{aligned} \quad (3.50)$$

Nous nous intéressons dans ce chapitre aux cas  $(p, q) \in \{(2, 2), (1, 2), (1, 1)\}$ . Le premier cas correspond à un a priori quadratique associé au laplacien non normalisé. Les deux derniers correspondent à des a priori de type variation totale.

Reprenons enfin le modèle de dégradation de la section précédente. Soit  $\mathbf{u}_0, \mathbf{f} \in \mathcal{X}$  deux signaux sur le graphe de matrice d'adjacence  $\mathbf{W}$ . Nous supposons que  $\mathbf{u}_0$  et  $\mathbf{f}$  sont reliés par le modèle de dégradation suivant :

$$\mathbf{f} = \mathbf{u}_0 + \boldsymbol{\eta}. \quad (3.51)$$

Ici aussi,  $\boldsymbol{\eta}$  désigne un vecteur aléatoire sur un espace probabilisé  $(\Omega, \mathcal{A}, \mathbb{P})$ . Comme dans la section précédente, nous nous limitons au modèle du bruit blanc gaussien. Le but est de construire un estimateur  $\hat{\mathbf{u}}$  de  $\mathbf{u}_0$ .

### 3.4.1 A priori quadratique

#### Formulation

L'équivalent de l'a priori de Sobolev pour les signaux sur  $G$  a été introduit dans le chapitre précédent. Avec les présentes notations il est donné par

$$\begin{aligned} J_{2,2}^w(\mathbf{u}) &= \frac{1}{2} \sum_{i,j=1}^N w_{i,j} (u_i - u_j)^2. \\ &= \langle \mathbf{L}\mathbf{u}, \mathbf{u} \rangle_{\mathcal{X}}, \end{aligned} \tag{3.52}$$

où  $\mathbf{L}$  désigne la matrice du laplacien non normalisé.

Posons pour  $\mathbf{u} \in \mathcal{X}$  et  $\lambda > 0$

$$E_{2,2}^w(\mathbf{u}; \mathbf{f}, \lambda) = \lambda J_{2,2}^w(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathcal{X}}^2. \tag{3.53}$$

Considérer, dans une approche bayésienne, un a priori dont l'énergie est proportionnelle à  $J_{2,2}^w$  amène à poser le problème d'optimisation suivant :

$$\underset{\mathbf{u} \in \mathbb{R}^N}{\text{Minimiser}} \quad E_{2,2}^w(\mathbf{u}; \mathbf{f}, \lambda). \tag{3.54}$$

Le problème (3.54) correspond à un cas particulier des fonctionnelles considérées dans [BEM07, LEB07, ELB08, BEM09] pour la simplification et le débruitage d'images, de maillages et de données en grande dimension. Les auteurs de [GO07] proposent une formulation variationnelle continue dont la discrétisation correspond à (3.54). Ils s'en servent pour le débruitage et la segmentation d'images.

#### Propriétés

Le problème (3.54) revient à minimiser la forme quadratique

$$Q_2^w(\mathbf{u}) = \frac{1}{2} \langle \mathbf{B}\mathbf{u}, \mathbf{u} \rangle - \langle \mathbf{u}, \mathbf{f} \rangle, \tag{3.55}$$

où

$$\mathbf{B} = 2\lambda\mathbf{L} + \mathbf{I}. \tag{3.56}$$

Soit  $\gamma \in \mathbb{R}$ . Notons que

$$\gamma \in \text{Sp}(\mathbf{B}) \iff \frac{\gamma - 1}{2\lambda} \in \text{Sp}(\mathbf{L}).$$

La matrice laplacienne étant symétrique semi-définie positive, nous avons  $\text{Sp}(\mathbf{B}) \subset [1, +\infty[$ , et donc la matrice symétrique  $\mathbf{B}$  est définie positive. Ceci permet d'affirmer que le problème (3.54) admet une unique solution elle-même solution du système linéaire

$$\mathbf{B}\mathbf{u} = \mathbf{f}. \quad (3.57)$$

Notons que la solution du problème (3.54) préserve la moyenne du signal original. En effet,

$$\begin{aligned} (\forall \mathbf{u} \in \mathbb{R}^N) \quad \langle \mathbf{B}\mathbf{u}, \mathbf{1} \rangle &= \langle \mathbf{u}, \mathbf{1} \rangle + 2\lambda \langle \mathbf{L}\mathbf{u}, \mathbf{1} \rangle, \\ &= \langle \mathbf{u}, \mathbf{1} \rangle + 2\lambda \sum_{i,j=1}^N w_{i,j} (u_i - u_j), \\ &= \langle \mathbf{u}, \mathbf{1} \rangle. \end{aligned}$$

Pour la résolution de (3.57) nous avons opté pour des méthodes itératives simples, à savoir la méthode de Jacobi et sa variante Gauss-Seidel.

### Méthode de Jacobi

La méthode de Jacobi [Cia88] permet de construire itérativement la solution d'un système linéaire  $\mathbf{C}\mathbf{u} = \mathbf{f}$ , où  $\mathbf{C} \in \mathbb{R}^{N \times N}$  est une matrice inversible. La méthode repose sur la décomposition :

$$\mathbf{C} = \mathbf{M} - \mathbf{N}, \quad (3.58)$$

où  $\mathbf{M}$  est la matrice diagonale telle que  $m_{i,i} = c_{i,i}$ , pour tout  $i \in \{1, \dots, N\}$ . L'algorithme est le suivant :

$$\begin{cases} \mathbf{u}^{(0)} & \text{donné} \\ \mathbf{u}^{(n+1)} & = \mathbf{M}^{-1}\mathbf{N}\mathbf{u}^{(n)} + \mathbf{M}^{-1}\mathbf{f}. \end{cases} \quad (3.59)$$

La convergence de l'algorithme (3.59) est assurée si et seulement si  $\rho(\mathbf{M}^{-1}\mathbf{N}) < 1$  (Théorème 5.1-1 de [Cia88]). Si la matrice  $\mathbf{C}$  du système linéaire est symétrique définie positive, la condition  $\rho(\mathbf{M}^{-1}\mathbf{N}) < 1$  est vérifiée en particulier si la matrice (forcément symétrique)  $\mathbf{M}^\top + \mathbf{N}$  est définie positive (Théorème 5.3-1 de [Cia88]).

Examinons le cas du système linéaire (3.57). Nous avons  $\mathbf{M} = 2\lambda\mathbf{D} + \mathbf{I}$  et  $\mathbf{N} = 2\lambda\mathbf{W}$ . Ainsi,  $\mathbf{M}^\top + \mathbf{N} = \mathbf{M} + \mathbf{N} = 2\lambda(\mathbf{W} + \mathbf{D}) + \mathbf{I}$ . Il suffit donc de montrer que  $\mathbf{W} + \mathbf{D}$  est symétrique semi-définie positive, ce qui est immédiat.

Écrivons les itérations de l'algorithme (3.59) pour le système (3.57) :

$$\begin{cases} \mathbf{u}^{(0)} \text{ donné} \\ u_i^{(n+1)} = \left( f_i + 2\lambda \sum_{j=1}^N w_{i,j} u_j^{(n)} \right) / (1 + 2\lambda d_{i,i}). \end{cases} \quad (3.60)$$

Avec le choix  $\mathbf{u}^{(0)} = \mathbf{f}$ , les itérations de (3.60) correspondent à un filtre de voisinage itératif dont les poids ne varient pas au cours des itérations.

### Gauss-Seidel

Une variante de (3.60) peut être obtenue en remplaçant la matrice  $\mathbf{M}$  dans (3.58) par  $\mathbf{M}'$  donnée par

$$(\mathbf{M}')_{i,j} = \begin{cases} c_{i,j}, & \text{si } i \leq j, \\ 0, & \text{sinon.} \end{cases}$$

On obtient alors la méthode Gauss-Seidel dont les itérations, dans notre cas, sont les suivantes :

$$\begin{cases} \mathbf{u}^{(0)} \text{ donné} \\ u_i^{(n+1)} = \left( f_i + 2\lambda \sum_{j=1}^{i-1} w_{i,j} u_j^{(n+1)} + 2\lambda \sum_{j=i}^N w_{i,j} u_j^{(n)} \right) / (1 + 2\lambda d_{i,i}). \end{cases} \quad (3.61)$$

### Version normalisée

L'approche bayésienne avec un a priori quadratique peut être développée en considérant l'a priori normalisé suivant :

$$\begin{aligned} \tilde{J}_{2,2}^w(\mathbf{u}) &= \langle \tilde{\mathbf{L}}\mathbf{u}, \mathbf{u} \rangle \\ &= \frac{1}{2} \sum_{i,j=1}^N w_{i,j} \left( \frac{f_i}{d_{i,i}^{1/2}} - \frac{f_j}{d_{j,j}^{1/2}} \right)^2 \end{aligned} \quad (3.62)$$

Le problème d'optimisation correspondant est alors le suivant :

$$\underset{\mathbf{u} \in \mathbb{R}^N}{\text{Minimiser}} \quad \tilde{E}_{2,2}^w(\mathbf{u}; \mathbf{f}, \lambda) \quad (3.63)$$

où

$$\tilde{E}_{2,2}^w(\mathbf{u}; \mathbf{f}, \lambda) = \lambda \tilde{J}_{2,2}^w(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathcal{X}}^2. \quad (3.64)$$



Le système linéaire (3.57) devient alors

$$\tilde{\mathbf{B}}\mathbf{u} = \mathbf{f}, \quad (3.65)$$

où

$$\tilde{\mathbf{B}} = 2\lambda\tilde{\mathbf{L}} + \mathbf{I}, \quad (3.66)$$

et l'algorithme de Jacobi est donné par

$$\begin{cases} \mathbf{u}^{(0)} \text{ donné} \\ u_i^{(n+1)} = \left( f_i + 2\lambda \sum_{j=1}^N \frac{w_{i,j}}{d_{i,i}^{1/2} d_{j,j}^{1/2}} u_j^{(n)} \right) / (1 + 2\lambda d_{i,i}). \end{cases} \quad (3.67)$$

Notons qu'ici aussi, la moyenne du signal original est préservée :

$$\begin{aligned} (\forall \mathbf{u} \in \mathbb{R}^N) \quad \langle \tilde{\mathbf{B}}\mathbf{u}, \mathbf{1} \rangle &= \langle \mathbf{u}, \mathbf{1} \rangle + 2\lambda \langle \tilde{\mathbf{L}}\mathbf{u}, \mathbf{1} \rangle \\ &= \langle \mathbf{u}, \mathbf{1} \rangle + 2\lambda \sum_{i,j=1}^N w_{i,j} \left( \frac{u_i}{d_{i,i}^{1/2}} - \frac{u_j}{d_{j,j}^{1/2}} \right) \\ &= \langle \mathbf{u}, \mathbf{1} \rangle. \end{aligned}$$

L'a priori  $\tilde{J}_2^w$  a été considéré dans divers travaux [ZS04, ZBL+04, ZS05] dans le cadre de la classification semi-supervisée [CSZ+]. Les auteurs de [ZS05] définissent une famille d'a priori dont  $\tilde{J}_2^w$  est un cas particulier.

### 3.4.2 A priori basé sur la variation totale

Nous considérons dans cette section un a priori de type variation totale pour des signaux définis sur  $G$ .

#### Formulation

L'équivalent de l'a priori TV pour les signaux sur  $G$  est donné par  $J_{1,q}^w$ . Nous considérons ici principalement le cas  $q = 2$  qui peut être relié au 1-laplacien isotrope [ELB08]. Le cas  $q = 1$  sera considéré en lien avec les algorithmes d'optimisation combinatoire qui peuvent y être employés ; il peut être relié au 1-laplacien anisotrope [LTE10].

Rappelons que

$$J_{1,2}^w(\mathbf{u}) = \|\nabla^w \mathbf{u}\|_{1,2} = \sum_{i=1}^N \left( \sum_{j=1}^N w_{i,j} (u_i - u_j)^2 \right)^{1/2}. \quad (3.68)$$

Comme nous l'avons vu, choisir, dans une approche bayésienne couplée à un estimateur MAP, un a priori dont l'énergie est proportionnelle à  $J_{1,2}^w$  amène à poser le problème suivant :

$$\underset{\mathbf{u} \in \mathbb{R}^N}{\text{Minimiser}} \quad E_{1,2}^w(\mathbf{u}; \mathbf{f}, \lambda), \quad (3.69)$$

où

$$E_{1,2}^w(\mathbf{u}; \mathbf{f}, \lambda) = \lambda J_{1,2}^w(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathcal{X}}^2, \quad (3.70)$$

et  $\lambda > 0$ .

Le problème (3.69) a été introduit par Osher (chapitre 16 de [Ana00] et [COS01]) dans le cas de graphes non pondérés. Dans [COS01], cette formulation permet d'introduire un filtre itératif, appelé *filtre TV digital*, pour le débruitage d'images. Le cadre de [Ana00] est plus général puisqu'il traite du débruitage de signaux définis sur des graphes. Notons que le modèle de graphe y est motivé par des applications où l'échantillonnage des données est irrégulier.

La formulation avec des coefficients de pondération a été proposée dans [BEM07, LEB07, ELB08, LTE10] pour le débruitage et la simplification d'images, de maillages et de données en grandes dimensions. Dans [GO08], les auteurs proposent une fonctionnelle continue dont la discrétisation est identique à (3.70).

### La TV comme fonction d'appui

Notons que  $J_{1,2}^w \in \Gamma_0(\mathcal{X})$ . Le problème (3.69) admet donc une unique solution  $\hat{\mathbf{u}}$ . Nous caractérisons dans ce qui suit cette solution.

L'a priori  $J_{1,2}^w$  est positivement homogène. Il peut donc s'écrire comme fonction d'appui d'un ensemble convexe  $\mathcal{K}^w$  [HUL01]. L'écriture de  $J_{1,2}^w$  à l'aide des normes mixtes permet d'identifier  $\mathcal{K}^w$ . En effet,

$$\begin{aligned} (\forall \mathbf{u} \in \mathcal{X}^N), \quad J_{1,2}^w(\mathbf{u}) &= \|\nabla^w \mathbf{u}\|_{1,2}, \\ &= \max_{\|\mathbf{F}\|_{\infty,2} \leq 1} \langle \nabla^w \mathbf{u}, \mathbf{F} \rangle_{\mathcal{Y}}, \\ &= \max_{\|\mathbf{F}\|_{\infty,2} \leq 1} \langle \mathbf{u}, \text{div}^w \mathbf{F} \rangle_{\mathcal{X}}, \\ &= \max_{\mathbf{v} \in \mathcal{K}^w} \langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{X}}, \\ &= \sigma_{\mathcal{K}^w}(\mathbf{u}). \end{aligned} \quad (3.71)$$

où

$$\mathcal{K}^w = \{\text{div}^w \mathbf{F} : \mathbf{F} \in \mathcal{Y}, \|\mathbf{F}\|_{\infty,2} \leq 1\}. \quad (3.72)$$

Ainsi, d'après (3.4), nous avons

$$\left( J_{1,2}^w \right)^* = \iota_{\mathcal{K}^w}. \quad (3.73)$$

## Espace $\mathcal{G}^w$

Notons

$$\mathcal{G}^w = \{\operatorname{div}^w \mathbf{F}, \mathbf{F} \in \mathcal{Y}\}. \quad (3.74)$$

Cet espace, que nous introduisons ici, est le pendant discret sur graphes de l'espace  $G$  introduit par Meyer dans [Mey01] pour modéliser les textures et les motifs oscillants. Dans notre cadre, on peut montrer que si le graphe  $G$  est connecté, l'espace  $\mathcal{G}^w$  s'identifie avec le sous-espace  $\mathcal{X}_0$  de  $\mathcal{X}$  constitué de signaux de moyenne nulle. En effet, si  $\mathbf{u} = \operatorname{div}^w \mathbf{F}$ , alors

$$\langle \mathbf{u}, \mathbf{1} \rangle_{\mathcal{X}} = \langle \operatorname{div}^w \mathbf{F}, \mathbf{1} \rangle_{\mathcal{X}} = -\langle \mathbf{F}, \nabla^w \mathbf{1} \rangle_{\mathcal{Y}} = 0. \quad (3.75)$$

Ainsi,  $\mathcal{G}^w \subset \mathcal{X}_0$ . D'autre part, nous avons

$$\operatorname{Im}(\Delta^w) = \operatorname{Im}(\operatorname{div}^w \circ \nabla^w) \subset \operatorname{Im}(\operatorname{div}^w) = \mathcal{G}^w. \quad (3.76)$$

D'où

$$\operatorname{rg}(\Delta^w) \leq \dim(\mathcal{G}^w). \quad (3.77)$$

Rappelons à ce stade que la multiplicité de la valeur propre nulle pour  $\Delta^w$  est égale au nombre de composantes connexes de  $G$ . Le graphe  $G$  étant supposé connexe, nous avons que  $\operatorname{rg}(\Delta^w) = N - 1 = \dim(\mathcal{X}_0)$ . Finalement,  $\mathcal{G}^w \subset \mathcal{X}_0$  et  $\dim(\mathcal{G}^w) \geq \dim(\mathcal{X}_0)$ , d'où  $\mathcal{G}^w = \mathcal{X}_0$ .

Notons que le résultat précédent a été énoncé dans [AABFC05] (proposition 2.2) pour une version discrète de la norme  $G$  issue de la discrétisation des opérateurs différentiels intervenant dans le modèle ROF, à la manière de [Cha04].

Remarquons enfin que le résultat précédent n'est en général pas vrai si l'on remplace l'opérateur de différence  $\nabla^w$  par sa version normalisée.

Nous munissons  $\mathcal{G}^w$  de la norme suivante :

$$(\forall \mathbf{u} \in \mathcal{G}^w) \quad \|\mathbf{u}\|_{\mathcal{G}^w} = \inf\{\|\mathbf{F}\|_{\infty,2}, \mathbf{F} \in \mathcal{Y}, \operatorname{div}^w \mathbf{F} = \mathbf{u}\}. \quad (3.78)$$

L'espace  $\mathcal{K}^w$  s'identifie alors avec la boule unité pour la norme  $\mathcal{G}^w$ , que l'on notera  $B_{\mathcal{G}^w}(1)$ .

### Caractérisation - Problème dual

L'identité de Moreau rappelée dans (3.11) permet de caractériser la solution  $\hat{\mathbf{u}}$  de (3.69). En effet,

$$\begin{aligned}\mathbf{f} &= \text{prox}_{\lambda J_{1,2}^w} \mathbf{f} + \lambda \text{prox}_{(J_{1,2}^w)^*/\lambda}(\mathbf{f}/\lambda) \\ &= \hat{\mathbf{u}} + \lambda \text{prox}_{\iota_{B_{\mathcal{G}^w}(1)}/\lambda}(\mathbf{f}/\lambda) \\ &= \hat{\mathbf{u}} + \text{proj}_{B_{\mathcal{G}^w}(\lambda)}(\mathbf{f})\end{aligned}\tag{3.79}$$

Nous avons ainsi la caractérisation suivante de  $\hat{\mathbf{u}}$  :

$$\hat{\mathbf{u}} = \mathbf{f} - \text{proj}_{B_{\mathcal{G}^w}(\lambda)}(\mathbf{f})\tag{3.80}$$

Cette caractérisation fournit une *dualisation* du problème (3.69) : on associe alors au problème primal (3.69) le problème, dit dual, du calcul de la projection de  $\mathbf{f}$  sur une boule de rayon prédéfini pour la norme  $\|\cdot\|_{\mathcal{G}^w}$ . Ce dernier problème s'énonce comme suit :

$$\underset{\|\mathbf{G}\|_{\infty,2} \leq 1}{\text{Minimiser}} \|\lambda \text{div}^w \mathbf{G} - \mathbf{f}\|_{\mathcal{X}}^2.\tag{3.81}$$

La fonctionnelle (3.69) n'est pas différentiable. Nous détaillons dans ce qui suit certains algorithmes permettant de calculer des solutions approchées.

### Algorithme de projection de Chambolle

L'algorithme de projection de Chambolle a été proposé dans [Cha04]. Cet algorithme construit une suite minimisante convergeant vers la solution d'un problème dual similaire à (3.81). Son adaptation au problème (3.81) est donnée par les itérations suivantes :

$$\begin{cases} \mathbf{G}^{(0)} = \mathbf{0}, \\ (\mathbf{G}^{(n+1)})_{i,j} = \frac{(\mathbf{G}^{(n)})_{i,j} + \tau (\nabla^w(\text{div}^w \mathbf{G}^{(n)} - \mathbf{f}/\lambda))_{i,j}}{1 + \tau \|(\nabla^w(\text{div}^w \mathbf{G}^{(n)} - \mathbf{f}/\lambda))_{i,\cdot}\|_X}. \end{cases}\tag{3.82}$$

Les itérations dans (3.82) convergent si la condition suivante sur  $\tau$  est vérifiée :

$$0 < \tau \leq \frac{1}{\|\text{div}^w\|^2},\tag{3.83}$$

où  $\|\operatorname{div}^w\|$  désigne la norme de l'opérateur divergence subordonnée aux normes  $\|\cdot\|_{\mathcal{X}}$  et  $\|\cdot\|_{\mathcal{Y}}$  :

$$\|\operatorname{div}^w\| = \sup\{\|\operatorname{div}^w \mathbf{G}\|_{\mathcal{X}} : \mathbf{G} \in \mathcal{Y}, \|\mathbf{G}\|_{\mathcal{Y}} \leq 1\}.$$

L'inégalité suivante a été démontrée dans [GO08]

$$\forall \mathbf{G} \in \mathcal{Y}, \|\operatorname{div}^w \mathbf{G}\|_{\mathcal{X}}^2 \leq 4m \|\mathbf{G}\|_{\mathcal{Y}}^2, \quad (3.84)$$

où  $m := \max_{1 \leq i \leq n} d_{i,i}$ . En conséquence, la convergence de l'algorithme (3.82) est assurée si

$$0 < \tau \leq \frac{1}{4m}.$$

Finalement, l'unique solution  $\hat{\mathbf{u}}$  du problème (3.69) est donnée par  $\hat{\mathbf{u}} = \mathbf{f} - \lambda \operatorname{div}^w \mathbf{G}^\infty$ , où  $\mathbf{G}^\infty$  est la limite de la suite  $(\mathbf{G}^n)_{n \geq 0}$  définie dans (3.82).

### Détails d'implémentation des algorithmes du gradient projeté et de celui de Beck et Teboulle appliqués au problème dual

L'algorithme du gradient projeté peut être obtenu comme cas particulier de l'algorithme explicite-implicite. Ici, il s'agit d'appliquer cet algorithme au problème dual (3.81). L'identification avec les notations du problème type (3.13) est comme suit

$$\mathcal{H} \equiv \mathcal{Y}, \quad F_1 \equiv \iota_{B_{\|\cdot\|_{\infty,2}}}, \quad F_2 \equiv \|\lambda \operatorname{div}^w \cdot - \mathbf{f}\|_{\mathcal{X}}^2. \quad (3.85)$$

Afin de pouvoir appliquer l'algorithme du gradient projeté et celui de Beck et Teboulle, nous avons besoin d'évaluer  $\operatorname{prox}_{\gamma F_1}$  et  $\nabla F_2$ . Commençons par le calcul de l'opérateur proximal.

Il s'agit ici de calculer la projection sur la boule unité  $B_{\infty,2}(1)$  pour la norme  $\|\cdot\|_{\infty,2}$ . Cette projection est donnée par

$$(\forall \mathbf{G} \in \mathcal{Y}) \quad \operatorname{proj}_{B_{\infty,2}(1)}(\mathbf{G}) = \tilde{\mathbf{G}}, \quad (3.86)$$

où

$$(\tilde{\mathbf{G}})_{i,j} = \frac{(\mathbf{G})_{i,j}}{\max(1, \|(\mathbf{G})_{i,\cdot}\|_2)}. \quad (3.87)$$

D'autre part, pour l'évaluation du gradient, nous avons

$$(\forall \mathbf{G} \in \mathcal{Y}) \quad F_2(\mathbf{G}) = -\lambda^2 \langle \mathbf{G}, \nabla^w \circ \operatorname{div}^w \mathbf{G} \rangle_{\mathcal{Y}} + 2\lambda \langle \mathbf{G}, \nabla^w \mathbf{f} \rangle_{\mathcal{Y}} + \langle \mathbf{f}, \mathbf{f} \rangle_{\mathcal{X}}. \quad (3.88)$$

D'où

$$(\forall \mathbf{G} \in \mathcal{Y}) \quad \nabla F_2(\mathbf{G}) = -2\lambda^2 (\nabla^w \circ \operatorname{div}^w) \mathbf{G} + 2\lambda \nabla^w \mathbf{f}. \quad (3.89)$$

## Détails d'implémentation de l'algorithme primal-dual de Chambolle et Pock

L'algorithme primal-dual de Chambolle et Pock a été rappelé dans la deuxième section de ce chapitre. Nous détaillons ici son application au problème (3.69). L'identification avec les notations du problème type (3.17) est la suivante :

$$K \equiv \nabla^w, \quad F \equiv \|\cdot\|_{1,2}, \quad G \equiv \frac{1}{2\lambda} \|\cdot - \mathbf{f}\|_{\mathcal{X}}^2.$$

L'adjoint de  $\nabla^w$  est donné par  $-\operatorname{div}^w$ .

Commençons par le calcul de l'opérateur proximal de  $\sigma F^*$ . Les normes  $\|\cdot\|_{1,2}$  et  $\|\cdot\|_{\infty,2}$  étant duales, nous avons, d'après (3.7), que  $F^* = \iota_{B_{\infty,2}(1)}$ . Ainsi,

$$\operatorname{prox}_{\sigma F^*} = \operatorname{prox}_{\sigma \iota_{B_{\infty,2}(1)}} = \operatorname{proj}_{B_{\infty,2}(1)}. \quad (3.90)$$

Nous sommes donc amenés, comme dans les deux algorithmes précédents, à calculer la projection sur la boule unité pour la norme  $\|\cdot\|_{B_{\infty,2}}$ . La formule est donnée dans (3.87).

Le calcul du second opérateur proximal est immédiat puisqu'il s'agit de celui d'une fonctionnelle quadratique. L'expression est donnée par

$$(\forall \mathbf{u} \in \mathcal{X}) \quad \operatorname{prox}_{\tau G}(\mathbf{u}) = \frac{\lambda \mathbf{u} + \tau \mathbf{f}}{\tau + \lambda}. \quad (3.91)$$

La convergence de cet algorithme est assurée si  $\tau \sigma \|\nabla^w\|^2 < 1$ , où  $\|\nabla^w\|$  désigne la norme de  $\nabla^w$  subordonnée aux normes  $\|\cdot\|_{\mathcal{X}}$  et  $\|\cdot\|_{\mathcal{Y}}$ .

## Minimisation par coupures de graphes

Nous montrons ici brièvement comment l'algorithme des  $\alpha$ -expansions [BVZ01] peut être employé pour résoudre de manière approchée le problème ROF pour des signaux sur graphes. La fonctionnelle de pénalisation que nous avons considérée jusqu'à présent est  $J_{1,2}^w$ . Afin de pouvoir utiliser l'algorithme des  $\alpha$ -expansions, nous remplaçons  $J_{1,2}^w$  par  $J_{1,1}^w$  dont l'expression est donnée par

$$(\forall \mathbf{u} \in \mathcal{X}) \quad J_{1,1}^w(\mathbf{u}) = \sum_{i,j=1}^N \sqrt{w_{i,j}} |u_i - u_j|.$$

Nous supposons aussi que les signaux  $\mathbf{u}$  sont *quantifiés*, c'est-à-dire à valeurs dans un ensemble fini  $\mathcal{L}$ . Ici, nous prenons  $\mathcal{L} = \{1, \dots, p\}$ . Le problème (3.69) devient alors

$$\underset{\mathbf{u} \in \mathcal{L}^N}{\operatorname{Minimiser}} E_{1,1}^w(\mathbf{u}; \mathbf{f}, \lambda), \quad (3.92)$$

où

$$E_{1,1}^w = \lambda \sum_{i,j=1}^N \sqrt{w_{i,j}} |u_i - u_j| + \frac{1}{2} \sum_{i=1}^N |u_i - f_i|^2. \quad (3.93)$$

Soit  $\mathbf{u}$  un signal sur  $G$  à valeurs dans  $\mathcal{L}$  et  $\alpha \in \mathcal{L}$ . Une  $\alpha$ -expansion de  $\mathbf{u}$  désigne tout signal  $\mathbf{v}$  tel que

$$\{i \in \{1, \dots, N\} : u_i = \alpha\} \subset \{i \in \{1, \dots, N\} : v_i = \alpha\}. \quad (3.94)$$

Notons  $\mathcal{X}_\alpha(\mathbf{u})$  l'ensemble des  $\alpha$ -expansions de  $\mathbf{u}$ .

Un minimum *local* de (3.92) au sens des expansions est un signal  $\hat{\mathbf{u}}$  tel que

$$(\forall \alpha \in \mathcal{L})(\forall \mathbf{u} \in \mathcal{X}_\alpha(\hat{\mathbf{u}})) \quad E_{1,1}(\hat{\mathbf{u}}) \leq E_{1,1}(\mathbf{u}). \quad (3.95)$$

Dans [BVZ01], les auteurs montrent que si  $\mathbf{u}^*$  désigne un *minimum global* alors il existe  $c > 0$  tel que

$$E_{1,1}(\hat{\mathbf{u}}) \leq cE_{1,1}(\mathbf{u}^*). \quad (3.96)$$

Afin de calculer un minimum local au sens des expansions, l'algorithme 3.1, dit des  $\alpha$ -expansions, a été proposé dans [BVZ01]. La boucle comprenant les instructions 3 à 8 s'appelle un *cycle*. Il a été montré dans [Vek99] que cet algorithme se termine en  $O(N)$  cycles.

L'étape cruciale dans l'algorithme 3.1 est l'instruction 4 consistant à trouver à partir d'un signal  $\mathbf{u}$  et de  $\alpha \in \mathcal{L}$  une  $\alpha$ -expansion de  $\mathbf{u}$  d'énergie  $E_{1,1}^w$  minimale. Les auteurs de [BVZ01] montrent que cette étape peut être implémentée de manière efficace en calculant une *s/t-coupe* minimale sur un graphe particulier. Nous renvoyons le lecteur intéressé à [BVZ01] pour les détails concernant cette partie.

L'algorithme que nous employons ici est générique : il peut être employé pour toute fonction d'énergie de la forme

$$E(\mathbf{f}) = \sum_{u \sim v} V_{u,v}(f_u, f_v) + \sum_{u \in \mathcal{L}} D_u(f_u), \quad (3.97)$$

telle que  $V_{u,v}(\cdot, \cdot)$  soit une distance pour tout  $u, v \in V$ . Pour la fonctionnelle (3.92), des algorithmes plus spécifiques ont été développés par Chambolle dans [Cha05] et Darbon et al. dans [DS06]. Une autre approche a été introduite par Ishikawa dans [Ish03]. L'intérêt de ces approches est qu'elles permettent de résoudre le problème de minimisation de manière exacte.

Un des inconvénients de l'utilisation des algorithmes de coupures de graphes sur les images est que ces graphes ont autant de sommets que l'image a de pixels. Des techniques de réduction de ces graphes commencent à émerger, notamment les travaux récents [ML12a, ML12b].

```

1: Initialization :  $\mathbf{u} \in \mathcal{L}^N$ .
2: success = 0
3: for all  $\alpha \in \mathcal{L}$  do
4:   Find  $\hat{\mathbf{u}} = \underset{\mathbf{u}' \in \mathcal{X}_\alpha(\mathbf{u})}{\operatorname{argmin}} E_{1,1}(\mathbf{u}; \mathbf{f}, \lambda)$ 
5:   if  $E_{1,1}(\hat{\mathbf{u}}; \mathbf{f}, \lambda) < E_{1,1}(\mathbf{u}; \mathbf{f}, \lambda)$  then
6:      $\mathbf{u} \leftarrow \hat{\mathbf{u}}$ ,   success  $\leftarrow 1$ 
7:   end if
8: end for
9: if success = 1 then
10:  goto 2
11: end if
12: Return  $\hat{\mathbf{u}}$ 

```

**Algorithme 3.1:** Algorithme des  $\alpha$ -expansions

## 3.5 Expériences

Nous présentons dans cette section des résultats de débruitage d'images et de simplification de maillages polygonaux.

### 3.5.1 Images

Nous testons les algorithmes basés sur les deux a priori  $J_1^w$  et  $J_2^w$  relativement à différentes configurations de graphes. La première ligne de la figure 3.3 présente l'image de test "Barbara" et une version artificiellement bruitée à l'aide d'un bruit blanc gaussien d'écart-type  $s = 20$ . La deuxième ligne de 3.3 correspond aux résultats obtenus par régularisation quadratique. Chaque image est obtenue en considérant un graphe particulier. L'image (c) est obtenue par régularisation quadratique sur le graphe de 4-voisinage non pondéré. L'image (d) est obtenue à l'aide d'un graphe de 8-voisinage dont les arêtes ont été pondérées par une similarité gaussienne entre pixels. L'image (e) est obtenue à l'aide du graphe des 25 plus proches voisins au sein d'une fenêtre  $21 \times 21$  au sens de la distance  $\ell_2$  entre patchs de taille  $5 \times 5$ . Enfin, l'image (f) est obtenue à l'aide du graphe obtenu en connectant chaque pixel à tous les pixels se trouvant à l'intérieur d'une fenêtre  $11 \times 11$ . Les poids sont calculés en transformant la distance entre patchs de taille  $5 \times 5$  en similarités gaussiennes. Le meilleur PSNR est obtenu avec le graphe des plus proches voisins au sens de la distance entre patchs. Le graphe, dont chaque pixel est connecté à tous les pixels se trouvant à l'intérieur d'une fenêtre, et les poids calculés à partir de patch, fournit également un très bon résultat. Visuellement, ce dernier résultat



semble plus artificiel que celui obtenu par plus proches voisins. Dans tous les cas où des noyaux gaussiens interviennent au niveau des poids, les largeurs des noyaux gaussiens ont été optimisées en terme de PSNR. Le paramètre de régularisation  $\lambda$  a également été optimisé relativement au même critère. Enfin, les résultats sont obtenus à l'aide de l'algorithme de Jacobi dont nous avons tronqué les itérations dès que la variation relative de la norme infinie entre deux itérés est inférieure à  $10^{-6}$ .

La troisième ligne de la figure 3.3 correspond aux résultats obtenus par régularisation basée sur la variation totale. Chaque image est obtenue en considérant un graphe particulier. Relativement aux résultats obtenus par régularisation quadratique, une même colonne correspond au même graphe (obtenu avec les mêmes paramètres, sauf pour les largeurs des noyaux qui ont été optimisées). Ainsi, les images (g), (h), (i), (j) correspondent respectivement au graphe de 4-voisinage non pondéré, au graphe de 8-voisinage pondéré, au graphe des plus proches voisins, et au graphe fenêtre. Ici encore, c'est le résultat obtenu à partir de graphe des plus proches voisins qui permet d'avoir le meilleur PSNR, vient ensuite le graphe basé sur de fenêtres de voisinage. Visuellement, il est difficile de distinguer des différences entre les images (i) et (j). L'algorithme utilisé est celui de Chambolle et Pock (version 1) avec 500 itérations pour chaque image. Dans tous les cas, nous avons vérifié qu'au bout de ce nombre d'itérations la variation relative de l'énergie est inférieure à  $10^{-6}$ .

Remarquons que le passage de l'a priori quadratique à celui de la variation totale améliore le PSNR pour les configurations en 4 et 8 voisinage. Cependant, pour des configurations de types plus proches voisins et graphes fenêtres, les résultats obtenus avec l'a priori quadratique fournissent un meilleur PSNR. L'adaptativité introduite par les poids permet ainsi d'obtenir de très bons résultats en considérant un opérateur linéaire.

### 3.5.2 Maillages

Nous illustrons dans cette sous-section l'utilisation de l'approche de régularisation sur graphes pour la simplification de maillages polygonaux. Un maillage polygonal fournit intrinsèquement un graphe (non pondéré) dont les sommets appartiennent à  $\mathbb{R}^3$ . Il est possible d'introduire une pondération des arêtes, par exemple en utilisant l'inverse de la distance euclidienne ( $w \equiv 1/d$ ). La figure 3.4 montre les résultats obtenus pour différentes valeurs du paramètre de régularisation  $\lambda$ . La deuxième ligne correspond aux résultats obtenus en considérant  $J_2^w$  comme fonction de pénalisation. La troisième ligne correspond aux résultats obtenus en considérant  $J_1^w$  comme fonction de pénalisation.

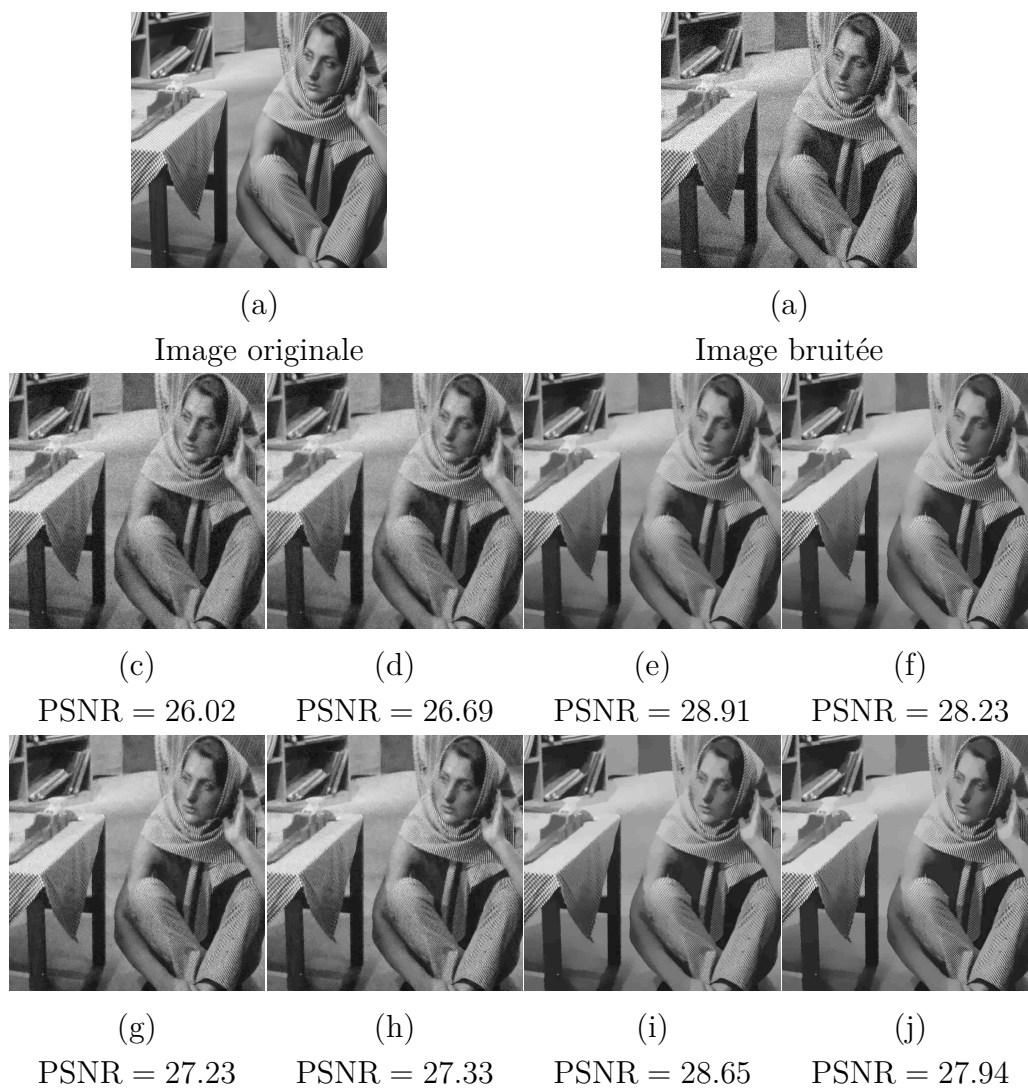


FIGURE 3.3 – Comparaison des résultats obtenus sur le débruitage de “Barbara” relativement à différentes configurations de graphes et deux différentes énergies.

Chaque composante à été simplifiée indépendamment des autres, la valeur de  $\lambda$  étant commune à toutes les composantes.

