

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : électronique
Option : Télécommunication

Réf:.....

**Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:**

MASTER

Thème

Compression d'images fixes par bandelettes et
transformée d'Alpert

Présenté par :
Behnas Rahma
Soutenu le : 10 Juin 2012

Devant le jury composé de :

Melle Toumi.A

Mr Baarir.Z.E

Mr Ouafi.A

MAA

MCA

MCB

Président

Encadreur

Examineur

Année universitaire : 2011 / 2012

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : électronique
Option : Télécommunication

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:

MASTER

Thème

Compression d'images fixes par bandelettes et transformée d'Alpert

Présenté par :

BEHNAS RAHMA

Avis favorable de l'encadreur :

BAARIR.Z.E

Avis favorable du Président du Jury

TOUMIA

Cachet et signature



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : électronique
Option : Télécommunication

Thème :

Compression d'images fixes par bandelettes et transformée d'Alpert

Proposé par : Behnas Rahma

Dirigé par : Baarir.Z.E

RESUMES (Français et Arabe)

Ce travail consiste à l'étude d'une nouvelle technique de représentations des images fixes permettant de capturer la régularité géométrique le long des contours et la régularité de zones, appelée bases de bandelettes.

Les bases de bandelettes sont construites à partir d'ondelettes bidimensionnelles déformées le long du flot géométrique dans chaque région de l'image. Le flot géométrique indique la direction où l'image est régulière dans une région, L'optimisation de la géométrie doit être effectuée par une nouvelle transformée de Alpert qu'on va appliquer son algorithme rapide de meilleure base qui permet d'obtenir des résultats de compression plus satisfaisants.

تلخيص

هذا العمل هو دراسة لتقنية جديدة في ضغط الصور الثابتة حيث تعمل على تتبع النظامية الهندسية على طول حواف الصور و نظامية المناطق، تدعى الشريطيات. أساس الشريطيات تعتمد في بنائها على المموجات المعدلة ثنائية البعد على طول التدفق الهندسي في كل منطقة من الصورة. تدفق الهندسية يشير إلى الإتجاه أين الصورة تكون منتظمة في منطقة ما. يتم تحسين الهندسية عن طريق خوارزمية سريعة للحصول على نتائج ضغط مرضية

Dédicace

à ma grand-mère, la plus chers fatma, qui j'aime beaucoup.

à mes parents Sabah et Noureddine, car c'est d'eux que je tiens la détermination et la persévérance qui m'ont permis de réaliser mon mémoire. Que sa réalisation leur soit un témoignage de reconnaissance et d'amour, eux qui, dans la simplicité et l'humilité ont su transmettre à leurs fils et à moi-même les plus grandes et les plus nobles des valeurs morales.

à mes frères : Abderraouf, Abderrahim, Mohamed Cherif et Walid

à mes amis : Khaoula, Imene, Radja et Abderrahmane.

à tous ceux qui m'ont aimé et me souhaitent le bonheur et la réussite.

Remerciement

Je remercie tout d'abord mon dieu qui m'a donné le courage, la puissance et la patience pour terminer ce travail.

Ensuite, je remercie spécialement mes chères parents qui sont toujours avec moi et qui m'ont encouragé durant toute ma vie et toujours.

*Un grand remerciement a mon encadreur **Dr. BAARIR Z.E** qui nous a suivi avec beaucoup d'attention et qui m'a appui constamment tout au long de ce travail, et de me faire profiter de ses savoirs, conseils, et remarques qui m'a permit de réaliser et améliorer ce travail.*

*Je tien à remercie vivement **Dr. OUAFI. A** pour ses précieux conseils, son soutien et son aide pendant notre travail de fin d'étude.*

Aussi un grand remerciement aux membres de jury pour l'honneur qu'ils nous font en acceptant de juger ce travail.

De plus, je remercie tous les enseignants de notre département de l'électronique.

Nous tenant également a remercier tous les étudiants de notre promotion.

Enfin, je tiens à remercier tous ceux qui m'ont aidés.

A tous merci.

Listes des figures

Chapitre1 : notions théoriques sur les bandelettes

Figure 1.1: Fonction 1D, transformée en ondelettes, et approximation obtenue en gardant 10% des coefficients

Figure 1.2 : Exemple d'un triplet de fonctions ondelettes en 2D

Figure.1.3 : Approximation d'une image dans une base d'ondelettes

Figure1.4 : Image avec de la régularité géométrique et ses coefficients en ondelettes.

Figure 1.5 : -(a) Exemple de flot dans une région-(b) Exemple d'une segmentation quadtree adaptée d'une image et son flot géométrique.

Figure 1.6 : (a) Image originale f . (b) Coefficients d'ondelettes de f . (c) déformation du flot géométrique.

Figure. 1.7 : Discrétisation d'une fonction 2D. Chaque pixel correspond à un coefficient $f[n]$ de la projection de f sur une fonction d'échelle

Figure1.8 : Les coefficients en ondelettes sont des échantillons d'une fonction régularisée

Figure. 1.9 : Déformation de l'échantillonnage des coefficients d'ondelettes suivant un flot *géométrique*.

Figure1.10 : Construction d'un quadtree par subdivisions successives.

Figure1.11 : Un exemple de quadtree adapté.

Chapitre 2 :l'algorithme de calcul de la transformée d'Alpert.

Figure 2.1 : Exemple de regroupements dyadiques suivant une courbe géométrique.

Chapitre 3 : simulations et résultats.

Figure 3 : Organigramme de la compression / décompression avec bandelette et Alpert.

Figure3.1 : Un exemple de quadtree calculée en utilisant la fonction *compute-quadtree*.

Figure3.2 : La compression d'image en utilisant mono-résolution bandelettes

Figure3.3 : Un exemple de quadtree calculée en utilisant la fonction *compute-wavelet-quadtree*.

Figure3.4 : La compression d'image en utilisant multi-résolution bandelettes

Lexique d'abréviations

- L^2 : L'espace de la fonction d'énergie finie.
- $\langle x, y \rangle$: Produit scalaire de deux variables x et y .
- P : Nombre des moments nuls.
- T : Seuil.
- α : Ordre de régularité.
- 2^j : Echelle.
- $K^{H,V,D}$: Orientation Horizontale, Verticale et Diagonale
- f_j^k : Coefficients d'ondelettes de chaque échelle et chaque orientation.
- DWT : Transformée en Ondelettes Discrètes.
- DCT : Transformée en Cosinus Discrètes.
- 1D, 2D : une seule dimension, deux dimensions.
- $C(x)$: Courbe intégrale du flot géométrique.
- \vec{c} : Vecteur de flot géométrique.
- $Wf(x_1, x_2)$: Déformation de l'image $f(x_1, x_2)$.
- $B(\Gamma)$: Base de bandelettes.
- Γ : Géométrie.
- Q_j^k : Segmentation en quadtree de chaque échelle et chaque orientation.
- $\tilde{\mathcal{Y}}_s$: Géométries approchées à l'intérieur des carrés de la segmentation.
- $\tilde{\mathcal{Y}}$: Géométries approchées.
- Ω_i : Région de la segmentation des flots.
- D_j^k : Dictionnaire de bases de bandelettes discrètes
- $\Delta_j(S)$: Base de Dirac.
- \mathcal{E} : Ensemble de géométrie.
- EQM : Erreur Quadratique Moyenne.
- W : warping : déformation.
- \mathcal{L} : Lagrangien.
- RC : taux de compression.
- PSNR : rapport signal sur bruit crête.

Résumé

Ce travail consiste à l'étude d'une nouvelle technique de représentations des images fixes permettant de capturer la régularité géométrique le long des contours et la régularité de zones, appelée bases de bandelettes.

Les bases de bandelettes sont construites à partir d'ondelettes bidimensionnelles déformées le long du flot géométrique dans chaque région de l'image. Le flot géométrique indique la direction où l'image est régulière dans une région, L'optimisation de la géométrie doit être effectuée par une nouvelle transformée de Alpert qu'on va appliquer son algorithme rapide de meilleure base qui permet d'obtenir des résultats de compression plus satisfaisants.

Mots clés

Transformée d'Alpert, Bandelettes, ondelettes géométriques, segmentation en quadtree, quantification de la géométrie.

Abstract

This work consists of the study of a new technique of fixed images, making it possible to capture both the geometrical regularity along the contours and the regularity the proposed zone.

The bandelets bases are built through wavelets two-dimensional deformations along the geometrical flow in each area of the image. This geometric flow indicates directions in which the image has regular variations. The geometric flow optimized with a new transform ‘Alpert transform’ fast best basis algorithm, which leads better image compression results.

Key words:

Alpert transform, bandelets, geometric wavelets, quadtree segmentation, quantification of the geometry.

Sommaire

Sommaire

Introduction générale

CHAPITRE 1-Notions théoriques sur les bandelettes

1.1. Introduction	15
1.2.Succès et échecs des bases d'ondelettes.....	15
1.2.1. Limitation des ondelettes.....	15
1.2.2.Bases d'ondelettes 1D.....	15
1.2.3.Bases d'ondelettes 2D.....	17
1.4. Succès des bases d'ondelettes.....	19
1.3. La Transformation En Bandelette.....	19
1.3.1Base de bandelettes première génération	19
1.3.2.Bandelettes orthogonales.....	21
1.4.Bandelettes pour la compression d'images.....	27
1.4.1.Compression.....	27
1.4.2.Codeur en bandelettes orthogonales.....	28
1.5.Conclusion.....	29

CHAPITRE 2-Transformée en bandelettes discrète

2.1. Introduction.....	30
2.2. Transformée en bandelettes discrète.....	31
2.2.1.Image discrétisée	31
2.2.2. Fonction d'échelles en ondelettes.....	31
2.2.3. Base de bandelettes discrètes	32
2.2.4. Transformée en bandelettes discrètes	32

2.3. ALGORITHME RAPIDE.....	33
2.4. Transformée D’Alpert 2D.....	36
2.4.1. Groupements dyadiques de points.....	37
2.4.2. Espaces de multi résolution polynomiaux	38
2.4.3. Construction de la base d’Alpert	38
2.5. Conclusion.....	39

CHAPITRE 3-programmation et résultat

3. Organigramme de la compression par bandelettes et transformée d’Alpert .	41
3.1. Introduction.....	42
3.2. Structure du programme.....	42
3.3 Algorithme de transformée en bandelettes et d’Alpert.....	43
3.4. Discussions.....	46
3.5 Conclusion	46
Conclusion générale.....	47
Annexe.....	49
Bibliographie.....	55

Introduction générale

Introduction générale

La recherche d'une bonne représentation est un problème central du traitement d'image. On cherche ici des bases permettant une représentation creuse des images, c'est-à-dire permettant de bien les approcher avec peu de coefficients.

Cette propriété est en effet à la base de nombreux algorithmes de compression ou de débruitage. Les bases de bandelettes permettent d'obtenir une telle représentation. Elles ne sont cependant pas optimales pour les images. Bien que très efficaces pour les zones régulières, les textures homogènes et les singularités ponctuelles, elles ne peuvent exploiter la régularité de nature géométrique des contours.

On cherche à obtenir la meilleure représentation possible avec le moins de coefficients possibles pour cela il va falloir approcher à la fois la géométrie via les contours, les problèmes de singularités via les bandelettes et la transformée d'Alpert et le résidu via les ondelettes.

Nous proposons ici la construction d'une nouvelle représentation permettant de capturer à la fois cette régularité le long des contours. Elle est basée sur une nouvelle fonction qui va améliorer la représentation des régularités des zones géométrique des images.

Cette technique se base sur la quantification de la géométrie et l'implémentation de la transformée d'Alpert ensuite la construction du quadtree.

Ce mémoire se compose de trois chapitres distincts :

Le premier chapitre est consacré aux concepts théoriques de la transformée en bandelettes.

Le deuxième chapitre comporte l'algorithme de calcul de la transformée de Alpert.

Le troisième chapitre comporte les simulations effectuées et la discussion des résultats obtenus.

Ce travail est achevé par une conclusion générale et des perspectives.

Chapitre 1

Notions théoriques sur
Les bandelettes

1.1. Introduction

Les représentations actuellement utilisées en traitement des images sont basées sur des transformées séparables.

Celles-ci ne capturent pas la régularité géométrique des images le long des contours, bien que ceci soit un aspect essentiel des images.

Nous proposons ici la construction d'une nouvelle représentation permettant de capturer à la fois cette régularité le long des contours et la régularité de zones. Elle est basée sur une nouvelle famille de bases, les bandelettes, vivant le long de contours et permettant de capturer les singularités le long de ceux-ci.

1.2. Succès et échecs des bases d'ondelettes

1.2.1. Limitation des ondelettes

Une approche courante pour analyser une image est de trouver une représentation compacte de son contenu souvent par le biais d'une transformation linéaire inversible. Récemment, il est devenu évident que les transformations usuelles de type séparable, telles que la transformée en ondelettes, ne sont pas nécessairement les mieux adaptées pour la représentation des images de nature géométrique. En effet, une des lacunes principales des transformées classiques réside dans leur incapacité de prendre en compte la régularité des structures géométriques des images [1]. En effet, les discontinuités d'une fonction bidimensionnelle présente souvent une structure géométrique que le noyau classique n'exploite pas. Dans le cas de l'image, ces singularités, ou contours, possèdent pourtant un fort poids sémantique et perceptuel.

Le support de l'ondelette ne capte pas cette régularité existant le long des contours, ce qui provoque une perte en compacité de représentation. Lors d'une approximation, cette perte se répercute visuellement par des rebonds tout autour de la singularité. Une adaptation du noyau au contours est également illustrée sur la droite et met en avant les deux propriétés qui font défaut au noyau classique : la directionalité et l'anisotropie.

Il est nécessaire de rechercher des techniques permettant de s'approcher plus encore de l'optimalité, tout en essayant de conserver les atouts des ondelettes. L'objectif est de rechercher de nouvelles bases de représentation exhibant ces deux propriétés tout en conservant celles ayant fait le succès des ondelettes classique : échantillonnage critique, localisation (spatiale et fréquentielle), multi-résolution [5].

1.2.2. Bases d'ondelettes 1D

Une base d'ondelettes B de $L^2([0; 1]^2)$ est obtenue en dilatant et translatant une fonction ψ [9]

$$B \stackrel{\text{def}}{=} \{ \psi_{jn} \mid j \leq 0, n = 0 \dots 2^{-j} - 1 \} \quad \text{avec} \quad \left\{ \psi_{jn}(x) \stackrel{\text{def}}{=} 2^{\frac{j}{2}} \psi(2^{-j}x - n) \right\}$$

Les ondelettes sont des fonctions oscillantes qui ont des moments nuls. Une ondelette a p moments nuls si elle est orthogonale aux polynômes de degré $p-1$

$$\forall k \leq p - 1, \int_0^1 \psi(x) x^k dx = 0 \dots \dots \dots (1,1)$$

Si une fonction f est régulière, par exemple de classe C^α sur un intervalle contenant le support d'une fonction $\psi_{j,n}$, alors le produit scalaire $\langle f, \psi_{j,n} \rangle$ va être quasiment nul.

