



ENSC-2006 N°95

**THESE DE DOCTORAT
DE L'ECOLE NORMALE SUPERIEURE DE CACHAN**

présentée par
Monsieur **Jérôme GILLES**

pour obtenir le grade de
**DOCTEUR DE L'ECOLE NORMALE SUPERIEURE DE
CACHAN**

Domaine :
Mathématiques

Sujet de la thèse :
**Décomposition et détection de structures
géométriques en imagerie**

Thèse présentée et soutenue le 22 juin 2006 devant le jury composé de :

M. Jean-Michel MOREL	Professeur	Président
M. Henri MAÎTRE	Professeur	Rapporteur
M. Gilles AUBERT	Professeur	Rapporteur
M. Frédéric PRADEILLES	Docteur	Examinateur
M. Yves MEYER	Professeur Emérite	Directeur de thèse
M. Bertrand COLLIN	Docteur	Co-directeur de thèse

CENTRE DE MATHÉMATIQUES ET DE LEURS APPLICATIONS
ENS CACHAN/CNRS/UMR 8536
61, Avenue du Président Wilson, 94235 CACHAN Cedex (FRANCE)

A mon Père ...

Table des matières

Remerciements	9
Introduction	11
1 Le bruit	25
1.1 Les ondelettes	26
1.1.1 Cas continu	27
1.1.2 Cas discret	28
1.1.3 Analyse multirésolution	28
1.1.4 Espaces de Besov	30
1.1.5 Pouvoir d'approximation	30
1.2 La transformée <i>ridgelet</i>	31
1.2.1 Définition	31
1.2.2 Transformée <i>ridgelet</i> et transformée de Radon	32
1.2.3 Espace de <i>ridgelet</i> - $R_{p,q}^s$	32
1.2.4 Remarques - Extensions	34
1.3 La transformée <i>curvelet</i>	35
1.3.1 Construction de la transformée	35
1.3.2 Aspects théoriques	37
1.4 La transformée <i>contourlet</i>	38
1.4.1 La décomposition pyramidale laplacienne (LP)	38
1.4.2 Filtrage directionnel	42
1.4.3 La transformée <i>contourlet</i>	42
1.4.4 Espace de <i>contourlets</i>	46
1.5 Débruitage d'image	47
1.5.1 Débruitage par seuillage de coefficients	47
1.5.2 Débruitage à base de variation totale	49
1.6 Mesure de la qualité du bruit extrait	50
1.6.1 Comparaison des différents algorithmes de débruitage	54
1.7 Bilan	57

2	L'analyse de texture	59
2.1	Quelques généralités.	59
2.2	Textures et espaces de fonctions oscillantes.	60
2.2.1	Au départ : l'algorithme de Rudin, Osher et Fatemi.	60
2.2.2	L'algorithme de Meyer.	62
2.2.3	Propriétés de l'algorithme de Meyer	64
2.2.4	L'algorithme de Osher-Vese.	72
2.2.5	L'algorithme de Aujol	74
2.3	Autres espaces fonctionnels	78
2.3.1	Algorithme de Aujol et Chambolle	78
2.3.2	L'algorithme de A.Haddad	79
2.4	Evaluation de la décomposition	82
2.5	Bilan	83
3	Décomposition u, v, w	85
3.1	Décomposition u, v, w adaptative	86
3.1.1	Algorithme numérique.	89
3.2	Algorithme de Aujol-Chambolle	91
3.3	Comparaison entre les deux modèles.	94
3.4	«Unification» des deux modèles.	94
3.5	Utilisation des <i>contourlets</i>	100
3.6	Evaluation des algorithmes	103
3.7	Bilan	105
4	Applications	107
4.1	Débruitage de bruit de type <i>speckle</i>	107
4.1.1	Principe	107
4.1.2	Tests et résultats	108
4.2	Détection d'objets longilignes	115
4.2.1	Une propriété inattendue!	115
4.2.2	Algorithme de détection des alignements.	117
4.2.3	Adaptation à la géométrie réelle	133
4.2.4	Bilan	133
	Conclusion	139
A	Méthode de projection non-linéaire	143
A.1	Notations - Définitions préliminaires	143
A.2	Variation totale	144
A.3	Algorithme	145

B	<i>Snake</i> probabiliste	149
B.1	Les premiers modèles	149
B.2	Formulation par <i>levelset</i>	150
B.2.1	La méthode des <i>levelsets</i>	150
B.2.2	Reformulation des <i>snakes</i>	152
B.3	Contour Actif Statistique Polygonal (CASP)	154
B.3.1	Modèle	155
B.3.2	Cas gaussien	156
B.3.3	Cas général pour la famille exponentielle	157
B.3.4	Algorithmes d'optimisation	157
B.4	<i>Snake</i> probabiliste	157
	Bibliographie	162

Remerciements

Cette thèse s'est déroulée en majeure partie au département Géographie-Imagerie-Perception (GIP) du Centre d'Expertise Parisien de la DGA à Arcueil et en collaboration avec le Centre de Mathématiques et de Leurs Applications (CMLA) de l'Ecole Normale Supérieure de Cachan.

Je tiens tout d'abord à remercier Yves Meyer d'avoir accepté d'encadrer cette thèse ainsi que pour son enthousiasme, permanente source de motivation. Je le remercie aussi vivement pour son excellente pédagogie qui restera un modèle pour moi.

Je remercie ensuite Bertrand Collin qui a co-dirigé cette thèse et qui a su m'inculquer le point de vue «numérique» du traitement de l'image. Son profil de physicien m'a poussé à m'interroger sur l'interprétation des modélisations mathématiques abordées dans cette thèse. Je le remercie aussi pour ses critiques «pédagogiques» lors de la préparation de ma soutenance.

Je remercie Messieurs Henri Maître et Gilles Aubert d'avoir accepté de relire ce manuscrit et pour les remarques qui m'ont été faites.

Je remercie aussi Jean-Michel Morel d'avoir accepté de présider mon jury. Même si nos rencontres durant ces années de thèse ont été peu nombreuses, celles-ci ont toujours été sources d'inspiration dans mon travail et je l'en remercie.

Je remercie également Frédéric Pradeilles, d'une part pour m'avoir permis de démarrer cette thèse dans les meilleures conditions possibles lorsqu'il était chef du département GIP, d'autre part pour avoir accepté de faire partie de ce jury en tant qu'examinateur.

Je tiens également à remercier l'ensemble de mes collègues du département GIP pour leur bonne humeur et leurs compétences diverses et variées en capteurs optroniques, systèmes de vision, traitement d'image qui m'ont été utiles lors de cette thèse. Enfin, je souhaite bon courage aux actuels thésards qui commencent à entrevoir le bout du tunnel !

Introduction

Les progrès de l'informatique ces dernières décennies et notamment la disponibilité d'une puissance de calcul de plus en plus accrue ont fait connaître au traitement d'image un développement fulgurant. A ce jour, le traitement d'image trouve des applications dans tous les domaines ; voici quelques exemples courants (cette liste n'étant pas exhaustive) :

- le monde du multimédia : à «l'heure du numérique», le grand public est équipé d'appareils photo ou vidéo et souhaite disposer de logiciels permettant de traiter toutes ces images (amélioration de la qualité, transformation, extraction de certains objets,...). La télévision numérique fait appel aussi à son lot de traitements spécifiques notamment en compression vidéo,...
- le monde industriel : le contrôle de qualité de pièces en sortie de chaîne de production est un travail fastidieux. Des solutions analysant les formes présentes dans une prise de vue des pièces permettent d'automatiser ce type de tâche. Le contrôle non destructif cherche à détecter des défauts à l'intérieur de structures ou objets, l'analyse d'images de type radiographiques permet d'obtenir des outils adaptés à cette problématique. L'industrie de l'automobile réfléchit à incorporer des systèmes de vision nocturne pour l'aide à la conduite,
- le monde médical : scanner, doppler, IRM, échographie, ... autant de procédés d'acquisition d'images dont le but est l'aide au diagnostic du médecin. Ces différents types d'imageurs nécessitent des traitements spécifiques pour permettre l'extraction d'information facilement interprétable,
- le monde de la Défense et de la sécurité : les applications dans ce domaine sont très nombreuses. Cette thèse s'étant déroulée dans ce contexte, nous allons en décrire dans le détail les différentes problématiques, celles-ci ayant très largement inspiré les travaux de recherche de cette thèse.

Les problématiques du traitement d'image dans le domaine militaire

Cette thèse a été réalisée dans le cadre du poste que j'occupe au sein de la Délégation Générale pour l'Armement (DGA) et plus précisément au département Géographie Imagerie Perception (GIP). L'une des missions de ce département est de conduire les différentes études sur les systèmes de perception potentiellement intéressants d'un point de vue militaire. Nous pouvons voir deux branches parallèles, mais finalement pas si éloignées l'une de l'autre : d'une part les aspects capteurs optroniques et leurs traitements associés, d'autre part les aspects traitements «automatiques» (détection de cible, reconnaissance de formes...).

Aspects capteurs optroniques et traitements associés

Un vieux rêve des opérationnels était d'acquiescer la possibilité de voir la nuit. Ce fut chose faite avec l'apparition des capteurs de type intensificateurs de lumière et des capteurs de type infrarouges¹. Ces derniers permettent de visualiser d'autres bandes spectrales complémentaires du domaine visible (comme par exemple la diffusion de chaleur par les êtres vivants ou les moteurs de véhicules), mais induisant par là même une nouvelle préhension des scènes observées. Ce type de capteur possède aussi ses défauts spécifiques dont les deux principaux sont :

- la non-uniformité des capteurs : un capteur est constitué de plusieurs pixels, ceux-ci n'ont pas tous exactement la même réponse (conversion photons / niveau de gris dans l'image). Ce phénomène est très nettement visible dans le domaine infrarouge, la figure 1 en illustre le principe. La bande de gauche correspond à un signal de type rampe. Si tous les pixels avait la même réponse, alors une ligne horizontale dans cette rampe devrait avoir les mêmes niveaux de gris dans l'image. Or on voit très clairement que ce n'est pas le cas ; on observe ici la non-uniformité des réponses des différents pixels du capteur. Ce phénomène se retrouve aussi dans la partie de droite censée être totalement uniforme et sur laquelle on observe une sorte de «texture». Ce problème peut être gênant dans la perception des objets de la scène observée. Des traitements de correction des non-uniformités sont nécessaires afin de corriger ces défauts.
- l'égalisation des contrastes dans une image : la présence d'un objet très chaud dans une scène vue en infrarouge va considérablement impacter la dynamique de l'image. Ainsi des détails de la scène peuvent ne plus être

¹les bandes infrarouges classiquement utilisées sont : le proche infrarouge ($0.8\mu m - 2\mu m$), la bande 2 ($3\mu m - 5\mu m$) et la bande 3 ($8\mu m - 12\mu m$)

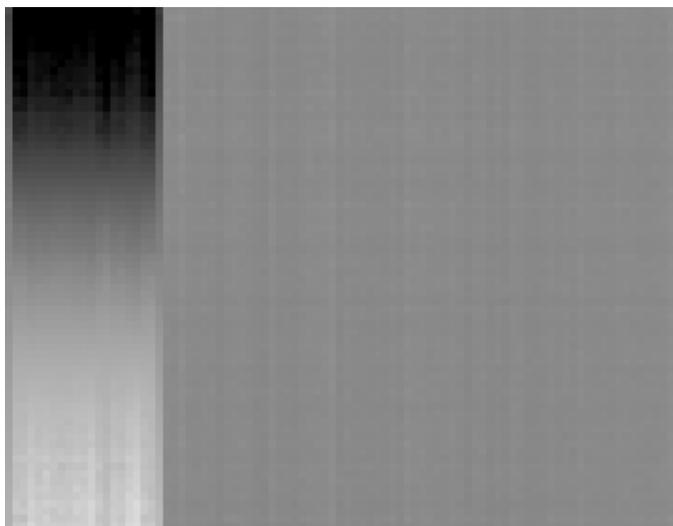


FIG. 1 – Non-uniformité des capteurs en imagerie infrarouge.

visibles malgré leur présence dans l'image. La figure 2 illustre ce comportement. L'image en haut à gauche montre le résultat obtenu d'une égalisation d'histogramme effectuée en tenant compte de la dynamique contenue dans la zone 1. En haut à droite le même type de traitement en tenant compte de la dynamique de la zone 2 et en bas à gauche en tenant compte de la zone 3. Enfin, l'image en bas à droite montre quant à elle le résultat obtenu grâce à un traitement adapté se basant sur une information locale de l'image. Les trois premiers cas ne permettent pas de voir l'ensemble de l'information contenue dans l'image. Le traitement adaptatif donnant l'image en bas à droite permet quant à lui d'obtenir pleinement l'ensemble des informations contenues dans la scène visualisée. Les fantassins et le véhicule blindé sont très clairement visibles mais aussi le pylône et les câbles électriques qu'il est indispensable de faire apparaître à l'opérateur humain si par exemple le système sert à la navigation de nuit d'un pilote d'hélicoptère.

Un autre type d'imagerie maintenant classiquement utilisée dans le domaine militaire est l'imagerie de type SAR² permettant de faire apparaître d'autres informations (comme certaines propriétés des matériaux des objets présents dans la scène). Ce type d'image est généralement très bruité (voir un exemple figure 3), ce qui introduit la nécessité de pouvoir disposer d'algorithmes de débruitage voire plus généralement de restauration d'image. Ce type d'image peut aussi être utilisé pour faire de la reconstruction 3D et réclame donc des algorithmes particuliers.

² *Synthetic Aperture Radar* (Radar à Ouverture Synthétique)

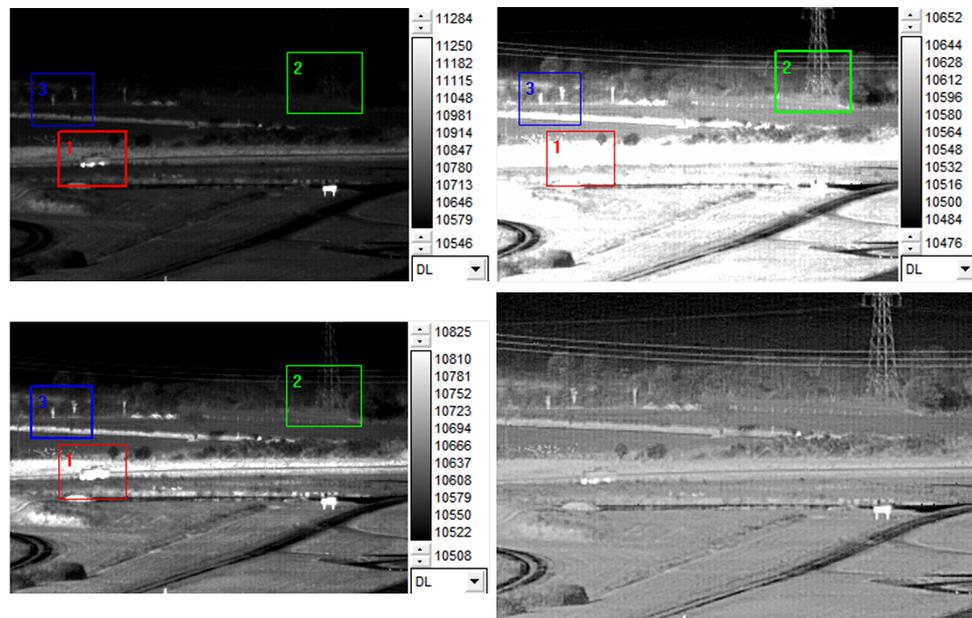


FIG. 2 – Illustration de la problématique de l'égalisation de l'histogramme (voir le texte pour les explications).

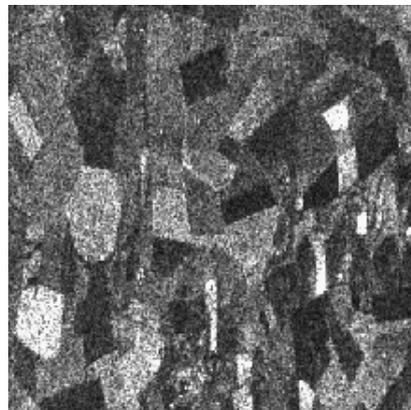


FIG. 3 – Exemple d'image SAR.

Plus récemment, de nouveaux types d'imagers ont fait leur apparition. Tout d'abord, les imageurs actifs qui utilisent un faisceau laser (qui peut être dans différentes bandes spectrales) éclairant la scène et se réfléchissant sur les objets pour être ensuite capté et former une image (voir le schéma figure 4). Outre la possible présence de bruit dans les images suivant les conditions d'acquisition, nous sommes confrontés à un autre problème qui est la dé-

formation des objets. Ce phénomène est essentiellement dû aux turbulences atmosphériques qui viennent modifier le trajet du faisceau laser. Cette déformation est gênante si l'on souhaite faire de la reconnaissance d'objets (voir la figure 5). Ici aussi il est nécessaire de développer des algorithmes permettant de corriger l'influence de ces turbulences. Pour cela, des modèles de propagation du faisceau laser en atmosphère turbulente sont actuellement à l'étude. Ils devraient permettre de comprendre le phénomène intervenant dans la formation de l'image et ainsi nous guider dans le développement de méthodes de restauration d'image spécifiques. On notera que cette problématique de l'effet des turbulences se retrouve aussi en imagerie passive et fait actuellement l'objet d'une thèse au sein du département (je remercie Magalie Lemaitre [49] de m'avoir fourni les images de la figure 6 présentant un exemple d'image déformée en infrarouge ainsi qu'une version restaurée obtenue par une méthode de filtrage adaptatif)

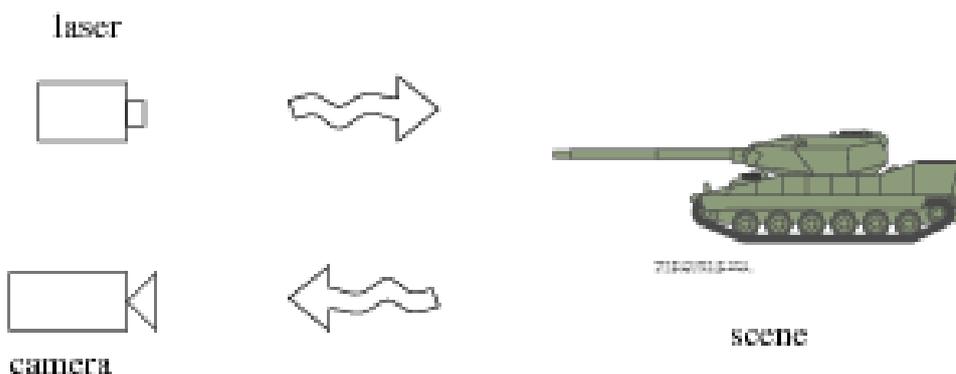


FIG. 4 – Schéma de principe de l'imagerie active.

D'autres types de systèmes optroniques actuellement en plein essor sont les systèmes multispectraux ou hyperspectraux. Ce type d'imageur fournit une visualisation d'une même scène dans N bandes spectrales différentes. La figure 7 donne un exemple d'images obtenues grâce à ce type d'imageur.

Chaque matériau a une réponse différente suivant la bande spectrale dans laquelle il est observé. Cette décomposition d'une même scène en N bandes spectrales distinctes doit permettre de réaliser une classification des matériaux présents dans la scène en vue de faire de la reconnaissance et de l'identification d'objets. Des travaux sont actuellement en cours tout d'abord dans le but d'identifier les bandes spectrales réellement intéressantes. Par ailleurs, il est aussi nécessaire d'étudier les différentes méthodes de classification existantes afin de trouver la mieux adaptée à ce type de données. Une autre application pouvant être envisagée serait de fusionner ces différentes informations en une seule et même image contenant un maximum d'informations



FIG. 5 – Exemple d'image acquise en imagerie active laser.

sur la scène et pouvant être visuellement compréhensible par un opérateur humain.

Traitements automatiques

Nous venons de faire une première allusion, ci-dessus, à la notion de traitement automatique en parlant de méthodes de classification dans le cadre de l'imagerie multispectrale. Le besoin de traitements automatiques ou semi-automatique est de plus en plus prépondérant. Un opérateur humain a maintenant à sa disposition plusieurs systèmes. Or il ne peut pas analyser d'une manière optimale l'ensemble des informations lui parvenant et ceci d'autant plus si il est dans une situation de stress (en situation de combat par exemple). D'autre part, les systèmes actuels permettent d'obtenir des informations, parfois de natures très différentes, en grande quantité, ce qui amène à se poser la question de savoir comment interpréter ces données proposant chacune une perception spécifique de la même scène. Par ailleurs, les temps de traitement pouvant être longs et fastidieux pour un opérateur humain, il est alors intéressant de disposer de traitements automatiques afin d'exploiter ces données.

Concernant les problématiques intéressant la Défense, les besoins concernent majoritairement les notions de Détection, Reconnaissance et Identification (DRI). Examinons cette chaîne dans l'ordre :

- Détection : les différents types d'imageurs fournissent chacun leur vision d'une scène. La première chose est de détecter les cibles³. Si quelques types

³On appellera cible tout ce qui ne concerne pas les éléments naturels d'une scène : véhicules, personnes, objets manufacturés,...

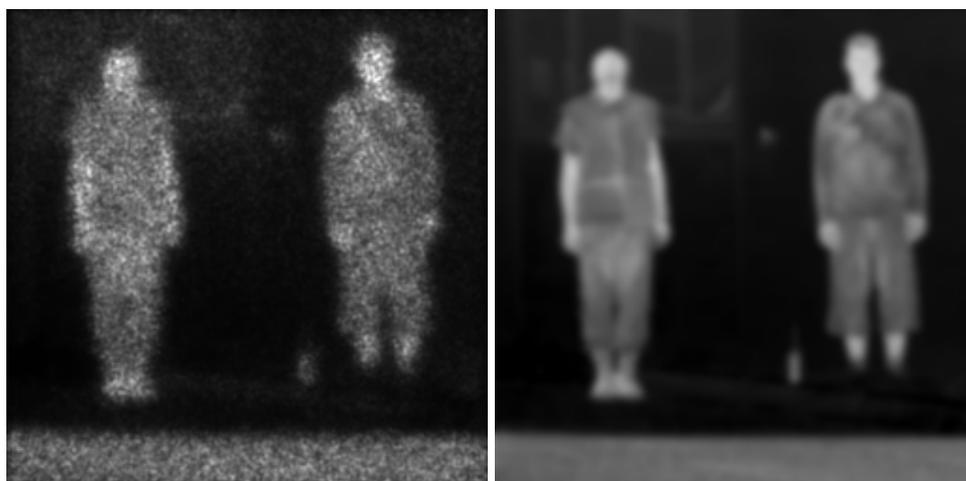


FIG. 6 – Exemple de turbulences en imagerie infrarouge (l'image dégradée à gauche et sa version restaurée à droite).

d'imageurs permettent assez aisément la détection de certaines cibles (en imagerie thermique, un véhicule ayant des parties chaudes générera des «points chauds» qu'il est souvent possible de détecter à l'aide de techniques de seuillage par exemple), d'autres ne sont pas aussi catégoriques. Par ailleurs les méthodes de camouflage font tous les jours des progrès, ce qui rend la tâche plus difficile. Prenons l'exemple de la figure 8 contenant quatre cibles qui, si certaines sont détectables assez facilement à l'œil, d'autres le sont nettement moins (à la fin de cette introduction la figure 10 donne la position des cibles). Ici nous sommes confrontés à une analyse du contenu de l'image et en particulier à l'analyse de texture dans ce cas (et ce sera très souvent le cas d'une manière générale sur ce type de problématique). Même si la littérature regorge d'articles abordant l'analyse ou la classification des textures dans une image, le fait est que la plupart d'entre eux le font sur des images de test créées par leurs auteurs ou étant à la disposition de la communauté. Malheureusement la plupart de ces algorithmes voient leur taux de réussite chuter si on les utilise sur des images de scènes réelles contenant des textures de nature assez proches les unes des autres.

- Reconnaissance : le but ici est de pouvoir dire si la cible précédemment détectée est une jeep, un véhicule type blindé à roues, type blindé à chenilles, un fantassin, ... Les algorithmes visés ici sont des algorithmes de reconnaissance des formes. La première étape de ce type d'algorithme est généralement une étape de segmentation de la cible afin d'en extraire les contours. Ensuite la forme de la cible doit être «reconnue» ; cela sous-

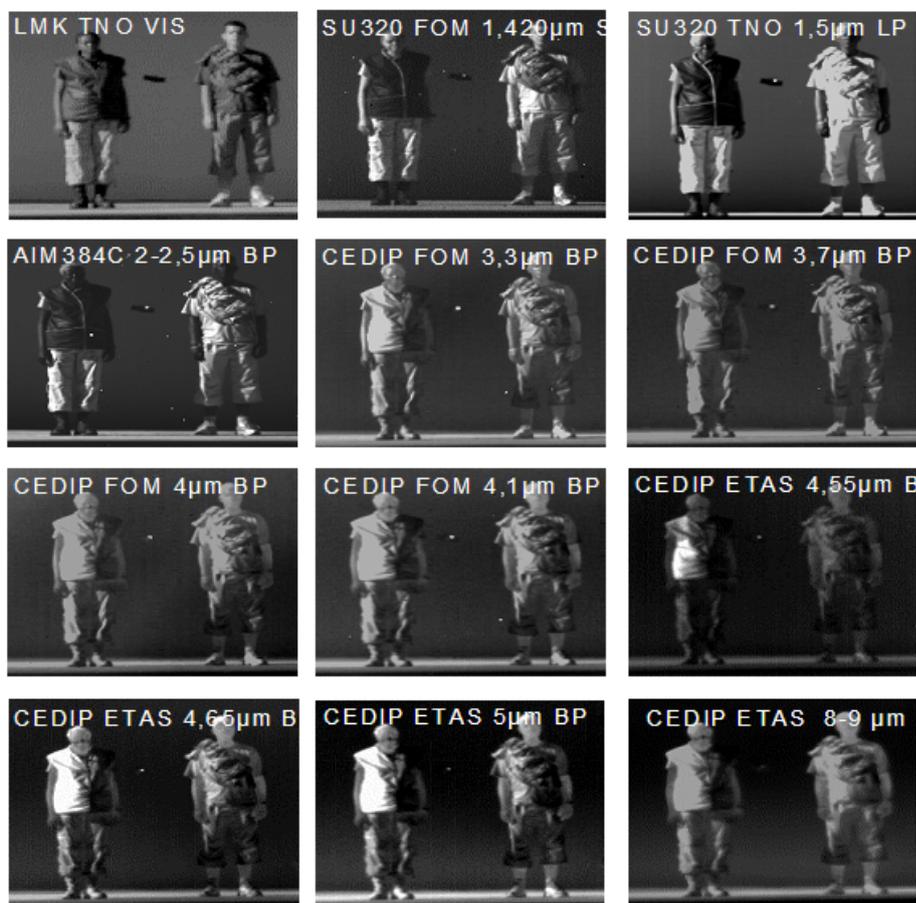


FIG. 7 – Résultats obtenus en imagerie multispectrale.

entend généralement qu'une base de donnée de formes admissibles est disponible et que des algorithmes de comparaison de formes vont retenir celle la plus probable correspondant à la cible détectée. La figure 9 donne un exemple de segmentation obtenue en imagerie infrarouge à l'aide d'un modèle de contour actif statistique [41] que nous avons proposé et dont nous rappelons le principe en annexe B.

- Identification : le but final est de pouvoir connaître le "modèle" de la cible. La première information primordiale est de savoir si la cible est amie ou ennemie. Ensuite vient la nécessité de caractériser la cible (comme identifier le modèle de char par exemple) afin de pouvoir choisir la méthode adéquate pour éventuellement la neutraliser. Cette problématique de l'identification est souvent basée sur la notion de signature. Ceci sous-entend que l'on doit être capable d'extraire certaines primitives et de pouvoir ensuite les



FIG. 8 – Exemple de scène en imagerie visible contenant des cibles.

comparer à celles contenues dans une base de données afin d'obtenir le résultat souhaité.

Les besoins mentionnés ci-avant se basent sur des scènes ou des séquences d'images d'une scène fixe. Le besoin de faire du suivi de cible en mouvement est aussi indispensable. Ce type de problème impose de s'intéresser à des algorithmes de détection de mouvement et aussi de suivi de changement de forme des cibles. Par exemple, dans une application de vidéo surveillance, on s'intéresse au comportement d'une personne afin de savoir si elle a une attitude menaçante ou non. Des algorithmes du type modèles déformables (ou *snake*) semblent très prometteur pour réaliser ce type de tâche car ces «objets» peuvent se mouvoir et se déformer au gré du comportement de la cible dans la séquence d'images.

Certaines autres problématiques attenantes ou complémentaires ont fait ou font leur apparition. Nous n'en citerons que quelques unes : reconnaissance de caractères (pour l'identification de plaque d'immatriculation, recherche de mots-clés dans des documents écrits...), la détection et reconnaissance de visages (pour l'identification de personnes), le recalage d'images, le comptage

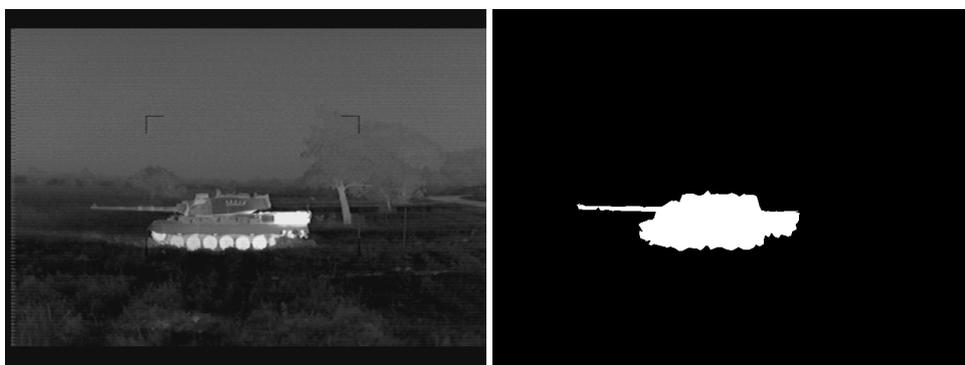


FIG. 9 – Extraction de la forme d'un char en imagerie infrarouge par contour actif statistique.

d'individus dans une foule,...

Un autre domaine utilisant la vision par ordinateur est la robotique. Un robot a besoin de percevoir son environnement d'une part pour se déplacer sans risque, d'autre part en vue d'explorer un lieu inconnu. Outre des capteurs du type télémètre (à ultrasons ou laser), les robots sont aujourd'hui traditionnellement équipés de caméras. Il est donc nécessaire de disposer d'algorithmes capables d'apprécier le contenu sémantique d'une scène.

Un autre aspect étudié au département GIP est la problématique de l'évaluation des algorithmes de traitement d'images. En effet l'une des missions du département au sein de la DGA est d'évaluer les performances de systèmes. La définition de métriques adaptées est nécessaire et doit être représentative des objectifs traités par les algorithmes. Nous abordons cette notion d'évaluation au cours de cette thèse pour comparer les différents algorithmes mis en jeu.

Objectifs de cette thèse

Comme nous l'avons vu dans la section précédente, la diversité des problèmes abordés pour des besoins de Défense sont nombreux et touchent toutes les branches de la recherche en matière de traitement des images. Même si certains outils mathématiques sont utilisés abondamment dans différentes problématiques, il n'existe pas de modèle visant à représenter d'un point de vue mathématique une image. Pourtant l'idée de posséder un modèle «générique» semble séduisante, car des algorithmes spécifiques pourraient sûrement être développés pour essayer de résoudre de manière adaptée différents types de problèmes.

De récents travaux en analyse fonctionnelle permettent de mettre en place un certain nombre d'outils en vue de créer un modèle d'image. L'idée de base est de décomposer l'image f en deux composantes : l'une (notée u) comprenant les structures de l'image et l'autre (notée v) les textures sous l'hypothèse $f = u + v$. Dans cette thèse, nous nous intéressons à cette notion de décomposition et étendons cette modélisation au cas trois composantes en rajoutant la notion de bruit dans le modèle (en effet des images issues directement d'un capteur sont naturellement bruitées, cette composante doit donc être traitée dans cette modélisation).

Dans le premier chapitre, nous rappelons quelques résultats d'analyse multi-résolution par ondelettes et leurs extensions comme les *ridgelets*, les *curvelets* et les *contourlets* (permettant de tenir compte de la géométrie présente dans les images) dans le but de faire du débruitage. Le choix de ce type d'outils s'est fait naturellement ; en effet, plus on diminue la résolution d'une image et plus la présence du bruit sera faible. Nous rappelons les propriétés d'approximation de ce type de fonctions ainsi que les définitions des espaces de Besov et des espaces de *ridgelets*. Nous proposons une définition des espaces de *contourlets* et de la norme associée. Pour chacun de ses outils, les résultats de débruitage par seuillage des coefficients sont rappelés et une comparaison en terme de théorie de l'approximation sera donnée. L'algorithme de Rudin-Osher-Fatemi basé sur la variation totale est aussi utilisé pour effectuer du débruitage. Enfin, nous introduisons une méthode utilisant la fonction d'autocorrélation pour nous permettre de juger de la «qualité» du bruit extrait. Cette méthode nous permettra de comparer les différents algorithmes entre eux en prenant un point de vue de décomposition et non seulement de débruitage.

Dans le deuxième chapitre, nous abordons plus directement la notion de décomposition et la modélisation des textures. L'algorithme de Rudin, Osher et Fatemi (ROF) rappelé dans le premier chapitre est le point de départ de la modélisation des textures par des fonctions oscillantes proposée par Yves Meyer [54]. Les propriétés et la mise en œuvre de cette nouvelle modélisation sont examinées notamment grâce à deux algorithmes : l'algorithme de Osher-Vese basé sur un système d'équations aux dérivées partielles couplées et l'algorithme de Jean-François Aujol quant à lui basé sur une méthode de projection non-linéaire. Le dernier modèle présenté est celui récemment proposé par Ali Haddad dans ses travaux de thèse [43]. Des espaces de Besov bien choisis sont utilisés pour modéliser les différentes composantes de l'image. Nous terminons ce chapitre en évaluant les performances des différents algorithmes de décomposition en l'erreur commise sur les composantes objets et textures lors de la décomposition appliquée sur une image synthétisée par nos soins (nous générons séparément une composante objet et une autre texture que nous assemblons au final).

Le troisième chapitre étudie le cas des images bruitées. Pour cela nous introduisons un nouvel algorithme permettant de décomposer une image f en trois parties : structures (notée u), textures (notée v) et bruit (notée w) sous l'hypothèse $f = u + v + w$. Pour cela, nous nous appuyons sur les résultats des chapitres précédents et y ajoutons l'idée de régularisation adaptative utilisée par Gilboa [39] pour débruiter des images texturées dans le cadre de l'algorithme ROF. Ce principe nous permet de jouer sur la norme dans l'espace des textures et ainsi séparer textures et bruit. En marge de ce travail, Aujol et Chambolle ont, eux aussi, proposé un algorithme permettant d'obtenir une décomposition en trois parties. Ils se basent sur une modélisation utilisant l'espace de Besov $\dot{B}_{-1,\infty}^\infty$ pour la composante bruit. Nous comparons les résultats en sortie des deux algorithmes et examinons une possible fusion des deux modélisations. Enfin, nous nous intéressons à l'utilisation des espaces de *contourlets* en lieu et place des espaces de Besov afin d'étudier l'apport de la prise en compte de l'information géométrique présente dans l'image. Toujours dans un souci d'évaluation des algorithmes et en suivant la même démarche qu'au deuxième chapitre, nous proposons une mesure des performances de ces différents algorithmes et concluons sur leurs capacités.

Le dernier chapitre, s'intéresse au côté applicatif de ces méthodes de décomposition d'image. Ces techniques étant relativement récentes, très peu d'applications ont été proposées à l'heure actuelle. Nous commençons par examiner le débruitage dans le cas d'un bruit de type multiplicatif en effectuant des tests sur des images de type SAR ou issues d'un imageur actif laser. Ensuite nous étudions une application de détection d'alignements. Pour cela, nous utiliserons une propriété particulière de la décomposition d'image (dont nous fournirons une preuve mathématique) qui servira de prétraitement à une phase de détection. Cette détection se fait en deux étapes. La première, «grossière», s'obtient en appliquant l'algorithme de détection d'alignements par événements ϵ -significatifs. Une deuxième étape dans laquelle les segments détectés sont transformés en contours actifs ce qui nous permettra d'obtenir une forme plus précise. Des tests seront effectués sur des images obtenues à partir d'un capteur aérien, les réseaux routiers pouvant être vus comme un réseau d'alignements.

Compte-tenu du regain d'intérêt pour ce type de technique à base de variation totale par rapport d'autres techniques plus classiques de traitement des textures, nous concluons sur cette notion de modèle pour les images et nous donnerons quelques perspectives aussi bien d'un point de vue théorique que d'un point de vue applicatif. En effet, outre certaines questions sur la modélisation elle-même apparaissant naturellement, cette nouvelle «vision» de l'image fait germer un certain nombre d'idées quant aux applications possibles et ouvre des voies de recherche intéressantes pour le futur.



FIG. 10 – Position des cibles dans la scène de l'image 8.

Chapitre 1

Le bruit

Dans ce chapitre, nous nous intéressons à la composante bruit dans les images. De nombreux algorithmes de débruitage existent dans la littérature utilisant des techniques différentes : filtrage passe-bas, filtrage de Wiener, filtrage anisotropique, variation totale, seuillage dans l'espace des coefficients d'ondelette, ... Dans ces articles, les auteurs s'intéressent généralement à la qualité de l'image débruitée et oublient le bruit extrait. Dans cette thèse, nous faisons le choix de voir le bruit comme étant une composante à part entière de l'image originale. On est donc en droit de se demander quel algorithme extrait un bruit de meilleure qualité ? Nous entendons par là, quelle composante de bruit contiendra un résidu de l'image originale minimum et aura des caractéristiques les plus proches du bruit original ?

Dans cette thèse, notre attention s'est portée sur deux types de techniques : l'approche multirésolution et l'utilisation de la variation totale. Nous avons choisi de travailler avec des outils d'analyse multirésolution parce que il est naturel de constater que le fait de regarder une image à différentes résolutions influe sur la notion de bruit. En effet, le bruit peut être vu comme des détails dans l'image ; aussi en utilisant les coefficients d'une décomposition multirésolution il est possible de traiter le bruit par des méthodes de seuillage des coefficients.

Ce chapitre se propose tout d'abord de rappeler les définitions ainsi que les propriétés des ondelettes. Nous examinerons les nouveaux outils d'analyse multirésolution que sont les *ridgelets*, les *curvelets* (initialement proposées par E.Candès et D.Donoho [15, 16, 37, 17, 18]) et les *contourlets* (définies par M.N.Do et M.Vetterli [33, 64, 35, 30, 32, 31, 29] directement dans le cadre discret). Ces nouvelles techniques cherchent à améliorer le pouvoir d'approximation des ondelettes en incorporant une notion de «directionnalité» ce qui leur permet de mieux tenir compte de la géométrie présente dans l'image (voir la figure 1.1).

Une facette intéressante pour notre démarche de modélisation est l'existence

d'espaces fonctionnels associés aux ondelettes (espaces de Besov B_{pq}^s) Nous verrons que les espace de *ridgelets* ont été définis par E.Candès et nous proposons de la même manière de définir les espaces de *contourlets*, notés CT_{pq}^s , et leur norme associée.

Nous exposerons ensuite l'algorithme de Rudin, Osher et Fatemi [67] basé sur la variation totale. Nous verrons dans le chapitre 2 que cet algorithme a été le point de départ de la modélisation des textures proposée par Y.Meyer.

Nous terminerons ce chapitre en tentant de répondre à la question de la qualité du bruit extrait. Pour cela, nous proposons une méthode basée sur la fonction d'autocorrélation permettant de mesurer la quantité de résidu dans le bruit. Nous donnons alors une comparaison des différentes méthodes de débruitage en prenant ce point de vue.

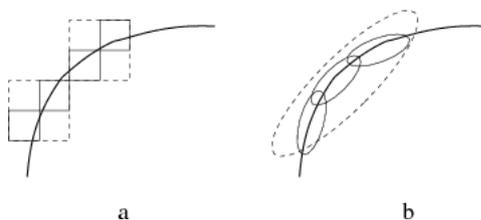


FIG. 1.1 – Fonctions d'approximation dans le cas des ondelettes (a) et dans le cas des *curvelets* (b).

1.1 Les ondelettes

Nous commençons par rappeler les notations et propriétés de l'analyse en ondelettes. La décomposition d'un signal en ondelette est apparue à la fin des années 80 [52, 44, 77]. Les grands acteurs de la formalisation de la théorie des ondelettes sont Y.Meyer, S.Mallat, I.Daubechies, A.Cohen (cette liste n'étant pas exhaustive).

Dans la suite, nous raisonnons sur des signaux 1D mais l'extension aux cas N-D se fait très naturellement par séparabilité de la transformée. L'analyse en ondelette permet de pallier certains "défauts" de l'analyse de Fourier. En effet, celle-ci donne la décomposition d'un signal sur une base de sinus-cosinus, ce qui permet d'obtenir une analyse très bien localisée en fréquence mais pas en temps (les fonctions sinus et cosinus n'étant pas des fonctions localisées). Prenons l'exemple d'un phénomène transitoire. Celui-ci sera décomposé sur l'ensemble du domaine fréquentiel alors que l'information contenue dans ce signal n'intervient qu'à un moment donné (très localisé dans le domaine temporel). Nous voyons facilement, ici, la nécessité d'avoir une représentation qui soit correctement localisée aussi bien en temps qu'en fréquence. La transfor-

mée de Fourier à fenêtre est apparue comme une première réponse à ce problème car elle permet d'obtenir un pavage du domaine temps-fréquence (un pavé étant aussi appelé un atome temps-fréquence). Toutefois, cette transformée n'est pas complètement satisfaisante car elle n'autorise pas un pavage du domaine fréquentiel par des atomes de taille variable. Or dans le cas où l'on souhaite étudier des phénomènes transitoires ayant des durées différentes, il est nécessaire d'avoir des atomes qui soient de taille variable. La transformée en ondelette permet d'obtenir ce type de pavage ; nous en rappelons les principales propriétés.

1.1.1 Cas continu

La transformée en ondelette permet de décomposer un signal sur une famille d'ondelettes translatées et dilatées. Une ondelette est une fonction $\psi \in L^2(\mathbb{R})$ devant répondre à certains critères :

$$\int_{\mathbb{R}} \psi(t) dt = 0 \quad \text{moyenne nulle} \quad (1.1)$$

$$\|\psi\|_2 = 1 \quad \text{normalisation} \quad (1.2)$$

et ψ doit être centrée autour de 0. Une famille d'atomes étant obtenue par dilatation (d'un coefficient a) et translation (d'un coefficient b) de l'ondelette ψ

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right) \quad (1.3)$$

Alors la transformée en ondelette d'une fonction $f \in L^2(\mathbb{R})$ au temps b et à l'échelle a est donnée par

$$\mathcal{WT}_f(a, b) = \langle f, \psi_{a,b} \rangle = \int_{\mathbb{R}} f(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-b}{a} \right) dt \quad (1.4)$$

On peut remarquer que la transformée en ondelette peut s'écrire comme un produit de convolution

$$\mathcal{WT}_f(a, b) = f \star \bar{\psi}_a(b) \quad \text{où} \quad \bar{\psi}_a(b) = \frac{1}{\sqrt{a}} \psi^* \left(\frac{-t}{a} \right) \quad (1.5)$$

Le théorème suivant donne les conditions permettant de reconstruire la fonction f à partir de sa transformée en ondelette.

Théorème 1.1.1 *Soit $\psi \in L^2(\mathbb{R})$ une ondelette réelle vérifiant la condition d'admissibilité suivante*

$$C_\psi = \int_0^{+\infty} \frac{|\hat{\psi}(\xi)|^2}{\xi} d\xi < +\infty \quad (1.6)$$

où $\hat{\psi}$ représente la transformée de Fourier de ψ . Toute fonction $f \in L^2(\mathbb{R})$ vérifie alors

$$f(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{\mathbb{R}} \mathcal{WT}_f(a, b) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) db \frac{da}{a^2} \quad (1.7)$$

et

$$\int_{\mathbb{R}} |f(t)|^2 dt = \frac{1}{C_\psi} \int_0^{+\infty} \int_{\mathbb{R}} |\mathcal{WT}_f(a, b)|^2 db \frac{da}{a^2} \quad (1.8)$$

Une preuve est donnée dans [52].

De nombreux articles sont disponibles dans la littérature concernant le choix de l'ondelette mère ψ . Suivant les applications visées, on cherchera à imposer certaines contraintes à l'ondelette (régularité, taille du support, nombre de moments nuls, ...).

1.1.2 Cas discret

En pratique, nous disposons de signaux numériques composés de N échantillons notés $f[n]$. Soit une ondelette continue $\psi(t)$ ayant pour support $[-K/2, K/2]$ alors l'ondelette discrète dilatée par 2^j est définie par

$$\psi_{jn}[k] = \frac{1}{\sqrt{2^j}} \psi[2^{-j}k - n] \quad (1.9)$$

La transformée en ondelette discrète s'écrit

$$\mathcal{WT}_f[n, j] = \sum_m f[m] \psi_{jn}^*[m] = \langle f, \psi_{jn} \rangle \quad (1.10)$$

et la formule de reconstruction n'est vraie que si ψ vérifie quelques propriétés supplémentaires [52]. On a alors

$$f[m] = \sum_{j=0}^{+\infty} \sum_n \mathcal{WT}_f[n, j] \psi_{jn}[m] \quad (1.11)$$

On voit sur ces relations que l'implémentation numérique de la transformée en ondelette et de sa transformée inverse peut se faire grâce à la mise en place de bancs de filtres définis à partir de ψ .

1.1.3 Analyse multirésolution

On définit une analyse multirésolution de la manière suivante [52] : soit une suite $\{V_j\}_{j \in \mathbb{Z}}$ de sous-espaces fermés de $L^2(\mathbb{R})$. On dira que cette suite est une approximation multirésolution si elle vérifie les six propriétés suivantes :

$$\forall (j, k) \in \mathbb{Z}^2, f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_j, \quad (1.12)$$

$$\forall j \in \mathbb{Z}, V_{j+1} \subset V_j, \quad (1.13)$$

$$\forall j \in \mathbb{Z}, f(t) \in V_j \Leftrightarrow f\left(\frac{t}{2}\right) \in V_{j+1}, \quad (1.14)$$

$$\lim_{j \rightarrow +\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\}, \quad (1.15)$$

$$\lim_{j \rightarrow -\infty} V_j = \overline{\bigcup_{j=-\infty}^{+\infty} V_j} = L^2(\mathbb{R}) \quad (1.16)$$

et il existe une fonction θ telle que $\{\theta(t-n)\}_{n \in \mathbb{Z}}$ soit une base de Riesz de V_0 .