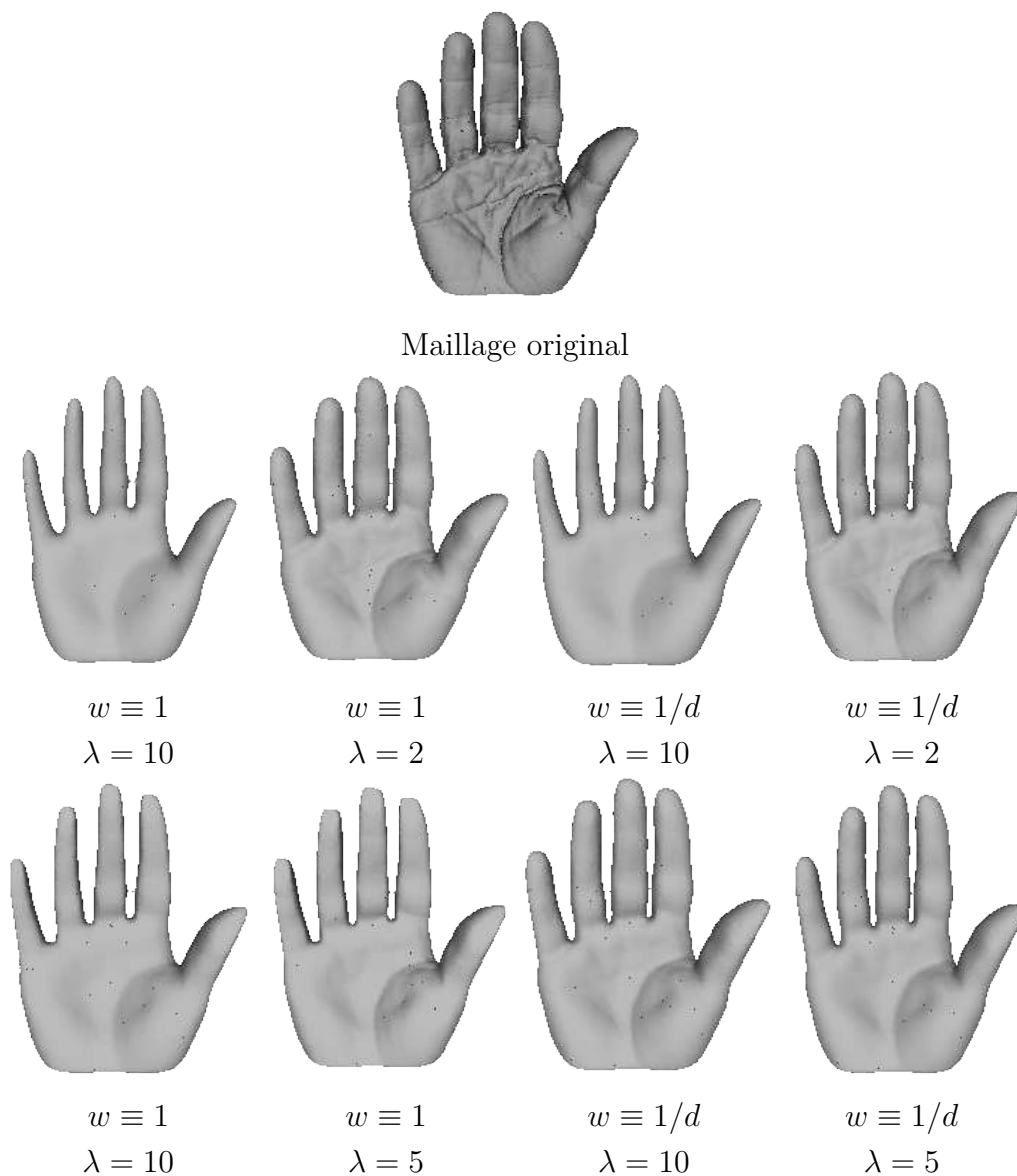


FIGURE 3.4 – Comparaison de la simplification d'un maillage triangulaire obtenue par deux énergies différentes et deux fonctions de poids différentes.

La figure précédente a été obtenue en manipulant les coordonnées associées à chaque sommet. À présent, le maillage original de la figure 3.4 inclut trois coordonnées de couleur (R,G,B) en chaque sommet. C'est cette information que l'on considère pour la simplification. La deuxième ligne est obtenue à partir de

l'a priori  $J_2^w$ . La troisième ligne est obtenue à partir de l'a priori  $J_1^w$ . Dans chaque ligne, le paramètre  $\lambda$  varie ainsi que la fonction de poids. La fonction de poids est ici égale à l'inverse de la distance euclidienne entre les coordonnées *spatiales* des sommets.

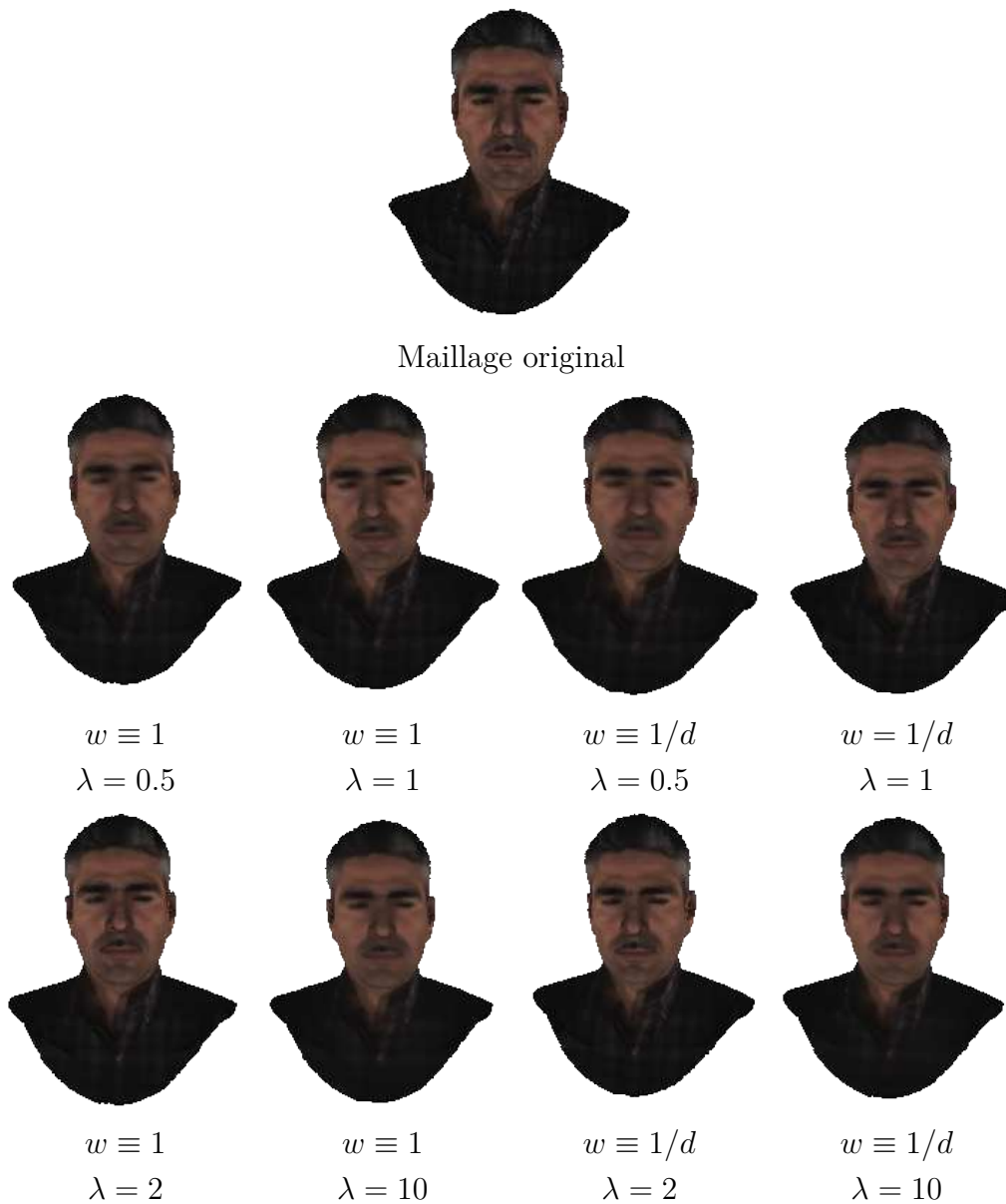


FIGURE 3.5 – Comparaison de la simplification des attributs de couleur d'un maillage triangulaire obtenue par deux énergies différentes et deux fonctions de poids différentes.

## 3.6 Conclusion

Nous avons abordé dans ce chapitre des méthodes variationnelles pour le débruitage de signaux définis sur l'ensemble des sommets d'un graphe pondéré général. Nous avons rappelé quelques éléments d'analyse et d'optimisation convexes utiles pour la suite, ainsi que le cadre de l'analyse bayésienne et les a priori classiques en traitement de l'image. Ces a priori ont été ensuite transposés sur des graphes, et ont donné lieu à deux fonctions d'énergie : l'une quadratique, basée sur les matrices laplaciennes, l'autre non différentiable, de type variation totale (TV). Nous avons caractérisé la solution du problème basé sur la pénalisation TV sur graphes. Cette caractérisation a fait apparaître un équivalent de l'espace  $G$  introduit par Meyer dans [Mey01]. Nous avons introduit cet espace et l'avons caractérisé à son tour. Nous avons également détaillé l'application de trois algorithmes proximaux récents au problème TV- $\ell_2$  sur graphes. Nous avons enfin effectué des tests de débruitage sur des images et de simplification de maillages avec attributs géométriques et colorimétriques. Les notions introduites dans ce chapitre sont à la base des décompositions multi-échelles que nous introduisons dans le chapitre suivant.

# Chapitre 4

## Décompositions multi-échelles hiérarchiques de signaux sur graphes

### 4.1 Introduction

Nous introduisons dans ce chapitre deux décompositions multi-échelles de signaux définis sur l'ensemble des sommets d'un graphe pondéré général. Ces décompositions sont basées sur un principe de régularisation itérative, introduit dans [TNV04], et que nous adaptons et généralisons à des signaux sur graphes. Nous étudions les propriétés des représentations obtenues et en proposons une application en *rehaussement de détails* pour les images numériques et les maillages 3D.

Ce chapitre est organisé de la manière suivante. Nous rappelons dans la section 4.2 des approches classiques de décompositions *mono-échelles* proposées pour les images et les maillages. Il s'agit de décompositions de la forme  $\mathbf{f} = \mathbf{u} + \mathbf{v}$ . L'intérêt de ces décompositions est révélé à travers des exemples d'applications en interpolation de données, en segmentation d'images, et en rehaussement de détails. Dans la section 4.3, on rappelle quelques approches permettant de générer des décompositions *multi-échelles* pour les images et les maillages. Nous présentons en particulier l'approche hiérarchique de Tadmor, Nezzar et Vese [TNV04] pour la décomposition d'images. Cette approche est transposée au cadre des signaux sur graphes à l'aide de l'a priori quadratique du chapitre précédent dans la section 4.4, et à l'aide de l'a priori TV sur graphes dans la section 4.5. Les propriétés de chacune des décompositions introduites sont étudiées selon deux axes : convergence des représentations et choix des paramètres d'échelle. Pour la décomposition basée sur l'a priori TV, le choix du

premier paramètre peut être caractérisé. Cette caractérisation fait apparaître la norme  $G^w$  définie dans le chapitre précédent. Nous proposons de la calculer à l'aide de l'algorithme primal-dual de Chambolle et Pock [CP11a]. Pour la décomposition basée sur l'a priori TV, nous énonçons également un résultat de *décomposition d'énergie*. Nous montrons dans la section 4.6 des exemples de décompositions obtenues avec l'a priori TV. La section 4.7 est dédiée à l'application de notre méthode au rehaussement de détails pour les images numériques et les maillages. Pour les maillages, nous considérons aussi bien le rehaussement de détails *géométriques* que celui de détails *photométriques* associés aux sommets du maillage. Une conclusion de ce chapitre est présentée dans la section 4.8.

Dans tout ce chapitre,  $G$  désigne un graphe pondéré sur  $\{1, \dots, N\}$  de matrice d'adjacence  $\mathbf{W} \in \mathbb{R}^{N \times N}$ .

## 4.2 Décompositions $u + v$

Nous considérons dans cette section des décompositions d'un signal  $\mathbf{f}$  sous la forme

$$\mathbf{f} = \mathbf{u} + \mathbf{v}. \tag{4.1}$$

Un premier exemple d'une telle décomposition a déjà été considéré dans le chapitre précédent : il s'agit du débruitage, qui apparaît ici comme un cas particulier de décomposition. Dans ce contexte, le signal  $\mathbf{u}$  correspond à un estimateur d'un signal "parfait" non observé, et  $\mathbf{v}$  correspond au bruit résiduel, du point de vue de l'algorithme utilisé et de ses paramètres.

Au-delà du cadre du débruitage, on peut être amené à considérer des décompositions de la forme (4.1) dans divers contextes, selon l'interprétation que l'on fait des signaux  $\mathbf{u}$  et  $\mathbf{v}$ . Le plus souvent, il s'agit de *simplifier* le signal  $\mathbf{f}$  ou d'en extraire des informations spécifiques, en vue du traitement, de l'analyse, de l'approximation ou de la compression.

Nous présentons dans cette section quelques exemples classiques de décompositions ainsi que certaines de leurs applications en traitement du signal, de l'image et des maillages et modèles 3D.

### Décomposition en hautes et basses fréquences

L'exemple canonique de décomposition en traitement du signal et de l'image est celui de la décomposition en hautes et basses fréquences. Si  $\mathbf{f} \in \mathbb{R}^N$  désigne un signal unidimensionnel, correspondant à l'échantillonnage régulier d'une fonction sur un intervalle borné, on obtient une composante basses fréquences

$\mathbf{u}$  en appliquant à  $f$  un filtre passe-bas linéaire et invariant [GW99, OS75]. La différence  $\mathbf{v} = \mathbf{f} - \mathbf{u}$  correspond alors à une composante hautes fréquences. Le filtrage numérique correspond dans ce cadre à une convolution, pouvant être implémentée ou analysée dans le domaine spectral à l'aide de la transformée de Fourier rapide.

Un exemple simple de décomposition est obtenu en considérant l'approximation linéaire à  $M$  termes. Dans ce cas, la composante basses fréquences est donnée par

$$\mathbf{u} = \frac{1}{N} \sum_{n=0}^{M-1} \langle \mathbf{f}, \mathbf{e}_n \rangle \mathbf{e}_n,$$

où  $(\mathbf{e}_n)_l = e^{i2\pi nl/N}$ , pour tout  $l \in \{0, \dots, N-1\}$ .

La représentation  $\mathbf{f} = \mathbf{u} + \mathbf{v}$  que l'on obtient ici est adaptée à des signaux stationnaires [Mal09]. Les applications concernent, par exemple, le débruitage et la compression [OS75]. Pour le traitement de l'image, des décompositions plus géométriques sont nécessaires.

## Décomposition structure-texture pour les images

Le principe de la décomposition structure-texture a été introduit par Meyer dans [Mey01]. Une image  $f$  est décomposée sous la forme  $f = u + v$ . La composante  $u$ , dite de structure, est censée contenir uniquement les formes contrastées des objets présents dans la scène visuelle capturée, tandis que la composante  $v$ , dite de texture, contient les "motifs oscillants", c'est-à-dire les textures et le bruit éventuel.

Remarquons qu'il s'agit d'une décomposition à caractère géométrique qui ne peut être obtenue par simple décomposition linéaire en hautes et basses fréquences car des fréquences élevées peuvent apparaître à la fois dans la partie structure, du fait des discontinuités dues aux frontières des objets, et dans la partie  $v$ , du fait des textures et éventuellement du bruit.

Les premières approches pour la décomposition structure-texture ont été développées dans un cadre fonctionnel continu. Nous considérons donc ici une image  $f$  définie sur un ouvert  $\Omega$  de  $\mathbb{R}^2$  et à valeurs dans  $\mathbb{R}$ . Dans [Mey01], Meyer propose un cadre général permettant de séparer les composantes  $u$  et  $v$ . Le principe est formulé de manière variationnelle et consiste à considérer le problème suivant

$$\operatorname{argmin}_{(u,v) \in \mathcal{X}_1 \times \mathcal{X}_2} \{ \lambda F_1(u) + F_2(v), f = u + v \}. \quad (4.2)$$

Dans (4.2),  $\mathcal{X}_1$  et  $\mathcal{X}_2$  désignent deux espaces de fonctions sur  $\Omega$ ,  $F_1$  et  $F_2$  deux fonctionnelles sur  $\mathcal{X}_1$  et  $\mathcal{X}_2$ , respectivement, telles que  $F_i(u) < +\infty$



si et seulement si  $u \in \mathcal{X}_i$ ,  $i \in \{1, 2\}$ . Un choix judicieux des espaces  $\mathcal{X}_i$  et des fonctionnelles  $F_i$  doit être tel que si  $u$  contient principalement des zones homogènes,  $F_1(u) \ll F_2(u)$ , et inversement si  $v$  est très texturée,  $F_2(v) \ll F_1(v)$ .

Les choix de  $\mathcal{X}_1$  et  $F_1$  ont très vite convergé vers l'espace des fonctions à variations bornées et la semi-norme de la variation totale [CL97]. Plusieurs choix ont été proposés pour le couple  $(\mathcal{X}_2, F_2)$ . Le modèle ROF pour le débruitage correspond au choix  $\mathcal{X}_2 = L_2(\Omega)$  et  $F_2 = \|\cdot\|_2^2$ . Dans [Mey01], Meyer propose l'espace suivant pour modéliser les "motifs oscillants"

$$\mathcal{X}_2 = G = \{\operatorname{div} g : g \in L^\infty \times L^\infty\}. \quad (4.3)$$

L'espace  $G$  ainsi défini est muni de la norme suivante, dite norme  $G$ ,

$$(\forall v \in G) \quad \|v\|_G = \inf\{\|g\|_{L^\infty} : v = \operatorname{div} g\}. \quad (4.4)$$

Du fait de la définition de la norme  $G$ , la minimisation du problème de Meyer se révèle difficile. Plusieurs travaux ont été proposés afin de pallier ce problème, parmi lesquels [AABFC05, AC05, CE05, AGCO06, GLMV07, MS89, She05]. Des états de l'art détaillés des approches variationnelles du type (4.2) pour la décomposition structure-texture, peuvent être trouvés dans [BLMV10, Gil09].

Une approche différente pour la décomposition structure-texture a été proposée dans [SED04, SED05]. Il s'agit de l'approche MCA (Morphological Component Analysis). Le lecteur intéressé pourra consulter [PFS<sup>+</sup>07, BSFM07] ainsi que l'article de synthèse [FSBM10].

L'intérêt des décompositions structure-texture a été illustré au travers de divers travaux récents. Une des premières applications a porté sur la problématique de la reconstruction de zones détériorées ou masquées dans des images (inpainting en anglais) [Mas02, BSCB00]. La problématique de l'inpainting est classée le plus souvent en deux catégories : inpainting de structure et inpainting de texture. Cette distinction est donc liée au type d'information présent dans l'image sous-jacente. Les images naturelles contenant le plus souvent à la fois des structures saillantes et des textures très oscillantes, l'inpainting simultané de ces deux types d'informations se révèle important. Une approche naturelle consiste alors à isoler les composantes structure  $u$  et texture  $v$  et à appliquer des algorithmes différents sur chaque composante. Cette approche a été adoptée dans [BVSO03], où les auteurs appliquent l'algorithme [BSCB00] afin de compléter la partie structure  $u$  et un algorithme de synthèse de texture [EL99] pour compléter la partie  $v$ . Les deux composantes sont alors recombinaées pour obtenir une reconstruction finale. Ce principe est illustrée



FIGURE 4.1 – Illustration de l'utilisation de la décomposition structure-texture pour l'inpainting simultané des structures et textures dans les images.

dans la figure 4.1 tirée de [BVSO03]. Signalons enfin, que d'autres algorithmes d'inpainting ont eu recours à la décomposition structure-texture, notamment dans [ESQD05].

L'application de la décomposition structure-texture à la détection de contours a été considérée dans [BLMV10] où les auteurs ont proposé d'effectuer la détection sur la partie  $u$ .

En segmentation d'images, les auteurs de [BT06] ont proposé de séparer les composantes  $u$  et  $v$  et d'effectuer la segmentation sur la partie structure uniquement. Récemment, les auteurs de [CPGN<sup>+</sup>12] ont combiné une décomposition structure-texture avec l'algorithme de segmentation basé sur les coupes normalisées de Shi et Malik [SM00]. Ce dernier est basé sur la décomposition

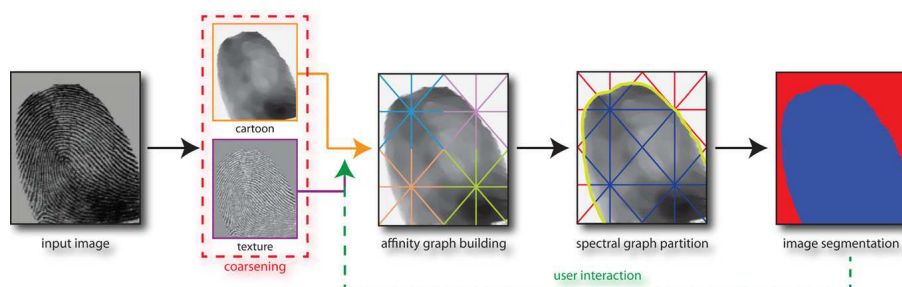


FIGURE 4.2 – Approche pour la segmentation d’images introduite dans [CPGN<sup>+</sup>12]. La première étape consiste en une décomposition structure-texture.

spectrale d’un graphe de similarité construit à partir de l’image à segmenter. Les auteurs de [CPGN<sup>+</sup>12] ont proposé de construire le graphe de similarité à partir de la composante structure de l’image originale. Ils montrent que ce choix permet d’avoir un graphe moins dense, permettant ainsi d’accélérer l’étape de décomposition spectrale nécessaire à l’application de cet algorithme. Les auteurs montrent aussi que, moyennant un choix judicieux de la fonction de poids, une telle approche n’altère pas la qualité de la segmentation finale. Enfin, ils proposent également un procédé de raffinement de la segmentation illustré dans la figure 4.2, issue de [CPGN<sup>+</sup>12].

Les applications de la décomposition structure-texture que nous avons rappelées jusqu’à présent appartiennent au domaine du traitement de l’image. En photographie numérique, il est aussi fréquent d’être amené à décomposer une image en une partie de texture et une autre, constante par morceaux, de structure. Les applications principales concernent le rehaussement ou l’atténuation de détails [BPD06, FAR07, FFLS08, GO11], la fusion d’images [FAR07], l’édition d’images [OCDD01, KRFB06] et le tone mapping [TT99, DD02] pour la visualisation d’images à grande gamme dynamique (High Dynamic Range Imaging).

Pour les applications en rehaussement de détails, il s’agit de faire apparaître, à partir de la manipulation des composantes  $u$  et  $v$ , des détails initialement présents dans la scène visuelle, mais dont le procédé d’acquisition a réduit l’importance. Un exemple est donnée dans la figure 4.3 issue de [FAR07]. À l’inverse du rehaussement, on peut être amené dans le contexte de la retouche d’images, à chercher à atténuer certains détails présents dans la scène visuelle capturée.

En photographie numérique, la composante  $u$  est obtenue généralement en appliquant un filtre non linéaire à  $f$  ou par diffusion anisotrope [PM90,



FIGURE 4.3 – Illustration de la problématique du rehaussement de détails.



FIGURE 4.4 – Édition du maillage à gauche.

[Tsc06]. L’usage du filtre bilatéral [TM98] est maintenant largement répandu et constitue l’approche standard pour obtenir la composante  $u$  ; la composante  $v$  étant obtenue par différence avec  $f$ .

### Décomposition de maillages et de modèles 3D

Dans le domaine de l’infographie (computer graphics) et de la vision par ordinateur, on est également amené à décomposer des modèles 3D représentés sous la forme de maillages polygonaux. La décomposition à laquelle nous faisons référence ici est différente de la décomposition d’un maillage polygonal en sous-maillages sémantiques comme dans [Sha08]. Il ne s’agit pas non plus d’une représentation multi-résolution où l’échantillonnage des points varie selon la résolution choisie [EDD<sup>+</sup>95]. La décomposition que nous considérons ici est une décomposition d’un maillage en deux composantes, l’une contenant une description sommaire de la géométrie de l’objet modélisé, et l’autre contenant les détails géométriques fins. Il s’agit là de l’équivalent de la décomposition structure-texture pour les modèles 3D. La motivation principale est d’améliorer la description d’un objet géométrique en modélisant explicitement sa surface à deux niveaux différents [PKG06]. Les applications appartiennent au domaine du débruitage et à l’édition de modèles 3D [BKP<sup>+</sup>10]. La figure 4.4, tirée de [PKG06], illustre un exemple d’édition d’un modèle 3D.

## 4.3 Décompositions multi-échelles

### 4.3.1 Définition et motivations

Les décompositions que nous avons rappelées dans la section précédente sont mono-échelles : un signal  $\mathbf{f}$  est représenté par deux composantes  $\mathbf{u}$  et  $\mathbf{v}$ . Dans cette section on s'intéresse à des décompositions multi-échelles de  $\mathbf{f}$ , c'est-à-dire des représentations de  $\mathbf{f}$  sous la forme

$$\mathbf{f} = \mathbf{s}_k + \mathbf{d}_k + \cdots + \mathbf{d}_1. \quad (4.5)$$

Dans (4.5), le signal  $\mathbf{s}_k$  représente une approximation *basse échelle*<sup>1</sup> de  $\mathbf{f}$  et les  $\mathbf{d}_i$  les détails nécessaires à la reconstruction de  $\mathbf{f}$  à partir de  $\mathbf{s}_k$ . En particulier, pour  $i \in \{1, \dots, k\}$ ,  $\mathbf{d}_i$  désigne la couche de détail permettant de passer de l'approximation  $\mathbf{s}_i$  à l'échelle  $i$  à l'approximation  $\mathbf{s}_{i-1}$  à l'échelle plus fine  $i - 1$ .

Les motivations amenant à considérer des décompositions de la forme (4.5) pour le traitement de l'image sont multiples. Elles ont pour point commun le constat que l'interprétation des images dépend de l'échelle à laquelle on les observe [Wit84, Lin93]. Par exemple, la séparation structure-texture dépend de manière intrinsèque de l'échelle d'observation : ce qui est considéré comme texture à une échelle donnée peut correspondre à des structures si on l'observe à des échelles plus fines.

La remarque importante qui précède est aussi valable au-delà du traitement du signal et de l'image, en particulier pour le traitement de maillages et de modèles 3D. Dans ce cadre, la notion d'échelle d'observation est simulée au niveau du rendu par la technique du niveau de détail (level-of-detail) [PGK02, BKP<sup>+</sup>10].

Plus généralement, pour le traitement, il est important de pouvoir contrôler l'échelle spatiale des détails que l'on manipule et il est souvent préférable d'opérer sur les détails à différentes échelles, au lieu d'une unique échelle.

Le concept d'analyse multi-résolution (AMR) introduit par Mallat dans [Mal89] permet de formaliser la notion de décomposition multi-échelle que nous venons de rappeler. L'AMR est à la base de la construction des bases d'ondelettes et de la transformée en ondelettes rapide [Mal09, D<sup>+</sup>92]. La généralisation de cette approche aux signaux sur graphes présente des difficultés et fait l'objet de beaucoup de travaux récents. Le chapitre suivant est consacré à cette approche. Dans le présent chapitre, nous nous intéressons à des approches où une représentation du type (4.5) est obtenue en considérant soit une famille de filtres à paramètre croissant, soit l'approche hiérarchique de [TNV04].

---

1. Nous réservons dans ce manuscrit le terme *résolution* à l'échantillonnage des signaux considérés

### 4.3.2 Décompositions basées sur le principe de la pyramide laplacienne

Cette approche est principalement utilisée en photographie numérique [R JW96, PFFG98, CPD07a, FFLS08]. Il s'agit de reprendre le principe de la pyramide laplacienne introduite dans [BA83]. Notons toutefois que dans ces approches, l'étape de sous-échantillonnage initialement proposée dans [BA83] n'est pas retenue car, d'une part, les applications ne concernent pas la compression d'images, et d'autre part, les approximations successives sont obtenues par un filtrage adaptatif et ne correspondent donc pas à l'application d'un filtre passe-bas. En particulier, l'opération de filtrage n'est pas linéaire.

Plus spécifiquement, le principe de ces approches est de construire une famille d'images  $(\mathbf{s}^1, \dots, \mathbf{s}^k)$ . Chaque image  $\mathbf{s}^i$  correspond à une approximation à l'échelle  $i$  de l'image originale  $\mathbf{f}$ . La notion d'échelle est ici capturée par un paramètre exogène, s'agissant généralement d'un paramètre de l'algorithme utilisé. La famille de détails est alors construite par différences successives en prenant  $\mathbf{d}_i = \mathbf{s}^{i-1} - \mathbf{s}^i$ .

Dans [CPD07a], chaque approximation  $\mathbf{s}_i$  est obtenue en appliquant le filtre bilatéral [TM98] à  $\mathbf{f}$ , les largeurs des noyaux spatial et photométrique augmentant à mesure que  $i$  progresse. Dans [FFLS08], les approximations  $\mathbf{s}_i$  sont obtenues comme solutions d'un problème d'optimisation quadratique générique, correspondant à une régularisation laplacienne pondérée, dont le paramètre de régularisation augmente à mesure que  $i$  devient plus grand.

Il est à noter enfin qu'une approche basée sur le même principe a également été proposée dans [HPFS11] où les auteurs utilisent l'estimateur de Nadayara-Watson [Nad64] pour obtenir les approximations  $\mathbf{s}_i$ . Les auteurs proposent également une approche bayésienne permettant d'inférer, en présence de bruit, les niveaux de détails crédibles et ceux correspondant à des artefacts dus au bruit.

Lorsque l'approche que nous venons de décrire est implémentée avec des filtres linéaires, il est équivalent d'appliquer les filtres successifs au signal initial  $\mathbf{f}$  ou récursivement à chaque approximation  $\mathbf{s}_i$ . Ceci n'est plus vrai si les filtres utilisés ne sont pas linéaires. Nous décrivons dans la sous-section suivante des approches où les couples  $(\mathbf{s}_i, \mathbf{d}_i)$  sont obtenus récursivement. Nous détaillons en particulier l'approche de Tadmor, Nezzar et Vese utilisant une pénalisation basée sur la variation totale.



### 4.3.3 L'approche hiérarchique de Tadmor, Nezzar et Vese pour la décomposition d'images

Cette approche de décomposition a été introduite dans [TNV04] dans un cadre continu. Ici, nous nous plaçons dans le cadre discret de la section 3.3 et présentons l'approche de [TNV04] dans ce cadre.

Rappelons la fonctionnelle ROF du chapitre précédent :

$$(\forall \mathbf{f}, \mathbf{u} \in \mathbb{R}^N) (\forall \lambda > 0) \quad E(\mathbf{u}; \mathbf{f}, \lambda) = \lambda J_1(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|^2, \quad (4.6)$$

où

$$J_1(\mathbf{u}) = \sum_{i,j=1}^n \|(\nabla \mathbf{u})_{i,j}\|, \quad (4.7)$$

La minimisation de (4.6) conduit à la décomposition à deux niveaux  $\mathbf{f} = \mathbf{u}_\lambda + \mathbf{v}_\lambda$  où  $\mathbf{u}_\lambda$  est le minimiseur et  $\mathbf{v}_\lambda$  le résidu. Cette décomposition est ici interprétée comme une décomposition structure-texture comme dans [Mey01]. Les auteurs de [TNV04] proposent d'étendre cette décomposition mono-échelle en une décomposition multi-échelle hiérarchique en décomposant les résidus successifs à des échelles de plus en plus fines.

#### Principe

Illustrons le principe avec deux paramètres d'échelles  $0 < \lambda_1 < \lambda_0$ . Une première décomposition  $\mathbf{f} = \mathbf{u}_0 + \mathbf{v}_0$  est obtenue en décomposant  $\mathbf{f}$  à l'échelle  $\lambda_0$  :

$$\mathbf{u}_0 = \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^N} E(\mathbf{u}; \mathbf{f}, \lambda_0), \quad (4.8)$$

et

$$\mathbf{v}_0 = \mathbf{f} - \mathbf{u}_0 \quad (4.9)$$

Dans [TNV04], la composante  $\mathbf{u}_0$  correspond à une description sommaire de l'image  $\mathbf{f}$ , contenant principalement les frontières des objets présents dans  $\mathbf{f}$ .

Ensuite, la décomposition est raffinée en décomposant le résidu  $\mathbf{v}_0$  à l'échelle  $\lambda_1$ . On obtient alors  $\mathbf{v}_0 = \mathbf{u}_1 + \mathbf{v}_1$ , où

$$\mathbf{u}_1 = \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^N} E_2(\mathbf{u}; \mathbf{v}_0, \lambda_1), \quad (4.10)$$

et

$$\mathbf{v}_1 = \mathbf{v}_0 - \mathbf{u}_1. \quad (4.11)$$

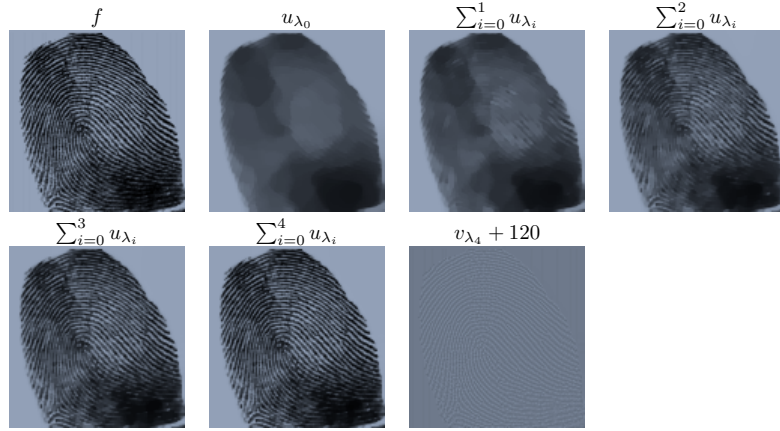


FIGURE 4.5 – Décomposition de l'image d'une empreinte en 5 étapes.

La composante  $\mathbf{u}_1$  contient maintenant la texture correspondant à l'échelle  $\lambda_1$ , du point de vue de la fonctionnelle utilisée. Nous disposons à ce stade de la décomposition à trois niveaux  $\mathbf{f} = \mathbf{u}_0 + \mathbf{u}_1 + \mathbf{v}_1$ .

Pour une famille de paramètres  $\lambda_0 > \lambda_1 > \dots > \lambda_n > 0$ , on obtient la décomposition

$$\mathbf{f} = \mathbf{u}_0 + \dots + \mathbf{u}_n + \mathbf{v}_n. \quad (4.12)$$

La suite  $(\mathbf{u}_i)_{i \geq 0}$  représente l'image  $\mathbf{f}$  aux différentes échelles  $\lambda_i$  tandis que la somme  $\mathbf{u}_0 + \dots + \mathbf{u}_n$  fournit une représentation multi-couche de  $\mathbf{f}$ . À mesure que les  $\lambda_i$  décroissent, les images  $\mathbf{u}_i$  intègrent de plus en plus de texture et la somme partielle  $\sum_{i=0}^n \mathbf{u}_i$  s'approche de  $\mathbf{f}$ . La figure 4.5, tirée de [TNV04], illustre le principe.

Dans [TNV04], la convergence de  $\sum_{i=0}^n \mathbf{u}_i$  vers  $\mathbf{f}$  est établie, dans un cadre fonctionnel, pour la suite dyadique  $\lambda_{i+1} = \lambda_i/2$ . La convergence de la série  $\sum_i \mathbf{u}_i$  permet d'interpréter la suite des sommes partielles comme un espace d'échelles inverse à temps discret [SG06, OBG<sup>+</sup>05]. La représentation commence avec une version simplifiée  $\mathbf{u}_0$  et progresse vers l'image originale à mesure qu'on effectue des décompositions successives.

Un espace d'échelles continu basé sur le principe de la représentation multi-échelle hiérarchique a été proposé dans [TA09]. Des applications en débruitage et en reconstruction y sont considérées. Le même principe est repris dans [AT11] en utilisant une pénalisation basée sur la variation totale et un terme d'attache aux données avec une norme  $\ell_1$ .

Nous reprenons dans la suite de ce chapitre le principe de la décomposition hiérarchique de [TNV04], et l'appliquons à des signaux sur graphes. Dans ce



qui va suivre nous considérerons les décompositions obtenues avec les a priori  $J_2^w$  et  $J_{1,q}^w$  du chapitre précédent et étudions leurs propriétés.

## 4.4 Décomposition multi-échelle hiérarchique de signaux sur graphes basée sur un a priori quadratique

Dans toute la suite de cette section,  $G$  désigne un graphe pondéré d'ordre  $N$  et  $\mathbf{f} \in \mathbb{R}^N$ .

### 4.4.1 Principe

Nous développons une décomposition multi-échelle de  $\mathbf{f}$  basée sur la décomposition hiérarchique de [TNV04]. Comme dans [TNV04], la notion d'échelle est ici extrinsèque au signal  $\mathbf{f}$  : elle est dictée par l'a priori choisi ainsi que par la suite  $(\lambda_i)_{i \geq 0}$ . L'a priori utilisé dans cette section est  $J_2^w$ , défini dans le chapitre précédent et basé sur la matrice laplacienne  $\mathbf{L}(G)$ . Afin d'alléger les notations, nous noterons  $\mathbf{L}$  au lieu de  $\mathbf{L}(G)$ .

Rappelons la fonctionnelle quadratique (non normalisée) du chapitre précédent :

$$(\forall \lambda > 0) (\forall \mathbf{u} \in \mathcal{H}(V)) \quad E_2^w(\mathbf{u}; \mathbf{f}, \lambda) = \lambda \langle \mathbf{L}\mathbf{u}, \mathbf{u} \rangle + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|_2^2. \quad (4.13)$$

Soit  $\lambda_0 > \lambda_1 > \dots > \lambda_n > 0$  une famille décroissante de paramètres d'échelle. La décomposition multi-échelle hiérarchique basée sur la fonctionnelle (4.13) est donnée par l'algorithme suivant :

$$\begin{cases} \mathbf{v}_{-1} &= \mathbf{f}, \\ \mathbf{u}_i &= \operatorname{argmin}_{\mathbf{u} \in \mathcal{H}(V)} E_2^w(\mathbf{u}; \mathbf{v}_{i-1}; \lambda_i), \quad 0 \leq i \leq n, \\ \mathbf{v}_i &= \mathbf{v}_{i-1} - \mathbf{u}_i, \quad 0 \leq i \leq n. \end{cases} \quad (4.14)$$

L'algorithme (4.14) donne lieu à la décomposition

$$\mathbf{f} = \sum_{i=0}^n \mathbf{u}_i + \mathbf{v}_n. \quad (4.15)$$

Intuitivement, à mesure que les  $\lambda_i$  décroissent, les résidus  $\mathbf{v}_i$  contiennent de moins en moins d'information. Nous précisons cette intuition dans la section suivante.

### 4.4.2 Convergence et choix des paramètres

Nous commençons par établir la convergence de la décomposition.

#### Convergence

Fixons pour la suite de cette section une base orthonormée  $(\mathbf{e}_0, \dots, \mathbf{e}_{N-1})$  de  $\mathcal{H}(V)$  formée par des vecteurs propres de  $\mathbf{L}$ . Chaque vecteur propre  $\mathbf{e}_i$  est associé à une valeur propre  $\alpha_i$  avec  $0 = \alpha_0 \leq \dots \leq \alpha_{N-1}$ . Nous exprimons tout d'abord dans cette base le résultat d'une seule étape de décomposition.

**Proposition 4.1.** Soit  $\lambda > 0$ . Notons  $\mathbf{u}_\lambda = \operatorname{argmin}_{\mathbf{u} \in \mathcal{H}(V)} E_2^w(\mathbf{u}; \mathbf{f}, \lambda)$  et  $\mathbf{v}_\lambda = \mathbf{f} - \mathbf{u}_\lambda$ .

Alors

$$\mathbf{u}_\lambda = \sum_{j=0}^{N-1} \frac{1}{1 + 2\lambda\alpha_j} \langle \mathbf{f}, \mathbf{e}_j \rangle \mathbf{e}_j, \quad (4.16)$$

et

$$\mathbf{v}_\lambda = \sum_{j=0}^{N-1} \left( 1 - \frac{1}{1 + 2\lambda\alpha_j} \right) \langle \mathbf{f}, \mathbf{e}_j \rangle \mathbf{e}_j. \quad (4.17)$$

*Démonstration.* Le résultat découle de la caractérisation (3.57) du chapitre précédent et du fait que la matrice  $2\lambda\mathbf{L} + \mathbf{I}$  est symétrique. En effet,

$$\begin{aligned} (\forall j \in \{0, \dots, N-1\}) \quad \langle \mathbf{f}, \mathbf{e}_j \rangle &= \langle (2\lambda\mathbf{L} + \mathbf{I})\mathbf{u}_\lambda, \mathbf{e}_j \rangle \\ &= \langle \mathbf{u}_\lambda, (2\lambda\mathbf{L} + \mathbf{I})\mathbf{e}_j \rangle \\ &= (1 + 2\lambda\alpha_j) \langle \mathbf{u}_\lambda, \mathbf{e}_j \rangle. \end{aligned}$$

□

**Remarque 4.1.** 1. La proposition précédente établit donc un lien entre la régularisation quadratique laplacienne et le filtrage spectral dans une base formée par des vecteurs propres de  $\mathbf{L}$ . Dans cette base, chaque composante  $\langle \mathbf{f}, \mathbf{e}_j \rangle$  est “modulée” indépendamment de autres et en fonction de la valeur propre  $\alpha_j$  et du paramètre d'échelle  $\lambda$ .