Elle a un support compact, de taille m . ainsi une fonction $\psi_{j,n}$ a un support de taille $k2^j$ et est localisée autour du point $2^j n \in [0, 1]$.

Ces deux propriétés font de la base d'ondelette un outil efficace pour analyser les fonctions 1D ayant des singularités ponctuelles.

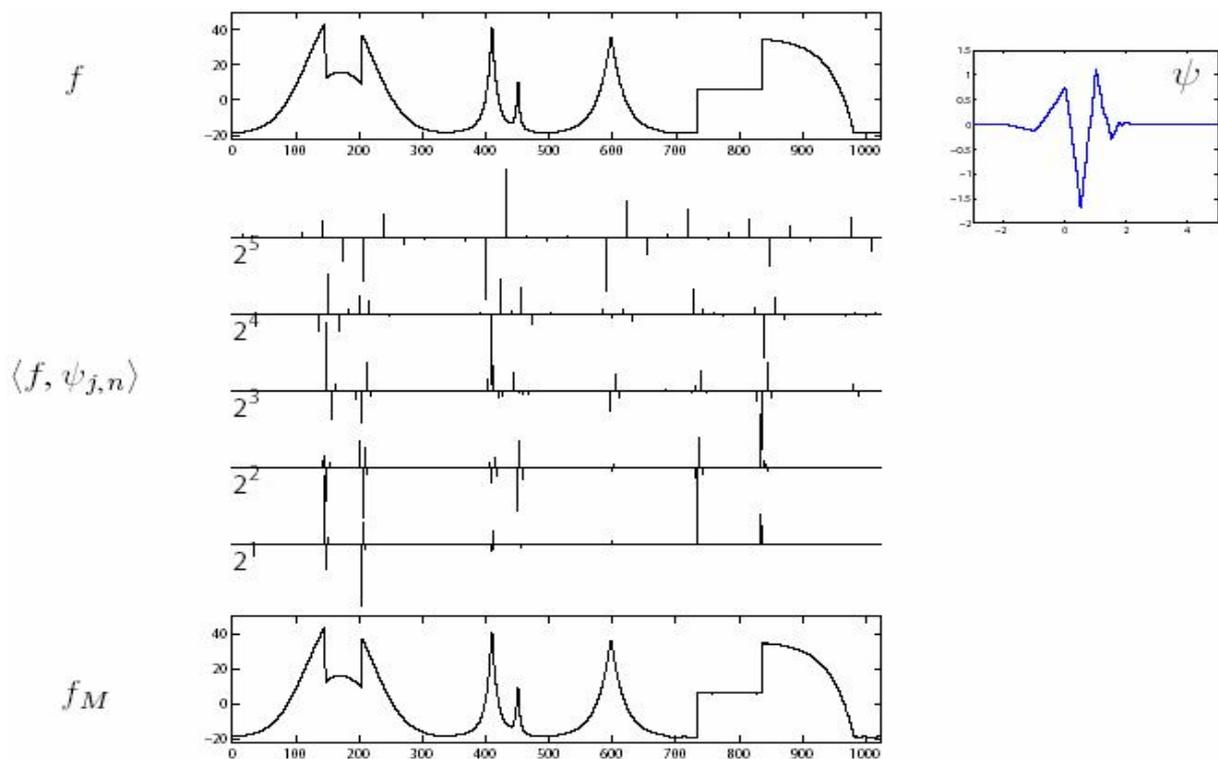


Figure 1.1: Fonction 1D, transformée en ondelettes, et approximation obtenue en gardant 10% des coefficients

La Figure 1.1 montre une fonction régulière par morceaux ainsi que ses coefficients en ondelettes $\langle f, \psi_{j,n} \rangle$. On peut constater que les grands coefficients sont peu nombreux et localisés au voisinage des singularités.

si la fonction est C^α par morceaux et que l'ondelette ψ à $p \geq \alpha$ moment nuls, alors la meilleure approximation f_M obtenue avec M coefficients d'ondelettes vérifie :

$$\|f - f_M\|_{L^2}^2 \leq CM^{-2\alpha} \dots\dots\dots(1,2)$$

Où C est constante qui ne dépend que de cette décroissance asymptotique est optimale pour les fonctions régulières par morceau [1].

1.2.3.Bases d'ondelettes 2D

Les bases d'ondelettes de $L^2([0; 1]^2)$ sont obtenues par translations et dilatations de 3 ondelettes élémentaires $\{\psi^H, \psi^V, \psi^D\}$ qui oscillent dans les directions horizontale, verticale, et diagonale. Ces ondelettes bidimensionnelles s'obtiennent par un produit séparable d'ondelettes monodimensionnelles. La figure 1.2 montre un exemple d'ondelettes 2D.

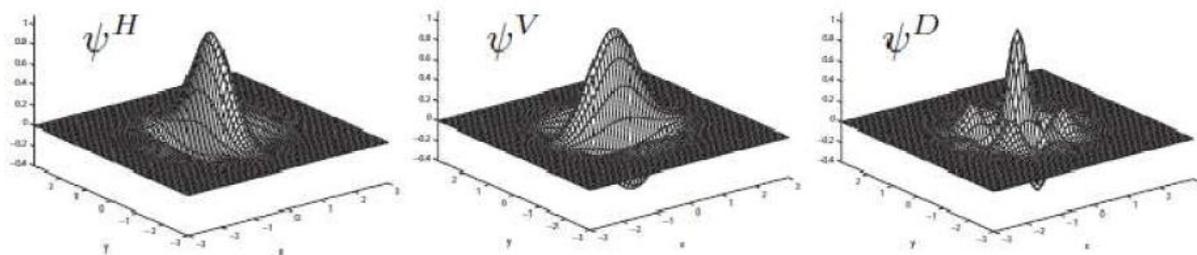


Figure 1.2 : Exemple d'un triplet de fonctions ondelettes en 2D

La transformée en ondelettes analyse une fonction $f \in L^2([0;1]^2)$ en calculant la décomposition de f dans une base d'ondelettes. Pour chaque échelle 2^j et orientation $k \in \{V, H, D\}$.

$$f_j^k[n_1, n_2] \triangleq \langle f, \Psi_{jn}^k \rangle \quad \text{Avec} \quad \Psi_{jn}^k(x_1, x_2) \triangleq 2^{-j} \psi^k(2^{-j}x_1 - n_1, 2^{-j}x_2 - n_2)$$

On peut interpréter cet ensemble de coefficient comme une image f_j^k contenant les coefficients d'ondelettes de f pour chaque échelle 2^j et orientation k. la fonction Ψ_{jn}^k est localisée au voisinage du point $2^j n$ sur un carré de taille $2^j k$.

La figure (1.3) montre les coefficients en ondelettes d'une image selon les 3 orientations. Ces coefficients ont été seuillés afin de ne garder que 15% et 2% des plus grands coefficients en (b) et (c). Le zoom effectué sur la reconstruction f_M montre qu'avec seulement 15% des coefficients, on obtient une

reconstruction précise de l'image, mais qui se dégrade lorsque ce pourcentage diminue. Le standard de compression d'images JPEG2000 décompose l'image dans une base d'ondelettes et effectue une quantification et un codage entropique des coefficients afin d'optimiser le code binaire [2].

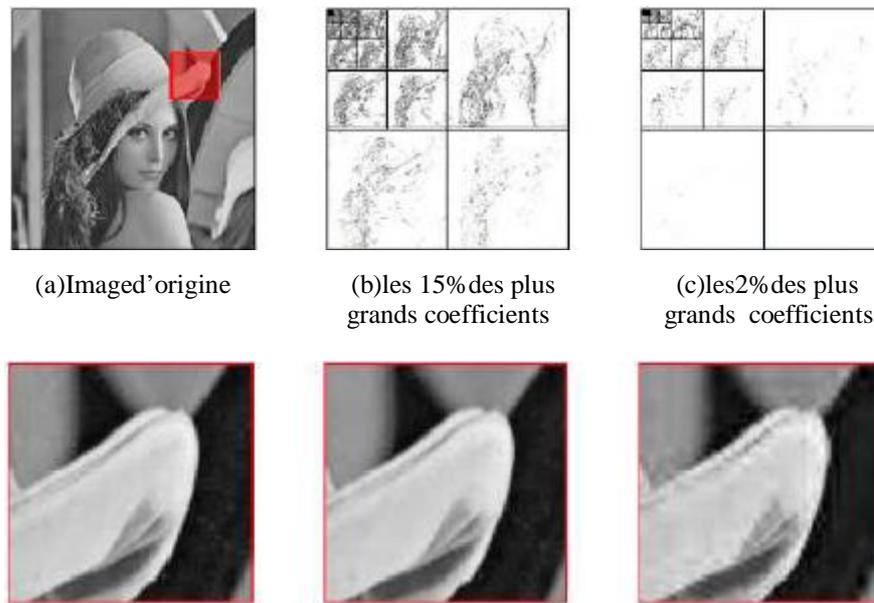


Figure.1.3 : Approximation d'une image dans une base d'ondelettes

Pour une fonction f ayant une régularité géométrique C^α la meilleure approximation f_M avec M coefficients dans une base d'ondelettes satisfait:

$$\|f - f_M\|_2^2 \leq CM^{-1} \dots\dots\dots(1,3)$$

Où C est une constante qui ne dépend que de f , ce taux d'approximation n'est pas optimal et la qualité de l'approximation est complètement dirigée par la présence de discontinuités.

Les bases d'ondelettes orthogonales sont capables de résoudre un problème essentiellement 1D, celui de l'analyse des singularités ponctuelles. En 2D, le problème devient beaucoup plus complexe, à cause de la présence de singularités curvilignes. Les ondelettes classiques ne sont pas capables de représenter de telles singularités de façon efficace à cause de leur support carré [3].

La figure (1.4) montre les coefficients d'ondelettes d'une image régulière par morceaux. On voit que les grands coefficients en ondelettes sont le long des contours (coefficients blancs et noirs) et le nombre de ces coefficients est donc proportionnel à la longueur des contours. Ce sont ces coefficients qui déterminent la précision de l'approximation [8]

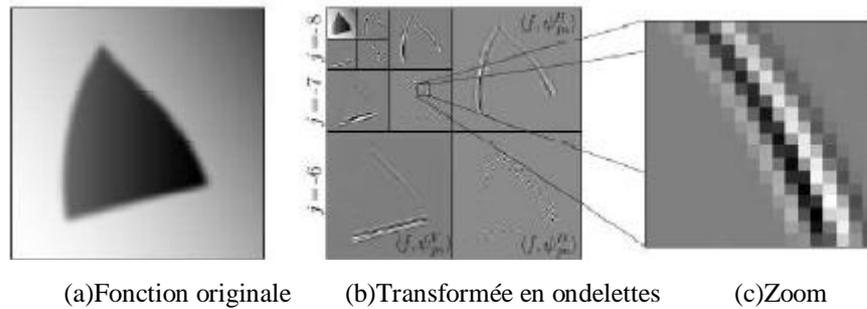


Figure 1.4 : Image avec de la régularité géométrique et ses coefficients en ondelettes.

L'enjeu des représentations géométriques est d'utiliser la régularité géométrique des « Singularités » de l'image afin d'améliorer le résultat d'approximation obtenu avec des ondelettes. En particulier, on voudrait obtenir des approximations qui satisfont $\|f - f_M\|_{L^2} \leq CM^{-\alpha}$, (Où C est une constante qui ne dépend que de f) comme s'il n'y avait pas de singularité dans l'image. C'est en effet le résultat obtenu en 1D.

1.4. Succès des bases d'ondelettes

Bien que non optimale pour l'approximation d'images géométriques, les bases d'ondelettes sont en pratique un outil très efficace.

Les ondelettes sont également employées en compression d'images avec succès. Elles sont en effet à la base du nouveau standard de compression d'image JPEG2000.

Le débruitage par seuillage dans une base d'ondelettes, utilise la capacité des ondelettes à représenter de façon compacte une image, et ainsi à bien séparer le signal du bruit.

1.3. La Transformation En Bandelette

1.3.1 Base de bandelettes première génération

Les premières bases de bandelettes construites par E. Le Pennec et S. Mallat [10] ont permis d'obtenir des résultats d'approximation asymptotiquement optimaux pour des fonctions géométriquement régulières [4, 11].

Au lieu de représenter la géométrie des images par contours, qui ne sont souvent pas bien définie, celle-ci est définie par des flots géométriques. Les vecteurs d'un flot indiquent des directions où l'image a localement des variations régulières. Les bases orthonormées de bandelettes sont construites en divisant l'image en régions où ces flots restent parallèles.

Les bandelettes sont construites à partir d'ondelettes bidimensionnelles déformées le long du flot géométrique.

Dans une région Ω , le flot géométrique est un champ de vecteur $\vec{v}(x_1, x_2)$, qui donne

une direction dans laquelle la fonction f est régulière dans un voisinage. Pour construire des bases orthonormées, une condition de parallélisme doit être imposée sur le flot. Celui-ci est choisi soit parallèle verticalement, $\vec{v}(x_1, x_2) = \vec{v}(x_1)$, soit parallèle horizontalement, $\vec{v}(x_1, x_2) = \vec{v}(x_2)$.

Pour permettre plus de flexibilité, cette condition de parallélisme n'est imposée que dans des sous-régions Ω_i de Ω . La région Ω est donc partitionnée en des régions Ω_i munies d'un flot parallèle soit verticalement soit horizontalement (figure (1.5)).

On définit alors une base pour chaque région Ω . Si aucun flot n'est défini, on utilise une base d'ondelettes bidimensionnelles de $L^2(\Omega)$. Sinon, cette base est remplacée par une base de bandelettes. La construction de cette base est maintenant présentée dans le cas où le flot est parallèle verticalement, le cas horizontal étant similaire.

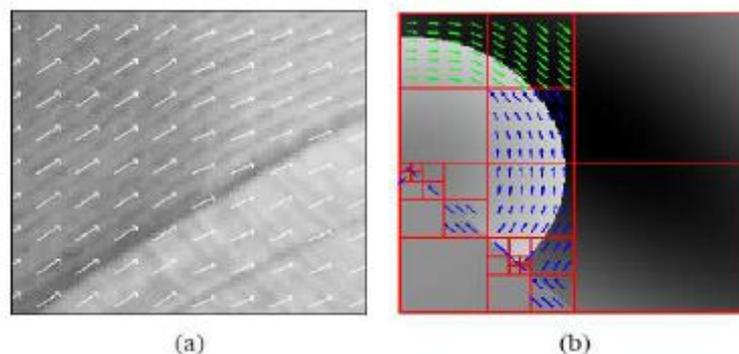


Figure 1.5 : - (a) Exemple de flot dans une région
 (b) Exemple d'une segmentation quadtree adaptée d'une image et son flot géométrique.

Le flot s'écrit donc, $\vec{v}(x_1, x_2) = \vec{v}(x_1)$ et quitte à le renormaliser, on peut l'écrire sous la forme $\vec{v}(x_1) = (1, C'(x_1))$. On pose alors $x_{min} = \inf_{x_1} \{x_1, x_2 \in \Omega\}$ et on définit la ligne de flot comme une courbe intégrale du flot. Celle-ci satisfait l'équation $x_2 = c(x_1) + c_0$ où

$$C(x) = \int_{x_{min}}^x c'(u) du \dots\dots\dots(1,4)$$

et c_0 est un paramètre de translation.

