

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMARI, TIZI-OUZOU  
FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE  
DEPARTEMENT D'ELECTRONIQUE



*MEMOIRE DE MAGISTER EN ELECTRONIQUE*

*OPTION : Télédétection*

Présenté par :

**Mme LAHDIR Née AKROUR LEILA**

THEME :

**Compression d'images par SPIHT appliqué dans le  
domaine des ondelettes entières :  
Application aux images MSG**

Devant les membres du jury :

<u>Président</u> :	Mr .	AMEUR	Soltane	Professeur	UMMTO
<u>Rapporteur</u> :	Mme.	AMEUR Née MAZOUZI	Zohra	Maître de conférences	UMMTO
<u>Examineurs</u> :	Mr.	LAGHROUCHE	Mourad	Maître de conférences	UMMTO
	Mr.	HADDAB	Salah	Maître de conférences	UMMTO

Soutenu le :                    /                    /

*À tous ceux qui font partie de ma vie*

*Et qui sont très chers à mon cœur*

*Plus particulièrement, mes anges : Mourad et Imane*



# REMERCIEMENTS



*Le travail que j'ai l'honneur de présenter, a été effectué en grande partie au laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires « L.A.M.P.A » de l'Université Mouloud MAMMERI de TIZI-OUZOU.*

*Avant tout, je remercie le bon dieu, le tout puissant de m'avoir donné la force nécessaire pour mener à terme ce travail.*

*Ensuite, je tiens à remercier Madame AMEUR née MAZOUZI Zohra, Maître de conférences à UMMTO, Directeur de mémoire, pour l'intérêt avec lequel elle a suivi ce travail, pour ses conseils et pour les encouragements précieux qu'elle n'a cessé de me prodiguer.*

*J'exprime ma profonde gratitude à Monsieur AMEUR Soltane, Professeur à UMMTO, d'avoir accepté la présidence du Jury chargé d'examiner ce mémoire. C'est un grand honneur pour moi de présenter mon travail devant notre directeur du laboratoire L.A.M.P.A.*

*Que Monsieur LAGHROUCHE Mourad, Maître de conférences à UMMTO, trouve ici, l'expression de ma profonde reconnaissance pour tout l'intérêt qu'il a manifesté à mon travail, pour ses conseils judicieux mais aussi pour avoir accepté de faire partie du Jury.*

*A Monsieur HADDAB Salah, Maître de conférences à UMMTO, j'exprime mes remerciements les plus sincères pour avoir accepté de faire partie du Jury examinant ce mémoire. C'est avec un énorme plaisir que je le vois prendre part à ce Jury.*

*Je tiens aussi à témoigner ma profonde gratitude ainsi que mes plus vifs remerciements à Monsieur LAHDIR Mourad, Maître de conférences à UMMTO, mon époux, pour sa patience, ses conseils judicieux, ses encouragements mais aussi pour son soutien moral.*

*Que toute personne qui, d'une manière ou d'une autre, m'a aidé et encouragé pour l'aboutissement de ce travail, trouve ici l'expression de ma sincère reconnaissance.*

*Leila AKROUR Ep. LAHDIR*

# Sommaire

<i>Introduction</i> .....	1
 <i>Chapitre I : Aperçu sur la compression d'images</i>	
1.1 Préambule.....	4
1.2. Généralités sur la compression des images .....	5
1.2.1 Définitions et notions de base.....	5
1.2.2. Structure générale d'un schéma de compression d'image.....	7
1.2.3 Concept d'évaluation d'une compression .....	11
-Taux de compression et débit binaire .....	11
-Compromis débit distorsion et critères de qualité .....	12
-Complexité .....	13
1.2.4 Classification des méthodes de compression d'images.....	13
1.3. Discussion.....	16
 <i>Chapitre II : La Transformée en ondelettes entières</i>	
2.1 Préambule.....	17
2.2. La Transformée en ondelettes par convolution classique.....	18
2.2.1. La transformée en ondelettes continue (CWT) .....	19
2.2.2 La transformée en ondelettes discrète (DWT) .....	20
2.2.3 Analyse Multirésolution .....	21
2.2.4. Choix d'ondelettes.....	24
2.2.5. Application aux signaux bidimensionnels.....	26
2.3. La Transformée en ondelettes par schéma de lifting (LWT) .....	30
2.3.1 La transformée polyphase .....	32
2.3.2 Opération de prédiction .....	32
2.3.3 Opération de mise à jour.....	33

2.3.4	Représentation polyphase des bancs de filtres.....	33
2.3.5	Algorithme de factorisation de la matrice polyphase.....	36
2.3.6	La transformée en ondelettes entières.....	39
2.3.6.a	Principe.....	39
2.3.6.b	Décomposition en ondelettes entière avec l'ondelette CDF 9/7.....	40
2.3.6.c	Décomposition en ondelettes entière avec l'ondelette Gall 5/3.....	42
2.4	Discussion.....	43

### ***Chapitre III : Compression d'images par ondelettes entières Et codeurs à arbres de zéros***

3.1	Préambule.....	44
3.2.	Méthode de compression décompression.....	45
3.2.1.	Principe.....	45
3.2.2	Transformation en ondelettes entières (LWT) .....	46
3.2.3	Etape de quantification.....	47
3.2.4	Etape de codage à arbre de zéro .....	48
3.2.4.1	Codeur EZW .....	49
3.2.4.2	Codeur SPIHT.....	55
3.3.	Discussion.....	59

### ***Chapitre IV: Résultats et Discussion***

4.1	Préambule.....	61
4.2.	Présentation des données.....	61
4.2.1.	Image Tests.....	62
4.2.2	Images Météosat Seconde Génération (MSG) .....	62
4.3.	Résultats expérimentaux et interprétations.....	65
4.3.1	Application aux images tests.....	66
4.3.2	Application aux images MSG.....	77
4.4.	Discussion.....	83

*Conclusion* .....85

*Annexe A : Les ondelettes biorthogonales*

*Annexe B : Exemples pour EZW et SPIHT*

*Références*

# Table des figures

1 Schéma de compression /décompression classique d'images.....	7
2 Principe de la quantification.....	9
3 L'ondelette de Morlet après dilatation et contraction.....	19
4 Représentation multirésolution de l'image Lena à la résolution trois.....	24
5 Analyse multirésolution à deux niveaux de décompositions d'une image.....	26
6.a Histogramme des intensités des pixels d'une image.....	27
6.b Histogramme des coefficients d'ondelettes d'une sous image.....	27
7 Transformée en ondelettes à une décomposition : Analyse et synthèse.....	28
8 Décomposition par bancs de filtres sur trois niveaux.....	29
9 Reconstitution par bancs de filtres sur trois niveaux.....	30
10 Transformation en ondelettes par schéma de lifting.....	31
11 Représentation polyphasée du banc de filtres d'analyse.....	35
12 Représentation polyphasée du banc de filtres de synthèse.....	36
13 Structure en échelle pour les bancs de filtres d'analyse.....	38
14 Structure en échelle pour les bancs de filtres de synthèse.....	38
15 Factorisation de l'opération de mise à l'échelle de $s(n)$ et $d(n)$ .....	40
16 Schéma du processus de compression.....	45
17 Illustration des dépendances inter échelles.....	46
18 Définition de la zone de quantification et mise en évidence de la zone morte.....	48
19 Illustration de l'approche inter bandes.....	49
20 Modèle de balayage des coefficients d'ondelettes.....	51
21 Test de signification des coefficients d'ondelettes.....	54
22 Terminologie SPIHT pour les descendants.....	56