Soit une fonction φ (appelée fonction d'échelle) et définie par :

$$\hat{\varphi}(\omega) = \frac{\hat{\theta}(\omega)}{\left(\sum_{k=-\infty}^{+\infty} |\hat{\theta}(\omega + 2k\pi)|^2\right)^{1/2}} \quad (1.17)$$

Alors la famille $\{\varphi_{jn}\}_{n \in \mathbb{Z}}$ définie par

$$\varphi_{jn}(t) = \frac{1}{\sqrt{2^j}} \varphi\left(\frac{t-n}{2^j}\right) \quad (1.18)$$

forme une base orthonormée de V_j . Si l'on définit $W_j = V_j \ominus V_{j+1}$, la famille d'ondelettes $\{\psi_{jn}\}_{n \in \mathbb{Z}}$ associée à cette fonction d'échelle (voir [44, 52, 77] pour la construction de telles fonctions) est une base orthonormée de W_j . Donc toute fonction $f \in L^2(\mathbb{R})$ peut se décomposer sous la forme

$$f(t) = \sum_n \alpha_n \varphi_{0n}(t) + \sum_{j=0}^{+\infty} \sum_n \beta_{jn} \psi_{jn}(t) \quad (1.19)$$

où les coefficients $\beta_{jn} = \langle f, \psi_{jn} \rangle$ sont les coefficients de la transformée en ondelette et $\alpha_n = \langle f, \varphi_{0n} \rangle$ sont les coefficients de la projection sur le sous-espace V_0 . En d'autres termes, on a

$$(1.19) \iff f \in V_0 \oplus \bigoplus_{j=0}^{\infty} W_j \quad (1.20)$$

1.1.4 Espaces de Besov

Les espaces de Besov $B_{p,q}^s$ sont des espaces adaptés aux ondelettes ($0 < s < \infty$, $0 < p \leq \infty$ et $0 < q \leq \infty$). Les fonctions dans $B_{p,q}^s$ ont s dérivées dans L^p , q permettant de jouer plus finement sur la régularité des fonctions. Si l'on considère que l'on utilise des ondelettes possédant au moins $s + 1$ moments nuls et de régularité au moins C^{s+1} alors en dimension d , la norme sur $B_{p,q}^s$ est définie par

$$\forall f \in B_{p,q}^s \quad \|f\|_{B_{p,q}^s} = \left[\sum_n |\alpha_n|^p \right]^{1/p} + \left\{ \sum_{j=0}^{+\infty} 2^{j\left(\frac{d}{2} - \frac{1}{p} + s\right)q} \left[\sum_n 2^{j\frac{p}{2}} |\beta_{jn}|^p \right]^{q/p} \right\}^{1/q} \quad (1.21)$$

Les coefficients α_n et β_{jn} sont les coefficients de la décomposition en ondelette exprimés en 1.19.

D'autre part, la version homogénéisée de ces espaces est définie par des conditions analogues à 1.21 où l'on somme cette fois de $j = -\infty$ à $j = +\infty$ (le premier terme disparaissant). Il par ailleurs possible de démontrer que le dual de $\dot{B}_{1,1}^1$ est l'espace $\dot{B}_{-1,\infty}^0$.

1.1.5 Pouvoir d'approximation

La théorie de l'approximation étudie le taux d'erreur résultant entre une fonction et son approximée. Dans notre cas, nous considérerons une méthode d'approximation non-linéaire qui consiste à ne garder que les M plus grands coefficients de la transformée en ondelette (nous garderons cette hypothèse de travail dans le cas des transformées *ridgelets*, *curvelets* et *contourlets* vues dans la suite de ce chapitre). Le résultat suivant donne le taux d'erreur après reconstruction dans le cadre de l'utilisation des ondelettes.

Proposition 1.1.2 *Soit $\tilde{f}_M^{wavelet}$ la fonction reconstruite à partir des M plus grands coefficients de la décomposition en ondelette de f (on supposera que ces coefficients appartiennent à l^1 faible) alors il existe une constante M telle que l'erreur d'approximation soit donnée par*

$$\|f - \tilde{f}_M^{wavelet}\|_{L^2}^2 \leq CM^{-1} \quad (1.22)$$

Cette proposition montre que plus le nombre M de coefficients gardés est grand et plus l'erreur d'approximation est faible.

1.2 La transformée *ridgelet*

1.2.1 Définition

L'extension au cas des images (2D) des ondelettes ne fait qu'utiliser le principe de séparabilité avec une même ondelette 1D en filtrant horizontalement puis verticalement. L'analyse de l'image se fait alors suivant trois directions principales (horizontale, verticale et oblique). Le contenu d'une image ne se limitant pas seulement à ces trois directions, il est aisé de comprendre le besoin de disposer d'outils gardant les avantages de l'analyse multirésolution proposés par les ondelettes et incorporant la notion de directionnalité de manière plus fine. E.Candès a proposé dans [15] une nouvelle transformée plus générale prenant en compte ces besoins : la transformée *ridgelet*. Les fonctions *ridgelet* $\psi_{a,b,\theta}$ sont définies de la même manière que les ondelettes mais en ajoutant la notion d'orientation (contrôlée par le paramètre θ)

$$\psi_{a,b,\theta} : \mathbb{R}^2 \longrightarrow \mathbb{R}^2 \quad (1.23)$$

$$\psi_{a,b,\theta} = \frac{1}{\sqrt{a}} \psi \left(\frac{x_1 \cos \theta + x_2 \sin \theta - b}{a} \right) \quad (1.24)$$

Donc $\psi_{a,b,\theta}$ est constant suivant les lignes $x_1 \cos \theta + x_2 \sin \theta = cste$ et est une ondelette ψ dans le sens transverse de ces lignes.

Définition 1.2.1 *La condition d'admissibilité pour une ridgelette est :*

$$K_\psi = \int \frac{|\hat{\psi}(\xi)|^2}{|\xi|^2} d\xi < \infty \quad (1.25)$$

et elle entraîne $\int \psi(t) dt = 0$.

Par ailleurs, on supposera ψ normalisée

$$\Rightarrow \int \frac{|\hat{\psi}(\xi)|^2}{|\xi|^2} d\xi = 1 \quad (1.26)$$

Sous ces hypothèses, E.Candès définit la transformée *ridgelet* d'une fonction f de la manière suivante

Définition 1.2.2 *Soit une fonction $f(x_1, x_2)$. Les coefficients de sa transformée *ridgelet* sont donnés par*

$$\mathcal{R}_f(a, b, \theta) = \int \psi_{a,b,\theta}^*(x_1, x_2) f(x_1, x_2) dx_1 dx_2 = \langle f, \psi_{a,b,\theta} \rangle \quad (1.27)$$

La formule de reconstruction est donnée par

$$f(x_1, x_2) = \int_0^{2\pi} \int_{-\infty}^{+\infty} \int_0^{+\infty} \mathcal{R}_f(a, b, \theta) \psi_{a,b,\theta}(x) \frac{da}{a^3} db \frac{d\theta}{4\pi} \quad (1.28)$$

D'autre part, la relation de Parseval est vérifiée :

Proposition 1.2.3 *Si $f \in L^1 \cap L^2(\mathbb{R}^2)$ et si ψ est admissible alors*

$$\|f\|_2^2 = c_\psi \int |\langle f, \psi_{a,b,\theta} \rangle|^2 \frac{da}{a^3} db \frac{d\theta}{4\pi} \quad (1.29)$$

où $c_\psi = (4\pi)^{-1} K_\psi^{-1}$

La preuve de cette proposition est donnée dans [15].

1.2.2 Transformée *ridgelet* et transformée de Radon

Nous commençons par rappeler la définition de la transformée de Radon.

Définition 1.2.4 *Transformée de Radon. Soit une fonction $f(x_1, x_2)$. Pour $(\theta, t) \in [0, 2\pi) \times \mathbb{R}$, la transformée de Radon de f est donnée par*

$$R_f(\theta, t) = \int_{\mathbb{R}^2} f(x_1, x_2) \delta(x_1 \cos \theta + x_2 \sin \theta - t) dx_1 dx_2 \quad (1.30)$$

La transformée *ridgelet* peut s'exprimer à partir de la transformée de Radon. Pour cela il suffit de remplacer l'expression de $\psi_{a,b,\theta}$ dans (1.27). On obtient alors

$$\mathcal{R}_f(a, b, \theta) = \int R_f(\theta, t) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt \quad (1.31)$$

autrement dit, la transformée *ridgelet* est obtenue par une transformée en ondelette 1D des lignes ($\theta = cste$) de la transformée de Radon de l'image. L'une des méthodes pour calculer la transformée de Radon est d'utiliser le **théorème de projection des coupes** : on calcule la transformée de Fourier 2D de l'image, on transforme cette image sur une grille cartésienne en une image en coordonnées polaires (droites passant par les fréquences nulles avec différentes orientations) puis on applique une transformée de Fourier inverse 1D sur chacune des lignes (pour $\theta = cste$). Il ne reste ensuite qu'à ajouter une transformée en ondelette 1D suivant ces mêmes lignes pour obtenir la transformée *ridgelet* (voir fig.1.2).

1.2.3 Espace de *ridgelet*- $R_{p,q}^s$

A l'image des espaces de Besov, E.Candès [15] définit des espaces de fonctions adaptés aux *ridgelets*. La définition est la suivante (note : $Ave_u H(u, \cdot)$ signifie que l'on calcule la moyenne de $H(u, \cdot)$ par rapport à la variable u sur le cercle unité)

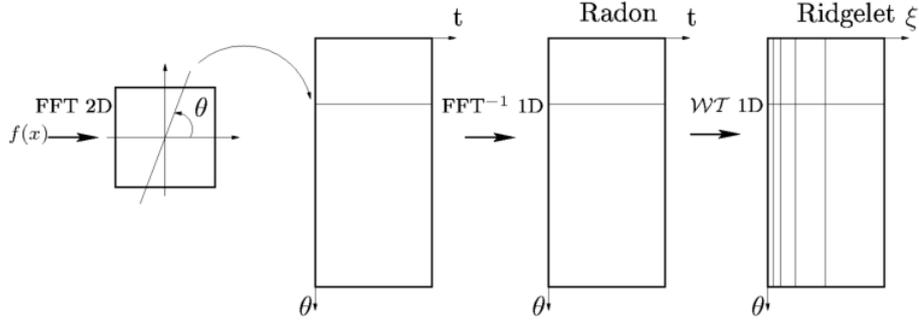


FIG. 1.2 – Construction de la transformée *ridgelet* par le théorème de projection des coupes.

Définition 1.2.5 Pour $s \geq 0$ et $p, q > 0$, on dira que $f \in R_{p,q}^s$ si $f \in L^1$ et

$$\begin{aligned} & Ave_u \|R_f(u, \cdot) \star \varphi\|_{L^p} < \infty \\ & \text{et } \left\{ 2^{js} 2^{j(d-1)/2} \left(Ave_u \|R_f(u, \cdot) \star \psi_j\|_{L^p}^p \right)^{1/p} \right\} \in l_q(\mathbb{N}) \end{aligned} \quad (1.32)$$

où $R_f(u, t) = \int_{u \cdot x = t} f(x) dx$ la transformée de Radon de f ($u = (\cos \theta; \sin \theta)$) et φ est la fonction d'échelle associée à ψ .

On définit alors

$$\begin{aligned} \|f\|_{R_{p,q}^s} &= Ave_u \|R_f(u, \cdot) \star \varphi\|_{L^p} \\ &+ \left\{ \sum_{j \geq 0} \left(2^{js} 2^{j(d-1)/2} \left(Ave_u \|R_f(u, \cdot) \star \psi_j\|_{L^p}^p \right)^{1/p} \right)^q \right\}^{1/q} \end{aligned} \quad (1.33)$$

et sa version homogène $\dot{R}_{p,q}^s$

$$\|f\|_{\dot{R}_{p,q}^s} = \left\{ \sum_{j \in \mathbb{Z}} \left(2^{js} 2^{j(d-1)/2} \left(Ave_u \|R_f(u, \cdot) \star \psi_j\|_{L^p}^p \right)^{1/p} \right)^q \right\}^{1/q} \quad (1.34)$$

Comme dans le cas des espaces de Besov caractérisés par les coefficients d'ondelette, la norme sur cet espace $R_{p,q}^s$ peut se calculer à partir des coefficients de la décomposition *ridgelet*. Pour cela, posons $w_j(u, b)(f) = \langle f(x), \psi_j(u \cdot x - b) \rangle$ pour $j \geq 0$ et $v(u, b)(f) = \langle f(x), \varphi(u \cdot x - b) \rangle$ les coefficients de la décom-

position *ridgelet*, alors

$$\|f\|_{R_{p,q}^s} = \left(\int |v(u,b)(f)|^p du db \right)^{1/p} + \left\{ \sum_{j \geq 0} \left(2^{js} 2^{j(d-1)/2} \left(\int |w_j(u,b)(f)|^p du db \right)^{1/p} \right)^q \right\}^{1/q} \quad (1.35)$$

1.2.4 Remarques - Extensions

Analyse globale vs analyse locale

Nous pouvons remarquer que la transformée *ridgelet* proposée en (1.27) autorise une analyse uniquement le long de droites traversant complètement l'image. Cela signifie que l'on effectue une analyse globale de l'image et non locale ce qui est gênant pour analyser des contours plutôt modélisés par des segments dans l'image. Pour pallier ce défaut, E.Candès et D.Donoho proposent de faire une analyse par bloc de l'image :

- on découpe l'image en bloc de taille $B = 2^s$
- on effectue la transformée *ridgelet* sur chacun des blocs.

Le paramètre s correspond à un facteur d'échelle, on peut ainsi construire une pyramide de *ridgelet* afin d'effectuer l'analyse à différentes échelles et de manière locale (par le découpage en bloc).

Afin d'éviter les effets de bord dûs au découpage en blocs, il est possible d'adopter une stratégie de recouvrement des blocs où les pixels qui se chevauchent sont affectés d'une pondération afin de pouvoir retrouver une reconstruction exacte.

Base de *ridgelets* orthonormales

Dans [79], D.Donoho et al. montrent qu'il est possible de construire une base orthonormale de *ridgelet* en utilisant la reformulation suivante dans le domaine de Fourier

$$\hat{\rho}_\lambda(\xi) = |\xi|^{-\frac{1}{2}} \left(\hat{\psi}_{j,k}(|\xi|) w_{i,l}^\epsilon(\theta) + \hat{\psi}_{j,k}(-|\xi|) w_{i,l}^\epsilon(\theta + \pi) \right) / 2 \quad (1.36)$$

où $\psi_{j,k}$ ($j, k \in \mathbb{Z}$) est l'ondelette de Meyer sur \mathbb{R} et ($w_{i_0,l}^\epsilon$ pour $l = 0, \dots, 2^{i_0} - 1$; $w_{i,l}^\epsilon$ pour $i \geq i_0, l = 0, \dots, 2^i - 1$) est une base orthonormale de $L^2[0, 2\pi]$ construite à partir d'une périodisation des fonctions d'échelle $w_{i_0,l}^0$ de Lemarié à l'échelle i_0 et une périodisation des ondelettes de Meyer $w_{i,l}^1$ pour les échelles $i \geq i_0$. Si on note $\lambda = (j, k, i, l, \epsilon)$ alors les fonctions $\rho_\lambda = TF^{-1}(\hat{\rho}_\lambda)$ sont les fonctions *ridgelet* orthonormales.

1.3 La transformée *curvelet*

1.3.1 Construction de la transformée

L'idée de la transformée *curvelet* [16, 17, 18, 37, 70] est d'appliquer la décomposition pyramidale *ridgelet* non plus sur l'image elle-même mais sur chacune des sous-bandes obtenues en sortie de la transformée en ondelette non décimée (transformée en ondelette à trou) de l'image.

D'un point de vue mathématique, la transformée se construit de la manière suivante (voir fig.1.3)

1. Décomposition en sous-bandes : on définit un banc de filtre $P_0, (\Delta_s)_{s \geq 0}$. On obtient donc la décomposition

$$f \mapsto (P_0 f, \Delta_1 f, \Delta_2 f, \dots) \quad (1.37)$$

2. Partition des sous-bandes : soit l'opérateur $w_Q(x_1, x_2)$ permettant d'obtenir une partition en carrés dyadiques :

$$Q_s = [k_1/2^s, (k_1 + 1)/2^s) \times [k_2/2^s, (k_2 + 1)/2^s) \quad (1.38)$$

Le découpage des sous-bandes est donc obtenu par

$$\Delta_s f \mapsto (w_Q \Delta_s f)_{Q \in \mathcal{Q}_s} \quad (1.39)$$

où \mathcal{Q}_s est l'ensemble des carrés dyadiques possibles sur l'image.

3. Renormalisation : soit l'opérateur défini sur un carré dyadique

$$(T_Q f)(x_1, x_2) = 2^s f(2^s x_1 - k_1, 2^s x_2 - k_2) \quad (1.40)$$

Cet opérateur ramène donc un carré Q sur un carré $[0, 1]^2$. En appliquant l'inverse de cet opérateur sur les carrés de la partition on obtient

$$g_Q = (T_Q)^{-1}(w_Q \Delta_s f)_Q \quad Q \in \mathcal{Q}_s \quad (1.41)$$

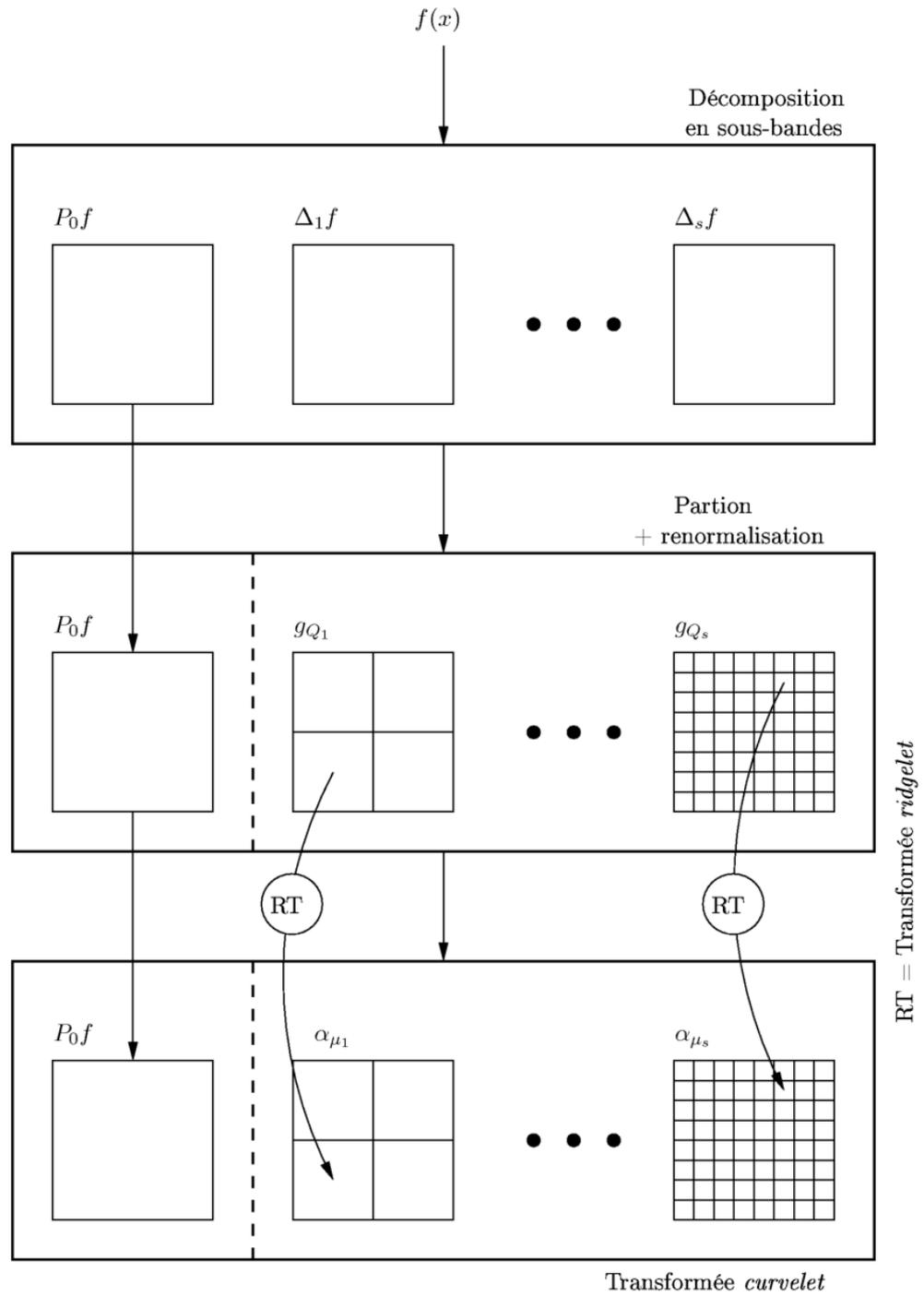
4. Transformée *ridgelet* : les coefficients *curvelet* sont obtenus en prenant la transformée *ridgelet* orthonormale de chacun des blocs de la partition :

$$\alpha_\mu = \langle g_Q, \rho_\lambda \rangle, \quad \mu = (Q, \lambda) \quad (1.42)$$

Les fonctions *curvelet* sont obtenues par

$$\gamma_\mu = \Delta_s \rho_{\lambda, Q} \quad \mu = (\lambda \in \Lambda, Q \in \mathcal{Q}_s) \quad (1.43)$$

La reconstruction est obtenue en déroulant l'algorithme dans l'ordre inverse (transformée *ridgelet* inverse, reconstruction de la partition initiale de l'image, filtrage dual pour la reconstruction de l'image à partir des sous-bandes).

FIG. 1.3 – Construction de la transformée *curvelet*.

1.3.2 Aspects théoriques

E.Candès et D.Donoho ont montré que la transformée décrite précédemment permet de construire un *tight frame* de $L_2(\mathbb{R}^2)$

$$f = \sum_{\mu} \langle f, \gamma_{\mu} \rangle \gamma_{\mu}. \quad (1.44)$$

La relation de Parseval est aussi vérifiée

$$\sum_{\mu} |\langle f, \gamma_{\mu} \rangle|^2 = \|f\|_2^2 \quad (1.45)$$

Il est aussi possible de montrer que les *curvelets* obéissent à une loi d'échelle parabolique :

$$\text{largeur} \approx \text{longueur}^2 \quad (1.46)$$

où

$$\text{longueur}(\gamma_{\mu}) \approx 2^{-s} \quad (1.47)$$

Transformée continue

Dans [17, 18], E.Candès et D.Donoho définissent la transformée *curvelet* continue et montrent que l'on obtient aussi un *frame* de $L_2(\mathbb{R}^2)$. Pour cela, ils effectuent un pavage du domaine fréquentiel par des fenêtres radiales ($W(r)$) et des fenêtres angulaires ($V(t)$). Ces fenêtres doivent vérifier les conditions d'admissibilité suivantes

$$\int_0^{\infty} W(ar)^2 \frac{da}{a} = 1, \quad \forall r > 0, \quad (1.48)$$

$$\int_{-1}^1 V(u)^2 du = 1 \quad (1.49)$$

A partir de ces fenêtres, on peut construire une famille d'éléments d'analyse contrôlés à l'aide de trois paramètres : le facteur d'échelle $a > 0$, la position $b \in \mathbb{R}^2$ et l'orientation $\theta \in [0, 2\pi)$. Si l'on note R_{θ} l'opérateur de rotation d'un angle de θ radians, l'élément de base est donné par

$$\gamma_{ab\theta}(x) = \gamma_{a00}(R_{\theta}(x - b)) \quad (1.50)$$

où

$$\hat{\gamma}_{a00}(r, \omega) = W(ar)V(w/\sqrt{a})a^{3/4}, \quad 0 < a < a_0 \quad (1.51)$$

La transformée *curvelet* continue est alors définie par

$$\Gamma_f(a, b, \theta) = \langle \gamma_{ab\theta}, f \rangle, \quad a < a_0 < \pi^2, b \in \mathbb{R}^2, \theta \in [0, 2\pi) \quad (1.52)$$

Les auteurs montrent que l'on peut reconstruire f à partir de la transformée $\Gamma_f(a, b, \theta)$ et que la formule de Parseval est vérifiée

$$f(x) = \int \Gamma_f(a, b, \theta) \gamma_{ab\theta}(x) \frac{da}{a^3} db d\theta \quad (1.53)$$

$$\|f\|_{L^2}^2 = \int |\Gamma_f(a, b, \theta)|^2 \frac{da}{a^3} db d\theta \quad (1.54)$$

Pouvoir d'approximation des *curvelets*

Dans [16], E.Candès et D.Donoho explorent les *curvelets* du point de vue de la théorie de l'approximation. Ils montrent notamment que les *curvelets* permettent d'avoir un meilleur taux d'approximation que les ondelettes. On note $\tilde{f}_M^{curvelet}$ la fonction reconstruite à partir des M plus grands coefficients de la transformée *curvelet* de f ; f sera supposée très régulière dans des domaines réguliers D_j , $1 \leq j \leq N$, avec des discontinuités de première espèce à travers les frontières Γ_j des D_j . Alors on a

$$\|f - \tilde{f}_M^{curvelet}\|_{L^2}^2 \leq CM^{-2}(\log M)^3 \quad (1.55)$$

1.4 La transformée *contourlet*

Les premières applications des *curvelets* [70, 71, 72] montrent un certain potentiel de la méthode; toutefois leur implémentation numérique n'est pas aisée du fait que la construction est faite dans le domaine continu. M.Do et M.Vetterli [33, 64, 30, 32, 29, 79] ont repensé d'un point de vue discret le problème du filtrage directionnel et l'aspect multirésolution. Pour cela, ils combinent deux méthodes complémentaires :

- une décomposition pyramidale laplacienne par l'algorithme de P.J.Burt et E.H.Adelson [14], afin d'obtenir l'aspect multirésolution,
- un filtrage directionnel 2D proposé par R.H.Bamberger et M.L.T.Smith [9], appliqué sur chacune des sous-bandes de la décomposition pyramidale.

L'intérêt de ces méthodes est qu'elles sont toutes les deux basées sur une construction par banc de filtres ayant la propriété d'être inversibles de manière exacte.

1.4.1 La décomposition pyramidale laplacienne (LP)

Le principe de la décomposition LP est le suivant (on notera f l'image d'entrée), la figure 1.4 décrit les étapes de décomposition et reconstruction :

- on génère une version à résolution plus faible de f par un filtrage passe-bas,

- on sous-échantillonne cette nouvelle image, on obtient alors une image c ,
- on obtient l'image des détails en faisant la soustraction entre f et une prédiction de f obtenue par sur-échantillonnage et filtrage dual de c . L'image d ainsi obtenue correspond à une version de l'image d'origine filtrée par un filtre passe-bande.

La reconstruction est le pseudo-inverse de l'algorithme de décomposition (valable uniquement dans le cas où les filtres sont orthogonaux), l'image reconstruite sera notée \tilde{f} .

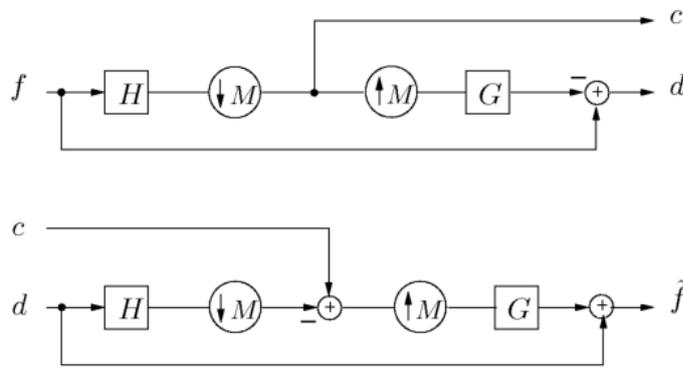


FIG. 1.4 – Décomposition (en haut) et reconstruction (en bas) pyramidale laplacienne (LP)

On applique cette décomposition à chaque niveau de résolution pour obtenir la structure de pyramide (un exemple de décomposition pyramidal est donné fig.1.5).

Le sous-échantillonnage et le sur-échantillonnage sont définis de la manière suivante :

Définition 1.4.1 Soit M la matrice de sous-échantillonnage de taille 2×2 alors le sous-échantillonnage est défini par

$$x_D[n] = x[Mn] \quad (1.56)$$

et le sur-échantillonnage par

$$x_U[n] = \begin{cases} x[k] & \text{si } n = Mk, k \in \mathbb{Z}^2 \\ 0 & \text{sinon} \end{cases} \quad (1.57)$$

En pratique on prendra

$$M = \text{diag}(2, 2) \quad (1.58)$$

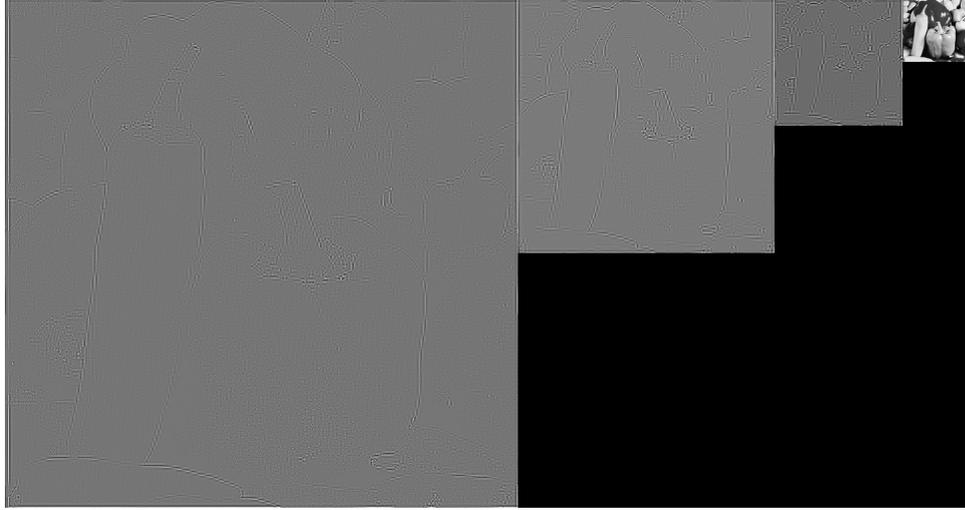


FIG. 1.5 – Exemple de décomposition pyramidale sur 3 niveaux avec un filtre “9-7”.

La décomposition laplacienne permet d’obtenir une analyse multirésolution (voir [33, 35] pour les détails), en effet on peut définir la fonction d’échelle suivante

$$\phi(t) = 2 \sum_{n \in \mathbb{Z}^2} g[n] \phi(2t - n) \quad (1.59)$$

$$\phi_{j,n} = 2^{-j} \phi\left(\frac{t - 2^j n}{2^j}\right) \quad j \in \mathbb{Z}, n \in \mathbb{Z}^2 \quad (1.60)$$

alors dans le cas où le filtre g est un filtre orthogonal on a

Proposition 1.4.2 *A une échelle 2^j , la famille $\{\phi_{j,n}\}_{n \in \mathbb{Z}^2}$ forme une base orthonormale de l’espace V_j . D’autre part la famille $\{V_j\}_{j \in \mathbb{Z}}$ forme une séquence multirésolution*

$$\dots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \dots \quad (1.61)$$

Si l’on note W_j l’espace orthogonal à V_j , on peut montrer que $V_{j-1} = V_j \oplus W_j$. En utilisant la notion de *domaine polyphasé* (voir [35]) on peut définir les fonctions

$$\psi^{(i)}(t) = 2 \sum_{n \in \mathbb{Z}^2} f_i[n] \phi(2t - n) \quad 0 \leq i \leq 3 \quad (1.62)$$

permettant ainsi de construire une famille de *framelet*, nous faisons remarquer que $\psi^{(0)}$ ne correspond pas à une fonction d’échelle (voir [29])

$$\psi_{j,n}^{(i)}(t) = 2^{-j} \psi^{(i)}\left(\frac{t - 2^j n}{2^j}\right) \quad j \in \mathbb{Z}, n \in \mathbb{Z}^2 \quad (1.63)$$

On a alors la proposition suivante

Proposition 1.4.3 *Soit la fonction $\psi_{j,n}^{(i)}$ définie en 1.63 alors*

- à l'échelle 2^j , la famille $\{\psi_{j,n}^{(i)}\}_{0 \leq i \leq 3, n \in \mathbb{Z}^2}$ est une *tight frame* pour W_j ,
- à toutes les échelles, la famille $\{\psi_{j,n}^{(i)}\}_{j \in \mathbb{Z}, 0 \leq i \leq 3, n \in \mathbb{Z}^2}$ est une *tight frame* pour $L^2(\mathbb{R}^2)$

et dans les deux cas la borne de frame est égale à 1.

La preuve est donnée dans [35].

Dans la suite, on notera

$$\mu_{j-1, 2n+k_i}(t) = \psi_{j,n}^{(i)}(t) \quad 0 \leq i \leq 3 \quad (1.64)$$

où

$$k_0 = (0, 0)^T ; k_1 = (1, 0)^T ; k_2 = (0, 1)^T ; k_3 = (1, 1)^T \quad (1.65)$$

De la proposition 1.4.3 découle directement que la famille $\{\mu_{j-1, n}\}_{n \in \mathbb{Z}^2}$ est une *frame* ajustée de W_j .

La décomposition pyramidale laplacienne permet d'obtenir un pavage du plan fréquentiel en bandes concentriques (voir fig.1.6).

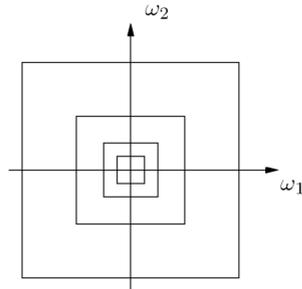


FIG. 1.6 – Pavage du plan fréquentiel obtenu par la décomposition pyramidale laplacienne.

1.4.2 Filtrage directionnel

Le filtrage directionnel 2D utilisé par M.Do et M.Vetterli est celui proposé par R.H.Bamberger et M.L.T.Smith dans [9]. Le principe du filtrage directionnel est de fournir les réponses au travers d'un banc de filtres orientés (*Directional Filter Bank ou DFB*) d'une image d'entrée d (voir fig.1.7).

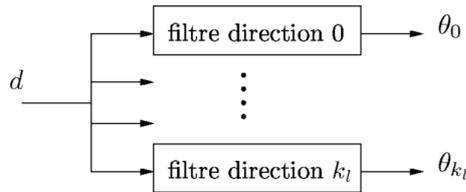


FIG. 1.7 – Banc de filtre directionnel

Proposition 1.4.4 Pour $d \in$ un espace Θ et $\forall l < \infty$, on définit

$$\theta_{k,n}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} g_k^{(l)}[m - S_k^{(l)}n]d \quad (1.66)$$

où l'opérateur $S_k^{(l)}$ "déforme" l'image par l'intermédiaire de filtres quincux (voir [29] et fig.1.8 pour un exemple).

La famille $\{\theta_{k,n}^{(l)}\}_{n \in \mathbb{Z}^2}$ est une base orthogonale des espaces directionnels

$$\Theta_k^{(l)}, \forall k = \{0, \dots, 2^l - 1\}$$

En conséquence, on a les propriétés suivantes

$$\Theta_k^{(l)} \perp \Theta_{k'}^{(l)} \quad \forall k \neq k' \quad (1.67)$$

$$\Theta_k^{(l)} = \Theta_{2k}^{(l+1)} \oplus \Theta_{2k+1}^{(l+1)} \quad (1.68)$$

$$\Theta = \bigoplus_{k=0}^{2^l-1} \Theta_k^{(l)} \quad (1.69)$$

Ce filtrage directionnel permet d'obtenir un pavage par "tranches orientées" du plan fréquentiel (voir fig.1.9).

1.4.3 La transformée *contourlet*

M.Do et M.Vetterli combinent les aspects d'analyse multirésolution et de filtrage directionnel pour construire la transformée en *contourlet* (aussi appelée *Pyramidal Directional Filter Bank* (PDFB)). Le principe étant d'appliquer

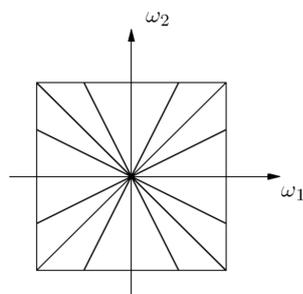
FIG. 1.8 – Exemple de déformation par un filtre *quincunx*.

FIG. 1.9 – Pavage du plan fréquentiel obtenu par filtrage directionnel.

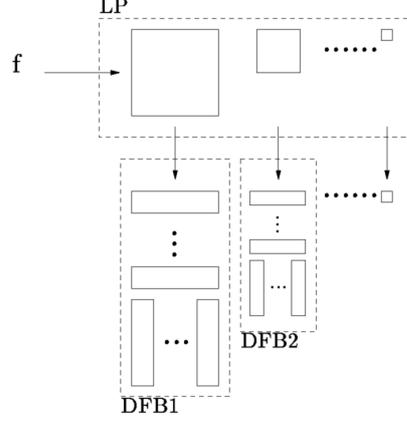
un filtrage directionnel sur chacune des images de détails issues de la décomposition laplacienne (voir la fig.1.10).

D'un point de vue mathématique, cela revient à appliquer un filtrage directionnel sur chacun des espaces W_j de la décomposition pyramidale. On obtient donc une famille de fonctions ($j =$ échelle, $k =$ direction, $n =$ localisation)

$$\rho_{j,k,n}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} g_k^{(l)}[m - S_k^{(l)}n] \mu_{j-1,m}(t) \quad (1.70)$$

On a alors les propositions suivantes

Proposition 1.4.5 *La famille des fonctions $\{\rho_{j,k,n}^{(l)}\}_{n \in \mathbb{Z}^2}$, telles que définies en (1.70), est un tight frame de $W_{j,k}^{(l)}$, $\forall k = 0, \dots, 2^l - 1$, dont la borne vaut 1. Les $W_{j,k}^{(l)}$ sont orthogonaux suivant les échelles et les directions.*

FIG. 1.10 – Principe de la transformée *contourlet*.

Proposition 1.4.6 *On note*

$$\rho_{j,k}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} g_k^{(l)}[m] \mu_{j-1,m}(t) \quad (1.71)$$

alors pour $l \geq 2$

$$\rho_{j,k,n}^{(l)}(t) = \rho_{j,k}^{(l)}(t - 2^{j-1} S_k^{(l)} n) \quad (1.72)$$

On en déduit

$$\rho_{j,k}^{(l)}(t) = \sum_{m \in \mathbb{Z}^2} \underbrace{\left(\sum_{i=0}^3 \sum_{n \in \mathbb{Z}^2} g_k^{(l)}[2n + k_i] f_i[m - 2n] \right)}_{c_k^{(l)}[n]} \phi_{j-1,m}(t) \quad (1.73)$$

d'autre part on peut montrer que

$$\left\| \rho_{j,k,n}^{(l)} \right\|_{L^2}^2 = \frac{3}{4} \quad \forall l \geq 2, j \in \mathbb{Z}, 0 \leq k < 2^l, n \in \mathbb{Z}^2 \quad (1.74)$$

M.Do et M.Vetterli montrent le théorème suivant

Théorème 1.4.7 $\forall \{l_j\}_{j \leq j_0}$ une famille donnant le nombre de directions de filtrage par niveau de résolution. Alors la famille

$$\left\{ \phi_{j_0,n}(t); \rho_{j,k,n}^{(l_j)}(t) \right\}_{j \leq j_0, 0 \leq k \leq 2^{l_j} - 1, n \in \mathbb{Z}^2} \quad (1.75)$$

est une tight frame de $L_2(\mathbb{R}^2)$, ayant pour borne 1.

Corollaire 1.4.8 $\forall \{l_j\}_{j \in \mathbb{Z}}$ une famille donnant le nombre de directions de filtrage par niveau de résolution. Alors la famille

$$\left\{ \rho_{j,k,n}^{(l_j)}(t) \right\}_{j \in \mathbb{Z}, 0 \leq k \leq 2^{l_j} - 1, n \in \mathbb{Z}^2} \quad (1.76)$$

est une tight frame de $L_2(\mathbb{R}^2)$, ayant pour borne 1.

Corollaire 1.4.9 On en déduit

$$L_2(\mathbb{R}^2) = V_{j_0} \oplus \left(\bigoplus_{j \leq j_0} W_j \right) \quad (1.77)$$

$$\Leftrightarrow f(t) = \sum_n \alpha_n \phi_{j_0,n}(t) + \sum_{j \leq j_0} \sum_{k=0}^{2^{l_j}-1} \sum_n \beta_{j,k,n} \rho_{j,k,n}^{(l_j)}(t) \quad (1.78)$$

et

$$L_2(\mathbb{R}^2) = \bigoplus_{j \in \mathbb{Z}} W_j \quad (1.79)$$

$$\Leftrightarrow f(t) = \sum_{j \in \mathbb{Z}} \sum_{k=0}^{2^{l_j}-1} \sum_n \beta_{j,k,n} \rho_{j,k,n}^{(l_j)}(t) \quad (1.80)$$

Les coefficients $\alpha_n = \langle f | \phi_{j_0,n} \rangle$ et $\beta_{j,k,n} = \langle f | \rho_{j,k,n}^{(l_j)} \rangle$ sont les coefficients de la décomposition *contourlet*.

On obtient alors un pavage du plan fréquentiel où chaque couronne représentant un niveau de résolution est elle-même redécoupée en portions correspondant aux directions que l'on se fixe pour chaque sous-bande (voir fig.1.11).

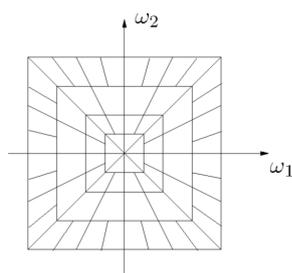


FIG. 1.11 – Découpage du plan fréquentiel dans le cas de la transformée *contourlet*.

On peut remarquer que si l'on double le nombre de directions toutes les deux sous-bandes de résolution alors on obtient un découpage équivalent à celui de la transformée *curvelet* de E.Candès et D.Donoho (à la différence près

que les bandes concentriques de résolution dans le domaine de Fourier sont carrées dans le cas des *contourlets* et circulaires pour les *curvelets* !

La figure 1.12 donne un exemple de transformée *contourlet*. En haut à droite on trouve l'approximation à basse résolution, puis de droite à gauche on voit apparaître trois sous-bandes elles-mêmes filtrées respectivement suivant 8, 8 et 16 directions.

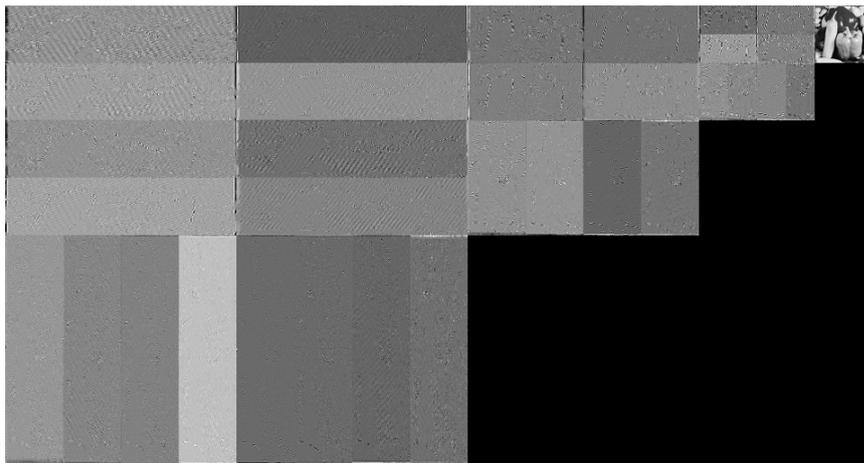


FIG. 1.12 – Exemple de transformée en *contourlet*.

En se plaçant dans le même cadre d'hypothèse que les *curvelets* (f sera supposée très régulière dans des domaines réguliers D_j , $1 \leq j \leq N$, avec des discontinuités de première espèce à travers les frontières Γ_j des D_j), M.Do et M.Vetterli montrent, d'un point de vue de la théorie de l'approximation et moyennant quelques hypothèses sur les *contourlets* [33], que si l'on ne conserve que les M plus grands coefficients de la transformée *contourlet* (on notera $\tilde{f}_M^{\text{contourlet}}$ l'image reconstruite) alors l'erreur d'approximation vaut

$$\|f - \tilde{f}_M^{\text{contourlet}}\|_{L^2}^2 \leq C(\log M)^3 M^{-2} \quad (1.81)$$

ce qui est le même taux d'erreur d'approximation que dans le cadre des *curvelets*.

1.4.4 Espace de *contourlets*

Nous proposons dans cette section de définir l'espace des *contourlets*, que l'on notera $CT_{p,q}^s$. Pour cela, nous nous inspirons de la démarche suivie dans le cas des espaces de Besov et de *ridgelets* (d étant la dimension).

Définition 1.4.10 Soient $s \geq 0$ et $p, q > 0$, si $f \in CT_{p,q}^s$ alors

$$\|f\|_{CT_{p,q}^s} = \left[\sum_n |\alpha_{j_0,n}|^p \right]^{1/p} + \left\{ \sum_{j \leq j_0} 2^{j \left(\frac{d}{2} - \frac{1}{p} + s \right) q} \left[\sum_{k=0}^{2^j-1} \sum_n 2^{j \frac{p}{2}} |\beta_{j,k,n}|^p \right]^{q/p} \right\}^{1/q} \quad (1.82)$$

ou

$$\|f\|_{CT_{p,q}^s} = \left\{ \sum_{j \in \mathbb{Z}} 2^{j \left(\frac{d}{2} - \frac{1}{p} + s \right) q} \left[\sum_{k=0}^{2^j-1} \sum_n 2^{j \frac{p}{2}} |\beta_{j,k,n}|^p \right]^{q/p} \right\}^{1/q} \quad (1.83)$$

où les coefficients $\alpha_{j_0,n}$ et $\beta_{j,k,n}$ sont les coefficients de la décomposition contourlet définis dans le corollaire 1.4.9.

1.5 Débruitage d'image

1.5.1 Débruitage par seuillage de coefficients

Le bruit présent dans une image vient altérer les détails de l'image. Cela signifie donc que les coefficients de la décomposition en ondelette du bruit seront d'amplitude faible par rapport aux coefficients du contenu sémantique de l'image. L'idée de base du débruitage est donc de supprimer les coefficients d'ondelette correspondants au bruit, pour cela on utilise généralement une méthode de seuillage des coefficients. Les deux méthodes les plus classiques dans la littérature sont les méthodes dites de seuillage doux (*Wavelet Soft Thresholding*) et de seuillage dur (*Wavelet Hard Thresholding*). Nous rappelons ci-après les définitions de ces deux méthodes de seuillage.

Définition 1.5.1 (*Wavelet Soft Thresholding*) L'opération de seuillage doux des coefficients d'ondelette β_{jn} d'une fonction f pour un seuil δ est définie par (on note $\tilde{\beta}_{jn}$ les coefficients seuillés)

$$\tilde{\beta}_{jn} = \begin{cases} \beta_{jn} - \delta & \text{si } \beta_{jn} > \delta \\ 0 & \text{si } |\beta_{jn}| \leq \delta \\ \beta_{jn} + \delta & \text{si } \beta_{jn} < -\delta \end{cases} \quad (1.84)$$

D. Donoho et al. [36] montrent que le seuil δ optimum dans le cas d'un bruit blanc gaussien est donné par

$$\delta = \sigma \sqrt{\log N} \quad (1.85)$$

où σ correspond à la variance du bruit et N est la taille de l'image ($N \times N$). On notera $WST(f; \delta)$ l'opérateur effectuant la transformée en ondelette, le seuillage doux et la reconstruction de l'image à partir des coefficients seuillés. Par ailleurs, on peut introduire un facteur de pondération κ permettant de jouer sur la valeur du seuil. On prendra alors le seuil de la manière suivante

$$\delta = \kappa \sigma \sqrt{\log N} \quad (1.86)$$

Définition 1.5.2 (*Wavelet Hard Thresholding*) L'opération de seuillage dur des coefficients d'ondelette β_{jn} d'une fonction f pour un seuil δ est définie par (on note $\tilde{\beta}_{jn}$ les coefficients seuillés)

$$\tilde{\beta}_{jn} = \begin{cases} \beta_{jn} & \text{si } |\beta_{jn}| > \delta \\ 0 & \text{sinon} \end{cases} \quad (1.87)$$

le seuil δ peut être choisi de la même manière que dans le cas du seuillage doux (voir définition 1.5.1).