Par construction, l'image a une variation régulière le long de ces lignes. Afin d'exploiter cette régularité, les ondelettes bidimensionnelles sont déformées pour suivre le flot (figure (1.6)).

L'image déformée $W(f(x_1, x_2 + c(x_1)))$ est régulière le long des lignes horizontales (x_2 fixé). On utilise alors pour la région déformée

$$\Omega' = W\Omega = \{(x_1, x_2) : (x_1, x_2 + c(x_1)) \in \Omega\} \dots\dots\dots(1,5)$$

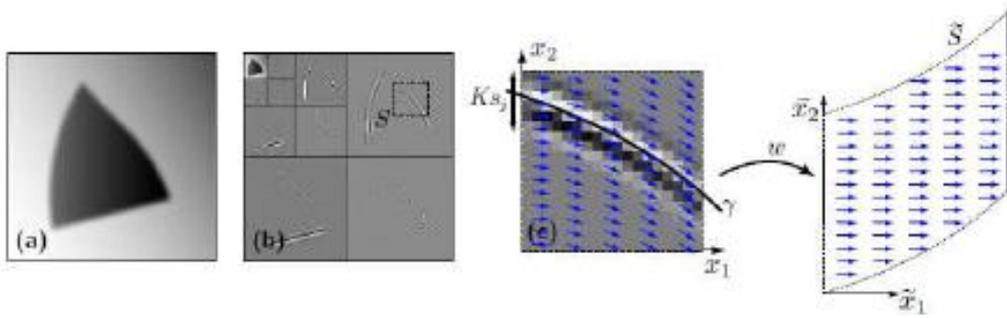


Figure 1.6 : (a) Image originale $_$. (b) Coefficients d'ondelettes de $_$. (c) déformation du flot géométrique.

la base d'ondelettes bidimensionnelles de $L^2(\Omega)$:

$$\begin{cases} \phi_{j,m_1}(x_1)\psi_{j,m_2}(x_2) \\ \psi_{j,m_1}(x_1)\phi_{j,m_2}(x_2) \\ \psi_{j,m_1}(x_1)\psi_{j,m_2}(x_2) \end{cases} \dots\dots\dots(1,6) \\ (j,m_1,m_2) \in I_\Omega$$

L'opérateur de déformation W étant orthogonal, l'application de son inverse à ces ondelettes donne une base orthonormée de $L^2(\Omega)$, que l'on appelle base d'ondelettes déformées :

$$\begin{cases} \phi_{j,m_1}(x_1)\psi_{j,m_2}(x_2 - c(x_1)) \\ \psi_{j,m_1}(x_1)\phi_{j,m_2}(x_2 - c(x_1)) \\ \psi_{j,m_1}(x_1)\psi_{j,m_2}(x_2 - c(x_1)) \end{cases} \dots\dots\dots(1,7) \\ (j,m_1,m_2) \in I_\Omega$$

Afin d'exploiter la régularité de la fonction f selon le flot, on remplace les ondelettes déformées de la forme (la fonction d'échelle $\phi_{j,m_1}(x_1)$, n'a pas des moments nuls):

$$\{\phi_{j,m_1}(x_1)\psi_{j,m_2}(x_2 - c(x_1))\}_{m_1,m_2} \dots\dots\dots(1,8)$$

Par la famille de fonctions engendrant le même espace qui ont des moments nuls le long de x_1

$$\{\psi_{l,m_1}(x_1)\psi_{j,m_2}(x_2 - c(x_1))\}_{l > j,m_1,m_2} \dots\dots\dots(1,9)$$

Cette opération est appelée **bandelétisation** et on vérifie qu'elle s'implémente par une simple transformée en ondelettes monodimensionnelles discrètes. Les fonctions:

$$\begin{cases} \psi_{l,m_1}(x_1)\psi_{j,m_2}(x_2 - c(x_1)) \\ \psi_{j,m_1}(x_1)\phi_{j,m_2}(x_2 - c(x_1)) \\ \psi_{j,m_1}(x_1)\psi_{j,m_2}(x_2 - c(x_1)) \end{cases} \dots\dots\dots(1,10) \\ j,l > j,m_1,m_2$$

Pour une partition donnée $\Omega = \cup_i \Omega_i$ et les flots géométriques correspondants, on définit ainsi une base orthonormée de bandelettes ou d'ondelettes (si il n'y a pas de flots) pour chaque Ω_i . L'union de ces bases forment une base par blocs de $L^2(\Omega)$. L'orthogonalité des bandelettes et des ondelettes peut-être relâchée : en utilisant une ondelette biorthogonale monodimensionnelle ψ et sa fonction d'échelle ϕ , la même construction donne une base biorthogonale pour chaque $L^2(\Omega_i)$ et donc une base biorthogonale par blocs de $L^2(\Omega)$ [4].

1.3.2. Bandelettes orthogonales

Géométrie Discrète. Les bandelettes orthogonales sont construites sur un domaine discret, celui des coefficients d'ondelettes. Ceci permet d'unifier à la fois la théorie mathématique et la pratique numérique sur des images ayant une précision finie.

En effet, il faut garder à l'esprit que, même si notre modèle mathématique suppose l'existence d'une fonction $f \in L^2$, en pratique, nous avons accès uniquement à un vecteur de coefficients discrétisés, noté f . Ce procédé de discrétisation concerne aussi bien la vision humaine (acquisition par les cônes de la rétine) que la prise de photos numériques (acquisition par les capteurs CCD). On peut modéliser cette étape de discrétisation comme une projection sur un ensemble de fonctions d'échelles translattées.

À la figure 1.7 on peut voir une fonction d'ondelette 1D ψ , la fonction d'échelle correspondante et l'effet de la discrétisation sur une fonction $f \in L^2$ (en réalité une image à haute résolution $2^{-j} = 512$) par projection à la résolution $2^{-j} = 32$.



Figure. 1.7 : Discrétisation d'une fonction 2D. Chaque pixel correspond à un coefficient $f[n]$ de la projection de f sur une fonction d'échelle

Géométrie multi-échelles. Dans le cas d'une image géométrique, comme on peut le voir à la figure 1.8, les grands coefficients sont localisés à chaque échelle près des courbes de singularité. En effet, si le support d'une ondelette $\psi_{j,m}^k$ ne rencontre pas une telle courbe, le coefficient d'ondelettes $f_j^k[n]$ va être petit.

On peut donc se concentrer sur les grands coefficients, qui sont localisés dans des tubes de largeur $m2^j$ (où m est la taille du support des ondelettes ψ^k) autour des courbes de singularité. Ainsi on a remplacé un problème **2D** (l'approximation de la fonction f) par un problème essentiellement **1D** (l'approximation des coefficients $f_j^k[n]$ autour des singularités).

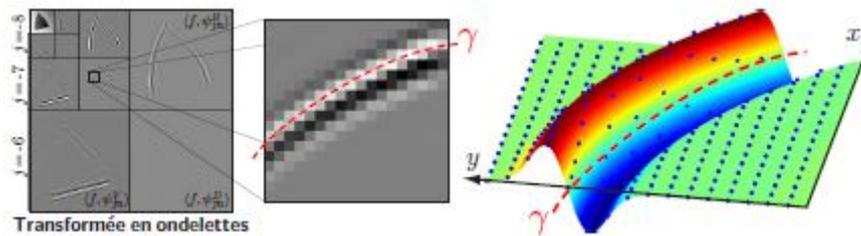


Figure 1.8 : Les coefficients en ondelettes sont des échantillons d'une fonction régularisée

Pour quantifier précisément la régularité de cet ensemble de coefficients, on remarque que l'on peut les écrire comme des échantillons d'une fonction régularisée

$$(f, \psi_{jm}^k) = f * \psi_j^k(n_1 2^j, n_2 2^j) \quad \text{où} \quad \psi_j^k(x) = 1/2^j \psi^k(-2^{-j}x)$$

Supposons que l'on définisse un champ de vecteurs approximativement parallèle au contour comme l'illustre la figure 1.9 $f * \psi_j^k$ hérite de régularité de $\psi_j^k(x)$ et de la régularité géométrique de $f(x)$ parallèlement au flot, comme le montre la surface sur la droite de figure 1.9. on rectifie le flot pour qu'il devienne horizontal à une déformation régulière du plan. Cette déformation modifie la grille carrée de l'échantillonnage des coefficients d'ondelettes orthogonaux, comme le montre la figure 1.9.

Dans le domaine déformé, le fait que le flot soit horizontal indique que la fonction $f * \psi_j^k$ a des variations régulières le long de droites horizontales.

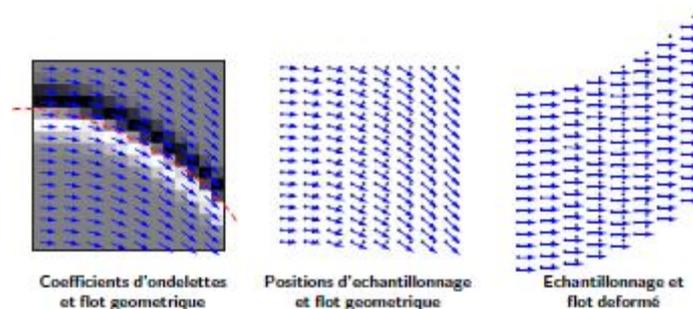


Figure. 1.9 Déformation de l'échantillonnage des coefficients d'ondelettes suivant un flot géométrique.

Segmentation en quadtree

Une segmentation en quadtree de $[0; 1]^2$ est obtenue par subdivisions successives du carré initial en quatre carrés de tailles égales. La figure 1.1 montre les étapes de subdivision menant à la construction d'un

quadtree, ainsi que l'arbre quaternaire permettant de représenter cette subdivision.

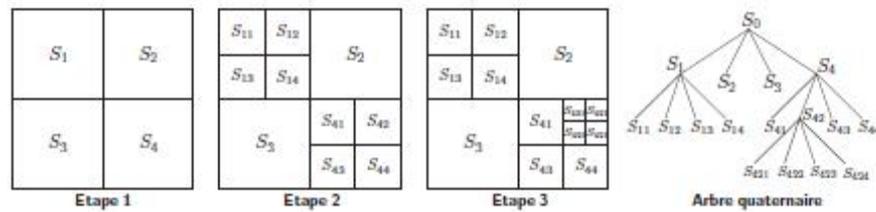


Figure1.10 : Construction d'un quadtree par subdivisions successives.

Ceci mène à une segmentation de $[0;1]^2$ en des carrés de tailles dyadiques. On cherche une segmentation qui encapsule les courbes de singularité par un ensemble de carrés de taille b et qui remplissent les zones restantes par des grands carrés. Pour une segmentation Q_j^k de $[0;1]^2$ en carrés de largeurs dyadiques plus grandes que 2^j , nous répartissons ces carrés en plusieurs groupes :

- L'ensemble $\mathcal{E}_b(Q_j^k)$ des carrés de largeur b contenant un contour. Par définition, un carré contenant un contour est un carré distant de moins de k_{ij} d'une et une seule courbe de contour. Cette courbe doit pouvoir être paramétrée soit horizontalement soit verticalement par une fonction γ satisfaisant $|\gamma'| \leq 2$.
- L'ensemble $\mathcal{E}_-(Q_j^k)$ des carrés contenant un contour de largeur inférieure à b .
- L'ensemble $\mathcal{C}(Q_j^k)$ des carrés contenant un coin. Par définition, un tel carré contient une jonction de deux courbes.
- L'ensemble $\mathcal{R}(Q_j^k)$ des carrés réguliers, qui sont les carrés restants.

Idéalement, nous souhaiterions n'avoir que des carrés réguliers ainsi que des carrés de taille b contenant une jonction. Mais des contraintes topologiques (comme des croisements de contours et des jonctions) et les variations de courbures (par exemple un contour horizontal qui devient vertical) nous obligent à subdiviser certains carrés de taille b en des carrés plus petits. Cependant, comme montré par Le Pennec et Mallat [12], on peut construire un quadtree possédant un faible nombre de carrés de petite taille. La figure (1.16) montre un quadtree adapté à une fonction géométrique.

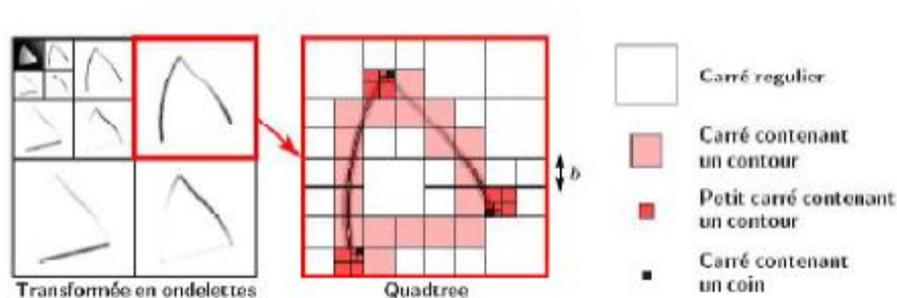


Figure1.11 : Un exemple de quadtree adapté

La famille de bandelettes orthogonales dépend du flot local à chaque échelle 2^j et orientation k . Ce flot est parallèle et caractérisé par une courbe intégrale γ illustrée en pointillé dans la figure 1.9. Cette courbe intégrale n'a pas besoin d'être strictement parallèle au contour grâce à la régularisation bidimensionnelle induit par l'ondelette ψ_j^k sur la fonction f . Localement, on démontre qu'il suffit de prendre une approximation polynomiale du contour γ cette géométrie approchée est assez précise pour la construction que l'on réalise selon lemme suivant :

Soit γ une fonction de classe C^α définie sur un intervalle I de longueur b et soit $\{\theta_i\}_i$ la base orthogonale des polynômes de Legendre sur I . alors le polynôme

$$\tilde{\gamma} = \sum_{i=0}^{p-1} (\gamma, \theta_i) \theta_i \quad \text{vérifie} \quad \forall x \in I, |\gamma(x) - \tilde{\gamma}(x)| \leq C b^\alpha$$

La segmentation en carré ($S \subset [0,1]^2$) de coefficients d'ondelettes est nécessaire afin d'approximer la géométrie par un flot polynomial sur chaque carré S . pour chaque échelle 2^j et orientation k de la transformée en ondelettes, une division récursive en carrés dyadiques, qui peuvent avoir des tailles variables, comme l'illustre la figure 1.11.

Une géométrie différente de l'image peut être choisie à chaque échelle pour s'adapter à l'évolution des structures géométriques de l'image aux différentes échelles .

Si la régularité de l'image n'a pas de direction préférentielle, comme cela peut être le cas dans des carré uniformément régulières ou au voisinage de jonctions, il n'y a pas de régularité géométrique directionnelle à exploiter . Il n'est donc pas nécessaire de modifier la base d'ondelettes. Dans ce cas aucun flot n'est défini (cela correspond aux carrés dans zones de « **régularité isotrope** » et aux carrés de « **jonction** » de la figure 1.11. ce n'est que dans les carrés le long des « **contours** » que le flot doit être calculé, pour obtenir une base de bandelettes qui exploite la régularité anisotrope de l'image [6].