23	Principe général de l'algorithme SPIHT.....	60
24	Images tests.....	62
25.a	Image visible 0.6.....	64
25.b	Image infrarouge 10.8.....	65
26	Evolution du PSNR en fonction du niveau de décomposition level.....	67
27	PSNR en fonction du taux en bpp avec le codeur EZW.....	69
28	PSNR en fonction du taux en bpp avec le codeur SPIHT.....	70
29	Images reconstruites avec Gall5/3_EZW et Gall5/3_SPIHT pour 0,5 bpp.....	71
30	Images reconstruites avec CDF9/7_EZW et CDF9/7_SPIHT pour 0,5 bpp.....	72
31	PSNR en fonction du taux en bpp pour l'image Lena avec CDF9/7 pour EZW et SPIHT....	73
32	Images reconstruites avec CDF9/7_EZW et CDF9/7_SPIHT pour 1 bpp.....	74
33	Temps de compression en fonction du taux en bpp dans le cas de l'image Lena.....	74
34	PSNR en fonction du taux en bpp du codeur SPIHT.....	76
35	PSNR en fonction du taux en bpp pour les images MSG.....	79
36	Image visible 0.6 reconstruite pour un taux de 0,5 bpp.....	80
37	Image visible 0.6 reconstruite pour un taux de 1 bpp.....	81
38	Image infrarouge 10.8 reconstruite pour un taux de 0,5 bpp.....	82
39	Image infrarouge 10.8 reconstruite pour un taux de 1 bpp.....	83

# *Liste des tableaux*

1	Comparaison des résultats obtenus sur l'image Lena.....	15
2	Bandes spectrales des douze canaux du satellite MSG.....	63
3	Résultats des PSNR en fonction du niveau de décomposition level.....	66
4	Résultats des PSNR en fonction du taux en bpp pour l'ondelette CDF9/7.....	68
5	Résultats des PSNR en fonction du taux en bpp pour l'ondelette Gall5/3.....	68
6	Résultats des PSNR en fonction du taux en bpp pour l'ondelette bior4.4.....	75
7	Comparaison des résultats obtenus sur l'image Lena avec ceux de la littérature.....	77
8	Résultats obtenus par CDF9/7_SPIHT sur les images MSG.....	78

# *Introduction*

L'essor de l'informatique durant les dernières décennies fit entrer l'humanité dans une nouvelle ère numérique. Aujourd'hui, la plupart des informations ou des données, que nous sommes amenés à manipuler dans notre vie quotidienne sont stockées, traitées et représentées sous forme numérique, car celle-ci fiabilise leur transmissions à travers les réseaux informatiques et facilite leur manipulation.

Toutefois, la puissance des processeurs augmente plus vite que les capacités de stockage, et énormément plus vite que la bande passante des réseaux, qui, malgré, les nouvelles technologies, a du mal à augmenter car cela demande d'énormes changements dans les infrastructures. Ainsi, nous préférons réduire la taille des données en exploitant la puissance des processeurs plutôt que d'augmenter les capacités de stockage et de télécommunication. Une grande partie de ces données sont des images, qui proviennent de différents domaines tels que la physique, l'industrie, la culture, la médecine et surtout le domaine d'observation de la terre qui ne cesse d'acquérir des nombres importants d'images satellitaires dans le but d'une analyse en vue de plusieurs applications comme l'agriculture, l'urbanisme, la météorologie,...

En effet, les images satellitaires météorologiques sont produites par des satellites qui peuvent avoir plusieurs canaux spectraux pendant des durées très courtes [Ameur *et al*, 2002 ; lahdir *et al*, 2006], conduisant ainsi à un nombre d'images assez important nécessitant de grandes capacités de stockage. Du coup, la compression semble donc incontournable pour résoudre le problème d'archivage. De plus, elle présente un intérêt évident pour la transmission des images qui peut s'avérer délicate du fait des bandes passantes existantes limitées [Gaudeau, 2006].

L'idée de base de la compression est de réduire le nombre moyen de bits par pixel nécessaire à leurs représentations. Il est possible dans une certaine limite de réduire ce nombre sans pour autant altérer l'image ; c'est le cas d'une compression sans perte. A titre d'exemple, citons la technique RLC (*Run Length Coding*), le codage de Huffman et la compression LZW (*Lempel Ziv Welch*) [Huffman, 1952 ; Ziv *et al*, 1977; Marsault, 1995]. Cependant, ces méthodes présentent des limitations en termes de taux de compression. Tolérer quelques pertes permet

de dépasser ces limites. Un autre avantage des algorithmes avec pertes est qu'ils permettent en général d'ajuster le débit [Christophe, 2006]. Les plus efficaces sont basées sur de puissants outils comme la transformée discrète en cosinus (DCT), la transformée en ondelettes ou l'approche fractale [Lahdir, 2008]. Avec de tels outils, l'image subit une forte compression mais avec des pertes se traduisant par des dégradations plus ou moins perceptibles à l'œil nu lors de la décompression [Lahdir *et al*, 2006]. C'est en utilisant ces méthodes que les membres du groupe JPEG se sont mis à l'élaboration de la nouvelle norme de codage d'images, la plus flexible et la plus performante: JPEG2000. Ce standard est basé principalement sur la transformée en ondelettes ; une transformée qui a montré de bonnes capacités de décorrélation, dans le but de compression sur les images réelles [Christophe, 2006].

Il existe principalement deux tendances pour tirer partie de la décorrélation après la transformée en ondelettes. La première, qui est appliquée dans le standard JPEG 2000 est d'utiliser un codeur arithmétique pour coder les coefficients d'ondelettes. La seconde, tire partie du lien existant entre les coefficients de la transformée, situés dans les différentes sous bandes (même si la corrélation est presque nulle). Cette dernière est appelée codage à arbres de zéros [Shapiro, 1993 ; Said et Pearlman, 1996]

Les arbres de zéros sur les coefficients d'ondelettes ont été développés pour utiliser la relation qui existe entre la position des singularités dans les différentes sous bandes. C'est dans ce contexte que s'inspire notre travail qui est défini par le développement d'une nouvelle approche de compression d'images basée sur les ondelettes entières associées aux codeurs à arbres de zéros.