On notera $WHT(f; \delta)$ l'opérateur effectuant la transformée en ondelette, le seuillage dur et la reconstruction de l'image à partir des coefficients seuillés.

L'opération de débruitage consiste simplement à appliquer les opérateurs $WST(f, \delta)$ ou $WHT(f, \delta)$ sur l'image bruitée. On peut voir immédiatement que l'on peut aussi définir le même type d'opérateur mais sur les coefficients *ridgelets*, *curvelets* ou *contourlets*. Ces transformées modélisant mieux les structures dans les images, les résultats de débruitage sont en théorie meilleurs que dans le cadre des ondelettes. En pratique, seule la transformée *contourlet* sera utilisée dans nos expérimentations (la transformée *ridgelet* étant moins efficace que la transformée *curvelet* et cette dernière peut être vue comme un cas particulier de la transformée *contourlet* dans le cas discret). On note alors $CST(f, \delta)$ et $CHT(f, \delta)$ les opérateurs de seuillage respectivement doux et durs des coefficients de la transformée *contourlet* (*Contourlet Soft Thresholding* et *Contourlet Hard Thresholding*).

Les tests qui suivent seront effectués à partir de l'image de droite de la figure 1.13 celle-ci étant obtenue à partir de l'image de gauche sur laquelle nous avons ajouté un bruit gaussien ($\sigma = 20$).

La figure 1.14 donne le résultat du débruitage obtenu par seuillage des coefficients d'ondelette (seuillage doux à gauche et seuillage dur à droite) ainsi que le bruit extrait.

Quant à la figure 1.15, elle expose les résultats du débruitage obtenu par seuillage des coefficients *contourlets* (seuillage doux à gauche et seuillage dur à droite) ainsi que le bruit extrait.

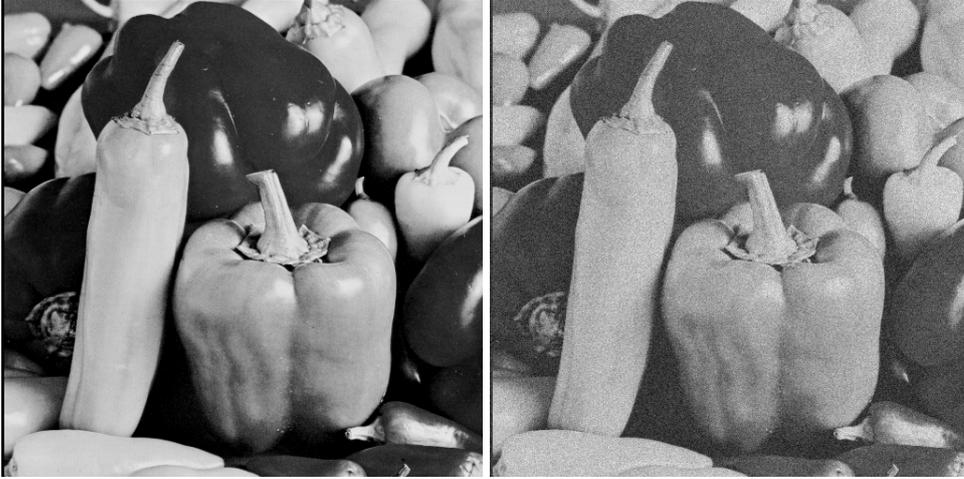


FIG. 1.13 – Image de base et sa version bruitée (bruit gaussien, $\sigma = 20$) servant de test pour le débruitage.

1.5.2 Débruitage à base de variation totale

Rudin et al. [67] se sont retrouvés confrontés au problème de la restauration d'image, le but étant de retrouver les objets présents dans l'image en éliminant le bruit et en tenant compte de l'éventuelle présence d'un noyau de convolution venant perturber l'image (ce dernier cas ne sera pas étudié dans le cadre de cette thèse). Pour retrouver l'image originale, les auteurs proposèrent de minimiser la fonctionnelle suivante :

$$F_{\lambda}^{ROF}(u) = J(u) + (2\lambda)^{-1} \|f - u\|_{L^2}^2 \quad (1.88)$$

où f est l'image dégradée mesurée, u est l'image restaurée et $J(u) = \|u\|_{BV}$. Ce modèle revient à considérer que la fonction u appartient à l'espace fonctionnel BV (espace des fonctions à variations bornées). Nous rappelons que $J(u) = \|u\|_{BV} = \int |\nabla u(x)| dx$ ($J(u)$ représente donc la variation totale de u).

La figure 1.16 montre le résultat obtenu sur l'image de test de la figure 1.13 (l'image de droite étant l'image débruitée et celle de gauche le bruit extrait). Ces différents résultats montrent que dans tous les cas, nous obtenons une partie u équivalente. Concernant la partie bruit, nous pouvons constater la présence plus ou moins importante d'un résidu. Ceci montre bien que pour juger des performances de ce type d'algorithme en tant que algorithme de décomposition, il faut tenir compte des deux parties extraites.

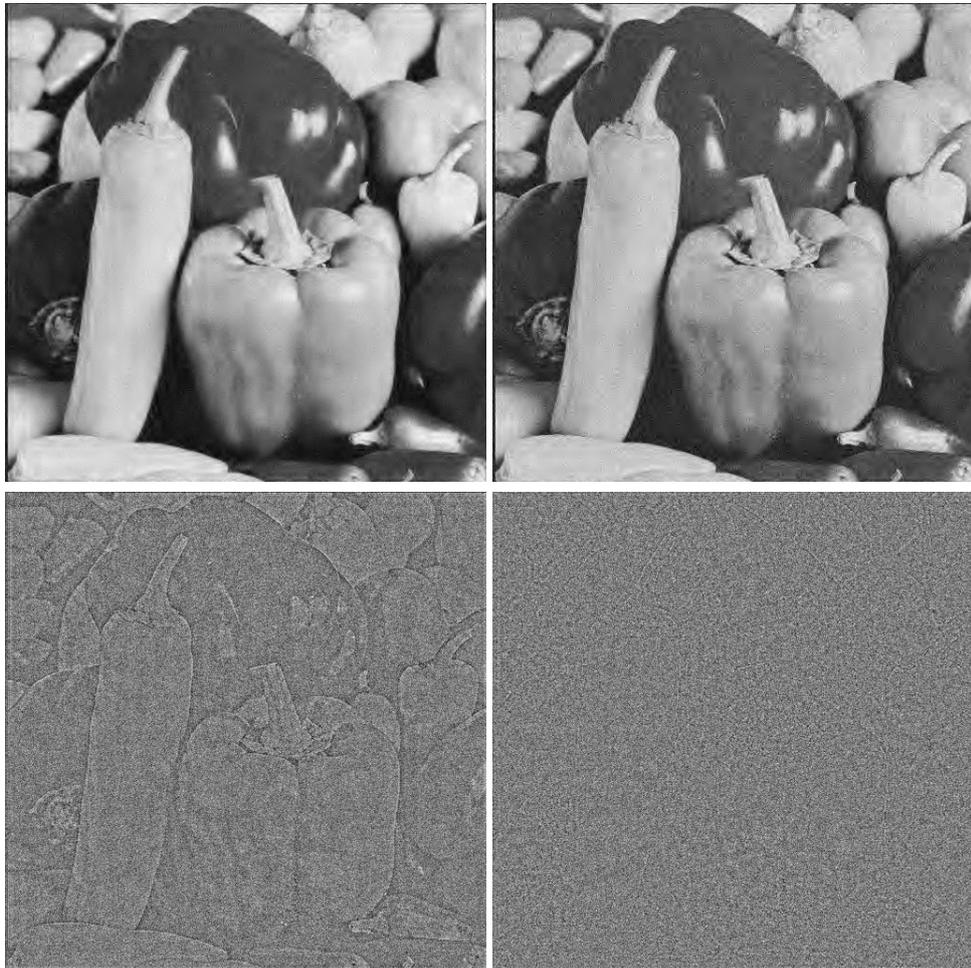


FIG. 1.14 – Exemple de débruitage par les seuillages dur (colonne de droite) et doux (colonne de gauche) des coefficients d'ondelettes. La deuxième ligne représentant le bruit extrait.

1.6 Mesure de la qualité du bruit extrait

Dans cette section, nous proposons une méthode permettant de quantifier la «qualité» du bruit extrait. Il nous faut tout d'abord définir ce que nous entendons par le terme «qualité». Pour cela supposons que l'image à débruitée f peut s'écrire comme la somme d'une image non bruitée et d'un bruit gaussien.

$$f = d + b \quad (1.89)$$

Visuellement les expériences montrent que lorsque l'on applique un algorithme de débruitage, le bruit extrait \hat{b} contient toujours un «reste» de

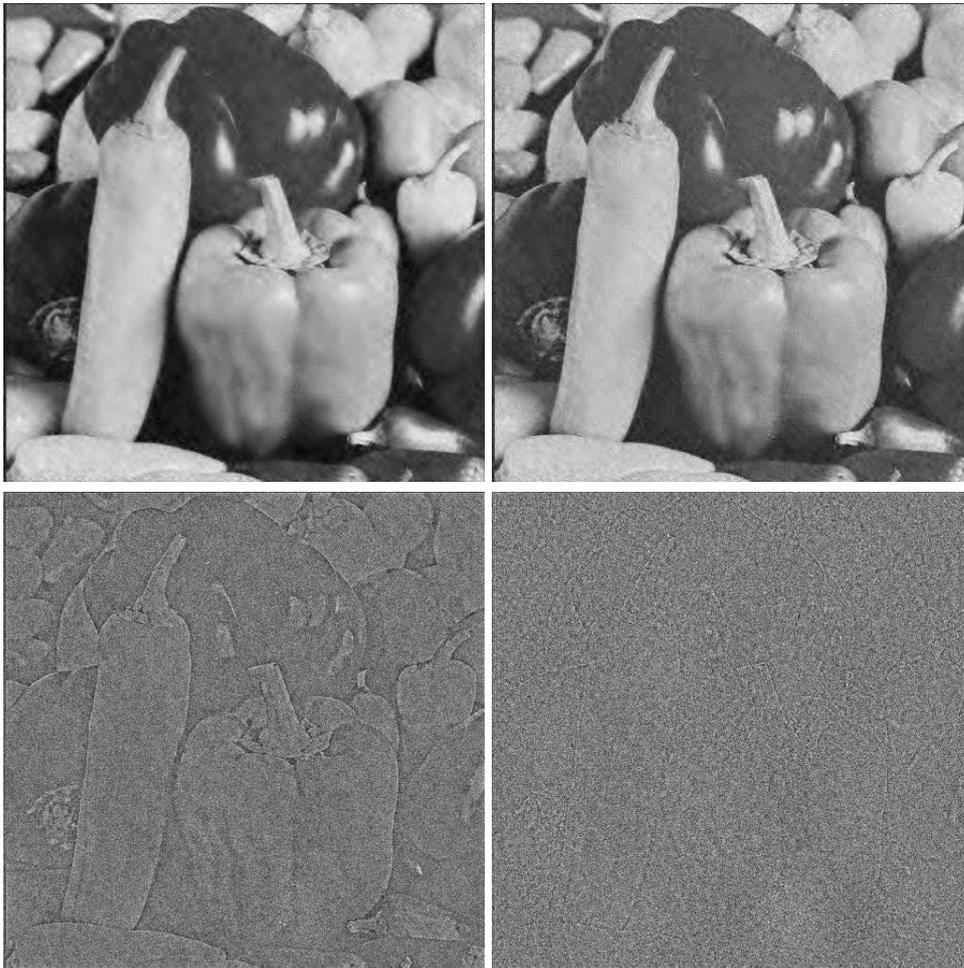


FIG. 1.15 – Exemple de débruitage par les seuillages dur (colonne de droite) et doux (colonne de gauche) des coefficients *contourlets*. La deuxième ligne représentant le bruit extrait.

l'image non bruitée d'origine et peut donc s'écrire

$$\tilde{b} = Ad + b \quad (1.90)$$

Où $A \in \mathbb{R}$. Dans la suite, nous appellerons «résidu» cette quantité Ad . Nous considérerons que la composante bruit sera d'autant meilleure qualité que cette quantité de résidu sera faible et que le bruit extrait se rapproche effectivement d'un bruit gaussien. La méthode proposée ci-après permet de quantifier cette quantité de résidu.

Nous faisons l'hypothèse d'un bruit gaussien de variance σ^2 . Le calcul de la

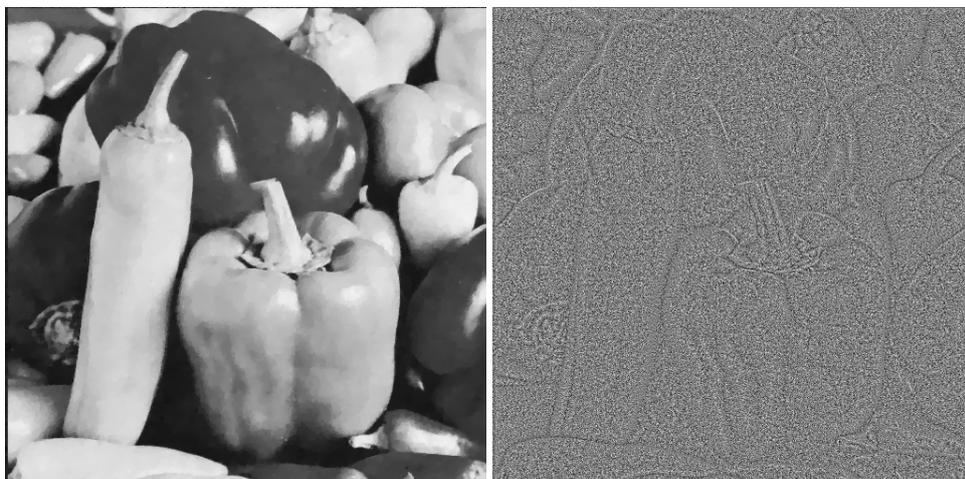


FIG. 1.16 – Algorithme ROF : à gauche l'image débruitée ($\lambda = 20$) et à droite le bruit extrait.

fonction d'autocorrélation nous permet de voir si le bruit extrait se rapproche effectivement d'un bruit gaussien (dans ce cas, cette fonction d'autocorrélation doit se résumer à un pic en $(0,0)$). La figure 1.17 montre les fonctions d'autocorrélation de chacun des bruits extraits sur les figures 1.14, 1.15 et 1.16.

Nous pouvons observer que l'on retrouve bien le pic central majoritairement dû au bruit. Le plus intéressant étant le fait que le résidu présent dans le bruit vient «perturber» la fonction d'autocorrélation. Nous pouvons vérifier théoriquement ce comportement dans un cas plus général. La proposition suivante donne un lien entre l'énergie de la fonction d'autocorrélation et la quantité de résidu présente dans le bruit.

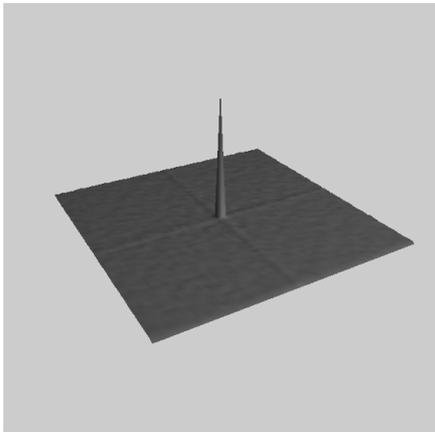
Proposition 1.6.1 *Soit un bruit gaussien de variance σ^2 noté $b(i, j)$ et une image non bruitée noté $d(i, j)$. Nous simulons alors l'image $f = Ad + b$ où $A \in \mathbb{R}$ et représente le niveau de résidu. Alors*

$$\|\gamma_f - \gamma_b\|_{L^2} \propto A^2 \quad (1.91)$$

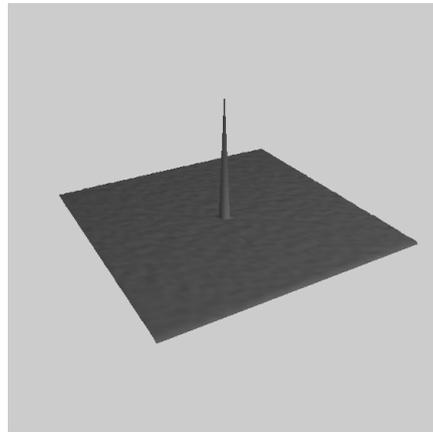
Preuve:

Nous commençons par calculer la fonction d'autocorrélation de f :

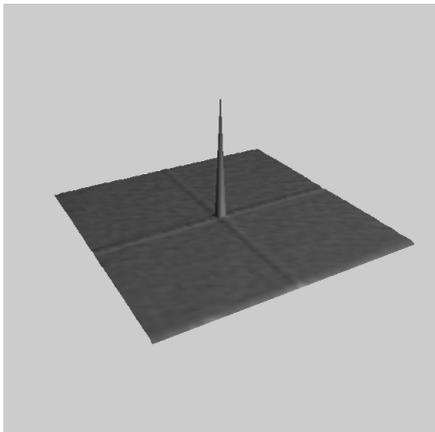
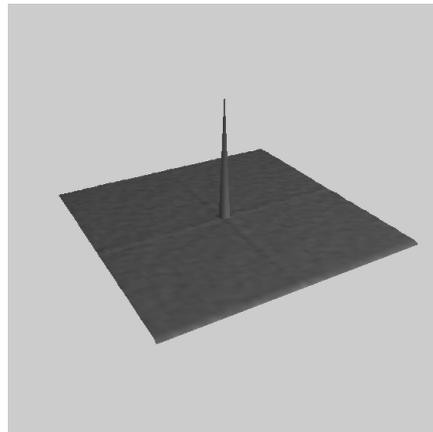
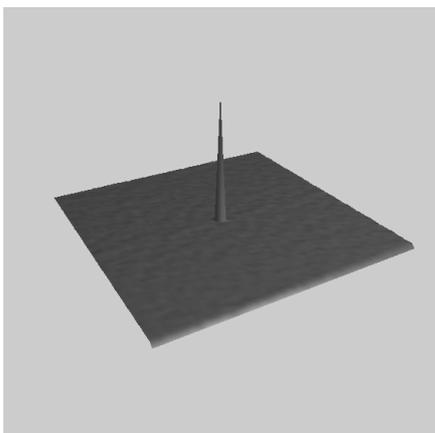
$$\gamma_f(k, l) = \sum_{(i, j) \in \mathbb{Z}^2} f(i, j) f^*(i + k, j + l) \quad (1.92)$$



Seuillage doux (cas ondelettes)



Seuillage dur (cas ondelettes)

Seuillage doux (cas *contourlets*)Seuillage dur (cas *contourlets*)

Cas variation totale

FIG. 1.17 – Fonctions d'autocorrélation des bruits extraits sur les figures 1.14, 1.15 et 1.16.

or nous travaillons avec des signaux réels donc $f(i, j) = f^*(i, j)$. On en déduit

$$\gamma_f(k, l) = \sum_{(i, j) \in \mathbb{Z}^2} [Ad(i, j) + b(i, j)] [Ad(i + k, j + l) + b(i + k, j + l)] \quad (1.93)$$

$$\begin{aligned} &= \sum_{(i, j) \in \mathbb{Z}^2} A^2 d(i, j) d(i + k, j + l) + \sum_{(i, j) \in \mathbb{Z}^2} b(i, j) b(i + k, j + l) + \\ &\quad \sum_{(i, j) \in \mathbb{Z}^2} [Ad(i, j) b(i + k, j + l) + Ad(i + k, j + l) b(i, j)] \quad (1.94) \end{aligned}$$

$$= A^2 \gamma_d(k, l) + \gamma_b(k, l) + A(\gamma_{db}(k, l) + \gamma_{bd}(k, l)) \quad (1.95)$$

Nous nous intéressons maintenant à la norme $\|\cdot\|_{L^2}$ de cette fonction d'auto-corrélation. Tout d'abord remarquons que $\gamma_b(k, l) = \sigma^2 \delta(k, l)$ (où $\delta(k, l)$ est le symbol de Kronecker) car nous avons supposé un bruit gaussien. En pratique, nous pouvons vérifier aisément que la quantité $A(\gamma_{db}(k, l) + \gamma_{bd}(k, l))$ est négligeable devant la quantité $A^2 \gamma_d(k, l)$. Si nous déduisons la contribution due au bruit, nous faisons alors l'approximation

$$\gamma_f(k, l) - \gamma_b(k, l) \approx A^2 \gamma_d(k, l) \quad (1.96)$$

donc en passant à la norme

$$\|\gamma_f - \gamma_b\|_{L^2} \approx A^2 \|\gamma_d\|_{L^2} \quad (1.97)$$

■

Afin de vérifier ce résultat en pratique, nous prenons pour $d(i, j)$ l'image de gauche de la figure 1.13 et une image $b(i, j)$ de bruit ($\sigma = 20$). Nous composons les images $f = Ad + b$ pour $A \in \{0.05; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9\}$ afin de faire apparaître de plus en plus de résidu dans le bruit (voir la figure 1.18)

La figure 1.19 donne les valeurs calculées de $\|\gamma_f - \gamma_b\|_{L^2}$ et trace la courbe associée. Nous constatons que cette norme suit bien une forme quadratique en A comme initialement prévu. Nous avons donc maintenant à notre disposition un moyen de quantifier la quantité de résidu présent dans le bruit et par là même de juger de la qualité du bruit extrait.

1.6.1 Comparaison des différents algorithmes de débruitage

Nous avons vu différentes méthodes de débruitage, généralement ces méthodes sont comparées entre elles en calculant la norme $\|f_{ref} - u\|_{L^2}$ (dans la suite nous notons f_{ref} l'image de départ, f sa version bruitée par le bruit b , u et v sont respectivement la version débruitée de f et le bruit extrait). Dans ce cas il n'est pas fait mention de la qualité du bruit extrait ce qui est logique étant donné que l'on s'intéresse au «pouvoir de débruitage» de la

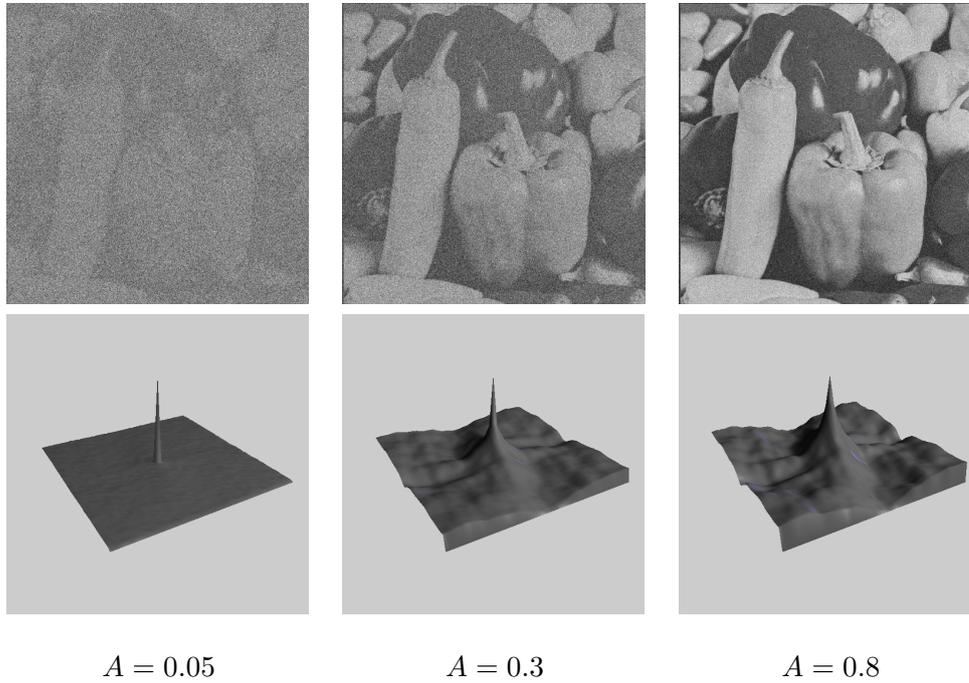


FIG. 1.18 – Images de bruit affectées de plusieurs niveaux de résidu et leurs fonctions d'autocorrélation associées.

A	$\ \gamma_f - \gamma_b\ _{L^2}$
0.05	849.093432
0.1	3312.071022
0.2	13099.095280
0.3	29367.800483
0.4	52118.223554
0.5	81350.371724
0.6	117064.247377
0.7	159259.851531
0.8	207937.184693
0.9	263096.247142

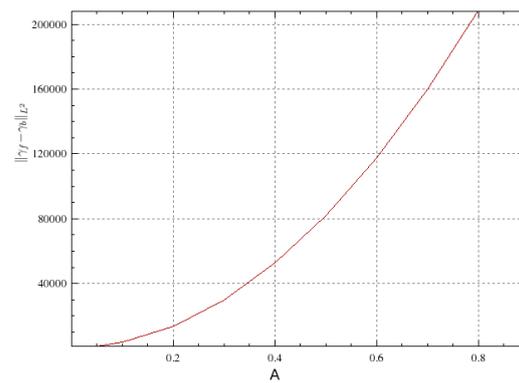


FIG. 1.19 – Mesure et tracé de l'évolution de la norme $\|\gamma_f - \gamma_b\|_{L^2}$ en fonction de la quantité de résidu.

méthode. Dans cette thèse, nous adoptons le point de vue de la décomposition de l'image et considérons le bruit comme une composante à part entière. Nous souhaitons juger de la qualité des deux composantes pour chacun des algorithmes, pour cela nous proposons d'effectuer l'expérience suivante : nous appliquons les différents algorithmes de débruitage sur l'image afin d'obtenir la même erreur $\|f - u\|_{L_2}$. Nous quantifions alors la qualité du bruit extrait par le principe de calcul vu à la section 1.6. Le tableau 1.20 récapitule les résultats obtenus (nous rappelons les notations suivantes : WST est le *Wavelet Soft Thresholding*, WHT est le *Wavelet Hard Thresholding*, CST est le *Contourlet Soft Thresholding*, CHT est le *Contourlet Hard Thresholding* et ROF est l'algorithme de Rudin-Osher-Fatemi).

Nous pouvons constater que les mesures effectuées suivent bien les résultats visuels obtenus sur les images en sortie des différents algorithmes. La composante de bruit est de meilleure qualité dans le cadre des seuillages durs (le résidu y est plus faible). Ici la transformée en ondelette donne de meilleurs résultats que les *contourlets* car nous contraignons ces dernières à être au même niveau de reconstruction que les ondelettes (nous gardons un pourcentage plus faible de coefficients *contourlets* que de coefficients d'ondelettes). Toutefois, l'expérience montre que les résultats obtenus ici par les *contourlets* et l'algorithme ROF sont sous optimaux.

	$\ f_{ref} - u\ _{L_2}$	$\ \gamma_v - \gamma_b\ _{L_2}$
WST ($s = 19.7$)	4992	287
WHT ($s = 51$)	4997	252
CST ($s = 18.9$)	4994	283
CHT ($s = 50$)	4998	271
ROF ($\lambda = 7.3$)	4994	344

FIG. 1.20 – Comparaison de la qualité du bruit extrait des divers algorithmes de débruitage.

1.7 Bilan

Dans ce chapitre, nous avons rappelé quelques résultats classiques du débruitage par seuillage doux et dur des coefficients d'ondelette. Nous examinons par ailleurs le même type de seuillage appliqué sur des coefficients *contourlet*. Cette transformée fournissant un outil équivalent des ondelettes mais permettant en plus d'ajouter la notion d'orientation, elle permet ainsi de mieux tenir compte de la géométrie dans les images.

Nous examinons ces algorithmes en prenant un point de vue de décomposition d'image. Nous proposons une méthode basée sur une mesure de l'énergie de la fonction d'autocorrélation du bruit permettant de juger de la qualité du bruit extrait. Les différents algorithmes peuvent ainsi être comparés en prenant ce point de vue. Il ressort qu'un seuillage dur donne un bruit de meilleure qualité qu'un seuillage doux. Concernant l'algorithme ROF, le fait de contraindre le résultat de reconstruction de u fait que le résultat fourni n'est pas l'optimum pour cet algorithme.

Ce chapitre nous permet de nous donner une voie sur la manière de traiter le bruit en tant que composante à part entière. Le chapitre suivant s'intéresse à la modélisation des textures dans les images.

Chapitre 2

L'analyse de texture

Nous avons vu dans le chapitre précédent différentes manières de traiter le bruit. Dans ce chapitre, nous nous intéressons au problème de la texture. Nous commençons par donner quelques généralités, puis nous examinons un modèle de texture proposé par Y.Meyer (inspiré de l'algorithme ROF) et étudions de plus près le comportement de ce nouvel algorithme. Nous présentons, ensuite, deux algorithmes numériques associés : l'un proposé par S.Osher et L.Vese, l'autre par J.F.Aujol et al. Ces deux algorithmes permettent de mettre en œuvre cette modélisation. Nous verrons enfin une autre approche proposée par A.Haddad utilisant certains espaces de Besov bien choisis.

2.1 Quelques généralités.

De nos jours, la notion même de texture n'est toujours pas définie de manière unique. Dans [22], les auteurs regroupent plusieurs définitions et propriétés possibles d'une texture ; nous pouvons toutefois dégager quelques grandes lignes caractéristiques assez générales :

- une texture peut être vue comme la répétition de motifs suivant certaines règles,
- une texture est une région de l'image ayant certaines propriétés statistiques locales particulières,
- une texture est composée de primitives réparties suivant à la fois une organisation spatiale particulière et certaines propriétés au travers des échelles.

Ces différentes définitions ont inspiré des pistes différentes pour analyser les textures : méthodes statistiques (calculs de moments, matrices de coocurrence), méthodes fréquentielles (spectre de densité de puissance, transformée de Fourier, transformée en ondelette, filtrage), modélisations autorégressives, champs de Markov, approches géométriques (pavage de Voronoï, modélisation fractale). Toutefois, le problème de l'analyse de textures n'a toujours

pas trouvé de solution de manière globale.

Une nouvelle voie a été proposée par Y. Meyer dans [54] en utilisant le point de vue de l'analyse fonctionnelle et des espaces de fonctions. C'est cette voie que nous avons choisi de suivre dans cette thèse, car elle nous semble fournir une alternative intéressante du point de vue de la modélisation mathématique des textures par rapport aux méthodes énoncées ci-dessus.

2.2 Textures et espaces de fonctions oscillantes.

2.2.1 Au départ : l'algorithme de Rudin, Osher et Fatemi.

La nouvelle modélisation proposée par Y. Meyer dans [54] trouve son point de départ dans les travaux de Rudin, Osher et Fatemi [67] et leur algorithme dénomé ROF dans ce manuscrit, décrit au chapitre précédent. Pour mémoire, nous rappelons la fonctionnelle utilisée

$$F_{\lambda}^{ROF}(u) = J(u) + \lambda \|f - u\|_{L^2}^2 \quad (2.1)$$

Du point de vue de Rudin et al. l'image mesurée f est une image u «dégradée». L'algorithme se propose de restaurer l'image initiale des objets $u \in BV$ ($J(u) = \|u\|_{BV}$).

Si l'on s'intéresse maintenant à un point de vue de décomposition où f est une «vraie» image somme de deux composantes : u regroupant l'ensemble des objets et v étant la somme des textures et du bruit. Le problème consiste maintenant en l'extraction de u et v .

La figure 2.1 montre deux exemples d'application de l'algorithme ROF sur une image contenant de la texture (l'image de Barbara et une image d'un véhicule blindé en milieu naturel). La colonne de gauche représente les images originales, celle de droite, les composantes u obtenues.

Nous pouvons constater que l'algorithme supprime les phénomènes «oscillants» de l'image. A ce stade, considérons que nous avons une image non bruitée, il semble raisonnable de penser que si l'on modélise notre image initiale f comme étant la somme d'une image u contenant des objets et d'une image v de phénomènes oscillants alors la composante $v = f - u$ doit représenter les textures de l'image initiale. La figure 2.2 montre les composantes v correspondantes aux exemples de la figure 2.1.

Cette composante v contient bien des textures mais aussi des objets (les pieds de la table, les bras de Barbara, son visage, les livres, ...). Cette modélisation n'est donc pas satisfaisante. Yves Meyer fait la remarque suivante, si l'on réécrit 2.1 à l'aide de v , alors

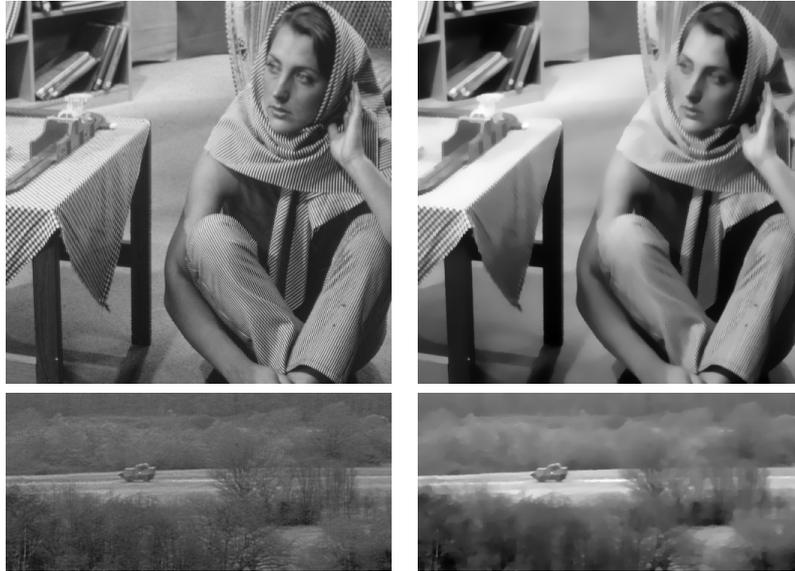


FIG. 2.1 – Algorithme ROF : la colonne de gauche contient les image originales, celle de droite la composante u .

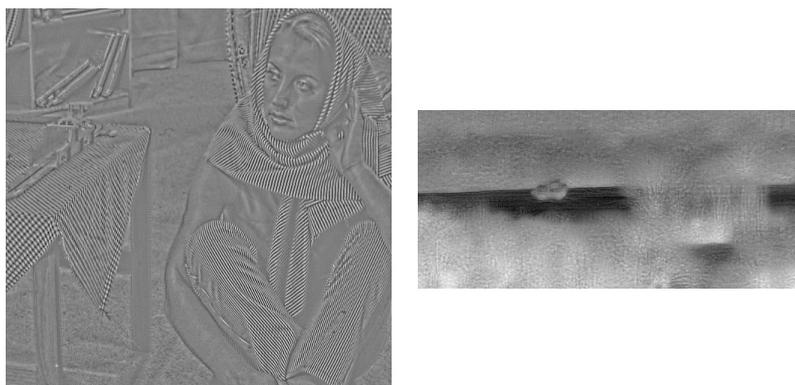


FIG. 2.2 – Algorithme ROF : composante «texture» $v = f - u$

$$F_\lambda^{ROF}(u, v) = J(u) + \lambda \|v\|_{L^2}^2 \quad \text{où } f = u + v \quad (2.2)$$

On peut donc remarquer que ce modèle ne décrit pas correctement les textures. En effet, de notre point de vue, les textures sont des fonctions oscillantes. Un exemple typique est le cas «toits de hangar» visible dans des images de type SPOT. On peut les modéliser comme suit :

$$g_N(x) = \cos(Nx_1)\theta(x) \quad (2.3)$$

où θ est la fonction indicatrice délimitant le hangar ($\theta(x) = 1$ si $x \in$ hangar, 0 sinon). Alors le calcul donne

$$\|g_N\|_{L^2} \approx \frac{1}{\sqrt{2}} \|\theta\|_{L^2} \quad (2.4)$$

qui est constante quelque soit la fréquence fixée par N . En revanche,

$$\|g_N\|_{BV} = \frac{N}{2\pi} \|\theta\|_{L^1}. \quad (2.5)$$

Ce qui signifie que plus la texture est oscillante, plus elle est rejetée de l'algorithme. L'algorithme ROF n'est donc pas adapté à un point de vue de la décomposition.

2.2.2 L'algorithme de Meyer.

Afin de pallier ce défaut, Yves Meyer dans [54] propose de changer d'espace fonctionnel pour modéliser la composante texture et préconise l'utilisation de l'espace \mathcal{BV}^* , qui est exactement le dual de l'espace \mathcal{BV} et correspond justement à un espace de fonctions oscillantes (une justification de ces résultats est disponible dans [43]). Nous rappelons quelques définitions et notations :

Définition 2.2.1 *Nous notons :*

- $BV = \{f \in L^2(\mathbb{R}^2), \nabla f = (f_1, f_2) \text{ est une mesure de Radon de masse totale finie}\}$,
- $\mathcal{BV} = \{f \in L^2(\mathbb{R}^2), \nabla f \in L^1(\mathbb{R}^2)\}$,
- $G = \mathcal{BV}^*$ espace dual défini comme l'ensemble des distributions tempérées $v = \partial_1 g_1 + \partial_2 g_2$ où $g_1 \in L^\infty(\mathbb{R}^2), g_2 \in L^\infty(\mathbb{R}^2)$. La norme est alors définie par (on note $g = (g_1, g_2)$, donc $v = \text{div } g$)

$$\|v\|_{\mathcal{BV}^*} = \|v\|_G = \inf_g \left\| \left(|g_1|^2 + |g_2|^2 \right)^{\frac{1}{2}} \right\|_{L^\infty} \quad (2.6)$$

- $E = \dot{B}_{-1, \infty}^\infty$, dual de $\dot{B}_{1, 1}^1$ (espaces de Besov définis à la section 1.1.4).

Rappelons, par ailleurs, la propriété d'inclusion suivante des espaces les uns dans les autres (voir [54])

Lemme 2.2.2 *On a les inclusions suivantes*

$$\dot{B}_{1,1}^1 \subset BV \subset L^2(\mathbb{R}^2) \subset G \subset E = \dot{B}_{-1,\infty}^\infty \quad (2.7)$$

Examinons deux exemples afin d'illustrer le choix de cette norme $\|\cdot\|_G$ dans le cadre des textures.

Exemple 1 : Supposons que

$$v(x) = \cos(\omega x + \varphi) \quad , \quad \omega \in \mathbb{R}^2 \quad (2.8)$$

alors

$$\|v\|_G = \frac{1}{|\omega|} \quad (2.9)$$

Pour s'en persuader, il faut observer que par construction, $\|v\|_G$ est invariante par rotation et par translation. De plus $\|\lambda v(\lambda \cdot)\|_G = \|v\|_G$ pour tout $\lambda > 0$. Il reste donc à vérifier que $\|\cos x_1\|_G < \infty$. Ceci est évident car $\cos x_1 = \frac{d}{dx_1} \sin x_1$.

Exemple 2 : Supposons que

$$v(x) = \cos(\omega x + \varphi)\theta(x) \quad (2.10)$$

où, par exemple, θ est la fonction indicatrice du carré unité. Alors on a encore, si $|\omega| \geq 1$,

$$\|v\|_G \leq \frac{C}{|\omega|} \quad (2.11)$$

mais ici si $0 \leq |\omega| \leq 1$, on a évidemment $\|v\|_G \leq C_0$ comme on le voit en observant que $L^2(\mathbb{R}^2) \subset (BV)^*$. L'inégalité isopérimétrique donne, en fait

$$\|f\|_G \leq \frac{1}{2\sqrt{\pi}} \|f\|_{L^2}. \quad (2.12)$$

Si donc une image f est la somme $g + \cos(\omega x + \varphi)\theta(x)$ où g est une image «simple» et θ est comme ci-dessus, alors on perd beaucoup plus que dans l'algorithme ROF en décidant que $\cos(\omega x + \varphi)\theta(x)$ est un «objet».

Y.Meyer propose donc de remplacer l'utilisation de la norme sur L^2 par la norme sur BV^* afin de ne pas pénaliser les fonctions oscillantes. Le nouveau modèle consiste donc à minimiser la nouvelle fonctionnelle

$$F_\lambda^{YM}(u, v) = J(u) + \lambda \|v\|_G \quad (2.13)$$

où $f = u + v$, $f \in G$, $u \in BV$, $v \in G$.

L'inconvénient de ce modèle est le calcul de la norme sur l'espace G . En effet, en regardant l'expression de cette norme (2.6), on s'aperçoit que le

calcul direct n'est pas possible du fait de la nécessité de connaître le champ de vecteurs g sous-jacent. L'implémentation numérique de l'algorithme n'est donc pas possible directement sous cette forme. Le problème est resté dans cet état quelques années, jusqu'aux travaux de S.Osher et L.Vese [58, 76] puis les travaux de J.F.Aujol, G.Aubert, L.Blanc-Féraud et A.Chambolle [2, 3, 20]. Avant de nous intéresser à ces travaux, remarquons que Y.Meyer propose aussi dans [54] d'utiliser d'autres espaces en lieu et place de G notamment les espaces de Besov définis à la section 1.1.4. Cette voie a récemment été explorée par les travaux de A.Haddad [43] et d'une autre manière par J.F.Aujol et A.Chambolle [3]. Avant d'explicitier ces travaux, nous allons donner quelques résultats théoriques sur ce type d'algorithme.

2.2.3 Propriétés de l'algorithme de Meyer

Nous commençons par rappeler le résultat important démontré par Y. Meyer [54].

Lemme 2.2.3 *Si $u \in L^2(\mathbb{R}^2)$ et $v \in BV(\mathbb{R}^2)$, alors*

$$\left| \int u(x)v(x)dx \right| \leq \|u\|_G \|v\|_{BV} \quad (2.14)$$

Maintenant, supposons que l'on part d'une image $f \in L^2(\mathbb{R}^2)$. On considère deux paramètres positifs λ et μ . On cherche à décomposer f en

$$f = u + v + w \quad (2.15)$$

en minimisant l'énergie $E(u, v)$ définie par

$$\|u\|_{BV} + \lambda \|v\|_{L^2}^2 + \mu \|w\|_G \quad (2.16)$$

Formellement, nous pouvons remarquer que le modèle ROF correspond à $\mu = +\infty$. Puisque $BV \subset L^2$, on a nécessairement $w \in L^2$. L'existence d'une décomposition optimale est facile à démontrer en utilisant la «méthode directe de D.Hilbert». En effet BV est un espace dual et de toute suite bornée $u_j \in BV$, on peut extraire une sous-suite convergente au sens des distributions vers $u \in BV$. Il en est de même pour L^2 et G . L'unicité n'est pas assurée, sauf en ce qui concerne la composante v qui est unique. Nous y reviendrons plus loin.

Le comportement de cet algorithme est donné par le théorème suivant.

Théorème 2.2.4 *Si $\|f\|_G \leq \frac{1}{2\lambda}$ et $\|f\|_{BV} \leq \frac{\mu}{2\lambda}$, alors $u = w = 0$ et la décomposition optimale est donc $f = 0 + f + 0$.*

Si $\|f\|_G \leq \frac{1}{2\lambda}$ mais $\|f\|_{BV} > \frac{\mu}{2\lambda}$, trois cas se présentent pour toute décomposition optimale $f = u + v + w$. On peut avoir

$$(1) \quad u = 0, \quad \|v\|_{BV} = \frac{\mu}{2\lambda}, \quad \|v\|_G < \frac{1}{2\lambda} \quad \text{et} \quad \langle v, w \rangle = \frac{\mu}{2\lambda} \|w\|_G,$$

$$(2) \quad w = 0, \quad \|v\|_{BV} \leq \frac{\mu}{2\lambda}, \quad \|v\|_G = \frac{1}{2\lambda} \quad \text{et} \quad \langle u, v \rangle = \frac{1}{2\lambda} \|u\|_{BV} \quad \text{et enfin,}$$

$$(3) \quad \|v\|_{BV} = \frac{\mu}{2\lambda}, \quad \|v\|_G = \frac{1}{2\lambda}, \quad \langle u, v \rangle = \frac{1}{2\lambda} \|u\|_{BV} \quad \text{et} \quad \langle v, w \rangle = \frac{\mu}{2\lambda} \|w\|_G.$$

Réciproquement tout triplet (u, v, w) vérifiant soit (1), soit (2), soit (3) est optimal pour $f = u + v + w$ et pour les valeurs de λ et μ associées.

On peut critiquer cet énoncé qui annonce ce qui va se produire en examinant les résultats que l'on annonce. Mais il y a des cas où le théorème 2.2.4 reste intéressant. Supposons par exemple que l'on ait, en outre, $\|f\|_G < \frac{\pi}{\lambda\mu}$. Alors la décomposition optimale est décrite par (1). En effet on compare la décomposition optimale $f = u + v + w$ à la décomposition triviale $f = 0 + 0 + f$. On a donc

$$\|u\|_{BV} + \lambda \|v\|_{L^2}^2 + \mu \|w\|_G \leq \mu \|f\|_G \quad (2.17)$$

ce qui entraîne

$$\|v\|_{L^2} \leq \sqrt{\frac{\mu}{\lambda} \|f\|_G} \quad (2.18)$$

Mais $\|v\|_G \leq \frac{1}{2\sqrt{\pi}} \|v\|_{L^2}$ et l'hypothèse faite entraîne $\|v\|_G < \frac{1}{2\lambda}$.

Avant de démontrer ce théorème, examinons le deuxième résultat important suivant.

Théorème 2.2.5 *Si $0 < \mu < 4\pi$, alors la décomposition optimale $f = u + v + w$ vérifie $u = 0$.*

Preuve:

Pour établir le théorème 2.2.5, nous partons du lemme suivant.

Lemme 2.2.6 *Il vient, pour toute fonction $f \in BV$,*

$$\|f\|_{L^2} \leq \frac{1}{2\sqrt{\pi}} \|f\|_{BV} \quad (2.19)$$

et il en résulte que

$$\|f\|_G \leq \frac{1}{2\sqrt{\pi}} \|f\|_{L^2} \leq \frac{1}{4\pi} \|f\|_{BV} \quad (2.20)$$

Ce lemme est une conséquence directe de l'inégalité isopérimétrique.

Revenons au théorème 2.2.5. On gèle v et l'on laisse flotter librement u . On pose $u + w = \sigma$ et l'on a donc $\sigma = f - v$. On cherche d'abord à minimiser $\|u\|_{BV} + \mu\|\sigma - u\|_G$. Il vient, sous l'hypothèse $0 < \mu < 4\pi$,

$$\|u\|_{BV} + \mu\|\sigma - u\|_G \geq 4\pi\|u\|_G + \mu\|\sigma - u\|_G \quad (2.21)$$

$$\geq \mu\|u\|_G + \mu\|\sigma - u\|_G \quad (2.22)$$

$$\geq \mu\|\sigma\|_G \quad (2.23)$$

Si par ailleurs, u n'est pas identiquement nulle, alors $\|u\|_G > 0$ et l'on a

$$\|u\|_{BV} + \mu\|\sigma - u\|_G > \mu\|\sigma\|_G \quad (2.24)$$

Le minimum sera donc atteint pour $u = 0$. ■

Examinons maintenant la preuve du théorème 2.2.4. Nous commençons par remarquer que le cas $0 < \mu < 4\pi$ revient à minimiser

$$\lambda\|v\|_{L^2}^2 + \mu\|f - v\|_G \quad (2.25)$$

On applique alors le résultat général suivant

Lemme 2.2.7 *Si l'on cherche à minimiser (pour E un espace de Banach arbitraire)*

$$\|u\|_E + \lambda\|v\|_{L^2}^2 \quad (2.26)$$

sur toutes les décompositions $f = u + v$ de $f \in L^2(\mathbb{R}^2)$, alors deux cas se présentent (on notera $\|\cdot\|_{E^}$ la norme dans le dual E^* de E)*

(1) *si $\|f\|_{E^*} \leq \frac{1}{2\lambda}$, alors le minimum est atteint pour $u = 0$ et $v = f$,*

(2) *si $\|f\|_{E^*} > \frac{1}{2\lambda}$, alors le minimum est atteint pour un v tel que $\|v\|_{E^*} = \frac{1}{2\lambda}$ et $\langle u, v \rangle = \frac{1}{2\lambda}\|u\|_E$.*

Appliquons le lemme 2.2.7 à $\lambda\|v\|_{L^2}^2 + \mu\|f - v\|_G$. Si $\|f\|_{BV} \leq \frac{\mu}{2\lambda}$, alors le minimum est atteint si $v = 0$.