2. Dans le cadre euclidien, la remarque précédente peut être exploitée pour calculer la solution  $\mathbf{u}_\lambda$ . Ceci est possible car, d'une part, les vecteurs propres de  $\mathbf{L}$  ont une forme analytique (il s'agit des exponentiels complexes) et, d'autre part, l'algorithme de la transformée de Fourier rapide permet un calcul aisé des composantes. Pour un graphe quelconque, les vecteurs propres n'ont pas de forme analytique particulière.

3. Rappelons que si  $A_0, \dots, A_{k-1}$  désignent les composantes connexes de  $G$ , alors  $\dim(\ker(\mathbf{L})) = k$  et  $\ker(\mathbf{L}) = \text{vect}(\mathbf{1}_{A_0}, \dots, \mathbf{1}_{A_{k-1}})$ . Ainsi,

$$\mathbf{u}_\lambda = \text{proj}_{\ker(\mathbf{L})} \mathbf{f} + \sum_{j=k}^{N-1} \left( \frac{1}{1 + 2\lambda\alpha_j} \langle \mathbf{f}, \mathbf{e}_j \rangle \mathbf{e}_j \right) \quad (4.18)$$

et

$$\text{proj}_{\ker(\mathbf{L})} \mathbf{f} = \sum_{j=1}^k \overline{\mathbf{f}}_{|A_j} \mathbf{1}_{A_j}, \quad (4.19)$$

où  $\overline{\mathbf{f}}_{|A_j}$  désigne la moyenne de la restriction de  $\mathbf{f}$  à la composante connexe  $A_j$ .

Nous pouvons à présent prouver la convergence de la décomposition obtenue à l'aide de l'algorithme (4.14).

**Proposition 4.2.** Soit  $(\lambda_n)_{n \geq 0}$  une suite décroissante dans  $\mathbb{R}^+$  telle que  $\lim_{n \rightarrow +\infty} \lambda_n = 0$ . Soit  $(\mathbf{u}_n)_{n \geq 0}$  et  $(\mathbf{v}_n)_{n \geq 0}$  les suites obtenues en appliquant l'algorithme (4.14) avec les  $\lambda_i$  comme paramètres d'échelle. Alors

$$\mathbf{f} = \sum_{i=0}^{+\infty} \mathbf{u}_i. \quad (4.20)$$

*Démonstration.* En utilisant la proposition 4.1 nous avons, par récurrence immédiate,

$$(\forall n \geq 0) (\forall j \in \{0, \dots, N-1\}) \quad \langle \mathbf{v}_n, \mathbf{e}_j \rangle = \left( \prod_{i=0}^n \left( 1 - \frac{1}{1 + 2\lambda_i \alpha_j} \right) \right) \langle \mathbf{f}, \mathbf{e}_j \rangle. \quad (4.21)$$

Notons  $P_n = \prod_{i=0}^n \left( 1 - \frac{1}{1 + 2\lambda_i \alpha_j} \right)$ . Les valeurs propres  $\lambda_i$  étant toutes positives,  $(P_n)_{n \geq 0}$  est décroissante et positive. De plus, le terme général ne tend pas vers 1, donc  $\lim_{n \rightarrow +\infty} P_n = 0$ .  $\square$

### Choix de la suite d'échelles $\lambda_i$

Nous optons ici pour une suite de paramètres de la forme  $\lambda_i = \lambda_{i-1}/2$ . Le premier paramètre  $\lambda_0$  doit être choisi suffisamment grand afin que  $\mathbf{u}_0$  corresponde à une version simplifiée du signal  $\mathbf{f}$ . Les décompositions suivantes viendront alors compléter  $\mathbf{f}$  en y rajoutant des couches successives de détail. Pour  $\lambda_0$  très grand, nous voyons, d'après le point 3 de la remarque 4.1, que  $\mathbf{u}_0$  correspondra au signal constant sur chaque composante connexe  $A_i$ . Cette

limite étant asymptotique, on pourrait envisager d'utiliser le critère suivant pour choisir  $\lambda_0$  :

$$\|\text{proj}_{\ker L}(\mathbf{f})\|^2 = (1 - \epsilon)\|\mathbf{u}_\lambda\|^2, \quad (4.22)$$

où  $\epsilon$  est petit (de l'ordre de  $10^{-1}$  ou  $10^{-2}$ ). Ici encore, on aurait besoin de calculer tout le spectre de la matrice laplacienne afin de pouvoir appliquer le critère (4.22). Nous verrons dans la section suivante que pour la décomposition obtenue avec un a priori de variation totale, le premier paramètre peut être caractérisé. Nous proposerons alors de le calculer.

## 4.5 Décomposition multi-échelle hiérarchique de signaux sur graphes basée sur la variation totale

### 4.5.1 Principe

Dans toute la suite de cette section,  $G$  désigne un graphe pondéré d'ordre  $N$  et  $\mathbf{f} \in \mathbb{R}^N$  un signal sur  $G$ . Comme dans le chapitre précédent, nous noterons la matrice d'incidence  $\mathbf{A}(G)$  simplement  $\mathbf{A}$ , et la matrice d'adjacence pondérée  $\mathbf{W}$ .

Nous développons une décomposition hiérarchique basée sur l'a priori  $J_{1,2}^w$  du chapitre précédent. Rappelons l'expression de  $J_{1,2}^w$  et de la fonctionnelle de régularisation associée  $E_{1,2}^w$  :

$$J_{1,2}^w(\mathbf{u}) = \sum_{i=1}^N \left( \sum_{j=1}^N w_{i,j} (u_i - u_j)^2 \right)^{1/2}$$

et

$$(\forall \lambda > 0) \quad E_{1,2}^w(\mathbf{u}; \mathbf{v}, \lambda) = \lambda J_{1,2}^w(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{v}\|_2^2. \quad (4.23)$$

Soit  $\lambda_0 > \lambda_1 > \dots > \lambda_n > 0$  une famille décroissante de paramètres d'échelle. La décomposition multi-échelle hiérarchique basée sur la fonctionnelle (4.23) est donnée par l'algorithme suivant :

$$\begin{cases} \mathbf{v}_{-1} &= \mathbf{f}, \\ \mathbf{u}_i &= \underset{\mathbf{u} \in \mathcal{H}(V)}{\text{argmin}} E_{1,q}^w(\mathbf{u}; \mathbf{v}_{i-1}; \lambda_i), \quad 0 \leq i \leq n, \\ \mathbf{v}_i &= \mathbf{v}_{i-1} - \mathbf{u}_i, \quad 0 \leq i \leq n. \end{cases} \quad (4.24)$$

L'algorithme (4.14) donne lieu à la décomposition

$$\mathbf{f} = \sum_{i=0}^n \mathbf{u}_i + \mathbf{v}_n. \quad (4.25)$$

Nous étudions les propriétés de cette décomposition dans les sections qui suivent.

### 4.5.2 Convergence

Nous commençons par établir la convergence de (4.25). Le résultat suivant est une conséquence immédiate de la caractérisation (3.80) du chapitre précédent.

**Proposition 4.3.** Soit  $(\lambda_n)_{n \geq 0}$  une suite décroissante dans  $\mathbb{R}^+$  telle que  $\lim_{n \rightarrow +\infty} \lambda_n = 0$ . Soit  $(\mathbf{u}_n)_{n \geq 0}$  et  $(\mathbf{v}_n)_{n \geq 0}$  les suites obtenues en appliquant l'algorithme (4.24) avec  $(\lambda_n)_{n \geq 0}$  comme suite de paramètres d'échelle. Alors

$$\mathbf{f} = \sum_{i=0}^{+\infty} \mathbf{u}_i. \quad (4.26)$$

*Démonstration.* Nous établissons le résultat suivant :  $\lim_{n \rightarrow +\infty} \|\mathbf{f} - \sum_{i=0}^n \mathbf{u}_i\|_{G_2^w} = 0$ . Le résultat final découle du fait que  $\|\cdot\|_{G_2^w}$  est une norme sur  $\mathbb{R}^N$  et de l'équivalence des normes en dimension finie.

D'après la caractérisation (3.80) du chapitre précédent, nous avons

$$\mathbf{f} - \sum_{i=0}^n \mathbf{u}_i = \mathbf{v}_n = \text{proj}_{G_2^w(\lambda_n)}. \quad (4.27)$$

Ainsi,  $\|\mathbf{f} - \sum_{i=0}^n \mathbf{u}_i\|_{G_2^w} \leq \lambda_n$ . D'où finalement

$$\lim_{n \rightarrow +\infty} \|\mathbf{f} - \sum_{i=0}^n \mathbf{u}_i\|_{G_2^w} = 0.$$

□

**Remarque 4.2.** La preuve précédente montre que, mesurée avec la norme  $\|\cdot\|_{G_2^w}$ , la vitesse de convergence de la décomposition hiérarchique vers  $\mathbf{f}$  est dictée par la vitesse à laquelle la suite d'échelles  $(\lambda_n)_{n \geq 0}$  décroît vers zéro.

La proposition suivante relie l'énergie de  $\mathbf{f}$  à celles des  $\mathbf{u}_i$ .

**Proposition 4.4.** Soit  $(\lambda_n)_{n \geq 0}$  une suite décroissante dans  $\mathbb{R}^+$  telle que  $\lim_{n \rightarrow +\infty} \lambda_n = 0$ . Soit  $(\mathbf{u}_n)_{n \geq 0}$  et  $(\mathbf{v}_n)_{n \geq 0}$  les suites obtenues en appliquant l'algorithme (4.24) avec les  $\lambda_i$  comme paramètres d'échelle. Alors

$$\sum_{i=0}^{\infty} \left( \|\mathbf{u}_i\|^2 + 2\lambda_i J_{1,2}^w(\mathbf{u}_i) \right) = \|\mathbf{f}\|^2. \quad (4.28)$$

La preuve de la proposition précédente repose sur le lemme suivant.

**Lemme 4.1.** Soit  $\lambda > 0$ ,  $\mathbf{u}_\lambda = \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^N} E_{1,2}^w(\mathbf{u}; \mathbf{f}, \lambda)$  et  $\mathbf{v}_\lambda = \mathbf{f} - \mathbf{u}_\lambda$ . Alors

$$\langle \mathbf{u}_\lambda, \mathbf{v}_\lambda \rangle = \lambda J_{1,2}^w(\mathbf{u}_\lambda). \quad (4.29)$$

*Démonstration.* D'après l'équation (3.80) du chapitre précédent, nous avons  $\mathbf{v}_\lambda = \operatorname{proj}_{G_2^w(\lambda)} f$ . En appliquant la caractérisation donnée dans la proposition 3.1 du chapitre 3, on voit que  $\mathbf{v}_\lambda$  est l'unique vecteur de  $G_2^w(\lambda)$  vérifiant

$$(\forall \mathbf{w} \in G_2^w(\lambda) \quad \langle \mathbf{w} - \mathbf{v}_\lambda, \mathbf{f} - \mathbf{v}_\lambda \rangle \leq 0. \quad (4.30)$$

Ainsi,

$$(\forall \mathbf{w} \in G_2^w(\lambda) \quad \langle \mathbf{w} - \mathbf{v}_\lambda, \mathbf{u}_\lambda \rangle \leq 0. \quad (4.31)$$

D'où

$$\begin{aligned} \langle \mathbf{u}_\lambda, \mathbf{v}_\lambda \rangle &= \sup_{\mathbf{w} \in G_2^w(\lambda)} \langle \mathbf{u}_\lambda, \mathbf{w} \rangle = \lambda \sup_{\mathbf{w} \in G_2^w(1)} \langle \mathbf{u}_\lambda, \mathbf{w} \rangle, \\ &= \lambda \sigma_{G_2^w(1)}(\mathbf{u}_\lambda), \\ &= \lambda J_{1,2}^w(\mathbf{u}_\lambda). \end{aligned} \quad (4.32)$$

□

Nous démontrons à présent le résultat de la proposition 4.4.

*Démonstration.* Pour  $i \geq -1$ , la décomposition  $\mathbf{v}_i = \mathbf{u}_{i+1} + \mathbf{v}_{i+1}$  implique que

$$\|\mathbf{v}_i\|^2 = \|\mathbf{v}_{i+1}\|^2 + \|\mathbf{u}_{i+1}\|^2 + 2\langle \mathbf{u}_{i+1}, \mathbf{v}_{i+1} \rangle. \quad (4.33)$$

En utilisant le résultat du lemme précédent, nous avons

$$\langle \mathbf{u}_{i+1}, \mathbf{v}_{i+1} \rangle = \lambda_{i+1} J_{1,2}^w(\mathbf{u}_{i+1}).$$

D'où

$$\|\mathbf{v}_i\|^2 - \|\mathbf{v}_{i+1}\|^2 = \|\mathbf{u}_{i+1}\|^2 + 2\lambda_{i+1} J_{1,q}^w(\mathbf{u}_{i+1}). \quad (4.34)$$

En sommant dans l'égalité précédente pour  $i$  allant de  $-1$  (avec  $\mathbf{v}_{-1} = \mathbf{f}$ ) à  $n-1$  on obtient

$$\|\mathbf{f}\|^2 - \|\mathbf{v}_n\|^2 = \sum_{i=0}^n \left( 2\lambda_i J_{1,q}^w(\mathbf{u}_i) + \|\mathbf{u}_i\|^2 \right). \quad (4.35)$$

□

### 4.5.3 Choix de la suite d'échelle $\lambda_i$

Dans cette sous-section nous considérons une progression dyadique de la suite d'échelles  $\lambda_i : \lambda_{i+1} = \lambda_i/2$ , avec  $\lambda_0$  donné. Le choix du premier paramètre  $\lambda_0$  est important puisqu'il détermine la performance de la décomposition. En particulier, si  $\lambda_0$  est choisi trop grand, beaucoup de niveaux seront nécessaires à la reconstruction du signal original. Pour un graphe d'ordre élevé, ou contenant beaucoup d'arêtes, le coût associé à une telle décomposition peut être très important. Si, à l'inverse,  $\lambda_0$  est trop petit, peu de niveaux de détails seront capturés. La proposition suivante montre qu'il existe une valeur critique  $\lambda$  au-delà de laquelle toute décomposition de  $\mathbf{f}$  est de la forme  $\mathbf{f} = \bar{\mathbf{f}} + \mathbf{v}$  où  $\bar{\mathbf{f}}$  désigne le signal dont les composantes sont toutes égales à la moyenne de  $\mathbf{f} : \bar{f}_i = \frac{1}{N} \sum_{j=1}^N f_j$ , et  $\mathbf{v} = \mathbf{f} - \bar{\mathbf{f}}$ .

**Proposition 4.5.** Soit  $f \in \mathbb{R}^N$  and  $\lambda > 0$ . Soit  $u_\lambda$  la solution du problème (4.23). Alors,

$$\begin{aligned} \|f - \bar{f}\|_{\mathcal{G}_w} \geq \lambda &\implies \|f - u_\lambda\|_{\mathcal{G}_w} = \lambda, \\ \|f - \bar{f}\|_{\mathcal{G}_w} \leq \lambda &\implies u_\lambda = \bar{f}, \end{aligned} \tag{4.36}$$

où la norme  $\|\cdot\|_{\mathcal{G}_w}$  est celle introduite dans (3.78).

La proposition précédente est l'analogue pour les signaux sur graphes du théorème 3 de [AABFC05]. Le résultat découle ici immédiatement de la caractérisation (3.80). Cette proposition fournit une interprétation de la norme  $\|\cdot\|_{\mathcal{G}_w}$  : l'intervalle ouvert  $]0, \|\mathbf{f} - \bar{\mathbf{f}}\|_{\mathcal{G}_w}[$  représente l'intervalle d'échelles qui induisent une décomposition non triviale, c'est-à-dire une décomposition où le minimiseur n'est pas la moyenne du signal considéré.

En adoptant une progression dyadique,  $\lambda_{i+1} = \lambda_i/2$ , de la suite d'échelles, et en considérant le résultat de la proposition précédente, il est facile d'encadrer la valeur de  $\lambda_0$  sous la forme :

$$\frac{1}{2} \|\mathbf{f} - \bar{\mathbf{f}}\|_{\mathcal{G}_w} \leq \lambda_0 < \|\mathbf{f} - \bar{\mathbf{f}}\|_{\mathcal{G}_w}. \tag{4.37}$$

Dans la suite, on prendra

$$\lambda_0 = \frac{3\|\mathbf{f} - \bar{\mathbf{f}}\|_{\mathcal{G}_w}}{4}, \tag{4.38}$$

ce qui correspond au point milieu.

Afin d'automatiser le choix du premier paramètre, nous sommes donc amenés à calculer la norme  $G^w$  du signal centré  $\mathbf{f} - \bar{\mathbf{f}}$ . Les détails sont donnés dans ce qui suit.

### Calcul de la norme $G^w$

Dans [AC05], les auteurs proposent de calculer la norme  $G$  d'une image par dichotomie, en exploitant précisément le résultat de la proposition 4.5. Cependant, les auteurs remarquent que cette méthode ne permet pas d'avoir une approximation précise de la norme  $G$ . Une approche similaire a également été proposée dans [SAC06]. Nous proposons dans cette section d'utiliser l'algorithme primal-dual de Chambolle et Pock [CP11a] afin de calculer la norme  $\mathcal{G}_w$  d'un signal centré.

Dans la suite de cette sous-section, les espace  $\mathcal{X}$  et  $\mathcal{Y}$  sont ceux de la section 3.4 du chapitre précédent. Soit  $\mathbf{u} \in G^w$ . Le calcul de  $\|\mathbf{u}\|_{G^w}$  revient à résoudre le problème convexe suivant :

$$\begin{aligned} & \underset{\mathbf{G} \in \mathcal{Y}}{\text{Minimiser}} \|\mathbf{G}\|_{\infty,2}, \\ & \text{sous contrainte } \text{div}^w(\mathbf{G}) = \mathbf{u}. \end{aligned} \quad (4.39)$$

Le problème (4.39) peut être réécrit sous la forme

$$\underset{\mathbf{G} \in \mathcal{Y}}{\text{Minimiser}} \|\mathbf{G}\|_{\infty,2} + \iota_{\{\mathbf{u}\}} \circ \text{div}^w(\mathbf{G}). \quad (4.40)$$

L'application de l'algorithme de Chambolle et Pock avec les identifications suivantes

$$F \equiv \iota_{\{\mathbf{u}\}}, \quad K \equiv \text{div}^w, \quad G = \|\cdot\|_{\infty,2},$$

et avec  $\theta = 1$  amène à considérer les itérations suivantes :

$$\begin{cases} \mathbf{G}^{n+1} = \text{prox}_{\tau\|\cdot\|_{\infty,2}}(\mathbf{G}^n + \tau\nabla_w \bar{\mathbf{v}}^n), \\ \mathbf{v}^{n+1} = \text{prox}_{\sigma\langle \cdot, \mathbf{u} \rangle_{\mathcal{X}}}(\mathbf{v}^n + \sigma \text{div}^w \mathbf{G}^{n+1}), \\ \bar{\mathbf{v}}^{n+1} = 2\mathbf{v}^{n+1} - \mathbf{v}^n. \end{cases} \quad (4.41)$$

Notons à ce stade que le problème dual associé à (4.39) est donné par

$$\begin{aligned} & \underset{\mathbf{v} \in \mathcal{X}}{\text{Minimiser}} \langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{X}}, \\ & \text{sous contrainte } J_w(\mathbf{v}) \leq 1, \end{aligned} \quad (4.42)$$

et présente donc une contrainte sur la variation totale de la variable de décision  $\mathbf{v}$ .

### Calcul des opérateurs proximaux

Le calcul du second opérateur proximal dans (4.41) est immédiat puisqu'il s'agit de l'opérateur proximal d'une fonction linéaire. L'expression est la suivante :

$$\forall \mathbf{v} \in \mathcal{X}, \text{prox}_{\sigma\langle \cdot, \mathbf{u} \rangle_{\mathcal{X}}}(\mathbf{v}) = \mathbf{v} - \sigma\mathbf{u}. \quad (4.43)$$



Afin de calculer le premier opérateur proximal dans (4.41), nous appliquons l'identité de Moreau, en utilisant le fait que la conjuguée d'une norme est la fonction support de la boule unité pour la norme duale et que les normes  $\|\cdot\|_{\infty,2}$  et  $\|\cdot\|_{1,2}$  sont duales. Ainsi, nous obtenons

$$\begin{aligned} \text{prox}_{\tau\|\cdot\|_{\infty,2}}(\mathbf{G}) &= \mathbf{G} - \tau \text{prox}_{\|\cdot\|_{\infty,2}^*/\tau}\left(\frac{1}{\tau}\mathbf{G}\right) \\ &= \mathbf{G} - \tau \text{prox}_{\ell_{B_{1,2}(1)}}\left(\frac{1}{\tau}\mathbf{G}\right) \\ &= \mathbf{G} - \text{proj}_{B_{1,2}(\tau)}(\mathbf{G}), \end{aligned} \quad (4.44)$$

où  $B_{1,2}(\tau)$  désigne la boule centrée en zéro et de rayon  $\tau$  pour la norme  $\|\cdot\|_{1,2}$ .

### Calcul de la projection sur une boule pour la $\|\cdot\|_{1,2}$

Nous venons de voir que, afin de calculer le premier opérateur proximal dans (4.41), nous avons besoin de calculer la projection d'une matrice  $\mathbf{G} \in \mathcal{Y}$  sur une boule pour la norme  $\|\cdot\|_{1,2}$ . Nous proposons ici d'étendre l'algorithme proposé dans [CR05, DFL08], qui calcule la projection  $\ell_2$  d'un vecteur dans  $\mathbb{R}^n$  sur une boule  $\ell_1$ . À cette fin, nous avons besoin d'introduire un opérateur de seuillage doux [Don95] agissant sur des matrices  $\mathbf{G} \in \mathcal{Y}$ . Une extension similaire à des champs de vecteurs a récemment été proposée dans [FP11] et utilisée dans une étape intermédiaire permettant de calculer la projection d'un vecteur sur une boule TV de rayon donné.

L'opérateur de seuillage doux au niveau  $\mu$ ,  $\mu > 0$ , que nous proposons est défini comme suit :

$$(S_\mu \mathbf{G})_{i,j} = \begin{cases} 0 & \text{if } \|\mathbf{G}_{i,\cdot}^\top\|_X \leq \mu, \\ \left(1 - \frac{\mu}{\|\mathbf{G}_{i,\cdot}^\top\|_X}\right) \mathbf{G}_{i,j} & \text{if } \|\mathbf{G}_{i,\cdot}^\top\|_X > \mu. \end{cases} \quad (4.45)$$

Ainsi,

$$\begin{aligned} \|S_\mu \mathbf{G}\|_{1,2} &= \sum_{i=1}^N \|(S_\mu \mathbf{G})_{i,\cdot}^\top\|_X = \sum_{i=1}^N \left( \sum_{j=1}^N (S_\mu \mathbf{G})_{i,j}^2 \right)^{1/2} \\ &= \sum_{\substack{i=1 \\ \|\mathbf{G}_{i,\cdot}^\top\|_X \geq \mu}}^N \left(1 - \frac{\mu}{\|\mathbf{G}_{i,\cdot}^\top\|_X}\right) \|\mathbf{G}_{i,\cdot}^\top\|_X \\ &= \sum_{\substack{i=1 \\ \|\mathbf{G}_{i,\cdot}^\top\|_X \geq \mu}}^N \left(\|\mathbf{G}_{i,\cdot}^\top\|_X - \mu\right). \end{aligned} \quad (4.46)$$

La proposition qui suit fait le lien entre l'opérateur de seuillage doux que nous venons de définir et le calcul de la projection sur une boule pour la norme  $\|\cdot\|_{1,2}$ .

**Proposition 4.6.** Soit  $\mathbf{G} \in \mathcal{Y}$  et  $r > 0$ . Si  $\|\mathbf{G}\|_{1,2} \leq r$ , alors  $\text{proj}_{B_{1,2}(R)}(\mathbf{G}) = \mathbf{G}$ . Sinon, il existe un unique  $\mu > 0$  tel que  $\text{proj}_{B_{1,2}(R)}(\mathbf{G}) = S_\mu \mathbf{G}$ .

*Démonstration.* La première assertion découle de la définition de l'opérateur de projection. Supposons à présent que  $\|\mathbf{G}\|_{1,2} > r$ . Notons tout d'abord que, au vu de (4.46), l'application  $\mu \mapsto \|S_\mu y\|_{1,2}$  définie sur  $[0, \infty[$  et à valeurs dans  $\mathbb{R}$  est affine par morceaux, continue et décroissante. De plus,  $\|S_0 y\|_{1,2} = \|\mathbf{G}\|_{1,2}$  et  $\|S_\mu \mathbf{G}\|_{1,2} = 0$  pour tout  $\mu \geq \max_{i=1,\dots,N} \|y_{i,\cdot}^\top\|_X$ . Il existe donc un unique  $\mu > 0$  tel que  $\|S_\mu \mathbf{G}\|_{1,2} = r$ . Afin de montrer que  $S_\mu \mathbf{G} = \text{proj}_{B_{1,2}(R)}(\mathbf{G})$ , il suffit d'observer que  $S_\mu \mathbf{G}$  est l'unique solution (c.f., par exemple, [CDVLL98] pour le cas vectoriel) du problème

$$\underset{\mathbf{H} \in \mathcal{Y}}{\text{Minimiser}} \|\mathbf{G}\|_{\mathcal{Y}}^2 + \mu \|\mathbf{H}\|_{1,2}. \quad (4.47)$$

D'où

$$(\forall \mathbf{H} \in B_{1,2}(r)) \quad \|S_\mu \mathbf{G} - \mathbf{G}\|_{\mathcal{Y}}^2 \leq \|\mathbf{H} - \mathbf{G}\|_{\mathcal{Y}}^2 + \mu(\|\mathbf{H}\|_{1,2} - r), \quad (4.48)$$

de telle sorte que

$$(\forall \mathbf{H} \in B_{1,2}(r)) \quad \|S_\mu \mathbf{G} - \mathbf{G}\|_{\mathcal{Y}}^2 \leq \|\mathbf{H} - \mathbf{G}\|_{\mathcal{Y}}^2. \quad (4.49)$$

Ce qui conclut la preuve.  $\square$

En examinant l'équation (4.46), nous voyons que la pente de l'application linéaire par morceaux  $\mu \mapsto \|S_\mu y\|_{1,2}$  change au points  $\|\mathbf{G}_{i,\cdot}^\top\|_X$ ,  $i = 1, \dots, N$ . L'algorithme 4.1 permet de calculer la projection.

## 4.6 Exemples de décompositions obtenues avec $J_{1,2}^w$

Nous présentons à présent quelques décompositions obtenues en appliquant notre proposition de décomposition pour uniquement l'énergie  $J_{1,2}^w$ . Nous considérons trois types de données : des images, des maillages 3D et des nuages de points. Afin d'obtenir ces décompositions, des graphes de similarités appropriés sont construits pour chaque type de données. Soit  $\mathbf{F}$  les données à décomposer. Nous considérons que  $\mathbf{F}$  est dans l'espace  $\mathbb{R}^{n \times m}$  avec  $n$  le

- 1: INPUT :  $\mathbf{G} \in \mathcal{Y}$ ,  $0 < r \leq \|\mathbf{G}\|_{1,2}$ .
- 2: OUTPUT :  $\text{proj}_{B_{1,2}(R)}(\mathbf{G})$ .
- 3: Calculer :  $\|\mathbf{G}_{i,\cdot}^T\|_{\mathcal{X}}$  for  $i = 1 \dots n$ .
- 4: Trier  $\|\mathbf{G}_{i,\cdot}^T\|_{\mathcal{X}}$  dans l'ordre décroissant afin d'obtenir  $\mathbf{A}_n \leq \mathbf{A}_{n-1} \leq \dots \leq \mathbf{A}_1$ .
- 5: Calculer  $b_i := \|S_{a_i} \mathbf{G}\| = \sum_{j=1}^i (\mathbf{A}_j - \mathbf{A}_i)$  for  $i = 1 \dots n$ .
- 6: Trouver  $k$  tel que  $b_k \leq r < b_{k+1}$ .
- 7:  $\mu \leftarrow a_k + \frac{b_k - R}{k}$ .
- 8:  $\text{proj}_{B_{1,2}(R)}(\mathbf{G}) \leftarrow S_{\mu} \mathbf{G}$ .

**Algorithme 4.1:** Projection sur une boule de rayon  $r$  pour la norme  $\|\cdot\|_{1,2}$

nombre d'instances (sommets du graphe) et  $m$  le nombre de composantes (la dimensionnalité du vecteur associé à un sommet). Quand  $m > 1$ , les données sont donc multivaluées : à chaque sommet  $i$  est associé un vecteur  $\mathbf{F}_{i,\cdot}^T \in \mathbb{R}^m$ . Dans ce cas la décomposition est effectuée composante par composante, mais les paramètres d'échelle  $(\lambda_i)_{i \geq 0}$  sont communs à toutes les composantes. Pour tous les exemples que nous présenterons, le premier paramètre d'échelle  $\lambda_0$  sera déterminé automatiquement comme nous l'avons proposé. Cependant, pour les maillages colorés, ce paramètre a été fixé manuellement. Quand les données à traiter sont uni-dimensionnelles ( $\mathbf{F} \in \mathbb{R}^{n \times 1}$ ), la valeur de  $\lambda_0$  est fixée à  $3\|\mathbf{F} - \bar{\mathbf{F}}\|_{\mathcal{G}_w}/4$ . Lorsque les données sont multivaluées, la valeur de  $\lambda_0$  est fixée à  $\max_{1 \leq i \leq m} 3\|\mathbf{F}_i^T - \bar{\mathbf{F}}_i^T\|_{\mathcal{G}_w}/4$ . Dans ces deux cas, la norme  $\mathcal{G}_w$  est calculée avec l'algorithme que nous avons proposé.

### 4.6.1 Images

Nous commençons par considérer un graphe de voisinage très simple : un graphe grille en 4-voisinage non pondéré ( $w = 1$ ). La figure 4.6 présente les sommes des couches successives obtenues avec cette construction de graphe pour l'image "Barbara". On peut clairement constater la reconstruction qui s'effectue au fur et à mesure de la sommation des couches successives. Les premiers niveaux donnent une représentation très lisse et contenant uniquement les larges variations d'intensité de l'image initiale. À partir de la cinquième couche, une première description géométrique de l'image est obtenue. La première ligne de la figure 4.8 présente les couches successives extraites. On peut constater comment l'information de texture est reconstruite au fur et à mesure des décompositions en couches. Dans cette figure, la constante 120 a été rajoutée aux images afin de faciliter leur visualisation. Afin d'illustrer l'importance du graphe considéré, nous considérons cette fois un graphe

particulier. Chaque pixel est connecté à ses 10 plus proches voisins (en terme de distance entre patches de taille  $5 \times 5$ ) recherchés dans une fenêtre de taille  $21 \times 21$ . Les arêtes de ce graphe sont fixées avec un poids unitaire comme pour le graphe grille précédent. La deuxième ligne de la figure 4.8 présente les sommes des couches successives obtenues avec cette construction de graphe pour l'image de Barbara. On peut constater, en comparaison des résultats obtenus avec un graphe grille, que les premiers niveaux présentent une meilleure description géométrique de l'image initiale et la partie géométrique de l'image est reconstruite plus tôt dans les sommations effectuées. On remarquera également que dans les parties texturées de l'image, la construction non locale du graphe permet une meilleure reconstruction comparativement à un graphe grille. Pour quantifier ceci, la figure 4.9 présente l'erreur quadratique moyenne entre l'image originale et les sommes des couches successives. La figure 4.10 permet d'illustrer à nouveau l'intérêt d'une représentation de graphe adaptée aux données. Partant d'une image d'empreinte digitale naturellement bruitée, notre approche de décomposition est appliquée en utilisant différentes topologies de graphes avec différents poids :

- Un graphe grille en 4-voisinage non pondéré,
- Un graphe grille en 8-voisinage pondéré par des poids gaussiens sur les niveaux de gris des pixels,
- Un graphe des 25 plus proches voisins sélectionnés dans une fenêtre de recherche  $21 \times 21$  et pondéré par des poids gaussiens sur des patches  $5 \times 5$ .

On constate que dans le cas d'un graphe grille en 4-voisinage non pondéré, les contours sont flous dans les premiers niveaux de la décomposition. Utiliser un graphe pondéré en 8-voisinage permet d'améliorer cet aspect (les contours sont moins flous) mais de la texture est préservée dans les premiers niveaux et le bruit n'est pas concentré dans le dernier résidu. Avec un graphe exploitant une configuration non locale (par la topologie et les poids), on constate que la texture a complètement disparu du premier niveau et seule l'information géométrique a été capturée. La reconstruction est également plus homogène et chaque décomposition fournit une partie distincte de la texture initiale, ce qui permet une meilleure décomposition.

### **4.6.2 Maillages et nuages de points 3D**

Nous montrons à présent l'intérêt de notre approche au-delà des données classiques d'images organisées sur une grille régulière. Nous considérons pour cela le cas des maillages triangulés et des nuages de points 3D. Un maillage triangulé est un ensemble de sommets et d'arêtes organisés en faces triangulaires. Chaque sommet est décrit par un ensemble d'attributs géométriques et/ou photométriques. Les attributs géométriques sont les coordonnées 3D de chaque

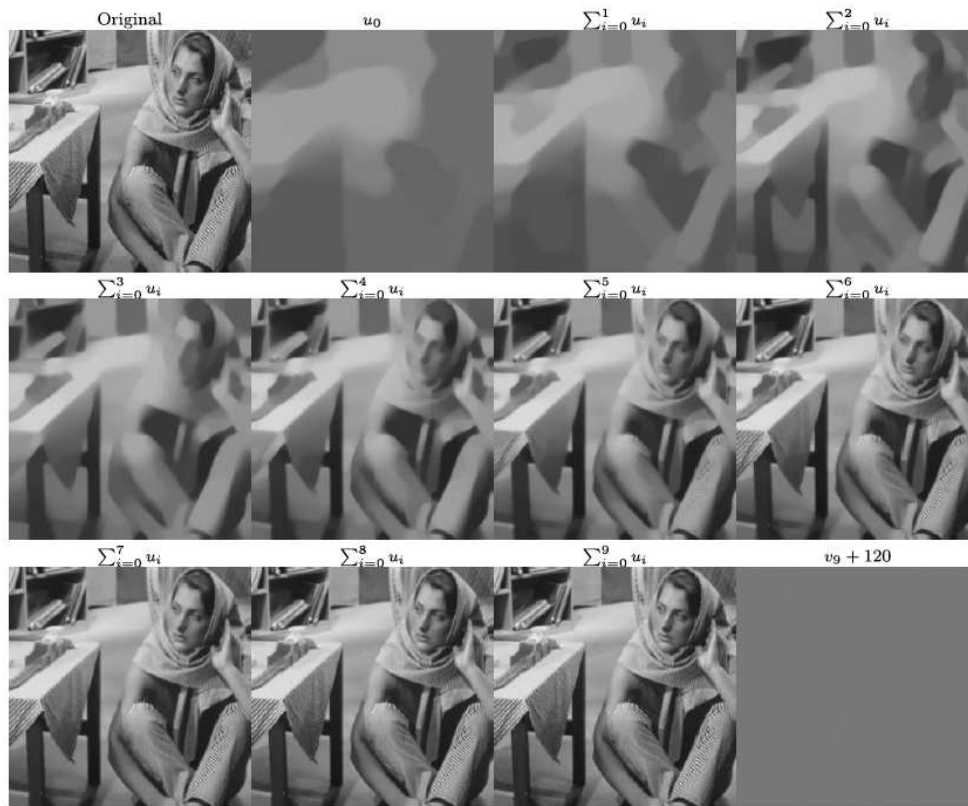


FIGURE 4.6 – Somme des couches successives extraites de l'image "Barbara". Les résultats sont obtenus avec un graphe grille non pondéré en 4-voisinage. La dernière image (en bas à droite) présente le dernier résidu.

sommet. Les attributs photométriques sont les couleurs de chaque sommet. Suivant le type d'attribut considéré, différents types de signaux sur graphes sont obtenus (le graphe est connu a priori). Nous considérerons, dans la suite, des maillages avec attributs géométriques et des maillages avec des attributs photométriques. Dans le premier cas (attributs géométriques), les poids sont fixés à  $w(u, v) = \frac{1}{1 + \|\mathbf{f}(u) - \mathbf{f}(v)\|}$ . Dans le second cas (attributs photométriques), les poids sont gaussiens.

La figure 4.11 présente les sommes des couches successives extraites à partir du signal géométrique d'un maillage 3D. Le résultat est donc obtenu en appliquant notre décomposition aux coordonnées des sommets du maillage. Comme pour les images, la représentation commence avec une version grossière des données initiales et les détails géométriques sont recouverts au fur et à mesure de la sommation des différentes couches. Au douzième niveau, le maillage original est complètement recouvert. Il faut noter toutefois que le

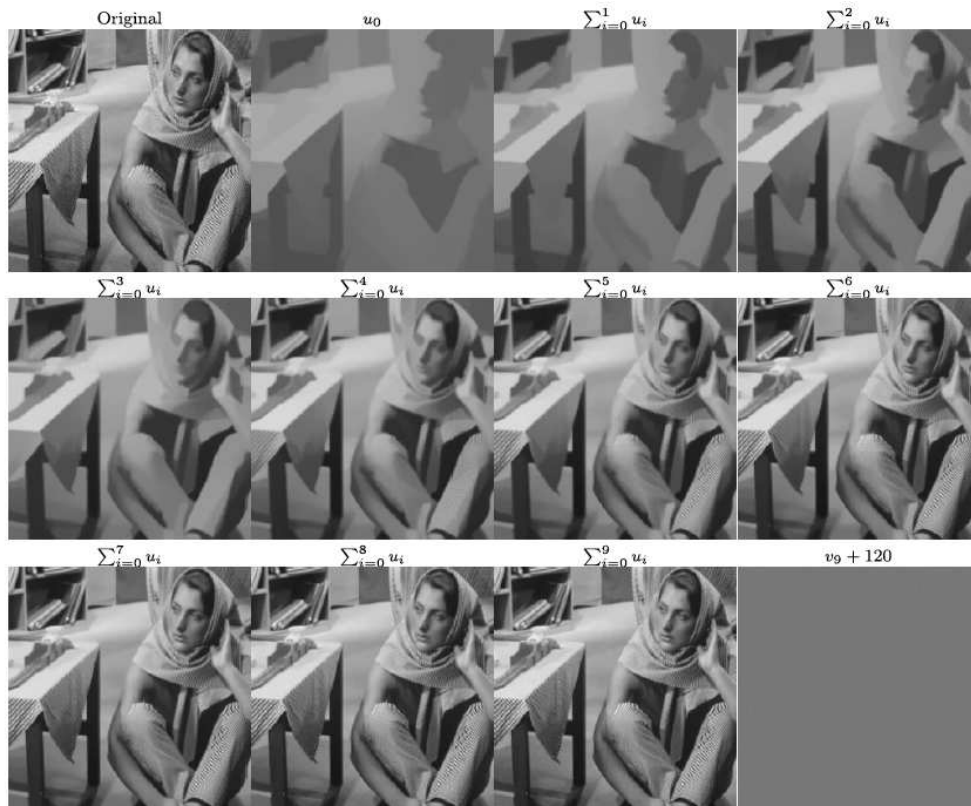


FIGURE 4.7 – Somme des couches successives extraites de l'image "Barbara". Les résultats sont obtenus avec un graphe non pondéré des 10 plus proches voisins. La dernière image (en bas à droite) présente le dernier résidu.

volume du maillage correspondant à  $u_0$  a été rétréci par rapport à celui du modèle original. Cela suggère que si cette propriété de préservation de volume du maillage est importante, elle devrait être ajoutée comme contrainte dans la fonctionnelle de régularisation.

La figure 4.12 présente les sommes des couches successives extraites à partir du signal photométrique d'un maillage 3D<sup>1</sup>. Le résultat est donc obtenu en appliquant notre décomposition aux couleurs *RGB* des sommets du maillage sans modifier les coordonnées de chaque sommet. On remarque que le dernier résidu contient encore de l'information de texture. La somme des six premières couches ( $i$  compris entre 0 et 5) peut être interprétée comme la partie structure d'une décomposition structure/texture du maillage coloré original. Ceci peut

1. Ce maillage 3D provient des données fournies par la société Cyberware Head & Face Color 3D Scanner et est disponible à <http://www.cyberware.com>.