Le travail effectué dans ce mémoire a donc consisté à l'élaboration et l'implémentation d'un nouvel algorithme se basant sur des codeurs particuliers, à savoir : EZW (Embedded Zerotree Wevelet coefficients) et SPIHT (Set Partitioning In Hierarchical Trees) [Shapiro, 1993 ; Said and Pearlman, 1996]. Néanmoins, pour mieux exploiter les avantages de ces deux codeurs, nous avons effectué, tout d'abord, une transformation par ondelettes entières tout en utilisant deux types de filtres principalement choisis pour leurs performances en compression.

A cet effet, le premier chapitre est un aperçu global sur la compression. Son objectif principal est de cerner les différentes techniques de compression existantes ainsi que les critères utilisés pour l'évaluation des performances d'une méthode, afin de pouvoir choisir la plus appropriée à notre schéma.

Sachant que le point de base de notre algorithme est l'utilisation des codeurs à arbres de zéros, le deuxième chapitre introduira la transformée la plus adaptative à ce type de codeurs ; celle qui offrira l'autosimilarité spatiale entre les sous bandes, critère essentiellement exploité par ces codeurs. Dans ce chapitre seront définis : la transformation par ondelettes sous ses deux catégories à savoir par convolution et par lifting, les propriétés qui les caractérisent, ainsi que la définition du critère essentiel pris pour fixer le choix sur les ondelettes par lifting et plus singulièrement le choix du filtre à utiliser.

L'algorithme élaboré est détaillé dans le troisième chapitre, dans lequel nous expliquerons les différentes étapes de notre schéma de compression, tout en développant les points de base des codeurs EZW et SPIHT.

Le quatrième chapitre sera consacré à l'évaluation de notre méthode en effectuant les différents tests sur des images références ainsi que le déploiement et l'analyse des différents résultats obtenus. Nous concluons ce chapitre par l'application de notre algorithme sur des images MSG. Les résultats obtenus mettront l'accent sur la performance réalisée par le codeur SPIHT dans le domaine des ondelettes entières qui s'avèrent très bien adaptées pour la localisation de l'énergie de l'image dans un nombre réduit de coefficients.

Enfin, nous terminerons ce mémoire par une conclusion générale tout en soulignant les limites de notre algorithme et en discutant les perspectives ouvertes par ce travail.

# *Chapitre I*

## *Aperçu sur la compression d'images*

### **1.1 Préambule**

Les multimédia ont fait entrer les images numériques partout dans notre vie quotidienne (appareils photos numériques, Internet, télévision, téléphones portables...). La tendance actuelle est à l'utilisation croissante de ces images digitalisées. La plupart des techniques modernes d'imagerie (spécialement dans le domaine médicale ou satellitaires) produisent des images intrinsèquement volumiques, alors que, d'autres correspondent au contraire à une succession d'images 2D (encore appelée pile d'images), à laquelle nous ajoutons une dimension supplémentaire, à savoir l'écart entre deux coupes successives. De ce fait, la majorité des images produites de nos jours peuvent être vues comme des images à au moins trois dimensions. Chaque jour, ces nombreuses images sont transférées sur Internet, d'un appareil d'acquisition à un ordinateur ou particulièrement d'un téléphone portable à un autre [Marsault, 1995]. La quantité importante de ces images volumiques que génère chaque jour un service classique d'acquisition d'images (multimédia, médicales ou un système d'acquisition embarqué dans un satellite) se chiffre à plusieurs Téra (Péta) octets de données en une année [Gaudeau, 2006]. Par ailleurs, ces données doivent être conservées un certain nombre d'années pour répondre aux contraintes réglementaires actuelles, mais aussi, être transmises dans le but d'une étude donnée. L'augmentation croissante et continue des capacités de stockage apporte une réponse partielle à ce problème mais demeure la plupart du temps insuffisante. Par conséquent, la compression dans ce cas est fatidique car elle présente un grand intérêt pour l'archivage et la transmission des images à la fois. Les techniques de compression se différencient par le fait qu'elles permettent ou

non de compresser sans pertes d'informations, c'est à dire de manière réversible ou irréversible [Akrou, 2003]. Ce chapitre a pour but de présenter les méthodes actuelles de compression avec et sans perte les plus efficaces pour les images satellitaires. L'ensemble de ces méthodes exploitent la corrélation des images afin d'améliorer la performance de compression.

## 1.2. Généralités sur la compression des images

L'objectif principal de la compression d'images est de réduire la quantité d'informations nécessaire à sa représentation. Cette réduction est obtenue en minimisant la redondance de l'information présente dans une image. Cette redondance peut être statistique, spatiale ou encore fréquentielle [Davoine, 1995; Mercier *et al.*, 2003; Brielmann, 2006]. Il existe une variété d'algorithmes de compression qui exploitent ces différents types de redondances. Ainsi, nous pouvons les classer en deux grandes familles : ceux qui peuvent reconstituer l'information exacte (algorithmes sans pertes) et ceux qui tolèrent une perte d'information (algorithmes avec pertes). Pour mieux comprendre les méthodes sans pertes et avec pertes d'information, il est nécessaire de rappeler quelques définitions et les notions de bases sur la théorie de l'information, en particulier les notions d'entropie et de distorsion [Christophe, 2006].

### 1.2.1 Définitions et notions de base

Une image numérique est une matrice composée d'échantillons élémentaires appelés pixels. A chaque pixel est associé une valeur numérique correspondant à la couleur ou à l'intensité lumineuse de celui-ci. Pour chaque couleur, 256 niveaux sont généralement utilisés pour le codage (nombres allant de 0 à 255). Ainsi, une image monochrome en niveaux de gris est codée sur 8 bits par pixel ( $2^8 = 256$  niveaux de gris). Par convention, le niveau de gris 0 correspond à du noir, et le niveau 255 à du blanc. Une image couleur possède trois canaux de couleur : le rouge, le vert et le bleu. Chaque pixel est ainsi codé sur 24 bits (soit 16 millions de couleurs). Un des enjeux de la compression est de réduire le nombre de bits nécessaire pour le codage d'un pixel. La phase de décompression appelée aussi décodage ou restitution est nécessaire pour représenter l'image décompressée sous forme de pixels visibles. Par abus de langage, nous appellerons une image compressée, une image qui, en fait, aura subi successivement l'opération de compression puis l'opération de décompression.

- **Eléments de base en théorie de l'information**

A la fin des années 40, les premières idées sur la compression des données émergeaient avec le développement de la théorie de l'information. Les chercheurs parlaient de concepts tels que : l'entropie, la quantité d'information, la redondance. La théorie de l'information est née des préoccupations techniques des ingénieurs de la télécommunication qui voulaient mesurer l'information et étudier à quelles lois cette information était soumise, et pouvoir par conséquent, quantifier le flux nécessaire à sa transmission et à son stockage [Shanon, 1984 ; Kubasov, 2008]. Ce n'est qu'en 1948 que Shannon et Weaver publièrent la théorie mathématique de la communication ; une théorie concrète associant à chaque message :

- Une mesure de sa quantité d'information.
- Son entropie.