Construction d'une base de bandelette

Soit $S \subset [0,1]^2$ un carré contenant des coefficients d'ondelettes à une échelle 2^j et orientation k . soit $\tilde{\gamma}_s$ une orientation approchée dans S . Dans la section précédente, nous avons construit une base de bandelette $B(S, \tilde{\gamma}_s) \stackrel{\text{def}}{=} \{\psi_{\ell i}\}_{\ell, i}$ de $\ell_j^2(S)$. On note toujours par $B(S, \tilde{\gamma}_s)$ les vecteurs de $\ell_j^2([0,1]^2)$ étendu à partir des vecteurs $\psi_{\ell i} \in B(S, \tilde{\gamma}_s)$ par

$$\forall (2^j n) \in S, \psi_{\ell i}[n] \stackrel{\text{def}}{=} 0 \dots \dots (1,12)$$

Soit $\Gamma_j^k \stackrel{\text{def}}{=} (Q_j^k, \{\tilde{\gamma}_s\}_{S \in Q_j^k})$ une segmentation en quadtree Q_j^k de $[0,1]^2$ à une échelle 2^j et orientation k , auquel on ajoute, pour chaque carré $S \in Q_j^k$, par une géométrie approchée $\tilde{\gamma}_s$.

Une base de bandelettes $B(\Gamma_j^k)$ de l'espace entier de coefficients $\ell_j^2([0,1]^2)$ est définie par

$$B(\Gamma_j^k) \stackrel{\text{def}}{=} \bigcup_{S \in Q_j^k} B(S, \tilde{\gamma}_S) \quad \dots\dots(1,13)$$

Un vecteur de bandelettes $\psi_\mu \in B(\Gamma_j^k)$ est ainsi spécifié par $\mu = (j, k, S, \tilde{\gamma}_S, \ell, i)$ où

- 2^j est l'échelle de la transformée en ondelettes 2D et $k \in \{V, H, D\}$ est l'orientation.
- $S \in Q_j^k$ est un carré dyadique de largeur $b = L 2^j$, où L est une puissance de 2.
- $\tilde{\gamma}_S$ est une géométrie approchée.
- $\ell \in \{-2 \log_2(L), \dots, 0\}$ et $i \in \{1, \dots, 2^\ell\}$ sont l'échelle et l'index d'un vecteur $\psi_{\ell i}$ dans la base de bandelettes orthogonales $B(S, \tilde{\gamma}_S)$.

Nous avons ainsi défini

$$\forall x_n \in S \in Q_j^k, \psi_\mu[n] = \psi_{\ell i}[n] \quad \text{où} \quad B(S, \tilde{\gamma}_S) = \{\psi_{\ell i}\}_{\ell, i}$$

Ces vecteurs discrets $\psi_\mu \in \ell^2([0,1]^2)$ peuvent être transformés en fonction $\psi_\mu \in L^2([0,1]^2)$ en définissant

$$\tilde{\psi}_\mu = \sum_n \psi_\mu[n] \psi_m^k \dots\dots(1,14)$$

On peut ainsi transformer un ensemble $\{B(\Gamma_j^k)\}_{k,j \geq 0}$ de bases de bandelettes discrètes en une unique base de fonctions de $L^2([0,1]^2)$ définie par

$$B(\Gamma) \stackrel{\text{def}}{=} \bigcup_{k,j \geq 0} \{\tilde{\psi}_\mu \mid \psi_\mu \in B(\Gamma_j^k)\} \quad \text{où} \quad \Gamma \stackrel{\text{def}}{=} \bigcup_{k,j \geq 0} \Gamma_j^k$$

4.2. Construction du dictionnaire de bandelettes

Quantification de la géométrie. Soit T le seuil en dessous duquel les coefficients en bandelettes vont être rejetés pour l'approximation. Pour réduire le nombre de bases de bandelettes potentielles, on ne va considérer qu'un petit nombre de géométries quantifiées à l'intérieur de chaque carré S du quadtree. De façon plus précise, à partir d'une paramétrisation $\tilde{\gamma}_S$, on définit une géométrie quantifiée $\tilde{\gamma}_{T^2}$ avec une précision de T^2 .

$$\tilde{\gamma}_{T^2} \stackrel{\text{def}}{=} \sum_{i=0}^{T-1} Q_{T^2}((\gamma, \theta_i) \theta_i) \dots\dots(1,15)$$

Où $\{\theta_i\}$ est la base de polynômes de Legendre. Nous avons utilisée le quantificateur

$$Q_\tau(x) = q\tau, \text{ si } (q-1/2)\tau \leq x \leq (q+1/2)\tau \dots\dots(1,16)$$

Pour un pas de quantification $\tau = T^2$. En restreignant l'amplitude des coefficients (γ, θ) ,

on obtient un nombre fini de géométries potentielles

$$\mathcal{G}_{T^2}(\mathbf{x}) \stackrel{\text{def}}{=} \left\{ \tilde{\gamma}_{T^2} \setminus \gamma \text{ géométrie telle que } \forall i, |\langle \gamma, \theta \rangle| \leq A_\theta T^{\frac{3\lambda}{2}} \right\} \cup \{\mathcal{E}\} \dots \dots (1,17)$$

Où A_θ est une constante définie par

$$A_\theta \stackrel{\text{def}}{=} 2b^{1/2} \max_{i < p} \|\theta_i\|_q \dots \dots (1,18)$$

On note que l'on a ajouté un symbole \mathcal{E} à cet ensemble de géométrie dans le but de prendre en compte d'une manière transparente le cas où le carré S ne contient pas de géométrie. Dans ce cas, on fixe $\mathbf{B}(S, \mathcal{E}) \stackrel{\text{def}}{=} \Delta_j(S)$ comme étant la base de Dirac définie par

$$\Delta_j(S) \stackrel{\text{def}}{=} \{\delta_n\}_{2^J n \in S} \quad \text{où} \quad \delta_n[m] = \begin{cases} 0 & \text{si } n \neq m \\ 1 & \text{si } n = m \end{cases} \dots \dots (1,19)$$

Construction du dictionnaire. Soit 2^J une échelle et k une orientation fixée. Le dictionnaire \mathbf{D}_j^k de bandelettes discrètes est composé des bases $\mathbf{B}(\Gamma_j^k)$ obtenues à l'aide de toutes les segmentations possibles en quadtree \mathbf{Q}_j^k et de toutes les géométries approchées $\{\tilde{\gamma}_s\}$ à l'intérieur des carrés de la segmentation.

En forçant les géométries $\tilde{\gamma}_s$ à appartenir à $\mathcal{G}_{T^2}(S)$, on définit un dictionnaire fini \mathbf{D}_{j,T^2}^k de bases de bandelettes discrètes

$$\mathbf{D}_{j,T^2}^k \stackrel{\text{def}}{=} \{ \mathbf{B}(\Gamma_j^k) \in \mathbf{D}_j^k \mid \Gamma_j^k = (\mathbf{Q}_j^k, \{\tilde{\gamma}_s\}) \text{ et } \forall s \in \mathbf{Q}_j^k, \tilde{\gamma}_s \in \mathcal{G}_{T^2}(S) \} \dots \dots (1,20)$$

En utilisant l'équation (1.13), on peut transformer un ensemble $\{\mathbf{B}(\Gamma_j^k)\}_{k,j \leq 0}$ de bases de bandelettes discrètes en une unique base de bandelettes de $\mathbf{L}^2([0,1]^2)$. L'union de toutes ces bases forme un dictionnaire de base de $\mathbf{L}^2([0,1]^2)$. Pour obtenir un dictionnaire de taille finie, on impose de plus que, pour toutes les échelles $2^J < T^2$, on ait $\mathbf{B}(\Gamma_j^k, T) = \Delta_j([0,1]^2)$, où $\Delta_j([0,1]^2)$ est la base de Dirac définie à l'équation (1.19). Le dictionnaire résultant \mathbf{D}_{T^2} est ainsi défini par

$$\mathbf{D}_{T^2}^k \stackrel{\text{def}}{=} \begin{cases} \mathbf{B}(\Gamma) & \forall 2^J < T^2, \mathbf{B}(\Gamma_j^k) = \Delta_j([0,1]^2) \\ & \forall 2^J \geq T^2, \mathbf{B}(\Gamma_j^k) \in \mathbf{D}_{j,T^2}^k \end{cases} \dots \dots (1,21)$$

1.4 Bandelettes pour la compression d'images

1.4.1 Compression

La compression d'image est une application immédiate des bases de bandelettes. Tout comme pour la transformée en ondelettes, ceci nécessite de quantifier les coefficients de bandelettes et d'optimiser le codage binaire de ces coefficients ainsi que les paramètres de la géométrie. Soit \mathbf{R} le nombre total de bits d'un tel

codage . on note par $D_b(R) = \|f - f_M\|^2$ l'erreur obtenue entre l'image f originale et l'image restituée f_R à partir de bandelettes [3].

1.4.2.Codeur en bandelettes orthogonales

Le codeur en ondelettes réalise la compression d'une image discrétisée \bar{f} de la manière suivante :

- pour chaque échelle 2^j et chaque orientation k , on subdivise en carrés S_M de $4*4$ les coefficients \bar{f}_j^k de la transformée en ondelettes.
- L'algorithme calcule le meilleur groupement $G^{(d)}$ tel que la transformée des coefficients f_S dans la base $B (G^{(d)}, S)$ minimise le lagrangien et stock $G_j^k[m]=d$ l'indice de ce groupement (en posant $G_j^k[m]=\infty$ si il n'y a pas de géométrie convenable).
- L'algorithme stocke en même temps les coefficients de la transformée en bandelettes.
- Ces coefficients sont quantifiés à l'aide du quantificateur uniforme décrit à l'équation (1.16).
- Ces coefficients quantifiés sont codés à l'aide d'un codeur arithmétique, ce qui produit un flot de bits de coefficients de bandelettes.
- Les cartes de géométries $G_j^k[m]$ sont également codées à l'aide d'un codeur arithmétique, ce qui crée un flot de bits de coefficients de géométrie [1].

1.5 Conclusion

La géométrie utilisée dans la construction des bandelettes est basée non pas sur la notion de contours mais sur celle d'un champ de vecteurs indiquant une direction de régularité locale des images. ce flot géométrique est simplifié par une segmentation adaptative en carré et des conditions de parallélisme.

La représentation en bandelettes représente efficacement les contours des images grâce aux bases orthogonales de bandelettes ainsi que les zones de régularité homogène sous la forme d'un résidu décomposée en ondelettes.

Mais nous avons vu que les bandelettes ne sont pas optimale pour les images de nature géométrique, alors l'optimisation de cette géométrie pour une application donnée s'obtient par les ondelettes d'Alpert [1] conduit à une amélioration par rapport aux bandelettes, et c'est ce que nous allons expliquer dans le deuxième chapitre.

Chapitre 2

Transformée en bandelettes discrète

2.1. Introduction

Dans ce chapitre nous présentons l'algorithme de transformée en bandelettes rapide d'une image discrétisée et la méthode de transformée d'Alpert qui est une des nombreux travaux qui étudient la bonne représentation de la géométrie des images.

2.2. Transformée en bandelettes discrète

2.2.1 Image discrétisée

Une image discrétisée \bar{f} de $N \times N$ pixels est obtenue en projetant une fonction $f \in L^2([0,1]^2)$ sur un ensemble de fonctions d'échelles $\{\Phi_{jn}\}_n$ à une résolution $2^j = N^{-1}$

$$\forall n \in \{0, \dots, N-1\}^2, \bar{f}[n] \stackrel{\text{def}}{=} \langle f, \Phi_{jn} \rangle$$

Où

$$\Phi_{jn}(x) \stackrel{\text{def}}{=} 2^{-j} \Phi(2^{-j} x - n).$$

2.2.2 Fonction d'échelles en ondelettes

Les espaces intermédiaires $V_j \subset L^2([0,1]^2)$ peuvent se décomposer comme des sommes d'espace de détails en ondelettes

$$V_j = \bigoplus_{k,p < j} W_j^k \quad \text{et} \quad V_j = \text{vect} \{ \varphi_{jn} \mid n \in \{0, \dots, 2^j - 1\}^2 \}$$

Où

$$\varphi_{jn} \stackrel{\text{def}}{=} 2^{-j} \varphi(2^{-j} x - n) \quad \dots\dots(2,1)$$

Les fonctions d'ondelettes et les fonctions d'échelle peuvent s'écrire à l'aide d'une cascade de filtrage passe-bas $\{g^k[n]\}_n$ et passe-haut $\{h^k[n]\}_n$, ce qui s'écrit dans le domaine de Fourier

$$\widehat{\psi}_j^k(\omega) = \prod_{p \leq j} \widehat{h}^k(2^p \omega) \quad \text{et} \quad \widehat{\varphi}_j^k(\omega) = \widehat{g}^k(2^j \omega) \prod_{p \leq j-1} \widehat{h}^k(2^p \omega)$$

2.2.3 Base de bandelettes discrètes

L'équation (2,1) transforme des vecteurs des bandelettes discrètes $\psi_\mu \in \mathcal{B}(\Gamma_j^k)$ de ℓ_j^2 en un ensemble de fonctions $\overline{\psi}_\mu$. Pour $j > J$, on peut aussi transformer ces vecteurs en des vecteurs $\overline{\psi}_\mu$ de l'espace de pixels de l'image ℓ_j^2 comme suit

$$\overline{\mathcal{B}}(\Gamma_j^k) \stackrel{\text{def}}{=} \{\overline{\psi}_\mu\}_\mu \quad \text{où} \quad \overline{\psi}_\mu \stackrel{\text{def}}{=} \sum_{(z',n) \in [0,1]^2} \psi_\mu[n] \overline{\psi}_{jn}^k$$

Où les vecteurs d'ondelettes discrètes $\overline{\psi}_{jn}^k$ sont définis dans le domaine de Fourier

$$\overline{\psi}_{jn}^k(\omega) \stackrel{\text{def}}{=} \prod_{p=j+1}^j \overline{h}^k(\omega).$$

Une base discrète de bandelettes est définie par

$$\overline{\mathcal{B}}(\Gamma) \stackrel{\text{def}}{=} \bigcup_{k,j>J} \overline{\mathcal{B}}(\Gamma_j^k) \quad \text{où} \quad \Gamma = \bigcup_{k,j>J} \Gamma_j^k.$$

L'ensemble de toutes les bases possibles de bandelettes discrètes $\overline{\mathcal{B}}(\Gamma)$ pour tous les ensembles de bases $\overline{\mathcal{B}}(\Gamma_j^k) \in \mathcal{D}_{j,\Gamma^2}^k, j > J$, forme le dictionnaire de bandelettes discrètes $\overline{\mathcal{D}}_{\Gamma^2}$.