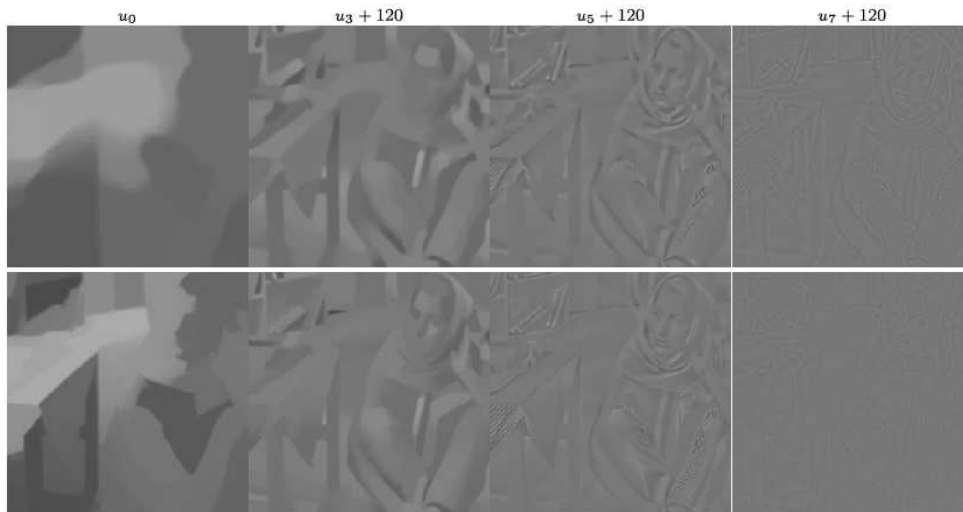


FIGURE 4.8 – Couches extraites de l'image "Barbara". La première ligne présente les résultats obtenus avec un graphe grille non pondéré en 4-voisinage. La seconde ligne présente les résultats obtenus avec un graphe non pondéré des 10 plus proches voisins.

être particulièrement intéressant si l'on souhaite effectuer une segmentation du maillage.

La figure 4.13 présente enfin les sommes des couches successives extraites à partir du signal géométrique d'un nuage de points 3D. Cette fois, le graphe n'est pas connu a priori et est construit comme un graphe des 10 plus proches voisins. Les arêtes sont pondérées par l'inverse de la distance euclidienne.

## 4.7 Applications en rehaussement de détails pour les images et les maillages

Afin de montrer l'intérêt de la décomposition multi-échelle que nous obtenons, nous montrerons comment celle-ci peut être exploitée dans le cadre d'applications en photographie numérique. En effet, de nombreux récents développements en photographie numérique [CM11, CT03, TT99, CPD07b, DD02, FLS08, FAR07, SSD09] décomposent une image en une couche de base lisse capturant les larges variations d'intensité et en une ou plusieurs couches de détails capturant les détails à plusieurs niveaux d'échelle. Dans la plupart de ces applications, les détails extraits sont ensuite manipulés à de multiples



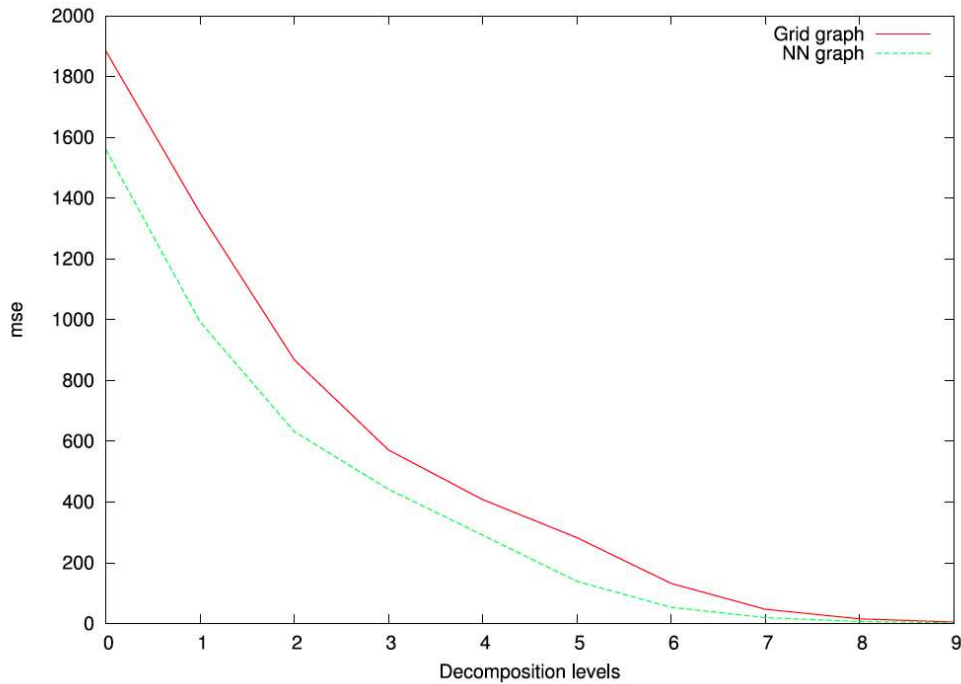


FIGURE 4.9 – MSE (Mean Squared Error) entre les somme des différentes couches et l'image originale "Barbara" pour les deux constructions de graphe (grille ou plus proches voisins).

échelles pour rehausser l'information initiale tout en évitant d'introduire des artéfacts visuels.

### 4.7.1 Comparaison avec l'état de l'art

Avant de présenter l'application de notre approche dans plusieurs applications de photographie numérique, nous comparons la décomposition que nous obtenons avec les approches les plus connues de l'état de l'art [CPD07b, FAR07, TT99, CT03, FFLS08]. Ces approches utilisent toutes des filtres dont les paramètres sont modifiés afin de simplifier de plus en plus l'image de résidu au fur et à mesure de l'extraction des couches de détails. Le principal défaut de ces approches est qu'il n'y a aucune garantie de convergence de la somme des couches obtenues vers l'image initiale. Ce n'est pas le cas de notre approche. Ceci est illustré par la figure 4.14. Nous avons considéré une décomposition d'une image initiale  $\mathbf{f}$  en 15 couches. La première colonne présente, pour chaque approche, une image avec sur la partie gauche la première couche obtenue  $\mathbf{u}_0$  (une version grossière de l'image) et sur la partie droite le



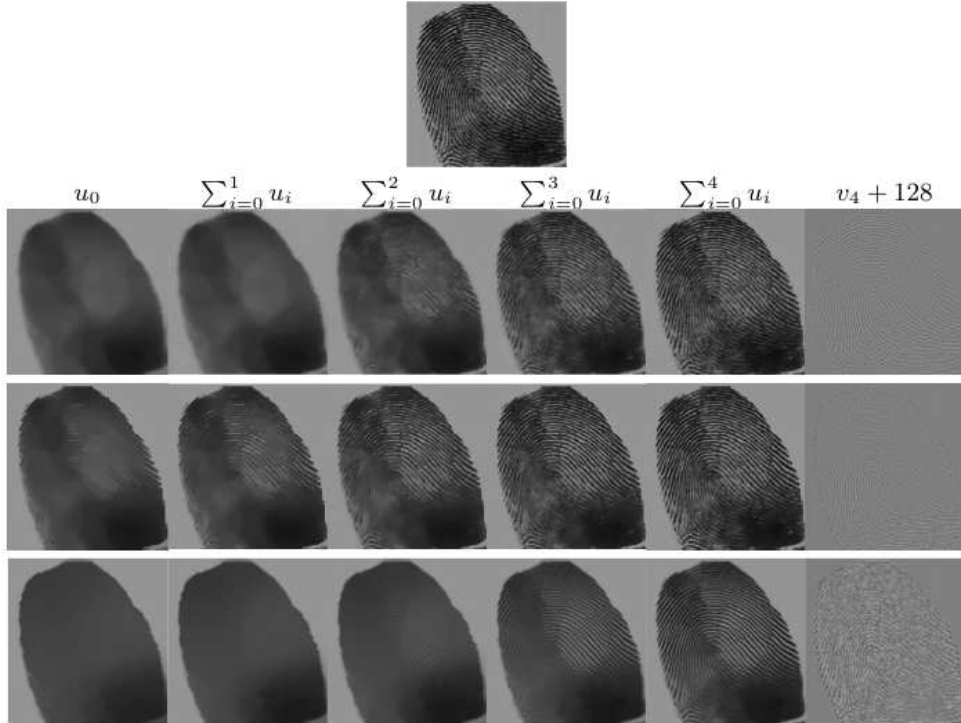


FIGURE 4.10 – Décomposition d’une image d’empreinte digitale. Première ligne : image originale. Deuxième ligne : graphe non pondéré de 4-voisinage. Troisième ligne : graphe de 8-voisinage pondéré par poids gaussiens. Quatrième ligne : 25-plus proches voisins avec poids gaussiens sur patches  $5 \times 5$ .

résidu  $\mathbf{v}_0$  (on a  $\mathbf{f} = \mathbf{u}_0 + \mathbf{v}_0$ ). La deuxième colonne présente sur la partie gauche une version intermédiaire de l’image initiale par la somme des 8 premières couches  $\left(\sum_{i=0}^7 \mathbf{u}_i\right)$  et sur la partie droite le résidu  $\left(\mathbf{f} - \sum_{i=0}^7 \mathbf{u}_i\right)$ . La dernière colonne présente sur la partie gauche la somme des 15 couches  $\left(\sum_{i=0}^{14} \mathbf{u}_i\right)$  et sur la partie droite le résidu  $\left(\mathbf{f} - \sum_{i=0}^{14} \mathbf{u}_i\right)$ . On constate tout de suite les bénéfices obtenus avec notre approche :

- Le dernier résidu ne contient quasiment pas de zones de contours contrairement aux autres approches.
- La première couche est réellement constante par morceaux, ne contient pas de texture ou de contours, mais bien uniquement les larges variations d’intensité de l’image initiale.

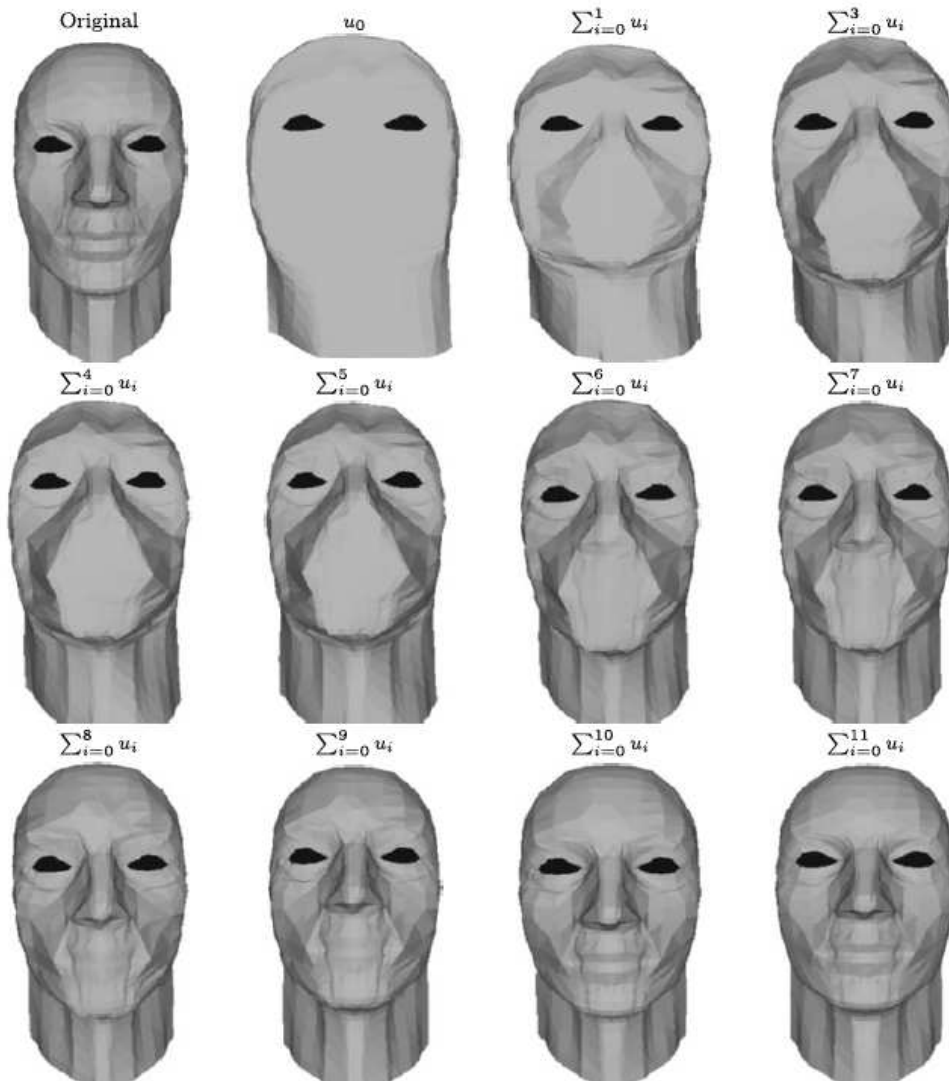


FIGURE 4.11 – Somme des couches successives extraites sur un maillage avec attributs géométriques (présenté en haut à gauche). La topologie du graphe est connue a priori et les arêtes sont pondérées par l'inverse de la distance euclidienne.



FIGURE 4.12 – Somme des couches successives extraites sur un maillage avec attributs photométriques (présenté sur la première ligne). La topologie du graphe est connue a priori et les arêtes sont pondérées par des poids gaussiens.

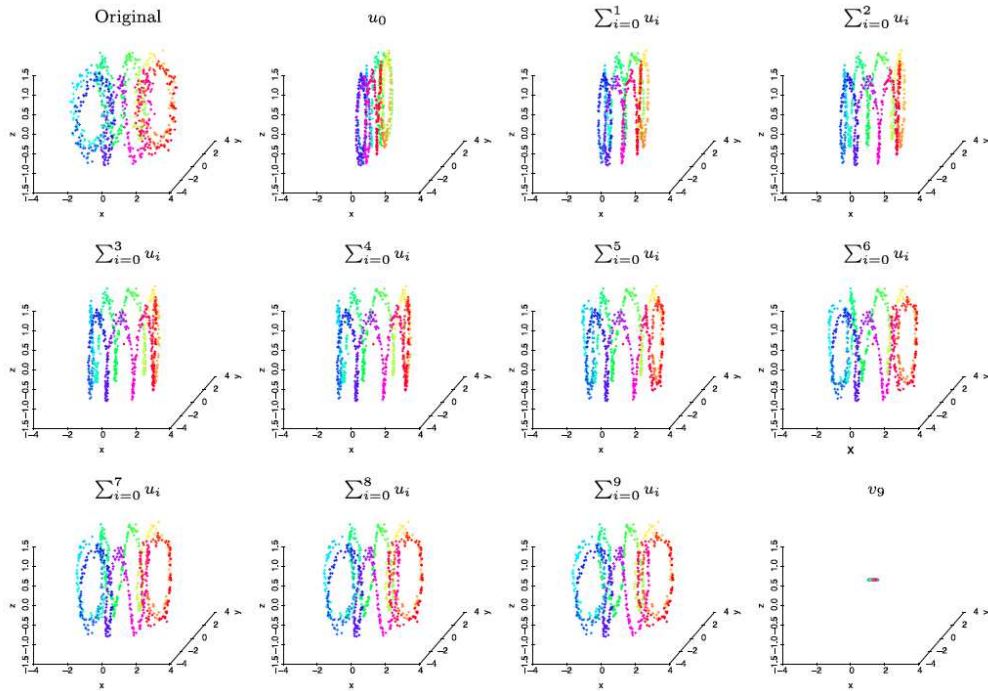


FIGURE 4.13 – Somme des couches successives extraites sur un nuage de points d’une hélice toroïdale bruitée. La topologie du graphe est un 10 plus proches voisins et les arêtes sont pondérées par l’inverse de la distance euclidienne. Chaque sommet du graphe est identifié par une unique couleur.

- Les sommes partielles obtenues ne contiennent pas d’effet de flou ou d’oscillations autour des arêtes. Ceci est particulièrement important car des oscillations présentes dans les couches produisent des halos lorsque les couches sont manipulées indépendamment et recombinaées ensuite [FFLS08].

## 4.7.2 Rehaussement de contours

Nous présentons tout d’abord une première application pour le rehaussement de contours dans une image. Une fois que l’image a été décomposée en  $n$  couches, nous proposons d’obtenir un rehaussement des contours (ou bien une amélioration de la netteté) de la manière suivante : trois niveaux de rehaussement sont considérés (grosier, intermédiaire, et fin) qui contiennent chacun un rehaussement des couches de détails. Partant d’une image originale

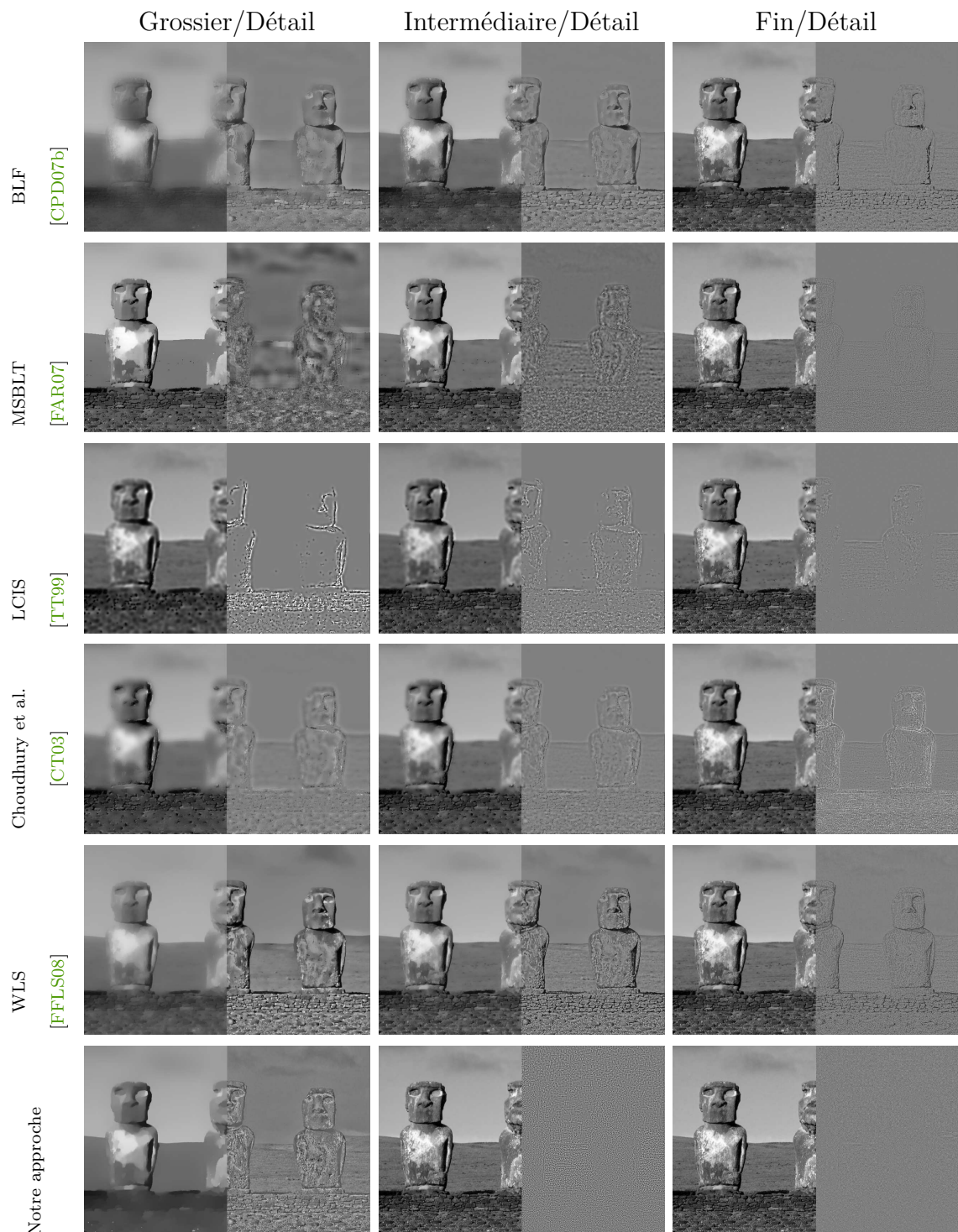


FIGURE 4.14 – Comparaison des résultats obtenus par notre approche (8-voisinage, poids gaussiens) sur une image avec différentes approches de l'état de l'art.

$\mathbf{f}$  et de sa décomposition en  $n$  couches  $\mathbf{u}_i$ , une version grossière est obtenue par :

$$g_{grossier} = \sum_{i=0}^{i=l_1} (1 + i\delta_1)\mathbf{u}_i \quad (4.50)$$

Une version intermédiaire est obtenue par :

$$g_{intermediaire} = g_{grossier} + \sum_{i=l_1+1}^{i=l_2} (\delta_2 + (i - l_1 - 1)\delta_1\delta_2)\mathbf{u}_i \quad (4.51)$$

Une version fine est obtenue par :

$$g_{fin} = g_{intermediaire} + \sum_{i=l_2+1}^{i=n-1} (\delta_2^2 + (i - l_2 - 1)\delta_1\delta_2)\mathbf{u}_i \quad (4.52)$$

Des coefficients de rehaussements croissants sont utilisés pour rehausser chacun de ces niveaux. Les valeurs de ces coefficients sont fixes et valent respectivement  $\delta_1 = 2.5$  et  $\delta_2 = 0.25$ . Afin d'automatiser la détermination des niveaux de décompositions qui sont considérés comme intermédiaires ou fins, nous déterminons automatiquement les valeurs de  $l_1$  et  $l_2$  par

$$l_1 = \left\{ i \left| \frac{255}{\max(\mathbf{u}_i) - \min(\mathbf{u}_i)} > s_1 \right. \right\} \quad (4.53)$$

et

$$l_2 = \left\{ i \left| \frac{255}{\max(\mathbf{u}_i) - \min(\mathbf{u}_i)} > s_2 \right. \right\} \quad (4.54)$$

Les valeurs par défaut que nous considérons sont  $s_1 = 5$  et  $s_2 = 20$  (déterminées expérimentalement). La figure 4.15 présente les résultats de notre approche en comparaison avec l'approche standard WLS [FFLS08]. Nous considérerons toujours des résultats obtenus avec un graphe grille en 8-voisinage avec poids gaussiens ( $s_1 = 10$ ). On constate bien tout d'abord que les détails sont accentués de manière croissante du niveau grossier vers le niveau fin. La dernière colonne de la figure 4.15 présente la combinaison moyenne des trois niveaux d'accentuation des détails. Nos résultats sont très différents de ceux obtenus avec WLS dont la méthode induit une forte décoloration du résultat relativement à l'image initiale.

La figure 4.16 présente une comparaison avec des méthodes de l'état de l'art pour un rehaussement fin. Nos résultats sont similaires à ceux de [CM11] mais bien meilleurs que les autres approches qui présentent des effets de halos blanc autour des feuilles de la fleur ainsi que des rehaussement artificiels de certaines zones dans le fond de l'image. Les différences entre notre approche



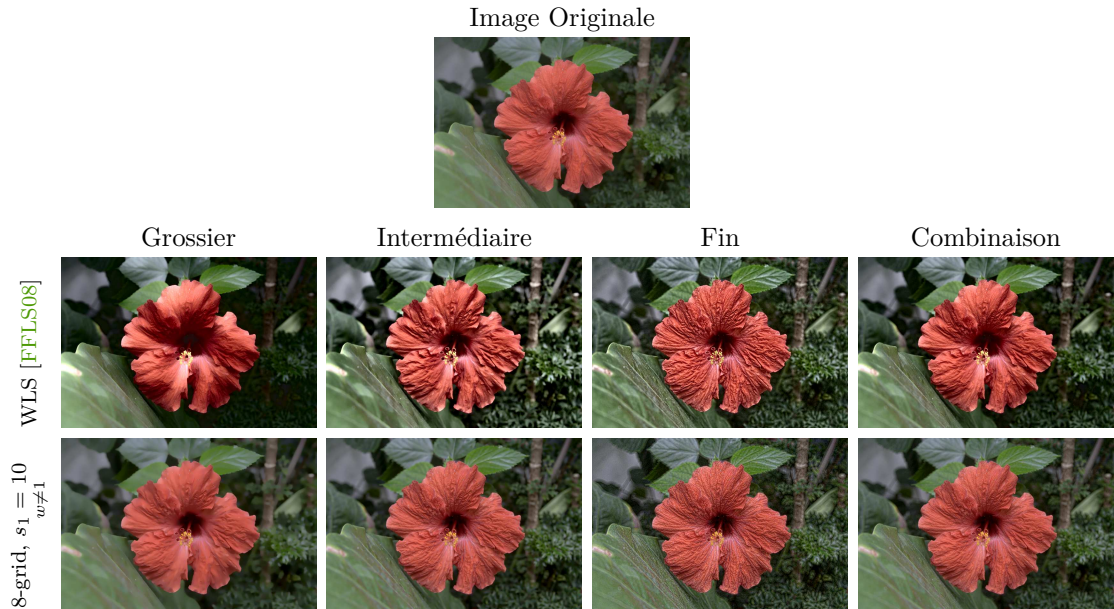


FIGURE 4.15 – Sharpening d’une image de fleur.

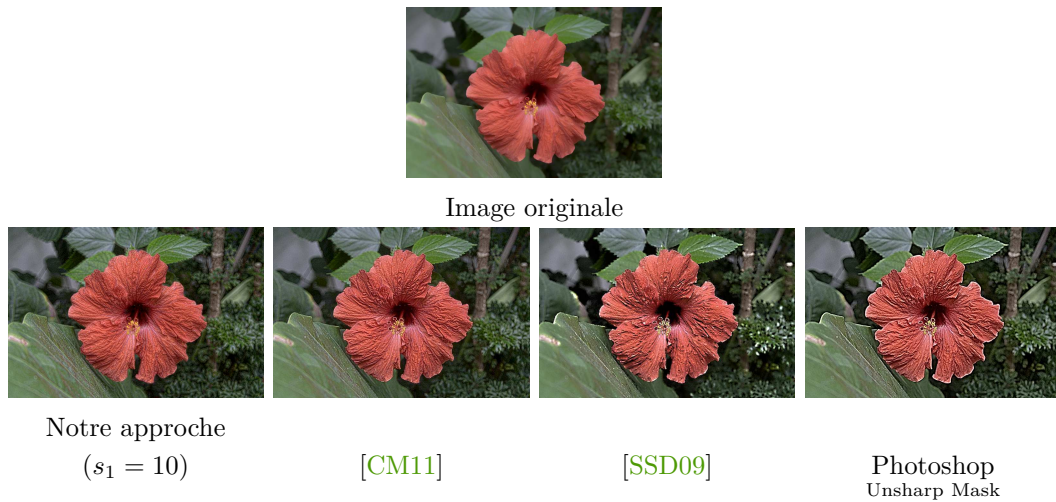


FIGURE 4.16 – Comparaison de notre approche de rehaussement de détails fins avec des approches de l’état de l’art sur l’image de fleur de la figure 4.15.

et [CM11] sont visibles dans les rainures de la fleur qui sont toutes rehaussées globalement par notre approche.

Ceci est particulièrement évident sur la figure 4.17 qui présente une image initiale d'une feuille de rhubarbe dont les détails fins ont été rehaussés par notre approche et les méthodes BLF [CPD07b], WLS [FFLS08]. On remarque que seule notre approche permet de conserver une luminance similaire à l'image initiale tout en accentuant de manière globale les détails même les plus fins.

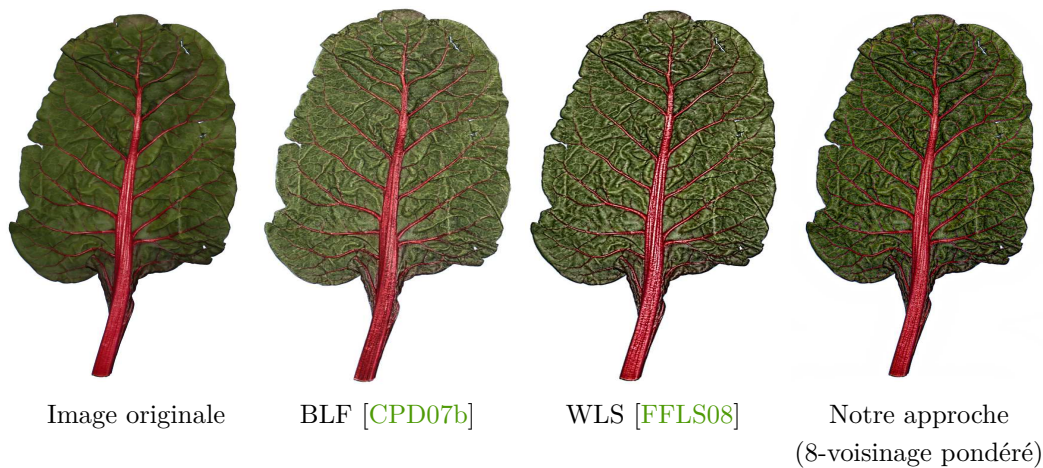


FIGURE 4.17 – Rehaussement des détails d'une feuille de rhubarbe.

La figure 4.18 met à nouveau en avant notre approche par rapport aux méthodes BLF et WLS. La partie gauche présente l'image avec rehaussement fin et la partie à droite du trait blanc présente la version avec rehaussement intermédiaire. Le rectangle blanc présente un zoom d'une zone de l'image mettant en avant l'effet (indésirable) de halo avec BLF. Notre approche ne produit pas de halos contrairement à BLF, et de plus, contrairement à WLS, elle ne modifie pas la luminosité de l'image ni ne rehausse artificiellement des zones blanches.

### 4.7.3 Tone mapping

Le tone mapping d'images HDR (High Dynamic Range) est une application en photographie numérique qui peut également être effectuée en décomposant une image HDR en une couche de base et une couche de détails [TT99, DD02]. La couche de base est ensuite transformée de manière non linéaire vers une image de plus faible dynamique et est recombinaée avec la couche de détails. La couche de base doit préserver les discontinuités et ne pas produire de halos [TT99]. Elle doit également être suffisamment lisse pour conserver un contraste



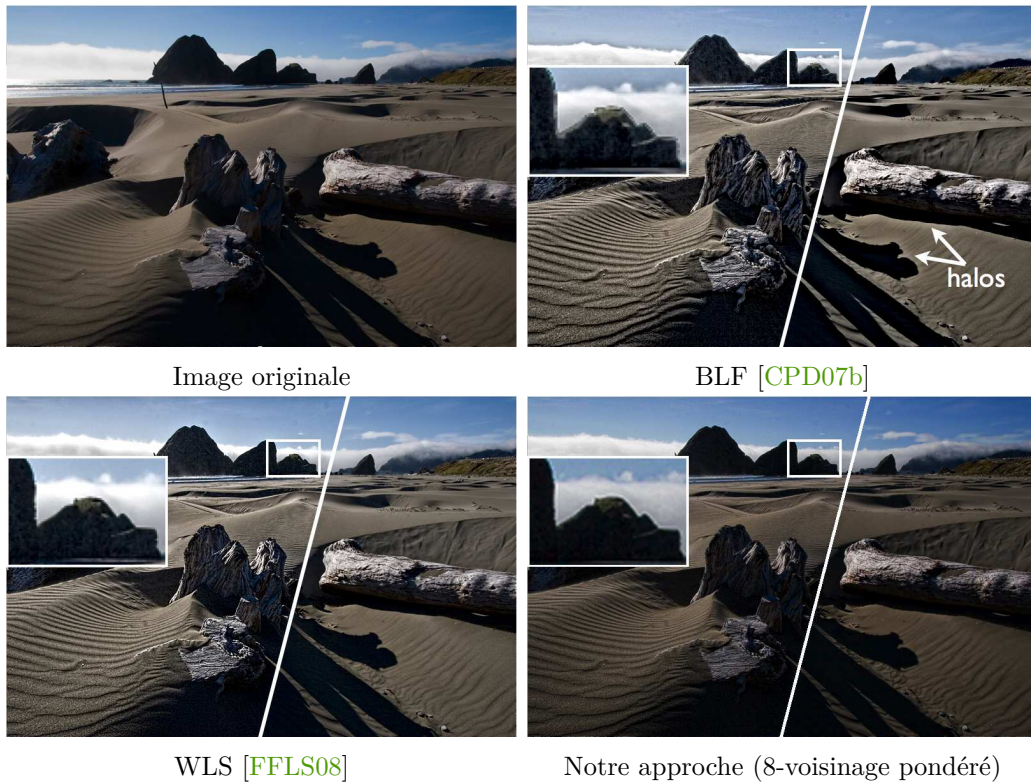


FIGURE 4.18 – Illustrations des problèmes de halo (BLF) et de modification d’intensité globale (WLS).  $\delta_1 = 2$ ,  $\delta_2 = 0.05$ .

raisonnable après réduction de la dynamique [CT03]. La figure 4.18 résume le principe du tone mapping selon l’approche de Durand et Dorsey [DD02]. En nous basant sur leur framework de tone mapping [DD02], nous utilisons notre méthode de décomposition en lieu et place de la leur reposant sur un filtre bilatéral. Un résultat est montré dans la figure 4.20. On peut constater que les structures importantes sont bien préservées ou rehaussées dans l’image finale obtenue après tone mapping. À nouveau le rehaussement global effectué par notre méthode est bien plus prononcé que pour les autres méthodes (à la fois l’intérieur et l’extérieur de la pièce ont leurs détails de rehaussés).

#### 4.7.4 Retouche virtuelle de portraits

Nous présentons à présent une application pour la retouche virtuelle de portraits. Il existe de nombreux programmes commerciaux, vendus sous forme de plugins dans Photoshop, qui permettent d’effectuer de la retouche d’images. Souvent cette retouche est limitée à du débruitage, mais certains plugins

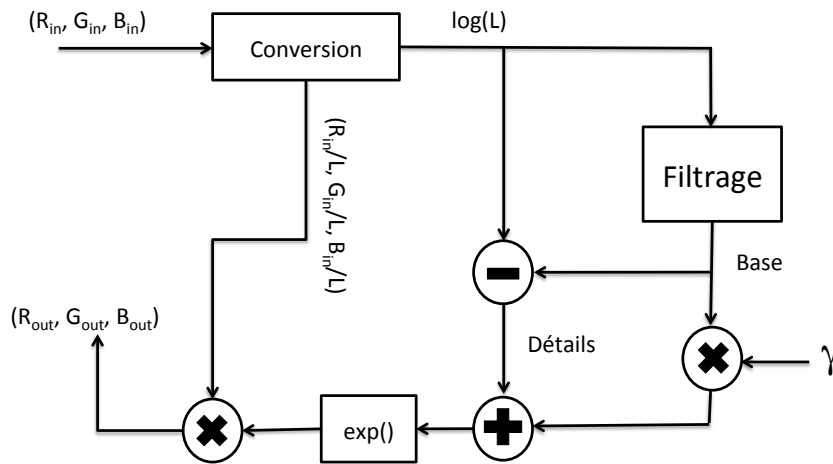
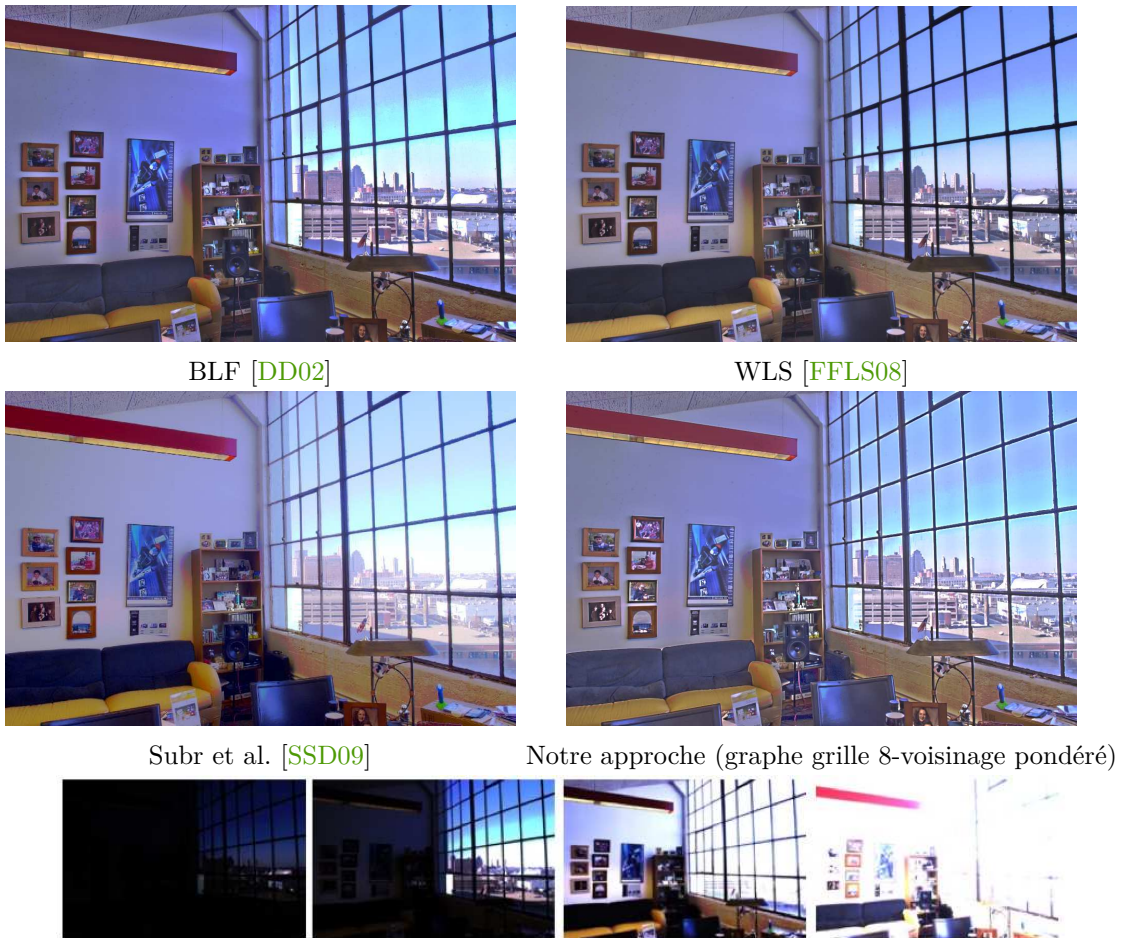


FIGURE 4.19 – Principe du tone mapping [DD02].

dédiés au domaine de la presse et de la mode ont pour objectif de retoucher virtuellement des photos de portraits. Ce type de retouches s'intéresse plus particulièrement à la peau des personnes apparaissant sur ces portraits. Il permet de lisser la peau, de supprimer certaines aspérités ou de l'acné, tout en améliorant le visuel final du portrait. L'effet recherché est exactement l'inverse de celui du rehaussement : on cherche ici à atténuer des détails de peau gênants. La plupart de ces techniques nécessitent une forte interaction avec l'utilisateur car elles requièrent de délimiter les zones de peau à retoucher et de spécifier le type de traitement ad hoc à appliquer à chaque zone. Nous proposons d'effectuer une retouche virtuelle de portraits, toujours de manière interactive, mais cette fois en limitant l'interaction à la sélection de couches de détails. Le principe est le suivant. Partant d'une décomposition en  $n$  couches d'une image, une nouvelle image est reconstruite en sommant les couches mais en supprimant quelques unes des couches de la décomposition. La figure 4.22 présente un exemple d'une telle manipulation où, pour supprimer l'acné, les 4 premières couches de détails ont été supprimées de la reconstruction de l'image. On peut constater que l'effet visuel de retouche est bien atteint : la peau est plus lisse et sans défauts.

Pour éviter un aspect trop abstrait, comme dans la figure 4.22, la manipulation des couches peut se faire de manière interactive et permettre d'obtenir des résultats visuellement très réalistes et moins artificiels que ceux obtenus avec le plugin Portraiture de photoshop [Plu] (voir figure 4.22).



L'image HDR originale obtenue à partir de différentes expositions de la même scène.

FIGURE 4.20 – Comparaison avec différents algorithmes de tone mapping.

#### 4.7.5 Rehaussement de détails sur des maillages

Pour terminer ces illustrations, nous montrons l'aspect générique de notre approche en effectuant un rehaussement de détails sur des maillages. Nous considérons deux types de rehaussement : celui de maillages classiques où une fonction associe à un noeud ses coordonnées 3D et celui de maillages colorés ou une fonction associe à un noeud sa couleur *RGB*. Effectuer une décomposition d'un maillage classique permettra d'obtenir des versions où l'apparence du maillage peut varier d'un maillage grossier à un maillage fin rehaussé (même si le nombre de noeuds du maillage ne varie pas). Effectuer une décomposition d'un maillage coloré permettra d'obtenir des versions où l'apparence des couleurs du maillage peut varier de grossières à fines rehaussées.