La théorie de l'information permet de définir et de mesurer la quantité d'information, présente dans un message, que nous considérons ici sous la forme d'une image. Chaque point de l'image représente une variable aléatoire, et une image de  $M*N$  points est représentée par une suite de  $M*N$  variables aléatoires. Soient  $n_i$  (avec  $0 \leq i \leq 255$ ), les niveaux de gris possibles d'un point particulier  $P$ , et  $p(n_i)$  la probabilité du niveau de gris  $n_i$ . L'information propre de  $n_i$ , notée  $I(n_i)$ , est telle que :

$$I(n_i) = \log_a \frac{1}{p(n_i)} \quad (1.1)$$

$I(n_i)$  mesure alors la quantité d'informations reçue lorsque nous savons que  $P$  a pour niveau de gris  $n_i$ . L'unité employée est le bit quand la base  $a$  vaut 2. Plus la probabilité est petite, plus la quantité d'information propre est grande. L'entropie de  $P$  est la valeur moyenne de  $I(n_i)$ , qui se note  $H(P)$  (en bits/symbole en base 2) :

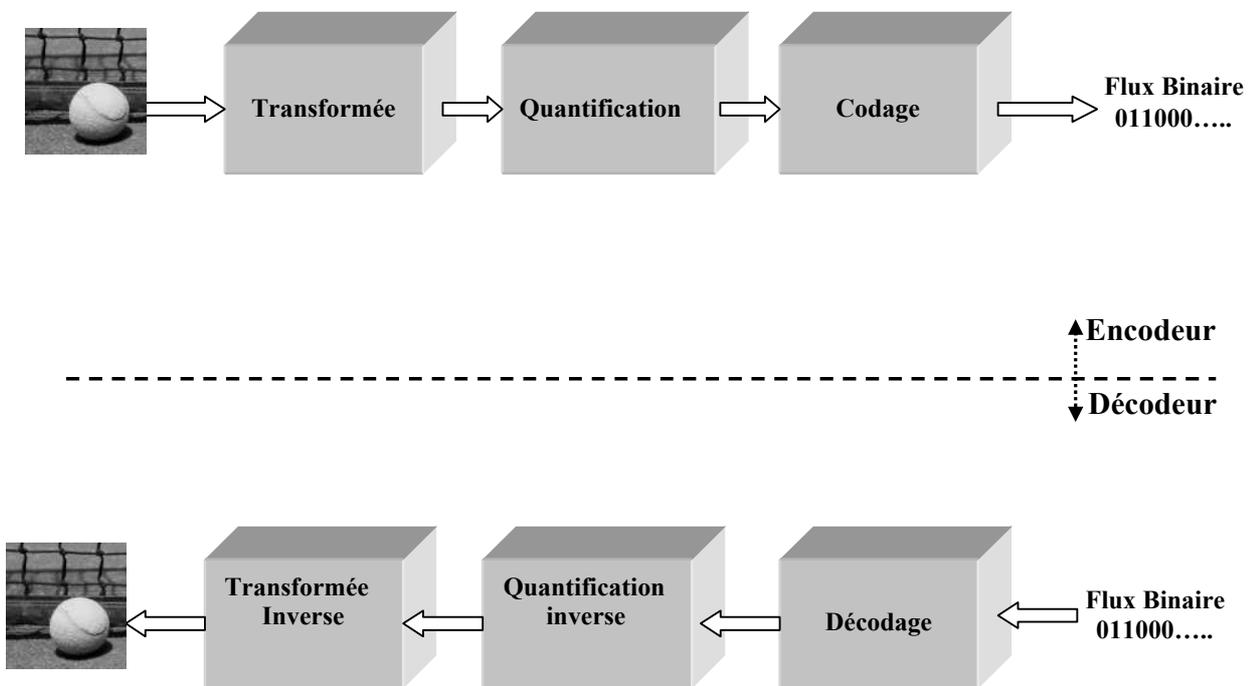
$$H(P) = \sum_{i=0}^{255} p(n_i) I(n_i) = - \sum_{i=0}^{255} p(n_i) \cdot \log_2(p(n_i)) \quad (1.2)$$

Si  $L$  est la longueur moyenne d'un code, il faut que  $L \geq H(P)$  pour préserver toute l'information nécessaire. L'entropie est maximale lorsque nous avons le maximum d'incertitude quant à l'apparition des symboles, c'est-à-dire, si les probabilités des différents états sont égales. L'entropie représente, alors, la quantité d'information minimale nécessaire à une bonne

représentation de l'image. Il s'agit de la limite inférieure du nombre moyen de digits binaires requis pour le codage de l'image.

### 1.2.2. Structure générale d'un schéma de compression d'image

Dans cette section, nous cherchons à établir un concept qui guidera l'étude des différentes méthodes de compression. Nous donnons un schéma dont les éléments sont couramment utilisés dans la communauté scientifique de la compression d'images. Ce schéma général souvent utilisé pour décrire le fonctionnement des algorithmes de compression est présenté dans la figure 1 [Gaudeau, 2006]. Les méthodes de compression d'images avec perte suivent les 3 étapes classiques de compression d'images naturelles. Elles débutent pour la plupart par une réorganisation du contenu de l'image, afin de séparer les composantes importantes (au sens visuel), des composantes contenant peu d'information. Cette tâche est remplie par une transformation mathématique. Cette étape est suivie par la quantification qui dégrade de manière irréversible le signal. La dernière étape de codage (sans perte) produit le flux binaire.



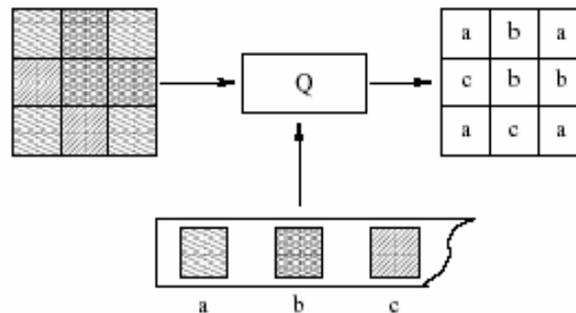
**Fig. 1** Schéma de compression /décompression classique d'images.

- **Etape de Transformation**

Le but de la transformée dans un schéma de compression est de réorganiser l'information, elle doit représenter les composantes importantes d'un signal avec le moins d'éléments possibles : c'est ce que nous appelons donner une représentation creuse du signal ou, de manière équivalente, compacter l'énergie. Dans le domaine spatial, l'information contenue dans l'image est distribuée sur toute la matrice image. Dans le domaine des fréquences, l'information (qui est strictement équivalente) est généralement plus concentrée. En principe, après la transformée, une grande partie de l'énergie de l'image est condensée sur un faible nombre de coefficients. De ce fait, il est approprié de construire un algorithme de compression dans le plan des fréquences. La transformée donc, est l'autre représentation possible de l'image dans laquelle celle-ci est mise sous forme de coefficients de fréquences spatiales. Le passage d'un domaine à l'autre se fait par des transformations mathématiques totalement réversibles, telles que la DCT (Discrete Cosine Transform) qui est utilisée dans le très populaire standard JPEG, la transformée en ondelettes est la base du nouveau standard JPEG 2000. Il existe également la KLT (Karhunen-Loeve Transform) adaptée aux données [Akrou, 2003 ; Cziho, 1999 ; Delgorge, 2005]. Ces transformations sont efficaces mais dépendent de l'image.