2.2.4 Transformée en bandelettes discrètes

Le calcul d'une transformée en bandelettes discrètes d'une image \overline{f} correspond à la projection de cette image sur les vecteurs d'une base $\overline{\mathcal{B}}(\Gamma)$. Cette transformée est calculée à l'aide tout d'abord d'une transformée en ondelettes discrète rapide de l'image $\overline{f} \in \ell_j^2$ puis d'une transformée en bandelettes discrète de l'ensemble de coefficients \overline{f}_j^k pour chaque échelle $2^j > 2^J$ et orientation k .

Lorsque les fonctions d'échelles $\overline{\psi}_{jn}$ sont en fait des fonctions d'échelles de la transformée en ondelettes $\overline{\psi}_j^k$ comme défini à l'équation (2,1), on a

$$\forall k, \quad \forall j > J, \quad \langle \overline{f}, \overline{\psi}_\mu \rangle_{\ell_j^2} = \langle \overline{f}, \overline{\psi}_\mu \rangle_{L^2} = \langle \overline{f}_j^k, \psi_\mu \rangle_{\ell_j^2}.$$

En générale lorsque Φ_{j_m} la fonction d'échelle est C^β à support compact, alors les coefficients en bandelettes $(\bar{f}, \bar{\Psi}_\mu)$ et les coefficients $(\bar{f}, \bar{\Psi}_\mu)$ sont proches. On peut ainsi énoncer le résultat d'approximation discrète suivant.

Théorème 2.1 Soit $T > 0$ et soit \bar{f} une image de $N \times N$ pixels discrétisée depuis une fonction f ayant une régularité géométrique C^α . Soit $\bar{\mathcal{B}}^*(\Gamma)$ la base de bandelettes discrète défini par

$$\bar{\mathcal{B}}^*(\Gamma) \stackrel{\text{def}}{=} \text{argmin} \left(\mathcal{L}(\bar{f}, \bar{\mathcal{B}}(\Gamma), T) \right).$$

$$\bar{\mathcal{B}}(\Gamma) \in \bar{\mathcal{D}}_{T^2}$$

L'approximation \bar{f}_M de \bar{f} par un seuillage à T dans la base $\bar{\mathcal{B}}^*(\Gamma)$ vérifie

$$\|\bar{f} - \bar{f}_M\|_{\ell_2^2} \leq CM^{-\alpha}$$

Où $M = M_B + M_G + M_S$ inclut le nombre de M_B de coefficients en bandelettes au-dessus de T , le nombre M_S de coefficients nécessaires pour spécifier les quadrees, le nombre M_G de coefficients nécessaires pour spécifier les géométries dans chaque carré des quadrees et C est une constante qui ne dépend que de f .

2.3. ALGORITHME RAPIDE

L'algorithme de transformée en bandelettes rapide direct inclut le calcul de la meilleure base de bandelettes $\bar{\mathcal{B}}^*(\Gamma)$ ainsi que le calcul de la décomposition de \bar{f} dans cette base. L'algorithme inverse consiste à inverser les transformées en ondelettes ainsi que les transformée de Alpert sur les carrés des quadrees.

- (1) **Paramètre de l'algorithme.** En utilisant une image discrétisée \bar{f} de taille $N \times N$ pixels, et un seuil T qui contrôle le taux de compression de l'algorithme. On note $2^j \stackrel{\text{def}}{=} N^{-1}$.
- (2) **Transformée en ondelettes 2D.** On calcule la transformée en ondelettes discrète 2D de l'image d'origine \bar{f} . ceci produit un ensemble de vecteurs de coefficients $\{\bar{f}_j^k\}_{k,j \geq j}$. Les vecteurs \bar{f}_j^k , pour chaque échelle 2^j et orientation $k \in \{V, H, D\}$ peuvent être stockés dans une unique image ayant la même taille que l'image d'origine \bar{f} . les étapes suivantes (3)-(7) implémente la bandelétisation, qui est répétée pour chaque échelle e orientation.

(3) Sélection de chaque carré dyadique. Un carré est par définition un carré obtenue en subdivisant de façon récursive le carré $[0,1]^2$ en quatre carrés de même taille. Pour chaque carré S, on regroupe les coefficients en ondelettes $\{F^k[n]\}_{(2^j n) \in S}$ dans un vecteur noté f_S . les étapes suivantes (4)-(7) sont répétées pour chaque carré dyadique S à chaque échelle 2^j et orientation k d'une transformée en ondelettes.

(4) Quantification de la géométrie. En essayant de trouver la meilleure géométrie approchée quantifiée $\tilde{\gamma}_s$ à l'intérieur d'un carré S. on testons toutes les bases de bandelettes $\mathcal{B}(S, \tilde{\gamma}_m^\Omega)$ de \mathcal{L}_j^2 pour les géométries

$$\tilde{\gamma}_m^\Omega \stackrel{\text{def}}{=} \sum_{i=0}^{p-1} (m_i T^2) \theta_i \quad \text{où} \quad |m_i| \leq A_\theta T^2 \quad \text{et} \quad \Omega \in \{V, H\}$$

Pour chaque ensemble possible de coefficients $m \stackrel{\text{def}}{=} \{m_i\}_i$ et orientation Ω . nous ajoutons aussi à ces géométries le symbole spécial $\tilde{\gamma} = \Xi$. Une géométrie horizontale $\tilde{\gamma}_m^H$ (resp. une géométrie verticale $\tilde{\gamma}_m^V$) paramétrise une courbe approchée horizontale $(x, \tilde{\gamma}_m^H(x))$ (resp. une courbe approchée verticale $(y, \tilde{\gamma}_m^V(y))$).

L'étape suivante (5) est répétée pour chaque géométrie potentielle $\tilde{\gamma}_m^\Omega$.

(5) Calcul de la transformée de Alpert. Pour une géométrie donnée $\tilde{\gamma} = \tilde{\gamma}_m^\Omega$, nous calculons l'ensemble des produits scalaires

$$A_{\tilde{\gamma}} = (f_S)[\ell, i] \stackrel{\text{def}}{=} \langle f_S, \Psi_{\ell i} \rangle \quad \text{où} \quad \mathcal{B}(S, \tilde{\gamma}) \stackrel{\text{def}}{=} \{\Psi_{\ell i}\}_{\ell, i}$$

Cette transformée $A_{\tilde{\gamma}}$ est calculée à l'aide d'un algorithme rapide comme expliqué à dans [1]. pour le symbole spécial $\tilde{\gamma} = \Xi$, la transformée correspondante A_Ξ est l'identité.

(6) Sélection de la meilleure géométrie. Pour un seuil donné T, nous devons choisir la meilleur géométrie $\tilde{\gamma} = \tilde{\gamma}_m^\Omega$ ou $\tilde{\gamma} = \Xi$ qui minimise le lagrangien

$$\mathcal{L}(f_S, \mathcal{B}(S, \tilde{\gamma}), T) \stackrel{\text{def}}{=} \sum_{(\ell, i) \in J_T} |A_{\tilde{\gamma}}(f_S)[\ell, i]|^2 + (\text{Card}(J_T) + M_C) T^2$$

Où

$$J_T \stackrel{\text{def}}{=} \{(\ell, i) \mid |A_{\tilde{\gamma}}(f_S)[\ell, i]| > T\}$$

Et $M_C = 1$ si $\tilde{\gamma} = \Xi$ et $M_C = p + 1$ sinon on peut ainsi définir la géométrie sélectionnée pour le carré S par

$$\tilde{\gamma}_S \stackrel{\text{def}}{=} \underset{\tilde{\gamma}_m^A}{\operatorname{argmin}} (\mathcal{L}(f_S, \mathcal{B}(S, \tilde{\gamma}_m^A), T))$$

(7) Construction du quadtree. Après avoir calculé les coefficients transformée $A_{\tilde{\gamma}_S}(f_S)$ pour chaque carré dyadique, on doit choisir la meilleure segmentation en carré. Ceci revient à conserver uniquement un sous-ensemble de carrés dyadiques de façon à obtenir une segmentation en quadtree. à partir de la segmentation complète en petits carrés, suivi d'une phase de regroupement par quatre des carrés. Grâce à l'additivité du lagrangien et à la structure hiérarchique du quadtree, la minimisation de \mathcal{L} s'effectue à l'aide d'un algorithme a été introduit par Donoho dans [14] et a déjà été utilisé par Lepennec et Mallat dans [12]. Il constitue un cas particulier de l'algorithme CART d'estimation statistique, introduit par Breiman et al.[13].

A l'étape précédente (6), nous avons enregistré, pour chaque carré dyadique S la valeur $\mathcal{L}(S) \stackrel{\text{def}}{=} \mathcal{L}(f_S, \mathcal{B}(S, \tilde{\gamma}_S), T)$ du lagrangien restreint à S, de même que la meilleure géométrie quantifiée $\tilde{\gamma}_S$. Alors, pour chaque échelle 2^j et orientation k, on calcule la structure de quadtree à l'aide de l'algorithme suivant

- Initialisation du quadtree : chaque carré S de taille $b=2^j$ est une feuille. Enregistrer les géométries optimales $\tilde{\gamma}_S$ et initialiser \mathcal{L}_0 , le lagrangien cumulatif du sous-arbre à $\mathcal{L}_0(S) = \mathcal{L}(S)$.
- Commencer avec des carrés S de taille $b=2 \times 2^j$.
- Pour chaque carré S, on note (S_1, S_2, S_3, S_4) ses quatre sous-carrés et

$$\mathcal{L}'(S) \stackrel{\text{def}}{=} \mathcal{L}_0(S_1) + \mathcal{L}_0(S_2) + \mathcal{L}_0(S_3) + \mathcal{L}_0(S_4) + T^2$$

Le lagrangien du sous-arbre (le terme additionnel T^2 est dû au coût de un coefficient pour la subdivision). Les sous-carrés doivent être rassemblés si $\mathcal{L}(S) < \mathcal{L}'(S)$. Si c'est le cas, déclarer S comme une feuille, enregistrer la géométrie optimale $\tilde{\gamma}_S$. Mettre à jour

$$\mathcal{L}_0(S) = \min(\mathcal{L}(S), \mathcal{L}'(S)).$$

-Tans que $b < 1$, faire $b \leftarrow 2 \times b$ et répéter l'étape précédente.

(8) **Fin de l'algorithme.** Dans chaque carré S du quadtree, nous avons enregistré les coefficients de la transformée de Alpert $A_{\mathcal{F}_S}(f_S)$. Comme la transformée de Alpert est une transformée orthogonale des coefficients d'origine f_S , nous pouvons les enregistrer au même endroit. L'ensemble de tous les coefficients en bandelettes peuvent être stockés dans une image de taille $N \times N$, tout comme une transformée en ondelettes. Il faut aussi garder trace des structures de quadtree, ainsi que les paramètres (m, Ω) ou Ξ des géométries optimales.

Complexité numérique Pour une image de $N \times N$ pixels, la complexité numérique de la transformée en ondelettes discrètes est de l'ordre de $O(N^2)$. Pour chaque échelle 2^j , chaque orientation k et pour chaque taille dyadique b, la complexité du calcul de la transformée de Alpert est de

$$C(j, b) = \underbrace{(1/b)^2}_{\text{nbr.carrés}} \times \underbrace{C_A(2^j b)^2}_{\text{complexité Alpert tr}} \times \underbrace{C_G T^{-\eta}}_{\text{nbr.géométries}} = C 2^{-2j} T^{-\eta}.$$

Comme le nombre de telles tailles de carrés, d'échelles 2^j et d'orientation k est proportionnel à $|\log_2(T)|$ et que $\eta < p-1$, la complexité globale de la transformée en bandelettes est $O(N^2 T^{-(p-1)})$ où p est le nombre de moments nuls de la transformée.

Bien que polynomiale en N t en T^{-1} , la complexité d'une recherche exhaustive des géométries est trop grande pour une image. Une implémentation typique (voir les chapitres 3 et 4) utilise des calculs heuristiques pour réduire le nombre de géométries potentielles et restreint les carrés à des tailles 4×2^j et 8×2^j seulement. Le chapitre 3 applique l'algorithme de transformée en bandelettes à la compression d'image et explique la modification faite à la transformée de régime pour la rendre plus efficace.

2.4. Transformée D'Alpert 2D

Dans cette sous-section, nous décrivons l'implémentation de la transformée de Alpert rapide. Rappelons que pour implémenter la transformée en bandelettes rapide, l'étape (5) de l'algorithme décrit à la section (2.3) nécessite le calcul de la transformée $A_{\mathcal{F}}(g)$ où $g = f_S$ est un ensemble de coefficients discrets et $S \subset [0,1]^2$ est un carrée dyadique.

On rappelle que P est le nombre de moments nuls que l'on souhaite obtenir pour les vecteurs de bandelettes discrètes. Dans cette section, on définit le degré d'un polynôme en deux variables comme le maximum des degrés selon chacun des axes. Ceci va nous amener à considérer des groupes de $N_p \stackrel{\text{def}}{=} p^2$ ondelettes d'Alpert. Il est possible d'utiliser une autre convention, ce qui nécessite l'utilisation de

$$N_p \stackrel{\text{def}}{=} \text{Card}\{(a_1, a_2) \mid \deg(X^a) < p\}$$

On suppose donc donné en rentrée de l'algorithme un ensemble de coefficients $\{g[n]\}_{n=0}^{N-1}$ qui sont des échantillons d'une certaine fonction $\{g[n]\} = g(x_n)$ où $\mathcal{U} \triangleq \{x_n\}_n \subset [0,1]^2$. Pour la transformée en bandelettes, on a $g = f_S$ et $x_n = 2^j n$. On suppose aussi donnée une courbe $\tilde{\gamma}$ qui va paramétrer la transformée de Alpert. Dans la suite, on note $\ell^2(\mathcal{U})$ l'espace des vecteurs indexés par les $x_n \in \mathcal{U}$ qui dans le cas de la transformée en bandelettes, s'identifie avec $\ell^2(S)$.

2.4.1. Groupements dyadiques de points la première étape de l'algorithme consiste à subdiviser l'ensemble des points \mathcal{U} de façon récursive en deux moitiés égales (où presque égales si le cardinal d'un ensemble intermédiaire est impair). Ainsi pour chaque échelle $\ell \leq 0$ de la transformée, on dispose d'une partition de \mathcal{U} en 2^ℓ sous-ensembles

$$\mathcal{U} = \bigcup_{i=0}^{2^\ell-1} \mathcal{U}_\ell^i \quad \text{Avec} \quad \mathcal{U}_\ell^i = \mathcal{U}_{\ell-1}^{2i} \cup \mathcal{U}_{\ell-1}^{2i+1} \dots (2.2)$$

Dans le cas qui nous intéresse, cette subdivision est calculée à l'aide de la courbe $\tilde{\gamma}$, que l'on suppose paramétrée horizontalement par $(x_1, \gamma(x_1))$. On rappelle que l'on a défini un opérateur de déformation ainsi que des positions déformées

$$\omega(x) \triangleq (x_1, x_2 - \gamma x_1) \quad \text{Et} \quad \tilde{x}_n = (\tilde{x}_{1n}, \tilde{x}_{2n}) \triangleq \omega(x_n)$$

On pose initialement $\mathcal{U}_0^0 = \mathcal{U}$. La subdivision $\mathcal{U}_\ell^i = \mathcal{U}_{\ell-1}^{2i} \cup \mathcal{U}_{\ell-1}^{2i+1}$ s'obtient en classant les ordonnées \tilde{x}_{2n} par ordre croissant et en plaçant la première moitié des points dans $\mathcal{U}_{\ell-1}^{2i}$ et la deuxième dans $\mathcal{U}_{\ell-1}^{2i+1}$. La figure 2.1 illustre ce procédé.