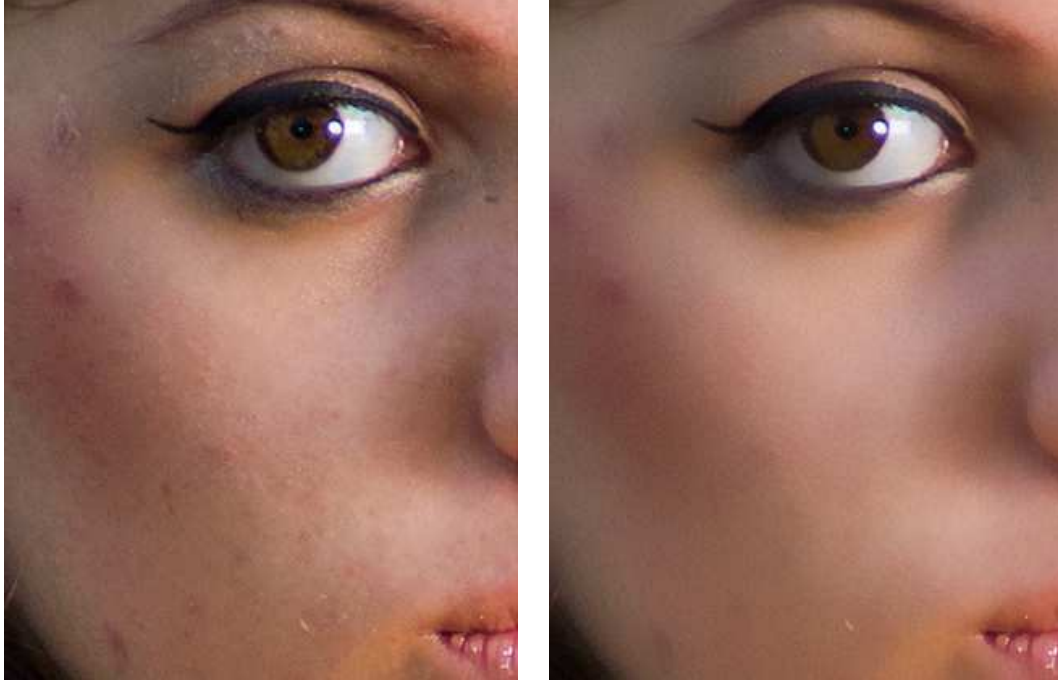


Image originale

Couches  $\mathbf{u}_1$  à  $\mathbf{u}_4$  supprimées.

FIGURE 4.21 – Retouche d'un portrait pour un effacement d'acné.

Tout comme pour les images, nous définissons trois niveaux de rehaussement de détails d'un maillage. Ces niveaux de rehaussement (grossier, intermédiaire et fin) sont obtenus par :

$$g_{grossier} = \sum_{i=0}^{i=l_1} (1 + i\delta_1)\mathbf{u}_i \quad (4.55)$$

$$g_{intermédiaire} = g_{grossier} + \sum_{i=l_1+1}^{i=l_2} (1 + i\delta_1)\mathbf{u}_i \quad (4.56)$$

$$g_{fin} = g_{intermédiaire} + \sum_{i=l_2+1}^{i=n-1} (1 + i\delta_1)\mathbf{u}_i \quad (4.57)$$

Les valeurs par défaut que nous considérons pour les paramètres de ce rehaussement sont  $\delta_1 = 0.15$ ,  $l_1 = n/3$  et  $l_2 = 2n/3$  (déterminés expérimentalement). Les figures 4.23 et 4.24 présentent des rehaussements obtenus pour deux maillages dont l'un comporte très peu de noeuds. On constate bien que les détails sont accentués de manière croissante du niveau grossier vers le niveau fin. Les figures 4.25 et 4.26 présentent des rehaussement de couleurs sur des





FIGURE 4.22 – Retouche interactive d'un portrait et comparaison avec le plugin Portraiture de photoshop.

maillages obtenus à partir de scanners 3D récents qui capturent simultanément la géométrie et la photométrie d'un maillage. Ces scanners ont souvent néanmoins le défaut de produire des couleurs qui paraissent globalement floues sur la surface du maillage. Notre approche de rehaussement de détails est alors tout particulièrement adaptée à ce cas de figure. On peut constater que la version rehaussée fine présentée dans les figures 4.25 et 4.26 permet d'obtenir un maillage dont la couleur globale paraît visuellement beaucoup plus nette. Il est important de noter qu'il n'existe à l'heure actuelle aucune méthode équivalente permettant d'effectuer un tel rehaussement de détails sur des maillages colorés. Ceci est d'un très grand intérêt notamment pour l'industrie cinématographique qui utilise maintenant fréquemment des répliques 3D des acteurs par un scan complet de leur corps pour les insérer ensuite dans des scènes réelles ou virtuelles.

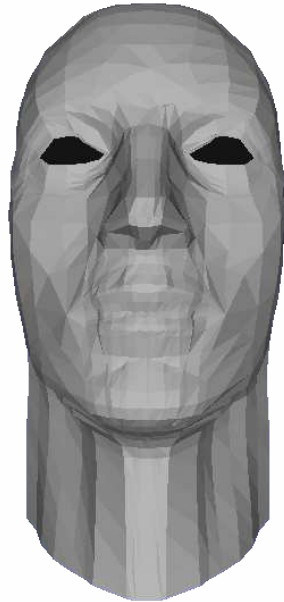
## 4.8 Conclusion

Nous avons introduit dans ce chapitre deux décompositions multi-échelles de signaux définis sur l'ensemble des sommets de graphes pondérés généraux. Ces décompositions sont inspirées de l'approche multi-échelle hiérarchique de [TNV04]. Nous avons étendu cette approche au cadre général des signaux sur graphes grâce aux outils variationnels introduits dans le chapitre précédent.

Nous avons étudié les propriétés des décompositions proposées. Pour la fonctionnelle quadratique, le lien avec le filtrage spectral permet de conclure rapidement à la convergence de la représentation. Le choix du premier paramètre n'est cependant pas immédiat, car il nécessite la connaissance de tout le spectre de la matrice laplacienne.

Les propriétés de la décomposition basée sur l'a priori TV ont été étudiées au moyen de techniques d'optimisation convexe. En particulier, nous avons vu que la caractérisation de la TV sur graphes comme fonction support permettait de conclure directement à la convergence de la représentation. La vitesse de convergence, mesurée avec la norme  $\mathcal{G}_w$ , que nous avons définie pour des signaux sur graphes, est reliée directement à la décroissance des paramètres d'échelle. Nous avons également proposé le calcul du premier paramètre en calculant la norme  $\mathcal{G}_w$  du signal (centré) que l'on souhaite décomposer.

Nous avons enfin considéré l'application de notre décomposition au rehaussement de détails. Nous avons pu, au sein d'un même cadre méthodologique, considérer le rehaussement de détails pour les images ainsi que pour les maillages 3D avec attributs géométriques et photométriques.



Maillage original



Maillage grossier



Maillage intermédiaire



Maillage fin

FIGURE 4.23 – Rehaussement d'un maillage comportant peu de noeuds (1487 noeuds, 2918 faces).

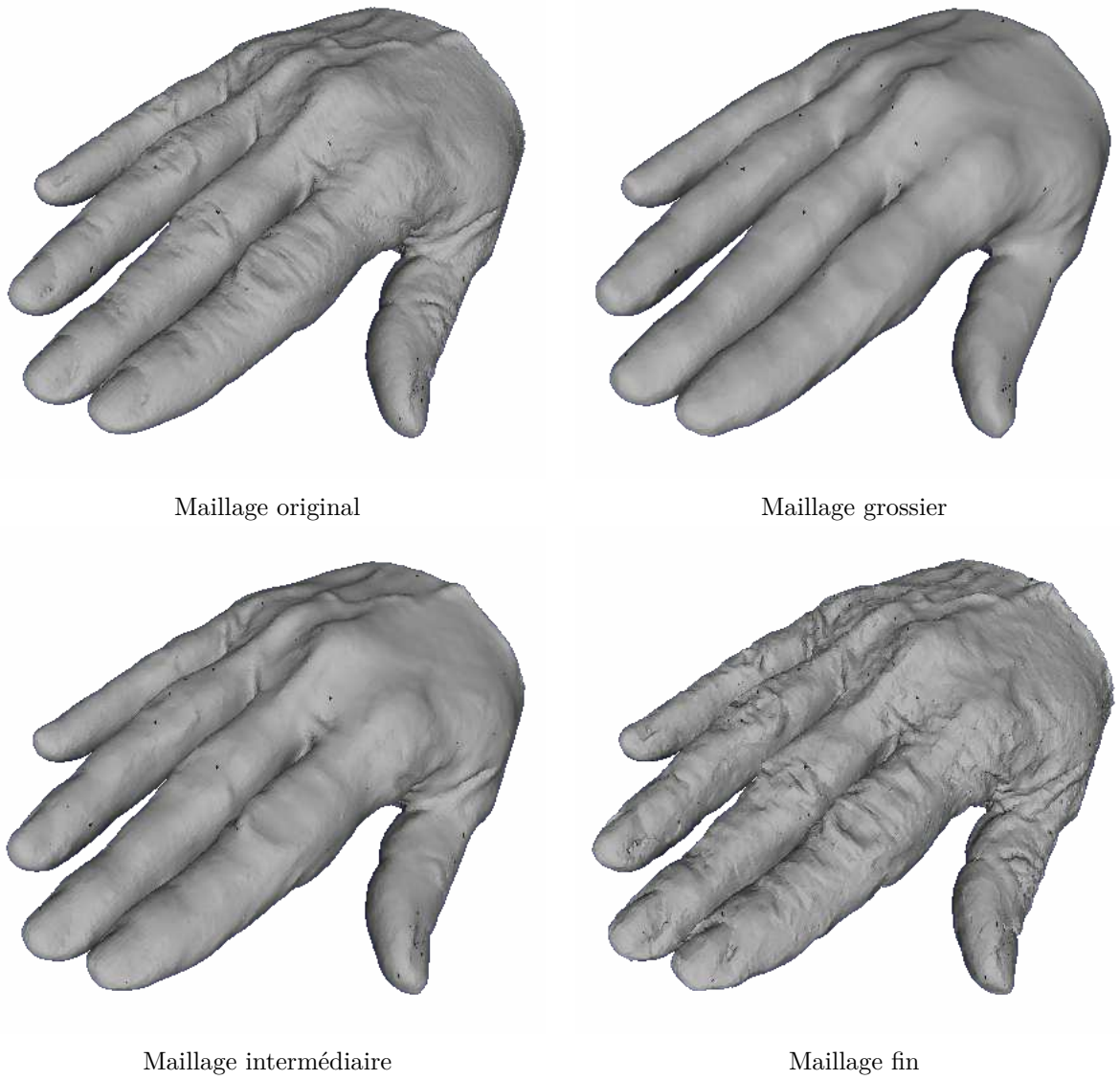


FIGURE 4.24 – Rehaussement d'un maillage (43533 noeuds, 88002 faces).





Maillage coloré original



Maillage coloré grossier



Maillage coloré intermédiaire



Maillage coloré fin

FIGURE 4.25 – Maillage coloré de faible qualité rehaussé (17215 noeuds, 29461 faces).



Maillage coloré Original



Maillage coloré grossier



Maillage coloré intermédiaire



Maillage coloré fin

FIGURE 4.26 – Maillage coloré de haute qualité rehaussé (553053 noeuds, 1105611 faces).



# Chapitre 5

## Transformées multi-échelles de signaux sur graphes

### 5.1 Introduction

Nous introduisons dans ce chapitre deux transformées multi-échelles agissant sur des signaux définis sur l'ensemble des sommets d'un graphe pondéré général. Ces transformées donnent lieu à des décompositions multi-échelles similaires à celles proposées dans le chapitre précédent. La spécificité des transformées que nous considérons ici est qu'elles correspondent à des développements dans des *bases multi-échelles* dont les éléments ont des *supports localisés*. Ces transformées correspondent donc à de nouvelles représentations des signaux en question.

Ce chapitre est organisé de la manière suivante. Dans la section 5.2, nous présentons rapidement l'intérêt des approches multi-échelles dans le cadre classique du traitement du signal et de l'image ainsi que les difficultés rencontrées lors de l'adaptation de ces approches aux signaux sur graphes. Plusieurs constructions de transformées multi-échelles sur des graphes ont été proposées récemment [CK03, CM06, HVG11, SRV12, NO12]. Nous examinons dans la section 5.3 les transformées basées sur les ondelettes de diffusion introduites par Coifman et Maggioni dans [CM06] et sur les ondelettes spectrales introduites par Hammond et al. dans [HVG11]. Ces deux constructions ont pour point commun d'interagir avec le graphe sous-jacent via des matrices associées (matrice laplacienne dans [HVG11] et matrice de transition dans [CM06]). Dans la section 5.4, nous abordons la construction d'ondelettes de Haar sur une hiérarchie de partitions. Cette approche est directement inspirée de [GNC10]. Nous rappelons dans cette section l'algorithme que nous avons introduit dans [HLE11b, HLE11a] permettant la construction d'une hiérarchie de partitions

à partir d'un graphe pondéré. Dans la section 5.5 nous développons l'approche basée sur le lifting. Nous détaillons l'adaptation de cet algorithme aux signaux sur graphes et discutons les difficultés d'une telle adaptation. La difficulté principale concerne l'étape de partitionnement des sommets. Nous optons ici pour une approche inspirée par [JNS09], où un unique coefficient de détail est calculé à chaque étape du lifting. Nous détaillons la sélection des sommets, l'effet sur les fonctions de base, ainsi que l'extension de la transformée à plusieurs cycles.

## 5.2 Transformées multi-échelles pour les signaux et les images

Pour des signaux discrets échantillonnés régulièrement, la base discrète de Fourier présente deux avantages importants. D'abord, elle diagonalise toutes les transformations linéaires invariantes par translation, et donc tous les filtres numériques. Ensuite, la transformée correspondante peut être effectuée de manière rapide et efficace grâce à l'algorithme de la transformée de Fourier rapide. Cependant, pour une large classe de signaux, la transformée de Fourier présente de nombreux problèmes dus principalement au fait que les vecteurs de la base associée ne sont pas à support compact. Ceci a pour effet que la variation d'une seule composante temporelle ou spatiale entraîne la modification de toutes les composantes dans le domaine spectral.

La base canonique de  $\mathbb{R}^N$  fournit une base parfaitement localisée en temps. Cependant elle présente le même problème que la base de Fourier discrète, cette fois dans le domaine spectral : la variation d'une seule composante dans le domaine spectral affecte toutes les composantes temporelles.

Les bases d'ondelettes [Mal09] permettent la localisation simultanée en temps et en échelle. L'impact le plus important de cette localisation est que ces bases permettent de représenter de larges classes de signaux de manière parcimonieuse, c'est-à-dire avec peu d'entrées non nulles, ou plus généralement de manière compressible, autrement dit avec peu de coefficients de magnitudes élevées.

Une application naturelle des transformées en ondelettes concerne la compression avec pertes de signaux digitaux ([VK95, DN+99]). Le procédé consiste à quantifier puis à effectuer un codage entropique des coordonnées du signal que l'on souhaite compresser dans la base d'ondelettes utilisée. L'approximation du signal original a lieu lors de l'étape de quantification. Les coefficients ainsi quantifiés et codés sont transmis ou stockés. L'approximation

du signal original est enfin effectuée en appliquant la transformée inverse aux coefficients stockés ou transmis.

Une autre application concerne le débruitage de signaux altérés par un bruit blanc gaussien. Initialement proposée par Donoho et Johnstone [DJ94], l'approche *wavelet shrinkage* consiste à *seuiller* individuellement les coefficients du signal bruité dans une base d'ondelettes. Cette approche est similaire à la quantification de ces coefficients en vue de la compression. Le seuillage doux des coefficients d'ondelettes, introduit dans [DJ94], peut être interprété de manière variationnelle [CDVLL98].

Les approches précédentes sont réalisables en pratique car les transformées directes et inverses peuvent être implémentées de manière efficace grâce à l'algorithme de la transformée en ondelettes rapide, développé par Mallat [Mal09]. Un autre algorithme, dit du *lifting*, introduit par Sweldens dans [Swe96, Swe98] permet également d'effectuer rapidement les transformées directes et inverses. Dans [DS98], Daubechies et Sweldens montrent que toute transformée en ondelettes basée sur des filtres finis peut être implémentée grâce à l'algorithme du *lifting*. Un premier intérêt de cet algorithme est qu'il ne nécessite pas de mémoire supplémentaire : tous les calculs intermédiaires peuvent être effectués en utilisant l'espace mémoire occupé par le signal original.

Au-delà de l'avantage algorithmique qu'il présente pour les signaux échantillonnés de manière régulière, l'algorithme du *lifting* permet également de construire des bases d'ondelettes dites de seconde génération. Il s'agit, en particulier, de bases pour des signaux définis sur des courbes ou des surfaces, ou encore des signaux dont l'échantillonnage est irrégulier. Nous détaillerons cet algorithme dans la section 5.5 où nous proposerons son application à des signaux définis sur l'ensemble des sommets de graphes pondérés généraux.

Une application récente de la parcimonie ou de la compressibilité des représentations des signaux dans des bases d'ondelettes concerne les solutions de problèmes inverses mal posés. Dans ce cadre ([Mal09, SMF10, DDDM04, DFL08]), la parcimonie est conçue comme un *a priori* : lors de l'inversion d'un opérateur linéaire, on cherche à favoriser les signaux dont la représentation est la plus parcimonieuse possible dans une base donnée. Cet *a priori* est mis en oeuvre en pratique en pénalisant la norme  $\ell_1$  de la nouvelle représentation.

En dimension 2, les transformées multi-échelles classiques sont généralement obtenues par produit tensoriel de transformées unidimensionnelles. Il s'agit là de transformées séparables, capables de détecter des discontinuités horizontales ou verticales. Pour le traitement de l'image, un nombre important de travaux a été dédié à la conception de transformées dites géométriques, c'est-à-dire capables de représenter de manière efficace des images présentant des

discontinuités le long de contours qui ne sont ni verticaux ni horizontaux. Le lecteur intéressé par une introduction à ce sujet pourra consulter l'article récent [JD<sup>CP</sup>11] ainsi que les multiples références y fournies.

Les décompositions que nous venons de mentionner plus haut correspondent à des développements des signaux dans des bases orthonormées ou des trames (frame en anglais). Récemment, un intérêt croissant a été porté à la construction de transformées redondantes. Dans ce cadre, on considère un dictionnaire constitué par la concaténation de plusieurs bases ou trames et on cherche à y exprimer de manière efficace les signaux que l'on souhaite étudier. Du fait de la redondance du dictionnaire, il existe une multitude de représentations d'un même signal. La recherche de la représentation la plus parcimonieuse est alors formulée de manière variationnelle; des algorithmes dits de *poursuite* sont alors employés. Le lecteur intéressé pourra consulter l'ouvrage récent [Ela10] pour une introduction à ce type d'approches.

Toutes les approches mentionnées précédemment considèrent des bases, des trames, ou des dictionnaires fixes et les transformées correspondantes sont linéaires. Un autre type d'approche consiste à construire une base à partir du signal dont on dispose. Cette approche donne lieu à des transformées multi-échelles non linéaires et permet d'adapter les fonctions de base au signal que l'on souhaite analyser. L'algorithme du lifting, introduit par Sweldens [Swe96, Swe98], permet de construire des transformées multi-échelles adaptatives et inversibles en considérant des opérateurs de prédiction non linéaires. Cette approche a été considérée notamment dans [PH02, CDSB03, HPPP05] pour la compression de signaux. Notons enfin que des transformées multi-échelles non linéaires ont été proposées dans [AD00, CM01, ADLT07]. Ces transformées ne sont plus basées sur l'algorithme du lifting mais sur une approche multirésolution très proche développée par Harten dans [Har93, Har96].

### 5.2.1 Difficultés d'adaptation pour les signaux sur graphes

L'efficacité des méthodes basées sur des transformées multi-échelles pour l'étude et le traitement du signal et de l'image justifie l'intérêt d'introduire des transformées multi-échelles pour des signaux définis sur l'ensemble des sommets de graphes pondérés généraux. Ces transformées seront alors être utilisées sur des données dès qu'une relation d'adjacence pondérée pertinente peut y être définie.

Les transformées en ondelettes classiques sont obtenues en calculant, à l'aide d'algorithmes rapides, des produits scalaires avec des translatés et dilatés d'une unique ondelette mère. Dans sa version continue, la transformée en ondelette est générée à partir d'une ondelette mère  $\psi$  admissible [D<sup>+</sup>92]. La transformée

directe associe donc à  $f$  la fonction  $W_f$  de deux variables, donnée pour tout  $t > 0$  et tout  $a \in \mathbb{R}$  par

$$W_f(t, a) = \int_{\mathbb{R}} \frac{1}{t} \psi^*\left(\frac{x-a}{t}\right) f(x) dx. \quad (5.1)$$

L'adaptation de cette approche à des signaux définis sur l'ensemble des sommets d'un graphe général pose des problèmes évidents. Tout d'abord, la définition d'une notion de translation pour les signaux sur graphes est problématique. La définition la plus naturelle consiste à poser pour un signal  $\mathbf{f} \in \mathbb{R}^N$  sur un graphe d'ordre  $N$ ,

$$\mathbf{f}(\cdot - k) = \mathbf{f}(\text{mod}(\cdot - k, N)). \quad (5.2)$$

Cette définition est insatisfaisante puisqu'elle dépend de manière directe de l'ordre des sommets du graphe, alors que cet ordre n'a pas de signification intrinsèque. Ensuite, le même problème se pose pour définir les opérations de *dilatation*, de *modulation* et de *sous-échantillonnage* d'un signal sur graphe [SNF<sup>+</sup>12]. Enfin, et en lien avec l'opération de sous-échantillonnage, la construction de sous-graphes capturant des propriétés structurelles des graphes initiaux est problématique.

## 5.3 Transformées multi-échelles définies dans le domaine spectral

Dans toute cette section,  $G$  désigne un graphe pondéré de matrice d'adjacence pondérée  $\mathbf{W} \in \mathbb{R}^{N \times N}$ .

### 5.3.1 Ondelettes de diffusion

Cette construction a été proposée par Coifman et Maggioni dans [CM06] dans un cadre très large incluant à la fois des fonctions sur des graphes, des variétés, et des espaces métriques généraux. Nous détaillons dans cette section cette approche dans le cadre de fonctions définies sur des graphes pondérés.

Notons, comme dans l'équation (2.16),  $\mathbf{K}$  la matrice de transition associée à la marche aléatoire naturelle sur  $G$ . La construction de [CM06] interagit avec  $G$  via le calcul de puissances dyadiques d'une matrice reliée à  $\mathbf{K}$ . Plus précisément, notons

$$\mathbf{T} = \mathbf{I} - \tilde{\mathbf{L}} = \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} = \mathbf{D}^{1/2} \mathbf{K} \mathbf{D}^{-1/2}.$$



La matrice  $\mathbf{T}$  est symétrique et similaire à  $\mathbf{K}$ ; elle a donc le même spectre que  $\mathbf{K}$ . La symétrie de  $\mathbf{T}$  facilite la représentation de puissances de l'opérateur associé.

La construction de [CM06] repose sur le fait que les matrices  $\mathbf{T}^{2^j}$  ont des rangs de plus en plus faibles à mesure que  $j$  est grand; asymptotiquement, ce rang est égal à celui de la matrice dont les lignes sont toutes égales à la distribution stationnaire de la chaîne de Markov de matrice de transition  $\mathbf{K}$ . Donc, asymptotiquement, la suite de matrices  $\mathbf{T}^{2^j}$  est de rang 1. La décroissance du rang des  $\mathbf{T}^{2^j}$  est exploitée pour construire une analyse multirésolution de  $\mathbb{R}^N$ . Une suite emboîtée de sous-espaces de  $\mathbb{R}^N$

$$\mathbb{R}^N = \mathcal{V}_0 \supset \mathcal{V}_1 \supset \dots \mathcal{V}_J, \quad (5.3)$$

est construite de la manière suivante. Notons  $\Phi_0 = (\delta_k)_{k \in X_0}$  la base canonique de  $\mathbb{R}^N$ , où  $X_0 = \{1, \dots, N\}$  désigne l'ensemble des indices. À ce stade, la matrice  $\mathbf{T}$  est interprétée comme celle représentant un opérateur de diffusion, que l'on notera  $T$ , dans  $\Phi_0$ . Les colonnes de  $\mathbf{T}$  engendrent l'image de l'opérateur  $T$ . Ces colonnes sont utilisées pour construire une famille orthonormée  $\Phi_1 = (\varphi_{1,k})_{k \in X_1}$  telle que tout élément de  $\text{Im}(T)$  soit approchable à  $\epsilon$  près par un vecteur dans  $\text{vect}(\Phi_1)$ , où  $\epsilon$  est un paramètre de précision, de l'ordre de  $10^{-4}$  dans [CM06]. À cette fin, les auteurs utilisent un algorithme d'orthonormalisation similaire au procédé de Gram-Schmidt et décrit dans la section 5 de [CM06]. L'espace  $\mathcal{V}_1$  correspond alors à  $\text{vect}(\Phi_1)$ .

Le même procédé est appliqué à l'opérateur  $T^2$ . Plus précisément, la restriction de  $T^2$  à  $\mathcal{V}_1$  est représentée dans la base  $\Phi_1$ . Notons  $\mathbf{T}_1^2$  la matrice correspondante. Les colonnes de cette matrice sont utilisées pour construire une famille orthonormée  $\Phi_2 = (\varphi_{2,k})_{k \in X_2}$  telle que tout élément de  $\text{Im}(T^2|_{\mathcal{V}_1})$  soit approchable à  $\epsilon$  près par un vecteur dans  $\text{vect}(\Phi_2)$ . À cette fin, le même algorithme d'orthonormalisation est employé. Remarquons à ce stade que  $\text{vect}(\Phi_2)$  est un sous-espace de  $\mathbb{R}^{|X_1|}$  et pas nécessairement de  $\mathbb{R}^N$ . Afin d'obtenir un sous-espace de  $\mathbb{R}^N$ , les auteurs de [CM06] proposent d'étendre les vecteurs de la famille  $\Phi_2$  à  $\mathbb{R}^N$ . En notant  $\tilde{\Phi}_2$  la famille de vecteurs ainsi étendus à  $\mathbb{R}^N$ , l'espace  $\mathcal{V}_2$  est alors défini comme étant égal à  $\text{vect}(\tilde{\Phi}_2)$ .

Plus généralement, pour la famille  $(T^{2^j})_{j=0, \dots, J}$ , le procédé précédent est itéré et donne lieu à une analyse multirésolution de  $\mathbb{R}^N$  sous la forme (5.3). Les espaces  $\mathcal{V}_j$  forment des espaces d'approximations engendrés par les fonctions d'échelle

$$(\varphi_{l,k})_{\substack{k \in X_l \\ l=1, \dots, J}}$$

Les espaces de détails sont obtenus, comme dans le cadre de l'analyse multirésolution classique de Mallat et Meyer [Mal89], en considérant les

supplémentaires orthogonaux des espaces d'échelle, et sont engendrés par les "ondelettes"

$$(\psi_{l,k})_{\substack{k \in X_l \\ l=1, \dots, J}}$$

En conclusion, les espaces d'échelle sont obtenus en examinant les puissances dyadiques d'un opérateur relié à la matrice de transition de la marche aléatoire naturelle sur le graphe pondéré sous-jacent.

### 5.3.2 Ondelettes spectrales

Cette approche a été introduite par Hammond et al. dans [HVG11] et peut être vue comme une adaptation de la transformée en ondelettes continue à des signaux sur graphes. Cette approche permet de construire pour tout paramètre d'échelle  $t > 0$  une ondelette  $\psi_{t,n} \in \mathbb{R}^N$  où  $n$  désigne l'index d'un sommet du graphe. Chaque  $\psi_{t,n}$  est de moyenne nulle et centrée autour du sommet de l'index  $n$ . De plus, les supports des  $\psi_{t,n}$  décroissent à mesure que  $t$  décroît.

La construction proposée dans [HVG11] interagit avec le graphe sous-jacent via l'opérateur laplacien  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  associé. Les valeurs et vecteurs propres de  $\mathbf{L}$  permettent de définir un équivalent de la transformée de Fourier pour les signaux sur graphes. Plus précisément, notons, comme dans le chapitre 2,  $0 = \lambda_0 \leq \dots \leq \lambda_{N-1}$  les valeurs propres de  $\mathbf{L}$  et  $(\mathbf{e}_i)_i$  une base orthonormée de  $\mathbb{R}^N$  formée par des vecteurs propres de  $\mathbf{L}$ . La transformée de Fourier associée à un signal  $\mathbf{f} \in \mathbb{R}^N$  défini sur les sommets de  $G$  le vecteur de ses composantes dans la base  $(\mathbf{e}_i)_i$  :

$$\hat{\mathbf{f}}(l) = \langle \mathbf{e}_l, \mathbf{f} \rangle = \sum_i \mathbf{f}(i) \mathbf{e}_l(i). \quad (5.4)$$

La transformée inverse correspond au développement de  $\mathbf{f}$  dans la base  $(\mathbf{e}_i)_i$  :

$$\mathbf{f} = \sum_l \hat{\mathbf{f}}(l) \mathbf{e}_l.$$

Le point d'entrée de la construction de [HVG11] est la remarque que le calcul de la transformée en ondelette (5.1) peut s'effectuer sans avoir à dilater  $f$  directement. Plus précisément,

$$\hat{W}_f(t, \omega) = \hat{\psi}^*(t\omega) \hat{f}(\omega). \quad (5.5)$$

La transformée proposée dans [HVG11] est construite précisément par analogie avec l'équation (5.5) et utilise les vecteurs propres de  $\mathbf{L}$  comme base de Fourier. Pour une fonction  $g : \mathbb{R} \rightarrow \mathbb{R}$  fixée, cette transformée associe à tout

signal  $\mathbf{f} \in \mathbb{R}^N$  sur  $G$  la famille de signaux définie pour tout paramètre d'échelle  $t > 0$  et en tout sommet  $n$  par

$$W_f(t, n) = (T_g^t \mathbf{f})(n) = \sum_l g(t\lambda_l) \hat{\mathbf{f}}(l) \mathbf{e}_l(n). \quad (5.6)$$

Les ondelettes sont obtenues en appliquant la transformée précédente aux signaux de la base canonique :

$$\boldsymbol{\psi}_{t,n} = T_g^t \delta_n.$$

Ainsi, les coefficients de la transformée (5.6) sont obtenus par produits scalaires avec les ondelettes :

$$W_f(t, n) = \langle \boldsymbol{\psi}_{t,n}, \mathbf{f} \rangle. \quad (5.7)$$

La fonction  $g$  joue le rôle de noyau et agit comme un filtre passe-bande. Comme pour le cadre continu, la transformée est inversible moyennant une hypothèse d'admissibilité sur  $g$ . Cette hypothèse est donnée dans le lemme 5.1 de [HVG11] et l'inversion permet alors de synthétiser un signal sur graphe à partir de la connaissance de sa transformée en ondelettes pour toutes les échelle  $t > 0$ . En pratique, une transformée discrète est nécessaire et les auteurs proposent d'échantillonner le paramètre d'échelle de manière régulière en échelle logarithmique sur un intervalle fini. Le théorème 5.6 de [HVG11] montre que la transformée correspond alors au développement dans une trame dont les bornes sont données en fonction du noyau  $g$  et du spectre de  $\mathbf{L}$ . Dans [LVDV11], les auteurs proposent une construction donnant lieu à trame de Parseval [Chr08] (tight frame en anglais), c'est-à-dire une trame dont les bornes sont égales. Cette transformée y est appliquée à l'analyse de données issues de l'imagerie par résonance magnétique fonctionnelle.

D'un point de vue algorithmique, la transformée en ondelettes proposée requiert la diagonalisation de la matrice  $\mathbf{L}$ . Pour des graphes d'ordre élevé, ceci est irréalisable et les auteurs de [HVG11] proposent une approximation polynomiale du noyau  $g$  afin d'éviter de calculer le spectre de  $\mathbf{L}$ . Les détails sont fournis dans la section 6 de [HVG11].

Enfin, du fait de la redondance de la transformée présentée ici, la reconstruction d'un signal à partir de sa transformée n'est pas unique. Les auteurs proposent de reconstruire la solution de norme minimale obtenue en appliquant la pseudo-inverse de la matrice représentant la transformée.

Dans [HJV12], la transformée en ondelettes spectrale a été appliquée au débruitage d'images modélisées comme des signaux sur graphes. Les auteurs proposent en particulier la minimisation d'une fonctionnelle de type  $\ell_1$ - $\ell_2$ , où la norme  $\ell_1$  est appliquée aux coefficients de la transformée et permet de promouvoir la parcimonie de la représentation dans la nouvelle base.

## 5.4 Transformée de Haar sur une hiérarchie

Comme dans la section précédente,  $G$  désigne un graphe pondéré d'ordre  $N$ . Le but est de construire des transformées multi-échelle pour les signaux sur  $G$  correspondant à des développements de ces signaux dans des bases ou des trames multi-échelles.

Les constructions que nous avons rappelées dans la section précédente interagissent avec le graphe sous-jacent via la matrice de transition associée à la marche aléatoire naturelle sur  $G$  pour la construction de [CM06] et via l'opérateur laplacien pour l'approche de [HVG11]. Dans cette section, nous nous intéressons à la construction d'une famille d'ondelettes directement dans le domaine spatial via l'organisation du graphe sous la forme d'une hiérarchie de partitions. Nous commençons tout d'abord par montrer comment une analyse multirésolution (AMR) de l'espace des fonctions sur un graphe découle immédiatement de l'organisation de ce graphe sous la forme d'une hiérarchie de partitions.

### 5.4.1 Transformée de Haar sur une hiérarchie

L'approche que nous décrivons ici est celle proposée par Gavish, Nadler et Coifman dans [GNC10]. Cette approche repose sur la construction d'une famille  $(\mathcal{V}^{(l)})_{l=1,\dots,L}$  de partitions de  $V(G)$  vérifiant

- $\mathcal{V}^{(1)} = \{V(G)\}$
- $\mathcal{V}^{(l)}$  est *plus fine* que  $\mathcal{V}^{(l-1)}$ , c'est-à-dire que tout élément de  $\mathcal{V}^{(l)}$  est inclus dans un élément de  $\mathcal{V}^{(l-1)}$ .

On dira alors que la famille  $(\mathcal{V}^{(l)})_{l=1,\dots,L}$  est une hiérarchie de partitions ou simplement une hiérarchie. Notons pour tout  $l \in \{1, \dots, L\}$ ,  $n_l$  le cardinal de  $\mathcal{V}^{(l)}$  et posons  $\mathcal{V}^{(l)} = \{\mathcal{V}_1^{(l)}, \dots, \mathcal{V}_{n_l}^{(l)}\}$ .

Les auteurs de [GNC10] montrent qu'une telle hiérarchie induit une AMR de l'espace des fonctions sur  $V(G)$ . En effet, notons,  $\mathcal{X}^{(l)}$  le sous-espace de  $\mathcal{H}(V) = \{f : V(G) \rightarrow \mathbb{R}\}$  formé par les signaux constants sur chaque partie  $\mathcal{V}_i^{(l)}$ ,  $i \in \{1, \dots, n_l\}$ . Nous avons alors

$$\text{vect}(\mathbf{1}) = \mathcal{X}^{(1)} \subset \mathcal{X}^{(2)} \subset \dots \subset \mathcal{X}^{(L)} = \mathcal{H}(V). \quad (5.8)$$

Ainsi, l'hypothèse d'emboîtement des partitions donne lieu à une famille croissante (au sens de l'inclusion) de sous-espaces de  $\mathcal{H}(V)$  pouvant être interprétée comme une AMR de cet espace.

Comme dans l'AMR classique [Mal89], une base d'ondelettes peut être construite en considérant les supplémentaires orthogonaux de chaque espace

d'échelle  $\mathcal{X}^{(l)}$ . En effet, notons  $\mathcal{W}^{(l)}$  le supplémentaire orthogonal de  $\mathcal{X}^{(l)}$  dans  $\mathcal{X}^{(l+1)}$  :

$$\mathcal{X}^{(l+1)} = \mathcal{X}^{(l)} \oplus \mathcal{W}^{(l)}. \quad (5.9)$$

Nous avons alors

$$\mathcal{H}(V) = \mathcal{W}^{(1)} \overset{\perp}{\oplus} \dots \overset{\perp}{\oplus} \mathcal{W}^{(L-1)} \overset{\perp}{\oplus} \mathcal{X}^{(1)}. \quad (5.10)$$

La représentation (5.10) de l'espace  $\mathcal{H}(V)$  permet de décomposer tout signal  $\mathbf{f} \in \mathcal{H}(V)$  sous la forme

$$\mathbf{f} = \mathbf{d}_1 + \dots + \mathbf{d}_{L-1} + \mathbf{s}_1, \quad (5.11)$$

où  $\mathbf{d}_i \in \mathcal{W}^{(i)}$ ,  $\mathbf{s}_i \in \mathcal{X}^{(i)}$  et  $\mathbf{s}_{i+1} = \mathbf{d}_i + \mathbf{s}_i$ .

Au vu de (5.10), on peut donc construire une base orthonormée de  $\mathcal{H}(V)$  en concaténant des bases orthonormées des espaces  $\mathcal{W}^{(l)}$ ,  $l \in \{1, \dots, L-1\}$  et de  $\mathcal{X}^{(1)}$ . Nous examinons plus en détail les signaux intervenant dans ces bases.

Afin de clarifier la nature des signaux de base, nous illustrons le passage d'un niveau  $l+1$  au niveau  $l$ . L'espace d'échelle de niveau  $l+1$ ,  $\mathcal{X}^{(l+1)}$ , est constitué des signaux constants sur les parties  $\mathcal{X}_i^{(l+1)}$ ,  $i = 1, \dots, n_{l+1}$ . Il est donc de dimension  $n_{l+1}$ . De même, l'espace d'échelle  $\mathcal{X}^{(l)}$  est de dimension  $n_l$ . L'espace de détail  $\mathcal{W}^{(l)}$  est alors de dimension  $n_{l+1} - n_l$ .

Chaque partie  $\mathcal{X}_i^{(l)}$  va donner lieu à une famille  $(\psi_{l,i,j})_j$  de fonctions de base toutes à support sur  $\mathcal{X}_i^{(l)}$ . De plus, ces fonctions de base doivent être :

- normées ;
- orthogonales aux fonctions d'échelle au niveau  $l$  ;
- orthogonales entre elles.

La première condition n'est pas restrictive puisque l'on peut normaliser les signaux de base à la fin de la procédure. La deuxième condition revient à imposer une moyenne nulle ; on parlera alors d'ondelettes. On perd alors un "degré de liberté" et l'on voit que si l'on note  $\text{sub}(l,i)$  le nombre de sous-parties de  $\mathcal{X}_i^l$  apparaissant au niveau  $l+1$ , on aura alors  $\text{sub}(l,i) - 1$  fonctions de base. Enfin, la dernière condition peut être vérifiée en appliquant un procédé d'orthogonalisation de Gram-Schmidt. La figure 5.1, tirée de [GNC10], illustre la construction des fonctions de base correspondant à une hiérarchie de partitions.

La transformée finale associe à tout signal  $\mathbf{f}$  sur  $G$  le vecteur dont les composantes sont données par les produits scalaires avec les signaux de base. Les auteurs de [GNC10] montrent que la décroissance des coefficients d'ondelettes de la transformée précédente peut être reliée à une notion de régularité du signal initial. Cette notion de régularité est définie relativement

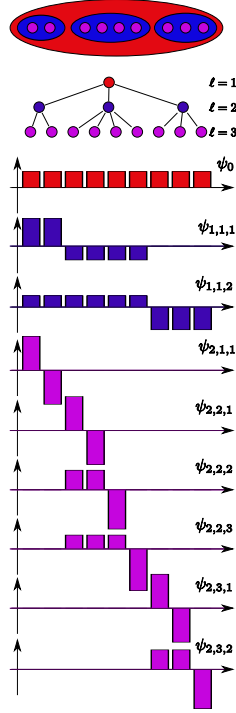


FIGURE 5.1 – Illustration des fonctions de base obtenues à partir d’une hiérarchie de partitions.

à l’arbre sous-jacent. Plus précisément, notons pour  $S \subset V(G)$ ,  $\nu(S) = |S|/|V(G)|$ . Les auteurs définissent la distance suivante sur  $V(G)$  :

$$(\forall x, y \in V(G)) \quad d(x, y) = \begin{cases} \nu(\text{partie}(x, y)), & \text{si } x \neq y \\ 0, & \text{sinon,} \end{cases} \quad (5.12)$$

où  $\text{partie}(x, y)$  est la plus petite partie contenant  $x$  et  $y$ .

Munis de la distance précédente, les auteurs définissent une notion de continuité höldérienne de la manière suivante. Soit  $C > 0$  et  $0 < \alpha < 1$ . On dira qu’un signal  $\mathbf{f}$  est  $(C, \alpha)$  Hölder continue si

$$(\forall i, j \in V(G)) \quad |\mathbf{f}_i - \mathbf{f}_j| \leq Cd(x, y)^\alpha. \quad (5.13)$$

Le théorème 1 de [GNC10] montre alors que<sup>1</sup>.

$$(\forall l, k, j) \quad |\langle f, \psi_{l,k,j} \rangle| \leq NC2^{\alpha+1}\nu(\mathcal{X}_k^l)^{\alpha+1/2}. \quad (5.14)$$

1. Le facteur  $N$  apparaissant dans le terme de droite est dû à la définition que nous adoptons ici pour le produit scalaire sur  $V(G)$  :  $\forall \mathbf{f}, \mathbf{g} \in V(G)$ ,  $\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{i=1}^N \mathbf{f}_i \mathbf{g}_i$ . Les auteurs de [GNC10] choisissent de travailler avec le produit scalaire suivant :  $\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{i=1}^N \mathbf{f}_i \mathbf{g}_i / N$

Dans un cadre où les signaux sont définis sur des ensembles finis, il n'est pas évident de décider si l'inégalité (5.14) peut avoir un impact pratique sur la construction de la transformée. En effet, la notion de régularité absolue présentée plus haut peut difficilement discriminer des classes de signaux : pour tout  $\alpha \in ]0, 1[$  et pour tout  $\mathbf{f} \in \mathbb{R}^N$ , il existe  $C > 0$  tel que  $\mathbf{f}$  soit  $(C, \alpha)$  Hölder continue.