Conclusion partielle : si $0 < \mu < 4\pi$ alors $u = 0$. Si, en outre, $\|f\|_{BV} \leq \frac{\mu}{2\lambda}$, alors $u = v = 0$.

La preuve du théorème est basée sur le lemme suivant :

Lemme 2.2.8 *Soient E_1 et E_2 deux espaces de Banach inclus dans un même espace vectoriel E . On définit alors l'espace de Banach E_3 qui se compose de toutes les sommes*

$$z = x + y, \quad x \in E_1, \quad y \in E_2 \quad (2.27)$$

et qui est muni de la norme

$$\|z\|_{E_3} = \inf\{\|x\|_{E_1} + \|y\|_{E_2}\} \quad (2.28)$$

où la borne inférieure porte sur toutes les décompositions de z .

Alors E_3 est le plus petit espace de Banach contenant E_1 et E_2 . Par ailleurs, E_3^ est le plus grand espace de Banach inclus dans E_1^* et E_2^* . En d'autres termes $E_3^* = E_1^* \cap E_2^*$ et la norme de g dans E_3^* est*

$$\|g\|_{E_3^*} = \sup\{\|g\|_{E_1^*}, \|g\|_{E_2^*}\}. \quad (2.29)$$

Nous devons minimiser $E(u, v) = \|u\|_{BV} + \lambda\|v\|_{L^2}^2 + \mu\|w\|_G$ sous la contrainte $f = u + v + w$. Fixons provisoirement v et optimisons sur u . Nous sommes conduits à poser $\sigma = u + w$ et

$$\|\sigma\| = \inf\{\|u\|_{BV} + \mu\|w\|_G; \sigma = u + w\} \quad (2.30)$$

Ceci étant, nous avons

$$\inf_{u,v} E(u, v) = \inf_{\sigma} \{\|\sigma\| + \lambda\|f - \sigma\|_{L^2}^2\} \quad (2.31)$$

Pour traiter le problème de minimisation $\|\sigma\| + \lambda\|f - \sigma\|_{L^2}^2$, nous appliquons le lemme 2.2.7. La norme duale de $\|\cdot\|$ est

$$\|\cdot\|_* = \sup\left\{\|\cdot\|_G, \frac{1}{\mu}\|\cdot\|_{BV}\right\} \quad (2.32)$$

On a donc

Lemme 2.2.9 *Si $\|f\|_G \leq \frac{1}{2\lambda}$ et $\|f\|_{BV} \leq \frac{\mu}{2\lambda}$, alors le minimum de $E(u, v)$ est atteint pour $u = w = 0$ et vaut donc $\lambda\|f\|_{L^2}^2$.*

Preuve:

En effet, étant donné que $\|f\|_G \leq \frac{1}{2\lambda}$ et $\|f\|_{BV} \leq \frac{\mu}{2\lambda}$, on a

$$\|f\|_* = \sup \left\{ \|f\|_G; \frac{1}{\mu} \|f\|_{BV} \right\} \leq \frac{1}{2\lambda}. \quad (2.33)$$

Celà nous ramène au cas (1) du lemme 2.2.7. Donc $\sigma = 0$ et $v = f$ ce qui impose par définition de la norme $\|\cdot\|$ que $u = w = 0$

■

Ceci conclut le premier point du théorème 2.2.4. Passons maintenant au cas suivant où

$$\|f\|_G \leq \frac{1}{2\lambda} \quad \text{et} \quad \|f\|_{BV} > \frac{\mu}{2\lambda}. \quad (2.34)$$

(Observons que $\|f\|_G \leq \frac{1}{4\pi} \|f\|_{BV}$. On ne peut donc avoir $\|f\|_G > \frac{1}{2\lambda}$ et $\|f\|_{BV} \leq \frac{\mu}{2\lambda}$ si $0 < \mu \leq 4\pi$).

Sous l'hypothèse 2.34, le lemme 2.2.7 nous assure que le σ optimal vérifie

$$\|v\|_* = \frac{1}{2\lambda} \quad \text{et} \quad \langle v, \sigma \rangle = \frac{1}{2\lambda} \|\sigma\|. \quad (2.35)$$

Par ailleurs $\|\sigma\| = \|u\|_{BV} + \mu\|w\|_G$, car u et w sont optimisés. On a soit

$$\|v\|_{BV} = \frac{\mu}{2\lambda} \quad \text{et} \quad \|v\|_G < \frac{1}{2\lambda} \quad (2.36)$$

soit

$$\|v\|_{BV} \leq \frac{\mu}{2\lambda} \quad \text{et} \quad \|v\|_G = \frac{1}{2\lambda} \quad (2.37)$$

Considérons d'abord le premier cas. On a alors

Lemme 2.2.10 *Si 2.34 et 2.36 ont lieu, alors la décomposition optimale $f = u + v + w$ vérifie $u = 0$ et $\langle v, w \rangle = \frac{\mu}{2\lambda} \|w\|_G$*

Preuve:

En effet, le lemme 2.2.7 nous apprend que

$$\langle v, u + w \rangle = \frac{1}{2\lambda} (\|u\|_{BV} + \mu\|w\|_G). \quad (2.38)$$

Mais

$$\langle v, w \rangle \leq \|v\|_{BV} \|w\|_G = \frac{\mu}{2\lambda} \|w\|_G \quad (2.39)$$

tandis que

$$\langle v, u \rangle \leq \|v\|_G \|u\|_{BV} < \frac{1}{2\lambda} \|u\|_{BV} \quad (2.40)$$

En additionnant ces deux inégalités, on doit tomber sur 2.38. Ces deux inégalités doivent être des égalités.

Cela entraîne $u = 0$ et $\langle v, w \rangle = \frac{\mu}{2\lambda} \|w\|_G$.

■

Passons maintenant au second cas. On distingue

$$\|v\|_{BV} < \frac{\mu}{2\lambda} \quad \text{et} \quad \|v\|_G = \frac{1}{2\lambda} \quad (2.41)$$

et

$$\|v\|_{BV} = \frac{\mu}{2\lambda} \quad \text{et} \quad \|v\|_G = \frac{1}{2\lambda} \quad (2.42)$$

Dans le cas 2.41, on reprend l'argumentation précédente et l'on conclut cette fois à $w = 0$. Alors $f = u + v$ est la décomposition optimale et l'on a donc (voir les résultats sur Osher-Rudin) $\|v\|_G = \frac{1}{2\lambda}$ et $\langle u, v \rangle = \|u\|_{BV} \frac{1}{2\lambda}$. Examinons la réciproque. Elle est fournie par le lemme suivant

Lemme 2.2.11 *Supposons que l'on ait $\|v_0\|_G = \frac{1}{2\lambda}$ et $\|v_0\|_{BV} \leq \frac{\mu}{2\lambda}$ avec $\langle u_0, v_0 \rangle = \frac{1}{2\lambda} \|u_0\|_{BV}$. Posons $f_0 = u_0 + v_0$. Alors pour toute fonction $\alpha \in BV$ et toute fonction $w \in L^2(\mathbb{R}^2)$, on a*

$$\|u_0 + \alpha\|_{BV} + \lambda \|v_0 - \alpha - w\|_{L^2}^2 + \mu \|w\|_G \geq \|u_0\|_{BV} + \lambda \|v_0\|_{L^2}^2. \quad (2.43)$$

Cela signifie que pour une telle fonction f_0 et pour ces valeurs de λ et de μ , la décomposition $u_0 + v_0$ est optimale.

Preuve:

Vérifions 2.43. On divise par $2\lambda = \|v_0\|_G^{-1}$ et l'on obtient

$$\begin{aligned} & \|u_0 + \alpha\|_{BV} \|v_0\|_G + \frac{1}{2} \|v_0 - \alpha - w\|_{L^2}^2 + \|v_0\|_{BV} \|w\|_G \\ &= \|u_0 + \alpha\|_{BV} \|v_0\|_G + \frac{1}{2} \|v_0\|_{L^2}^2 - \langle v_0, \alpha \rangle + \frac{1}{2} \|\alpha\|_{L^2}^2 - \langle w, v_0 - \alpha \rangle \\ &+ \frac{1}{2} \|w\|_{L^2}^2 + \|v_0\|_{BV} \|w\|_G \\ &\geq \langle u_0, v_0 \rangle + \langle \alpha, v_0 \rangle + \frac{1}{2} \|v_0\|_{L^2}^2 - \langle \alpha, v_0 \rangle + \frac{1}{2} \|\alpha\|_{L^2}^2 - \langle w, v_0 - \alpha \rangle \\ &+ \frac{1}{2} \|w\|_{L^2}^2 + \|v_0\|_{BV} \|w\|_G \\ &= \frac{1}{2\lambda} \|u_0\|_{BV} + \frac{1}{2} \|v_0\|_{L^2}^2 + \frac{1}{2} \|\alpha\|_{L^2}^2 - \langle w, v_0 - \alpha \rangle + \frac{1}{2} \|w\|_{L^2}^2 + \|v_0\|_{BV} \|w\|_G \\ &= \frac{1}{2\lambda} \|u_0\|_{BV} + \frac{1}{2} \|v_0\|_{L^2}^2 + \frac{1}{2} \|\alpha + w\|_{L^2}^2 - \langle w, v_0 \rangle + \|v_0\|_{BV} \|w\|_G \\ &\geq \frac{1}{2\lambda} \|u_0\|_{BV} + \frac{1}{2} \|v_0\|_{L^2}^2. \end{aligned}$$

■

Remarquons que si l'on a égalité, on a nécessairement $\alpha = -w$ et $\langle w, v_0 \rangle = \|v_0\|_{BV} \|w\|_G$. On a également $\mu = 2\lambda \|v_0\|_{BV}$. Revenons alors à 2.43 qui s'écrit

$$\|u_0 - w\|_{BV} + \lambda \|v_0\|_{L^2}^2 + \mu \|w\|_G \geq \|u_0\|_{BV} + \lambda \|v_0\|_{L^2}^2 \quad (2.44)$$

ou encore

$$\|u_0 - w\|_{BV} + 2\lambda \|v_0\|_{BV} \|w\|_G \geq \|u_0\|_{BV} \quad (2.45)$$

c'est à dire

$$\|u_0 - w\|_{BV} \|v_0\|_G + \langle w, v_0 \rangle \geq \|u_0\|_{BV} \frac{1}{2\lambda}. \quad (2.46)$$

Mais

$$\|u_0 - w\|_{BV} \|v_0\|_G \geq \langle u_0 - w, v_0 \rangle = \langle u_0, v_0 \rangle - \langle w, v_0 \rangle. \quad (2.47)$$

Enfin

$$\langle u_0, v_0 \rangle = \|u_0\|_{BV} \frac{1}{2\lambda}. \quad (2.48)$$

Si l'on a égalité, on doit avoir aussi

$$\|u_0 - w\|_{BV} \|v_0\|_G = \langle u_0 - w, v_0 \rangle. \quad (2.49)$$

Passons maintenant à la réciproque du lemme 2.2.10.

Lemme 2.2.12 *Supposons que $f = v_0 + w_0$ avec $\|v_0\|_{BV} = \frac{\mu}{2\lambda}$, $\|v_0\|_G < \frac{1}{2\lambda}$ et $\langle v_0, w_0 \rangle = \frac{\mu}{2\lambda} \|w_0\|_G$. Alors $f = v_0 + w_0$ est la décomposition optimale.*

Preuve:

Posons $w = w_0 + \tilde{w}$ et $v = v_0$ alors le problème 2.16 devient

$$\|u\|_{BV} + \lambda \|v_0 - u - \tilde{w}\|_{L^2}^2 + \mu \|w_0 + \tilde{w}\|_G = J(u, \tilde{w}). \quad (2.50)$$

Il vient alors

$$\|w_0 + \tilde{w}\|_G \|v_0\|_{BV} \geq \langle w_0 + \tilde{w}, v_0 \rangle = \langle w_0, v_0 \rangle + \langle \tilde{w}, v_0 \rangle \quad (2.51)$$

or par hypothèse $\langle w_0, v_0 \rangle = \frac{\mu}{2\lambda} \|w_0\|_G$. Donc

$$\|w_0 + \tilde{w}\|_G \|v_0\|_{BV} \geq \frac{\mu}{2\lambda} \|w_0\|_G + \langle \tilde{w}, v_0 \rangle \quad (2.52)$$

et comme $\|v_0\|_{BV} = \frac{\mu}{2\lambda}$, on a

$$\|w_0 + \tilde{w}\|_G \geq \|w_0\|_G + \frac{2\lambda}{\mu} \langle \tilde{w}, v_0 \rangle. \quad (2.53)$$

Par ailleurs

$$\begin{aligned}
 \lambda\|v_0 - u - \tilde{w}\|_{L^2}^2 &= \lambda\|v_0\|_{L^2}^2 - 2\lambda\langle\tilde{w}, v_0\rangle - 2\lambda\langle u, v_0\rangle + \lambda\|u + \tilde{w}\|_{L^2}^2 \\
 &= \langle v_0 - u - \tilde{w}, v_0 - u - \tilde{w}\rangle = \langle v_0, v_0\rangle - \langle v_0, u\rangle - \langle v_0, \tilde{w}\rangle - \langle u, v_0\rangle \\
 &\quad + \langle u, u\rangle + \langle u, \tilde{w}\rangle - \langle \tilde{w}, v_0\rangle + \langle \tilde{w}, u\rangle + \langle \tilde{w}, \tilde{w}\rangle \\
 &= \|v_0\|_{L^2}^2 - 2\langle u, v_0\rangle - 2\langle v_0, \tilde{w}\rangle + 2\langle u, \tilde{w}\rangle + \langle u, u\rangle + \langle \tilde{w}, \tilde{w}\rangle
 \end{aligned}$$

or $2\langle u, \tilde{w}\rangle + \langle u, u\rangle + \langle \tilde{w}, \tilde{w}\rangle = \|u + \tilde{w}\|_{L^2}^2$, ce qui entraîne

$$J(u, \tilde{w}) \geq \lambda\|v_0\|_{L^2}^2 + \mu\|w_0\|_G + \|u\|_{BV} - 2\lambda\langle u, v_0\rangle + \lambda\|u + \tilde{w}\|_{L^2}^2. \quad (2.54)$$

Pour conclure, il suffit d'observer que

$$|\langle u, v_0\rangle| \leq \|u\|_{BV}\|v_0\|_G < \frac{1}{2\lambda}\|u\|_{BV} \quad (2.55)$$

et

$$\|u + w\|_{L^2}^2 = \|f - v_0 - w_0\|_{L^2}^2 = 0 \quad \text{car par hypothèse } f = v_0 + w_0. \quad (2.56)$$

■

A ce stade, nous avons donc obtenu les points (1) et (2) du théorème 2.2.4. Pour terminer la preuve, il nous faut établir (3). La partie directe s'obtient par la méthode utilisée pour démontrer (1) ou (2). Voici l'étude de la réciproque. On doit calculer

$$\|u + \alpha\|_{BV} + \lambda\|v + \beta\|_{L^2}^2 + \mu\|w - \alpha - \beta\|_G \quad (2.57)$$

Sachant que u, v et w vérifient (3). Les fonctions $\alpha \in BV$ et $\beta \in L^2$ sont arbitraires. Comme $\|v\|_{BV} = \frac{\mu}{2\lambda}$ et $\|v\|_{BV}\|w - \alpha - \beta\|_G \geq \langle v, w - \alpha - \beta\rangle$, on a

$$\mu\|w - \alpha - \beta\|_G \geq 2\lambda(\langle v, w\rangle - \langle v, \alpha\rangle - \langle v, \beta\rangle). \quad (2.58)$$

Par ailleurs comme $\|v\|_G = \frac{1}{2\lambda}$, on a aussi

$$\|u + \alpha\|_{BV} \geq 2\lambda(\langle u, v\rangle + \langle \alpha, v\rangle) \quad (2.59)$$

Finalement

$$\lambda\|v + \beta\|_{L^2}^2 = \lambda\|v\|_{L^2}^2 + 2\lambda\langle v, \beta\rangle + \lambda\|\beta\|_{L^2}^2, \quad (2.60)$$

$$\langle v, w\rangle = \frac{\mu}{2\lambda}\|w\|_G \quad \text{et} \quad (2.61)$$

$$\langle v, u\rangle = \frac{1}{2\lambda}\|u\|_{BV} \quad (2.62)$$

ce qui permet de conclure, tous les termes disparaissent et il reste (le minimum étant atteint pour $\beta = 0$)

$$\|u\|_{BV} + \lambda\|v\|_{L^2}^2 + \mu\|w\|_G. \quad (2.63)$$

Ceci conclut la preuve du théorème 2.2.4.

Remarquons que nous n'avons pas unicité de la décomposition, le contre exemple suivant permet de s'en persuader.

On appelle θ la fonction indicatrice du disque unité. On a alors $\|\theta\|_G = \frac{1}{2}$ et l'on pose $f = 3\theta$. On considère les deux décompositions $f = \theta + \theta + \theta$ et $f = 2\theta + \theta + 0$. On pose $\lambda = 1$ et $\mu = 4\pi$.

Pour la première, on a bien $\|v\|_{BV} = \frac{\mu}{2\lambda}$, $\|v\|_G = \frac{1}{2\lambda}$, $\langle u, v \rangle = \pi = \frac{\|u\|_{BV}}{2\lambda}$ et $\langle v, w \rangle = \pi = \frac{\mu}{2\lambda} \|w\|_G$.

Pour la seconde, on a bien $\|v\|_{BV} \leq \frac{\mu}{2\lambda}$, $\langle u, v \rangle = \frac{1}{2\lambda} \|u\|_{BV}$ et $\|v\|_G = \frac{1}{2\lambda}$. Ces deux décompositions respectent bien le théorème 2.2.4.

Applications.

Le premier exemple est une route longue et fine : $f(x_1, x_2) = 1$ si $0 \leq x_1 \leq L$, $0 \leq x_2 \leq \epsilon$ où $L \gg 1$ et $0 < \epsilon \ll 1$. Alors $\|f\|_G \leq \frac{\epsilon}{2}$ tandis que $\|f\|_{BV} = 2(L + \epsilon)$. Nous sommes alors dans le cadre du théorème 2.2.4. Plus précisément, nous sommes dans le cas 1 du théorème 2.2.4 si $\frac{\epsilon}{2} < \sqrt{\frac{\pi}{\lambda\mu}}$ et $\mu < 4\lambda(L + \epsilon)$, c'est à dire si L est assez grand devant μ . Alors $u = 0$ et l'on a $\|w\|_{BV} \geq \|f\|_{BV} - \|v\|_{BV} \geq 2(L + \epsilon) - \frac{\mu}{2\lambda}$ qui est grand. Dans ce cas la composante w est la plus importante. On pourra aussi observer que $\|w\|_G \leq \|f\|_G \leq \frac{\epsilon}{2}$.

Enfin, comme nous l'apprend A.Chambolle, v est défini par le problème variationnel

$$\inf \left\{ \|f - v\|_{L^2}^2 \quad ; \quad \|v\|_{BV} \leq \frac{\mu}{2\lambda} \right\}. \quad (2.64)$$

Supposons que l'on augmente μ de sorte que $\|f\|_{BV} = 2(L + \epsilon) < \frac{\mu}{2\lambda}$. Alors la décomposition optimale devient $f = 0 + f + 0$ et w a disparu.

Dans le second exemple $f(x_1, x_2) = \cos(Nx_1)\theta(x_1, x_2)$ où θ est la fonction indicatrice du carré unité. Alors d'après 2.11, $\|f\|_G \leq \frac{C}{N}$. Nous avons également $\|f\|_{BV} \simeq N$. Nous sommes à nouveau dans le cadre du théorème 2.2.4 si N est très grand. Plus précisément, c'est la conclusion (1) qui fournit la décomposition lorsque l'on a

$$\frac{C}{2N} < \sqrt{\frac{\pi}{\lambda\mu}} \quad \text{et} \quad \mu \leq C'N, \quad (2.65)$$

c'est à dire si N est grand devant μ et devant $\sqrt{\lambda\mu}$.

2.2.4 L'algorithme de Osher-Vese.

Dans [58, 76], les auteurs proposent d'apporter quelques modifications au modèle décrit en 2.13 afin de pouvoir rechercher numériquement une solution.

Le nouveau modèle proposé est le suivant

$$F_{\lambda,\mu,p}^{OV}(u, g) = J(u) + \lambda \|f - (u + \operatorname{div} g)\|_{L^2}^2 + \mu \left\| \sqrt{g_1^2 + g_2^2} \right\|_{L^p} \quad (2.66)$$

L'idée est d'utiliser la propriété classique

$$\forall f \in L^\infty(\Omega), \|f\|_{L^\infty} = \lim_{p \rightarrow \infty} \|f\|_{L^p} \quad (2.67)$$

La reformulation vise alors à rechercher la composante $u \in BV$ et le champ de vecteurs $g = (g_1, g_2) \in L^\infty \times L^\infty$ tel que $v = \operatorname{div} g$. D'autre part, afin de pouvoir effectuer des calculs, la norme $\|\cdot\|_{L^\infty}$ est remplacée par une norme $\|\cdot\|_{L^p}$ (en considérant que $p \rightarrow \infty$ pour retrouver l'expression de la norme sur l'espace G). Le terme supplémentaire $\|f - (u + \operatorname{div} g)\|_{L^2}^2$ permet de s'assurer que l'on aura bien la contrainte $f = u + v$.

Grâce à cette nouvelle formulation, il est possible d'utiliser le formalisme d'Euler-Lagrange dérivant de $F_{\lambda,\mu,p}^{OV}(u, g)$. On obtient alors le système de trois équations aux dérivées partielles couplées suivant

$$\begin{cases} u = f - \partial_x g_1 - \partial_y g_2 + \frac{1}{\lambda} \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \\ \mu \left(\left\| \sqrt{g_1^2 + g_2^2} \right\|_{L^p} \right)^{1-p} \left(\sqrt{g_1^2 + g_2^2} \right)^{p-2} g_1 = 2\lambda \left[\frac{\partial}{\partial x} (u - f) + \partial_{xx}^2 g_1 + \partial_{xy}^2 g_2 \right] \\ \mu \left(\left\| \sqrt{g_1^2 + g_2^2} \right\|_{L^p} \right)^{1-p} \left(\sqrt{g_1^2 + g_2^2} \right)^{p-2} g_2 = 2\lambda \left[\frac{\partial}{\partial y} (u - f) + \partial_{xy}^2 g_1 + \partial_{yy}^2 g_2 \right] \end{cases} \quad (2.68)$$

La discrétisation de ces équations ne pose pas de problèmes particuliers, leurs expressions sont disponibles dans [76]. La figure 2.3 montre les résultats obtenus grâce à l'algorithme de Osher-Vese ($p = 5$, $\mu = 0.1$ dans les deux cas, $\lambda = 0.1$ dans le cas de Barbara et $\lambda = 0.001$ dans le cas du blindé).

Si l'on compare ces résultats à ceux obtenus avec l'algorithme ROF, on voit clairement que la composante v ne contient effectivement plus que des textures. Les objets tels que le pied de la table, les bras de Barbara ont été très nettement (mais pas complètement) rejetés de cette composante. Cela nous conforte dans l'idée de modéliser les textures par des espaces de fonctions oscillantes. Toutefois, d'un point de vue pratique, on peut constater que l'algorithme n'est pas toujours stable, ceci étant dû en partie au fait que l'on utilise des EDP couplées du second ordre. D'autre part, la propriété sous-jacente de la convergence de la norme sur L^p vers la norme sur L^∞ quand $p \rightarrow \infty$ utilisée au départ par les auteurs est ensuite non-respectée par ceux-ci puisqu'ils préconisent la valeur $p = 1$ dans leurs expérimentations et en faisant remarquer que cette valeur convenait très bien. Nous avons nous-même expérimenté cet algorithme en faisant varier la valeur de p , on peut effectivement noter de très faibles différences sur les résultats à partir de $p = 5$ mais surtout un accroissement de l'instabilité numérique de l'algorithme.

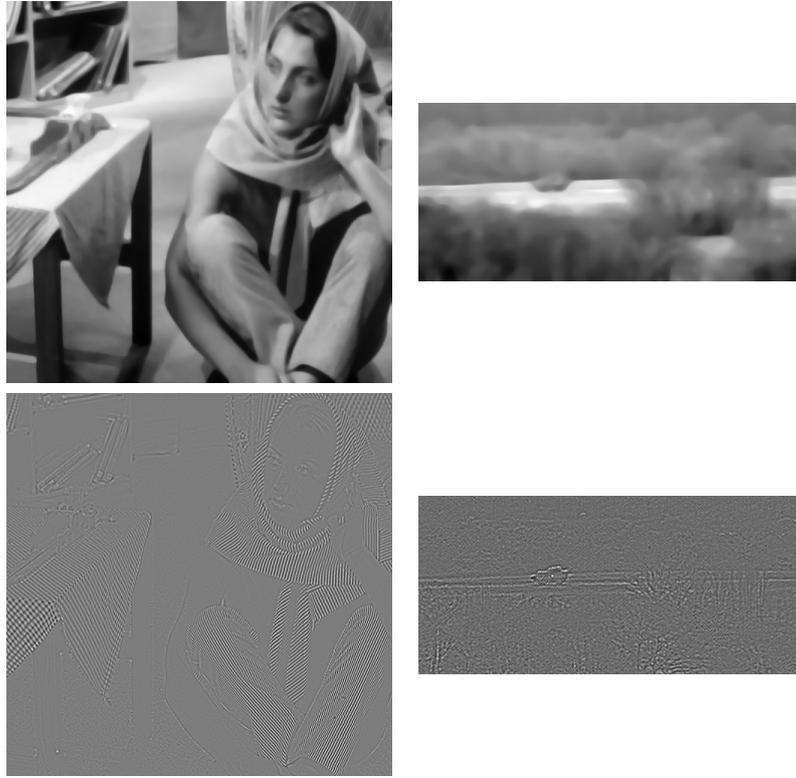


FIG. 2.3 – Algorithme Osher-Vese

2.2.5 L'algorithme de Aujol

Dans [1, 2], J.F.Aujol, G.Aubert, L.Blanc-Féraud et A.Chambolle proposent aussi une nouvelle formulation du problème. Pour cela, les auteurs considèrent un domaine fini Ω . La nouvelle fonctionnelle est la suivante

$$F_{\lambda, \mu}^{AU}(u, v) = J(u) + J^*\left(\frac{v}{\mu}\right) + (2\lambda)^{-1} \|f - u - v\|_{L^2}^2 \quad (2.69)$$

où

$$(u, v) \in BV(\Omega) \times G_\mu(\Omega) \quad (2.70)$$

L'ensemble G_μ est le sous-ensemble de G défini par $\|v\|_G \leq \mu$. La fonction J^* représente la fonction caractéristique sur l'espace G_1

$$J^*(v) = \begin{cases} 0 & \text{si } v \in G_1 \\ +\infty & \text{sinon} \end{cases} \quad (2.71)$$

Il est possible de montrer [1, 2, 20] que J^* est l'opérateur dual de J ce qui correspond bien à la notion de dualité entre les espaces BV et G . Reste la

question de la résolution numérique de $F_{\lambda,\mu}^{AU}(u, v)$. Une solution originale, basée sur les travaux de A.Chambolle [20], utilisant un opérateur de projection sur l'espace G_μ est adoptée (cet opérateur sera noté P_{G_μ}). L'annexe A, donne les résultats obtenus par A.Chambolle et notamment l'algorithme permettant de calculer la projection d'une fonction sur l'espace G_μ . Un théorème spécifiant les conditions de convergence de l'algorithme de projection est aussi donné en annexe A (théorème A.3.1).

La recherche du couple (\hat{u}, \hat{v}) solution de la minimisation de $F_{\lambda,\mu}^{AU}(u, v)$ est obtenue de manière itérative

– on fixe v et on recherche u solution de

$$\inf_u (J(u) + (2\lambda)^{-1} \|f - u - v\|_{L^2}^2) \quad (2.72)$$

– on fixe u et on recherche v solution de

$$\inf_v J^* \left(\frac{v}{\mu} \right) + \|f - u - v\|_{L^2}^2 \quad (2.73)$$

D'après les résultats démontrés par A.Chambolle (rappelés en annexe A), la solution de 2.72 est donnée par

$$\hat{u} = f - \hat{v} - P_{G_\lambda}(f - \hat{v}) \quad (2.74)$$

et la solution de 2.73 est donnée par

$$\hat{v} = P_{G_\mu}(f - \hat{u}) \quad (2.75)$$

On obtient donc l'algorithme numérique correspondant suivant

1. Initialisation :

$$u_0 = v_0 = 0$$

2. Itérations :

$$\begin{aligned} v_{n+1} &= P_{G_\mu}(f - u_n) \\ u_{n+1} &= f - v_{n+1} - P_{G_\lambda}(f - v_{n+1}) \end{aligned}$$

3. On arrête l'algorithme si

$$\max(|u_{n+1} - u_n|, |v_{n+1} - v_n|) \leq \epsilon$$

ou si l'on atteint un nombre maximal d'itérations prescrit.

Dans [1, 2], les auteurs montrent l'existence d'une solution, la convergence de l'algorithme ainsi que le lien avec le modèle proposé par Y. Meyer (le

paramètre λ doit être proche de 0 et de plus $\lambda < \mu$). De récents travaux de Aujol [5] proposent une méthode permettant de sélectionner une valeur optimale de λ au sens d'un certain critère. La question de l'ordre de calcul choisi dans le déroulement de l'algorithme (v_{n+1} avant u_{n+1}) est aussi abordée. Les auteurs démontrent que le choix pris ci-dessus permet de se départir d'une borne inférieure sur le temps de convergence.

La figure 2.4 illustre les résultats obtenus par l'algorithme de Aujol ($\mu = 100$ dans les deux cas, $\lambda = 1$ pour Barbara et $\lambda = 10$ pour le blindé).

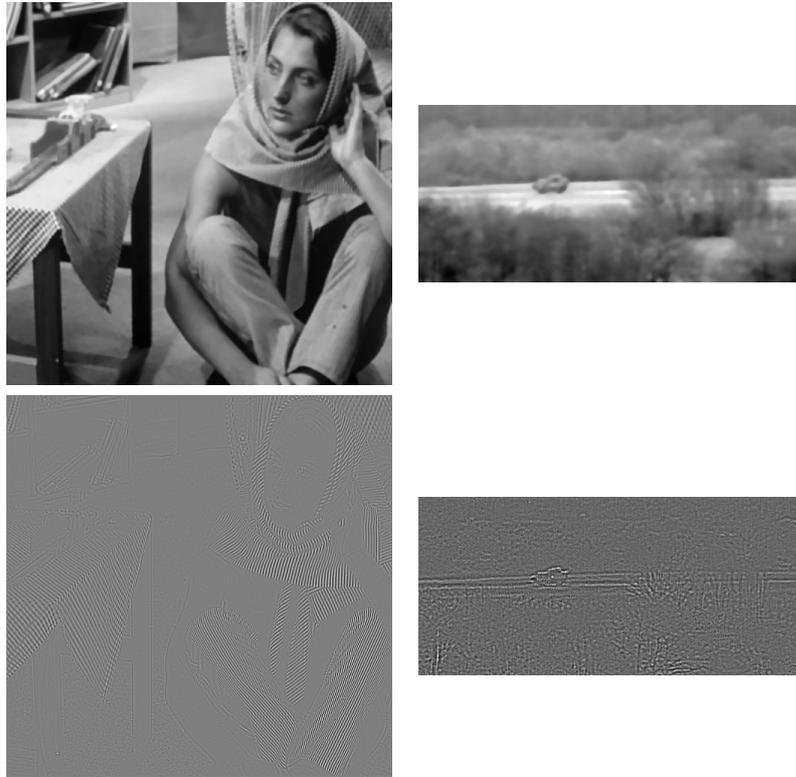


FIG. 2.4 – Algorithme de Aujol

On peut constater que l'algorithme permet bien d'extraire les textures de la même manière que l'algorithme de Osher-Vese. De plus, cet algorithme a quelques avantages par rapport à celui de Osher-Vese :

- aucun problème de stabilité et de convergence à partir du moment où l'on respecte les conditions énoncées par le théorème A.3.1,
- facile à implémenter (ne nécessite que quelques lignes de code).

Afin de faire le parallèle avec les résultats énoncés dans la section 2.2.3, examinons d'un point de vue théorique l'algorithme de J.F.Aujol. Il s'agit

d'optimiser la décomposition $f = u + v + w$ de $f \in L^2(\mathbb{R}^2)$ en minimisant la fonctionnelle

$$\|u\|_{BV} + \lambda\|v\|_{L^2}^2 \quad (2.76)$$

sous la contrainte $\|w\|_G \leq \mu$. Dans l'algorithme précédent, le coefficient μ multipliait $\|w\|_G$ et la fonctionnelle à minimiser était

$$\|u\|_{BV} + \lambda\|v\|_{L^2}^2 + \mu\|w\|_G \quad (2.77)$$

Il est donc nécessaire de changer μ en $\frac{1}{\mu}$ pour passer d'un problème à l'autre. Les deux problèmes ont une structure semblable, mais nous allons voir que les résultats sont tout à fait différents. On posera donc

$$\tilde{E}(u, v) = \|u\|_{BV} + \lambda\|v\|_{L^2}^2 + \frac{1}{\mu}\|w\|_G \quad (2.78)$$

Le premier cas envisagé est celui où

$$\|f\|_G \leq \mu. \quad (2.79)$$

Alors $u = v = 0$ est évidemment la solution optimale de l'algorithme de J.F.Aujol. On a donc alors (Aujol)

$$f = 0 + 0 + f. \quad (2.80)$$

Que fournirait, dans les mêmes conditions, l'algorithme 2.78 ? Le test s'écrit alors

$$\|f\|_G \leq \frac{1}{2\lambda} \quad (2.81)$$

et deux cas doivent être envisagés. Si l'on a, en outre,

$$\|f\|_{BV} \leq \frac{1}{2\lambda\mu} \quad (2.82)$$

alors $u = w = 0$ et la décomposition optimale, au sens de 2.78, est $f = 0 + f + 0$. Elle est donc très différente de celle fournie par le nouvel algorithme de J.F.Aujol et al.

Remarquons ici qu'en vertu du théorème 2.2.5, si $\mu > \frac{1}{4\pi}$, minimiser $\|u\|_{BV} + \lambda\|v\|_{L^2}^2 + \mu^{-1}\|w\|_G$ conduit nécessairement à $u = 0$.

Supposons maintenant que l'on ait $\|f\|_{BV} > \frac{1}{2\lambda\mu}$, mais que l'on ait $\|f\|_G < \frac{\pi\mu}{\lambda}$. Alors on tombe sur le cas (1) du théorème 2.2.4 et l'on a nécessairement

$$u = 0, \quad \|v\|_{BV} = \frac{1}{2\lambda\mu}, \quad \|v\|_{L^2} \leq \sqrt{\frac{\|f\|_G}{\lambda\mu}} \quad (2.83)$$

$$\text{et } \langle v, w \rangle = \frac{\|w\|_G}{2\lambda\mu} \quad (2.84)$$

Illustrons ces remarques par l'exemple d'une route, de largeur ϵ et de longueur $L \gg 1$. Nous savons alors que $\|f\|_G \leq \frac{\epsilon}{2}$ (f est la fonction indicatrice de la route). Supposons $\frac{\epsilon}{2} \leq \mu$ et $\epsilon \leq \frac{1}{\lambda}$.

Alors l'algorithme de J.F.Aujol fournit $f = 0 + 0 + f$. En ce qui concerne 2.78, la décomposition est un peu différente et s'écrit $f = 0 + v + w$ où $\|v\|_{L^2} \leq \sqrt{\frac{\epsilon}{2\lambda\mu}}$ et donc $\|f - w\|_{L^2} \leq \sqrt{\frac{\epsilon}{2\lambda\mu}}$.

Voyons maintenant ce qui se passe lorsque $f(x) = \cos(Nx_1)\theta(x)$ où $\theta(x)$ est, par exemple, la fonction indicatrice du carré unité. Alors

$$\|f\|_G \leq \frac{C}{N}, \quad N \gg 1 \quad (2.85)$$

Dans le nouvel algorithme de J.F.Aujol, on obtient encore $f = 0 + 0 + f$. En ce qui concerne 2.78, la décomposition optimale est $f = 0 + v + w$ et l'on a encore (en comparant à la décomposition triviale $f = 0 + 0 + f$),

$$\lambda\|v\|_{L^2}^2 \leq \frac{1}{\mu}\|f\|_G \quad (2.86)$$

ce qui entraîne

$$\|v\|_{L^2} \leq \sqrt{\frac{C}{N\lambda\mu}}. \quad (2.87)$$

Dans ces deux situations, les deux algorithmes donnent asymptotiquement la même décomposition.

2.3 Autres espaces fonctionnels

2.3.1 Algorithme de Aujol et Chambolle

Comme nous l'avons mentionné à la section 2.2.2, Y.Meyer propose aussi d'utiliser l'espace de Besov $E = \dot{B}_{-1,\infty}^\infty$ qui est lui même plus grand que l'espace G (voir le lemme 2.2.2 sur l'inclusion des différents espaces). Donc les fonctions de G sont aussi dans E .

Du fait de la définition de la norme $\|\cdot\|_G$ incalculable directement en pratique, l'idée est alors d'utiliser la norme sur les espaces de Besov. En effet, nous avons vu au premier chapitre que les normes associées aux différents espaces de Besov sont définies grâce aux coefficients de la transformée en ondelette.

Le modèle proposé est le suivant

$$F_\lambda^{YM2}(u, v) = J(u) + \lambda\|v\|_E \quad (2.88)$$

Les premiers à s'être inspirés de ce modèle sont J.F.Aujol et A.Chambolle qui dans [1, 3] proposent d'utiliser une fonctionnelle similaire à (2.69) mais en remplaçant l'espace G par E .

$$F_{\lambda, \mu}^{AC}(u, v) = J(u) + B^* \left(\frac{v}{\mu} \right) + (2\lambda)^{-1} \|f - u - v\|_{L^2}^2 \quad (2.89)$$

Si l'on note E_μ le sous-espace des fonctions $f \in E$ telles que $\|f\|_E \leq \mu$ alors $B^*(f)$ est la fonction indicatrice sur l'espace E_1 , définie de la même manière que $J^*(\cdot)$.

$$B^*(v) = \begin{cases} 0 & \text{si } v \in E_1 \\ +\infty & \text{sinon} \end{cases} \quad (2.90)$$

Dans ce cas, la projection sur l'espace G_μ est remplacée par la projection sur l'espace E_μ . A.Chambolle et al. dans [21] montrent que cette projection s'exprime en fonction d'un seuillage doux des coefficients (*Wavelet Soft Thresholding* noté WST et défini par la définition 1.5.1) de la décomposition en ondelette par un seuil μ :

$$P_{E_\mu}(f) = f - WST(f, \mu) \quad (2.91)$$

l'algorithme de décomposition est donc

<p>1. Initialisation :</p> $u_0 = v_0 = 0$ <p>2. Itérations :</p> $v_{n+1} = P_{E_\mu}(f - u_n) = f - u_n - WST(f - u_n, \mu)$ $u_{n+1} = f - v_{n+1} - P_{G_\lambda}(f - v_{n+1})$ <p>3. On arrête l'algorithme si</p> $\max(u_{n+1} - u_n , v_{n+1} - v_n) \leq \epsilon$ <p>ou si l'on atteint un nombre maximal d'itérations prescrit.</p>

La figure 2.5 illustre la décomposition obtenue grâce à cet algorithme. Les paramètres utilisés sont $\lambda = 1$, $\kappa = 0.3$, la valeur de sigma a été fixée proche de la valeur de la variance de l'image initiale ($\sigma = 50$).

2.3.2 L'algorithme de A.Haddad

Dans ses travaux de thèse [43], A.Haddad propose l'utilisation d'un autre espace de Besov : $\dot{B}_{1, \infty}^1$. En effet, il est possible de montrer que les normes sur BV et $\dot{B}_{1, \infty}^1$ sont équivalentes sur l'ensemble de toutes les images dont le niveau de gris ne prend que les valeurs 0 et 1. Le théorème suivant est démontré dans [43] :



FIG. 2.5 – Algorithme de Aujol utilisant les espaces de Besov.

Théorème 2.3.1 *Si l'on considère l'ensemble \mathcal{E}_N des images dont le niveau de gris est assujéti à ne prendre que N valeurs (non spécifiées), il existe une constante C_N telle que*

$$\frac{1}{2}\|f\|_{\dot{B}_{1,\infty}^1} \leq \|f\|_{BV} \leq C_N\|f\|_{\dot{B}_{1,\infty}^1} \quad (2.92)$$

On ne connaît pas, à l'heure actuelle, la valeur optimale de C_N . On sait cependant que $C_N \leq CN$. Cela permet de remplacer la norme sur l'espace BV dans l'algorithme ROF par la norme sur l'espace $\dot{B}_{1,\infty}^1$.

$$F_\lambda^{HD}(u, v) = \|u\|_{\dot{B}_{1,\infty}^1} + \lambda\|v\|_2^2 \quad \text{où } f = u + v \quad (2.93)$$

L'espace BV n'est pas caractérisé par une condition portant sur les modules des coefficients d'ondelette. Si $f \in BV$, alors les coefficients d'ondelette $c_\lambda(f)$, $\lambda \in \Lambda$, appartiennent à l^1 -faible (une fois réarrangés par ordre décroissant, ces coefficients vérifient $c_m^* \leq \frac{C}{m}$, $m \geq 1$). Inversement, si $c_\lambda \in l^1(\Lambda)$, alors $\sum c_\lambda \psi_\lambda \in BV$. En quelque sorte BV est coincé entre l^1 et l_w^1 (observons que $\frac{1}{n \log^2 n}$, $n \geq 2$, appartient à l^1 tandis que $\frac{1}{n}$ appartient à l^1 -faible).

L'espace $\dot{B}_{-1,\infty}^1$, qui est un espace de distributions tempérées, peut être considéré comme le dual de $\dot{B}_{1,\infty}^1$. Par exemple, $f(x) = \sum_{j=1}^{\infty} j^{-2} 2^j \cos(2^j x_1)$, $x = (x_1, x_2)$, appartient à $\dot{B}_{-1,\infty}^1$. L'avantage de ces espaces est qu'ils sont caractérisés par une condition portant sur les modules de leurs coefficients d'ondelette. Par ailleurs, les normes équivalentes sur chacun de ces espaces se calculent de la manière suivante :

$$\|f\|_{\dot{B}_{1,\infty}^1} = \sup_j \sum_k |c_{j,k}| \quad (2.94)$$

$$\|f\|_{\dot{B}_{-1,\infty}^1} = \sum_j \sup_k |c_{j,k}| \quad (2.95)$$

où les $c_{j,k}$ sont les coefficients de la décomposition en ondelette de f . L'auteur montre quelques propriétés de cette fonctionnelle et de ses solutions. La résolution pratique du problème consistant à trouver (\hat{u}, \hat{v}) tels que

$$(\hat{u}, \hat{v}) = \inf_{(u,v) \in (\dot{B}_{1,\infty}^1 \times \dot{B}_{-1,\infty}^1)} F_\lambda^{HD}(u, v) \quad (2.96)$$

correspond à effectuer un seuillage sur les coefficients d'ondelette, en prenant un seuil adapté suivant les sous-bandes considérées. Le théorème suivant expose l'algorithme.

Théorème 2.3.2 *Soit $f \in L^2(\mathbb{R})$ telle que $f = u + v$ (u, v étant la décomposition optimale). On note $c_{j,k}, u_{j,k}, v_{j,k}$ les coefficients de la décomposition en ondelette de f, u, v respectivement. De plus nous supposons que $\|f\|_{\dot{B}_{-1,\infty}^1} \geq (2\lambda)^{-1}$. Alors il existe une unique suite $(v_j)_{j \in \mathbb{Z}}$ à termes positifs vérifiant $\sum_j v_j = (2\lambda)^{-1}$ et*

$$\begin{aligned} v_{j,k} &= \epsilon_{j,k} \min(v_j, |f_{j,k}|) \\ u_{j,k} &= \epsilon_{j,k} \max(|f_{j,k}| - v_j, 0) \\ \epsilon_{j,k} &= \text{sign}(f_{j,k}) \end{aligned}$$

Des détails pratiques pourront être trouvés au chapitre 7 de [43]. Je tiens à remercier Ali Haddad de m'avoir fourni, afin de pouvoir illustrer son algorithme, les images de la figure 2.6 ($\lambda = 25$).



FIG. 2.6 – Algorithme de A. Haddad ($\lambda = 25$).

Ces espaces de Besov permettent bien d'obtenir la décomposition recherchée. Comme pour l'algorithme ROF, nous pouvons constater qu'une part de géométrie est toujours présente dans la partie texture. Cette propriété a effectivement été démontrée par A. Haddad par l'intermédiaire du théorème suivant :

Théorème 2.3.3 Soit $f \in L^2(\mathbb{R})$ et $u_0 + v_0$ la décomposition optimale au problème 2.96. Si $\|f\|_{\dot{B}_{-1,\infty}^1} \leq (2\lambda)^{-1}$ alors $u_0 = 0$. Dans le cas contraire, $f = u + v$ est la solution optimale au problème 2.96 si et seulement si $\|v\|_{\dot{B}_{-1,\infty}^1} = (2\lambda)^{-1}$ et $\int uv dx = \|u\|_{\dot{B}_{1,\infty}^1} \|v\|_{\dot{B}_{-1,\infty}^1}$.

Ce théorème montre que la composante v ne peut pas être nulle (sauf dans le cas trivial où $f = 0$). Donc si l'on considère une image uniquement composée d'objets, cela signifie qu'une partie de ces objets va apparaître dans la composante texture. Cet algorithme possède le même défaut que l'algorithme ROF.

2.4 Evaluation de la décomposition

Nous venons de voir qu'il était possible d'utiliser soit l'espace G , soit les espaces de Besov pour séparer les textures du reste de l'image. Il est donc légitime de se demander lequel de ces espaces est le mieux adapté. Afin de pouvoir évaluer les performances de ces espaces, nous proposons de tester les algorithmes $F_{\lambda,\mu}^{AU}(u, v)$ et $F_{\lambda,\mu}^{AC}(u, v)$ sur une image synthétisée par nos soins. Cette image est constituée de la somme d'une image d'objets et d'une image de textures (fig 2.7).

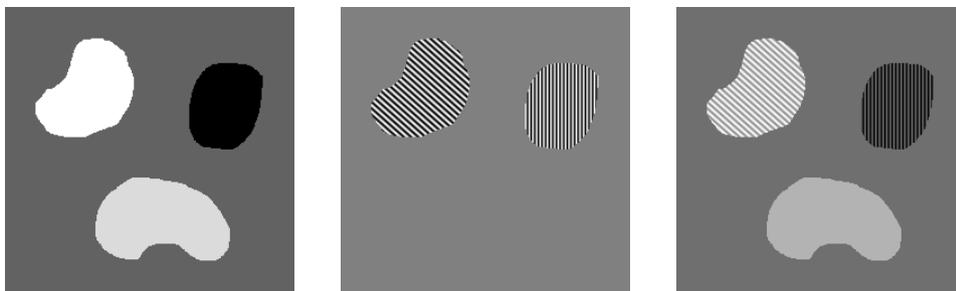


FIG. 2.7 – A droite l'image de test utilisée pour évaluer les algorithmes de décomposition, composée d'une image d'objets (à gauche) et de textures (au centre).