On arrête la subdivision à l'échelle ℓ_0 la plus petite satisfaisant

$$\forall i, \quad N_p/2 < \text{Card}(\mathcal{U}_{\ell_0}^i) \leq N_p.$$

On constate que les points x_n se répartissent dans des bandes b_{ℓ_i} .

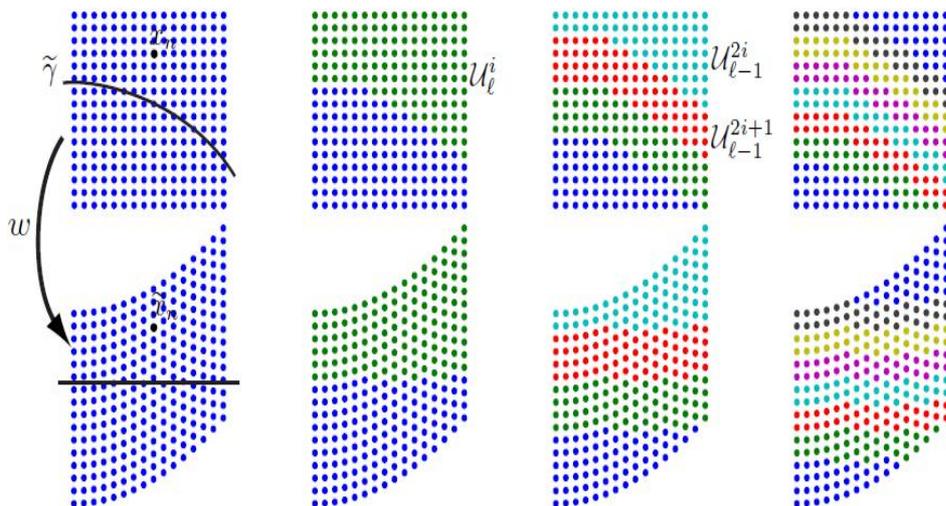


Figure 2.1 : Exemple de regroupements dyadiques suivant une courbe géométrique

2.4.2. Espaces de multi résolution polynomiaux Dans la suite, si $X^a = X_1^{a_1} X_2^{a_2}$ est un monôme, on note de même la fonction $X^a : x \mapsto (x_1)^{a_1} (x_2)^{a_2}$. de plus, si g est une fonction continue, on note $[g]_{\ell}^i$ le vecteur discret échantillonné et restreint aux points de \mathcal{U}_{ℓ}^i , c'est-à-dire, pour $\ell > \ell_0$

$$\forall n, \quad [g]_{\ell}^i[n] \stackrel{\text{def}}{=} \begin{cases} g(x_n) & \text{si } x_n \in \mathcal{U}_{\ell}^i, \\ 0 & \text{sinon.} \end{cases}$$

Comme il est expliqué dans [1], on souhaite projeter le vecteur $g \in \ell^2(\mathcal{U})$ sur des espace multi résolution

$$G_{\ell} \stackrel{\text{def}}{=} \left\{ \begin{array}{l} \varphi \in \ell^2(\mathcal{U}) \quad \setminus \quad \forall x_n \in \mathcal{U}_{\ell}^i, \quad \varphi[n] = P_i(\tilde{x}_n) \\ \text{et } P_i \text{ est polynomiale de degré inférieur à } p-1 \end{array} \right\}$$

Ces espaces satisfont $G_{\ell} \subset G_{\ell-1}$ puisque les ensembles \mathcal{U}_{ℓ}^i satisfont la propriété d'emboîtement (2.2). On note de plus que $G_0 = \ell^2(\mathcal{U})$.

La base orthogonale d'Alpert est alors définie comme une base orthogonale $\{\Psi_{\ell i}\}_{i=0}^{m_{\ell}-1}$ de l'espace de détails H_{ℓ} :

$$\text{Vect}_i(\Psi_{\ell i}) = H_{\ell} \quad \text{où} \quad H_{\ell} \oplus^{\perp} G_{\ell} = G_{\ell-1}$$

On constate que l'on a les décompositions orthogonales

$$G_{\ell} = \bigoplus_{i=0}^{2^{\ell-1}-1} G_{\ell}^i \quad \text{et} \quad H_{\ell} = \bigoplus_{i=0}^{2^{\ell-1}-1} H_{\ell}^i$$

Où G_{ℓ}^i (resp. H_{ℓ}^i) est formée par l'ensemble des vecteurs de G_{ℓ} (resp. H_{ℓ}) nuls en dehors de G_{ℓ}^i . on remarque que l'on a

$$\forall \ell > \ell_0 + 1, \quad \dim(H_{\ell}) = 2^{-\ell} N_p \quad \text{et} \quad \dim(H_{\ell}^i) = N_p,$$

Et que $\dim(H_{\ell_0}^i)$ peut éventuellement être plus petit que N_p .

2.4.3. Construction de la base d'Alpert Notre but est de construire de façon algorithmique une base orthogonale de H_{ℓ}^i pour chaque ℓ et chaque i . Pour simplifier on suppose que $\ell > \ell_0 + 1$ (pour le cas initial $\ell = \ell_0 + 1$, il faut prendre en compte le fait que l'on peut avoir $\text{Card}(\mathcal{U}_{\ell}^i) < N_p$).

On constate que l'on dispose d'une base de G_{ℓ}^i :

$$\mathcal{B}(G_{\ell}^i) \stackrel{\text{def}}{=} \{[X^a]_{\ell}^i \mid \deg(X^a) < p\}.$$

On peut compléter cette base à l'aide d'une base de H_{ℓ}^i :

$$\mathcal{B}(H_{\ell}^i) \stackrel{\text{def}}{=} \{[X^a]_{\ell-1}^{2i} - [X^a]_{\ell-1}^{2i+1} \mid \deg(X^a) < p\}$$

On constate que l'on obtient bien une base de $G_{\ell-1}$ en réunissant toutes les bases ainsi obtenues puisque

$$\text{Vect}(\mathcal{B}_\ell^i) \stackrel{\text{def}}{=} G_{\ell-1}^{2i} \oplus G_{\ell-1}^{2i+1} \quad \text{Avec} \quad \mathcal{B}_\ell^i \stackrel{\text{def}}{=} \mathcal{B}(G_\ell^i) \cup \mathcal{B}(H_\ell^i)$$

Pour construire une base orthonormée de H_ℓ^i qui est orthogonale à G_ℓ^i , il suffit d'appliquer le procédé d'orthogonalisation de Gram-Schmidt à la famille \mathcal{B}_ℓ^i pour obtenir une famille orthogonale $\tilde{\mathcal{B}}_\ell^i$. L'orthogonalisation gardant l'ordre des vecteurs, les N_p premiers vecteurs de $\tilde{\mathcal{B}}_\ell^i$ fournissent une base orthonormée de G_ℓ^i et les N_p derniers vecteurs fournissent une base orthonormée de H_ℓ^i : ce sont les fonctions multi-ondelettes Ψ_ℓ^i pour $k=0, \dots, N_p - 1$.

On pourra noter qu'il n'y a bien sur pas unicité d'une telle famille de fonctions d'ondelettes. dans la construction originale, en dimension 1, Alpert propose de fixer ces fonctions en imposant que la fonction $\Psi_{\ell_i}^{(k)}$ ait en fait $p+k-1$ moments nuls. Ceci revient à utiliser une base $\mathcal{B}(H_\ell^i)$ légèrement différente (composée de vecteurs de degrés supérieurs à p).

L'algorithme de calcul que l'on vient de décrire nécessite à chaque échelle $2^{-\ell}$ orthogonalisations de Gram-Schmidt sur $2N_p$ vecteurs de taille $N2^\ell$, ce qui demande $(2N_p)^2 N2^\ell$ opérations. Ainsi la complexité totale de l'algorithme est de

$$\sum_{\ell=-\log_2(N)}^0 2^{-\ell} ((2N_p)^2 N2^\ell) = O(N \log_2(N)).$$

2.5. Conclusion

A la recherche de nouvelle méthode pour l'amélioration de la représentation des images de nature géométriques, et a travers des recherches on a trouvez que la transformée de Alpert et son application vont nous donnés des bons résultats peut-on dire optimale avec la bonne prise on main et l'exploitation de l'algorithme de cette dernière, on va récolter des résultats plus optimale que les bandelettes et ces ce qu'on va vous expliquez dans le chapitre suivant.

Chapitre 3

Simulations et résultats

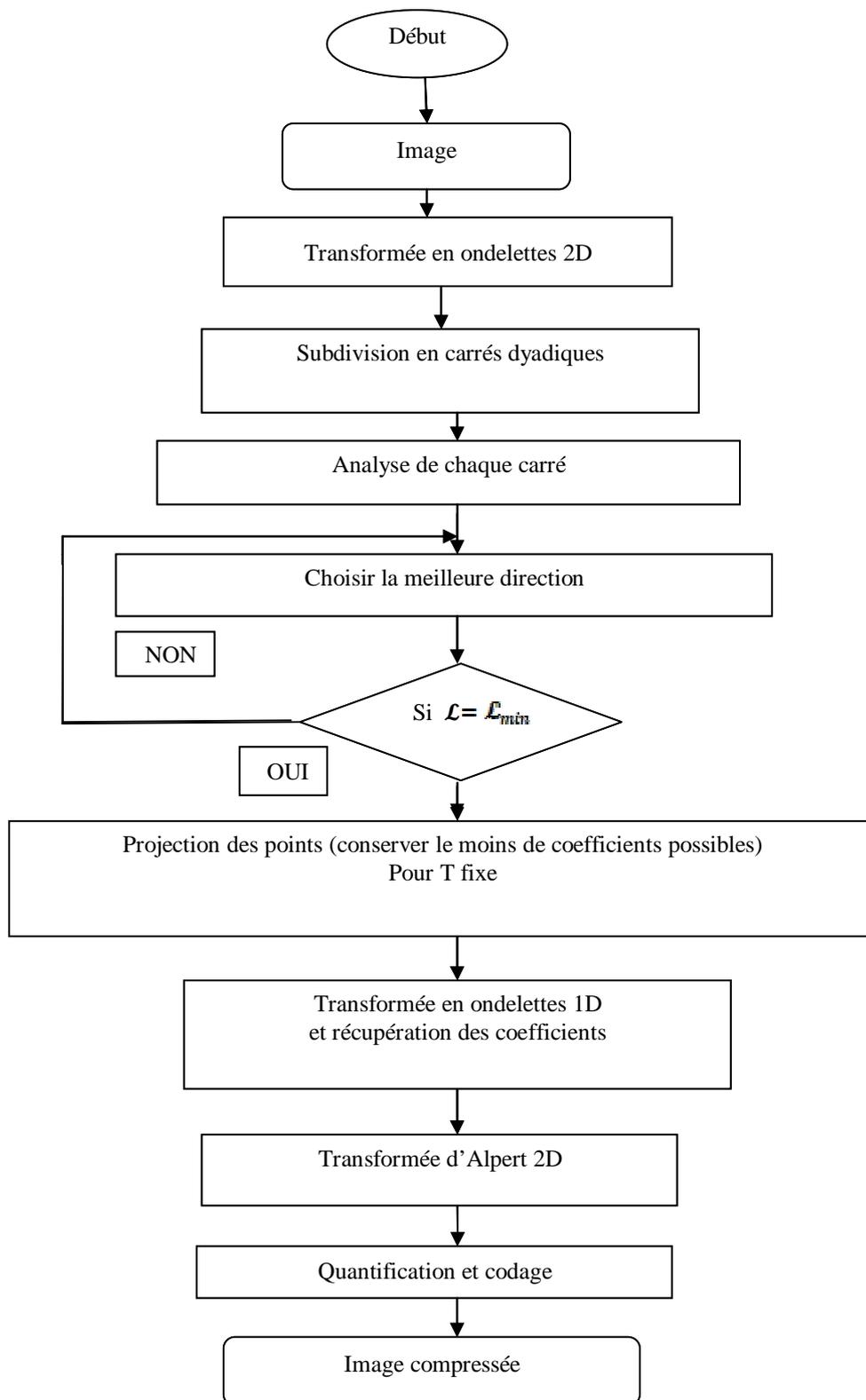


Figure : organigramme de la compression par bandelettes et transformée d'Alpert

3.1 Introduction

La deuxième génération des bandelettes [1] c'est une transformée orthogonale multi échelles qui est capable de capturer le contenu géométrique des images des surface le but de cette note c'est de donner quelques données sur la réalité implémentation matlab. Le code source accompagnateur (contient des fonctions décrites dans ce papier avec d'autre outils et textes écrits) .dans ce chapitre nous présentons l'algorithme générale de compression d'image par bandelettes et transformée d'Alpert sur lequel se base notre application, avec quelque exemples illustrant les images de test (image gaussienne, image objet), avec une étude comparative.