Intuitivement, les valeurs absolues des coefficients de la transformée doivent être reliées à la variance du signal analysé sur chaque partie de la hiérarchie. Ce lien est explicitement formulé dans [GNC10] où l'inégalité suivante est démontrée<sup>2</sup> :

$$(\forall l, k, j) \quad |\langle f, \psi_{l,k,j} \rangle| \leq N\sigma(\mathbf{f}, \mathcal{X}_k^l), \quad (5.15)$$

où  $\sigma(\mathbf{f}, \mathcal{X}_k^l)$  désigne l'écart-type empirique de  $\mathbf{f}$  sur la partie  $\mathcal{X}_k^l$ .

Notons enfin que d'autres transformées multi-échelles basées sur l'organisation d'un ensemble de données sous la forme d'une hiérarchie de partitions ont également été proposées dans des travaux récents. En particulier, la transformée de [GNC10] peut être vue comme une généralisation de celle proposée dans [EA02] pour des arbres binaires complets. Dans [Mur07], Murtagh propose une transformée multi-échelle basée sur la représentation d'un ensemble de données sous la forme d'un dendrogramme (section 14.3.12 de [HTF01]). Cette transformée est appliquée à l'ensemble de données lui-même et non à des signaux définis sur cet ensemble de données. Une approche similaire à [GNC10] a été proposée dans [LNW08], et la transformée correspondante est appliquée à l'ensemble de données et non à des signaux sur l'ensemble de données.

Lorsque la transformée de [GNC10] est appliquée à un arbre binaire complet, elle correspond à une transformée de Haar 1D appliquée à une permutation des données. Dans [REC11], les auteurs proposent une généralisation de cette transformée. Une version redondante est introduite dans [REC12]. Les transformées de [REC11, REC12] sont appliquées au débruitage d'images : les coefficients d'ondelettes sont seuillés et la transformée inverse est appliquée. Une approche de type "cycle-spinning" [CD+95] est enfin appliquée. Notons enfin que l'approche de [REC11] est similaire à celle proposée récemment dans [Pl09].

## 5.4.2 Construction adaptative d'une hiérarchie à partir d'un graphe pondéré

Nous proposons dans cette sous-section un algorithme permettant la construction d'une hiérarchie de partitions à partir d'un graphe pondéré.

---

2. Voir note 1.

Cette construction, proposée dans [HLE11b, HLE11a] est adaptative, c'est-à-dire qu'elle dépend du signal que l'on souhaite analyser. Formellement, cette construction dépend donc d'un graphe pondéré et d'un signal sur ce graphe. Appliquée à la hiérarchie ainsi obtenue, la transformée de la sous-section précédente sera globalement non linéaire, bien qu'elle soit linéaire si l'on considère la hiérarchie fixée.

Au-delà de l'aspect adaptatif de la construction de la hiérarchie, il est également souhaitable que cette construction soit robuste au bruit qui peut s'ajouter aux observations dont on dispose. Nous détaillons dans ce qui suit les différentes étapes de notre algorithme.

## **Description de l'algorithme**

L'algorithme que nous introduisons ici produit, à partir d'un graphe pondéré  $G_0 = G$  et d'un signal  $\mathbf{f}_0 \in \mathbb{R}^N$  sur l'ensemble des sommets de  $G$ , une hiérarchie de graphes pondérés  $(G_i)_i$  et de signaux  $(\mathbf{f}_i)_i \in \mathbb{R}^{|V(G_i)|}$  définis sur ces graphes. En particulier, cet algorithme fournit un partitionnement hiérarchique des sommets du graphe initial  $G_0$ . C'est ce dernier aspect qui nous intéressera pour la suite de cette section.

L'algorithme procède par agrégations successives des sommets. L'agrégation des sommets d'un graphe donné est précédée par une étape de partitionnement de ses sommets. Dans notre cas, ce partitionnement est réalisé de manière adaptative en fonction du signal dont on dispose. De plus, il repose sur une étape de simplification préliminaire elle-même basée sur un procédé de régularisation sur graphes décrit dans le chapitre 3 de ce manuscrit.

Après chaque agrégation, une relation d'adjacence et un signal sont définis sur les agrégats. Le processus est itéré sur le nouveau graphe pondéré ainsi obtenu, de manière adaptative, relativement au nouveau signal construit. Nous détaillons dans la suite de cette sous-section les différentes étapes de l'algorithme.

## **Utilisation de l'a priori TV**

Du fait de l'hétérogénéité des valeurs de  $\mathbf{f}$ , nous proposons d'appliquer une étape de régularisation. Cette étape doit pouvoir, d'une part, révéler les possibles groupements des sommets de  $G$  en réduisant les variations s'apparentant à du bruit, et d'autre part, préserver les discontinuités importantes du signal en question. Nous avons opté pour une approche basée sur la minimisation du



modèle ROF sur graphes défini dans la chapitre 3. Rappelons ici qu'il s'agit de considérer le problème suivant :

$$\underset{u \in \mathbb{R}^N}{\text{Minimiser}} \quad E_{1,2}^w(\mathbf{u}; \mathbf{f}, \lambda), \quad (5.16)$$

où

$$E_{1,2}^w(\mathbf{u}; \mathbf{f}, \lambda) = \lambda J_{1,2}^w(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathcal{X}}^2, \quad (5.17)$$

et  $\lambda > 0$ .

Notons  $\mathbf{f}^*$  l'unique solution du problème (5.16). Le partitionnement des sommets de  $G$  est alors obtenu en considérant les classes d'équivalence d'une relation d'équivalence  $\mathcal{R}_{\mathbf{f}^*}^\epsilon$  que nous décrivons maintenant.

### Partitionnement adaptatif des sommets d'un graphe

On propose ici de partitionner les sommets de  $G$  en prenant en compte à la fois la topologie du graphe et le signal  $\mathbf{f}^*$  obtenu après l'étape de régularisation.

Intuitivement, deux sommets voisins appartiennent à un même sommet si les valeurs prises par  $\mathbf{f}$  en ces deux sommets sont proches. Fixons un réel positif  $\epsilon$ . Notons, pour deux sommets  $i$  et  $j$  de  $G$ ,  $d_f(i, j) = |\mathbf{f}_i - \mathbf{f}_j|$ . Nous considérons la relation binaire  $\mathcal{R}_f^\epsilon$  définie de la manière suivante :  $i \mathcal{R}_f^\epsilon j$  si et seulement si  $i = j$  où s'il existe un chemin  $(l_1, \dots, l_n)$  reliant  $i$  et  $j$  tel que pour tout  $k \in \llbracket 1, n-1 \rrbracket$   $d_f(l_k, l_{k+1}) < \epsilon$ . Cette relation binaire est réflexive, symétrique ( $G$  est non orienté), et transitive. Il s'agit donc d'une relation d'équivalence. Les classes d'équivalence correspondantes forment ainsi une partition de l'ensemble des sommets de  $G$ .

La partition précédente peut être vue comme une croissance de région, et le paramètre  $\epsilon$  permet de contrôler la finesse de cette partition.

### Construction d'un graphe pondéré et d'un signal sur ce graphe par agrégation

Notons  $P^\epsilon(G_0) = \{P_{0,1}, \dots, P_{0,n_0}\}$  les classes d'équivalence distinctes obtenues à l'étape précédente. Nous noterons  $P_{0,i} \sim P_{0,j}$  si  $P_{0,i}$  et  $P_{0,j}$  ont des représentants adjacents dans le graphe  $G_0$ .

On construit un nouveau graphe  $G_1$  dont les sommets correspondent aux classes d'équivalence  $P_{0,i}$ . De manière équivalente, les sommets de  $G_0$  appartenant à une même partie  $P_{0,i}$  sont agrégés pour former un unique sommet de  $G_1$ . On obtient ainsi un ensemble de sommets dont le cardinal est égal aux nombres de classes d'équivalence obtenues précédemment. Si  $u \in V(G_1)$ , on notera  $R_u^0$  l'ensemble des sommets de  $G_0$  qui ont été agrégés pour former  $u$ .

Deux sommets  $u, v \in V(G_1)$  sont connectés par une arête s'ils représentent deux classes d'équivalence ayant des représentants adjacents dans  $G_0$ . Afin de tenir compte des volumes des parties agrégées, nous proposons de pondérer les arêtes du graphe  $G_1$  de la manière suivante. Pour tout couple de sommets adjacents  $u, v \in V(G_1)$ ,

$$w_1(u, v) = \frac{\text{cut}(R_u^0, R_v^0)}{|R_u^0|} + \frac{\text{cut}(R_u^0, R_v^0)}{|R_v^0|}.$$

où

$$\text{cut}(A, B) = \sum_{a \in A, b \in B} w(a, b),$$

et  $A$  et  $B$  désignent deux parties d'un graphe pondéré par la fonction  $w$ .

Nous disposons à ce stade d'un graphe pondéré  $G_1 = (V(G_1), E(G_1), w_1)$ . Nous définissons enfin un signal  $\mathbf{f}_1$  sur  $G_1$  en considérant le moyennes de  $\mathbf{f}_0$  sur chaque classe d'équivalence  $P_{0,\cdot}$  :

$$(\forall u \in V(G_1)) \quad \mathbf{f}_1(u) = \frac{1}{|R_u^0|} \sum_{v \in R_u^0} \mathbf{f}_0(v).$$

### Itération du procédé et terminaison de l'algorithme

Nous venons de montrer comment un graphe pondéré d'agrégats peut être construit de manière adaptative à partir d'un graphe initial. Ce procédé est alors itéré sur la paire  $(G_1, \mathbf{f}_1)$  : une étape de régularisation/simplification est effectuée, suivie d'un partitionnement des nouveaux sommets et enfin de la construction du graphe des agrégats et du nouveau signal. Afin de garantir un nombre croissant d'agrégats au cours de la hiérarchie, nous modifions le paramètre de régularisation  $\lambda_i$  à chaque étape en le divisant par deux :  $\lambda_{i+1} = \lambda_i/2$ .

Enfin, l'algorithme se termine lorsqu'on arrive à un graphe d'agrégats à un seul sommet. Nous illustrons dans ce qui suit la transformée de Haar sur la hiérarchie dont nous venons de décrire la construction.

### 5.4.3 Expériences

Afin de montrer le comportement de la décomposition multi-échelle que nous obtenons, nous présentons plusieurs exemples de son utilisation sur des images et bases de données. Partant d'un signal initial  $\mathbf{f}$ , nous présentons les différents niveaux de décomposition obtenus, à savoir les fonctions de détails  $\mathbf{d}_{L-i}$  et d'échelle  $\mathbf{s}_{L-i}$ , avec  $\mathbf{s}_{L-i} = \mathbf{d}_{L-i-1} + \mathbf{s}_{L-i-1}$ .

## Images

Nous considérons tout d'abord le cas des images. La figure 5.2 présente la décomposition multi-échelle que nous obtenons sur les 6 premiers niveaux ( $i \in [0, 6]$ ) d'une hiérarchie adaptative construite à partir d'un graphe en 8-voisinage pondéré avec des poids gaussiens. La figure 5.3 présente la décomposition multi-échelle obtenue sur la même image mais en utilisant un graphe des 7 plus proches voisins pris dans un voisinage  $11 \times 11$  avec des poids gaussiens sur des patchs  $5 \times 5$ . Comme cela est attendu, quel que soit le graphe considéré, chaque niveau de détail permet de capturer des informations de plus en plus fines et localisées au fur et à mesure de la décomposition multi-échelle. On constate également l'importance d'utiliser une représentation adaptée des pixels. En effet, l'image originale de "Barbara" contenant beaucoup de zones de textures, la décomposition obtenue est bien plus pertinente lorsque l'on utilise un graphe de voisinage exploitant des patchs (à la fois dans sa construction, et dans sa pondération). Ceci se constate à la fois dans les fonctions de détails extraits qui concernent des zones de texture proches, mais également dans les fonctions d'échelle dans lesquelles les frontières sont mieux délimitées. Cet effet est également mis en avant sur la décomposition obtenue dans la figure 5.4.

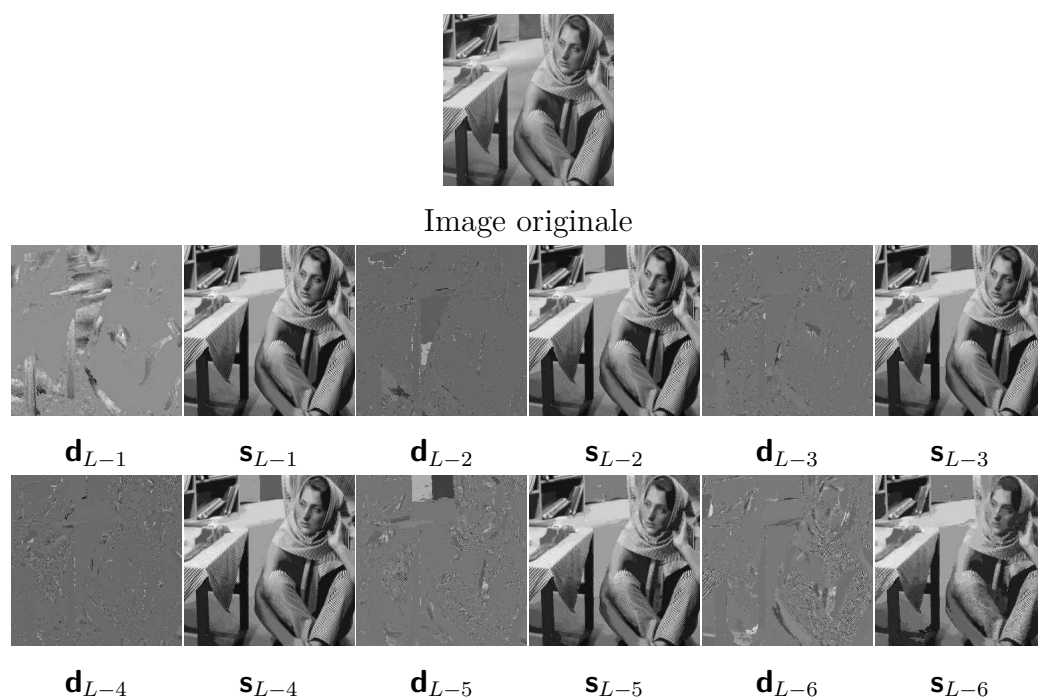


FIGURE 5.2 – Décomposition multi-échelle de "Barbara" : graphe en 8-voisinage, poids gaussiens.

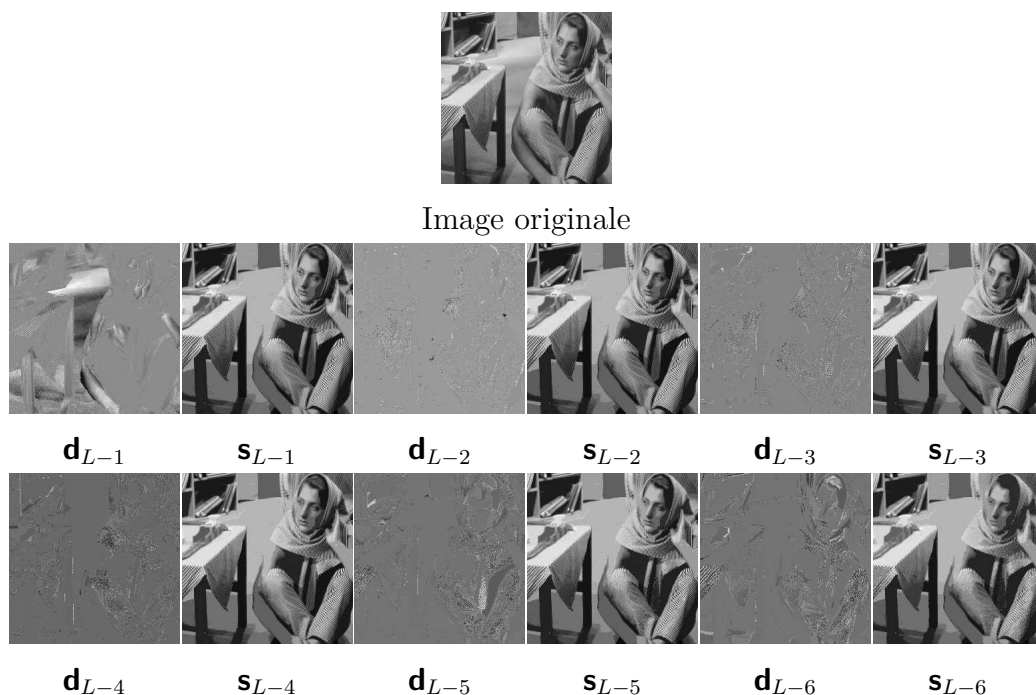


FIGURE 5.3 – Décomposition multi-échelle de "Barbara" : graphe des 7-plus proches voisins, poids gaussiens sur patches  $5 \times 5$  dans un voisinage  $11 \times 11$ .

### Bases d'images

Afin d'illustrer la généralité de l'approche, considérons à présent des données beaucoup moins communes et difficiles à traiter dans leur ensemble : des bases de données d'images. Nous considérons deux bases d'images. La première est la base USPS (United States Postal Service) contenant des images de chiffres manuscrits scannés d'enveloppes de l'U.S. Postal Service. Les images sont de taille  $16 \times 16$ . La fonction associée à un noeud est donc un vecteur de  $\mathbb{R}^{16^2}$ . Nous considérons 400 images de cette base (40 par chiffre manuscrit). La seconde base est la Olivetti Face Image Database qui contient 400 images de visages (40 personnes avec 10 images par personne). Les images de chaque personne présentent des variations en temps d'acquisition, illumination et expression. Les images sont de taille  $92 \times 112$ . La fonction associée à un noeud est donc un vecteur de  $\mathbb{R}^{92 \times 112}$ .

Pour ces deux bases, la construction d'une hiérarchie adaptative est effectuée à partir d'un graphe initial des 7 plus proches voisins avec une fonction de poids entre deux images  $u$  et  $v$  définie par  $w(u, v) = \frac{1}{1 + \|\mathbf{f}(u) - \mathbf{f}(v)\|}$ . Nous présentons tout d'abord les 5 premiers niveaux de décimation obtenus, c'est-à-

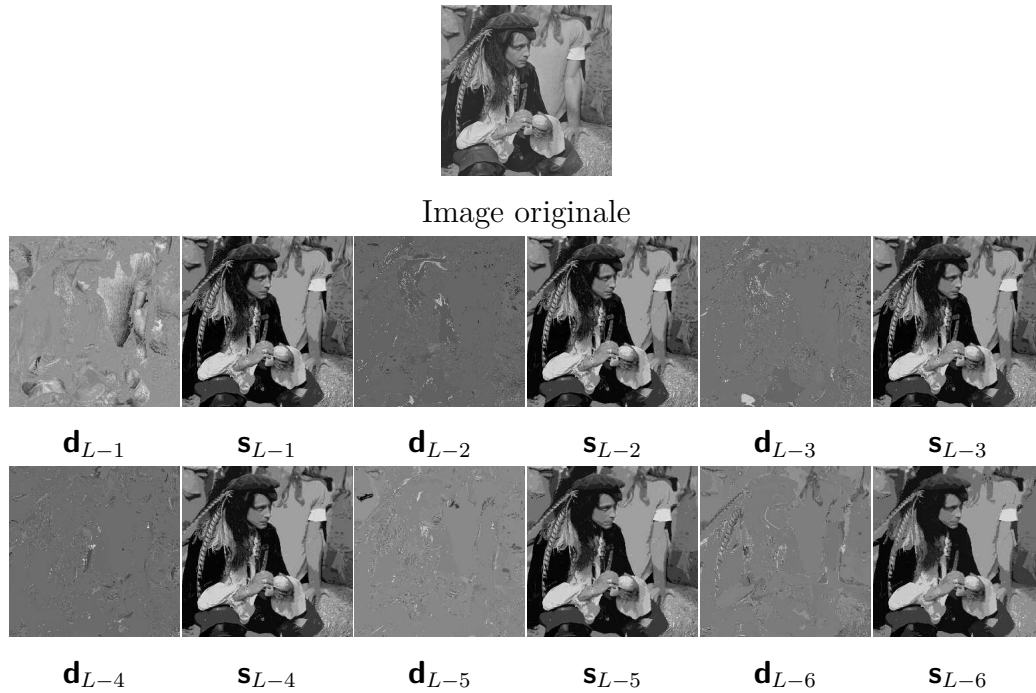


FIGURE 5.4 – Décomposition multi-échelle de "Man" : graphe des 7-plus proches voisins, poids gaussiens sur patches  $5 \times 5$  dans un voisinage  $11 \times 11$ .

dire les graphes après agrégation de noeuds. Les figures 5.5 et 5.6 présentent les résultats obtenus pour les deux bases d'images. Chaque graphe  $(G_i, \mathbf{f}_i)$  présenté après le premier niveau est donc une version décimée du graphe  $G_{i-1}$  avec une fonction  $\mathbf{f}_{i-1}$  qui est obtenue comme la moyenne des images du cluster correspondant à une classe d'équivalence. Ceci permet d'illustrer comment l'agrégation des graphes opère au fur et à mesure de la construction de la hiérarchie. On constate bien à la fois une décimation du graphe et une modification de la fonction associée aux noeuds survivants.

Les figures 5.7 et 5.8 présentent les fonctions d'échelle et de détail obtenues pour les niveaux de décomposition  $L - 3$  et  $L - 5$  sur les bases USPS et Olivetti. Il est cette fois ci beaucoup plus difficile d'évaluer visuellement la pertinence de la décomposition pour de telles masses de données, d'autant que la distance considérée pour comparer deux images est très simple. Néanmoins, les expériences conduites mettent en avant l'intérêt de notre approche formulée sur des graphes : une fois que la topologie et les poids du graphe initial décrivant les données sont fixés, l'algorithme est identique.

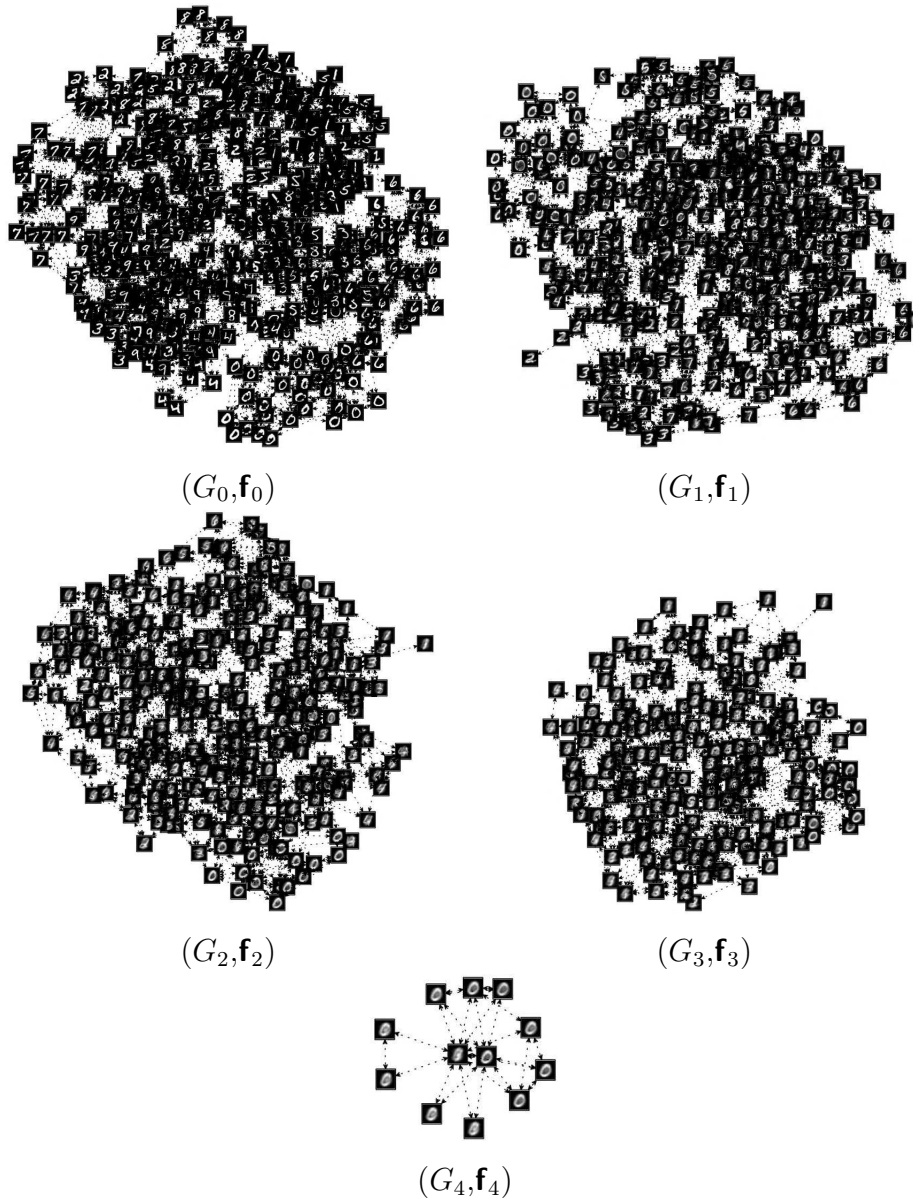


FIGURE 5.5 – Graphes des niveaux de décomposition des 400 images de la base USPS.

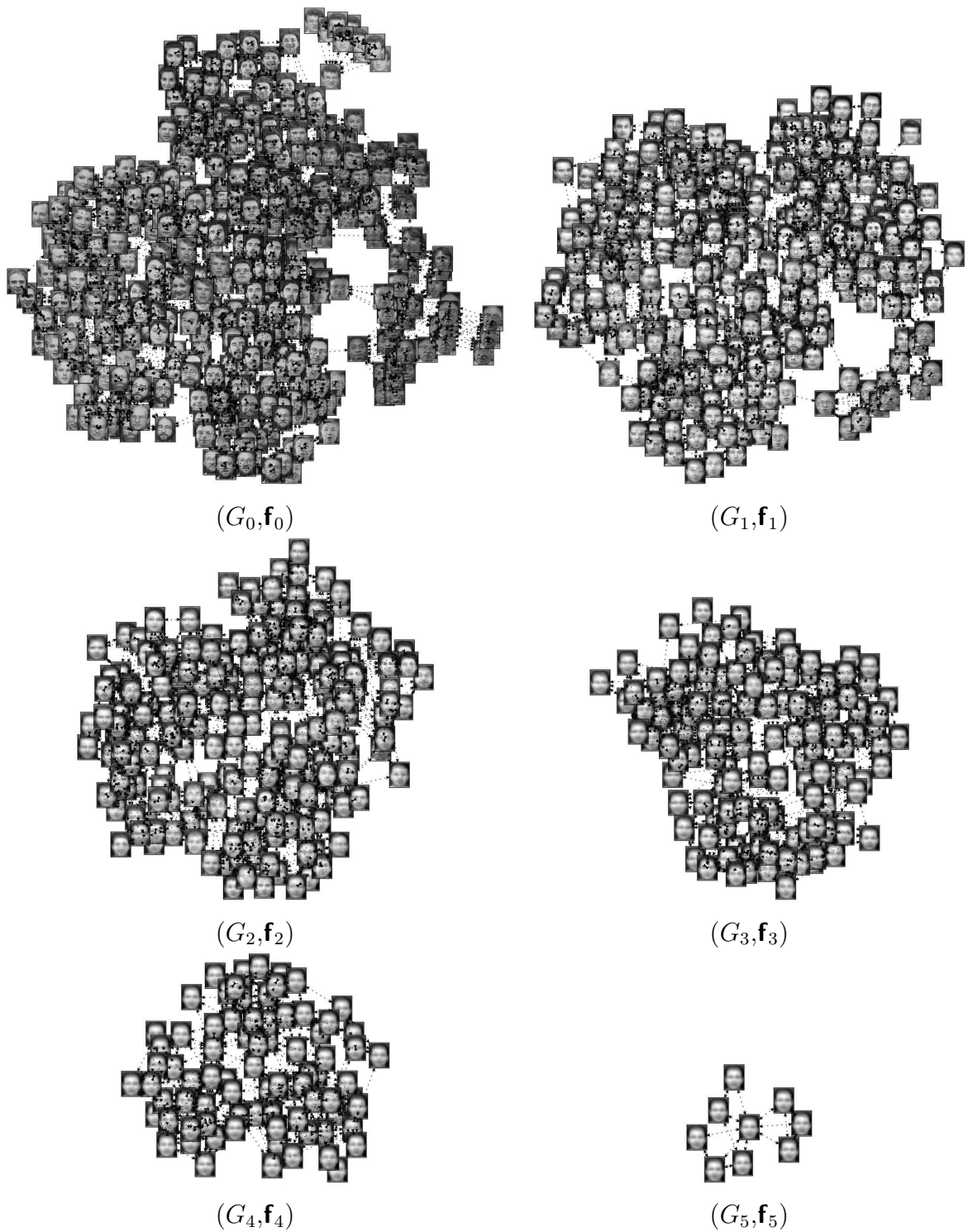


FIGURE 5.6 – Graphes des niveaux de décomposition des 400 images de la base Olivetti.



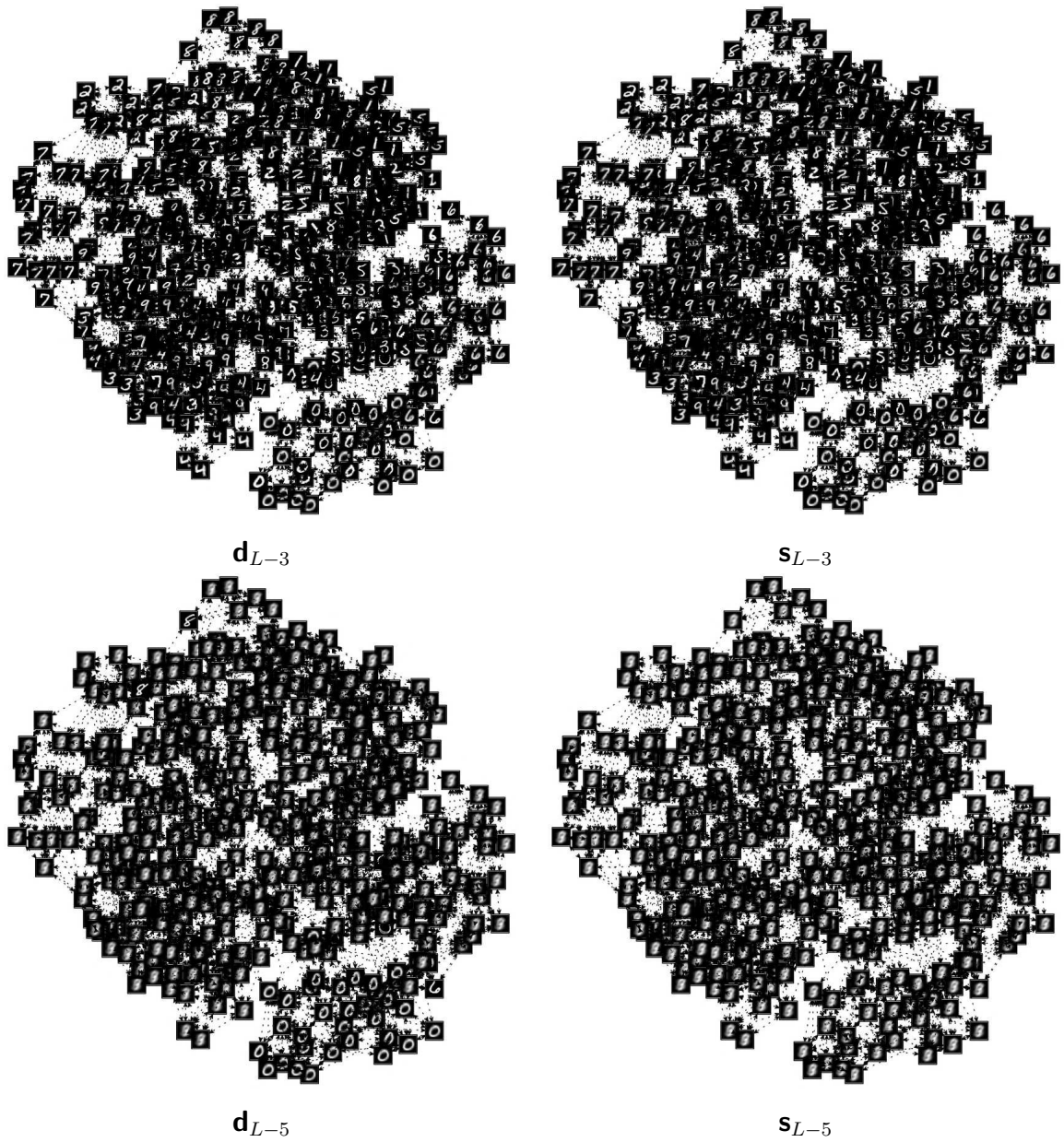


FIGURE 5.7 – Décomposition de 400 images de la base USPS : fonctions de détail et d'échelle des niveaux 3 et 5.



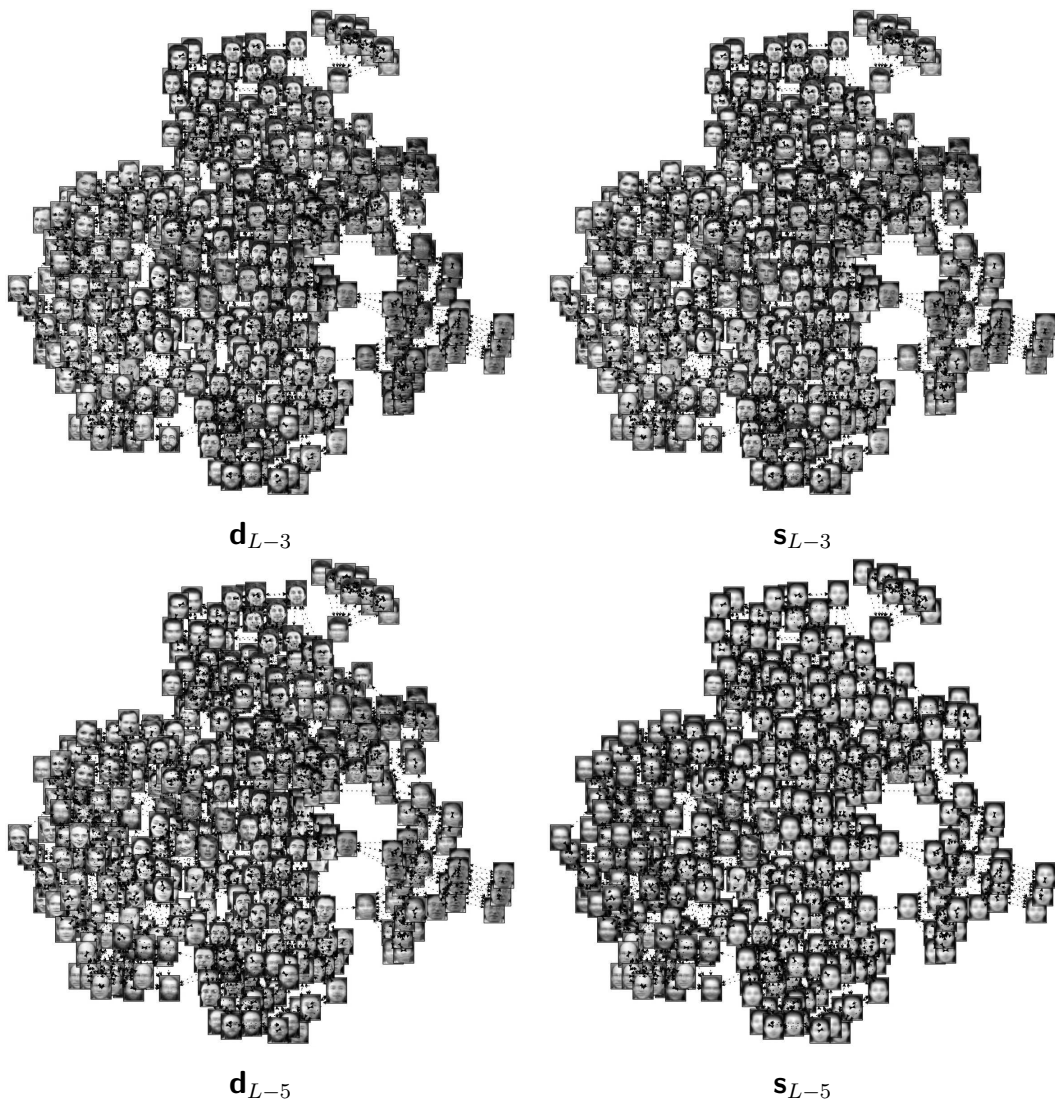


FIGURE 5.8 – Décomposition de 400 images de la base Olivetti : fonctions de détail et d'échelle des niveaux 3 et 5.

## 5.5 Transformées multi-échelles par lifting sur graphes

Nous avons vu dans la section précédente que l'organisation des sommets d'un graphe  $G$  sous la forme d'une hiérarchie de partitions induisait une analyse multirésolution naturelle de l'espace  $\mathcal{H}(V)$  des fonctions sur  $V(G)$ . Nous avons également proposé un algorithme permettant la construction adaptative d'une telle hiérarchie. Nous nous intéressons dans cette section à une approche alternative, visant à construire des transformées multi-échelles en utilisant le schéma du lifting introduit par Sweldens dans [Swe96, Swe98].

### 5.5.1 Schéma du lifting pour les signaux 1D

Le schéma du 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. De plus, d'un point de vue pratique, les transformées directes et inverses peuvent être implémentées *in situ*, c'est-à-dire sans occuper de mémoire autre que celle allouée au signal initial. La deuxième application du schéma du 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 [SS95].

Illustrons rapidement le principe pour un signal 1D  $\mathbf{x} = (x_1, \dots, x_N)$ . Le lecteur pourra se référer à la figure 5.9, qui illustre la transformée du lifting directe, et à la figure 5.10, qui illustre la transformée inverse.

La première étape du schéma du 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  $\mathbf{x}^I$  :

$$\tilde{\mathbf{x}}^I = P(\mathbf{x}^J), \quad (5.18)$$

puis on évalue l'erreur de prédiction  $(\mathbf{x}^I)^*$  :

$$(\mathbf{x}^I)^* = \mathbf{x}^I - \tilde{\mathbf{x}}^I. \quad (5.19)$$

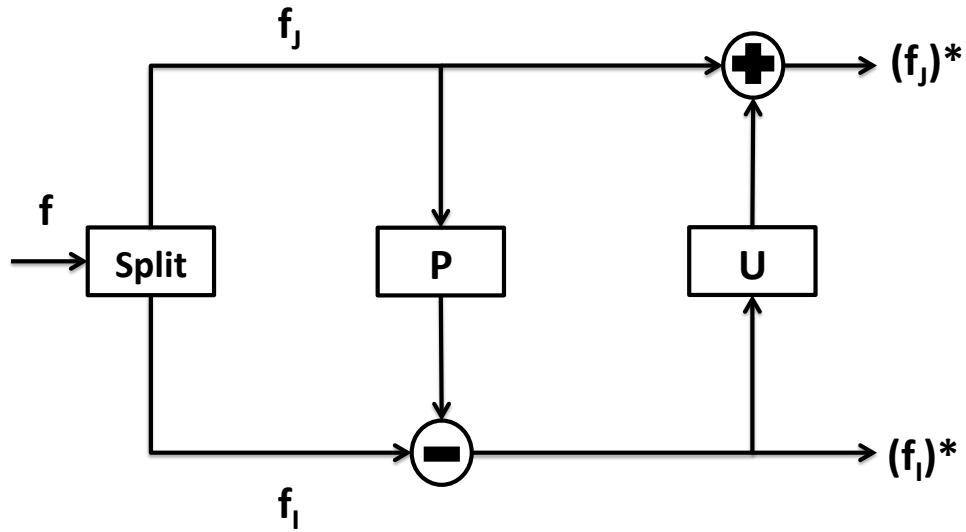


FIGURE 5.9 – Transformée du lifting directe.

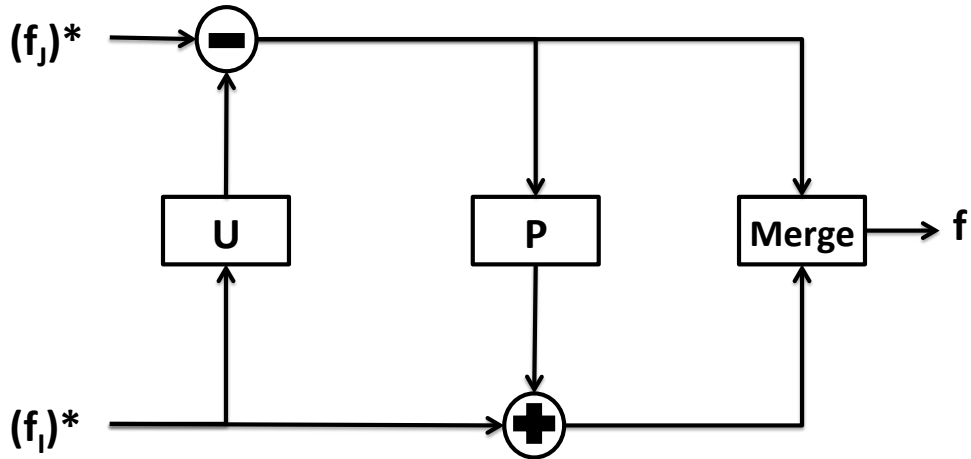


FIGURE 5.10 – Transformée du lifting inverse.