- **Etape de Quantification des coefficients**

Dans le schéma de compression, l'étape de quantification est celle qui dégrade de manière irréversible le signal. Elle est cependant d'une importance capitale dans la réduction du débit binaire, car, c'est dans cette étape qu'on se ramène à un nombre limité de coefficients. De nombreux travaux ont été développés sur le codage des coefficients issus de la transformée. Ce passage du continu au discret peut s'effectuer échantillon par échantillon, dans ce cas nous parlerons de quantification scalaire (QS), ou bloc par bloc : c'est ce que nous appelons la quantification vectorielle (QV). En général, c'est le degré de quantification qui détermine le niveau de compression obtenu. Une quantification grossière donne une compression importante mais introduit une forte dégradation [Guillemot, 2004 ; Gaudeau, 2006].



**Fig. 2** Principe de la quantification

Dans le cas de la quantification scalaire, nous distinguons la quantification scalaire uniforme dans laquelle les intervalles de quantification sont tous de la même taille. Cette quantification est optimale au sens de la distorsion pour une distribution uniforme des coefficients. Toutefois, il est possible de construire une quantification scalaire non uniforme optimale. Par exemple, pour la quantification des coefficients de la transformée en ondelettes d'une image, il a été montré que le quantificateur scalaire à zone morte est un très bon choix au sens de la distorsion engendrée. Cependant, pour la même transformée, des algorithmes de codage par plans de bits basés sur des structures d'arbres de zéros donnent actuellement de très bons résultats. Il s'agit principalement de EZW et SPIHT [Shapiro, 1993 ; Said and Pearlman, 1996 ; Pearlman, 1997]. La quantité et l'ordre des données dépendent des valeurs codées. Ces algorithmes seront détaillés plus précisément par la suite.

- **Etape de Codage**

Nous pouvons présenter deux grandes familles de codeurs sans perte : les codeurs entropiques et les codeurs par plages. Ils sont utilisés dans une chaîne de compression sans perte, directement sur l'image de départ, ou après une transformée. Ils sont également employés à la dernière étape de la chaîne de compression avec pertes afin d'exploiter les redondances présentes à la sortie du quantificateur. Le codage entropique est effectué sur les coefficients quantifiés afin d'aboutir au flux binaire de sortie. Plusieurs types de codages entropiques existent. Le codage par Run-Length est utilisé dans la télécopie mais aussi dans le standard JPEG [Borko, 1995 ; Akrou, 2003]. Le codage par dictionnaire comme LZW (Lempel-Ziv-Welch) est utilisé dans le format d'image GIF

(Graphics Interchange Format). Le codage par statistique comme le codage de Huffman est aussi utilisé dans le format JPEG, mais qui présente néanmoins deux limitations principales :

- Il n'est pas toujours facile d'évaluer les probabilités  $p(s_i)$ , notamment lorsque la taille des symboles augmente (pour des symboles codés sur 16 bits, il y a  $2^{16} = 65536$  symboles possibles, il faut donc disposer d'une grande quantité de données pour évaluer les probabilités) ou lorsque la loi de probabilité change au cours de la transmission ;
- La construction du code de Huffman prend en considération un nombre entier de bits pour coder un symbole. Or, la valeur optimale pourrait être fractionnaire. Pour atteindre une longueur fractionnaire, il faudrait utiliser le théorème de Shannon. Aussi, le codage arithmétique remédie à cette dernière limitation du codage de Huffman. C'est actuellement, une des méthodes de codage entropique les plus efficaces pour les données binaires.

Comme nous l'avons déjà évoqué, les méthodes de codage entropique prennent en compte les probabilités d'apparition des symboles pour construire leurs codes. Toutefois, il est courant qu'en sortie d'un quantificateur, la source ne soit pas totalement décorrélée, c'est le cas des coefficients d'ondelettes, ceux de faible amplitude ont tendances à se lier. Les codeurs entropiques évoqués, ci-dessus, ne peuvent exploiter cette mémoire de la source quantifiée. Il existe néanmoins des codeurs capables d'exploiter les statistiques d'ordre supérieures: les codeurs entropiques adaptatifs et les codeurs par plages de zéros. Les codeurs par plages (ou run-length en anglais) sont particulièrement adaptés pour coder les coefficients d'ondelettes. L'idée de base consiste à coder la longueur d'une série d'échantillons nuls, plutôt que, de coder chaque échantillon indépendamment.

### I.2.3 Concept d'évaluation d'une compression

Dans le cas des images classiques, plusieurs méthodes d'évaluation sont utilisées [Christophe, 2006] :

- **Méthodes subjectives** : basées sur l'évaluation de la qualité par des observateurs humains. Elles consistent à faire attribuer une note de qualité par un panel d'observation.
- **Méthodes objectives** : basées sur des critères mathématiques pour évaluer la qualité des images, ou plus précisément, la performance d'une méthode de compression. Les principaux paramètres utilisés pour cette évaluation sont : le taux de compression, le débit, le temps d'exécution et les mesures de distorsions.

En pratique, il s'agit de réaliser le meilleur compromis possible entre le taux de compression et la qualité de l'image décompressée tout en minimisant le temps d'exécution [Lahdir *et al.*, 2006]. De plus, l'algorithme permettant d'implémenter ces méthodes sur ordinateur, doit être robuste aux erreurs de transmission, flexible, et permet une transmission progressive, c'est-à-dire, transmettre une image dont la qualité à la réception s'affine progressivement.

- **Taux de compression et débit binaire**

Sachant que, l'intérêt principal de la compression est de minimiser la quantité d'informations nécessaire à la représentation d'une image, nous pouvons définir le rapport de compression  $R_c$  comme un rapport entre la taille de l'image originale et la taille de l'image compressée en bits.

$$R_c = \frac{\text{Taille de l'image originale en bits}}{\text{Taille de l'image compressée en bits}} \quad (1.3)$$

De là, le taux de compression en pourcentage  $C$  aura la valeur qui suit :

$$C = (1 - 1/R_c) \times 100 \quad (1.4)$$

Le rapport de compression  $R_c$  peut être relié au débit binaire exprimé en bits par pixel (bpp) : c'est le volume des données compressées divisé par le nombre de pixels dans l'image ; plus

précisément, c'est la résolution numérique de l'image originale divisée par le rapport de compression  $R_c$ .

$$D(\text{bpp}) = \frac{\text{Nombre de bits par pixels dans l'image originale}}{R_c} \quad (1.5)$$