Nous appliquons alors les deux algorithmes de décomposition. Nous retenons les valeurs des paramètres donnant les meilleurs résultats visuels. Les paramètres retenus sont pour l'algorithme $F_{\lambda,\mu}^{AU}(u, v)$, $\mu = 500$, $\lambda = 1$ et pour l'algorithme $F_{\lambda,\mu}^{AC}(u, v)$, $\mu = 40$, $\lambda = 20$. Les images issues des algorithmes sont données sur la figure 2.8.

Nous calculons alors la norme de l'erreur entre chacune des composantes et leur référence respective. Nous obtenons les résultats du tableau 2.1.

Nous voyons clairement que l'espace G est le mieux adapté pour modéliser

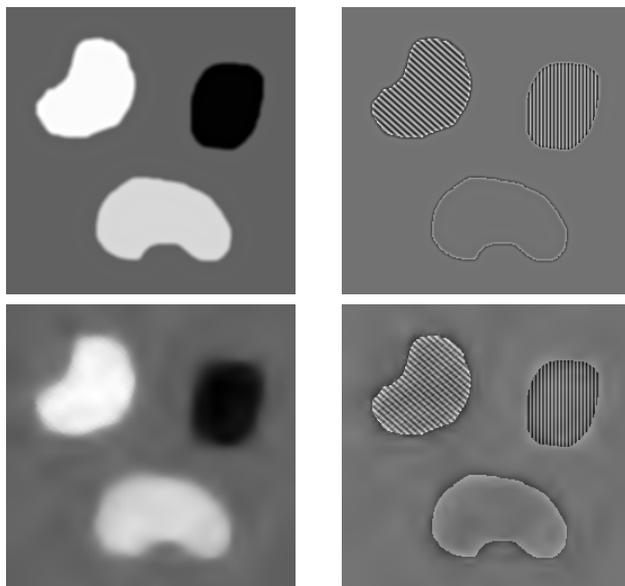


FIG. 2.8 – Composantes obtenues à partir des algorithmes $F_{\lambda,\mu}^{AU}(u,v)$ (première ligne) et $F_{\lambda,\mu}^{AC}(u,v)$ (deuxième ligne).

	$\ \tilde{u} - u_{ref}\ _{L^2}$	$\ \tilde{v} - v_{ref}\ _{L^2}$
Algorithme $F_{\lambda,\mu}^{AU}(u,v)$	668.4	624.4
Algorithme $F_{\lambda,\mu}^{AC}(u,v)$	1309.1	1245.1

TAB. 2.1 – Résultats de l'évaluation de la décomposition.

les textures. En effet les ondelettes ont tendance à abimer à la fois les bords des objets et les textures.

2.5 Bilan

Nous avons vu dans ce chapitre la définition d'un espace adapté à la modélisation des textures. Un algorithme permettant de séparer les textures des objets de l'image est aussi présenté. L'utilisation des espaces de Besov est aussi proposée dans la littérature. Nous avons mené une évaluation des algorithmes utilisant ces espaces. Il en ressort que l'espace initialement proposé est le mieux adapté pour la modélisation de la texture.

A ce stade, nous disposons donc d'un côté d'algorithmes permettant de traiter le bruit et d'un autre côté les textures. Le chapitre suivant s'intéresse donc à la fusion de ces deux aspects afin d'obtenir une modélisation plus générale.

Chapitre 3

Décomposition u, v, w

Jusqu'à présent, nous avons considéré des images non bruitées. Or dans un processus réel d'acquisition, les images sont très souvent bruitées. Il est donc légitime de se poser la question de savoir ce qui se passe pour les algorithmes du chapitre précédent avec ce type d'image. Nous avons vu que plus la composante v est oscillante, plus sa norme dans l'espace G est faible ce qui favorise l'apparition de ce type de fonctions dans v . Or le bruit peut être vu comme un signal aléatoire très oscillant par rapport au contenu textuel de l'image.

En conclusion, non seulement le bruit sera présent dans la composante v mais il a toutes les chances d'être prépondérant vis à vis des textures extraites de l'image (voir la figure 3.1 qui illustre le résultats de la décomposition u, v sur une image bruitée).



FIG. 3.1 – Composantes obtenues sur une image bruitée par un modèle u, v .

Il est naturel de se demander s'il est possible de séparer le bruit et les textures afin d'obtenir une décomposition plus «évoluée». Dans ce chapitre, nous

présentons un nouveau modèle permettant d'effectuer une décomposition en trois composantes : une composante structurelle $u \in BV$, une composante de textures et une composante de bruit. Pour cela, nous commençons par nous inspirer de l'idée de G.Gilboa et al. [39] où les auteurs utilisent le modèle ROF avec un coefficient de régularisation adaptatif en vue de faire du débruitage d'image préservant les textures.

L'idée est la suivante : soit λ_R ce coefficient de régularisation adaptatif (*ie* $\lambda_R = \lambda_R(f)(x, y)$ où f est l'image à débruiter) :

- si l'on est dans une région non texturée alors on peut augmenter la régularisation en prenant λ_R élevé.
- *a contrario* si l'on est dans une région texturée alors la régularisation doit être plus faible sous peine de rejeter les textures du résultat.

Pour réaliser la décomposition u, v, w nous allons reprendre cette idée de coefficient de régularisation non constant mais qui sera représentatif du contenu local de l'image.

Nous étudierons ensuite l'utilisation des méthodes par seuillage de coefficients, tout d'abord en examinant le travail de J.F.Aujol et A.Chambolle [3]. Les auteurs proposent eux aussi un nouveau modèle de décomposition en trois composantes en utilisant l'opérateur de seuillage doux WST . Nous donnerons une description de leur modèle puis nous comparerons les deux modèles (nous verrons qu'il existe un lien entre eux).

Enfin, nous remplacerons l'utilisation des ondelettes par les *contourlets* vues au chapitre 1 afin d'en examiner l'influence sur les performances de la décomposition.

3.1 Décomposition u, v, w adaptative

En nous basant sur les résultats proposés par J.F.Aujol sur la décomposition en deux composantes et l'idée de G.Gilboa et al. [39] (idée de coefficient de régularisation adaptatif), nous proposons le modèle suivant (nous rappelons que $G_\mu = \{v \in G / \|v\|_G \leq \mu\}$) :

$$F_{\lambda, \mu_1, \mu_2}^{JG}(u, v, w) = J(u) + J^* \left(\frac{v}{\mu_1} \right) + J^* \left(\frac{w}{\mu_2} \right) + (2\lambda)^{-1} \|f - u - \nu_1 v - \nu_2 w\|_{L^2}^2 \quad (3.1)$$

où $\nu_1 = \nu_1(f)(x, y), \nu_2 = \nu_2(f)(x, y)$ sont deux fonctions de $\mathbb{R}^2 \rightarrow]0; 1[$ généralisant l'idée de coefficient de régularisation adaptatif suivant la zone de l'image dans laquelle on se situe. Les fonctions v et w pouvant être vues toutes les deux comme oscillantes, nous les prendrons respectivement dans G_{μ_1} et G_{μ_2} . La fonction u , quant à elle, sera prise dans l'espace BV .

Dans [3], J.F.Aujol et A.Chambolle présentent un travail sur le calcul des différentes normes. On peut voir que la norme sur l'espace G est plus faible dans le cas d'un bruit que dans le cas d'une texture (plus une fonction est oscillante, plus sa norme $\|\cdot\|_G$ est faible). Il nous faut donc choisir $\mu_2 < \mu_1$ afin que la composante v représente bien les textures et w la composante bruit.

La solution du problème précédent est donné par la proposition suivante :

Proposition 3.1.1 *Soient $u \in BV$, $v \in G_{\mu_1}$, $w \in G_{\mu_2}$ représentant respectivement les composantes géométrique, texture et bruit issues de la décomposition de l'image et $(\nu_1(f)(x, y), \nu_2(f)(x, y))$ deux fonctions de $\mathbb{R}^2 \rightarrow]0; 1[$ fixées et supposées être beaucoup plus régulières que les fonctions v et w . Alors la solution de*

$$(\hat{u}, \hat{v}, \hat{w}) = \arg \inf_{(u,v,w) \in BV \times G_{\mu_1} \times G_{\mu_2}} F_{\lambda, \mu_1, \mu_2}^{JG}(u, v, w) \quad (3.2)$$

est donnée par

$$\hat{u} = f - \nu_1 \hat{v} - \nu_2 \hat{w} - P_{G_\lambda}(f - \nu_1 \hat{v} - \nu_2 \hat{w}) \quad (3.3)$$

$$\hat{v} = P_{G_{\mu_1}} \left(\frac{f - \hat{u} - \nu_2 \hat{w}}{\nu_1} \right) \quad (3.4)$$

$$\hat{w} = P_{G_{\mu_2}} \left(\frac{f - \hat{u} - \nu_1 \hat{v}}{\nu_2} \right) \quad (3.5)$$

où les P_{G_μ} sont les projecteurs non-linéaires introduits par A. Chambolle (voir [20]).

Preuve:

Commençons par chercher la solution par rapport à u (le calcul est similaire à celui effectué dans [20], nous allons donc en donner une version rapide ici) : on applique le principe d'Euler-Lagrange à (3.2) par rapport à u :

$$-\frac{1}{\lambda}(f - u - \nu_1 v - \nu_2 w) + \partial J(u) \ni 0 \quad (3.6)$$

$$\Leftrightarrow u \in \partial J^* \left(\frac{1}{\lambda}(f - u - \nu_1 v - \nu_2 w) \right) \quad (3.7)$$

on rajoute $f - u - \nu_1 v - \nu_2 w$ de chaque coté puis on multiplie par $\frac{1}{\lambda}$

$$f - u - \nu_1 v - \nu_2 w + u \in \begin{aligned} & f - u - \nu_1 v - \nu_2 w + \\ & \partial J^* \left(\frac{1}{\lambda}(f - u - \nu_1 v - \nu_2 w) \right) \end{aligned} \quad (3.8)$$

$$\frac{1}{\lambda}(f - \nu_1 v - \nu_2 w) \in \begin{aligned} & \frac{1}{\lambda}(f - u - \nu_1 v - \nu_2 w) + \\ & \frac{1}{\lambda} \partial J^* \left(\frac{1}{\lambda}(f - u - \nu_1 v - \nu_2 w) \right) \end{aligned} \quad (3.9)$$

posons $\eta = (f - u - \nu_1 v - \nu_2 w)/\lambda$ on a alors

$$\frac{1}{\lambda}(f - \nu_1 v - \nu_2 w) \in \eta + \frac{1}{\lambda} \partial J^*(\eta). \quad (3.10)$$

La solution est alors donnée par

$$\eta = P_G \left(\frac{1}{\lambda}(f - \nu_1 v - \nu_2 w) \right) = \frac{1}{\lambda} P_{G_\lambda}(f - \nu_1 v - \nu_2 w). \quad (3.11)$$

En réinjectant η et en isolant u on obtient le résultat :

$$\hat{u} = f - \nu_1 v - \nu_2 w - P_{G_\lambda}(f - \nu_1 v - \nu_2 w). \quad (3.12)$$

Démontrons le lemme suivant afin de démontrer les relations donnant \hat{v} et \hat{w} :

Lemme 3.1.2 *Soient $f \in L^2(\mathbb{R}^2)$, $v \in G_\mu$ et $\nu(x, y)$ une fonction de $\mathbb{R}^2 \rightarrow]0; 1[$ fixée et suffisamment régulière par rapport à v , alors*

$$\hat{v} = \arg \inf_{v \in G_\mu} \left\{ (2\lambda)^{-1} \|f - \nu v\|_{L^2}^2 + J^* \left(\frac{v}{\mu} \right) \right\} \quad (3.13)$$

est donné par

$$\hat{v} = P_{G_\mu} \left(\frac{f}{\nu} \right) \quad (3.14)$$

Preuve:

Rappelons que J^* est la fonction indicatrice sur l'espace G_1 ; en posant $\eta = \frac{v}{\mu}$ et en prenant l'équation d'Euler-Lagrange par rapport à η de (3.13), nous obtenons

$$-\frac{\mu\nu}{\lambda}(f - \mu\nu\eta) + \partial J^*(\eta) \ni 0. \quad (3.15)$$

Donc

$$\mu^2 \nu^2 \eta - \mu\nu f + \lambda \partial J^*(\eta) \ni 0 \quad (3.16)$$

$$\Leftrightarrow \eta - \frac{f}{\mu\nu} + \frac{\lambda}{\mu^2 \nu^2} \partial J^*(\eta) \ni 0. \quad (3.17)$$

Or l'on suppose que ν est suffisamment régulière par rapport à v (on entend par là que par rapport à l'échelle des variations rapides de v , ν se comporte comme une constante). On peut donc considérer que le terme $\frac{\lambda}{\mu^2 \nu^2}$ se comporte comme une constante face à $\partial J^*(\eta)$.

Donc

$$\hat{\eta} = P_G \left(\frac{f}{\mu\nu} \right) = \frac{1}{\mu} P_{G_\mu} \left(\frac{f}{\nu} \right). \quad (3.18)$$

Finalement on obtient

$$\hat{v} = P_{G_\mu} \left(\frac{f}{\nu} \right) \quad (3.19)$$

■

En appliquant ce lemme tout d'abord par rapport à v , puis par rapport à w à $F_{\lambda, \mu_1, \mu_2}^{JG}(u, v, w)$ on obtient facilement les expressions de \hat{v} et \hat{w} énoncées dans la proposition 3.1.1.

■

3.1.1 Algorithme numérique.

Suivant le contenu de l'image, le comportement de l'algorithme doit être le suivant :

- Dans une partie de l'image contenant de la texture et du bruit : ν_1 doit être faible afin de renforcer v et *a contrario* ν_2 doit être proche de 1 afin de modéliser l'absence de bruit,
- Dans une partie de l'image ne contenant que du bruit : ν_1 doit être proche de 1 afin de modéliser l'absence de texture et ν_2 doit être proche de 0 pour prendre en compte le bruit.

Si nous partons du fait que les rôles des fonctions ν_1 et ν_2 sont complémentaires, une hypothèse évidente consiste à prendre $\nu_2 = 1 - \nu_1$.

Concernant le choix des fonctions ν_i , nous nous plaçons dans le cadre d'un bruit additif et nous pouvons donc considérer la composante texture orthogonale au bruit. La variance d'une zone comportant à la fois de la texture et du bruit sera plus importante que la variance d'une zone ne possédant pas de texture. Nous pouvons donc nous servir d'un calcul de variance «local» pour construire une carte des zones texturées ou non. En pratique, la valeur de ν prise aux coordonnées (i, j) , correspondra à la variance calculée sur une fenêtre carrée de taille $L \times L$, centrée en (i, j) dans l'image f . La taille de cette fenêtre ne devant être ni trop petite, ni trop grande afin d'obtenir un bon compromis entre estimation correcte de la variance et localité dans l'image. Un exemple de fonction ν est illustré sur la figure 3.2. Nous requantifions cette fonction de pondération ν de telle sorte que $\nu \in]0; 1[$ afin de respecter les hypothèses de la proposition 3.1.1.

L'algorithme numérique associé est le suivant

Étape 1 : Initialisation : $u_0 = v_0 = w_0 = 0$

Étape 2 : Calcul de ν_1 et ν_2 à partir de f



FIG. 3.2 – Exemple de fonction de pondération ν (à gauche l'image initiale bruitée, à droite la fonction de pondération correspondante)

Etape 3 : Calcul de $w_{n+1} = P_{G_{\mu_2}} \left(\frac{f - u_n - \nu_1 v_n}{\nu_2} \right)$

Etape 4 : Calcul de $v_{n+1} = P_{G_{\mu_1}} \left(\frac{f - u_n - \nu_2 w_{n+1}}{\nu_1} \right)$

Etape 5 : Calcul de $u_{n+1} = f - \nu_1 v_{n+1} - \nu_2 w_{n+1} - P_{G_\lambda}(f - \nu_1 v_{n+1} - \nu_2 w_{n+1})$

Etape 6 : On stoppera la décomposition, soit par un critère d'arrêt sur l'erreur commise ($\max\{|u_{n+1} - u_n|, |v_{n+1} - v_n|, |w_{n+1} - w_n|\} \leq \epsilon$) ou par un critère sur un nombre prescrit d'itérations. Si le critère d'arrêt n'est pas satisfait on retourne à l'étape 3.

Nous présentons sur la figure 3.3 les résultats obtenus avec cet algorithme. La première ligne donne l'image initiale ainsi que sa version bruitée (bruit gaussien, $\sigma = 20$) utilisées. La deuxième ligne présente respectivement les composantes u et v , w étant visible sur la troisième ligne. Les paramètres utilisés sont : $\lambda = 10$, $\mu_1 = 1000$, $\mu_2 = 1$, deux itérations pour la convergence de la décomposition et une taille de quinze pixels pour la fenêtre d'analyse lors de la construction des ν_i .

On peut voir que les résultats obtenus sont conformes à ce que l'on pouvait attendre de cette modélisation. Le fait d'imposer des valeurs de normes très différentes pour les fonctions v et w permet bien de séparer le bruit des textures. On peut toutefois remarquer que du bruit persiste dans la composante v et que des résidus de texture sont aussi présents dans la composante bruit. Afin de pouvoir apprécier l'intérêt de la notion de localité apportée par les fonctions ν_i , nous présentons sur la figure 3.4 les résultats du même algorithme dans lequel les fonctions ν_i ont été supprimées du modèle (les autres paramètres restent inchangés) :

$$\tilde{F}_{\lambda, \mu_1, \mu_2}^{JG}(u, v, w) = J(u) + J^* \left(\frac{v}{\mu_1} \right) + J^* \left(\frac{w}{\mu_2} \right) + (2\lambda)^{-1} \|f - u - v - w\|_{L^2}^2 \quad (3.20)$$

On peut constater que sans la notion de localité, une quantité importante de bruit reste présente dans la composante textures. Le fait d'utiliser ces fonctions de pondération ν_i permet donc d'améliorer très nettement la séparation du bruit et des textures.

3.2 Algorithme de Aujol-Chambolle

Dans [3], J.F.Aujol et A.Chambolle proposent le modèle suivant pour effectuer la décomposition en trois composantes.

$$F_{\lambda, \mu, \delta}^{AC2}(u, v, w) = J(u) + J^* \left(\frac{v}{\mu} \right) + B^* \left(\frac{w}{\delta} \right) + (2\lambda)^{-1} \|f - u - v - w\|_{L^2}^2 \quad (3.21)$$

où $u \in BV, v \in G_\mu, w \in E_\delta$ avec

$$E_\delta = \left\{ w \in \dot{B}_{-1, \infty}^\infty / \|w\|_{\dot{B}_{-1, \infty}^\infty} \leq \delta \right\} \quad (3.22)$$

et $B^*(w)$ est défini par 2.90

Les auteurs montrent alors la proposition suivante

Proposition 3.2.1 *La solution au problème*

$$(\hat{u}, \hat{v}, \hat{w}) = \arg \inf_{u \in BV, v \in G_\mu, w \in E_\delta} F_{\lambda, \mu, \delta}^{AC2}(u, v, w) \quad (3.23)$$

est donnée par

$$\hat{u} = f - \hat{v} - \hat{w} - P_{G_\lambda}(f - \hat{v} - \hat{w}) \quad (3.24)$$

$$\hat{v} = P_{G_\mu}(f - \hat{u} - \hat{w}) \quad (3.25)$$

$$\hat{w} = P_{E_\delta}(f - \hat{u} - \hat{v}) = f - \hat{u} - \hat{v} - WST(f - \hat{u} - \hat{v}, 2\delta) \quad (3.26)$$

où $WST(f - \hat{u} - \hat{v}, 2\delta)$ est l'opérateur de seuillage doux des coefficients d'ondelette (*Wavelet Soft Thresholding*).

J.F.Aujol et al. donnent une preuve de cette proposition dans [3].

L'algorithme numérique associé est décrit ci-après

Étape 1 : Initialisation : $u_0 = v_0 = w_0 = 0$

Étape 2 : Calcul de $w_{n+1} = f - u_n - v_n - WST(f - u_n - v_n, 2\delta)$

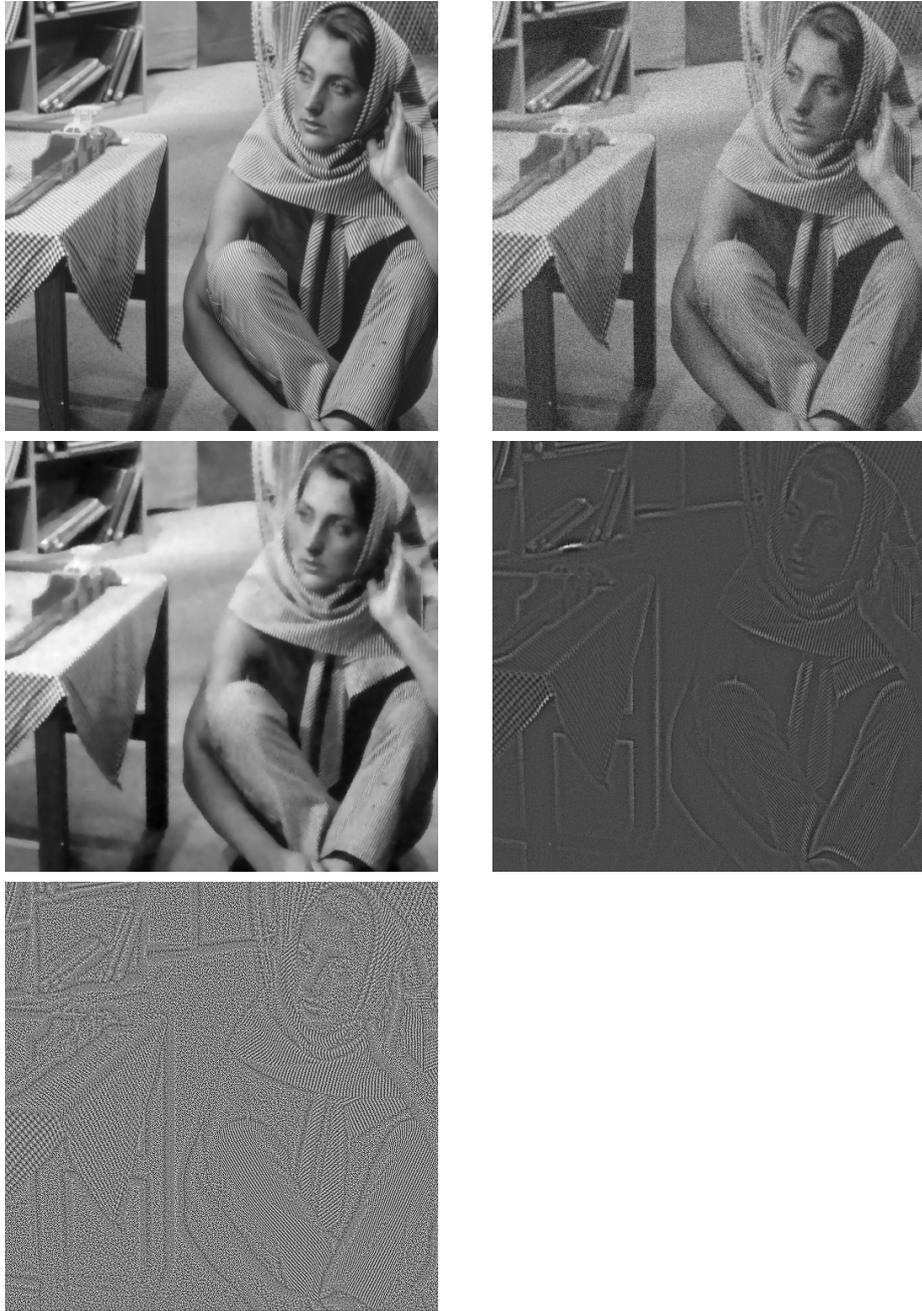


FIG. 3.3 – Algorithme JG : résultats obtenus à partir de la fonctionnelle $F_{\lambda, \mu_1, \mu_2}^{JG}$

Étape 3 : Calcul de $v_{n+1} = P_{G_\mu}(f - u_n - w_{n+1})$

Étape 4 : Calcul de $u_{n+1} = f - v_{n+1} - w_{n+1} - P_{G_\lambda}(f - v_{n+1} - w_{n+1})$

Étape 5 : On stoppera la décomposition, soit par un critère d'arrêt sur l'er-



FIG. 3.4 – Algorithme JG : résultats obtenus à partir de la fonctionnelle $\tilde{F}_{\lambda, \mu_1, \mu_2}^{JG}$ (sans les fonctions ν_i)

reur commise ($\max\{|u_{n+1} - u_n|, |v_{n+1} - v_n|, |w_{n+1} - w_n|\} \leq \epsilon$) ou par un critère sur un nombre prescrit d'itérations. Si le critère d'arrêt n'est pas satisfait on retourne à l'étape 2.

La figure 3.5 illustre la décomposition obtenue par cet algorithme sur l'image de test de la figure 3.3 (la première ligne rappelant l'image initiale et sa version bruitée). La deuxième ligne donne les composantes u et v , la troisième quant à elle illustre la composante bruit. Les paramètres utilisés sont $\mu = 100$, $\lambda = 1$, $\kappa = 0.3$, $\sigma = 20$.

On peut voir que la partie texture est mieux débruitée par cet algorithme que dans le cas de notre modélisation, toutefois un plus gros résidu de texture est aussi perdu dans la composante bruit.

3.3 Comparaison entre les deux modèles.

Nous pouvons constater que les résultats obtenus en sortie des deux algorithmes sont visuellement très proches. Nous nous sommes donc demandé si un lien pouvait exister entre les deux modèles. Si l'on examine de plus près les résultats obtenus par les deux algorithmes d'une part sur l'images globale et d'autre part sur des zooms (voir la figure 3.6) nous pouvons constater que le bruit semble mieux modélisé dans l'algorithme de Aujol-Chambolle (AC2) (par l'utilisation des espaces de Besov). Toutefois l'algorithme AC2 rejette une plus grande part d'information sur la texture dans la composante bruit (ce qui représente une information perdue si l'on se place dans le cadre de la restauration dont le but est de recomposer $u + v$ dans le cadre du modèle AC2 ou $u + \nu_1 v$ dans notre cas, voir la figure 3.6).

Le bruit est mieux modélisé dans le cas de l'algorithme AC2 car l'espace de Besov correspond plus à la vraie nature du bruit (du type distributions) à l'instar de notre modélisation qui utilise l'espace G . A l'inverse, la notion de décomposition adaptative locale dans l'image que nous avons introduit nous permet de réduire la quantité de résidu de texture extraite dans la composante bruit (w). Une nouvelle question apparaît naturellement : peut-on «unifier» ces deux modèles afin de conserver cette notion de décomposition adaptative locale avec l'utilisation de l'espace $\dot{B}_{-1,\infty}^\infty$?

Nous examinons cette possibilité dans la section suivante.

3.4 «Unification» des deux modèles.

Nous avons vu précédemment que chacun des modèles JG et AC2 possédaient leur point fort :

- la notion de localité de la décomposition dans notre modèle (modèle JG),
- l'utilisation de l'espace $\dot{B}_{-1,\infty}^\infty$ mieux adapté à la modélisation du bruit dans le modèle de J.F.Aujol et A.Chambolle (AC2).

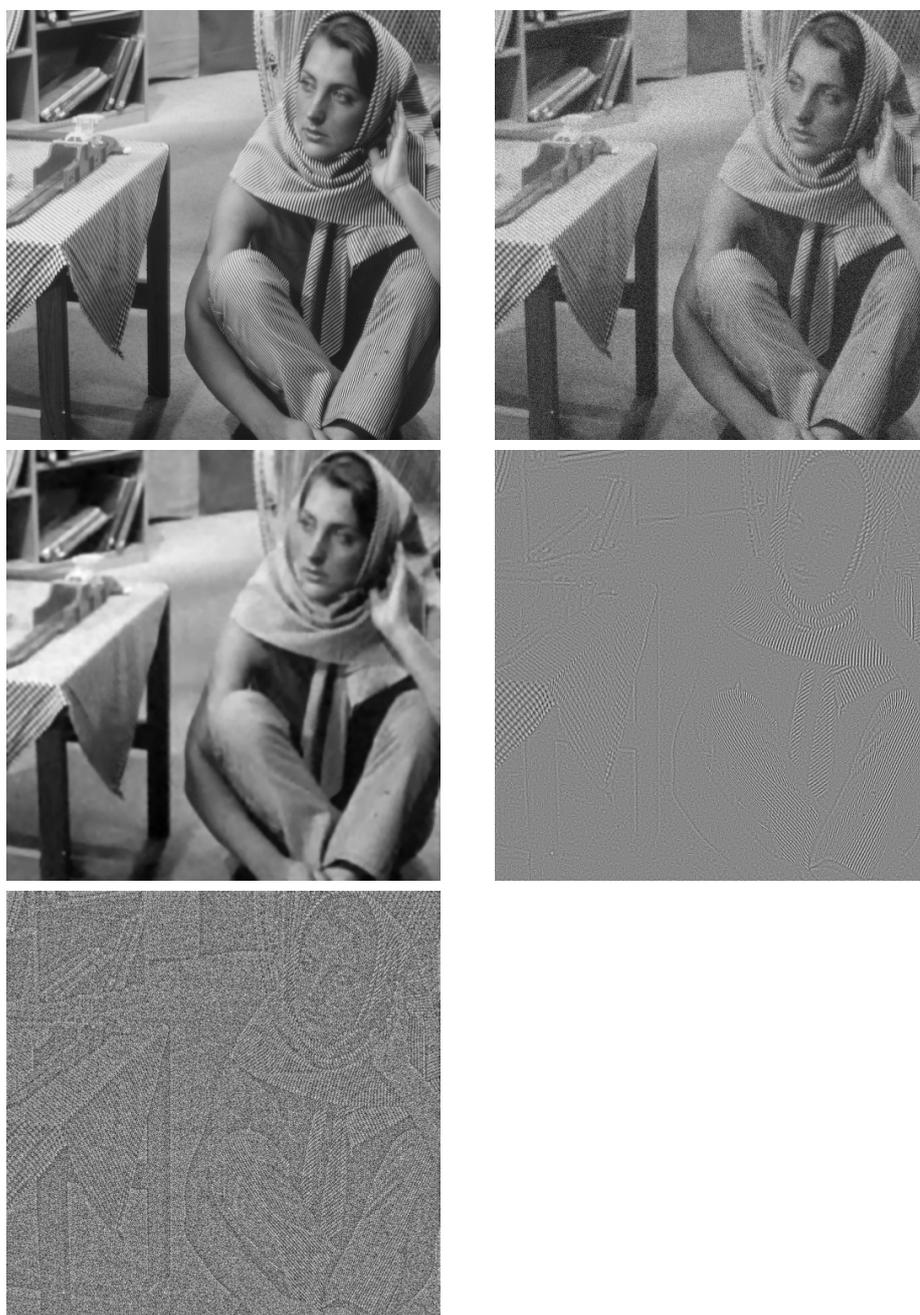


FIG. 3.5 – Algorithme AC2 : résultats de décomposition obtenus par la fonctionnelle $F_{\lambda,\mu,\delta}^{AC2}$

Il semble donc intéressant d'examiner la possibilité de réunir ces deux points forts dans une même modélisation. Nous proposons donc d'utiliser le modèle

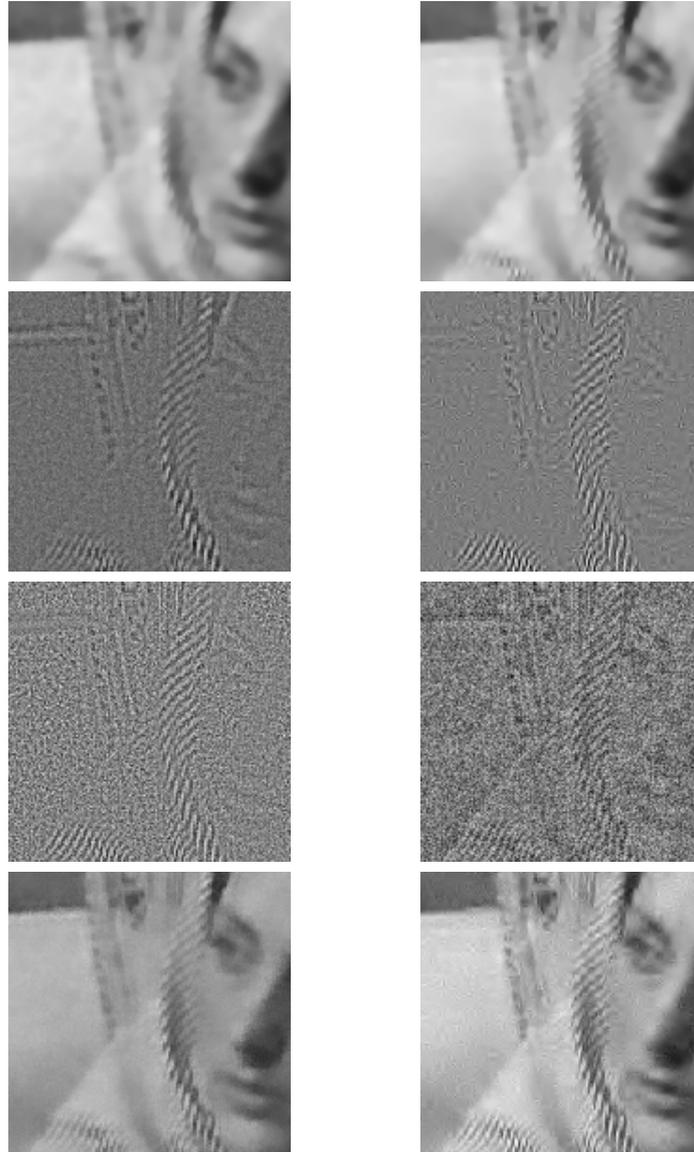


FIG. 3.6 – Comparaison entre l’algorithme de Aujol et al. et notre algorithme sur des zooms. La colonne de droite correspond à l’algorithme AC, la colonne de gauche à l’algorithme JG (les images étant respectivement de haut en bas : u , v , w , partie restaurée).

décris par 3.27.

$$F_{\lambda, \mu, \delta}^{JG2}(u, v, w) = J(u) + J^* \left(\frac{v}{\mu} \right) + B^* \left(\frac{w}{\delta} \right) + (2\lambda)^{-1} \|f - u - \nu_1 v - \nu_2 w\|_{L^2}^2 \quad (3.27)$$

Nous cherchons alors la solution du problème suivant

$$(\hat{u}, \hat{v}, \hat{w}) = \arg \inf_{(u,v,w) \in BV \times G_\mu \times B_{E_\delta}} F_{\lambda,\mu,\delta}^{JG^2}(u, v, w) \quad (3.28)$$

Nous retrouvons dans ce modèle généralisé nos fonctions de régularisation adaptatives ν_1 et ν_2 pondérant les composantes v et w et nous imposons bien que $w \in B_{E_\delta}$. La solution à ce modèle est donnée par la proposition suivante.

Proposition 3.4.1 *Soient $u \in BV$, $v \in G_\mu$, $w \in B_{E_\delta}$ représentant respectivement les composantes géométrique, texture et bruit issues de la décomposition de l'image et $(\nu_1(f)(x, y), \nu_2(f)(x, y))$ deux fonctions de $\mathbb{R}^2 \rightarrow]0; 1[$ fixées et supposées être beaucoup plus régulières que les fonctions v et w . Alors la solution de*

$$(\hat{u}, \hat{v}, \hat{w}) = \arg \inf_{(u,v,w) \in BV \times G_\mu \times B_{E_\delta}} F_{\lambda,\mu,\delta}^{JG^2}(u, v, w) \quad (3.29)$$

est donnée par

$$\begin{aligned} \hat{u} &= f - \nu_1 \hat{v} - \nu_2 \hat{w} - P_{G_\lambda}(f - \nu_1 \hat{v} - \nu_2 \hat{w}) \\ \hat{v} &= P_{G_\mu} \left(\frac{f - \hat{u} - \nu_2 \hat{w}}{\nu_1} \right) \\ \hat{w} &= \frac{f - \hat{u} - \nu_1 \hat{v}}{\nu_2} - \frac{\lambda}{\delta \nu_2^2} WST \left(\frac{\delta \nu_2}{\lambda} (f - \hat{u} - \nu_1 \hat{v}); \frac{2\delta^2 \tilde{\nu}_2^2}{\lambda} \right) \end{aligned}$$

où P_{G_λ} est le projecteur non-linéaire introduit par A.Chambolle (voir [20]) et $WST(f, \delta)$ est l'opérateur de seuillage doux des coefficients de la décomposition en ondelette de f et $\tilde{\nu}_2$ est une version pyramidale de ν_2 (voir la fig 3.7).

Preuve:

La démonstration concernant la minimisation par rapport à u et à v se fait exactement de la même manière que dans la démonstration de la proposition 3.1.1 en utilisant le lemme 3.1.2 pour la composante v . Il nous reste à démontrer le résultat dans le cas de la composante w .

Supposons u et v fixés, il nous faut donc trouver

$$\hat{w} = \arg \inf_{w \in B_{E_\delta}} \left\{ (2\lambda)^{-1} \|f - u - \nu_1 v - \nu_2 w\|_{L^2}^2 + B^* \left(\frac{w}{\delta} \right) \right\} \quad (3.30)$$

Pour cela commençons par montrer le lemme suivant

Lemme 3.4.2 Soient $f \in L^2$, $w \in B_{E_\delta}$ et une fonction $\nu(x, y)$ de $\mathbb{R}^2 \rightarrow]0; 1[$ fixée et supposée être très régulière par rapport à w . La solution \hat{w} de

$$\hat{w} = \arg \inf_{w \in B_{\delta E}} \left\{ (2\lambda)^{-1} \|f - \nu w\|_{L^2}^2 + B^* \left(\frac{w}{\delta} \right) \right\} \quad (3.31)$$

est donnée par

$$\hat{w} = \frac{f}{\nu} - \frac{\lambda}{\delta \nu^2} WST \left(\frac{\delta f \nu}{\lambda}; \frac{2\delta^2 \tilde{\nu}^2}{\lambda} \right) \quad (3.32)$$

Preuve:

On différencie par rapport à w la fonctionnelle définie en (3.31) en utilisant le principe d'Euler-Lagrange

$$0 \in -\frac{\nu}{\lambda}(f - \nu w) + \frac{1}{\delta} \partial B^* \left(\frac{w}{\delta} \right) \quad (3.33)$$

$$\Leftrightarrow \frac{w}{\delta} \in \partial B \left(\frac{\delta \nu (f - \nu w)}{\lambda} \right). \quad (3.34)$$

Posons $\eta = \frac{\delta \nu (f - \nu w)}{\lambda}$. Alors

$$0 \in \frac{\lambda}{\delta^2 \nu^2} \eta - \frac{f}{\delta \nu} + \partial B(\eta) \quad (3.35)$$

$$\Leftrightarrow 0 \in \eta - \frac{\delta f \nu}{\lambda} + \frac{\delta^2 \nu^2}{\lambda} \partial B(\eta). \quad (3.36)$$

Or ceci découle de la formulation par Euler-Lagrange de

$$\arg \inf_{\eta \in L^2} \left\{ \frac{1}{2} \left\| \eta - \frac{\delta f \nu}{\lambda} \right\|_{L^2}^2 + \frac{\delta^2 \nu^2}{\lambda} \|\eta\|_{B_{1,1}^1} \right\}. \quad (3.37)$$

D'après [21] et du fait que ν peut être considérée comme constante par rapport à η , la solution est donnée par

$$\hat{\eta} = WST \left(\frac{\delta f \nu}{\lambda}; \frac{2\delta^2 \tilde{\nu}^2}{\lambda} \right) \quad (3.38)$$

où $\tilde{\nu}$ est une version pyramidale de ν (voir la figure 3.7) afin d'avoir un seuillage adaptatif propre à chacune des sous-bandes de la décomposition en ondelette.

En réinjectant l'expression de $\hat{\eta}$ on obtient

$$\frac{\delta f \nu}{\lambda} - \frac{\delta \nu^2 \hat{w}}{\lambda} = WST \left(\frac{\delta f \nu}{\lambda}; \frac{2\delta^2 \tilde{\nu}^2}{\lambda} \right) \quad (3.39)$$

FIG. 3.7 – Fonction de pondération ν et sa version pyramidale $\tilde{\nu}$

$$\Rightarrow \hat{w} = \frac{f}{\nu} - \frac{\lambda}{\delta\nu^2} WST \left(\frac{\delta f \nu}{\lambda}; \frac{2\delta^2 \tilde{\nu}^2}{\lambda} \right) \quad (3.40)$$

■

Grâce au lemme précédent, il suffit de remplacer f par $f - u - \nu_1 v$ et ν par ν_2 pour obtenir la fin de la démonstration de la proposition 3.4.1 :

$$\hat{w} = \frac{f - u - \nu_1 v}{\nu_2} - \frac{\lambda}{\delta\nu_2^2} WST \left(\frac{\delta\nu_2}{\lambda} (f - u - \nu_1 v); \frac{2\delta^2 \tilde{\nu}_2^2}{\lambda} \right) \quad (3.41)$$

■

L'algorithme numérique associé est décrit ci-après

Étape 1 : Initialisation : $u_0 = v_0 = w_0 = 0$

Étape 2 : Calcul de ν_1, ν_2 et $\tilde{\nu}_2$

Étape 3 : Calcul de $w_{n+1} = \frac{f - u - \nu_1 v}{\nu_2} - \frac{\lambda}{\delta\nu_2^2} WST \left(\frac{\delta\nu_2}{\lambda} (f - u - \nu_1 v), \frac{2\delta^2 \tilde{\nu}_2^2}{\lambda} \right)$

Étape 4 : Calcul de $v_{n+1} = P_{G_\mu} \left(\frac{f - u_n - \nu_2 w_{n+1}}{\nu_1} \right)$

Étape 5 : Calcul de $u_{n+1} = f - \nu_1 v_{n+1} - \nu_2 w_{n+1} - P_{G_\lambda} (f - \nu_1 v_{n+1} - \nu_2 w_{n+1})$

Étape 6 : On stoppera la décomposition, soit par un critère d'arrêt sur l'erreur commise ($\max\{|u_{n+1} - u_n|, |v_{n+1} - v_n|, |w_{n+1} - w_n|\} \leq \epsilon$) ou par un critère sur un nombre prescrit d'itérations. Si le critère d'arrêt n'est pas satisfait on retourne à l'étape 3.

La figure 3.8 donne les résultats obtenus à partir de cet algorithme unifié. Les paramètres utilisés sont $\mu = 1000$, $\lambda = 5$, $\sigma = 20$, $\kappa = 0.3$ (nous rappelons que κ est un paramètre de permettant d'ajuster le seuil optimal calculé par D. Donoho) et une taille de fenêtre d'analyse de 7×7 pixels.



FIG. 3.8 – Algorithme JG2 : en haut se trouvent l'image bruitée ainsi que la composante u ; en bas les composantes v et w .

On peut clairement constater que l'ajout de la notion de localité dans le seuillage des coefficients d'ondelette permet de réduire la quantité d'information texture présente dans la composante bruit. Par ailleurs, l'utilisation de l'espace $\dot{B}_{-1,\infty}^\infty$ permet d'obtenir des textures qui soient mieux débruitées.

3.5 Utilisation des *contourlets*

Nous venons de voir que l'espace de Besov $\dot{B}_{-1,\infty}^\infty$ était un bon choix pour modéliser le bruit. Par ailleurs, dans le chapitre 1 nous avons testé les *contourlets* comme une alternative aux ondelettes et mieux adaptées au cas de certaines images. Nous proposons dans cette section d'examiner l'influence de ce nouveau type de fonctions. Nous avons défini dans la section 1.4.4 les espaces de *contourlets* et avons proposé une norme (inspirée de celle sur les espaces de Besov) associée à ces espaces. D'autre part, nous avons vu que nous pouvions également définir les opérateurs de seuillage doux et dur sur

les coefficients *contourlets*.

Nous proposons donc de remplacer l'utilisation des ondelettes dans le modèle de Aujol-Chambolle et ainsi obtenir le modèle suivant,

$$F_{\lambda,\mu,\delta}^{JG^3}(u, v, w) = J(u) + J^* \left(\frac{v}{\mu} \right) + J_{CT}^* \left(\frac{w}{\delta} \right) + (2\lambda)^{-1} \|f - u - v - w\|_{L^2}^2 \quad (3.42)$$

où $J_{CT}^*(f)$ est la fonction indicatrice sur l'ensemble des fonctions CT_1 (où $CT_\delta = \left\{ f / \|f\|_{CT_{-1,\infty}} \leq \delta \right\}$, voir la section 1.4.4 pour la définition de cette norme).

$$J_{CT}^*(f) = \begin{cases} 0 & \text{si } f \in CT_1 \\ +\infty & \text{sinon} \end{cases} \quad (3.43)$$

Le triplet $(\hat{u}, \hat{v}, \hat{w})$ minimisant $F_{\lambda,\mu,\delta}^{JG^3}(u, v, w)$ est donné par la proposition suivante.

Proposition 3.5.1 *Soient $u \in BV$, $v \in G_\mu$, $w \in CT_\delta$ représentant respectivement les composantes géométrique, texture et bruit issues de la décomposition de l'image. Alors la solution de*

$$(\hat{u}, \hat{v}, \hat{w}) = \arg \inf_{(u,v,w) \in BV \times G_\mu \times CT_\delta} F_{\lambda,\mu,\delta}^{JG^3}(u, v, w) \quad (3.44)$$

est donnée par

$$\begin{aligned} \hat{u} &= f - \hat{v} - \hat{w} - P_{G_\lambda}(f - \hat{v} - \hat{w}) \\ \hat{v} &= P_{G_\mu}(f - \hat{u} - \hat{w}) \\ \hat{w} &= f - \hat{u} - \hat{v} - CST(f - \hat{u} - \hat{v}; \delta) \end{aligned}$$

où P_{G_λ} est le projecteur non-linéaire introduit par A.Chambolle (voir [20]) et $CST(f, \delta)$ est l'opérateur de seuillage doux des coefficients de la décomposition contourlet de $f - u - v$.