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

$$(\mathbf{x}^J)^* = \mathbf{x}^J + U((\mathbf{x}^I)^*). \quad (5.20)$$

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 lifting 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 :

$$\mathbf{x}^J = (\mathbf{x}^J)^* - U \left( (\mathbf{x}^I)^* \right), \quad (5.21)$$

et

$$\mathbf{x}^I = (\mathbf{x}^I)^* + \tilde{\mathbf{x}}^I. \quad (5.22)$$

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 sous-section suivante.

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 :

$$\tilde{x}_{2m-1} = x_{2m}, \quad x_{2m-1}^* = x_{2m-1} - x_{2m}, \quad (5.23)$$

et, d'autre part, la mise à jour suivante

$$x_{2m}^* = x_{2m} + \frac{1}{2}x_{2m-1}^*, \quad (5.24)$$

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 [Mal09].

Notons que le schéma du 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 [CDSB03, PH02, KPPBB<sup>+</sup>12]. Notons cependant que ces transformées multi-échelles non linéaires ne correspondent généralement plus à des changements de base.

Notons également que le schéma du 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 [DSvS04] pour la régression non paramétrique avec design aléatoire. Des techniques similaires ont également été proposées dans [WSB05, WCBD05, SO08, BCW08] 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 du lifting à des signaux définis sur graphes.

### 5.5.2 Lifting sur graphes

L'adaptation du schéma du lifting à des signaux définis sur des graphes généraux a été proposée en particulier dans [She10, NO09]. Bien que cette adaptation soit immédiate, son application pose certains problèmes. Nous commençons par spécifier le schéma du lifting dans le contexte des signaux sur graphes. On se limite dans toute la suite à des opérateurs linéaires pour l'étape de prédiction. De plus, les opérateurs  $P$  et  $U$  sont *locaux* : la prédiction et la mise à jour d'un signal en un sommet  $k \in V(G)$  ne dépend que de ses voisins  $l \sim k$ .

#### Partitionnement des sommets

Soit  $G$  un graphe pondéré sur  $V(G) = \{1, \dots, N\}$  et  $\mathbf{W}$  sa matrice d'adjacence pondérée. Afin de pouvoir effectuer une étape de lifting, il est nécessaire de partitionner  $V(G)$  en deux parties. Comme nous le verrons dans la suite, le critère de partitionnement dépend, d'une part, du domaine d'application et, d'autre part, des propriétés de la transformée que l'on souhaite obtenir. Nous fixons pour l'instant deux parties  $I$  et  $J$  formant une partition de  $V(G)$ .

#### Prédiction

Rappelons que les graphes sur lesquels sont définis les signaux que nous étudions sont des *graphes de similarité*. Ainsi, nous utilisons les poids des arêtes pour définir l'étape de prédiction.

L'étape de prédiction du schéma du lifting peut s'écrire à l'aide d'une matrice de prédiction  $\mathbf{P} \in \mathbb{R}^{N \times N}$  vérifiant

$$(i) \quad (\forall j \in J) (\forall l \in V(G)) \quad p_{j,l} = \delta_{j,l} = \begin{cases} 1, & \text{si } j = l, \\ 0, & \text{sinon,} \end{cases} \quad (5.25)$$

(ii)

$$(\forall i \in I) (\forall l \in V(G)) \quad p_{i,l} = \begin{cases} \frac{w_{i,l}}{\sum_{\substack{j \in J \\ j \sim i}} w_{i,j}}, & \text{si } l \in J \text{ et } l \sim i \\ 0, & \text{sinon.} \end{cases} \quad (5.26)$$

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

### Mise à jour

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

$$(i) \quad (\forall i \in I) (\forall l \in V(G)) \quad u_{i,l} = \delta_{i,l}, \quad (5.27)$$

$$(ii) \quad (\forall j \in J) (\forall l \in V(G)) \quad u_{j,l} \neq 0 \implies l \sim j \text{ et } l \in I. \quad (5.28)$$

### Transformée globale

Une étape de la transformée globale est représentée par la matrice

$$\mathbf{T} = (\mathbf{I} + \mathbf{U}) (\mathbf{I} - \mathbf{P}). \quad (5.29)$$

Posons  $\mathbf{g} = \mathbf{Tf}$ . Les composantes  $(\mathbf{g}_i)_{i \in I}$  correspondent à des *coefficients de détail* ou d'*ondelettes* alors que les composantes  $(\mathbf{g}_j)_{j \in J}$  correspondent à des *coefficients d'échelle*.

Plus généralement, plusieurs étapes de prédiction/mise à jour peuvent être effectuées :

$$\mathbf{T} = \prod_{m=1}^k (\mathbf{I} + \mathbf{U}_m) (\mathbf{I} - \mathbf{P}_m). \quad (5.30)$$

La transformée précédente peut être itérée sur le signal  $(\mathbf{g}_j)_{j \in J}$  à condition de définir un nouveau graphe sur les sommets dans  $J$ .

### Remarques concernant le partitionnement

Nous identifions à ce stade une première difficulté concernant l'application du schéma du 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 du 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, comme nous l'avons vu au chapitre 2, est équivalent au problème de la coupe maximale. Ce dernier problème est NP-dur. Dans [NO09], les auteurs utilisent l'algorithme randomisé proposé dans [FM01] pour calculer une solution approchée. Les auteurs de [MEO11, MEDdMO11] utilisent l'algorithme proposé dans [Hsu83]. Les travaux [MEO11, MEDdMO11] 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 [JNS09], 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.

### 5.5.3 Transformée multi-échelle par suite de liftings élémentaires

Afin de pallier le problème du partitionnement des sommets du graphe, nous adoptons ici une stratégie qui consiste à calculer les coefficients un à un. On parlera alors d'étapes élémentaires de lifting. Cette approche est inspirée par [JNS09] et a été initialement proposée dans [JNS01] pour des signaux 2D échantillonnés de manière irrégulière.

#### Description de l'algorithme

L'approche proposée dans [JNS09] est résumée<sup>1</sup> dans l'algorithme 5.1.

Initialement, les coefficients d'échelle correspondent au signal  $\mathbf{f}$ . Chaque itération de la boucle 3 de l'algorithme 5.1 correspond à une unique étape de

---

1. Nous adoptons une numérotation ascendante, contrairement à celle utilisée dans [JNS09].

- 1: Entrée : Un graphe pondéré  $G$  sur  $\llbracket 1, N \rrbracket$  et un signal  $\mathbf{f} \in \mathcal{H}(V)$ .
- 2: Initialisation :  $\mathcal{S}_0 := V(G)$ ,  $\mathcal{D}_0 := \emptyset$ ,  $c_{0,i} := f_i$ ,  $\forall i \in \mathcal{S}_0$ .
- 3: **for all**  $r \in \llbracket 1, m \rrbracket$  **do**
- 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 :

$$d_r := c_{r-1,i_r} - \sum_{\substack{k \sim i_r \\ k \in \mathcal{S}_r}} \alpha_{r,k} c_{r-1,k}. \quad (5.31)$$

- 8: Mise à jour :

$$(\forall k \in \mathcal{S}_r) \quad c_{r,k} = \begin{cases} c_{r-1,k} + \beta_{r,k} d_r, & \text{si } k \sim i_r \\ c_{r-1,k}, & \text{sinon.} \end{cases} \quad (5.32)$$

- 9: **end for**

**Algorithme 5.1:** Transformée multi-échelles par suite de liftings élémentaires

lifting. Un sommet  $i_r$  est sélectionné, et un coefficient de détail est calculé en  $i_r$ . Ce coefficient de détail correspond à une erreur de prédiction comme expliqué plus haut. Ici, on se restreint à des opérateurs de prédiction locaux correspondant à des combinaisons linéaires convexes reliées aux poids des arêtes concernées :

$$(\forall k \in \mathcal{S}_r, k \sim i_r) \quad \alpha_{r,k} = \frac{w_{i_r,k}}{\sum_{\substack{l \sim i_r \\ l \in \mathcal{S}_r}} w_{i_r,l}}. \quad (5.33)$$

Le choix de la normalisation est motivé par la volonté d'obtenir des coefficients de détail nuls pour des fonctions localement constantes. 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. Le choix des coefficients  $\beta_{r,k}$  sera adressé dans la suite.

Remarquons également que dès lors qu'un coefficient de détail a été calculé en un sommet  $i_r$ , ce sommet n'intervient plus dans les itérations suivantes. Ceci revient à retirer ce sommet du graphe; en particulier toutes les arêtes incidentes à  $i_r$  sont également retirées. Cette remarque a une incidence sur la sélection des sommets (ligne 4 de l'algorithme 5.1).

Avant de détailler la sélection des sommets de détail et l'opérateur de mise à jour, nous étudions l'effet du lifting sur la construction des signaux de base.



### Effet sur la base primale

Comme pour le schéma du lifting classique [Swe96, Swe98], la transformée que nous venons de décrire est bijective. La transformée inverse s'obtient simplement en inversant chaque étape de lifting élémentaire comme décrit précédemment.

La transformée que nous venons de présenter est ainsi linéaire et inversible. Elle correspond donc à un *changement de base*. On s'intéresse ici à l'effet des étapes de prédiction et de mise à jour sur les éléments de cette base.

Notons  $T_r : \mathbb{R}^N \rightarrow \mathbb{R}^N$  la transformée correspondant à l'application de  $r$  étapes de liftings élémentaires :

$$(\forall \mathbf{f} \in \mathbb{R}^N) \quad (T_r \mathbf{f})_l = \begin{cases} c_{r,l}, & \text{si } l \in \mathcal{S}_r, \\ d_l, & \text{si } l \in \mathcal{D}_r. \end{cases} \quad (5.34)$$

L'application linéaire  $T_r$  peut s'écrire sous la forme

$$T_r = E_r \circ \dots \circ E_1, \quad (5.35)$$

où les  $E_i$  désignent les étapes de liftings élémentaires. Notons  $(\phi_{0,i})_{1 \leq i \leq N}$  la base canonique de  $\mathbb{R}^N$  :  $\phi_{0,i}(j) = \delta_{i,j}$ . Le lemme suivant sera utile pour la suite.

**Lemme 5.1.** Pour tout  $i \in \llbracket 1, N \rrbracket$

$$E_r^{-1}(\phi_{0,i}) = \begin{cases} \phi_{0,i} + \alpha_{r,i} \phi_{0,i_r}, & \text{si } i \sim i_r \text{ et } i \in \mathcal{S}_r, \\ - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} \phi_{0,i} + \left( 1 - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \alpha_{r,i} \beta_{r,i} \right) \phi_{0,i_r}, & \text{si } i = i_r, \\ \phi_{0,i}, & \text{sinon.} \end{cases} \quad (5.36)$$

*Démonstration.* La démonstration est immédiate. Pour inverser une étape de lifting élémentaire, il faut d'abord annuler la mise à jour, puis ensuite calculer la prédiction et l'additionner à l'erreur.  $\square$

À chaque étape  $r \in \llbracket 1, m \rrbracket$ , les coefficients de  $T_r \mathbf{f}$  dans la base canonique correspondent aux coordonnées de  $\mathbf{f}$  dans une *nouvelle base*  $\text{MSB}_r := \{\phi_{r,i}, \psi_{r,j}\}_{i \in \mathcal{D}_r, j \in \mathcal{D}_r}$  :

$$\mathbf{f} = \sum_{i \in \mathcal{S}_r} c_{r,i} \phi_{r,i} + \sum_{i \in \mathcal{D}_r} d_i \psi_{r,i}. \quad (5.37)$$

Du fait de la linéarité de  $T_r$ , la base  $\text{MSB}_r$  est obtenue, pour tout  $r \in \llbracket 1, m \rrbracket$ , en appliquant  $T_r^{-1}$  aux vecteurs de la base canonique :

$$\begin{cases} (\forall i \in \mathcal{S}_r) & \phi_{r,i} = T_r^{-1}(\phi_{0,i}), \\ (\forall j \in \mathcal{D}_r) & \psi_{r,j} = T_r^{-1}(\phi_{0,i}). \end{cases} \quad (5.38)$$

La proposition suivante établit une relation entre les vecteurs des bases  $\text{MSB}_r$ .

**Proposition 5.1.** Soit  $r \in \llbracket 2, m \rrbracket$ . La relation de récurrence entre les fonctions d'échelle aux niveaux  $r$  et  $r - 1$  est donnée par :

$$(\forall i \in \mathcal{S}_r) \quad \phi_{r,i} = \begin{cases} \phi_{r-1,i} + \alpha_{r,i} \phi_{r-1,i_r} & \text{si } i \sim i_r \\ \phi_{r-1,i} & \text{sinon.} \end{cases} \quad (5.39)$$

La fonction de détail au niveau  $r$  est donnée par :

$$(\forall i \in \mathcal{D}_r) \quad \psi_{r,i} = \begin{cases} \phi_{r-1,i_r} - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} \phi_{r,i}, & \text{si } i = i_r, \\ \psi_{r-1,i} & \text{sinon.} \end{cases} \quad (5.40)$$

*Démonstration.* La démonstration repose sur le fait que  $T_r = E_r \circ T_{r-1}$  et sur le lemme 5.1. En effet, pour tout  $i \in \mathcal{S}_r$

$$\begin{aligned} \phi_{r,i} &= T_r^{-1}(\phi_{0,i}) = T_{r-1}^{-1} \circ E_r^{-1}(\phi_{0,i}) \\ &= \begin{cases} T_{r-1}^{-1}(\phi_{0,i} + \alpha_{r,i} \phi_{0,i_r}) & \text{si } i \sim i_r, \\ T_{r-1}^{-1}(\phi_{0,i}), & \text{sinon,} \end{cases} \\ &= \begin{cases} \phi_{r-1,i} + \alpha_{r,i} \phi_{r-1,i_r} & \text{si } i \sim i_r, \\ \phi_{r-1,i}, & \text{sinon.} \end{cases} \end{aligned}$$

Pour tout  $i \in \mathcal{D}_r$ ,

$$\begin{aligned} \psi_{r,i} &= T_r^{-1}(\phi_{0,i}) = T_{r-1}^{-1} \circ E_r^{-1}(\phi_{0,i}) \\ &= \begin{cases} T_{r-1}^{-1} \left( - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} \phi_{0,i} + (1 - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \alpha_{r,i} \beta_{r,i}) \phi_{0,i_r} \right) & \text{si } i = i_r, \\ T_{r-1}^{-1}(\phi_{0,i}), & \text{sinon,} \end{cases} \\ &= \begin{cases} T_{r-1}^{-1} \left( \phi_{0,i_r} - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} (\beta_{r,i} \phi_{0,i} - \alpha_{r,i} \phi_{0,i_r}) \right) & \text{si } i = i_r, \\ T_{r-1}^{-1}(\phi_{0,i}), & \text{sinon.} \end{cases} \\ &= \begin{cases} \phi_{r-1,i_r} - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} (\phi_{r-1,i} - \alpha_{r,i} \phi_{r-1,i_r}) & \text{si } i = i_r, \\ \psi_{r-1,i}, & \text{sinon.} \end{cases} \\ &\stackrel{(5.40)}{=} \begin{cases} \phi_{r-1,i_r} - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} \phi_{r,i} & \text{si } i = i_r, \\ \psi_{r-1,i}, & \text{sinon.} \end{cases} \end{aligned}$$

□

### Sélection des sommets de détail

Nous détaillons ici l'étape 4 de l'algorithme 5.1. Au vu de la discussion précédente, nous énonçons deux critères de sélection des sommets. Le premier critère consiste simplement à sélectionner, à chaque étape  $r \in \llbracket 1, m \rrbracket$ , le sommet ayant le plus grand degré parmi les sommets restants :

$$i_r = \arg \max_{i \in \mathcal{S}_{r-1}} \sum_{j \in \mathcal{S}_{r-1}} w_{i,j}. \quad (5.41)$$

Intuitivement, l'erreur de prédiction en un sommet de degré élevé doit être faible. En appliquant ce critère, on cherche donc à favoriser la propriété de parcimonie de la transformée.

Rappelons que la sélection d'un sommet  $i_r$  à une étape  $r$  invalide toutes les arêtes incidentes à  $i_r$  aux étapes suivantes. Afin de prendre en compte cet aspect, le deuxième critère que nous proposons considère le ratio entre le degré et le nombre de voisins :

$$i_r = \arg \max_{i \in \mathcal{S}_{r-1}} \frac{\sum_{j \in \mathcal{S}_{r-1}} w_{i,j}}{\text{Card}(\{j \sim i, j \in \mathcal{S}_{r-1}\})}. \quad (5.42)$$

En particulier, à degré égal, le critère (5.42) favorise le sommet ayant le plus petit nombre de voisins.

### Étape de mise à jour

L'équation (5.40) permet de clarifier l'effet de l'étape de mise à jour ainsi que le calcul des coefficients qui y interviennent. Notons, pour  $\mathbf{f} \in \mathbb{R}^N$ ,  $f$  la valeur moyenne de  $\phi_{r,i}$  et posons  $I_{r,i} = \int \phi_{r,i}$ . Notons que  $I_{0,i} = 1$  pour tout  $i \in \llbracket 1, N \rrbracket$ . Comme pour les transformées multi-échelles classiques, nous souhaitons que le moment d'ordre zéro des fonctions de détail  $\psi_{r,i_r}$  soit nul. Ceci est équivalent, d'après (5.40), à

$$(\forall r \in \llbracket 1, m \rrbracket) \quad I_{r-1,i_r} = \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} I_{r,i}. \quad (5.43)$$

Nous optons, comme [JNS01, JNS09], pour la solution de *norme euclidienne minimale* donnée par

$$\beta_{r,i} = \frac{I_{r,i} I_{r-1,i_r}}{\sum_{\substack{j \sim i_r \\ j \in \mathcal{S}_r}} I_{r,j}^2}. \quad (5.44)$$

## Terminaison de l'algorithme et notion de cycle

Le nombre de liftings élémentaires a été spécifié dans l'algorithme 5.1 par le paramètre  $m$ . Nous précisons ici ce paramètre.

Notons tout d'abord que  $m$  dépend de la topologie du graphe. En effet, chaque lifting élémentaire invalide les arêtes incidentes au sommet où le coefficient de détail est calculé. Ainsi, selon la topologie du graphe considéré, il peut arriver qu'au bout d'un certain nombre  $m$  de liftings élémentaires, toutes les arêtes du graphe soient invalidées. Autrement dit, tous les sommets appartenant à  $\mathcal{S}_m$  n'ont pour voisins que des sommets appartenant à  $\mathcal{D}_m$ . Il n'est alors plus possible d'effectuer de prédiction, et l'algorithme se termine. On dira alors qu'on a effectué *un seul cycle* de lifting.

Nous proposons dans la suite une extension de cet algorithme à plusieurs cycles.

## Transformée à plusieurs cycles

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  $j$ . Nous illustrons dans ce qui suit l'effet de la transformée du lifting que nous proposons sur des images. Nous nous intéressons à l'approximation à  $M$  termes et comparons avec la transformée orthogonale de Haar et celle, biorthogonale de Cohen, Daubechies et Feauveau, CDF(2,2).

### 5.5.4 Expériences

Nous illustrons dans cette sous-section le comportement de la transformée du lifting que nous venons de décrire. Nous nous limitons aux cas des images numériques, bien que l'approche décrite plus haut s'applique à des signaux plus généraux.

L'illustration que nous donnons est l'erreur d'approximation non linéaire à  $M$  termes : la transformée du lifting est appliquée aux images<sup>2</sup> de la figure 5.11. Ensuite, seuls les  $M$  plus grands coefficients de détail (en valeur absolue) sont

---

2. Ces images sont de taille 128x128.

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 :

$$psnr(f, g) = 10 \log_{10} \left( \frac{255^2}{EQM(f, g)} \right), \quad (5.45)$$

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. Enfin, le critère de sélection des sommets est le degré.

La figure 5.12 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 du 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 :

1. Le graphe de 4 voisinage non pondéré ;
2. Le graphe de 8 voisinage pondéré avec des poids gaussiens ( $\sigma = 15$ ) ;
3. Le graphe des 25 plus proches voisins basé sur la distance photométrique dans une fenêtre 21x21. Les distances sont transformées en similarités gaussiennes ( $\sigma = 10$ ).

Les transformées basées sur les configurations 2 et 3 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 du lifting. Les performances montrées dans la figure 5.12 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 du lifting que nous décrivons.

La figure 5.13 et 5.14 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 5.12.

Enfin, les mêmes tests ont été effectués sur l'image "Lena". Les résultats, reportés dans les figures 5.15, 5.16 et 5.17, sont similaires.



FIGURE 5.11 – Images de test.

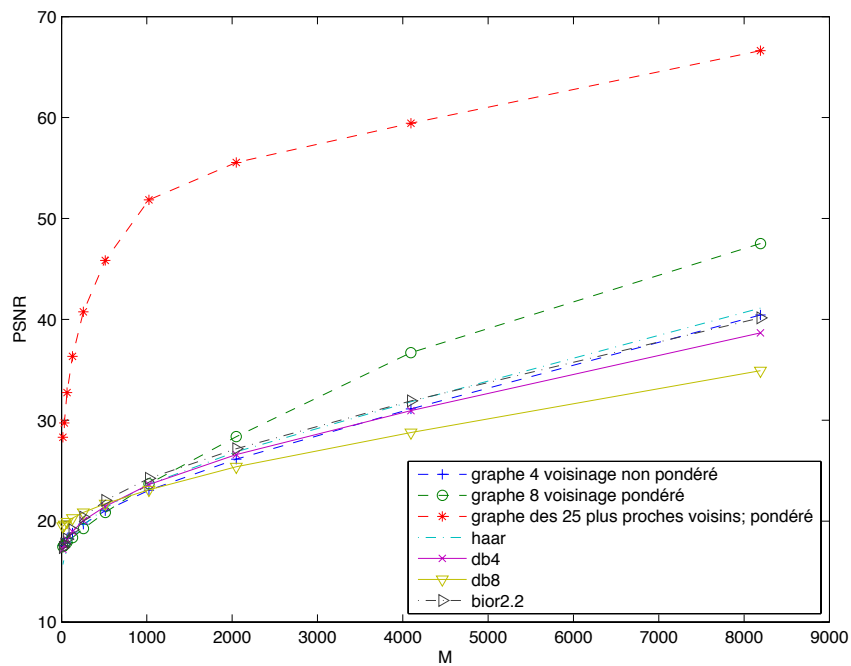


FIGURE 5.12 – Erreur d’approximation non linéaire à  $M$  termes pour l’image “Barbara”. Transformées considérées : lifting sur graphe (4 et 8 voisinage, plus proches voisins), Haar, Daubechies 4, Daubechies 8, CDF(2,2).

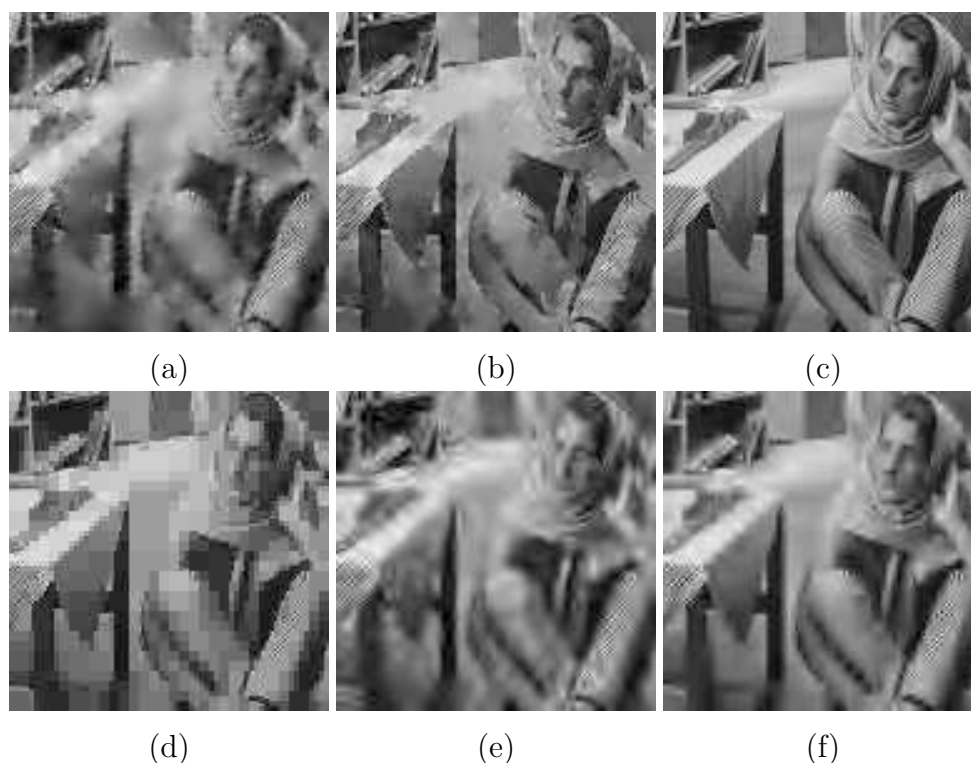


FIGURE 5.13 – Reconstructions de l’image “Barbara” à partir de 1024 coefficients. (a) lifting sur graphe 4 voisinage non pondéré. (b) lifting sur graphe 8 voisinage pondéré. (c) lifting sur graphe des 25 plus proches voisins ; poids gaussiens. (d) Haar. (e) Daubechies 4. (f) : CDF(2.2).

## 5.6 Conclusion

Nous avons présenté dans ce chapitre deux transformées multi-échelles agissant sur des signaux définis sur des graphes généraux. Ces transformées donnent lieu à des décompositions multi-échelles des signaux en questions et correspondent à des changements de base.

La première approche que nous avons développée est inspirée de [GNC10]. Ce travail montre qu’une AMR de l’espace des fonctions sur un graphe peut être obtenue dès que l’on dispose d’une hiérarchie de partitions des sommets de ce graphe. Nous avons proposé un algorithme permettant la construction de telles hiérarchies. Nous avons montré son intérêt au travers des exemples sur des images et des base de données d’images.

Dans la seconde approche, nous avons opté pour une adaptation de l’algorithme du lifting. Nous avons détaillé cette adaptation ainsi que les difficultés qu’elle pose et opté pour une approche gloutonne présentée dans

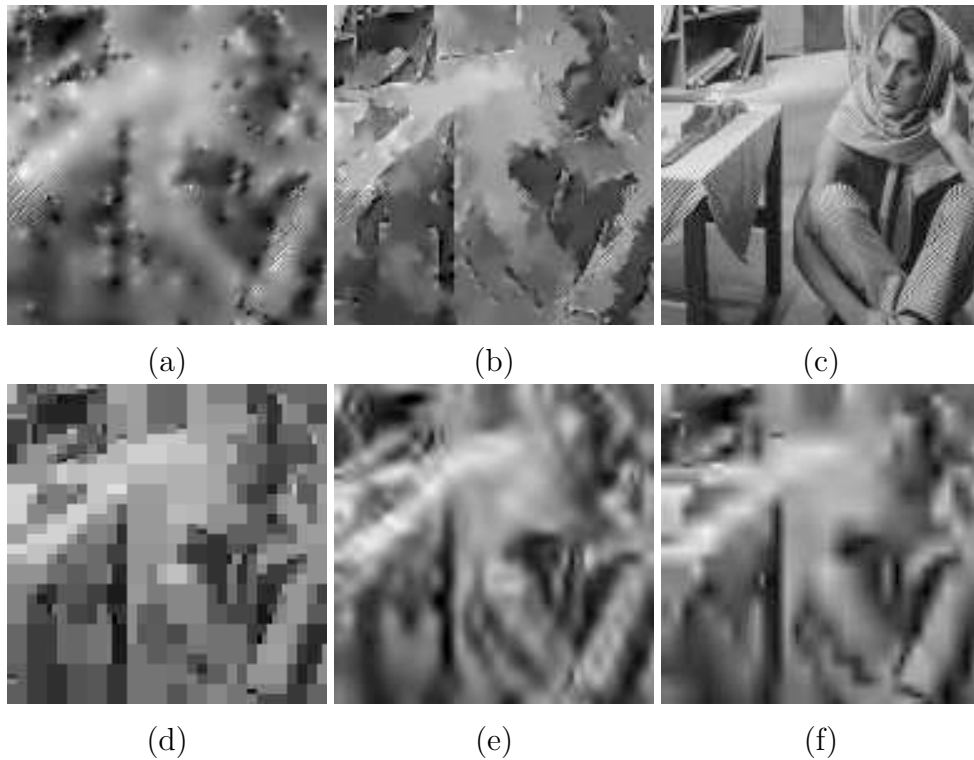


FIGURE 5.14 – Reconstructions de l’image “Barbara” à partir de 256 coefficients. (a) lifting sur graphe 4 voisinage non pondéré. (b) lifting sur graphe 8 voisinage pondéré. (c) lifting sur graphe des 25 plus proches voisins ; poids gaussiens. (d) Haar. (e) Daubechies 4. (f) CDF(2.2).

[JNS09]. En particulier, nous avons proposé un critère de sélection des sommets et avons étendu l’approche de [JNS09] à plusieurs cycles. Des comparaisons concernant l’erreur d’approximation à  $M$  termes ont été effectuées afin, d’une part, d’étudier l’effet du choix du graphe sous-jacent, et, d’autre part, de comparer avec d’autres transformées classiques.



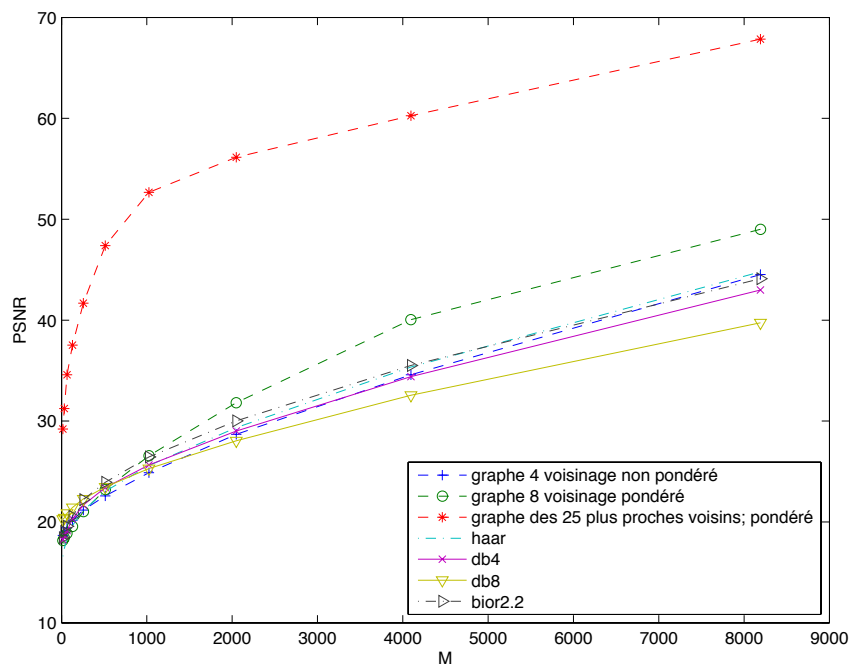


FIGURE 5.15 – Erreur d’approximation non linéaire à  $M$  termes pour l’image “Lena”. Transformées considérées : lifting sur graphe (4 et 8 voisinage, plus proches voisins), Haar, Daubechies 4, Daubechies 8, CDF(2,2).

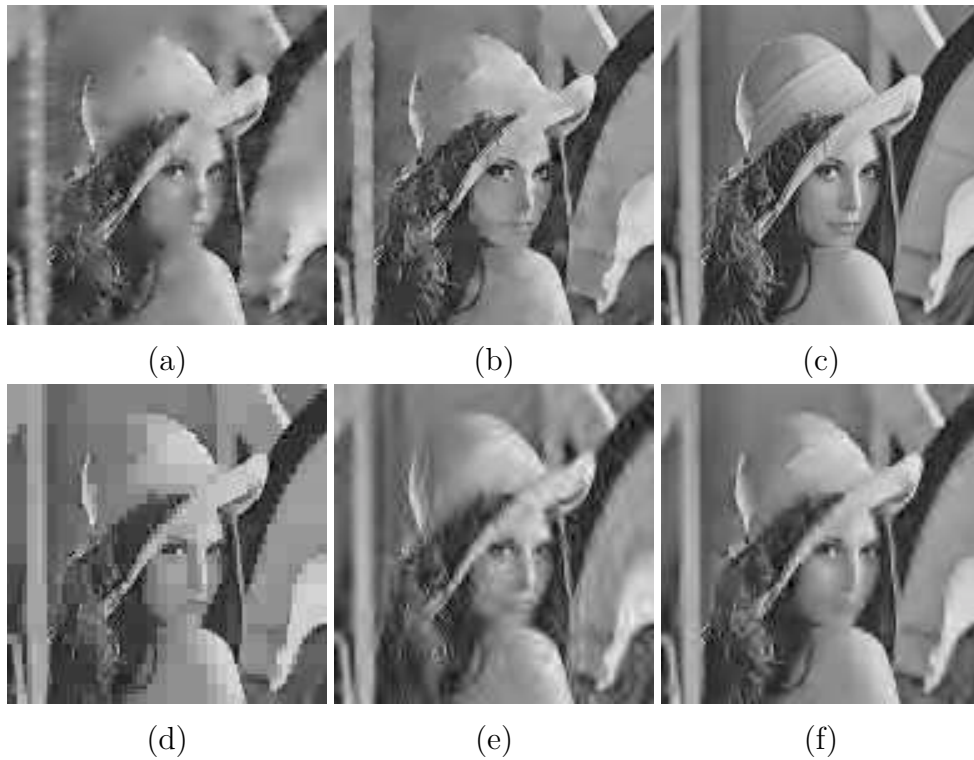


FIGURE 5.16 – Reconstructions de l’image “Lena” à partir de 1024 coefficients. (a) lifting sur graphe 4 voisinage non pondéré. (b) lifting sur graphe 8 voisinage pondéré. (c) lifting sur graphe des 25 plus proches voisins ; poids gaussiens. (d) Haar. (e) Daubechies 4. (f) CDF(2.2).

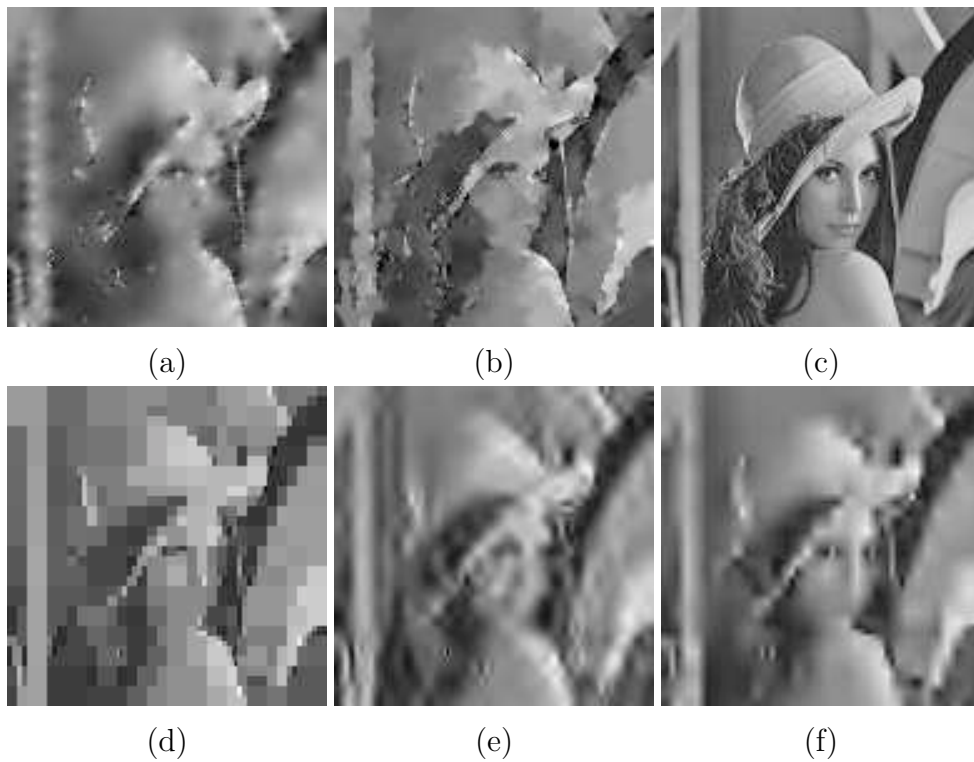


FIGURE 5.17 – Reconstructions de l’image “Lena” à partir de 256 coefficients. (a) lifting sur graphe 4 voisinage non pondéré. (b) lifting sur graphe 8 voisinage pondéré. (c) lifting sur sur graphe des 25 plus proches voisins ; poids gaussiens. (d) Haar. (e) Daubechies 4. (f) CDF(2.2).

# Chapitre 6

## Conclusion et perspectives

Ce dernier chapitre présente plusieurs remarques et perspectives concernant les travaux que nous avons présentés dans les chapitres précédents.

Dans ce manuscrit, nous nous sommes intéressés à la décomposition de signaux définis sur des graphes pondérés généraux. Nous avons développé trois algorithmes permettant de générer de telles décompositions. Nous rappelons chacune d'entre elles dans la suite.

### Décompositions multi-échelles hiérarchiques

Ces décompositions sont basées sur un principe de régularisation itérative introduit dans [TNV04] et que nous avons adapté et généralisé à des signaux sur graphes. Ces décompositions ont été déclinées sous deux formes : l'une provenant d'une fonctionnelle de régularisation quadratique basée sur la matrice laplacienne et l'autre d'une fonction non différentiable de type variation totale. Dans les deux cas, la convergence de la représentation a été établie et le choix des paramètres discuté. La preuve de convergence de la décomposition basée sur l'a priori quadratique s'obtient en considérant l'effet de cette régularisation sur les composantes du signal dans la base des vecteurs propres de la matrice laplacienne. La preuve de convergence de la décomposition basée sur l'a priori de variation totale s'obtient en considérant le problème dual associé au modèle ROF sur graphe. Pour cet a priori, le choix du premier paramètre de décomposition est encadré. Cet encadrement fait apparaître la norme  $G_w$  du signal (centré) que l'on souhaite décomposer. Au niveau des applications, nous avons détaillé l'application de notre approche au travers de multiples applications en rehaussement de détails dans les images et les maillages. La comparaison de notre méthode avec celles de l'état de l'art

s'est révélée satisfaisante et ouvre de nombreuses applications en photographie numérique.

Concernant les perspectives, il nous semble naturel de nous servir de la décomposition que nous obtenons afin de générer des segmentations hiérarchiques d'images et de maillages. L'approche que nous voudrions adopter consiste à appliquer un même algorithme de segmentation aux sommes successives des couches que nous obtenons. La segmentation obtenue à chaque étape sera ainsi raffinée après l'ajout de chaque nouvelle couche de détail, garantissant ainsi l'emboîtement des partitions. Une autre perspective qu'il serait intéressant d'étudier est le choix du premier paramètre d'échelle pour la décomposition avec l'a priori quadratique. Disposer d'une méthode rapide permettant l'automatisation de ce choix serait intéressant. Dans la section portant sur les expériences de cette partie, nous avons utilisé exclusivement des décompositions obtenues avec l'a priori TV. Il serait utile de comparer plus en détail ces résultats à ceux que l'on obtient avec l'a priori quadratique. Une extension de ces travaux au modèle  $(TV, \ell_1)$  (tout comme dans [AT11]), modèle reconnu comme plus apte à capturer les textures, est également un axe de recherche privilégié. Enfin, les applications des décompositions obtenues semblent, d'après les résultats que nous avons présentés, d'un intérêt fort dans le domaine de la photographie numérique.

## **Transformée de Haar sur une hiérarchie**

Nous avons abordé dans cette partie de notre travail la construction d'ondelettes de Haar sur une hiérarchie de partitions comme proposé dans [GNC10]. Nous avons présenté ensuite un algorithme permettant la construction adaptative d'une hiérarchie de partitions à partir d'un graphe pondéré. Au niveau des expériences, nous avons montré les décompositions obtenues sur des images ainsi que sur des bases d'images.

Nous projetons d'utiliser les résultats de cette partie pour le débruitage d'images. L'approche que nous voudrions adopter serait de type "wavelet shrinkage", c'est-à-dire qu'elle consisterait à seuiller individuellement les coefficients de détail avant d'appliquer la transformée inverse. Dans le cas d'un bruit blanc gaussien, du fait de l'orthogonalité de la transformée, le problème de l'estimation dans le domaine spatial (ou bien du graphe) est équivalent à celui de l'estimation dans le domaine des coefficients de la transformée. Notons que ce type d'approche a été introduit récemment, pour une construction d'arbre différente, dans [REC11, REC12]. Une autre question qui nous semble importante concerne la construction de la hiérarchie. Les théorèmes 2 et 3 de [GNC10] fournissent respectivement des encadrements des magnitudes des

---

coefficients de détail et de l'erreur  $\ell_1$  d'approximation. Ces deux résultats restent à étudier afin de savoir s'ils peuvent guider l'étape de construction de la hiérarchie, en vue d'une meilleure compressibilité des coefficients de détail.