- **Compromis débit distorsion et critères de qualité**

Le taux de compression n'est pas le seul critère de performance d'un système de compression. Dans le cas d'une compression avec pertes, la qualité de l'image reconstruite doit aussi être prise en compte. Il y a donc un compromis à trouver entre le taux de compression et la qualité. Nous parlons de compromis débit-distorsion. La distorsion est la mesure de l'erreur commise entre l'image originale et l'image reconstruite. Pour cette mesure, nous utiliserons l'erreur quadratique moyenne *MSE* (*Mean Square Error*) entre l'image originale et l'image compressée de taille  $N \times N$  :

$$MSE = \frac{\sum_{j=0}^{N-1} \sum_{i=0}^{N-1} [I(i, j) - \tilde{I}(i, j)]^2}{N^2} \quad (1.6)$$

Avec :

$N^2$  : Taille de l'image ;

$I(i, j)$  : Un pixel de l'image originale ;

$\tilde{I}(i, j)$  : Un pixel de l'image reconstruite

Dans le cas des mesures de qualité, nous utiliserons le *PSNR* (Peak Signal to Noise Ratio) en décibels défini à partir de l'*MSE* par :

$$PSNR = 10 \log_{10} \frac{(2^R - 1)^2}{MSE} \quad (1.7)$$

$R$  représente le nombre de bits par pixel

- **Complexité**

Un troisième critère de performance d'un système de compression est la complexité. La complexité calculatoire peut être mesurée par le temps d'exécution du processus de compression ou en nombre d'opérations par pixel : c'est le nombre moyen d'opérations qui sont nécessaires à la compression de l'image.

#### 1.2.4 Classification des méthodes de compression d'images

Les méthodes de compression d'images sont divisées en deux classes, à savoir, les méthodes sans pertes et celles avec pertes. Le choix d'une méthode de compression dépend des images à traiter et de leurs entropies. Comme indiqué sur la figure 1, la seule étape qui différencie les méthodes de compression avec pertes de celles sans pertes, est celle de quantification qui n'intervient pas dans ces dernières. En effet, c'est le processus de quantification qui introduit les distorsions qui apparaissent sur les images décompressées. [Cziho, 1999; Mercier *et al.*, 2003; Delgorge, 2005; Rodrigues, 2006].

- Les algorithmes sans pertes paraissent préférables à première vue. Cependant, ils ne peuvent être appliqués sur les images satellitaires pour leurs complexités. Le standard JPEG sans pertes est l'une des applications de la compression. Ce type de compression avec une reconstruction exacte de l'image de départ, garantissant l'intégrité des données, demeure le préféré des analystes pour des raisons évidentes d'analyse. Cependant, il offre de faibles performances en terme de débit binaire et les taux de compression ( $C$ ) potentiels varient suivant le contenu informatif de l'image. Les algorithmes sans pertes présentent donc des limitations assez importantes en terme de compression. Ces limitations sont dues, en partie, au bruit introduit par les capteurs haute résolution. En pratique, les méthodes sans pertes basées sur les codeurs entropiques restent les plus utilisées. Ces codeurs ont été mis en œuvre pour coder les pixels de l'image avec la contrainte d'obtenir des mots de code de longueur aussi proche que possible de l'entropie de l'image [Shannon, 1948]. Les codeurs les plus utilisés en compression d'images sont le codeur prédictif, les codeurs à longueurs variables (Huffman et arithmétique) [Huffman, 1952], ceux à base de dictionnaires (LZW) [Ziv *et al.*, 1977] et les codeurs par longueur de plages (RLE) [Lahdir, 2008].

- Afin d'atteindre de forts taux de compression, les méthodes avec pertes sont utilisées. Celles-ci emploient l'un des codeurs entropiques que nous avons décrit précédemment qui est associé, soit à une quantification (méthodes spatiales), soit à une transformation et une quantification (méthodes par transformation). L'un des avantages de ces algorithmes est qu'ils permettent en général d'ajuster le débit. Tolérer des distorsions assez importantes, c'est pouvoir comprimer plus (débit plus faible) et inversement. Cependant, le principal problème de cette compression pour les images, d'une façon générale, est dû, au fait que, des détails importants pourraient disparaître (d'autres pourraient éventuellement apparaître). Ces détails sont généralement des structures difficiles à discerner car elles entraînent de faibles changements de contraste. Certaines applications en imagerie n'ont pas nécessairement besoin de la même précision que les applications de diagnostic (domaine médicale par exemple). Nous pouvons donc raisonnablement envisager une compression avec pertes maîtrisées pour ce type d'application. Les méthodes les plus efficaces sont basées sur de puissants outils tels que : la transformée discrète en cosinus (DCT), la transformée en ondelettes ou l'approche fractale [Davoine, 1995 ; Marsault, 1995 ; Averbuch *et al.*, 1996]. Avec de tels outils, le fichier image subit une forte compression mais avec des pertes se traduisant par des dégradations plus ou moins perceptibles à l'œil nu lors de la décompression. Parmi les algorithmes à fort taux de compression, nous distinguons le standard JPEG (*Joint Photographic Expert Group*) [Borko, 1995], qui est actuellement, le plus couramment employé. Son principe est de diviser l'image en blocs de 8 x 8 pixels et de coder les valeurs les plus proches dans chaque bloc sur quelques bits. En procédant de la sorte, nous pouvons atteindre des taux de compression supérieurs à 90 %, tout en restituant des images avec une perte d'informations à peine perceptible. Toutefois, à l'instar d'autres algorithmes, le temps d'exécution est très lent surtout dans le cas d'images de grande taille. De plus, des artefacts apparaissent dans celles-ci sous forme de mosaïques. Dans le but de réduire les distorsions causées par les méthodes à fort taux de compression, une méthode de compression basée sur la DCT et la quantification vectorielle a été développée au laboratoire LAMPA. Cette technique appliquée à des images de références comme l'image Lena ou les images de Brodatz et à des images de télédétection prises par Météosat 4, a donné des résultats satisfaisants [Ameur *et al.*, 2002]. En effet, elle a permis de réaliser un bon compromis entre le taux de compression et la qualité de restitution des images. Grâce à cette méthode, le taux de compression et le PSNR atteignent respectivement 94 % et 33 dB. Quoique, l'apparition de nouvelles technologies de

communication impose la manipulation d'un volume de données plus important, qui doit être traité dans des délais de plus en plus courts, avec le meilleur compromis possible entre le taux de compression et la qualité de restitution des images. Cependant, l'utilisation de ces algorithmes engendre des problèmes très complexes dans le monde de la télédétection, comme par exemple la question de responsabilité dans le cas d'un mauvais diagnostic issu d'une image décompressée. Malgré cela, de nombreux chercheurs ont développé un certain nombre de méthodes hybrides ; particulièrement, celles qui sont basées sur les ondelettes, dans le but d'avoir un taux de compression élevé, une meilleure qualité de restitution de l'image et un temps de calcul minimal lors du processus de codage. Les résultats obtenus sur l'image test Lena sont répertoriés dans le tableau 1. Le but de la comparaison est d'identifier et de cerner les méthodes qui conduisent aux critères de performances (un taux de compression et un PSNR) les plus élevés, avec le temps d'exécution le plus court possible.