3.2 Structure du programme

Dans notre application pour la compression d'images avec les algorithmes géométriques, nous utilisons une transformée d'Alpert en plus des bandelettes, cette transformée a pour but de capturer la régularité géométriques le longs des contours. L'organigramme suivant représente les étapes à suivre pour la réalisation d'une compression utilisant la transformée d'Alpert :

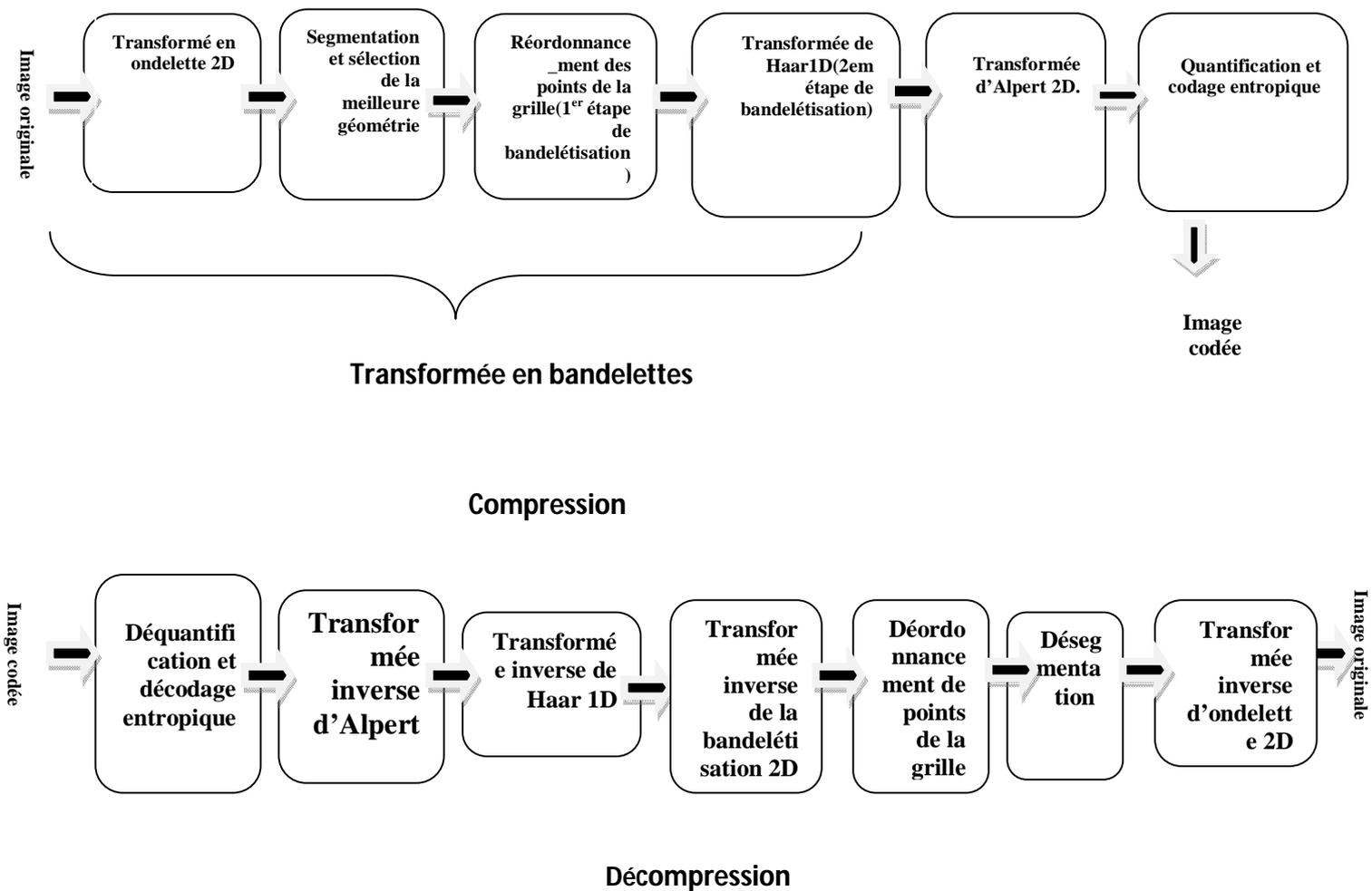


Figure 3 : Organigramme de la compression / décompression avec bandelette et Alpert.

3.3 Algorithme de transformée en bandelettes et d'Alpert

1-l'implémentation de la transformée a une échelle fixé : dans cette section on explique comment computer la transformée en bandelettes d'une image donnée sur une échelle fixe, c'est la clé ingrédient de tout l'algorithme d bandelettes ,qui utilise en premier une transformée en ondelettes et puis une transformée en bandelettes dans chaque échelle et orientation.la mono-résolution de la (bandelets transform) peu être utiliser directement sur l'image et donne bien de bon résultat, l'inconvénient essentiel de cette mono-résolution approchée c'est qu'elle produise un blocage artefact due au pièce constat nature des fonction de base. Dans la prochaine section on va montrer comment la (wavelet-bandelets transform) fixe le problème par performer l'approximation dans le domaine des ondelettes.

Warped Haar transform : en premier vous allez avoir besoin d'une transformée de Haar 1D ceci est un code essentiel et il doit être plus rapide que possible :1-c'est une simple implémentation dans matlab function (perform-warped-Haar). 2- appliquer la transformée de Haar dans une direction donnée spécifier par son angle θ Ceci correspond aux étapes (5)-(7) de l'algorithme rapide, le paramètre : dir c'est +1 pour la transformation direct et -1 pour l'inverse.

La meilleur sélection géométrique : pour trouver la meilleur direction dans un petit carré on a besoin de vérifier un ensemble de directions ceci est l'étape (8) de l'algorithme. La fonction (compute-best-direction) sélectionne un nombre de direction ensuite les mets dans θ et calcule le lagrangien pour chaque direction, ensuite va choisir la direction où le lagrangien est le plus petit.

La construction du Quadtree : le quadtree est calculé par la fonction (compute-quadtree) le quadtree est représenté en utilisant deux : 2D image (de même taille que l'image originale). L'image QT représente la profondeur du quadtree pour chaque pixels, notez que chaque QT et θ sont constant dans chaque carré du quadtree finale. Pour calculer ce quadtree en premier en doit calculer le lagrangien, pour chaque carré dyadique pour la petit taille (la taille de ces carrés doit être spécifique par le paramètre j_{min}) puis l'algorithme part de la petit taille a la grande taille, est essaye d'associer chaque groupe des quatre carrés . pour faire ainsi c'est simplement calculer le lagrangien du carré associé et le comparer a la somme des 4 lagrangien(plus 1 qui est le prix du divisement de l'arbre ceci est l'étape 10 de l'algorithme).dans la figure 3.1,on peut voir un quadtree calculer avec cette procédure pour $T=10$,on peut noter que c'est ainsi dans les carrés avec les paramètres géométriques, l'algorithme choisi quelques orientations arbitraires ceci c'est parce que dans ces carrés la fonction n'a pas une moyenne zéro, alors la transformée en bandelettes(avec n'importe quelle directions) est mieux que laisser les données intransformée ,cette situation n'apparait pas dans la wavelet-bandelet algorithme (voir2) depuis dans les espaces-plat la transformée en ondelettes à la moyenne zéro.

Bandelet transform : la transformée directe et inverse utilise exactement le même code Matlab par simplement mentionner dir soit à +1 ou -1 la fonction (perform-bandelet-transform) fait le travail par transformée chaque feuille du quadtree, cette fonction aussi retourne m.geom qui est le nombre des paramètres pour décrire la base de la bandelettes c'est la somme des coefficients géométriques (un coefficient par feuille

avec géométrie) et une description quadtree des coefficients (un bit par 1 :4 une règle de division qui peut être remplacé par approximativement 1/7 th du coefficient. Dans la figure 3.2 on peut voir la compression géométrique de l'image avec plusieurs seuil T. ceci montre clairement la limitation de l'application directe de la transformée en bandelettes dans une scène mono-résolution : artefacts de blocage sont clairement visibles à des débits faibles.

2-L'implémentation complète du schéma de bandelettes Pour performer un algorithme complet présenté dans [1] on doit appliquer la transformée en bandelettes présenté dans l'ex section dans chaque échelle et l'orientation de la transformée d'ondelettes de chaque image.

Computing the wavelet transform : vous allez avoir besoin d'une transformée en Ondelettes 2D comme celui implémenter dans wavelab [14]. La fonction (perform-wavelet-transform) est un simple enveloppe pour wavelab, mais vous pouvez le remplacer par votre propre ondelette orthogonale ou biorthogonale, vous devez juste respecter les mêmes données dans le plan.

Computing the quadtree for each scale : la fonction (compute-wavelet-quadtree) calcule un seul quadtree pour chaque échelle et pour chaque orientation. Partageant le même quadtree pour les trois orientations menant au peu d'augmentation d'efficacité. Pour le prix d'une implémentation complexe la structure du quadtree QT est la géométrie thêta sont mémorisé dans une image qui a la même taille que l'image originale M (et ils ont la même structure hiérarchique comme l'image transformée en ondelette MW).

Performing the wavelet-bandelet transform : comme pour la construction du quadtree, le calcul du (wavelet-bandelet transform) est directe, elle implique simplement une boucle pour chaque échelle et orientation, et appelle la fonction mono-résolution (perform-bandelet-transform), ceci est fait dans le scripte (perform-wav-band-transform). Comme en peut le voir dans la figure 3.4 le résultat de la compression en utilisant l'algorithme (wavelet-bandelet transform) sur une image géométrique.

Perform the Alpert 2D transform : la transformée de Alpert est une transformée multi-ondelettes sur la base de polynômes orthogonaux, elle a été originalement désigner pour la résolution des dérivées partielles et équations intégrales. l'algorithme de Alpert proposé peut transformée un signal (1D, 2D, nD) par une longueur arbitraire, et échantillonnage arbitraire (on doit chaque fois fournir des échantillons locaux dans le paramètre pos). Le nombre de moment nuls (qui est aussi le degré d'approximation polynomial+1) qui est posé par le paramètre K.

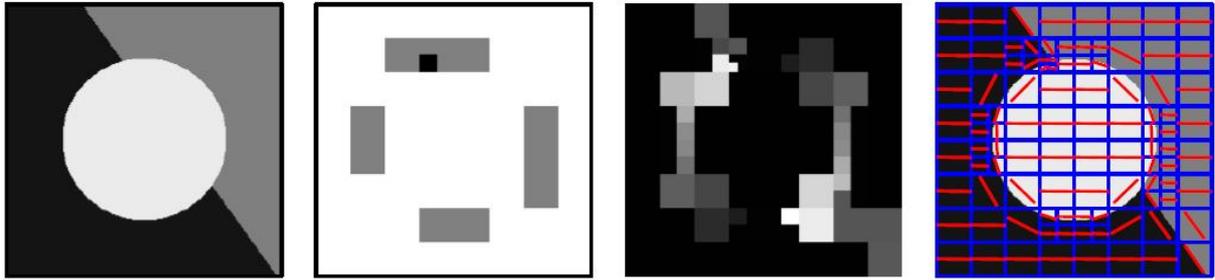


Figure3.1 : Un exemple de quadtree calculée en utilisant la fonction *compute-quadtree*.

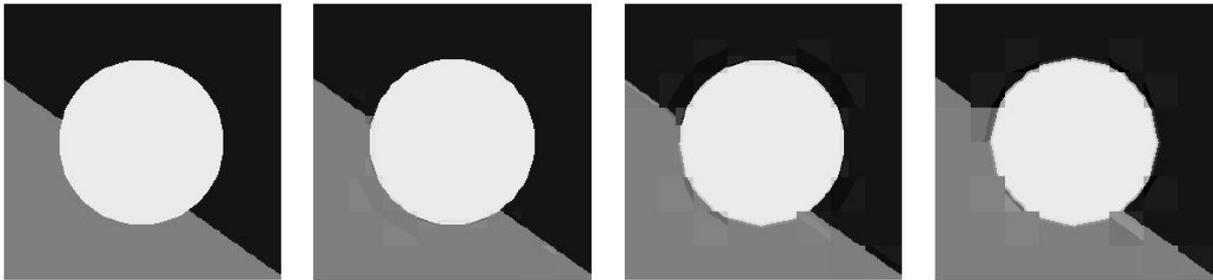


Figure3.2 : La compression d'image en utilisant mono-résolution bandelettes

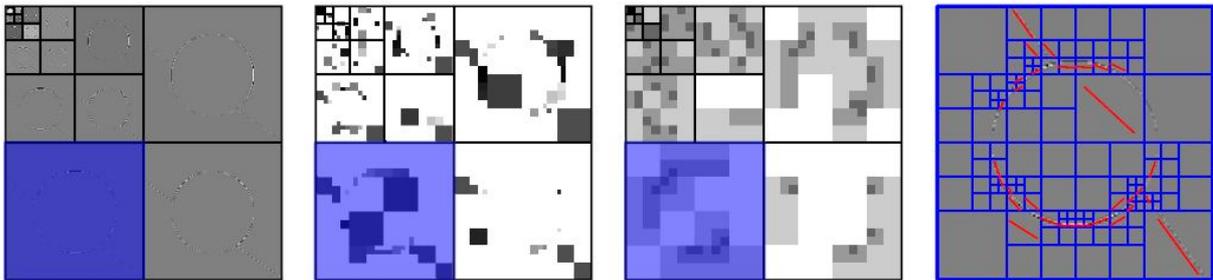


Figure3.3 : Un exemple de quadtree calculée en utilisant la fonction *compute-wavelet-quadtree*.

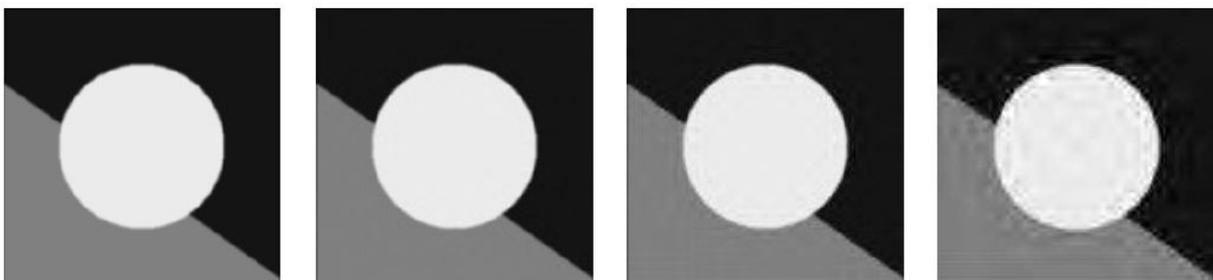


Figure3.4 : La compression d'image en utilisant multi-résolution bandelettes

Mesure de la distorsion

La distorsion entre l'image originale et l'image reconstituée est donnée par l'erreur quadratique moyenne (EQM) et le rapport signal sur bruit crête (PSNR) :

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{EQM}} \right) \dots \dots \dots (3.1)$$

$$\text{EQM} = \frac{1}{n \times m} \sum_{i=1}^n \sum_{j=1}^m (x_{ij} - y_{ij})^2 \dots \dots \dots (3.2)$$

- n, m : longueur et largeur de l'image, respectivement.
- x_{ij}, y_{ij} : valeur (niveau de gris) d'un pixel de l'image originale et d'un pixel de l'image reconstituée respectivement, à la même position spatial.

$$\text{RC}(\%) = 100 \times \left(1 - \frac{\text{taille de l'image après le codage entropique}}{\text{taille de l'image originale}} \right) \dots \dots \dots (3.3)$$

- RC : taux de compression.

Discussion

On a pas obtenus de résultat pour la transformée de Alpert, car le programme que j'ai utilisée n'a pas fonctionner.

Conclusion

Conclusion générale

Conclusion générale

Le travail réalisé dans ce mémoire, a pour objectif l'application d'une méthode de représentation des images par la transformée de Alpert dans le domaine de compression d'image.

Après l'étude théorique qu'on a menée, et le développement du programme, j'ai trouvais des difficultés pour appliqué la fonction de Alpert sur les bandelettes et aussi je n'ai pas eu assez de temps et l'aide qu'il faut pour terminer ce que j'ai commencer aussi la conclusion pour se mémoire reste ouverte jusqu'a obtention prochainement de nouveaux résultats.