## Lifting sur graphes

Nous avons développé dans la dernière partie de ce travail une transformée multi-échelle basée sur le schéma de lifting. L'adaptation de cet algorithme pose des problèmes. En particulier, le problème du partitionnement des sommets du graphe se révèle particulièrement délicat. Nous avons opté pour une approche basée sur des étapes de liftings élémentaires, où un unique coefficient de détail est calculé à chaque étape. Au niveau des expériences, nous nous sommes limités au calcul de l'erreur d'approximation non linéaire sur des images.

Parmi les perspectives que nous envisageons, la première concerne l'intégration d'une étape d'orthonormalisation à la fin de chaque lifting élémentaire. Ces étapes permettent d'obtenir une transformée unitaire et d'envisager une approche de type "wavelet shrinkage" pour le débruitage. Une autre perspective serait de définir une transformée de lifting redondante. Un moyen simple pour obtenir de la redondance consisterait à appliquer plusieurs fois la transformée décrite ici, en modifiant le critère de sélection des sommets à chaque itération.

## Remarques finales

Nous concluons ce chapitre par quelques remarques générales. Les ensembles de données que nous avons considérés dans ce manuscrit ont été définis par des images, des bases d'images, des maillages 3D et des nuages de points en dimension 3. Nous souhaiterions dans le futur appliquer nos transformées à d'autres signaux sur graphes, par exemple à des réseaux sociaux ou de communication. Le challenge applicatif est très actuel et important mais également différent de celui que nous avons considéré dans cette thèse. En effet, dans de tels exemples de réseaux complexes, les signaux ne résident plus sur les sommets du graphe mais uniquement sur les arêtes à travers lesquelles est encodée une notion d'interaction. Nous projetons dans une première approche de considérer le graphe dual pour explorer de telles données directement à l'aide de nos algorithmes.



# Liste des figures

2.1	s/t coupe d'un graphe pondéré orienté. . . . .	19
2.2	50 points générés aléatoirement dans $[0, 5] \times [0, 5]$ . . . . .	22
2.3	Graphe complet sur les points de la figure 2.2. . . . .	23
2.4	Triangulation de Delaunay des points de la figure 2.2. . . . .	23
2.5	Un MST sur les points de la figure 2.2. . . . .	24
2.6	Graphe des $k$ plus proches voisins avec $k = 2$ . . . . .	24
2.7	Graphe des $k$ plus proches voisins mutuels avec $k = 2$ . . . . .	25
2.8	$\epsilon$ -graphe avec $\epsilon = 1$ . . . . .	25
2.9	Partie de l'image "Lena". . . . .	35
2.10	Graphe-grille construit de à partir de l'image "Lena" de la figure 2.9. . . . .	35
2.11	Graphe des 4 plus proches voisins au sens de la distance photométrique. La recherche des voisins est limitée à une fenêtre de taille 11x11. . . . .	36
2.12	$\epsilon$ -graphe au sens de la distance photométrique. La recherche de voisins est limitée à une fenêtre de taille 11x11. . . . .	36
2.13	Graphe des 4 plus proches voisins au sens de la distance euclidienne entre patches. La recherche des voisins est limité à une fenêtre de taille 11x11, et les patches sont de taille 5x5. . . . .	37
3.1	Images de test et leur version bruitée avec un bruit blanc gaussien d'écart-type $\sigma = 20$ . . . . .	52
3.2	Images débruitées par minimisation d'énergie. La colonne de gauche présente les résultats obtenus avec l'a priori $J_2$ . Celle de droite présente les résultats obtenus avec une pénalisation TV. Les images montrées sont obtenues pour des valeurs de $\lambda$ optimales en terme de PSNR. . . . .	53



3.3	Comparaison des résultats obtenus sur le débruitage de “Barbara” relativement à différentes configurations de graphes et deux différentes énergies. . . . .	69
3.4	Comparaison de la simplification d’un maillage triangulaire obtenue par deux énergies différentes et deux fonctions de poids différentes. . . . .	70
3.5	Comparaison de la simplification des attributs de couleur d’un maillage triangulaire obtenue par deux énergies différentes et deux fonctions de poids différentes. . . . .	71
4.1	Illustration de l’utilisation de la décomposition structure-texture pour l’inpaiting simultané des structures et textures dans les images. . . . .	77
4.2	Approche pour la segmentation d’images introduite dans [CPGN <sup>+</sup> 12]. La première étape consiste en une décomposition structure-texture. . . . .	78
4.3	Illustration de la problématique du rehaussement de détails. . . . .	79
4.4	Édition du maillage à gauche. . . . .	79
4.5	Décomposition de l’image d’une empreinte en 5 étapes. . . . .	83
4.6	Somme des couches successives extraites de l’image "Barbara". Les résultats sont obtenus avec un graphe grille non pondéré en 4-voisinage. La dernière image (en bas à droite) présente le dernier résidu. . . . .	96
4.7	Somme des couches successives extraites de l’image "Barbara". Les résultats sont obtenus avec un graphe non pondéré des 10 plus proches voisins. La dernière image (en bas à droite) présente le dernier résidu. . . . .	97
4.8	Couches extraites de l’image "Barbara". La première ligne présente les résultats obtenus avec un graphe grille non pondéré en 4-voisinage. La seconde ligne présente les résultats obtenus avec un graphe non pondéré des 10 plus proches voisins. . . . .	98
4.9	MSE (Mean Squared Error) entre les somme des différentes couches et l’image originale "Barbara" pour les deux constructions de graphe (grille ou plus proches voisins). . . . .	99
4.10	Décomposition d’une image d’empreinte digitale. Première ligne : image originale. Deuxième ligne : graphe non pondéré de 4-voisinage. Troisième ligne : graphe de 8-voisinage pondéré par poids gaussiens. Quatrième ligne : 25-plus proches voisins avec poids gaussiens sur patches 5 × 5. . . . .	100

4.11	Somme des couches successives extraites sur un maillage avec attributs géométriques (présenté en haut à gauche). La topologie du graphe est connue a priori et les arêtes sont pondérées par l'inverse de la distance euclidienne. . . . .	101
4.12	Somme des couches successives extraites sur un maillage avec attributs photométriques (présenté sur la première ligne). La topologie du graphe est connue a priori et les arêtes sont pondérées par des poids gaussiens. . . . .	102
4.13	Somme des couches successives extraites sur un nuage de points d'une hélice toroïdale bruitée. La topologie du graphe est un 10 plus proches voisins et les arêtes sont pondérées par l'inverse de la distance euclidienne. Chaque sommet du graphe est identifié par une unique couleur. . . . .	103
4.14	Comparaison des résultats obtenus par notre approche (8-voisinage, poids gaussiens) sur une image avec différentes approches de l'état de l'art. . . . .	104
4.15	Sharpening d'une image de fleur. . . . .	106
4.16	Comparaison de notre approche de rehaussement de détails fins avec des approches de l'état de l'art sur l'image de fleur de la figure 4.15. . . . .	106
4.17	Rehaussement des détails d'une feuille de rhubarbe. . . . .	107
4.18	Illustrations des problèmes de halo (BLF) et de modification d'intensité globale (WLS). $\delta_1 = 2$ , $\delta_2 = 0.05$ . . . . .	108
4.19	Principe du tone mapping [DD02]. . . . .	109
4.20	Comparaison avec différents algorithmes de tone mapping. . . . .	110
4.21	Retouche d'un portrait pour un effacement d'acné. . . . .	111
4.22	Retouche interactive d'un portrait et comparaison avec le plugin Portraiture de photoshop. . . . .	112
4.23	Rehaussement d'un maillage comportant peu de noeuds (1487 noeuds, 2918 faces). . . . .	114
4.24	Rehaussement d'un maillage (43533 noeuds, 88002 faces). . . . .	115
4.25	Maillage coloré de faible qualité rehaussé (17215 noeuds, 29461 faces). . . . .	116
4.26	Maillage coloré de haute qualité rehaussé (553053 noeuds, 1105611 faces). . . . .	117
5.1	Illustration des fonctions de base obtenues à partir d'une hiérarchie de partitions. . . . .	129

5.2	Décomposition multi-échelle de "Barbara" : graphe en 8-voisinage, poids gaussiens. . . . .	134
5.3	Décomposition multi-échelle de "Barbara" : graphe des 7-plus proches voisins, poids gaussiens sur patchs $5 \times 5$ dans un voisinage $11 \times 11$ . . . . .	135
5.4	Décomposition multi-échelle de "Man" : graphe des 7-plus proches voisins, poids gaussiens sur patchs $5 \times 5$ dans un voisinage $11 \times 11$ . . . . .	136
5.5	Graphes des niveaux de décomposition des 400 images de la base USPS. . . . .	137
5.6	Graphes des niveaux de décomposition des 400 images de la base Olivetti. . . . .	138
5.7	Décomposition de 400 images de la base USPS : fonctions de détail et d'échelle des niveaux 3 et 5. . . . .	139
5.8	Décomposition de 400 images de la base Olivetti : fonctions de détail et d'échelle des niveaux 3 et 5. . . . .	140
5.9	Transformée du lifting directe. . . . .	142
5.10	Transformée du lifting inverse. . . . .	142
5.11	Images de test. . . . .	153
5.12	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). . . . .	153
5.13	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). . . . .	154
5.14	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). . . . .	155
5.15	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). . . . .	156

5.16	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). . . . .	157
5.17	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). . . . .	158



# Liste des algorithmes

3.1	Algorithme des $\alpha$ -expansions . . . . .	67
4.1	Projection sur une boule de rayon $r$ pour la norme $\ \cdot\ _{1,2}$ . . . . .	94
5.1	Transformée multi-échelles par suite de liftings élémentaires . . . . .	147



# Liste des Publications

## Articles de revues internationales

1. M. Hidane, O. Lézoray, and A. Elmoataz. Nonlinear multilayered representation of graph-signals. *Journal of Mathematical Imaging and Vision*, 45(2) :114–137, 2013.

## Articles de conférences internationales

1. M. Hidane, O. Lézoray, V.-T. Ta, and A. Elmoataz. Nonlocal multiscale hierarchical decomposition on graphs. In *European Conference on Computer Vision (ECCV)*, volume LNCS 6314, pages 638–650, 2010.
2. M. Hidane, O. Lézoray, and A. Elmoataz. Hierarchical representation of discrete data on graphs. In *International Conference on Computer Analysis of Images and Patterns (IAPR CAIP)*, volume LNCS 6854, pages 186–193, 2011.
3. M. Hidane, O. Lézoray, and A. Elmoataz. A scale-space based hierarchical representation of discrete data. In *International Conference on Image Processing (IEEE ICIP)*, pages 285–288, 2011.
4. M. Hidane, O. Lézoray, A. Elmoataz, and V.-T. Ta. Représentation multi-échelle hiérarchique de données discrètes définies sur des graphes. In *colloque GRETSI*, 2011.





# Bibliographie

- [AABFC05] Jean-François Aujol, Gilles Aubert, Laure Blanc-Féraud, and Antonin Chambolle. Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vis.*, 22(1) :71–88, January 2005.
- [AC05] Jean-François Aujol and Antonin Chambolle. Dual norms and image decomposition models. *Int. J. Comput. Vision*, 63(1) :85–104, June 2005.
- [AD00] F. Arandiga and R. Donat. Nonlinear multiscale decompositions : The approach of a. harten. *Numerical Algorithms*, 23(2) :175–216, 2000.
- [ADLT07] S. Amat, R. Donat, J. Liandrato, and J.C. Trillo. A fully adaptive multiresolution scheme for image processing. *Mathematical and computer modelling*, 46(1) :2–11, 2007.
- [AGCO06] Jean-François Aujol, Guy Gilboa, Tony Chan, and Stanley Osher. Structure-texture image decomposition–modeling, algorithms, and parameter selection. *Int. J. Comput. Vision*, 67(1) :111–136, April 2006.
- [AK06] G. Aubert and P. Kornprobst. *Mathematical problems in image processing : partial differential equations and the calculus of variations*, volume 147. Springer, 2006.
- [Ana00] G. Anastassiou. *Handbook of Analytic Computational Methods in Applied Mathematics*. Chapman & Hall/CRC, 2000.
- [AT11] P. Athavale and E. Tadmor. Integro-differential equations based on  $(bv, l^1)$  image decomposition. *SIAM Journal on Imaging Sciences*, 4(1) :300–312, 2011.
- [BA83] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4) :532–540, 1983.
- [BC11] H.H. Bauschke and P.L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

- [BCM10] A. Buades, B. Coll, and J-M. Morel. Image denoising methods. A new non-local principle. *SIAM Review*, 52(1) :113–147, 2010.
- [BCW08] R. Baraniuk, A. Cohen, and R. Wagner. Approximation and compression of scattered data by meshless multiscale decompositions. *Applied and Computational Harmonic Analysis*, 25(2) :133–147, 2008.
- [BEM07] S. Bogleux, A. Elmoataz, and M. Melkemi. Discrete regularization on weighted graphs for image and mesh filtering. *Scale Space and Variational Methods in Computer Vision*, pages 128–139, 2007.
- [BEM09] S. Bogleux, A. Elmoataz, and M. Melkemi. Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. *International journal of computer vision*, 84(2) :220–236, 2009.
- [Bes93] J. Besag. Towards bayesian image analysis\*. *Journal of Applied Statistics*, 20(5-6) :107–119, 1993.
- [BK04] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9) :1124–1137, 2004.
- [BKP<sup>+</sup>10] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon mesh processing*. AK Peters Limited, 2010.
- [BL06] Jonathan M Borwein and Adrian S Lewis. *Convex Analysis and Nonlinear Optimization : theory and examples*. Springer, 2006.
- [BLMV10] A. Buades, T.M. Le, J.M. Morel, and L.A. Vese. Fast cartoon+ texture image filters. *Image Processing, IEEE Transactions on*, 19(8) :1978–1986, 2010.
- [BPD06] S. Bae, S. Paris, and F. Durand. Two-scale tone management for photographic look. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 637–645. ACM, 2006.
- [BSCB00] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*, pages 417–424, 2000.
- [BSFM07] J. Bobin, J.L. Starck, J. Fadili, and Y. Moudden. Sparsity and morphological diversity in blind source separation. *Image Processing, IEEE Transactions on*, 16(11) :2662–2674, 2007.

- [BT06] X. Bresson and J.-P. Thiran. Image segmentation model using active contour and image decomposition. In *Image Processing, 2006 IEEE International Conference on*, pages 1657–1660, oct. 2006.
- [BT09a] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18(11) :2419–2434, 2009.
- [BT09b] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1) :183–202, 2009.
- [BVSO03] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *Image Processing, IEEE Transactions on*, 12(8) :882–889, 2003.
- [BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11) :1222–1239, 2001.
- [CCC+10] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An introduction to total variation for image analysis. *Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter, Radon Series Comp. Appl. Math*, 9 :263–340, 2010.
- [CD<sup>+</sup>95] R.R. Coifman, D.L. Donoho, et al. Translation-invariant denoising. *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG-*, pages 125–125, 1995.
- [CDSB03] R.L. Claypoole, G.M. Davis, W. Sweldens, and R.G. Baraniuk. Nonlinear wavelet transforms for image coding via lifting. *Image Processing, IEEE Transactions on*, 12(12) :1449–1459, 2003.
- [CDVLL98] A. Chambolle, R.A. DeVore, Nam-Yong Lee, and B.J. Lucier. Nonlinear wavelet image processing : variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7(3) :319–335, march 1998.
- [CE05] Tony F. Chan and Selim Esedoglu. Aspects of total variation regularized  $l_1$  function approximation. *SIAM J. Appl. Math*, 2005.
- [Cha04] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20 :89–97, 2004.

- [Cha05] Antonin Chambolle. Total variation minimization and a class of binary mrf models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757 of *LNCS*, pages 136–152. 2005.
- [Chr08] O. Christensen. *Frames and bases : An introductory course*. Birkhäuser Boston, 2008.
- [Chu97] F.R.K. Chung. *Spectral graph theory*, volume 92. Amer Mathematical Society, 1997.
- [Cia88] P.G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. 1988.
- [CK03] M. Crovella and E. Kolaczyk. Graph wavelets for spatial traffic analysis. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1848–1857. IEEE, 2003.
- [CL97] A Chambolle and P. L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2) :167–188, 1997.
- [CLL11] A. Chambolle, S.E. Levine, and B.J. Lucier. An upwind finite-difference method for total variation-based image smoothing. *SIAM Journal on Imaging Sciences*, 4(1) :277–299, 2011.
- [CLRS01] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2001.
- [CM01] A. Cohen and B. Matei. Compact representation of images by edge adapted multiscale transforms. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 8–11. IEEE, 2001.
- [CM06] R.R. Coifman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1) :53–94, 2006.
- [CM11] Anustup Choudhury and Gérard G. Medioni. Perceptually motivated automatic sharpness enhancement using hierarchy of non-local means. In *ICCV Workshops*, pages 730–737. IEEE, 2011.
- [COS01] T.F. Chan, S. Osher, and J. Shen. The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10(2) :231 –241, feb 2001.
- [CP11a] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1) :120–145, 2011.

- [CP11b] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, volume 49, pages 185–212. Springer, 2011.
- [CPD07a] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graph.*, 26(3), July 2007.
- [CPD07b] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graph.*, 26(3) :103, 2007.
- [CPGN<sup>+</sup>12] W. Casaca, A. Paiva, E. Gomez-Nieto, P. Joia, and L.G. Nوناتo. Spectral image segmentation using image decomposition and inner product-based metric. *Journal of Mathematical Imaging and Vision*, pages 1–12, 2012.
- [CR05] Emmanuel J. Candes and Justin K. Romberg. Signal recovery from random projections. volume 5674, pages 76–86. SPIE, 2005.
- [CSZ<sup>+</sup>] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*, volume 2.
- [CT03] Prasun Choudhury and Jack Tumblin. The trilateral filter for high contrast images and meshes. In Per H. Christensen, Daniel Cohen-Or, and Stephen N. Spencer, editors, *Rendering Techniques*, volume 44 of *ACM International Conference Proceeding Series*, pages 186–196. Eurographics Association, 2003.
- [CVGO10] R. Choksi, Y. Van Gennip, and A. Oberman. Anisotropic total variation regularized  $l^1$ -approximation and denoising/deblurring of 2d bar codes. *arXiv preprint arXiv :1007.1035*, 2010.
- [CW05] Patrick L. Combettes and Valerie R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4) :1168–1200, 2005.
- [CYY<sup>+</sup>10] B. Cheng, J. Yang, S. Yan, Y. Fu, and T.S. Huang. Learning with  $l_1$  graph for image analysis. *IEEE Transactions on Image Processing*, 19(4), 2010.
- [D<sup>+</sup>92] I. Daubechies et al. *Ten lectures on wavelets*, volume 61. SIAM, 1992.

- [DD02] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM transactions on graphics*, 21(3) :257–266, 2002.
- [DDDM04] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11) :1413–1457, 2004.
- [DFKE07] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8) :2080–2095, 2007.
- [DFL08] Ingrid Daubechies, Massimo Fornasier, and Ignace Loris. Accelerated projected gradient method for linear inverse problems with sparsity constraints. *Journal of Fourier Analysis and Applications*, 14 :764–792, 2008.
- [DJ94] D.L. Donoho and J.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3) :425–455, 1994.
- [DKS09] S.I. Daitch, J.A. Kelner, and D.A. Spielman. Fitting a graph to vector data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 201–208. ACM, 2009.
- [DN<sup>+</sup>99] G.M. Davis, A. Nosratinia, et al. Wavelet-based image coding—an overview. *Applied and computational control, signals, and circuits.*, 1 :369–434, 1999.
- [Don95] D.L. Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3) :613–627, 1995.
- [DS98] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier analysis and applications*, 4(3) :247–269, 1998.
- [DS06] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part i : Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3) :261–276, 2006.
- [DSvS04] V. Delouille, J. Simoens, and R. von Sachs. Smooth design-adapted wavelets for nonparametric stochastic regression. *Journal of the American Statistical Association*, 99(467) :643–658, 2004.
- [EA02] K. Egiazarian and J. Astola. Tree-structured haar transforms. *Journal of Mathematical Imaging and Vision*, 16(3) :269–279, 2002.

- [EDD<sup>+</sup>95] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182. ACM, 1995.
- [EL99] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.
- [Ela10] M. Elad. Sparse and redundant representations. *From theory to applications in signal and image processing*, Springer-New York, 2010.
- [ELB08] Abderrahim Elmoataz, Olivier Lézoray, and Sébastien Bougleux. Nonlocal discrete regularization on weighted graphs : A framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7) :1047–1060, jul 2008.
- [ESQD05] M. Elad, J.L. Starck, P. Querre, and D.L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis*, 19(3) :340–358, 2005.
- [FAR07] R. Fattal, M. Agrawala, and S. Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. *ACM Transactions on Graphics*, 26(3) :51, 2007.
- [FFLS08] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM Transactions on Graphics (TOG)*, volume 27, page 67. ACM, 2008.
- [FM01] S. Fitzpatrick and L. Meertens. An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs. *Stochastic Algorithms : Foundations and Applications*, pages 345–347, 2001.
- [FP11] J.M. Fadili and G. Peyré. Total variation projection with first order schemes. *IEEE Transactions on Image Processing*, 20(3) :657–669, march 2011.
- [FSBM10] M.J. Fadili, J.L. Starck, J. Bobin, and Y. Moudden. Image decomposition and separation using sparse representations : an overview. *Proceedings of the IEEE*, 98(6) :983–994, 2010.
- [Gem90] D. Geman. Random fields and inverse problems in imaging. *École d’Été de Probabilités de Saint-Flour XVIII-1988*, pages 113–193, 1990.



- [GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6) :721–741, 1984.
- [GG86] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. AMS, Providence, RI, 1986.
- [Gil09] J. Gilles. Image decomposition : Theory, numerical schemes, and performance evaluation. *Advances in Imaging and Electron Physics*, 158 :89–137, 2009.
- [GLMV07] John B. Garnett, Triet M. Le, Yves Meyer, and Luminita A. Vese. Image decompositions using bounded variation and generalized homogeneous besov spaces. *Applied and Computational Harmonic Analysis*, 23(1) :25 – 56, 2007.
- [GNC10] M. Gavish, B. Nadler, and R.R. Coifman. Multiscale wavelets on trees, graphs and high dimensional data : Theory and applications to semi supervised learning. In *Proc. International Conference on Machine Learning, Haifa, Israel*, 2010.
- [GO07] G. Gilboa and S. Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation*, 6(2) :595–630, 2007.
- [GO08] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling and Simulation*, 7(3) :1005–1028, 2008.
- [GO11] E.S.L. Gastal and M.M. Oliveira. Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics (TOG)*, 30(4) :69, 2011.
- [GP10] L.J. Grady and J.R. Polimeni. *Discrete Calculus : Applied Analysis on Graphs for Computational Science*. Springer, 2010.
- [GPS89] DM Greig, BT Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.
- [GW99] C. Gasquet and P. Witomski. *Fourier analysis and applications : filtering, numerical computation, wavelets*. Texts in applied mathematics. Springer, 1999.
- [Har93] A. Harti. Discrete multi-resolution analysis and generalized wavelets. *Applied numerical mathematics*, 12(1) :153–192, 1993.

- [Har96] A. Harten. Multiresolution representation of data : A general framework. *SIAM Journal on Numerical Analysis*, 33(3) :1205–1256, 1996.
- [HJ07] B. Huang and T. Jebara. Loopy belief propagation for bipartite maximum weight b-matching. *Artificial Intelligence and Statistics (AISTATS)*, 2007.
- [HJV12] D.K. HAMMOND, L. Jacques, and P. Vandergheynst. *Image Processing and Analysis with Graphs : Theory and Practice*, chapter 1. CRC press, 2012.
- [HLE11a] M. Hidane, O. Lézoray, and A. Elmoataz. Hierarchical representation of discrete data on graphs. In *Computer Analysis of Images and Patterns*, pages 186–193. Springer, 2011.
- [HLE11b] M. Hidane, O. Lezoray, and A. Elmoataz. A scale-space based hierarchical representation of discrete data. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 285–288. IEEE, 2011.
- [HLE12] M. Hidane, O. Lézoray, and A. Elmoataz. Nonlinear multilayered representation of graph-signals. *Journal of Mathematical Imaging and Vision*, pages 1–24, 2012.
- [HPFS11] L. Holmström, L. Pasanen, R. Furrer, and S.R. Sain. Scale space multiresolution analysis of random signals. *Computational Statistics & Data Analysis*, 55(10) :2840–2855, 2011.
- [HPPP05] H.J.A.M. Heijmans, B. Pesquet-Popescu, and G. Piella. Building nonredundant adaptive wavelets by update lifting. *Applied and Computational Harmonic Analysis*, 18(3) :252–281, 2005.
- [Hsu83] C.P. Hsu. Minimum-via topological routing. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 2(4) :235–246, 1983.
- [HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning : data mining, inference, and prediction*, 2001.
- [HUL01] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.
- [HVG11] D.K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2) :129–150, 2011.

- 
- [Ish03] H. Ishikawa. Exact optimization for markov random fields with convex priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10) :1333–1336, 2003.
- [JDCP11] L. Jacques, L. Duval, C. Chaux, and G. Peyré. A panorama on multiscale geometric representations, intertwining spatial, directional and frequency selectivity. *Signal Processing*, 91(12) :2699–2730, 2011.
- [JNS01] M. Jansen, G.P. Nason, and B.W. Silverman. Scattered data smoothing by empirical bayesian shrinkage of second generation wavelet coefficients. In *Proc. SPIE*, volume 4478, pages 87–97, 2001.
- [JNS09] M. Jansen, G.P. Nason, and BW Silverman. Multiscale methods for data on graphs and irregular multidimensional situations. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 71(1) :97–125, 2009.
- [JWC09] T. Jebara, J. Wang, and S.F. Chang. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 441–448. ACM, 2009.
- [KPPBB<sup>+</sup>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 Signal Processing : Special Issue on New Image and Video Representations Based on Sparsity*, 2012 :22, 2012.
- [KRFB06] E.A. Khan, E. Reinhard, R.W. Fleming, and H.H. Bühlhoff. Image-based material editing. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 654–663. ACM, 2006.
- [KZ04] V. Kolmogorov and R. Zabini. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2) :147–159, 2004.
- [LEB07] O. Lezoray, A. Elmoataz, and S. Bougleux. Graph regularization for color image processing. *Computer Vision and Image Understanding*, 107(1) :38–55, 2007.
- [LG12] O. Lezoray and L. Grady. *Image Processing and Analysis With Graphs : Theory and Practice*, volume 5. CRC Press, 2012.
- [Lin93] T. Lindeberg. *Scale-space theory in computer vision*. Springer, 1993.

- [LNW08] A.B. Lee, B. Nadler, and L. Wasserman. Treelets—an adaptive multi-scale basis for sparse unordered data. *The Annals of Applied Statistics*, 2(2) :435–471, 2008.
- [LTE10] Olivier Lézoray, Vinh-Thong Ta, and Abderrahim Elmoataz. Partial differences as tools for filtering data on graphs. *Pattern Recognition Letters*, 31(14) :2201 – 2213, 2010.
- [LVDV11] N. Leonardi and D. Van De Ville. Wavelet frames on graphs defined by fmri functional connectivity. In *Biomedical Imaging : From Nano to Macro, 2011 IEEE International Symposium on*, pages 2136–2139. IEEE, 2011.
- [Mal89] S.G. Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7) :674–693, 1989.
- [Mal09] S. Mallat. *A Wavelet Tour of Signal Rrocessing, the Sparse way*. Academic Press, 2009.
- [Mas02] S. Masnou. Disocclusion : a variational approach using level lines. *Image Processing, IEEE Transactions on*, 11(2) :68–76, 2002.
- [MEDdMO11] E. Martinez-Enriquez, E. Diaz-de Maria, and A. Ortega. Video encoder based on lifting transforms on graphs. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3509–3512. IEEE, 2011.
- [MEO11] E. Martínez-Enríquez and A. Ortega. Lifting transforms on graphs for video coding. In *Data Compression Conference (DCC), 2011*, pages 73–82. IEEE, 2011.
- [Mey01] Yves Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations : The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. American Mathematical Society, 2001.
- [Mil11] P. Milanfar. A tour of modern image filtering. *IEEE Signal Processing Magazine*, 2, 2011.
- [ML12a] F. Malgouyres and N. Lermé. Non-heuristic reduction of the graph in graph-cut optimization. In *Journal of Physics : Conference Series*, volume 386, page 012002. IOP Publishing, 2012.
- [ML12b] Francois Malgouyres and Nicolas Lermé. A Non-Heuristic Reduction Method For Graph Cut Optimization. Technical report, April 2012. to appear in NCMIP.

- [Moh91] Bojan Mohar. The laplacian spectrum of graphs. *Graph theory combinatorics and applications*, 2(6) :871–898, 1991.
- [Moh97] B. Mohar. Some applications of laplace eigenvalues of graphs. *Graph symmetry : Algebraic methods and applications*, 497(22) :227, 1997.
- [Mor62] J J Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *C. R. Acad. Sci. Paris Sér. A Math.*, 255 :2897–2899, 1962.
- [MS89] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5) :577–685, 1989.
- [Mur07] F. Murtagh. The haar wavelet transform of a dendrogram. *Journal of Classification*, 24(1) :3–32, 2007.
- [Nad64] E.A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1) :141–142, 1964.
- [Nes03] Y. Nesterov. *Introductory lectures on convex optimization : A basic course*, volume 87. Springer, 2003.
- [NO09] S.K. Narang and A. Ortega. Lifting based wavelet transforms on graphs. In *Proceedings : APSIPA ASC 2009 : Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*, pages 441–444. Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, 2009.
- [NO12] S.K. Narang and A. Ortega. Perfect reconstruction two-channel wavelet filter banks for graph structured data. *Signal Processing, IEEE Transactions on*, 60(6) :2786–2799, 2012.
- [OBG<sup>+</sup>05] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2) :460–489, 2005.
- [OCDD01] B.M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 433–442. ACM, 2001.
- [OS75] Alan V. Oppenheim and Ronald W. Schaffer. *Digital Signal Processing*. Prentice–Hall, 1975.

- [PBC08] Gabriel Peyré, Sébastien Bogleux, and Laurent Cohen. Non-local regularization of inverse problems. In *Computer Vision ECCV 2008*, volume 5304 of *LNCS*, pages 57–68. 2008.
- [PFFG98] S.N. Pattanaik, J.A. Ferwerda, M.D. Fairchild, and D.P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 287–298. ACM, 1998.
- [PFS<sup>+</sup>07] G. Peyré, J. Fadili, J.L. Starck, et al. Learning adapted dictionaries for geometry and texture separation. *Proceedings of SPIE, Wavelets XII*, 6701, 2007.
- [PGK02] M. Pauly, M. Gross, and L.P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Visualization, 2002. VIS 2002. IEEE*, pages 163–170. IEEE, 2002.
- [PH02] G. Piella and H.J.A.M. Heijmans. Adaptive lifting schemes with perfect reconstruction. *Signal Processing, IEEE Transactions on*, 50(7) :1620–1630, 2002.
- [PKG06] M. Pauly, L.P. Kobbelt, and M. Gross. Point-based multiscale surface representation. *ACM Transactions on Graphics (TOG)*, 25(2) :177–193, 2006.
- [Pl09] G. Plonka. The easy path wavelet transform : A new adaptive wavelet transform for sparse representation of two-dimensional data. *Multiscale Modeling & Simulation*, 7(3) :1474–1496, 2009.
- [Plu] Photoshop Image Portraiture Plugin. <http://imagenomic.com/pt.aspx>.
- [PM90] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7) :629–639, 1990.
- [REC11] I. Ram, M. Elad, and I. Cohen. Generalized tree-based wavelet transform. *Signal Processing, IEEE Transactions on*, 59(9) :4199–4209, 2011.
- [REC12] I. Ram, M. Elad, and I. Cohen. Redundant wavelets on graphs and high dimensional data clouds. *Signal Processing Letters, IEEE*, 19(5) :291–294, 2012.
- [RJW96] Z. Rahman, D.J. Jobson, and G.A. Woodell. Multi-scale retinex for color image enhancement. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 1003–1006. IEEE, 1996.

- [Roc96] R.T. Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1996.
- [ROF92] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4) :259–268, 1992.
- [RS00] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500) :2323–2326, 2000.
- [SAC06] David M. Strong, Jean-Francois Aujol, and Tony F. Chan. Scale recognition, regularization parameter selection, and meyer’s  $g$  norm in total variation regularization. *Multiscale Modeling & Simulation*, 5(1) :273–303, 2006.
- [SB97] S.M. Smith and J.M. Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, 23(1) :45–78, 1997.
- [Sch10] O. Scherzer. *Handbook of mathematical methods in imaging*, volume 1. Springer, 2010.
- [SED04] J.L. Starck, M. Elad, and D. Donoho. Redundant multiscale transforms and their application for morphological component separation. *Advances in Imaging and Electron Physics*, 132 :287–348, 2004.
- [SED05] J.L. Starck, M. Elad, and D.L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *Image Processing, IEEE Transactions on*, 14(10) :1570–1582, 2005.
- [SG06] O. Scherzer and C. Groetsch. Inverse scale space theory for inverse problems. *Scale-Space and Morphology in Computer Vision*, pages 317–325, 2006.
- [Sha08] A. Shamir. A survey on mesh segmentation techniques. In *Computer graphics forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008.
- [She05] J.J. Shen. Piecewise  $h^{-1} + h^0 + h^1$  images and the mumford-shah-sobolevmodel for segmented image decomposition. *Applied Mathematics Research Express*, 2005(4) :143–167, 2005.
- [She10] G. Shen. *LIFTING TRANSFORMS ON GRAPHS : THEORY AND APPLICATIONS*. PhD thesis, UNIVERSITY OF SOUTHERN CALIFORNIA, 2010.

- [SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8) :888–905, 2000.
- [SMF10] J.L. Starck, F. Murtagh, and J.M. Fadili. *Sparse image and signal processing : wavelets, curvelets, morphological diversity*. Cambridge University Press, 2010.
- [SNF<sup>+</sup>12] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. Signal processing on graphs : Extending high-dimensional data analysis to networks and other irregular data domains. *CoRR*, abs/1211.0053, 2012.
- [SO08] G. Shen and A. Ortega. Optimized distributed 2d transforms for irregularly sampled sensor network grids using wavelet lifting. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 2513–2516. IEEE, 2008.
- [Spi07] D.A. Spielman. Spectral graph theory and its applications. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 29–38. IEEE, 2007.
- [SRV12] D.I. Shuman, B. Ricaud, and P. Vandergheynst. A windowed graph fourier transform. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 133–136. IEEE, 2012.
- [SS95] P. Schröder and W. Sweldens. Spherical wavelets : Efficiently representing functions on the sphere. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 161–172. ACM, 1995.
- [SSD09] Kartic Subr, Cyril Soler, and Frédo Durand. Edge-preserving multiscale image decomposition based on local extrema. *ACM Trans. Graph.*, 28(5), 2009.
- [SSN09] A. Singer, Y. Shkolnisky, and B. Nadler. Diffusion interpretation of nonlocal neighborhood filters for signal denoising. *SIAM Journal on Imaging Sciences*, 2(1) :118–139, 2009.
- [SWAC12] J. Salmon, R. Willett, and E. Arias-Castro. A two-stage denoising filter : the preprocessed yaroslavsky filter. *arXiv preprint arXiv :1208.6516*, 2012.
- [Swe96] W. Sweldens. The lifting scheme : A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2) :186, 1996.



- [Swe98] W. Sweldens. The lifting scheme : A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2) :511–546, 1998.
- [TA] AN Tikhonov and V.Y. Arsenin. Solutions of ill-posed problems. 1977. *WH Winston, Washington, DC*.
- [TA09] E. Tadmor and P. Athavale. Multiscale image representation using novel integro-differential equations. *Inverse problems and imaging*, 35(4) :693, 2009.
- [TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.
- [TNV04] E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using hierarchical (BV,L2) decompositions. *Multiscale Modeling & Simulation*, 2(4) :554–579, 2004.
- [Tsc06] D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving pde’s. *International Journal of Computer Vision*, 68(1) :65–82, 2006.
- [TT99] J. Tumblin and G. Turk. Lcis : A boundary hierarchy for detail-preserving contrast reduction. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 83–90. ACM Press/Addison-Wesley Publishing Co., 1999.
- [Vek99] O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. PhD thesis, Cornell University, 1999.
- [VK95] M. Vetterli and J. Kovačević. *Wavelets and subband coding*. Prentice Hall PTR Upper Saddle River, NJ, 1995.
- [vL07] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17 :395–416, 2007.
- [WCBD05] R. Wagner, H. Choi, R. Baraniuk, and V. Delouille. Distributed wavelet transform for irregular sensor network grids. In *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*, pages 1196–1201. IEEE, 2005.
- [Wei08] P. Weiss. *Algorithmes rapides d’optimisation convexe. Applications à la reconstruction d’images et à la détection de changements*. PhD thesis, Université de Nice Sophia-Antipolis, 2008.
- [Wit84] A. Witkin. Scale-space filtering : A new approach to multi-scale description. In *Acoustics, Speech, and Signal Processing, IEEE*

- International Conference on ICASSP'84.*, volume 9, pages 150–153. IEEE, 1984.
- [WL11] J. Wang and B.J. Lucier. Error bounds for finite-difference methods for rudin-osher-fatemi image smoothing. *SIAM Journal on Numerical Analysis*, 49(2) :845–868, 2011.
- [WSB05] R. Wagner, S. Sarvotham, and R. Baraniuk. A multiscale data representation for distributed sensor networks. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 4, pages iv–549. IEEE, 2005.
- [YE96] L.P. Yaroslavsky and M. Eden. *Fundamentals of Digital Optics : Digital Signal Processing in Optics and Holography*. Springer-Verlag New York, Inc., 1996.
- [ZBL<sup>+</sup>04] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16 :321–328, 2004.
- [ZS04] D. Zhou and B. Schölkopf. A regularization framework for learning from graph data. 2004.
- [ZS05] Dengyong Zhou and Bernhard Schölkopf. Regularization on discrete spaces. In *Pattern Recognition*, volume 3663 of *LNCS*, pages 361–368. 2005.

# Décompositions multi-échelles de données définies sur des graphes.

## Résumé

Cette thèse traite d'approches permettant la construction de décompositions multi-échelles de signaux définis sur des graphes pondérés généraux. Ce manuscrit traite de trois approches que nous avons développées.

La première approche est basée sur un procédé variationnel itératif et hiérarchique et généralise la décomposition structure-texture, proposée initialement pour les images. Deux versions sont proposées : l'une basée sur un a priori quadratique et l'autre sur un a priori de type variation totale. L'étude de la convergence est effectuée et le choix des paramètres discuté dans chaque cas. Nous détaillons l'application des décompositions que nous obtenons au rehaussement de détails dans les images et les modèles 3D.

La deuxième approche fournit une analyse multirésolution de l'espace des signaux sur un graphe donné. Cette construction repose sur l'organisation du graphe sous la forme d'une hiérarchie de partitions. Nous avons développé un algorithme permettant la construction adaptative de telles hiérarchies.

Enfin, dans la troisième approche, nous adaptons le schéma de lifting à des signaux sur graphes. Cette adaptation pose divers problèmes pratiques. Nous nous sommes intéressés d'une part à l'étape de sous-échantillonnage, pour laquelle nous avons adopté une approche gloutonne, et d'autre part à l'itération de la transformée sur des sous-graphes induits.

**Mots-clés : Analyse multi-échelle, Optimisation convexe, Laplacien, Ondelettes, Traitement d'images.**

## Multiscale Decompositions of Data Defined on Graphs

### Abstract

This thesis is concerned with approaches to the construction of multiscale decompositions of signals defined on general weighted graphs. This manuscript discusses three approaches that we have developed.

The first approach is based on a variational and iterative process. It generalizes the structure-texture decomposition, originally proposed for images. Two versions are proposed: one is based on a quadratic prior while the other is based on a total variation prior. The study of the convergence is performed and the choice of parameters discussed in each case. We describe the application of the decompositions we get to the enhancement of details in images and 3D models.

The second approach provides a multiresolution analysis of the space of signals on a given graph. This construction is based on the organization of the graph as a hierarchy of partitions. We have developed an adaptive algorithm for the construction of such hierarchies.

Finally, in the third approach, we adapt the lifting scheme to signals on graphs. This adaptation raises a number of practical problems. We focused on the one hand on the subsampling step for which we adopted a greedy approach, and on the other hand on the iteration of the transform on induced subgraphs.

**Keywords: Laplacian, Wavelets (Mathematics), Image processing.**

**Université de Caen Basse-Normandie : Esplanade de la Paix 14032 CAEN  
CEDEX**

**Groupe de Recherche en Informatique, Image, Automatique et Instrumentation  
de Caen : 6 Boulevard du Maréchal Juin 14050 CAEN CEDEX**