<b>Types d'algorithmes</b>	<b>PSNR (dB)</b>	<b>Taux de compression C (%)</b>
Ondelettes et fractales sans itération [Lahdir <i>et al.</i> , 2006]	32,90	93,75
DCT et QV en sous bandes [Ameur <i>et al.</i> , 2002]	32,20	93,75
JPEG [Charier <i>et al.</i> , 1999]	28,89	93,75
Ondelettes et fractales multirésolutions [sadoun <i>et al.</i> , 2006]	30,08	72,36
Fractales et les Algorithmes Génétiques [Lahdir <i>et al.</i> , 2008]	34,15	85.76

**Tableau 1** Comparaison des résultats obtenus sur l'image Lena.

### **1.3. Discussion**

Dans ce chapitre, nous avons fait le point sur la compression et présenté brièvement les principales méthodes de compression. Lors de l'application des méthodes avec pertes, les taux de compression obtenus varient entre 80 % et 95 %. Ces méthodes éliminent certaines informations qui ne sont pas indispensables pour l'appréciation visuelle des images par un œil humain. Les pertes dues à cette compression se traduisent par des artefacts ou des distorsions. Ces dernières deviennent plus importantes avec l'augmentation du taux de compression. Les résultats illustrés dans tableau 1 montrent que l'association des ondelettes avec les diverses techniques de compression, réalise de bons résultats. En effet, les propriétés de l'approche multirésolution basée sur la transformée en ondelettes peuvent rendre très intéressante la deuxième étape du schéma général de compression d'images. La transformée projette les pixels d'une image sur une variété de bases d'ondelettes, permettant ainsi un bon compactage d'énergie sur un nombre réduit de coefficients. La compression d'images basée sur cette approche effectue une reconstruction progressive de l'image ; ce qui nous permet d'avoir une image décompressée à n'importe quelle résolution.

Dans le but de développer de nouvelles approches de compression ou des versions améliorées de celles existantes, il est important d'exploiter les avantages des différentes méthodes présentées précédemment. En effet, les performances d'une méthode de compression dépendent des techniques utilisées dans les étapes de transformation, de quantification et de codage. Tenant compte de toutes ces constatations, et fixant comme objectif : l'obtention du meilleur compromis possible entre le taux de compression, la qualité de l'image restituée et le temps de calcul, nous avons élaboré une méthode de compression basée sur un codage par arbres de zéros dans le domaine des ondelettes, plus particulièrement les ondelettes entières. Cette méthode qui sera appliquée à des images météorologiques sera décrite dans le chapitre 3.

# *Chapitre II*

## *La Transformée en ondelettes entières*

### **2.1 Préambule**

Les transformations linéaires ont toujours joué un très grand rôle en traitement du signal [Ayad, 2007]. Le but d'une transformation est de projeter un signal sur une base de fonctions dont les propriétés sont adaptées à la nature et aux caractéristiques du signal à analyser. La projection est généralement orthogonale. La première transformation mathématique employée pour analyser les signaux est la transformée de Fourier (TF). Celle-ci occupe une place centrale, notamment dans le domaine du traitement du signal, en raison de l'universalité liée au concept de fréquence et de son optimalité pour traiter les signaux stationnaires. Malgré cela, elle rend difficile l'analyse des signaux dits transitoires [Guillemot, 2004]. Dans la majorité des cas, le signal doit être pris dans sa globalité. En revanche, la transformée de Fourier ne permet pas de représenter des événements concentrés dans le temps ou dans l'espace, de manière équivalente pour les signaux 2D. Or, la plupart des signaux réels tels que les sons ou les images ont des caractéristiques non stationnaires. Pour traiter ces types de signaux transitoires, nous pouvons découper le signal en plusieurs morceaux sur lesquels sera appliquée une TF (ou une transformation dérivée) : l'hypothèse que le signal est "stationnaire par morceaux" [Ayad, 2007]. C'est la méthode employée en particulier dans le standard de compression JPEG avec la transformée en cosinus par bloc [Ammar, 2002 ; Akrou, 2003]. Toutefois, même si cette stratégie donne de bons résultats, elle présente l'inconvénient de créer des artefacts dus au découpage artificiel en blocs.

La seconde approche consiste à utiliser une base de fonctions qui ont une localisation temporelle (ou spatiale dans le cas des images) et fréquentielle. Obtenir une localisation

fréquentielle consiste à utiliser une fonction dont l'énergie est localisée sur un intervalle particulier, c'est-à-dire une fonction dont la valeur est négligeable en dehors de cet intervalle. Une base de fonctions capable d'analyser en temps et fréquence tout le signal qui peut être obtenu par translation de la fonction utilisée. La taille du support est un point très important car c'est lui qui détermine la résolution temporelle. Il existe de nombreuses recherches liées à la détermination de ce type de base et à la généralisation de ce principe. Un des points culminants de ces travaux est la théorie des ondelettes.

La transformée en ondelettes (TO) [Daubechies, 1992] est un outil largement utilisé en traitement du signal et d'image. L'une de ses applications principales est la compression d'images du fait sa capacité à compacter l'énergie sur un petit nombre de coefficients permettant un codage efficace de l'image.

Elle correspond à la première étape de la chaîne de compression classique. Elle est couramment utilisée dans le cas des images 2D par application de filtres séparables dans les deux directions (axes x et y) [Antonini, 1992]. L'objectif de cette partie est de présenter les deux principales façons de calculer les coefficients d'ondelettes : par convolution classique et par lifting.

## 2.2. La Transformée en ondelettes par convolution classique :

Les fonctions  $\psi_{(a,b)}$  qui composent une base d'ondelettes  $\{\psi_{(a,b)}(t)\}_{(a,b)}$  sont définies de la manière suivante [Savaton, 1999] :

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (2.1)$$

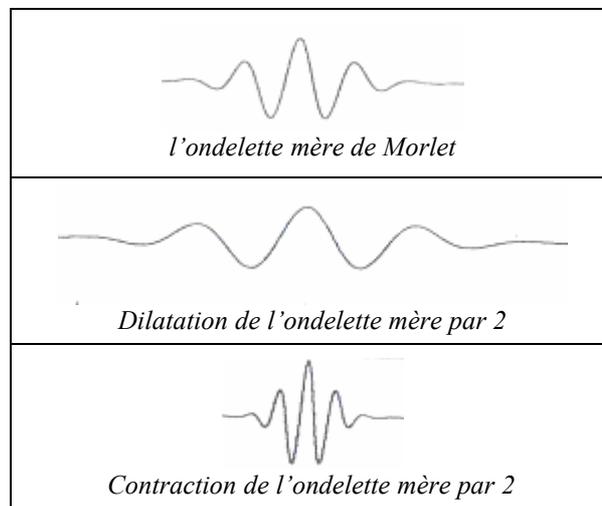
Avec :  $a$  est appelé facteur d'échelle et  $b$  est le paramètre de translation.