ANNEXE

Les algorithmes avec les quels on a travaillés :

```
function x = perform haar transform(x, dir);
x = x(:); % to be sure we have a column vector
J = floor( log2(length(x)) ); % number of scales
if dir==1 % forward transform
for j=1:J
c = x(1:2^(j-1):end); % previous coarse signal
x(1:2^j:end) = ... % new coarse signal
( c(1:2:end) + c(2:2:end) )/sqrt(2);
x(1+2^(j-1):2^j:end) = ... % new details
(c(1:2:end)-c(2:2:end))/sqrt(2);
end
else % backward transform
for j=J:-1:1
y = x(1:2^(j-1):end);
x(1:2^j:end) = ...
( y(1:2:end) + y(2:2:end) )/sqrt(2);
x(1+2^(j-1):2^j:end) = ...
( y(1:2:end) - y(2:2:end) )/sqrt(2);
end
end
```

Listing 1: Function perform-haar-transform

```
function M = perform warped haar(M,theta,dir)
if theta==Inf % special token : no geometry
return; % nothing to do
end
n = size(M,1);
% sampling location
[Y,X] = meshgrid(1:n,1:n);
% projection on orthogonal direction
t = -sin(theta)*X(:) + cos(theta)*Y(:);
% order points in increasing order
[tmp,I] = sort(t);
M(I) = perform haar transform(M(I),dir);
```

Listing 2: Function perform-warped-haar

```
function [MW,theta,L] = compute best direction(M,T)
% samples the direction
[Y,X] = meshgrid(0:n-1, 0:n-1); X = X(:); Y = Y(:);
X(1) = []; Y(1) = [];
Theta = atan2(Y(:),X(:)); Theta = unique(Theta);
Theta = [-Theta(end-1:-1:2); Theta]';
% take mid points
Theta = ( Theta + [Theta(2:end),Theta(1)+pi] )/2;
Theta = [Theta, Inf]; % add 'no geometry' token
% now check for each direction
n = size(M,1);
t = pi/(2*n*s);
Theta = [t/2:t:pi-t/2, Inf];
% compute the lagrangian
L = []; % to store the
for theta = Theta
MW = perform warped haar(M,theta,1);
% compute the error
MWt = MW .* (abs(MW)<T); % residual
m = sum( abs(MW(:))>T ); % number of coef above T
% compute the lagrangian
l = sum(MWt(:).^2) + m*T^2;
if theta~=Inf
% here we need to store a geometric coefficient
l = l + T^2;
end
L = [L; l];
end
```

```

% find minimum of lagrangian
[L,I] = min(L);
theta = Theta(I(end));
MW = perform warped haar(M,theta,1);

```

Listing 3: Function compute-best-direction

```

function [QT,Theta] = ...
compute quadtree(M,T,j min,j max)
n = size(M);
% we assume that a split in the QT is
% coded with 1/7 th of coefficients.
gamma = 1/7;
QT = zeros(n)+j min; Theta = zeros(n);
L = zeros(n/2^j min); % the current lagrangian
% compute bandelet approximation for each square
for kx=0:n/2^j min-1
for ky=0:n/2^j min-1
selx = kx*2^j min+1:(kx+1)*2^j min;
sely = ky*2^j min+1:(ky+1)*2^j min;
% compute the optimal direction on this square
[tmp,Theta(selx,sely),L(kx+1,ky+1)] = ...
compute best direction(M(selx,sely),T);
end
end
% perform the bottom-up procedure, trying to merge
% 4 squares into 1 if it decreases the lagrangian
for j=j min+1:j max
% new lagrangian for this size of squares
L1 = zeros(n/2^j);
for kx=0:n/2^j-1
for ky=0:n/2^j-1
selx = kx*2^j+1:(kx+1)*2^j;
sely = ky*2^j+1:(ky+1)*2^j;
% the lagrangian of the 4 squares splited (add
% gamma penalty because of the split)
l sum = L(2*kx+1,2*ky+1) + ...
L(2*kx+2,2*ky+1) + L(2*kx+1,2*ky+2) + ...
L(2*kx+2,2*ky+2) + gamma*T^2;
% the lagrangian of the 4 squares once merged
[tmp,theta,l] = ...
compute best direction(M(selx,sely),T);
% perform the conditional merging
if l < l sum
L1(kx+1,ky+1) = l; QT(selx,sely) = j;
Theta(selx,sely) = theta;
else
L1(kx+1,ky+1) = l sum;
end
end
end
L = L1;
end

```

Listing 4: Function compute-quadtree

```

function [MB,m geom] = ...
perform bandelet transform(M,QT,Theta,dir)
n = size(M,1); MB = zeros(n); m geom = 0;
gamma = 1/7; % 4:1 split cost
j min = min(QT(:)); j max = max(QT(:));
% display subdivision
for j=j max:-1:j min
w = 2^j/n;
for kx=0:n/2^j-1
for ky=0:n/2^j-1

selx = kx*2^j+1:(kx+1)*2^j;

sely = ky*2^j+1:(ky+1)*2^j;
if QT(kx*2^j+1, ky*2^j+1)==j
% this is a leaf, transform it

```

```

theta = Theta(kx*2^j+1, ky*2^j+1);
MB(selx,sely) = ...
perform warped haar(M(selx,sely),theta,dir);
% one coefficient for geometry
m geom = m geom + (theta~=Inf);
else % add split cost
m geom = m geom + gamma;
end
end
end
end

```

Listing 5: Function perform bandelet transform

```

function MW = perform wavelet transform(M,Jmin, dir)
% retrieve the 7-9 CDF biorthogonal filters
[qmf,dqmf] = MakeBSFilter( 'CDF', [4,4] );
% compute biorthogonal wavelet transform
if dir==1
MW = FWT2 SBS(M,Jmin,qmf,dqmf);
else
MW = IWT2 SBS(M,Jmin,qmf,dqmf);
end
end

```

Listing 6: Function perform wavelet transform

```

function [QT,Theta] = ...
compute wavelet quadtree(M,Jmin,T,j min,j max)
% perform the wavelet transform
MW = perform wavelet transform(M,Jmin, 1);
n = size(M,1); Jmax = log2(n)-1;
QT = zeros(n); Theta = zeros(n);
% compute the transform for each scale and direction
for j=Jmax:-1:Jmin % for each scale
j max = min(j max, j);
for q=1:3 % for each orientation
if q==1 % 1st quadrant
selx = 1:2^j; sely = (2^j+1):2^(j+1);
elseif q==2
selx = (2^j+1):2^(j+1); sely = 1:2^j;
else
selx = (2^j+1):2^(j+1); sely = (2^j+1):2^(j+1);
end
[QT(selx,sely),Theta(selx,sely)] = ...
compute quadtree(MW(selx,sely),T,j min,j max);
end
end

```

Listing 7: Function compute-wavelet-quadtree

```

function [MB,m geom] = ...
perform wav band transform(M,Jmin,QT,Theta,dir)
MB = M;
if dir==1
% perform the wavelet transform
MB = perform wavelet transform(M,Jmin, 1);
end
n = size(M,1); Jmax = log2(n)-1;
m geom = 0;
% compute the transform for each scale and direction
for j=Jmax:-1:Jmin % for each scale
for q=1:3 % for each orientation
if q==1 % 1st quadrant
selx = 1:2^j; sely = (2^j+1):2^(j+1);
elseif q==2
selx = (2^j+1):2^(j+1); sely = 1:2^j;
else
selx = (2^j+1):2^(j+1); sely = (2^j+1):2^(j+1);
end

```

```

[MB(selx,sely),m] = ...
perform bandelet transform(MB(selx,sely), ...
QT(selx,sely),Theta(selx,sely),dir);
m geom = m geom + m;
end
end
if dir==-1
% perform the inverse wavelet transform
MB = perform wavelet transform(MB,Jmin, -1);
end

```

Listing 8: Function perform wav-band-transform

```

function [w,info] = perform_alpert_transform_2d(v,pos,alpert_vm, dir, options)

% perform_alpert_transform_2d - transform a 2D signal.
%
%
% [w,info] = perform_alpert_transform_2d(v,pos,alpert_vm, dir, options);
%
% 'v' is a 1D vector, the value of the function at each sampling location.
% 'pos' is a 2D vector, pos(:,i) is the ith point.
% 'alpert_vm' is the number of vanishing moments (1=>Haar, 2=>linear basis ...).
% * 'alpert_vm' can be an integer, and then the algorithm will use the same
% order for X and Y direction.
% * 'alpert_vm' can be a couple of integer alpert_vm=[kx,ky] and 'kx' will be the
% order on the X direction, and 'ky' the order on the Y direction.
% * 'alpert_vm' can be a set of monomial, see below ('degree_type') for
% further comments.
% 'dir' is 1 for fwd transform and -1 for bwd.
%
% 'options' is a structure that can contains the followind field :
% 'degree_type' : Polynomial degree. By default, the multiresolution spaces are defined
% as piecewise polynomials P that satisfy
% degX(P)<alpert_vm(1) and degY(P)<alpert_vm(2).
% If you want to use spaces defined by
% degX(P)+degY(P) < alpert_vm(1)
% then you should specify degree_type='sum'
% (default is degree_type='max').
% If you want to define your own multiresolution space,
% you can provide your own monomials in 'alpert_vm' and then
% set degree_type='user_defined'. It's a bit tricky
% because you have to provide an even number of monomials,
% twice more than needed. Suppose you want to use
% as multiresolution basis the polynomials {1,X}, then you can set
% alpert_vm = [[0;0],[1;0],[0;1],[1;1]];
% 'part_type': for automatic partition, the way the algorithm
% will perform the grouping (can be either 'laxis', '2axis' or 'kmeans',
% type 'help dichotomic_grouping' for more info).
% 'part': if you don't want to use automatic grouping, then
% you can provide a cell array that contains a binary grouping of the points
% (same format as 'dichotomic_grouping' function).
%
% 'w' is the transformed data.
% 'info' is a struct containing the localisation information for each
% basis Alpert vector.
% 'info.l' is the scale of the vector (0=coarse scale).
% 'info.n' is the space location of the vector.
% 'info.k' is the number of multiwavelet (in [1,...,alpert_vm(1)*alpert_vm(2)]).
%
% WARNING: the function will try to use the mex-compiled function
% perform_moment_transform.dll if possible, and then it
% won't retrun 'info'. Otherwise, it will use the slower
% function 'perform_moment_transform_slow' and
% 'info' will be returned.
%
% Copyright (c) 2004 Gabriel Peyr
v = v(:);

if nargin<2
error('You must provide sampling location in pos.');
```

```

if nargin<3
    alpert_vm = 3;
end
if nargin<4
    dir=1;
end

options.null = 0;

if isfield(options, 'degree_type')
    degree_type = options.degree_type;
else
    if length(alpert_vm(:))<=2
        degree_type='max';
    else
        degree_type='user_defined';
    end
end

use_mex = getoptions(options, 'use_mex', 1);
part_type = getoptions(options, 'part_type', '2axis');
part = getoptions(options, 'part', []);

if length(alpert_vm)==1
    % use same order for X and Y
    alpert_vm = [alpert_vm alpert_vm];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% number of points
n = size(pos,2);
if n==0
    w = [];
    info.l = [];
    info.n = [];
    info.k = [];
    return;
end

switch lower(degree_type)
    case 'max'
        if n<alpert_vm(1)*alpert_vm(2)
            % special case, not enough data.
            % We want kx*ky<n subject to the same ratio, ie kx/ky=alpert_vm(1)/alpert_vm(2);
            kk = alpert_vm; % keep old one
            kx = floor( sqrt(n*alpert_vm(1)/alpert_vm(2)) );
            if kx==0
                kx=1;
            end
            ky = floor( n/kx );
            if ky==0
                ky=1;
            end
            alpert_vm = [kx ky];
        end
        % define the monomial basis
        [MX,MY] = meshgrid(0:2*alpert_vm(1)-1, 0:alpert_vm(2)-1);
        monomials = [MX(:)';MY(:)'];
    case 'sum'
        alpert_vm = alpert_vm(1);
        while n<alpert_vm*(alpert_vm+1)/2
            alpert_vm = alpert_vm-1;
        end
        % define the monomial basis
        s = 0; m = 0; i = 0;

```

```

monomials = [];
while s<alpert_vm*(alpert_vm+1)
    s = s+1;
    monomials = [monomials, [i;m-i]];
    i = i+1;
    if i>m
        m = m+1;
        i = 0;
    end
end
case 'user_defined'
    monomials = alpert_vm;
otherwise
    error('Unknown degree type.');
```

end

```

if size(monomials,1)~=2
    monomials = monomials';
end
if size(monomials,1)~=2
    error('monomials must be of size 2x(2*k2).');
```

end

```

k2 = size(monomials,2)/2;    % equivalent to k^2 in Alpert paper.
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% find a regroupement
if isempty(part) && (~strcmp(part_type, 'laxis') || exist('perform_moment_transform')==0 ||
use_mex==0 )
    clear options;
    options.ptmax = k2;
    options.part_type = part_type;
    [part,B,G] = dichotomic_grouping(pos,options);
end
```

```

if exist('perform_moment_transform')>0 && use_mex
    if strcmp(part_type, 'laxis') && isempty(part)
        % will automatically compute the partition using X axis
        w = perform_moment_transform(v,pos, monomials, dir);
    else
        w = perform_moment_transform(v,pos, monomials, dir, part);
    end
    info = [];
else
    [w,info] = perform_moment_transform_slow(v,pos, monomials, dir, part);
end
```

listing9: perform_alpert_transform_2d

Bibliographie

- [1] Gabriel PEYRÉ. « Géométrie multi-échelles pour les images et les textures », thèse de Doctorat, Ecole polytechniques, Paris, 13 Déc. 2005.
- [2] A.Saoud «compression d'image fixes par ondelettes géométriques (base de bandelettes)» thèse de magister en électronique, université de Biskra 2010.
- [3] L. Seriani et S.Mansouri «compression d'image fixes par la transformée en bandelettes» thèse d'ingénieur d'état en électronique, université de Biskra 2008.
- [4] E. Le Pennec and S.Mallat, « Sparse Geometrical Image Approximation with Bandelets », IEEE Transaction on Image Processing, 14(4), p. 423–438, avril 2005.
- [5] Benjamin LEGUEN, Stéphane Pateux et Jaque Weiss. «Modèle énergétique pour la représentation d'image par ondelettes déformées».France télécom.
- [6] Stéphane Mallat & Gabriel Peyré. « Traitements géométriques des images par bandelettes », Journée annuelle 2006.
- [7] OUAFI Abdelkarim « compression d'images fixes biomédicales par les transformées en ondelettes associées aux algorithmes de quantification vectorielle et de codage entropique » ; thèse magister en électronique ; université de Biskra ; Mai 2001.
- [8]. S. MALLAT and G. PEYRE. « Orthogonal Bandlet Bases for Geometric Images Approximation », CMAP, Ecole Polytechnique, CEREMADE, Université Paris Dauphine, 2006.
- [9]. Mallat, S. « A Wavelet Tour of Signal Processing ». Academic, San Diego (1998).
- [10]. Erwan Le Pennec. « Bandelettes et représentation géométrique des images », thèse de doctorat, Ecole polytechniques, Paris, 19 Déc 2002.
- [11]. Erwan Le Pennec, Stéphane Mallat . « Bandelettes et représentation géométrique des images », CMAP Polytechnique. 91128 PALAISEAU – France.
- [12]. E. Le Pennec and S. Mallat. « Bandelet Image Approximation and Compression », SIAM Multiscale Modeling and Simulation, 2005.
- [13]. L. Breiman, J. H. Friedman, R. A. Olshen, , and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.
- [14]. D. Donoho. Wedgelets : Nearly-minimax estimation of edges. *Ann. Statist*, 27 :353–382, 1999.