Preuve:

Les expressions de \hat{u}, \hat{v} sont obtenues exactement de la même manière que dans la démonstration de la proposition 3.1.1. Le point particulier concerne l'expression de \hat{w} en fonction du seuillage des coefficients *contourlets*. Si l'on cherche à minimiser $F_{\lambda,\mu,\delta}^{JG^3}(u, v, w)$ par rapport à w , le problème revient à trouver w solution de (on pose $g = f - u - v$)

$$\hat{w} = \arg \min_{w \in CT_\delta} \{ \|g - w\|_{L^2}^2 \} \quad (3.45)$$

nous allons remplacer ce problème par son problème dual : $\hat{w} = g - \hat{h}$ tel que

$$\hat{h} = \arg \min_{h \in CT_{1,1}^1} \left\{ 2\delta \|h\|_{CT_{1,1}^1} + \|g - h\|_{L^2}^2 \right\} \quad (3.46)$$

Nous utilisons alors la même approche que A.Chambolle et al. dans [21], notons $(c_{j,k,n})_{j \in \mathbb{Z}, 0 \leq k \leq 2^{l_j}, n \in \mathbb{Z}^2}$ et $(d_{j,k,n})_{j \in \mathbb{Z}, 0 \leq k \leq 2^{l_j}, n \in \mathbb{Z}^2}$ les coefficients issus des décompositions *contourlets* respectives de g et h . Etant donné que les *contourlets* forment une *frame* ajustée de borne égale à 1, on a (on notera $\Omega = \mathbb{Z} \times \llbracket 0, 2^{l_j} \rrbracket \times \mathbb{Z}^2$)

$$\|g\|_{L^2}^2 = \sum_{(j,k,n) \in \Omega} |c_{j,k,n}|^2 \quad (3.47)$$

Alors on peut réécrire 3.46 de la manière suivante

$$\sum_{(j,k,n) \in \Omega} |c_{j,k,n} - d_{j,k,n}|^2 + 2\delta \sum_{(j,k,n) \in \Omega} |d_{j,k,n}| \quad (3.48)$$

ce qui est équivalent à la formulation «coefficient à coefficient»

$$|c_{j,k,n} - d_{j,k,n}|^2 + 2\delta |d_{j,k,n}| \quad (3.49)$$

Or dans [21], les auteurs montrent que la solution à ce type de problème correspond au seuillage doux des coefficients $(c_{j,k,n})$ avec un seuil égal à δ . Donc $\hat{h} = CST(g, \delta)$, ce qui entraîne par dualité que $\hat{w} = g - CST(g, \delta)$, soit

$$\hat{w} = f - \hat{u} - \hat{v} - CST(f - \hat{u} - \hat{v}, \delta) \quad (3.50)$$

ce qui termine la démonstration. ■

L'algorithme numérique correspondant est donc le même que celui de J.F.Aujol et al. à ceci près que l'on remplace le seuillage des coefficients d'ondelette par un seuillage des coefficients *contourlets*.

Étape 1 : Initialisation : $u_0 = v_0 = w_0 = 0$

Étape 2 : Calcul de $w_{n+1} = f - u - v - CST(f - u - v, 2\delta)$

Étape 3 : Calcul de $v_{n+1} = P_{G_\mu}(f - u_n - w_{n+1})$

Étape 4 : Calcul de $u_{n+1} = f - v_{n+1} - w_{n+1} - P_{G_\lambda}(f - v_{n+1} - w_{n+1})$

Étape 5 : On stoppera la décomposition, soit par un critère d'arrêt sur l'erreur commise ($\max\{|u_{n+1} - u_n|, |v_{n+1} - v_n|, |w_{n+1} - w_n|\} \leq \epsilon$) ou par un critère sur un nombre prescrit d'itérations. Si le critère d'arrêt n'est pas satisfait on retourne à l'étape 2.

Les résultats obtenus avec l'utilisation des *contourlets* sont exposés figure 3.9. L'effet des *contourlets* permet d'obtenir des bords plus réguliers dans l'image géométrique u et ainsi atténuer une partie de l'information géométrique restante dans la partie texture.

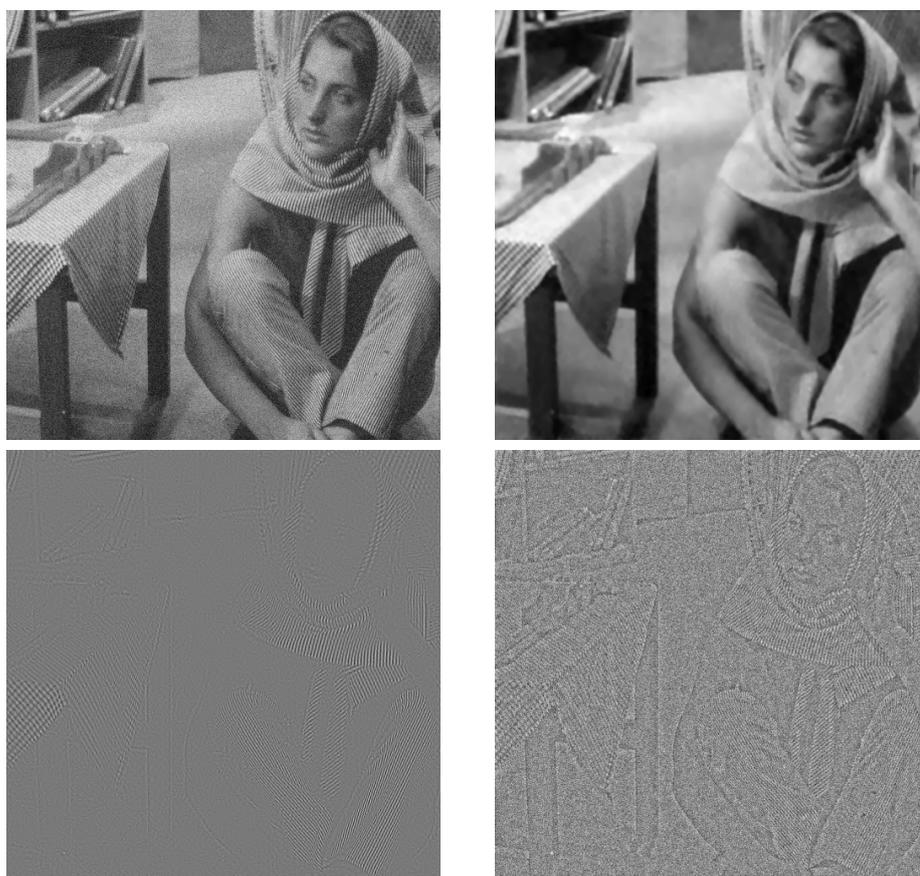


FIG. 3.9 – Algorithme JG3 : en haut se trouvent l'image bruitée ainsi que la composante u ; en bas les composantes v et w obtenue en utilisant un débruitage par *contourlets*.

3.6 Evaluation des algorithmes

Comme à la section 2.4 du chapitre précédent, nous proposons d'évaluer les performances des algorithmes F^{JG} , F^{AC2} et F^{JG3} . Pour cela, nous utilisons les mêmes images d'objets et de textures qu'au chapitre précédent auxquelles nous rajoutons une composante de bruit gaussien de moyenne nulle et $\sigma = 20$ (voir la figure 3.10). Nous mesurons la norme L^2 des erreurs tout d'abord entre le fond de référence et les composantes \tilde{u} puis entre les textures de référence et les composantes \tilde{v} obtenues par les différents algorithmes. De plus nous rajoutons les mesures, proposées à la section 1.6, effectuées sur la fonction d'autocorrélation des composantes \tilde{w} afin de pouvoir juger de la qualité du bruit extrait.

Les paramètres ont été choisis comme étant ceux donnant les meilleurs résultats visuels. Ces paramètres sont listés ci-après pour chacun des algorithmes :

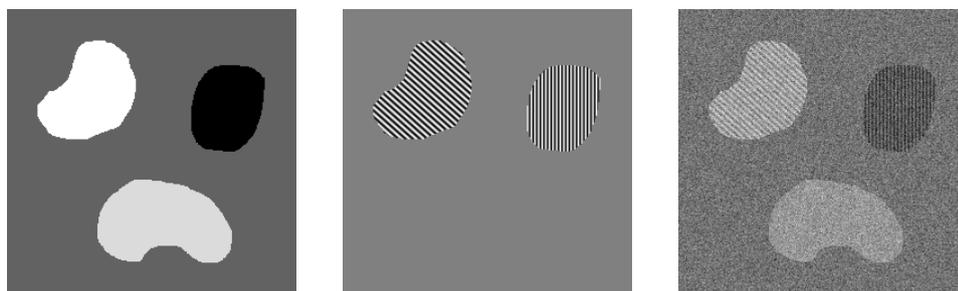


FIG. 3.10 – Les images de gauche sont les composantes objets et textures de référence. L'image finale de test à droite est composée des images de référence additionnées d'un bruit gaussien.

- Algorithme F^{JG} : $\lambda = 10$, $\mu_1 = 1000$, $\mu_2 = 100$ et une taille de fenêtre de 3×3 pixels,
- Algorithme F^{AC2} : $\lambda = 1$, $\mu = 500$ et $\delta = 9.4$ ($\kappa = 0.2$ et $\sigma = 20$),
- Algorithme F^{JG3} : $\lambda = 1$, $\mu = 500$ et $\delta = 23.5$ ($\kappa = 0.5$ et $\sigma = 20$).

La figure 3.11 donne les résultats obtenus en sortie des algorithmes.

Le tableau 3.1 donne les différentes mesures obtenues lors des tests.

Algorithme	F^{JG}	F^{AC2}	F^{JG3}
$\ \tilde{u} - u_{ref}\ _{L^2}$	792.8	873.5	984.6
$\ \tilde{v} - v_{ref}\ _{L^2}$	1844.9	2832.4	1598.6
$\ \gamma_w - \gamma_{w_{ref}}\ _{L^2}$	423.2	423.5	255.3

TAB. 3.1 – Mesure obtenues pour l'évaluation des algorithmes de décomposition u, v, w .

Nous pouvons constater que l'algorithme F^{JG} donne l'erreur la plus faible pour l'image des objets, une qualité de texture légèrement moins bonne que l'algorithme F^{JG3} à base de *contourlets*. La qualité du bruit est quant à elle équivalente à celle obtenue avec l'utilisation des espaces de Besov. L'algorithme F^{JG3} fournit une qualité de bruit nettement meilleure pour une qualité de texture elle aussi meilleure à l'instar d'une erreur sur les objets plus importante. Toutefois l'information contenue dans les composantes objets issues des algorithmes F^{JG} et F^{JG3} sont visuellement équivalentes.

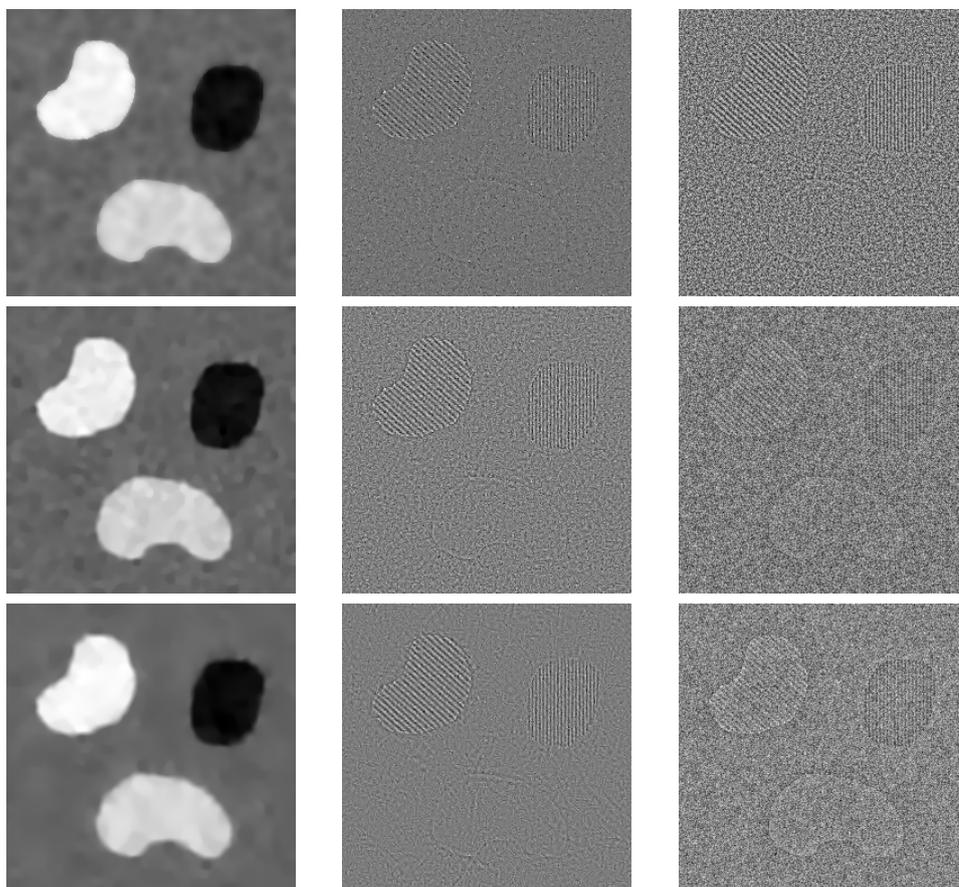


FIG. 3.11 – Résultats des différents algorithmes. Première ligne : algorithme F^{JG} , deuxième ligne : algorithme F^{AC2} et dernière ligne : algorithme F^{JG3} .

3.7 Bilan

Dans ce chapitre, nous proposons d'étendre le principe de décomposition d'image vu au chapitre précédent au cas d'images bruitées. Nous considérons le bruit comme étant une composante à part entière dans le sens où nous nous intéressons à ce que la qualité du bruit extrait soit elle aussi bonne.

Nous proposons dans un premier temps un algorithme basé sur une notion de régularisation locale adaptative. Nous rappelons le modèle proposé par J.F.Aujol et A.Chambolle utilisant les espaces de Besov pour modéliser le bruit (par seuillage des coefficients d'ondelettes). Nous proposons de remplacer les ondelettes par les *contourlets* vues au chapitre 1.

Des tests faits à partir d'images de référence nous montrent que globalement l'algorithme basé sur les *contourlets* nous donne les meilleurs résultats sur la qualité des textures et du bruit extrait. L'algorithme utilisant la régularisa-

tion locale adaptative donne des résultats équivalent au niveau des textures, une meilleure restitution des objets mais un bruit de moins bonne qualité. L'algorithme basé sur les ondelettes donne les moins bon résultats, les deux autres étant meilleurs, car ils respectent mieux la directionalité contenue dans les images.

Chapitre 4

Applications

Nous avons vu dans les chapitres précédents que les travaux disponibles dans la littérature sur la décomposition d'image montrent un gros effort en terme de modélisation de l'image. Même si à l'origine, l'idée était de faire de la restauration d'images, le champ d'applications possible de ce type de technique a été assez peu exploré. Nous proposons dans ce chapitre deux applications utilisant la décomposition en deux parties vue au chapitre 2. Nous nous intéressons dans un premier temps au traitement du bruit de type multiplicatif appliqué à des images issues de capteurs de type radar et imageur laser. Dans un deuxième temps, nous examinons une propriété particulière de la décomposition d'image mettant en avant les structures longilignes. Celle-ci, couplée à des algorithmes de détection, nous a permis d'envisager une application originale de détection de réseaux routiers en imagerie aérienne ou satellitaire.

4.1 Débruitage de bruit de type *speckle*

4.1.1 Principe

Cette application fut étudiée en collaboration avec J.F.Aujol avant la mise au point des algorithmes de décomposition u, v, w . Le bruit de type *speckle* est un bruit du type multiplicatif. On le rencontre par exemple dans des images du type radar (SAR) ou en imagerie active laser.

La méthode de décomposition exposée dans le chapitre 2 fait l'hypothèse de l'additivité $f = u + v$ qui est inadaptée à un bruit de type multiplicatif. La méthode la plus simple pour obtenir un modèle multiplicatif et conserver les algorithmes de décomposition u, v est tout simplement d'utiliser la fonction logarithme. Alors $f = uv$ devient :

$$\log(f) = \log(u) + \log(v) \tag{4.1}$$

Si l'on utilise les notations suivantes

$$\tilde{f} = \log(f) \quad (4.2)$$

$$\tilde{u} = \log(u) \quad (4.3)$$

$$\tilde{v} = \log(v) \quad (4.4)$$

on retrouve alors un modèle additif

$$\tilde{f} = \tilde{u} + \tilde{v} \quad (4.5)$$

Rq : en pratique on rajoutera 1 à f afin d'éviter d'avoir le logarithme de 0.

Il ne reste qu'à obtenir la décomposition \tilde{u} et \tilde{v} et de prendre l'exponentielle de \tilde{u} afin de retrouver la composante débruitée u .

4.1.2 Tests et résultats

Pour tester cet algorithme, nous disposons d'une image de synthèse sur laquelle un bruit multiplicatif de type gamma a été rajouté, une image de type SAR fournie par le CNES, une image flash laser. Ces images sont données respectivement fig.4.1, fig.4.2 et fig.4.3.

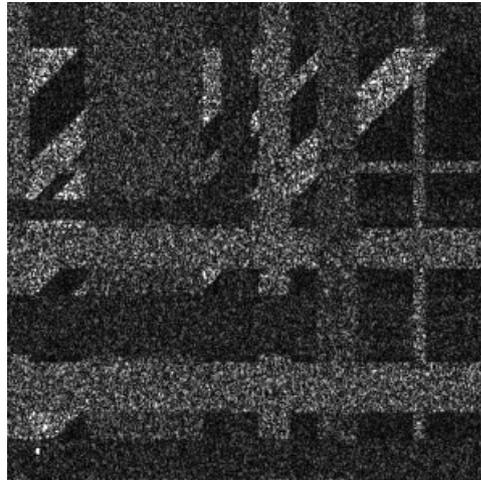


FIG. 4.1 – Image synthétique avec bruit multiplicatif de type Gamma.

Nous disposons également des images de référence pour les cas SAR et image synthétique (voir les figures 4.4 et 4.5).

Les figures 4.6, 4.7 et 4.8 donnent les résultats obtenus par l'algorithme de débruitage par décomposition d'images.

L'image laser donne le sentiment de n'être que partiellement débruitée mais les «artefacts» restants sont dûs aux effets de turbulences de l'atmosphère

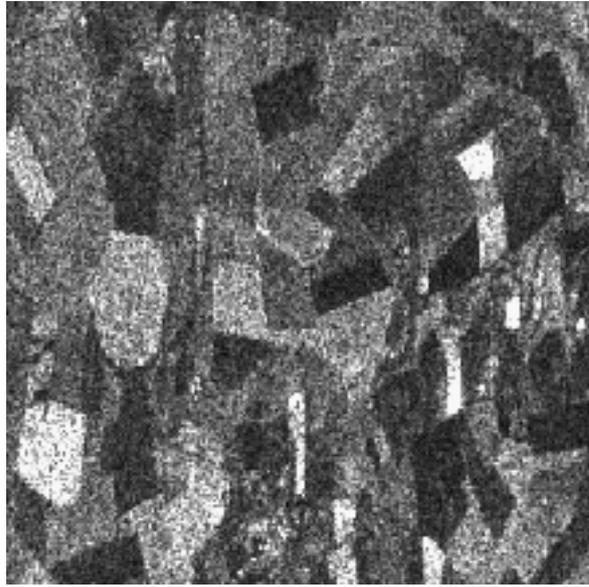


FIG. 4.2 – Image SAR initiale.

qui influent sur la propagation du rayon laser (d'où une sorte de texture surimposée à l'image).

Nous avons comparé nos résultats avec des images débruitées par les algorithmes de filtrage de Frost et Kuan (le choix de ces algorithmes s'est fait principalement sur un critère de disponibilité, ces filtres étant classiquement utilisés dans le domaine de l'imagerie SAR, nous avons pu y avoir accès au travers du logiciel ENVI à notre disposition). Nous rappelons les définitions de ces deux filtres en 4.1.1 et 4.1.2. La figure 4.9 regroupe les résultats obtenus en appliquant ces filtres sur l'image synthétique et sur l'image SAR.

Définition 4.1.1 *Le filtre de Frost est un filtrage de Wiener adaptatif qui revient à convoluer les pixels de l'image f par un masque m de taille fixée. Le masque m est une réponse impulsionnelle de type exponentielle définie par*

$$m(t) = \exp(-KC_f(t_0)|t|) \quad \text{où} \quad C_f(t_0) = \frac{\sigma_f(t_0)}{\bar{f}(t_0)} \quad (4.6)$$

où t_0 représente les coordonnées du centre de la fenêtre, $|t|$ la distance entre le pixel considéré et t_0 , K est un paramètre du filtre. Les quantités $\sigma_f(t_0)$ et $\bar{f}(t_0)$ sont respectivement l'écart-type et la moyenne de f sur la fenêtre.

Définition 4.1.2 *Le filtre de Kuan est basé sur le critère MMSE (Minimum Mean Square Error). On peut alors montrer que l'image estimée est donnée*

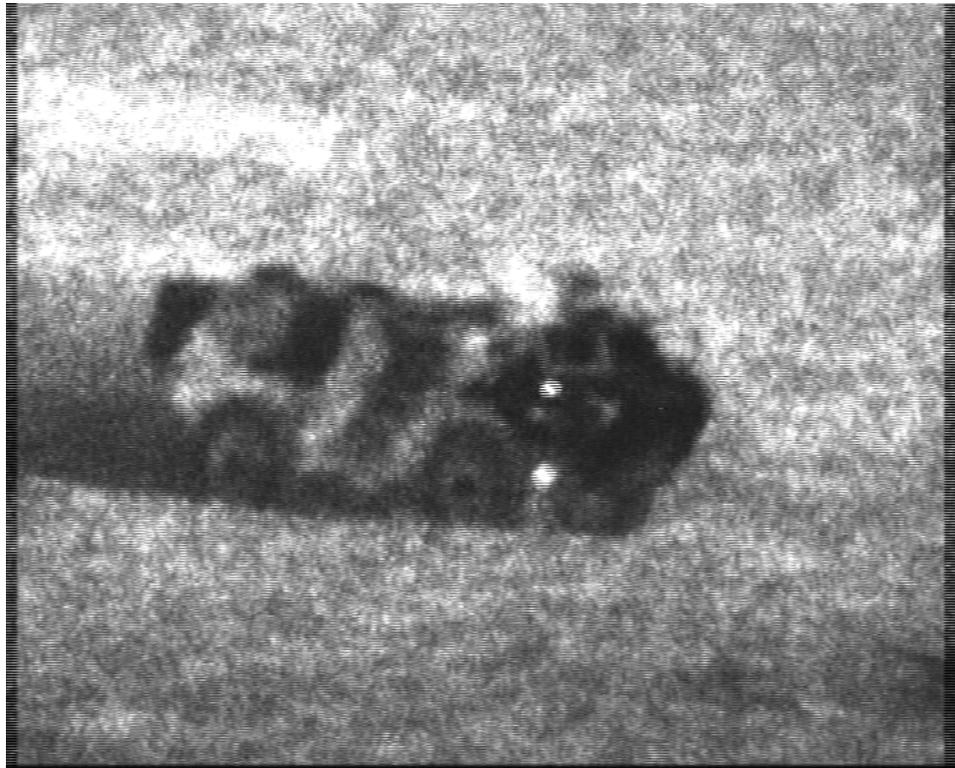


FIG. 4.3 – Image active laser initiale d'un blindé.



FIG. 4.4 – Image synthétique de référence.

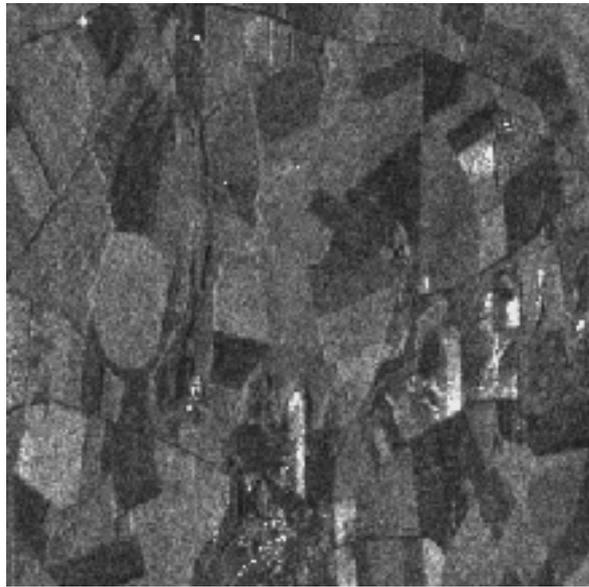
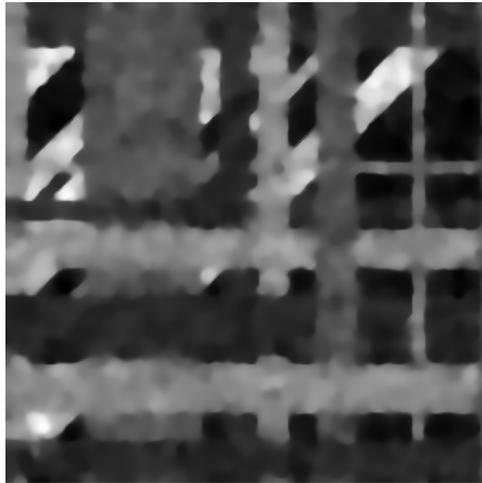


FIG. 4.5 – Image SAR de référence.

FIG. 4.6 – Image synthétique débruitée ($\mu = 500$, $\lambda = 0.5$, 50 itérations).

par

$$\hat{f} = \bar{f} + \frac{\sigma^2(f - \bar{f})}{\sigma^2 + (\bar{f}^2 + \sigma^2)/L} \quad \text{où} \quad \sigma^2 = \frac{L\sigma_f^2 - \bar{f}^2}{L+1} \quad (4.7)$$

où \bar{f} et σ_f représentent respectivement la moyenne et la variance de l'image f estimés sur une fenêtre de taille fixée et centrée sur le pixel à traiter, le paramètre L est le rapport (moyenne du bruit)/(écart-type du bruit). Ce

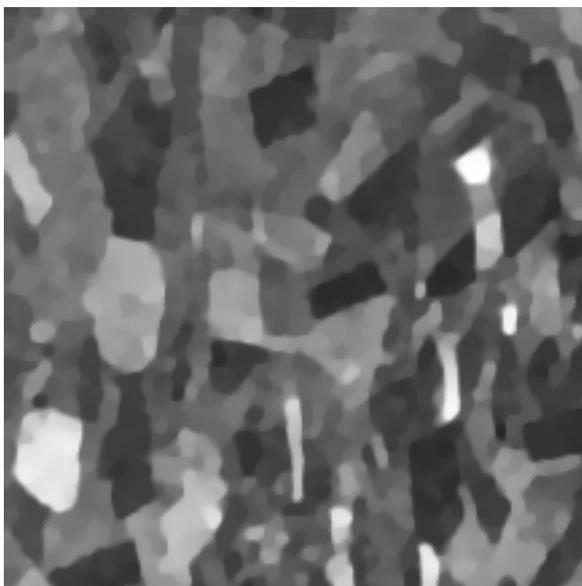


FIG. 4.7 – Image SAR débruitée ($\mu = 500$, $\lambda = 0.1$, 50 itérations).

paramètre peut être soit fournit par l'opérateur ou être estimé dans une région uniforme.

Si nous nous basons sur un critère purement visuel, nous constatons que les résultats obtenus dans le cas de la décomposition d'image sont meilleurs que ceux obtenus avec les filtres de Kuan et Frost. Toutefois ce résultat est à nuancer car si le contenu informationnel paraît effectivement mieux exploitable, nous constatons que les contrastes ne sont pas respectés ce qui peut être gênant dans le cas où la notion de signature de certains matériaux devrait être exploitée.



FIG. 4.8 – Image active laser blindé débruitée ($\mu = 100$, $\lambda = 1$, 50 itérations).

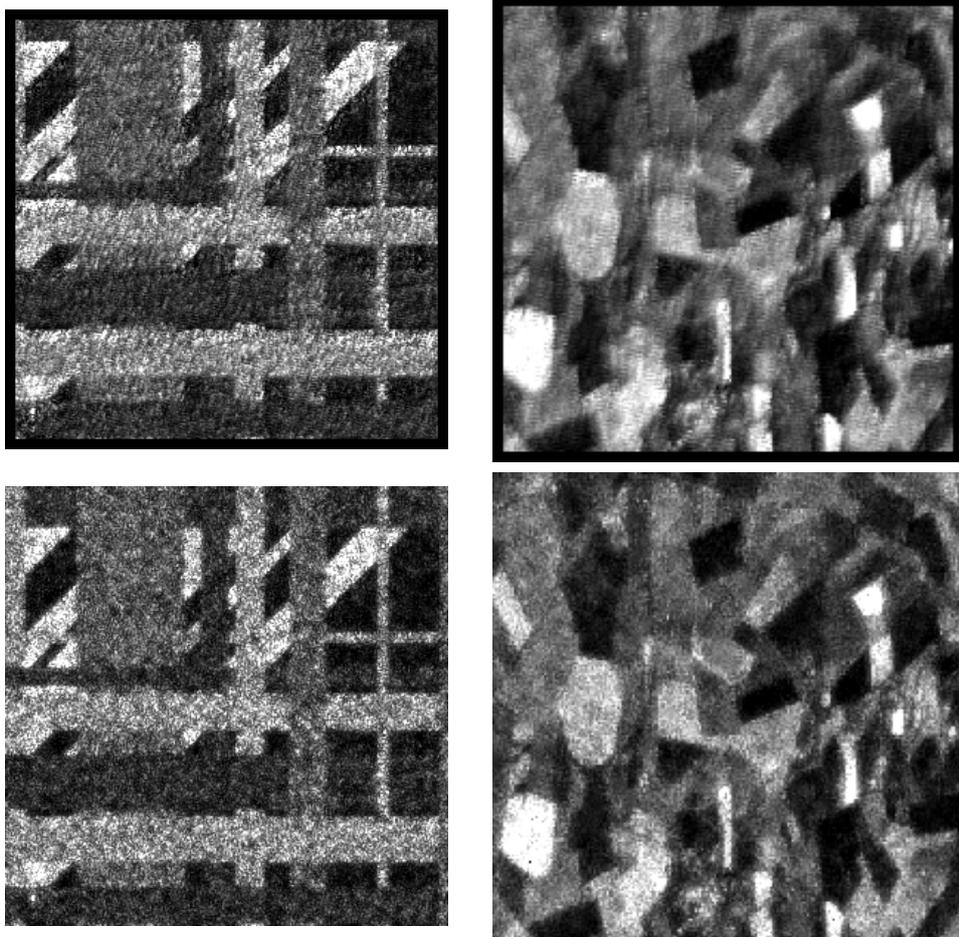


FIG. 4.9 – Filtrage SAR classiques : Première ligne : Filtre de Frost (masque de 15 pixels de large), Filtre de Frost (masque de 11 pixels de large), deuxième ligne : Filtre de Kuan (masque de 7 pixels de large), Filtre de Kuan (masque de 7 pixels de large).

4.2 Détection d'objets longilignes

Cette application se base sur une propriété de la décomposition permettant «d'intensifier» la présence d'objets du type longilignes. Cette propriété a trouvé une application dans la mise en place d'un outil destiné à soulager le travail des interprètes images chargés d'extraire les réseaux routiers sur des images aériennes ou satellitaires. En effet, malgré certaines avancées dans le domaine de la détection de réseaux routiers, ceux-ci se chargent toujours d'effectuer cette tâche fastidieuse manuellement. La quantité d'images à traiter étant en constante augmentation, le besoin de disposer d'outils d'aide semi-automatique est bien réel.

Le principe proposé se base sur une succession de trois phases :

- une première étape de décomposition d'image exacerbant les objets longilignes,
- une deuxième étape permettant d'effectuer une première phase de détection, basée sur l'algorithme de détection d'alignement issu de la théorie de la *Gestalt*,
- enfin une dernière étape dans laquelle chaque segment précédemment détecté est «transformé» en courbe déformable sur lesquelles nous appliquons un algorithme de type *snake*. Le but de cette phase est d'obtenir une détection finale la plus proche possible du réseau réel.

Dans la suite de ce chapitre, nous présentons ces différentes étapes ainsi que quelques résultats obtenus.

4.2.1 Une propriété inattendue !

Lors d'expérimentations sur la décomposition $u + v$, nous avons remarqué la propriété suivante : si l'on regarde la décomposition obtenue à partir de l'image en haut de la figure 4.10, nous pouvons remarquer que la portion de chemin sur laquelle roule le véhicule est plus visible dans la composante texture (cf. image du bas).

Au premier abord, ce comportement semble étrange car dans notre esprit une route est plutôt considérée comme un «objet» fin et longiligne. D'un point de vue mathématique, il est donc naturel de modéliser une route par une fonction indicatrice longue et peu large que l'on s'attend donc à retrouver dans la composante u . Le lemme suivant montre pourtant qu'il est normal que ce type d'objet apparaisse effectivement dans l'espace des textures. Pour cela, il suffit de montrer que la norme de ce type de fonction indicatrice dans l'espace des textures n'est pas nulle.

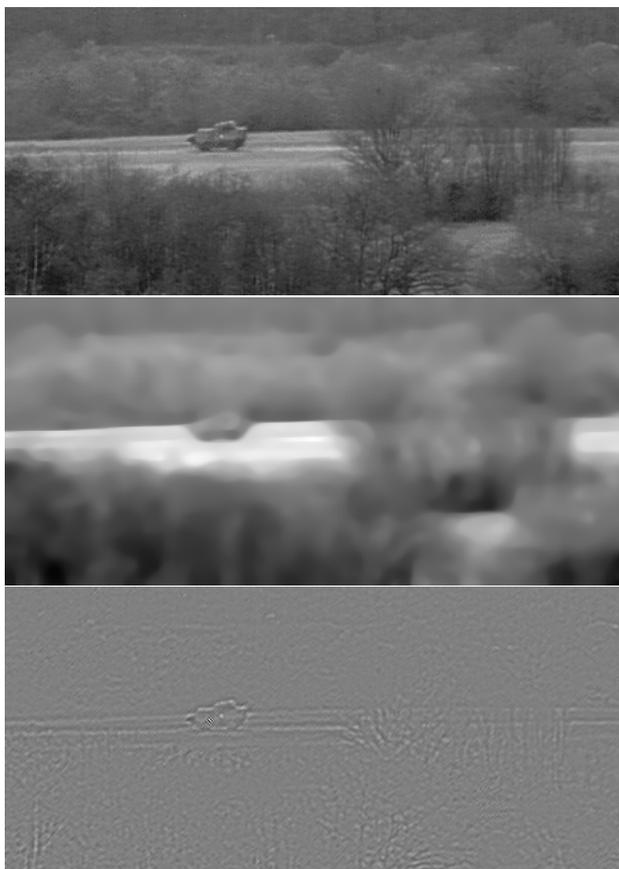


FIG. 4.10 – Constatation expérimentale : le chemin est mis en avant dans la partie texture.

Lemme 4.2.1 Soit f la fonction indicatrice de l'ensemble E_N défini par

$$E_N = [0, 1] \times [0, N] \quad (4.8)$$

quand N est grand f correspondra à une route dans l'image. On a alors

$$\|f\|_G \in \left[\left(2 + \frac{2}{N}\right)^{-1}, \frac{1}{2} \right] \quad (4.9)$$

Preuve:

Soit la fonction θ définie par

$$\theta(t) = \begin{cases} t - \frac{1}{2} & \text{si } 0 < t < 1 \\ \frac{1}{2} & \text{si } t \geq 1 \\ -\frac{1}{2} & \text{si } t \leq 0 \end{cases} \quad (4.10)$$

alors

$$f = \chi_{E_N} = \frac{\partial}{\partial x_1} (\theta(x_1)\chi_{[0,N]}(x_2)) \quad (4.11)$$

on a alors

$$\|\theta(x_1)\chi_{[0,N]}(x_2)\|_{L^\infty} = \frac{1}{2} \quad (4.12)$$

Ce qui nous fournit la borne supérieure. La borne inférieure est obtenue en appliquant la propriété suivante

$$\|f\|_G \|f\|_{BV} \geq \int f^2(x) dx = |E_N| \quad (4.13)$$

On en déduit

$$\|f\|_G \geq \frac{N}{2N+2} \quad (4.14)$$

ce qui achève la démonstration. ■

La figure 4.11 illustre cette propriété sur une imagerie extraite d'une image aérienne. On vérifie bien le fait que le réseau routier est mis en avant dans l'image des textures.

Nous devons maintenant extraire le réseau routier proprement dit. Le choix d'un algorithme de détection sera basé sur le fait que nous cherchons à détecter des objets longilignes.

4.2.2 Algorithme de détection des alignements.

Principe

Au paragraphe précédent, nous avons vu que les objets longilignes pouvaient être mis en avant dans la partie texture de l'image, il nous faut maintenant mettre en place un algorithme permettant de détecter des alignements de pixels dans l'image. Une méthode bien adaptée à ce type d'objet est la méthode de détection des alignements issue de la théorie de la Gestalt. Dans les pages qui suivent, nous commençons par rappeler le principe et les résultats théoriques de la méthode, pour cela nous donnons quelques définitions de base en reprenant les références [26],[27].

La méthode est basée sur le principe de Helmholtz : un événement¹ sera d'autant plus significatif qu'il aura peu de chance d'apparaître (autrement dit un événement ayant une probabilité d'apparition très faible sera effectivement significatif si il apparaît).

¹Un événement peut, par exemple, être une configuration géométrique particulière («*x* points sont alignés le long d'un segment de longueur *l*»).

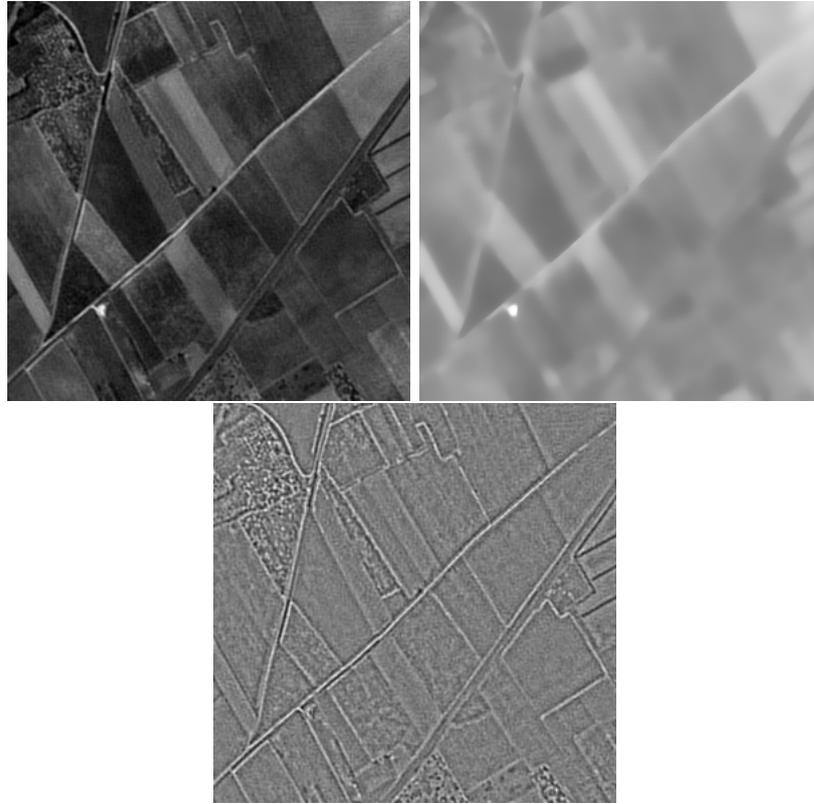


FIG. 4.11 – Image aérienne (en haut à gauche) et sa décomposition (composante BV en haut à droite, composante texture en bas).

En pratique, il est souvent difficile de calculer la probabilité d'un évènement, on préfère calculer l'espérance de ces évènements étant donné que l'on cherche des probabilités faibles (inégalité de Markov : soit C un évènement, $\mathbb{P}(C) \leq \mathbb{E}(C)$, de plus en pratique le calcul d'une espérance est un simple comptage des différents évènements). En se basant sur ce principe, il est possible de définir la notion d'évènement ϵ -significatif :

Définition 4.2.2 (*Evènement ϵ -significatif*)

Nous dirons qu'un évènement est ϵ -significatif si l'espérance dans l'image du nombre d'occurrences de cet évènement est inférieure à ϵ (sous l'hypothèse d'une distribution de la fréquence d'apparition uniforme).

Appliquons cette définition à un évènement E_i ayant certaines caractéristiques, nous notons sa probabilité d'apparition p . En faisant l'hypothèse d'indépendance entre les évènements, nous pouvons déduire que la probabilité qu'au moins k évènements ayant les mêmes caractéristiques parmi les

n observés est

$$\mathcal{B}(n, k, p) = \sum_{i=k}^n \binom{l}{k} p^i (1-p)^{n-i} \quad (4.15)$$

(\Leftrightarrow queue de la loi binômiale)

Nous pouvons ensuite définir la notion du Nombre de Fausses Alarmes (*NFA*) (c'est à dire l'espérance du nombre d'évènements arrivant par hasard) :

$$NFA = N_{conf} \mathcal{B}(n, k, p) \quad (4.16)$$

où N_{conf} est le nombre de tests effectués sur l'image. On dira alors qu'un évènement est ϵ -significatif si $NFA \leq \epsilon$.

Pour pouvoir appliquer ce principe à la détection d'alignement, il nous faut commencer par définir d'un point de vue mathématique la notion d'alignement de points. Pour cela, nous utilisons des outils de géométrie appliquée. Soit v l'image en niveau de gris de taille $N \times N$ à analyser, nous pouvons définir le vecteur direction du gradient

$$dir(i, j) = \frac{1}{\|\vec{D}\|} \vec{D} \quad (4.17)$$

où

$$\vec{D} = \frac{1}{2} \begin{pmatrix} -[v(i, j+1) + v(i+1, j+1)] + [v(i, j) + v(i+1, j)] \\ [v(i+1, j) + v(i+1, j+1)] - [v(i, j) + v(i, j+1)] \end{pmatrix} \quad (4.18)$$

on dira alors que deux points X et Y ont la même direction avec la précision $p = \frac{1}{n}$ (en pratique on choisit $2 < n \leq 16$) si

$$Angle(dir(X), dir(Y)) \leq \frac{2\pi}{n} \quad (4.19)$$

En se basant sur le principe de Helmholtz vu précédemment, on considère les directions en chacun des points d'une image comme étant une variable aléatoire distribuée de manière uniforme. On peut donc interpréter p comme étant la probabilité que deux points indépendants aient la même direction. Soit A un segment de longueur l , composé des points x_1, \dots, x_l , on notera X_i la variable aléatoire telle que

$$X_i = \begin{cases} 1 & \text{si le pixel } x_i \text{ est aligné avec la direction de } A. \\ 0 & \text{sinon} \end{cases} \quad (4.20)$$

Les variables X_i sont alors distribuées selon une loi de Bernoulli

$$P[X_i = 1] = p \quad \text{et} \quad P[X_i = 0] = 1 - p \quad (4.21)$$

On peut ensuite définir une variable aléatoire représentant le nombre de points x_i ayant la bonne direction par

$$S_l = X_1 + X_2 + \dots + X_l \quad (4.22)$$

En faisant l'hypothèse de l'indépendance des X_i , la variable S_l est donnée par une distribution binômiale

$$P[S_l = k] = \binom{l}{k} p^k (1-p)^{l-k} \quad (4.23)$$

Nous cherchons maintenant à savoir si un segment de longueur l donné est ϵ -significatif ou non par rapport aux autres segments de l'image. Pour cela, notons $m(l)$ le nombre de segments orientés de longueur l dans l'image (que l'on supposera de taille $N \times N$). Il est possible de montrer que

$$\sum_{l=1}^{l_{max}} m(l) \approx N^4 \quad (4.24)$$

On définit alors le seuil de détection par

Définition 4.2.3 On appellera «seuil de détection» la famille des valeurs positives notées $w(l, \epsilon, N)$, pour $1 \leq l \leq l_{max}$ telle que

$$\sum_{l=1}^{l_{max}} w(l, \epsilon, N) m(l) \leq \epsilon \quad (4.25)$$

Les auteurs de [26, 27] donnent alors une définition générale d'un segment ϵ -significatif.

Définition 4.2.4 Un segment de longueur l est ϵ -significatif dans une image $N \times N$ si il contient au moins $k(l)$ points ayant leur direction alignée avec celle du segment. La variable $k(l)$ est donnée par

$$k(l) = \min\{k \in \mathbb{N}, \mathbb{P}[S_l \geq k] \leq w(l, \epsilon, N)\} \quad (4.26)$$

Le problème de cette formulation est que l'on ne connaît pas, a priori, la loi de probabilité \mathbb{P} . Afin de palier ce problème, on remarque que si l'on note, pour $1 \leq i \leq N^4$, e_i l'évènement «le i -ème segment est ϵ -significatif» et χ_{e_i} la fonction caractéristique de l'évènement e_i alors on a

$$\mathbb{P}[\chi_{e_i} = 1] = \mathbb{P}[S_{l_i} \geq k(l_i)], \quad l_i = \text{longueur du } i\text{-ème segment} \quad (4.27)$$

Soit R la variable aléatoire représentant le nombre exact de fois où les évènements e_i apparaissent simultanément dans l'image. On a donc $R = \chi_{e_1} + \chi_{e_2} + \dots + \chi_{e_{N^4}}$ et le calcul de l'espérance donne

$$\mathbb{E}(R) = \mathbb{E}(\chi_{e_1}) + \mathbb{E}(\chi_{e_2}) + \dots + \mathbb{E}(\chi_{e_{N^4}}) = \sum_{l=1}^{l_{max}} m(l) \mathbb{P}[S_l \geq k(l)] \quad (4.28)$$

En pratique, il est beaucoup plus facile de calculer l'espérance de R que de connaître la loi de probabilité $\mathbb{P}[S_l \geq k(l)]$ (cette loi n'étant pas triviale à déterminer, notamment car les événements e_i ne sont pas indépendants). En utilisant les relations 4.26 et 4.25, on obtient

$$\mathbb{E}(R) \leq \sum_{l=1}^{l_{max}} w(l, \epsilon, N) m(l) \leq \epsilon \quad (4.29)$$

Il reste à discuter du choix de la famille de seuils de détection $w(l, \epsilon, N)$. Dans [26, 27] les auteurs font remarquer que le nombre de segments de longueur l dans une image de taille $N \times N$ est de l'ordre de N^3 donc $m(l) \approx N^3$. Si l'on ne s'intéresse qu'aux segments de longueur l alors on peut prendre $w(l, \epsilon, N) = \frac{\epsilon}{N^3}$. Si maintenant nous faisons le choix de ne pas privilégier des longueurs de segment particulières, un choix possible est de supposer que la longueur des segments dans l'image est répartie suivant une loi uniforme, ce qui nous donnera :

$$\forall l \geq 1 \quad w(l, \epsilon, N) = \frac{\epsilon}{N^4} \quad (4.30)$$

La définition finale d'un segment ϵ -significatif est la suivante :

Définition 4.2.5 *Un segment A de longueur l est ϵ -significatif dans une image $N \times N$ si il contient au moins $k(l)$ points ayant leur direction alignée avec celle du segment A , où $k(l)$ est donné par*

$$k(l) = \min\{k \in \mathbb{N}, \mathbb{P}[S_l \geq k] \leq \frac{\epsilon}{N^4}\} \quad (4.31)$$

Enfin, nous pouvons définir le Nombre de Fausses Alarmes :

Définition 4.2.6 *Soit A un segment de longueur l_0 composé de k_0 points ayant leur direction alignée sur celle de A . Le Nombre de Fausses Alarmes est donné par*

$$NFA(l_0, k_0) = N^4 \mathbb{P}[S_{l_0} \geq k_0] = N^4 \sum_{k=k_0}^{l_0} \binom{l_0}{k} p^k (1-p)^{l_0-k} \quad (4.32)$$

Dans [26, 27] les auteurs montrent la proposition suivante

Proposition 4.2.7 *Soit $A = (l_0, k_0)$ un segment alors $NFA(A)$ est le plus petit ϵ tel que A soit ϵ -significatif.*

Les expériences menées par les auteurs de [26, 27] montrent qu'un même événement peut donner lieu à plusieurs détections (par exemple pour un même alignement on aura plusieurs segments parallèles, des problèmes sur la précision des directions,...) la cause étant en fait la précision limitée dans les images (dûe au procédé d'acquisition). Or des études psychovisuelles montrent que l'être humain corrige ce problème grâce au principe d'exclusion :

Définition 4.2.8 (*Principe d'exclusion*) Soient A et B deux évènements obtenus à partir de la même loi de la Gestalt. Alors aucun point x n'est autorisé à appartenir à A et B simultanément. Donc $x \in A$ ou $x \in B$.

On va donc appliquer ce principe à la détection des alignements. Pour cela, il nous faut tout d'abord définir la notion de segment maximal :

Définition 4.2.9 (*Segment maximal*) Un segment A est dit maximal si

1. Il ne contient pas d'autres segments significatifs

$$\Leftrightarrow \forall B \subset A, NFA(B) \geq NFA(A) \quad (4.33)$$

2. Il n'est pas contenu par un autre segment significatif

$$\Leftrightarrow \forall B \supset A, NFA(B) > NFA(A) \quad (4.34)$$

On dira qu'un segment est maximal significatif si il est à la fois maximal et significatif.

Les mêmes auteurs montrent qu'un segment maximal a les propriétés suivantes :

- les deux extrémités de A ont leurs directions alignées avec celles de A ,
- les deux points suivant les extrémités de A n'ont pas leur directions alignées avec celle de A .

Le principe d'exclusion appliqué dans le cadre de la détection d'alignements nous donne l'algorithme suivant

1. On établit une liste L de tous les intervalles I sur une ligne commençant par un point aligné précédé par un point non-aligné et finissant par un point aligné suivi par un point non aligné.
2. On considère toutes les paires (I, J) (où $I, J \in L$) telles que $I \subset J$. Si J est plus ou autant significatif que I alors on retire I de la liste.
3. On retourne à l'étape 1 jusqu'à ce qu'il n'y ait plus de paires possibles.

Au final, on obtient tous les segments maximaux significatifs de la ligne. On applique la méthode pour toutes les lignes possibles de l'image afin d'obtenir tous les segments maximaux significatifs de l'image.

Expérimentations

Les images à notre disposition sont des images acquises dans le domaine optique sur la région de la ville de Marseille (©CNES-DGA). Dans un premier temps, nous avons extrait une portion contenant peu de routes (voir figure 4.12) afin de valider le concept sur une image de taille raisonnable

(300×300). Ensuite nous appliquerons le même algorithme de détection sur deux images dont les tailles correspondent plus aux images réellement manipulées. L'une de ces deux images contient majoritairement une zone de type agricole et l'autre image contient à la fois une zone de type agricole et une zone de type urbaine (voir les figures 4.13 et 4.14). Les images présentées ici ont été retouchées afin d'augmenter leur contraste pour des raisons de visibilité dans ce manuscrit.



FIG. 4.12 – Imagerie de test

Nous considérons ici que les images ne sont pas bruitées et par conséquent utilisons une décomposition en deux parties obtenue grâce à la fonctionnelle $F_{\lambda, \mu}^{AU}(u, v)$ vue au paragraphe 2.2.5. La première étape a consisté tout d'abord à étudier l'influence des paramètres de l'algorithme de décomposition d'image. Pour cela, nous avons fait varier ces paramètres dans les gammes suivantes $\mu \in [0.1; 5000]$ et $\lambda \in [0.1; 100]$ avec un pas variable augmentant au fur et à mesure que les paramètres eux-mêmes augmentaient. Pour des raisons de temps de calcul, nous avons fait ce test uniquement sur l'imagerie de test. En pratique, nous avons obtenu 1080 décompositions différentes. Les figures 4.15, 4.16 et 4.17 illustrent le comportement de la décomposition vis à vis de ces paramètres.

Tout d'abord, si nous examinons l'influence du paramètre μ , nous pouvons observer pour $\mu = 0.2$ (donc μ faible) que la partie texture est plus proche d'une image de «bruit» où les routes ne sont pas spécialement mise en avant par rapport au reste de l'image. A l'inverse, le cas $\mu = 5000$ permet d'observer nettement le réseau routier et quelques textures présentes dans l'image. Ce comportement s'explique tout simplement par la définition même de la norme sur l'espace G ; en effet nous avons vu que, plus le signal était «oscillant», plus sa norme sur cet espace était faible. Or rappelons que lorsque



FIG. 4.13 – Zone agricole



FIG. 4.14 – Zone agricole + urbain

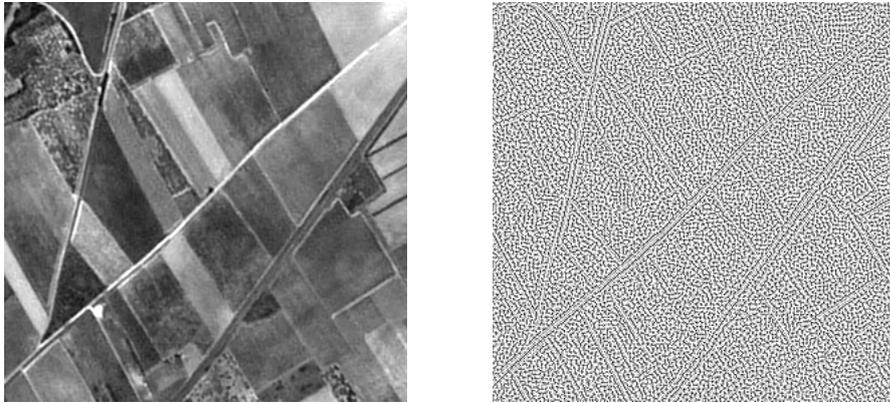


FIG. 4.15 – Décomposition de l'imagette obtenue pour $\mu = 0.2$ et $\lambda = 0.1$.

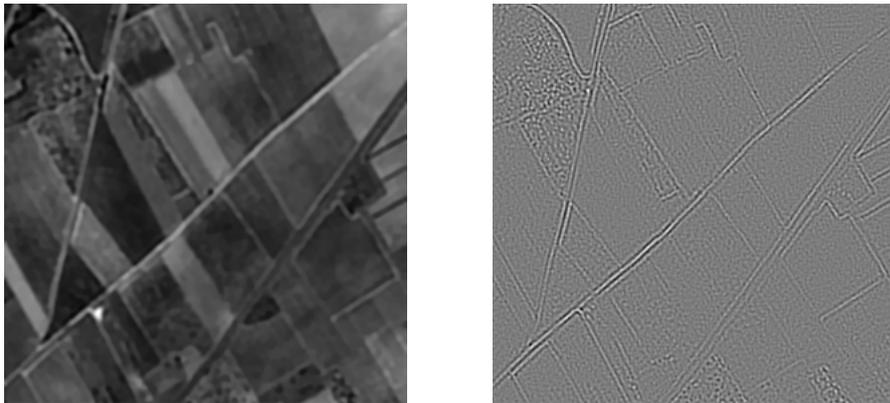


FIG. 4.16 – Décomposition de l'imagette obtenue pour $\mu = 5000$ et $\lambda = 0.1$

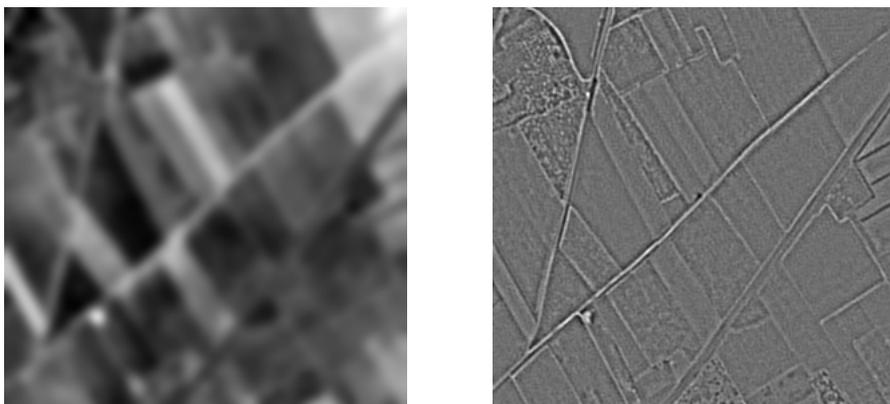


FIG. 4.17 – Décomposition de l'imagette obtenue pour $\mu = 5000$ et $\lambda = 90$

l'on fixe une valeur de μ faible dans l'algorithme, cela correspond à la norme maximale de la partie texture. Par conséquent cette partie extraite contiendra en réalité les parties bruitées de l'image. Tandis que lorsque l'on fixe une valeur élevée de μ , nous allons effectivement obtenir uniquement des textures et en particulier le réseau routier.

Deuxième examen, l'influence du paramètre λ . Nous voyons très nettement que ce paramètre joue sur le «lissage» de la composante dans l'espace BV . En effet, plus λ est important, plus la partie BV sera lissée, cela signifie que ce qui peut être vu comme des transitions dans l'image (notamment comme les bords des parcelles des champs) sera considéré comme une oscillation dans l'image et donc rejeté dans la composante texture. L'inconvénient pour l'application que nous visons dans ce paragraphe est que les bords de ces parcelles deviennent elles-aussi non négligeables par rapport au réseau routier. Nous risquons donc de détecter des alignements qui ne font pas partie de ce que nous cherchons.

Cela nous amène maintenant à la deuxième étape : la détection elle-même. Nous appliquons l'algorithme de détection des alignements à chacune des composantes textures vues précédemment. Premier résultat : la détection sur la partie texture de la figure 4.15 ne donne aucun résultat (aucun segment détecté), en effet l'image étant trop bruitée, aucun alignement n'est considéré comme significatif. La figure 4.18 montre les segments détectés sur les deux autres composantes textures (à gauche pour $\lambda = 0.1$, à droite pour $\lambda = 90$). Les paramètres de l'algorithme de détection des alignements sont les suivants : une précision de 16 (valeur par défaut), un nombre de directions de recherche de 192 (afin d'avoir une résolution en terme de direction suffisamment fine), une norme minimum du gradient définissant une direction égale à 2 (valeur par défaut) et $\epsilon = 10^{-15}$ (cette valeur a été retenue après quelques essais du fait qu'elle donnait le minimum de fausses alarmes, des valeurs plus faibles donnent des résultats similaires).

Le comportement de l'algorithme de détection des alignements est conforme à ce que l'on attendait, les routes sont effectivement détectées. Dans le cas où $\lambda = 90$, l'algorithme détecte effectivement les bords les plus prononcés des parcelles agricoles. Le principe de détection de réseau routier est donc viable sur cette image de test ; il nous faut donc le tester sur les images plus complexes des figures 4.13 et 4.14. Etant donné les résultats précédents et la motivation de minimiser les temps de calculs, nous choisissons directement les paramètres μ et λ respectivement égaux à 5000 et 0.2. Les composantes textures obtenues sont visibles sur les figures 4.19 et 4.20 (la dynamique de ces images a été retouchée afin d'en améliorer la visualisation dans ce manuscrit).

Les résultats de l'algorithme de détection des alignements sont donnés sur les figures 4.21 et 4.22. Les paramètres utilisés sont respectivement une pré-

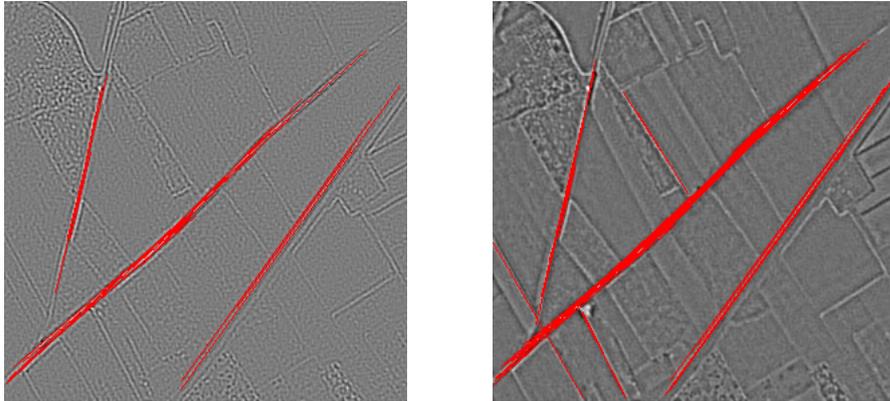


FIG. 4.18 – Segments détectés pour $\mu = 5000$, $\lambda = 0.1$ à gauche et $\lambda = 90$ à droite.

cision de 8, 192 orientations de recherche, une norme minimum du gradient définissant une direction égale à 2 et $\epsilon = 10^{-12}$ pour la figure 4.21 ; une précision de 12, 192 orientations de recherche, une norme minimum du gradient définissant une direction égale à 2 et $\epsilon = 10^{-6}$ pour la figure 4.22. Dans les deux cas, le réseau routier est relativement bien extrait hormis quand celui-ci a une forme courbée. En effet, l'algorithme de détection favorise l'extraction des segments longs, donc une route ayant un rayon de courbure non négligeable ne peut pas être décomposée comme une succession de segments. Afin d'extraire le réseau routier, les paramètres de l'algorithme (notamment le choix de ϵ) sont choisis de telle sorte que l'algorithme détecte bien des segments qui ne sont pas de longueur importante. Ceci a pour conséquence la détection de quelques bords de parcelles agricoles (un compromis est à faire entre pouvoir de détection et fausses détections).

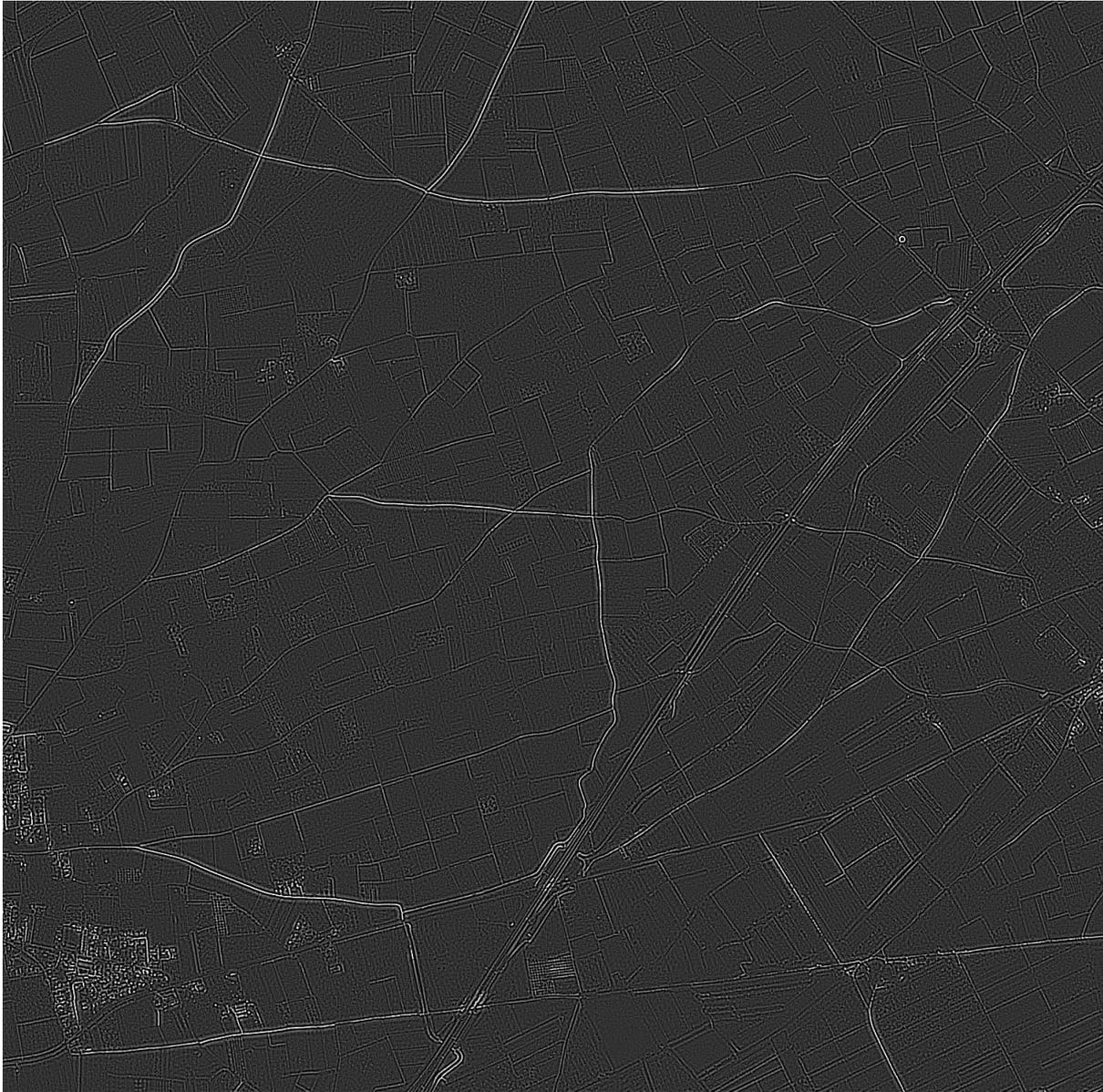


FIG. 4.19 – Composante texture pour la zone agricole



FIG. 4.20 – Composante texture pour la zone agricole et urbaine

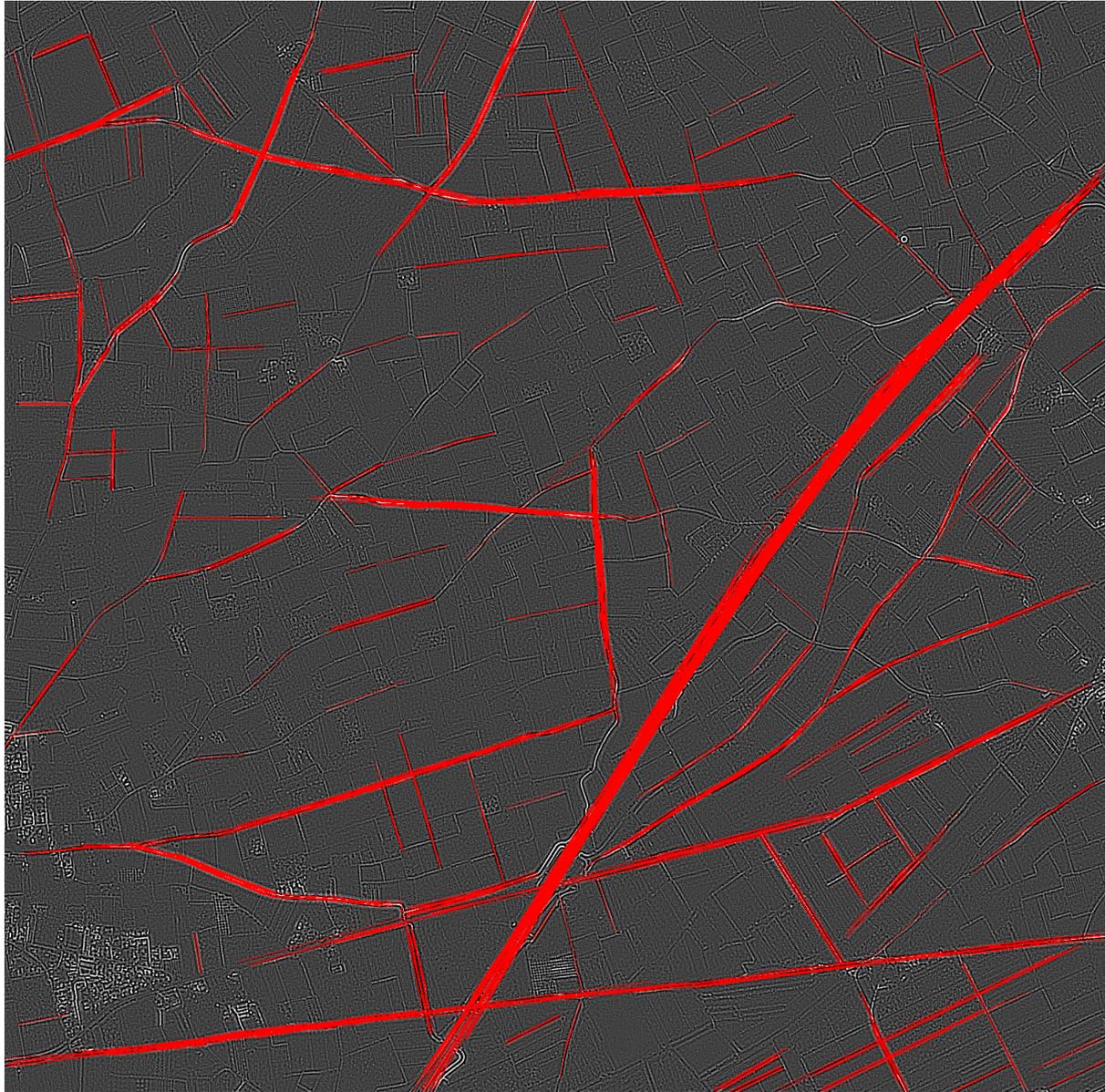


FIG. 4.21 – Segments détectés pour la zone agricole

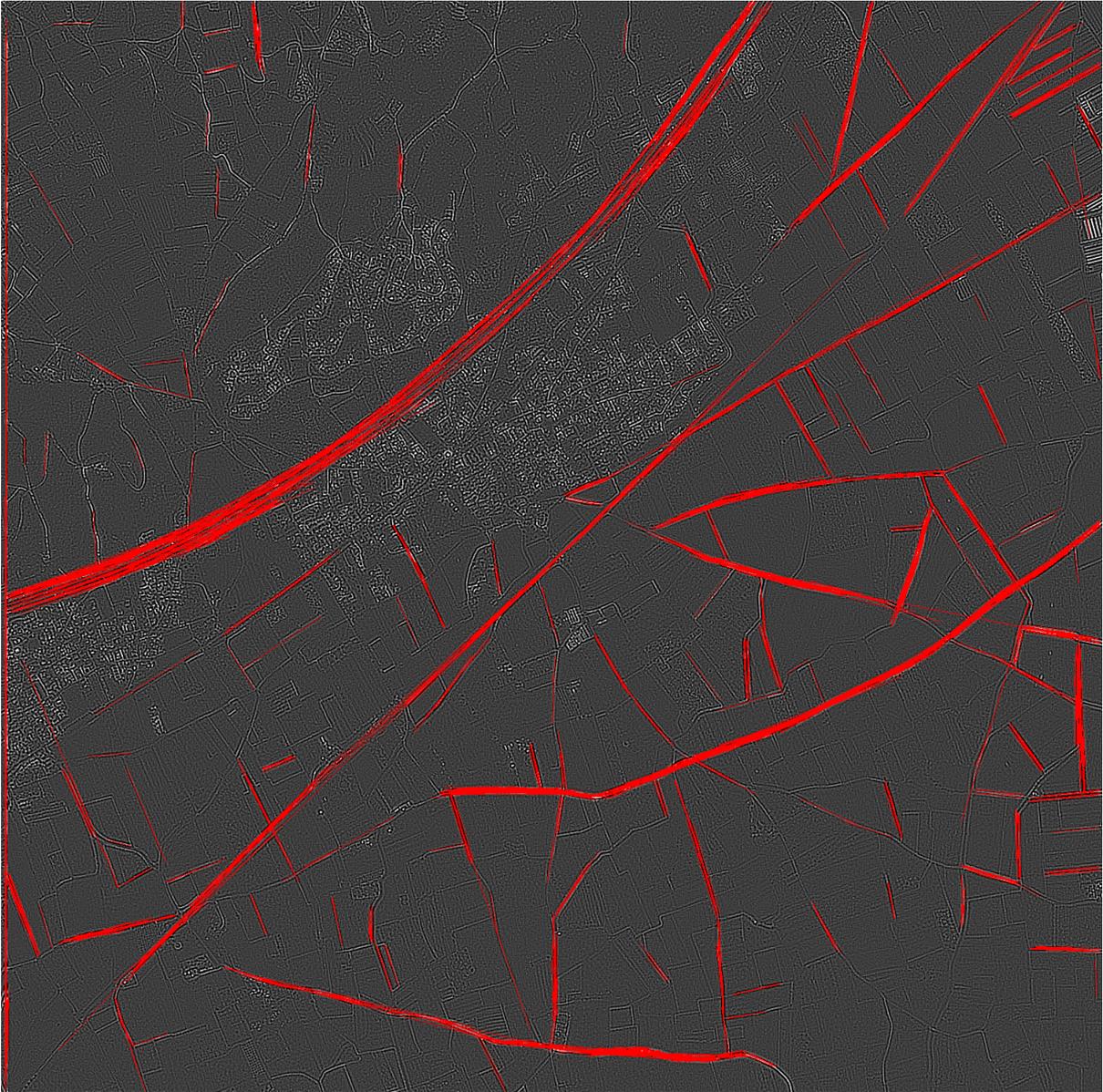


FIG. 4.22 – Segments détectés pour la zone agricole et urbaine

Fusion de segments

Nous venons de voir que plusieurs segments pouvaient être fournis par l'algorithme pour un même alignement. Nous proposons ici de mettre en place quelques règles permettant de fusionner certaines configurations de groupes de segments. Ces règles sont regroupées dans le tableau 4.23.

Ces règles nous permettent de réduire le taux de sur-détection. Un exemple est donné sur la figure 4.24. L'étape suivante va consister à affiner la géométrie des routes extraites à l'aide d'un algorithme de type *snake*.

4.2.3 Adaptation à la géométrie réelle

Nous venons de voir qu'il était possible de détecter des segments correspondant aux portions longilignes du réseau routier en appliquant l'algorithme de détection des alignements sur la composante texture de l'image initiale. Toutefois ce résultat n'est pas encore tout à fait satisfaisant car sur des images aussi complexes que celle présentées sur les figures 4.19 et 4.20, le réseau routier n'a pas nécessairement une forme purement longiligne. L'idée est donc de convertir nos segments détectés en courbes polygonales déformables et d'appliquer un algorithme du type *snake*.

Etant donné que le but initial de cette application est de proposer un outil interactif pouvant être utilisé par des opérateurs non spécialisés dans le domaine des contours déformables, nous avons choisi d'utiliser un modèle basé sur une déformation de courbes polygonales. Le modèle choisit est rappelé à l'annexe B, celui-ci a été développé dans le cadre d'un besoin en extraction de formes dans des images relativement bruitées (d'où l'utilisation de la notion de statistique pour guider la courbe) et avec une contrainte de temps de calcul. Les étapes du processus sont donc les suivantes :

- transformation de chaque segment en une courbe polygonale ayant un pas d'échantillonnage fixé,
- évolution de la courbe,
- retouche éventuelle de l'opérateur via une interface appropriée.

La figure 4.25 illustre sur l'imagerie le type de résultat final obtenu. Les figures 4.26 et 4.27 quant à elles illustrent les résultats obtenus sur les images à plus grande échelle 4.13 et 4.14.

4.2.4 Bilan

Nous venons de voir que le principe exposé dans les paragraphes précédents donnait des résultats prometteurs. En accord avec la théorie et afin de privilégier les objets longilignes dans la composante texture, nous avons vu qu'il

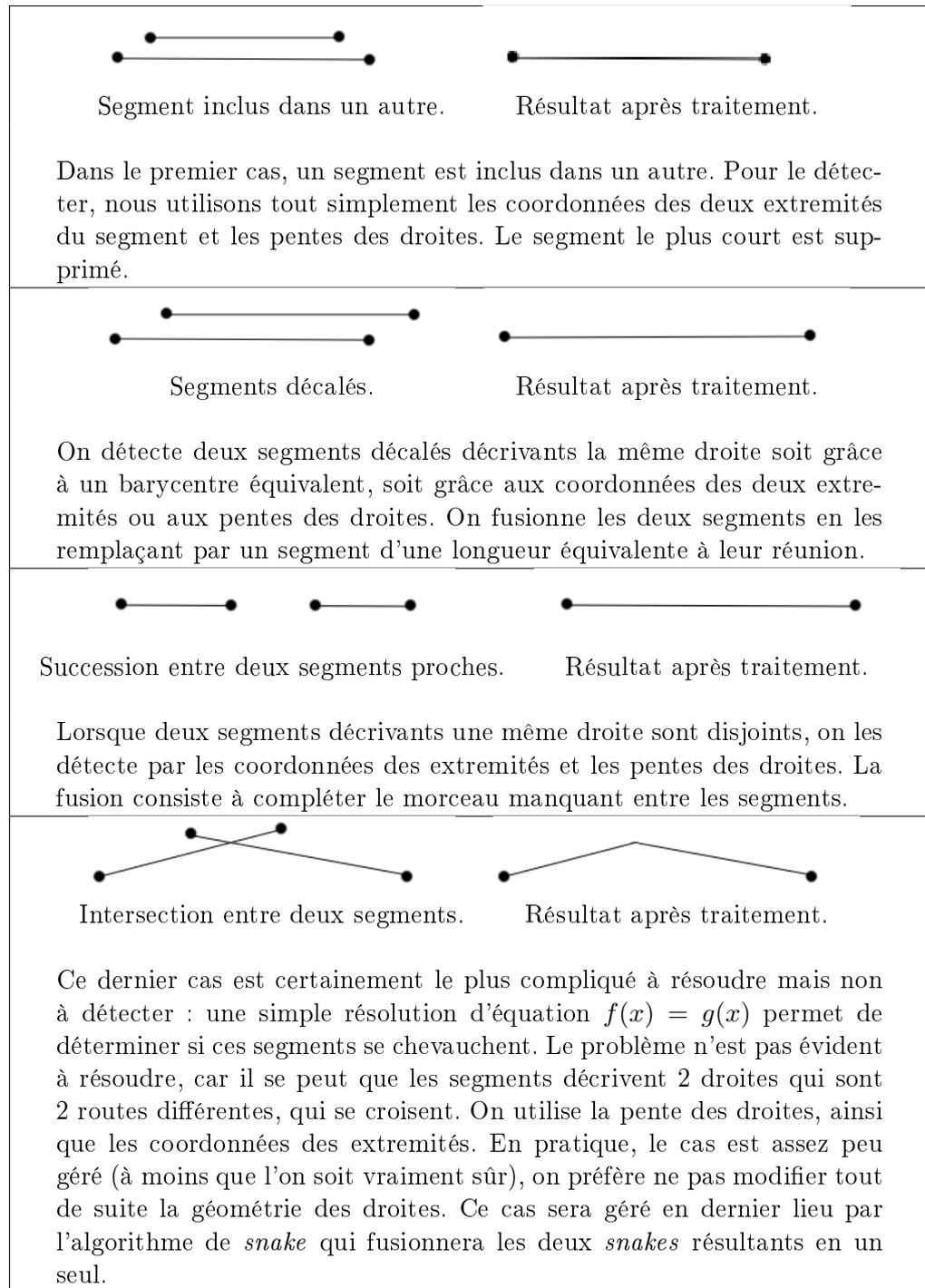


FIG. 4.23 – Règles de fusion des segments.

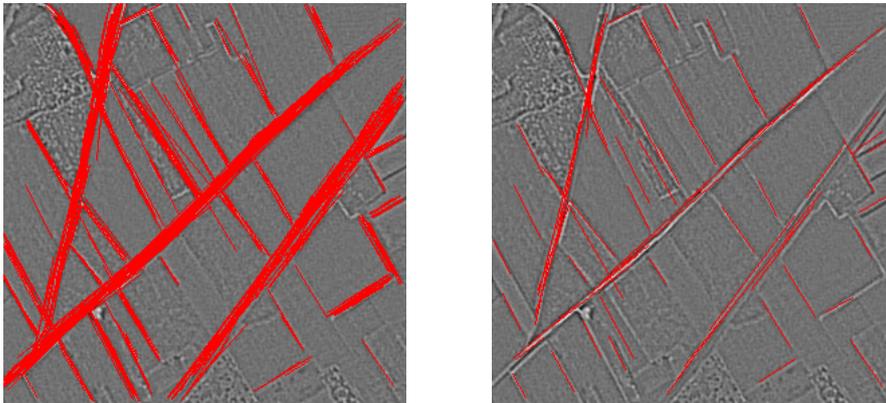


FIG. 4.24 – Exemple de résultat obtenu après application des règles de fusion des segments.

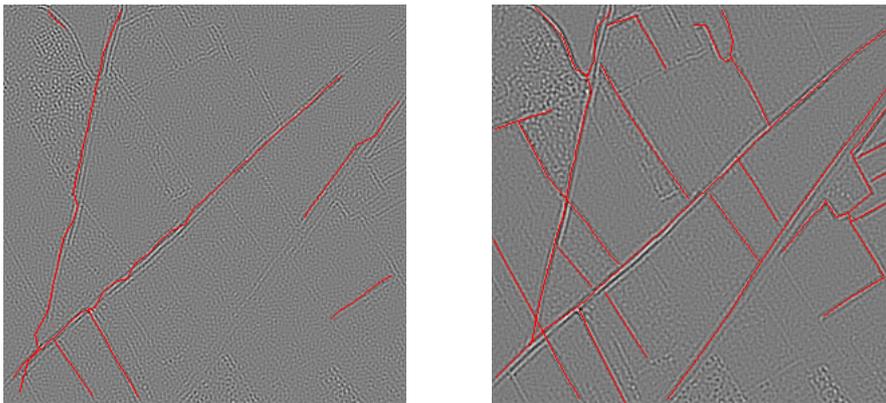


FIG. 4.25 – Réseau routier final détecté par courbes déformables.

suffisait de choisir le paramètre μ élevé (afin de réellement extraire les textures et pas seulement du bruit) et le paramètre λ faible (afin de ne pas trop «lisser» l'image et ainsi rejeter certains bords dans la partie texture qui seraient alors détectés par l'algorithme). Du côté de l'algorithme de détection des alignements, la valeur par défaut de 2 pour la norme minimum du gradient définissant une direction semble tout à fait appropriée. De manière à avoir une résolution suffisante en terme de directionalité, le choix de 192 orientations de recherche convient tout à fait. Reste le choix de la précision et de la valeur de ϵ . Concernant le choix de la précision, une valeur comprise entre 8 et 12 donne un bon compromis entre la détection d'alignements comportant des points s'écartant de la direction principale et une redondance de détection pour un même alignement. Quant à la valeur de ϵ , celle-ci condi-



FIG. 4.26 – Routes finales détectés pour la zone agricole

tionne le fait que l'on va vouloir privilégier des alignements plus ou moins longs. Plus ϵ est faible et plus les alignements détectés seront longs. Le choix de ce paramètre se fait donc essentiellement suivant le type d'image à traiter, toutefois les différentes expérimentations que nous avons menées montrent qu'il est raisonnable de choisir $10^{-15} \leq \epsilon \leq 10^{-6}$.

Si nous regardons les travaux disponibles dans la littérature, nous pouvons



FIG. 4.27 – Routes finales détectés pour la zone agricole et urbaine

constater que cette problématique est très largement étudiée. Les travaux récents de C.Lacoste [48], montrent que ce problème est loin d'être complètement résolu. Toutefois, nous pouvons noter que les techniques couramment utilisées se basent sur des approches probabilistes. Ces méthodes requièrent la plupart du temps une première phase de «prétraitement» visant à exacerber le tracé des routes dans les images. Dans sa thèse [48], l'auteur propose d'utiliser des processus ponctuels marqués (basés sur une modélisation Mar-

kovienne) pour effectuer la tâche de détection proprement dite. L'aspect intéressant est qu'il est nécessaire de modéliser des contraintes géométriques d'un point de vue probabiliste, chose qui n'est pas triviale. La dernière partie de cette thèse étend la méthode à une notion de lignes brisées afin de pouvoir détecter des réseaux sinueux. Nous pouvons donc voir que la démarche adoptée dans [48] est proche de la notre dans le sens où nous commençons aussi par une étape de prétraitement (l'extraction de la composante texture), une étape de détection d'objets linéiques (détecteur d'alignements de la Gestalt) et enfin une étape de «raffinement géométrique» par l'emploi de contours déformables. Malheureusement, n'ayant pas accès aux images utilisées par C.Lacoste, nous n'avons pu mener une réelle évaluation des résultats. Toutefois, pour des images de nature aériennes, nous pouvons constater visuellement que nos résultats sont très encourageant vis à vis de ceux obtenus par C.Lacoste. Par contre dans le cas d'image de type radar, notre méthode est mise en défaut, le problème étant que la modélisation sous-jacente à la décomposition de l'image ne correspond pas un modèle d'image radar.

Conclusion

Les problématiques intervenant sur les projets dans lesquels je suis impliqué à la DGA nous ont fait nous demander si il était possible de voir le traitement d'image de manière plus globale. La première étape a donc consisté à étudier une modélisation mathématique des images notamment à partir des travaux de Y.Meyer suivis de ceux de S.Osher et al. puis de ceux de J.F.Aujol et al. Ces travaux nous permettent de décomposer une image en sa partie «objets», modélisés par des fonctions appartenant à l'espace des fonctions à variations bornées (BV), sa partie textures, modélisées par l'espace dual de BV .

Dans le cadre de cette thèse, nous avons étendu cette modélisation au cas d'images bruitées. Ainsi une image peut être vue comme étant constituée de trois composantes : les objets, les textures et le bruit. Par ailleurs, J.F.Aujol et al. ont aussi proposé un modèle permettant d'obtenir ces trois composantes, modèle que nous avons par la suite modifié en remplaçant l'utilisation des ondelettes par les *contourlets*, celles-ci nous permettant d'inclure la notion de géométrie au modèle.

Ces différents modèles nous ont incité à proposer des méthodes permettant de juger de la qualité des décompositions obtenues par chacun des algorithmes. Cette démarche s'inscrit dans la problématique de «l'évaluation», actuellement en plein essor dans la communauté scientifique (notamment via le programme TECHNOVISION), importante car il est nécessaire de pouvoir juger de la qualité des résultats obtenus par les différentes méthodes développées.

Enfin nous présentons deux applications de cette décomposition d'image. La première, étudiée avant la disponibilité des modèles prenant en compte le bruit, dans le cadre du débruitage de bruit multiplicatif (appliqué à des images du type radar ou active laser). Nous comparons les résultats obtenus à quelques filtres classiques de la littérature.

La deuxième application présentée a été étudiée dans le cadre de la recherche d'outils d'aide pour les photo-interprètes. Nous cherchons à extraire les réseaux routiers d'images de nature aériennes ou satellitaires. Pour cela nous démontrons une propriété particulière de la décomposition et l'utilisons comme une première étape de «mise en forme» de l'image. Nous utilisons

ensuite l'algorithme de détection d'alignements de la Gestalt, proposée par l'équipe de J.M.Morel. Pour finir nous transformons les segments détectés en modèles déformables du type «snake». Nous proposons un algorithme très simple et surtout très rapide pour faire évoluer le *snake* en nous basant sur des mesures statistiques locales dans l'image.

Plusieurs perspectives s'offrent à ces travaux. Tout d'abord concernant la modélisation elle-même. Les modèles à trois composantes sont du type additif, hors dans le cas d'une image de type radar ou active laser nous pourrions être tentés de considérer les textures dans l'image et le bruit multiplicatif. Pour cela, il faudrait modifier le modèle de la façon suivante :

$$f = (u + v)w$$

Une autre extension du modèle serait la prise en compte d'un champ de mouvement dans une séquence. Celui serait «superposé» à l'image observée :

$$f(t) = u + v + m(t)$$

Ce type de modèle pourrait être utilisé par exemple dans le cas de l'imagerie active. En effet avec certains systèmes et niveaux de turbulences atmosphérique, il est possible de «voir» l'atmosphère se déplacer dans la séquence d'images. Il serait intéressant de tester ce modèle car la composante $u + v$ correspondrait à la scène observée sans l'effet des turbulences.

Ce type de techniques ayant retrouvé un regain d'intérêt ces dernières années, il reste à les utiliser dans des cadres se rapprochant d'autres techniques aujourd'hui classiques (modélisation markovienne, analyse de textures par ondelettes,...). Une première question venant rapidement à l'esprit quant à l'utilisation de ces espaces de textures : qu'en est-il de l'analyse et la classification des textures? En effet, étant donné qu'il est possible d'extraire les textures présentes dans l'image il est légitime de se demander si le fait d'avoir les textures seules apporte un plus. Des travaux ont été entamés afin d'explorer cette question. Plusieurs voies sont possibles

- utiliser des outils de caractérisation de textures classiques (filtrage de Gabor, ondelettes,... + classifieur (plus proches voisins, SVM,...)) sur la composante texture extraite,
- utiliser conjointement les parties objets et textures pour faire l'analyse (voir [4] pour un premier travail),
- développer de nouvelles méthodes de caractérisation des textures adaptées aux composantes extraites.

Enfin concernant le champ des applications de la décomposition, plusieurs domaines peuvent être explorés :

- développement d’algorithmes de compression adaptés à la composante texture (que l’on pourrait ensuite combiner avec des algorithmes plus «standards» fonctionnant sur les parties objets, une première approche a été explorée par J.F.Aujol et B.Matei [7]),
- extension au cas d’images multispectrales (J.F.Aujol propose une extension au cas des images couleurs dans [6]),
- Poursuivre les travaux sur l’extraction de réseaux routiers, notamment en adaptant la méthode au cas des images de type radar. Par ailleurs, il serait intéressant de pouvoir réellement comparer nos résultats avec ceux de la littérature en nous basant sur des images de test dont nous disposerions des vérités terrains associées (extraction manuelle par un photo-interprète)

En conclusion, ces techniques de décomposition sont encore jeunes et n’ont pas encore montré tout leur potentiel. Par ailleurs des applications pratiques commencent à apparaître ou feront très certainement leur apparition et promettent de passionnants travaux dans le futur.

Annexe A

Méthode de projection non-linéaire

Dans [20] Antonin Chambolle propose un algorithme basé sur une technique de projection non-linéaire pour résoudre une certaine catégorie de fonctionnelle à base de variation totale.

A.1 Notations - Définitions préliminaires

On suppose que l'on traite une image de taille $M \times N$. On notera $X = \mathbb{R}^{M \times N}$ et $Y = X \times X$.

Définition A.1.1 Soit $u \in X$ alors le gradient discret de u , noté $\nabla u \in Y = X \times X$ est défini par :

$$(\nabla u)_{i,j} = ((\nabla u)_{i,j}^1, (\nabla u)_{i,j}^2) \quad (\text{A.1})$$

avec $\forall i, j \in \llbracket 0, \dots, M-1 \rrbracket \times \llbracket 0, \dots, N-1 \rrbracket$

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{si } i < M-1 \\ 0 & \text{si } i = M-1 \end{cases} \quad (\text{A.2})$$

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{si } j < N-1 \\ 0 & \text{si } j = N-1 \end{cases} \quad (\text{A.3})$$

Définition A.1.2 Soit $p \in Y$ ($p = (p^1, p^2)$), on définit la divergence discrète $\text{div} : Y \rightarrow X$ telle que $\text{div} = -\nabla^*$ (∇^* est l'opérateur adjoint de ∇) par :

$$(\text{div } p)_{i,j} = \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{si } 0 < i < M-1 \\ p_{i,j}^1 & \text{si } i = 0 \\ -p_{i-1,j}^1 & \text{si } i = M-1 \end{cases} + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{si } 0 < j < N-1 \\ p_{i,j}^2 & \text{si } j = 0 \\ -p_{i,j-1}^2 & \text{si } j = N-1 \end{cases} \quad (\text{A.4})$$

On rappelle que : $\langle -\operatorname{div} p, u \rangle_X = \langle p, \nabla u \rangle_Y$

A.2 Variation totale

Dans le cas discret, la variation totale s'exprime par :

$$J(u) = \sum_{\substack{0 < i < M-1 \\ 0 < j < N-1}} |(\nabla u)_{i,j}| \quad (\text{A.5})$$

$$= \sum_{\substack{0 < i < M-1 \\ 0 < j < N-1}} \sqrt{\left((\nabla u)_{i,j}^1\right)^2 + \left((\nabla u)_{i,j}^2\right)^2} \quad (\text{A.6})$$

Or J est une fonction 1-homogène ($J(\lambda u) = \lambda J(u)$), si on applique la transformée de Legendre-Fenchel on obtient :

$$J^*(v) = \sup_u \langle u, v \rangle_X - J(u) \quad (\text{A.7})$$

avec

$$\langle u, v \rangle_X = \sum_{i,j} u_{i,j} v_{i,j} \quad (\text{A.8})$$

où J^* est la fonction caractéristique de l'ensemble convexe fermé K :

$$J^*(v) = \chi_K(v) = \begin{cases} 0 & \text{si } v \in K \\ +\infty & \text{sinon} \end{cases} \quad (\text{A.9})$$

Rq : on a la propriété $J^{**} = J$.

Dans le cas continu (voir les propriétés de l'espace BV), on a :

$$K = G_1 = \left\{ \operatorname{div} \xi : \xi \in C_c^1(\Omega, \mathbb{R}^2); |\xi(x)| \leq 1, \forall x \in \Omega \right\} \quad (\text{A.10})$$

alors

$$J(u) = \sup_{\xi} \left\{ \int_{\Omega} u(x) \operatorname{div} \xi(x) dx : \xi \in C_c^1(\Omega, \mathbb{R}^2); |\xi(x)| \leq 1, \forall x \in \Omega \right\} \quad (\text{A.11})$$

or $\int_{\Omega} u(x) \operatorname{div} \xi(x) dx = \langle u, \operatorname{div} \xi \rangle_X$ on peut réécrire :

$$J(u) = \sup_{\xi} \langle u, \operatorname{div} \xi \rangle_X \quad (\text{A.12})$$

ou encore si l'on pose $v = \operatorname{div} \xi$,

$$J(u) = \sup_{v \in K} \langle u, v \rangle_X \quad (\text{A.13})$$

On aimerait une expression identique dans le cas discret. Pour cela Antonin Chambolle a montré le lemme suivant :

Lemme A.2.1 Dans le cas discret, on a :

$$J(u) = \sup_{v \in G_1} \langle v, u \rangle \quad (\text{A.14})$$

$$\text{où } G_1 = \{\text{div } p; p \in Y; |p_{i,j}| \leq 1\} \quad (\text{A.15})$$

Définition A.2.2 On définit le produit scalaire sur Y : soient $p \in Y, q \in Y$ tel que $p = (p^1, p^2)$ et $q = (q^1, q^2)$ alors

$$\langle p, q \rangle_Y = \sum_{\substack{0 < i < M-1 \\ 0 < j < N-1}} (p_{i,j}^1 q_{i,j}^1 + p_{i,j}^2 q_{i,j}^2) \quad (\text{A.16})$$

A.3 Algorithme

On veut donc résoudre

$$\min_{u \in X} \frac{\|u - g\|^2}{2\lambda} + J(u) \quad (\text{A.17})$$

avec $g \in X, \lambda > 0, \|\cdot\|$ la norme euclidienne définie par $\|u\|^2 = \langle u, u \rangle_X$.

Si l'on applique Euler-Lagrange à A.17 on obtient

$$\frac{2(u - g)}{2\lambda} + \partial J(u) \ni 0 \quad (\text{A.18})$$

$$\iff u - g + \lambda \partial J(u) \ni 0 \quad (\text{A.19})$$

où ici ∂J est le pseudo-différentiel de J défini par

$$w \in \partial J(u) \iff J(v) \geq J(u) + \langle w, v - u \rangle_X \quad \forall v \quad (\text{A.20})$$

alors A.19 peut être réécrit comme

$$\frac{g - u}{\lambda} \in \partial J(u) \quad (\text{A.21})$$

$$\iff \partial J^* \left(\frac{g - u}{\lambda} \right) \ni u \quad (\text{A.22})$$

$$\iff \frac{u}{\lambda} \in \frac{1}{\lambda} \partial J^* \left(\frac{g - u}{\lambda} \right) \quad (\text{A.23})$$

$$\iff \frac{g}{\lambda} \in \frac{g - u}{\lambda} + \frac{1}{\lambda} \partial J^* \left(\frac{g - u}{\lambda} \right) \quad (\text{A.24})$$

Supposons que l'on cherche un minimiseur de

$$\frac{\|w - (\frac{g}{\lambda})\|^2}{2} + \frac{1}{\lambda} J^*(w) \quad (\text{A.25})$$

on applique Euler-Lagrange à A.25, on obtient alors

$$w - \frac{g}{\lambda} + \frac{1}{\lambda} \partial J^*(w) \ni 0 \quad (\text{A.26})$$

$$\iff w + \frac{1}{\lambda} \partial J^*(w) \ni \frac{g}{\lambda} \quad (\text{A.27})$$

On voit donc grâce à A.24 que

$$w = \frac{g - u}{\lambda} \quad (\text{A.28})$$

est un minimiseur de A.25

Or comme $J^*(w) = \chi_{G_1}(w)$ et si $w = P_{G_1}(\frac{g}{\lambda})$ (l'opérateur de projection sur G_1) alors $J^*(w) = 0$ et $\|w - \frac{g}{\lambda}\|$ est minimum. Donc

$$P_{G_1}\left(\frac{g}{\lambda}\right) = \frac{g - u}{\lambda} \quad (\text{A.29})$$

$$u = g - \lambda P_{G_1}\left(\frac{g}{\lambda}\right) \quad (\text{A.30})$$

On note $P_{G_\lambda}(\frac{g}{\lambda}) = \lambda P_{G_1}(\frac{g}{\lambda})$, on a alors

$$u = g - P_{G_\lambda}\left(\frac{g}{\lambda}\right) \quad (\text{A.31})$$

Il reste donc à trouver le moyen de calculer $P_{G_\lambda}(g)$. A.Chambolle donne le résultat suivant :

$$\text{calculer } P_{G_\lambda}(g) \iff \min_{p \in Y} \{ \|\lambda \operatorname{div}(p) - g\|^2; |p_{i,j}|^2 \leq 1 \quad \forall i, j \} \quad (\text{A.32})$$

Les conditions de Karush-Kuhn-Tucker montrent l'existence d'un multiplicateur de Lagrange $\alpha_{i,j} \geq 0$ associé à chaque contrainte de A.32 tel que l'on ait $\forall i, j$:

$$-(\nabla(\lambda \operatorname{div}(p) - g))_{i,j} + \alpha_{i,j} p_{i,j} = 0 \quad (\text{A.33})$$

avec

$$\alpha_{i,j} > 0 \quad \text{et} \quad |p_{i,j}| = 1 \quad (\text{A.34})$$

$$\alpha_{i,j} = 0 \quad \text{et} \quad |p_{i,j}| < 1. \quad (\text{A.35})$$

On voit alors que si $\alpha_{i,j} = 0$, alors $(\nabla(\lambda \operatorname{div}(p) - g))_{i,j} = 0$; donc ce cas n'est pas intéressant. Donc passons au cas $\alpha_{i,j} \neq 0$:

$$\alpha_{i,j} p_{i,j} = (\nabla(\operatorname{div}(p) - g))_{i,j} \quad (\text{A.36})$$

$$\Rightarrow |\alpha_{i,j}| |p_{i,j}| = \left| (\nabla(\operatorname{div}(p) - g))_{i,j} \right| \quad (\text{A.37})$$

or $|\alpha_{i,j}| = \alpha_{i,j}$ car $\alpha_{i,j} > 0$ et $|p_{i,j}| = 1$ donc

$$\alpha_{i,j} = \left| (\nabla (\operatorname{div}(p) - g))_{i,j} \right| \quad (\text{A.38})$$

On utilise alors un algorithme de descente du gradient avec $\tau > 0$; $p^0 = 0$; $n \geq 0$

$$p_{i,j}^{n+1} = p_{i,j}^n + \tau \left[\left(\nabla \left(\operatorname{div}(p^n) - \frac{g}{\lambda} \right) \right)_{i,j} - \left| \left(\nabla \left(\operatorname{div}(p^n) - \frac{g}{\lambda} \right) \right)_{i,j} \right| p_{i,j}^{n+1} \right] \quad (\text{A.39})$$

On obtient donc l'équation itérative

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau \left(\nabla \left(\operatorname{div}(p^n) - \frac{g}{\lambda} \right) \right)_{i,j}}{1 + \tau \left| \left(\nabla \left(\operatorname{div}(p^n) - \frac{g}{\lambda} \right) \right)_{i,j} \right|} \quad (\text{A.40})$$

Antonin Chambolle démontre le théorème important suivant

Théorème A.3.1 *Si $\tau < \frac{1}{8}$ alors $\lambda \operatorname{div}(p^n)$ converge vers $P_{G_\lambda}(g)$ quand $n \rightarrow +\infty$*

La démonstration de ce théorème est disponible dans [20]. En pratique, on constate que le choix $n = 20$ est suffisant à obtenir la convergence souhaitée.

Annexe B

Snake probabiliste

Le problème de la segmentation a engendré des efforts importants pour la mise au point de techniques adéquates. Les algorithmes de type modèles déformables ont connu un développement très important ces dernières années car elles ont l'avantage de fournir directement un ensemble de courbes correspondant aux contours détectés.

Après avoir rappelé les diverses approches disponibles dans la littérature, nous introduisons un modèle généralisant celui de l'équipe de P.Réfrégier [24], [23], [38], [66], basé sur un principe de mesure statistique. Cet algorithme a été développé dans le cadre d'un besoin pour la DGA dont le but était d'extraire certaines formes dans de longues séquences d'images. Cela nous a obligé à garder à l'esprit la contrainte du temps de calcul et nous a permis de développer un modèle relativement simple et fonctionnant en temps réel.

B.1 Les premiers modèles

Les premiers à avoir proposé la notion de *snake* sont M.Kass et al. [47]. Celle-ci à base d'une courbe polygonale C , paramétrée par $v(s)$ où s représente l'abscisse curviligne, peut se mouvoir grâce à une «force» appliquée aux nœuds qui relient les segments la composant (voir la figure B.1).

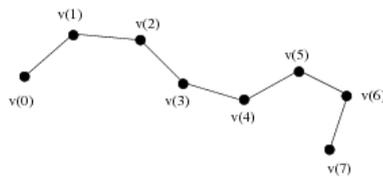


FIG. B.1 – Définition d'un *snake* polygonal

Le modèle est guidé par

$$E(C) = \int_a^b \left(\alpha(s) \left\| \frac{\partial v}{\partial s}(s) \right\|_{L^2}^2 + \beta(s) \left\| \frac{\partial^2 v}{\partial s^2}(s) \right\|_{L^2}^2 - \|\nabla I(v(s))\|_{L^2}^2 \right) ds \quad (\text{B.1})$$

Dans la pratique, $\alpha(s)$ et $\beta(s)$ sont pris constants. En appliquant le principe d'Euler, on obtient le système d'équations suivant :

$$-\alpha \frac{\partial^2 v}{\partial s^2}(s) + \beta \frac{\partial^4 v}{\partial s^4}(s) = \nabla P(v(s)) \quad (\text{B.2})$$

où

$$P(v(s)) = f(v(s)) = \frac{1}{2} \|\nabla I(v(s))\|_{L^2}^2 \quad (\text{B.3})$$

représente une force d'attraction due à l'image (divers choix sont possibles pour guider le *snake* : force ballon [25], *Gradient Vecteur Flow* [80],...).

Dans [10], l'auteur montre que la solution numérique à ce système peut se mettre sous une forme matricielle donnant ainsi un algorithme itératif. Il est par ailleurs aussi possible de remplacer les courbes polygonales par des courbes *B-Splines* [74],[75] pour obtenir des modèles nommés *B-Snakes* [13],[68],[12]. Dans ce modèle, la force est appliquée sur les points de contrôle de la courbe *B-Splines*. Il est aussi possible d'obtenir un algorithme itératif convergent vers la position finale de la courbe.

B.2 Formulation par *levelset*

Les modèles précédents ont deux inconvénients majeurs. D'une part, la position de la courbe initiale doit être proche du résultat final. D'autre part si plusieurs objets sont présents dans l'image, il faut mettre en place autant de courbes que d'objets recherchés et appliquer à chacun l'algorithme d'évolution (pas de changement topologiques possibles). Pour ces raisons, plusieurs travaux se sont attachés à reformuler le problème des contours déformables et se sont dirigés vers des méthodes à base de *levelsets*.

B.2.1 La méthode des *levelsets*

Cette méthode a été introduite pour résoudre des équations du type Hamilton-Jacobi

$$\frac{\partial \phi}{\partial t} + [H(\phi)]_x = 0 \quad (\text{B.4})$$

où H est une fonction qui dépend des dérivées partielles de ϕ . L'idée est alors d'utiliser un schéma itératif pour résoudre cette équation.

Ce type d'équation intervient notamment dans la problématique de l'évolution de courbes. En effet, soit une courbe $C(p)$ régie par une équation du type :

$$\begin{cases} \frac{\partial C(p,t)}{\partial t} = F(\kappa)\mathcal{N} \\ C(p,0) = C_0(p) \end{cases} \quad (\text{B.5})$$

où κ représente la courbure et \mathcal{N} la normale à la courbe.

Si l'on résout ce système par la méthode des différences finies, on exclut les changements topologiques, sans compter la possibilité de voir apparaître des instabilités numériques dans certains cas. On utilise donc la méthode des *levelsets* :

Soit $\varphi(x, y, t) : \mathbb{R}^2 \times [O, T] \rightarrow \mathbb{R}$ la fonction dite *levelset*. On pose alors :

$$\begin{cases} C(p,0) = \{(x,y)/\varphi(x,y,0) = 0\} \\ C(p,t) = \{(x,y)/\varphi(x,y,t) = 0\} \end{cases} \quad (\text{B.6})$$

Ceci est équivalent à dire que $C(p,t) = \varphi^{-1}(0)$. Concrètement à un instant donné t , on a une surface définie par $(x,y,z = \varphi(x,y,t))$ dans \mathbb{R}^3 avec laquelle on prend l'intersection avec le plan $z = 0$ pour retrouver $C(p,t)$ (Cf. Fig.B.2).

Il faut maintenant remplacer $C(p,t)$ par $\varphi(p,t)$ dans B.5, pour cela on écrit :

$$\varphi(C(t), t) = 0 \quad (\text{B.7})$$

$$\Rightarrow \frac{\partial \varphi}{\partial C(t)} \frac{\partial C(t)}{\partial t} + \frac{\partial \varphi}{\partial t} = 0 \quad (\text{B.8})$$

or on a :

$$\frac{\partial C(t)}{\partial t} = F(\kappa)\mathcal{N} \quad (\text{B.9})$$

et

$$\frac{\partial \varphi}{\partial C(t)} = \nabla \varphi \quad (\text{B.10})$$

et l'on pose la notation :

$$\frac{\partial \varphi}{\partial t} = \varphi_t \quad (\text{B.11})$$

où \mathcal{N} est le vecteur normal à la courbe mais aussi à la surface φ

$$\Rightarrow \mathcal{N} = -\frac{\nabla \varphi}{|\nabla \varphi|} \quad (\text{B.12})$$

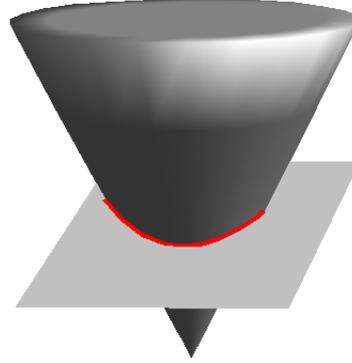


FIG. B.2 – Principe de coupe d'une fonction *levelset* par le plan image.

Donc

$$\varphi_t = -\langle \nabla \varphi, F(\kappa) \mathcal{N} \rangle = -F(\kappa) \langle \nabla \varphi, -\frac{\nabla \varphi}{|\nabla \varphi|} \rangle \quad (\text{B.13})$$

$$\Rightarrow \varphi_t = F(\kappa) |\nabla \varphi| \quad (\text{B.14})$$

On obtient bien une équation de type Hamilton-Jacobi. L'évolution de la courbe consiste donc à faire évoluer la surface φ et à en prendre son intersection avec le plan $z = 0$. En pratique, il n'est pas nécessaire de faire évoluer la surface entière, il suffit de s'intéresser au voisinage du plan de coupe. Divers algorithmes ont été proposés afin de réaliser cette tâche (*Fast Marching*, *Narrow Band* . . .) ; voir les références suivantes : [19], [42], [61], [59], [62], [60], [57], [69].

Pour utiliser cette méthode, il reste à formuler le problème des contours actifs sous forme d'une équation de Hamilton-Jacobi.

B.2.2 Reformulation des *snakes*

Nous prenons comme point de départ la formulation classique B.1 (on considère que le paramétrage de la courbe est normalisé) :

$$E(C) = \int_0^1 \left(\alpha(p) \left\| \frac{\partial C}{\partial p}(p) \right\|_{L^2}^2 + \beta(p) \left\| \frac{\partial^2 C}{\partial p^2}(p) \right\|_{L^2}^2 - \|\nabla I(C(p))\|_{L^2}^2 \right) dp \quad (\text{B.15})$$

Or il est possible de montrer que le terme $\left\| \frac{\partial^2 C}{\partial p^2}(p) \right\|_{L^2}^2$ correspond en fait à une reparamétrisation de la courbe, il n'est donc pas utile au bon fonctionnement du modèle. Nous partons donc de :

$$E(C) = \alpha \int_0^1 \left\| \frac{\partial C}{\partial p}(p) \right\|_{L^2}^2 dp - \beta \int_0^1 \|\nabla I(C(p))\|_{L^2}^2 dp \quad (\text{B.16})$$

On peut même généraliser ce modèle en considérant la fonction $g(x)$ monotone décroissante :

$$g : [0; +\infty[\rightarrow \mathbb{R}^+ \quad \text{telle que : } g(0) = 1 \text{ et } \lim_{x \rightarrow +\infty} g(x) = 0 \quad (\text{B.17})$$

et en posant $\beta = 1 - \alpha$, on obtient :

$$E(C) = \alpha \int_0^1 \left\| \frac{\partial C}{\partial p}(p) \right\|_{L^2}^2 dp - (1 - \alpha) \int_0^1 g(\|\nabla I(C(p))\|_{L^2})^2 dp \quad (\text{B.18})$$

Minimiser cette fonctionnelle d'énergie est équivalent à minimiser :

$$E(C) = \int_0^{\mathcal{L}(C)} g(\|\nabla I(C(s))\|_{L^2}) ds \quad (\text{B.19})$$

$$= \int_0^1 g(\|\nabla I(C(p))\|_{L^2}) \left\| \frac{\partial C}{\partial p} \right\|_{L^2} dp \quad (\text{B.20})$$

où le terme $g(\|\nabla I(C(p))\|_{L^2})$ correspond au terme d'attraction dû à l'image, le terme $\left\| \frac{\partial C}{\partial p} \right\|_{L^2}$ correspond au terme de régularité de la courbe et $\mathcal{L}(C)$ est la longueur de la courbe.

En appliquant le principe d'Euler-Lagrange à B.20 on obtient l'équation aux dérivées partielles :

$$\frac{\partial C}{\partial t} = g(\|\nabla I\|_{L^2}) \kappa \mathcal{N} - (\nabla g(\|\nabla I\|_{L^2}) \cdot \mathcal{N}) \mathcal{N} \quad (\text{B.21})$$

où $\kappa = \text{div} \left(\frac{\nabla I}{\|\nabla I\|_{L^2}} \right)$ (courbure euclidienne). Classiquement, on choisit :

$$g(\|\nabla I\|_{L^2}) = \frac{1}{1 + \|\nabla \hat{I}\|_{L^2}^q} \quad (\text{B.22})$$

où $q \in \{1; 2\}$ et \hat{I} est l'image I filtrée par une gaussienne. Si l'on pose alors

$$F(\kappa) = g(\|\nabla I\|_{L^2}) \kappa - (\nabla g(\|\nabla I\|_{L^2}) \cdot \mathcal{N}) \quad (\text{B.23})$$

on obtient

$$\frac{\partial C}{\partial t} = F(\kappa) \mathcal{N} \quad (\text{B.24})$$

qui est bien une équation d'évolution de courbe identique à B.5. On peut donc utiliser la méthode des *levelsets* pour résoudre cette EDP et le problème (en prenant la formulation des *levelsets*) devient alors identique à :

$$\frac{\partial \varphi}{\partial t} = F(\kappa) |\nabla \varphi| \quad (\text{B.25})$$

Il reste à prendre l'intersection avec le plan $z = 0$ à chaque itération pour obtenir l'évolution de la courbe.

B.3 Contour Actif Statistique Polygonal (CASP)

L'équipe de P.Réfrégier propose de prendre le point de vue de l'inférence statistique pour reformuler le problème des contours déformables. Dans les pages qui suivent, nous nous contenterons (pour la commodité du lecteur) de paraphraser P.Réfrégier et al. Les auteurs considèrent le *snake* comme étant un paramètre statistique θ à estimer à partir de l'information mesurée disponible χ (l'image).

Pour cela, ils prennent en compte le modèle physique de formation de l'image ; on a donc une fonction de vraisemblance : $P(\chi | \theta)$.

Le *snake* final est la solution optimale au sens du maximum de vraisemblance (MV). On peut de plus rajouter de l'information a priori sur θ (*prior* $\prod(\theta)$), la solution s'obtenant alors au sens du maximum a priori (MAP).

En pratique $P(\chi | \theta)$ n'est pas toujours disponible, on essaie alors de construire une vraisemblance $P_\mu(\chi | \theta)$ dont les paramètres μ (paramètres de nuisance) seront estimés par le biais de la théorie de l'information par MV ou MAP en utilisant seulement χ . On pourra aussi estimer μ par une approche Bayésienne Marginale (BM) où l'on considère les paramètres μ comme des variables aléatoires distribuées suivant une loi de probabilité ; on intègre alors sur toutes les valeurs possibles des paramètres de nuisance. Les approches par MAP ou BM permettent d'inclure de l'information a priori sur μ .

On considère le modèle d'image suivant : soit une scène $s = \{s_i | i \in [1, N]\}$ de N pixels, on fait l'hypothèse qu'une scène est composée de deux régions : l'objet à segmenter et le fond. La scène s peut donc être représentée par

$$s_i = a_i w_i + b_i (1 - w_i) \quad (\text{B.26})$$

où \mathbf{w} est un vecteur de dimension N , support de l'objet :

$$w_i = \begin{cases} 1 & \text{si } s_i \in \text{objet} \\ 0 & \text{sinon} \end{cases} \quad (\text{B.27})$$

On obtient donc deux régions :

$$\begin{aligned} \text{objet : } \Omega_a &= \{i | w_i = 1\} && \text{composée de } N_a(\mathbf{w}) \text{ pixels} \\ \text{fond : } \Omega_b &= \{i | w_i = 0\} && \text{composée de } N_b(\mathbf{w}) \text{ pixels} \end{aligned} \quad (\text{B.28})$$

Les vecteurs \mathbf{a} et \mathbf{b} sont constitués de variables aléatoires (de probabilités $p^{\mu_a}(a_i)$ et $p^{\mu_b}(b_i)$) représentant respectivement les niveaux de gris de l'objet et du fond.

Les auteurs font alors l'hypothèse du modèle SIR (*Statistically Independent Region*) qui consiste à prendre p^{μ_a} et p^{μ_b} indépendantes.

Le but de la segmentation est donc de trouver la forme binaire \mathbf{w} . On voit apparaître deux cas :

1. On n'a pas d'information a priori sur $\mathbf{w} \Rightarrow$ MV.
2. On a de l'information a priori sur $\mathbf{w} \Rightarrow$ MAP ou BM.

B.3.1 Modèle

Soit $H_{\mathbf{w}}$ l'hypothèse qui consiste à attribuer une certaine forme à l'objet ; la vraisemblance s'écrit alors :

$$L[s | H_{\mathbf{w}}, \mu_a, \mu_b] = P(\chi_a | \mu_a)P(\chi_b | \mu_b) \quad (\text{B.29})$$

où

$$P(\chi_u | \mu_u) = \prod_{i \in \Omega_u} p^{\mu_u}(s_i) \quad \text{avec} \quad \chi_u = \{s_i | i \in \Omega_u\} \quad (\text{B.30})$$

Faisons l'hypothèse que p^{μ_a} et p^{μ_b} appartiennent à la famille exponentielle (voir [23]) ; on peut donc écrire :

$$P(\chi_u | \mu_u) = g(T(\chi_u) | \mu_u) \prod_{i \in \Omega_u} f(s_i) \quad (\text{B.31})$$

$$\Rightarrow L[s | H_{\mathbf{w}}, \mu_a, \mu_b] = g(T(\chi_a) | \mu_a)g(T(\chi_b) | \mu_b) \prod_{i \in \Omega} f(s_i) \quad (\text{B.32})$$

En prenant la log-vraisemblance on obtient

$$l(s | H_{\mathbf{w}}) = \ln G(T(\chi_a)) + \ln G(T(\chi_b)) + \sum_{i \in \Omega} \ln f(s_i) \quad (\text{B.33})$$

Rq : on a supposé ici que l'on avait le même a priori pour μ_a et μ_b .

D'autre part $\sum_{i \in \Omega} \ln f(s_i)$ ne dépend pas de \mathbf{w} , on ne s'en occupera plus par la suite.

Maximiser B.33 revient donc à minimiser

$$J(s | \mathbf{w}) = -\ln G(T(\chi_a)) - \ln G(T(\chi_b)) \quad (\text{B.34})$$

Il faut donc trouver \mathbf{w} (et donc $T(\chi_a)$ et $T(\chi_b)$) qui minimise $J(s | \mathbf{w})$

Rq : par abus de langage J pourrait être appelé énergie externe du *snake*.

B.3.2 Cas gaussien

On considère donc que

$$p^{\mu_u}(x) = \frac{1}{\sqrt{2\pi}\sigma_u} \exp\left\{\frac{-1}{2\sigma_u^2}(x - m_u)^2\right\} \quad (\text{B.35})$$

Les vecteurs des paramètres de nuisance sont donc

$$\mu_a = (m_a, \sigma_a) \quad (\text{B.36})$$

$$\mu_b = (m_b, \sigma_b) \quad (\text{B.37})$$

On obtient alors

$$\begin{aligned} l(s, \mathbf{w}, m_a, \sigma_a, m_b, \sigma_b) = & -N_a(\mathbf{w}) \ln(\sqrt{2\pi}\sigma_a) - \frac{1}{2\sigma_a^2} \sum_{i \in \Omega_a} (s_i - m_a)^2 \\ & -N_b(\mathbf{w}) \ln(\sqrt{2\pi}\sigma_b) - \frac{1}{2\sigma_b^2} \sum_{i \in \Omega_b} (s_i - m_b)^2 \end{aligned} \quad (\text{B.38})$$

Il faut maintenant commencer à estimer les paramètres de nuisance par MV, donc en appliquant

$$\frac{\partial l(s, \mathbf{w}, m_a, \sigma_a, m_b, \sigma_b)}{\partial m_u} = 0 \quad (\text{B.39})$$

$$\frac{\partial l(s, \mathbf{w}, m_a, \sigma_a, m_b, \sigma_b)}{\partial \sigma_u} = 0 \quad (\text{B.40})$$

on obtient

$$\hat{m}_a = \frac{1}{N_a(\mathbf{w})} \sum_{i \in \Omega_a} s_i \quad (\text{B.41})$$

$$\hat{m}_b = \frac{1}{N_b(\mathbf{w})} \sum_{i \in \Omega_b} s_i \quad (\text{B.42})$$

$$\hat{\sigma}_a^2 = \frac{1}{N_a(\mathbf{w})} \sum_{i \in \Omega_a} (s_i - \hat{m}_a)^2 \quad (\text{B.43})$$

$$\hat{\sigma}_b^2 = \frac{1}{N_b(\mathbf{w})} \sum_{i \in \Omega_b} (s_i - \hat{m}_b)^2 \quad (\text{B.44})$$

En utilisant ces résultats on obtient

$$l(s, \mathbf{w}, m_a, \sigma_a, m_b, \sigma_b) = -\frac{N}{2} \ln(2\pi) - \frac{N}{2} - N_a(\mathbf{w}) \ln[\hat{\sigma}_a^2(\mathbf{w})] - N_b(\mathbf{w}) \ln[\hat{\sigma}_b^2(\mathbf{w})] \quad (\text{B.45})$$

Donc maximiser B.45 par rapport à \mathbf{w} revient à minimiser

$$J(s, \mathbf{w}) = N_a(\mathbf{w}) \ln[\hat{\sigma}_a^2(\mathbf{w})] + N_b(\mathbf{w}) \ln[\hat{\sigma}_b^2(\mathbf{w})] \quad (\text{B.46})$$

B.3.3 Cas général pour la famille exponentielle

On peut montrer, en tenant un raisonnement similaire, que trouver une solution au problème dans le cas général de la famille exponentielle, revient à minimiser

$$J(s, \mathbf{w}) = N_a(\mathbf{w})H(\hat{\theta}_a) + N_b(\mathbf{w})H(\hat{\theta}_b) + K_l \quad (\text{B.47})$$

avec

loi	$H(x)$	$\hat{\theta}_u$
Gaussien	$\log(x)$	$\frac{1}{N_u(w)} \sum_{i \in \Omega_u} s_i^2$ $-\left\{ \frac{1}{N_u(w)} \sum_{i \in \Omega_u} s_i \right\}^2$
Gamma	$\log(x)$	$\frac{1}{N_u(w)} \sum_{i \in \Omega_u} s_i$
Rayleigh	$\log(x)$	$\frac{1}{N_u(w)} \sum_{i \in \Omega_u} s_i^2$
Bernouilli	$x \log(x) + (1-x) \log(1-x)$	$\frac{1}{N_u(w)} \sum_{i \in \Omega_u} s_i$
Poisson	$-x \log(x)$	$\frac{1}{N_u(w)} \sum_{i \in \Omega_u} s_i$

B.3.4 Algorithmes d'optimisation

L'équipe de Marseille a utilisé deux types d'algorithme :

- un algorithme déterministe.
- un algorithme stochastique.

Le premier fait tous les tests de déplacement de chaque nœud pour trouver le minimum du critère, l'inconvénient majeur est le temps de calcul qui peut devenir prohibitif. Le deuxième algorithme fait un tir aléatoire (uniformément) du nœud à traiter ainsi qu'un tir aléatoire du déplacement associé. On améliore très nettement la rapidité de la convergence mais on n'est pas sûr d'obtenir le minimum global.

B.4 Snake probabiliste

La méthode précédente est très bien adaptée à des images fortement bruitées ou dégradées de part le fait qu'elle utilise des mesures statistiques. Dans le cadre d'une étude pour la DGA, nous avons été amenés à traiter des images ayant une qualité médiocre ce qui nous a incité à conserver ce principe de mesure statistique. Par ailleurs, le but était de pouvoir traiter une grande quantité d'images ce qui imposa des contraintes de temps de calcul. Enfin, nous souhaitions pouvoir disposer d'un modèle pouvant fonctionner aussi bien avec des courbes fermées que ouvertes (ce qui n'est pas le cas du modèle CASP, celui-ci ne fonctionnant que dans le cadre de courbes fermées).

Dans [41] nous nous proposons de reformuler le problème d'un *snake* statistique mais dans un cadre plus général, c'est à dire qui tiendrait compte aussi bien du cas «courbe fermée» que du cas «courbe ouverte».

Posons les notations suivantes :

- $C(s)$ le contour actif
- B un contour dans l'image
- $p(C|B)$ la densité de probabilité que la position actuelle du contour actif soit sur un contour dans l'image
- $p(B|C)$ la densité de probabilité qu'un contour soit présent à la position actuelle du *snake*

Alors en utilisant la règle de Bayes on a :

$$p(C|B) = \frac{p(B|C)p(C)}{p(B)} \quad (\text{B.48})$$

où $p(B)$ représente la probabilité des contours dans l'image, quantité qui n'est pas importante car constante dans notre problème. La quantité $p(C)$ représente la probabilité du contour actif ; on peut voir cette quantité comme la probabilité que le contour actif soit de telle ou telle forme (c'est grâce à cette quantité que l'on pourra notamment jouer sur le *snake* pour obtenir une régularisation du contour actif).

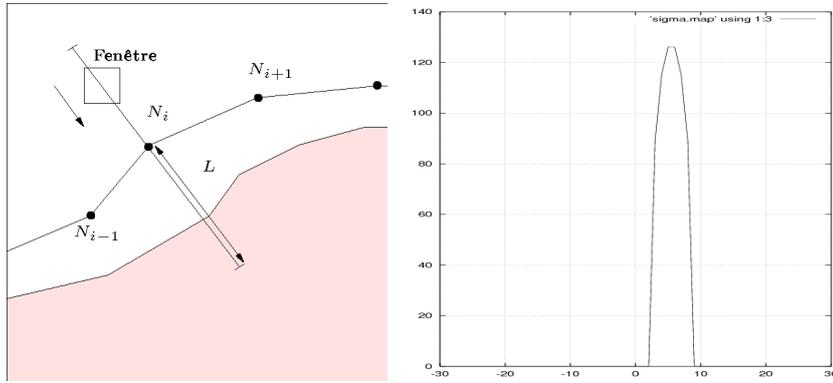
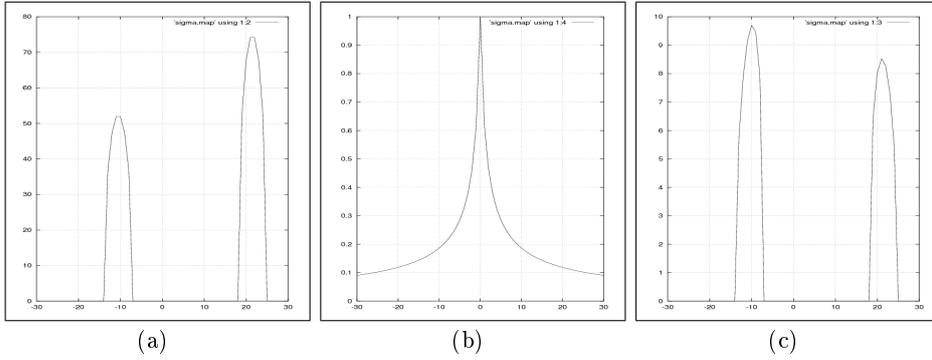
Il faut maintenant définir la quantité $p(B|C)$. Un exemple simple de choix est venu d'une constatation simple : un contour dans une image est une zone qui n'est pas homogène, par conséquent la variance calculée sur une fenêtre de taille raisonnable aux alentours d'un contour doit nous donner une valeur importante.

Nous construisons $p(B|C)$ en considérant, nœud par nœud, une mesure de variance sur une fenêtre centrée sur le nœud courant (à une constante de renormalisation près). Ce qui revient à considérer que chaque nœud est indépendant (du point de vue de l'influence de l'image sur le *snake*) ; cela revient donc à écrire :

$$p(B|C) = \prod_{i \in [0 \dots N-1]} p(N_i|I) \quad (\text{B.49})$$

où $p(N_i|I)$ sera par exemple construite à partir des mesures de variance le long de la normale à la courbe au nœud N_i (voir la fig.B.3) car uniquement le déplacement suivant la normale à la courbe est importante (la composante tangentielle correspondant en fait à un reparamétrage de la courbe).

Concernant la régularisation du contour actif, on peut utiliser une formula-


 FIG. B.3 – Principe d'évolution dans le cadre du *snake* statistique.

 FIG. B.4 – Exemple de régularisation : (a) $p(N_i|I)$, (b) la fonction de pondération, (c) $p(N_i|I)$ régularisée.

tion standard en inférence statistique :

$$P(\mathbf{w}) = A \exp\left\{-\frac{1}{2\varphi^2} U_{in}(\mathbf{w})\right\} \quad (\text{B.50})$$

$$\text{avec } U_{in}(\mathbf{w}) = \sum_{i=0}^{N_{\text{nœuds}}-1} d_i^2 \quad (\text{B.51})$$

ou l'on peut aussi jouer directement sur la construction de $p(N_i|I)$ en utilisant une fonction de pondération associée au critère de régularisation que l'on souhaite mettre en place. La figure B.4 donne un exemple de régularisation de ce type dans le cas où l'on souhaite privilégier les contours les plus proches de la position initiale du nœud.

La fonction régularisante utilisée peut être de la forme :

$$R(x) = \frac{1}{|x - L|r} \quad (\text{B.52})$$

où x représente l'abscisse le long de la normale de longueur $2L$, $x = 0$ correspondant à la position initiale du nœud, r correspondant au paramètre de régularisation.

Afin d'obtenir une meilleure convergence, on fait une première itération avec un petit nombre de points pour le *snake*. On se fixe une longueur L convenable pour que le contour actif s'accroche sur le contour désiré. Ensuite on rééchantillonne la courbe ; on finit par une nouvelle étape de convergence avec une longueur L plus restreinte afin d'obtenir un résultat ayant une bonne résolution. Les figures B.5 et B.6 donnent les résultats obtenus dans les cas de courbes aussi bien fermées que ouvertes.

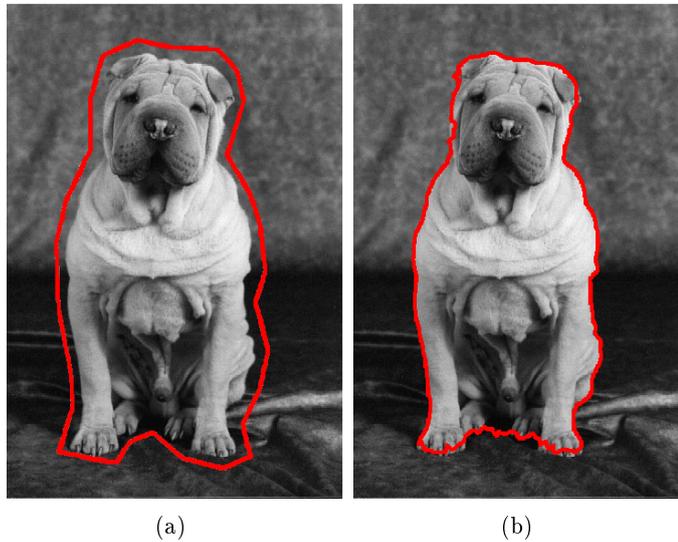
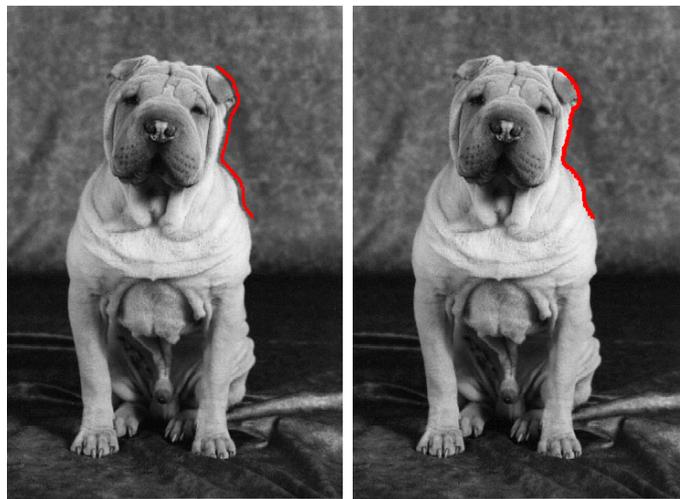


FIG. B.5 – *Snake* statistique dans le cas fermé : initialisation (a), résultat (b)

Ce modèle donne de très bon résultats malgré sa simplicité. Il a par ailleurs le très gros avantage de nécessiter très peu de calculs et par conséquent d'être très rapide (quelques micro-secondes sur les expérimentations que nous avons menées). Son inconvénient, le même que pour tous les modèles utilisant des courbes polygonales, est qu'il ne gère pas les changements topologiques (il faudra autant de courbes que d'objets recherchés).



(a)

(b)

FIG. B.6 – *Snake* statistique dans le cas ouvert : initialisation (a), résultat (b)

Bibliographie

- [1] J.F. Aujol. Contribution à l'analyse de textures en traitement d'images par méthodes variationnelles et équations aux dérivées partielles. Thèse de Doctorat, Université de Nice Sophia Antipolis, 2004.
- [2] J.F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Decomposing an image. application to textured images and SAR images. Rapport technique, Université de Nice Sophia-Antipolis, 2003.
- [3] J.F. Aujol and A. Chambolle. Dual norms and image decomposition models. *International Journal of Computer Vision*, 63(1) :85–104, 2005.
- [4] J.F. Aujol and T. Chan. Combining geometrical and textured information to perform image classification. à paraître dans *Journal of Visual Communication and Image Representation*, 2006.
- [5] J.F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition-modeling, algorithms and parameter selection. *International Journal of Computer Vision*, 67(1) :111–136, 2006.
- [6] J.F. Aujol and S.H. Kang. Color image decomposition and restoration. à paraître dans *Journal of Visual Communication and Image Representation*, 2006.
- [7] J.F. Aujol and B. Matei. Simultaneous structure and texture compact representation. *Conférence Advanced Concepts for Imaging Vision Systems (ACIVS)*, 2004.
- [8] A. Averbuch, R.R. Coifman, D.L. Donoho, M. Israeli, and J. Waldén. Fast slant stack : a notion of radon transform for data in a cartesian grid wich is rapidly computible, algebraically exact, geometrically faithful and invertible. Rapport technique, Stanford University, 2001.
- [9] R.H. Bamberger and M.L.T. Smith. A filter bank for the directional decomposition of images : Theory and design. *IEEE Trans. Signal Processing*, 40(4) :882–893, 1992.
- [10] M.O. Berger. Les contours actifs : modélisation, comportement et convergence. Thèse de Doctorat, INP Lorraine, INRIA Lorraine, 1991.
- [11] H. Bresiz. Analyse fonctionnelle. DUNOD, 1996.

- [12] P. Brigger, R. Engel, and M. Unser. B-spline snakes and a JAVA interface : an intuitive tool for general contour outlining. pages 277–281. International Conference on Image Processing (ICIP), 1998.
- [13] P. Brigger, J. Hoeg, and M. Unser. B-spline snakes : a flexible tool for parametric contour detection. *IEEE Trans on Image Processing*, 9(9) :1484–1496, 2000.
- [14] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31(4) :532–540, 1983.
- [15] E. Candès. Ridgelets : Theory and Applications. Thèse de Doctorat, Department of Statistics, Stanford University, 1998.
- [16] E. Candès and D. Donoho. Curvelets : A surprisingly effective non-adaptive representation of objects with edges. Rapport technique, Department of Statistics, Stanford University, 1999.
- [17] E. Candès and D. Donoho. Continuous curvelet transform, part I : Resolution of the wavefront set. Rapport technique, Department of Statistics, Stanford University, 2003.
- [18] E. Candès and D. Donoho. Continuous curvelet transform, part II : Discretization and frames. Rapport technique, Department of Statistics, Stanford University, 2003.
- [19] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. Dans International Conference in Computer Vision (ICCV), pages 694–699, 1995.
- [20] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2) :89–97, mars 2004.
- [21] A. Chambolle, R.A. DeVore, N. Lee, and B.J. Lucier. Nonlinear wavelet image processing : variational problems, compression and noise removal through wavelet shrinkage. *IEEE Trans. on Image Processing*, 7 :319–335, 1998.
- [22] C.H. Chen, L.F. Pau, and P.S.P. Wang. The handbook of pattern recognition and computer vision (2nd edition). World Scientific Publishing, 1998.
- [23] C. Chesnaud. Techniques statistiques de segmentation par contour actif et mise en oeuvre rapide. Thèse de Doctorat, ENSPM, Université d’Aix-Marseille, 2000.
- [24] C. Chesnaud, P. Réfrégier, and V. Boulet. Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Trans on PAMI*, 21(11) :1145–1157, 1999.
- [25] L.D. Cohen. Etude des modèles de contours actifs et d’autres techniques de traitement d’image. Thèse de Doctorat, Université Paris VI - Jussieu, 1990.

- [26] A. Desolneux, L. Moisan, and J.-M. Morel. Maximal meaningful events and applications to image analysis. *Annals of Statistics*, 31(6) :1822–1851, 2003.
- [27] A. Desolneux, L. Moisan, and J.-M. Morel. Analyse d’images et géométrie stochastique. Polycopié de cours DEA MVA, 2004–2005.
- [28] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. Rapport technique, University of Massachusetts-INRIA, 1994.
- [29] M.N. Do. Directional Multiresolution Image Representations. Thèse de Doctorat, Department of Communication Systems, Swiss Federal Institute of Technology Lausanne, 2001.
- [30] M.N. Do. Contourlets and sparse image representations. SPIE conference on Wavelet Applications in Signal and Image Processing X, San Diego, USA, 2003.
- [31] M.N. Do and M. Vetterli. Pyramidal directional filter banks and curvelets. IEEE International Conference on Image Processing (ICIP), Thessaloniki, Greece, 2001.
- [32] M.N. Do and M. Vetterli. Contourlets : A directional multiresolution image representation. IEEE International Conference on Image Processing (ICIP), Rochester, 2002.
- [33] M.N. Do and M. Vetterli. The contourlet transform : an efficient directional multiresolution image representation. IEEE Trans. Image Processing, 14(12) :2091–2106, 2003.
- [34] M.N. Do and M. Vetterli. The finite ridgelet transform for image representation. IEEE Trans. Image Processing, 12 :16–28, 2003.
- [35] M.N. Do and M. Vetterli. Framing pyramids. IEEE Trans. Signal Processing, 51 :2329–2342, 2003.
- [36] D. Donoho. De-noising by soft-thresholding. IEEE Trans. Inform. Theory, 41 :613–627, 1995.
- [37] D. Donoho and M. Duncan. Digital curvelet transform : strategy, implementation and experiments. Rapport technique, Department of Statistics, Stanford University, 1999.
- [38] O. Germain. Segmentation d’image radar : caractérisation des détecteurs de bord et apport des contours actifs statistiques. Thèse de Doctorat, ENSPM, Université d’Aix-Marseille, 2001.
- [39] G. Gilboa, Y. Zeevi, and N. Sochen. Texture preserving variational denoising using an adaptive fidelity term. pages 137–144. Proceedings of VLISM, 2003.
- [40] J. Gilles and J.F. Aujol. Suppression de bruit multiplicatif par décomposition d’images - application aux images SAR. Journées imagerie polarimétrique - GdR ISIS - GdR Ondes, 2004.

- [41] J. Gilles and B. Collin. Fast probabilist snake algorithm. International Conference on Image Processing (ICIP), Barcelona, 2003.
- [42] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. Dans *Scale-Space Theories in Computer Vision*, pages 34–45, 1999.
- [43] A. Haddad. Méthodes variationnelles en traitement d'image. Thèse de Doctorat, Ecole Normale Supérieure de Cachan, 2005.
- [44] W. Härdle, G. Kerkyacharian, D. Picard, and A. Tsybakov. Wavelets, approximation and statistical applications. *Proceeding du séminaire Paris-Berlin*, 1997.
- [45] A.K. Jain. *Fundamentals of digital image processing*. Prentice Hall, 1989.
- [46] A.C. Kak and M. Slaney. *Principles of computerized tomographic imaging*. IEEE Press, 1999.
- [47] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active contours models. *International Journal of Computer Vision*, 1 :312–331, 1988.
- [48] C. Lacoste. Extraction de réseaux linéiques à partir d'images satellitaires et aériennes par processus ponctuels marqués. Thèse de Doctorat, INRIA, Université de Nice-Sophia Antipolis, 2004.
- [49] M. Lemaitre, F. Mériaudeau, O. Laligant, and J. Blanc-Talon. Distant horizontal ground observation : atmospheric perturbation simulation and image restoration. *Conférence Signal Image Technologie and Internet Based Systems (SITIS)*, 2005.
- [50] D.P.K. Lun, T.C. Hsung, and T.W. Shen. Orthogonal discrete periodic radon transform. Part I : theory and realization. *Signal Processing*, 83 :941–955, 2003.
- [51] D.P.K. Lun, T.C. Hsung, and T.W. Shen. Orthogonal discrete periodic radon transform. Part II : applications. *Signal Processing*, 83 :957–971, 2003.
- [52] S. Mallat. *Une exploration des signaux en ondelettes*. Number ISBN 2-7302-0733-3. Ecole Polytechnique, 2000.
- [53] H. Maître. *Le traitement des images*. Hermes - Lavoisier, 2003.
- [54] Y. Meyer. Oscillating patterns in image processing and in some non-linear evolution equations. *The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, American Mathematical Society, 2001.
- [55] Y. Meyer. Modélisation des textures en traitement d'image. Notes du séminaire du laboratoire Jacques-Louis Lions, 2003.
- [56] Y. Meyer and J. Gilles. Remarques sur un modèle de texture de L.Vese et S.Osher. *Journées texture - GdR ISIS*, 2003.

- [57] S. Osher and R. Fedkiw. Level set methods and dynamic implicit surfaces. Springer, 2002.
- [58] S. Osher, A. Sole, and L. Vese. Image decomposition and restoration using total variation minimization and the H^{-1} norm. *Multiscale Modeling and Simulation : A SIAM Interdisciplinary Journal*, 1(3) :349–370, 2002.
- [59] N. Paragios. Geodesic active regions and level set methods : contributions and applications in artificial vision. Thèse de Doctorat, University of Nice Sophia Antipolis, France, 2000.
- [60] N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. Dans *International Conference in Computer Vision (ICCV)*, pages 926–932, 1999.
- [61] N. Paragios and R. Deriche. Unifying boundary and region-based information for geodesic active tracking. Dans *IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, pages 1034–1040, 1999.
- [62] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 22(3) :266–280, 2000.
- [63] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans on PAMI*, 22(3), 2000.
- [64] D.D.Y. Po and M.N. Do. Directional multiscale modeling of images using the contourlet transform. à paraître dans *IEEE Trans. Image Processing*, 2006.
- [65] A.D. Poularikas. The transforms and applications handbook, second edition. CRC Press, IEEE Press, 1999.
- [66] O. Ruch. Reconnaissance des formes par contour actif statistique - Application à l'imagerie optronique active. Thèse de Doctorat, ENSPM, Université d'Aix-Marseille, 2001.
- [67] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60 :259–268, 1992.
- [68] D. Rueckert and P. Burger. Contour fitting using an adaptive spline model. volume 1, pages 207–216. *British Machine Vision Conference*, 1995.
- [69] G. Sapiro. Geometric partial differential equations and image analysis. Cambridge University Press, 2001.
- [70] J.L. Starck, E. Candès, and D. Donoho. The curvelet transform for image denoising. *IEEE Trans. Image Processing*, 11(6) :670–684, 2002.
- [71] J.L. Starck, M. Elad, and D.L. Donoho. Image decomposition : separation of texture from piecewise smooth content. *SPIE conference on*

- Signal and Image Processing : Wavelet Applications in Signal and Image Processing X, SPIE's 48th Annual Meeting, San Diego, 2003.
- [72] J.L. Starck, F. Murtagh, E. Candes, and D.L. Donoho. Gray and color image contrast enhancement by the curvelet transform. *IEEE Trans. Image Processing*, 12(6) :706–717, 2003.
- [73] P. Toft. The Radon transform. Theory and implementation. Thèse de Doctorat, Technical University of Denmark, 1996.
- [74] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing : Part I - theory. *IEEE Trans on Signal Processing*, 41(2) :821–833, 1993.
- [75] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing : Part II - efficient design and applications. *IEEE Trans on Signal Processing*, 41(2) :833–848, 1993.
- [76] L. Vese and S. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19(1-3) :553–572, 2002.
- [77] B. Vidakovic and P. Mueller. Wavelets for kids. <http://www.isye.gatech.edu/~brani/wp/kidsA.ps>, 1991. Cours d'introduction aux ondelettes.
- [78] J. Weickert and G. Kühne. Fast Methods for Implicit Active Contour, pages 43–58. Springer, 2003.
- [79] G.V. Welland. Beyond wavelets. Academic Press, 2003.
- [80] C. Xu and J.L. Prince. Snakes, shapes and gradient vector flow. *IEEE Trans on Image Processing*, 7(3) :359–369, 1998.