Une base d'ondelettes est obtenue par translation et dilation d'une unique fonction  $\psi$ , appelée ondelette mère. Nous obtenons de cette manière une analyse où la résolution fréquentielle et temporelle est variable. C'est grâce à cette propriété que la théorie des ondelettes a connu un tel succès : faire varier  $b$  permet d'analyser localement le signal, et ce, à différentes échelles grâce à  $a$ . Cette échelle  $a$  permet de faire varier la résolution temps-fréquence de l'analyse en faisant varier la taille du support de  $\psi_{(a,b)}$  d'un rapport de  $\frac{1}{a}$ . Ensuite, nous nous intéresserons

à définir et à décrire les principales propriétés des ondelettes qui ont fait de cet outil un élément central dans les schémas actuels de compression d'images.

### 2.2.1. La transformée en ondelettes continue (CWT)

La transformation en ondelette propose une gamme de fenêtre dont le support varie avec la résolution des détails que nous souhaitons mettre en évidence. Une telle fenêtre est appelée ondelette, c'est une fonction oscillante d'intégrale nulle à décroissance rapide ; elle peut être dilatée ou contractée selon sa longueur (comme un accordéon) cette propriété est illustrée par la figure 3.



**Fig .3** L'ondelette de Morlet après dilatation et contraction

La transformée en ondelette continue  $C_f(a,b)$  d'une fonction  $f$  est définie comme le produit scalaire de  $f$  avec l'ondelette mère  $\psi_{ab}$  [Mayer, 1992 ; Langlet, 2004 ; Chaux, 2006] :

$$C_f(a,b) = \langle f, \psi_{ab} \rangle = \int_{-\infty}^{+\infty} f(t) \overline{\psi_{ab}(t)} dt \quad (2.2)$$

Les coefficients  $C_f(a,b)$  sont appelés coefficients d'ondelettes. Cette transformation permet de mesurer la corrélation entre la fonction  $f$  étudiée et l'ondelette mère  $\psi_{ab}$  pour différentes largeurs de son support. Lorsque  $a$  croît, le support des ondelettes  $\psi_{ab}$  s'élargit et les coefficients  $C_f(a,b)$  traduisent la présence de détails de moins en moins fins.

La transformation en ondelette inverse de  $C_f(a, b)$  s'écrit :

$$f(t) = \iint_{\mathbb{R}^2} C_f(a, b) \psi_{ab}(t) \cdot \frac{da \cdot db}{a^2} \quad (2.3)$$

### 2.2.2 La transformée en ondelettes discrète (DWT)

Afin d'éviter les problèmes de redondance de la CWT, et les calculs d'intégrales de la transformée en ondelettes continue, nous définissons la transformée en ondelettes discrète [Langlet, 2004]. Ce qui nous amène à chercher une nouvelle base de représentation, une base d'ondelettes  $(\psi_{jk})_{(j,k) \in \mathbb{Z}^2}$ . En 1987, Y. Meyer a démontré qu'il était possible de construire des bases d'ondelettes orthonormales en discrétisant les paramètres de dilatation et de translation  $a$  et  $b$  en utilisant l'équation (2.4) :

$$(a, b) = (a_0^j, kb_0 a_0^j) \quad (2.4)$$

Avec :  $\begin{cases} (j, k) \in \mathbb{Z}^2 \\ a_0 > 1 \text{ et } b_0 > 0 \end{cases}$  : pas de dilatation et de translation respectivement

Ainsi, la nouvelle famille d'ondelettes peut s'écrire :

$$\psi_{j,k}(t) = a_0^{-j/2} \psi(a_0^{-j} t - kb_0) \quad (2.5)$$

Avec  $(j, k) \in \mathbb{Z}^2$

La transformée en ondelettes discrète d'une fonction est donnée par l'équation 2.6 :

$$C_f(j, k) = \langle f, \psi_{j,k} \rangle = a_0^{-j/2} \int_{-\infty}^{+\infty} \psi(a_0^{-j} t - kb_0) f(t) dt \quad (2.6)$$

Meyer a montré qu'il existe des familles d'ondelettes discrètes formant des bases orthonormées de  $L^2(\mathbb{R})$  [Daubechies, 1988 ; Meyer, 1992].

Les paramètres de dilatation et de translation sont choisis comme suit :

$$(a, b) = (a_0^j, k b_0 a_0^j) = (2^{-j}, k 2^{-j}) \quad \text{Avec : } a_0=2 \text{ et } b_0=1$$

Nous obtenons ainsi des bases dans  $L^2(\mathbb{R})$  de la forme :

$$\{\psi_{j,k}\}_{j,k \in \mathbb{Z}} = \left\{ 2^{j/2} \psi(2^j t - k) \right\}_{j,k \in \mathbb{Z}} \quad (2.8)$$

La décomposition de  $f(t) \in L^2(\mathbb{R})$  peut s'écrire alors :

$$f(t) = \sum_{j,k \in \mathbb{Z}} \langle f(t), \psi_{j,k}(t) \rangle \psi_{j,k}(t) \quad (2.9)$$

$$\text{où} \quad \langle f(t), \psi_{j,k}(t) \rangle = 2^{-j/2} \int_{-\infty}^{+\infty} f(t) \bar{\psi}_{j,k}(2^{-j} t - k) dt$$

$\langle f(t), \psi_{j,k}(t) \rangle$  Représente les coefficients d'ondelettes qui sont décorrélés entre eux.

La fonction continue  $f$  est alors entièrement représentée par la fonction discrète  $C_f(j, k)$  :

$$C_f(j, k) = \langle f, \psi_{jk} \rangle \quad (2.10)$$

### 2.2.3 Analyse Multirésolution

L'utilisation des ondelettes à travers l'analyse multirésolution permet la construction des bases orthonormées. Cette méthode offre la possibilité de représenter un signal (une image) à des échelles différentes, c'est-à-dire, représenter les signaux avec des niveaux de résolutions différents ou encore avec des niveaux de détails différents. Une analyse multirésolution prend pour point de départ une fonction d'échelle  $\varphi$  et une ondelette mère  $\psi$  qui doivent vérifier les propriétés de dilations suivantes :

$$\varphi(t) = 2 \sum_{k=-\infty}^{+\infty} h_k^0 \cdot \varphi(2t - k) \quad (2.11)$$

$$\psi(t) = 2 \sum_{k=-\infty}^{+\infty} h_k^1 \cdot \varphi(2t - k) \quad (2.12)$$

L'analyse multirésolution est définie comme étant une suite de sous-espaces  $(V_j)_{j \in \mathbb{Z}}$  de  $L^2(\mathbb{R})$  construite à partir de  $\varphi$  en vérifiant les propriétés suivantes :

$$V_j : \begin{cases} v(t) \in V_j \Leftrightarrow v(2t) \in V_{j+1} \\ V_{j+1} \supset V_j \\ \lim_{j \rightarrow +\infty} V_j = L^2(\mathbb{R}) \\ \lim_{j \rightarrow -\infty} V_j = 0 \end{cases}$$

Examinons plus en détail chacune de ces propriétés :

- $v(t) \in V_j \Leftrightarrow v(2t) \in V_{j+1}$

Pour construire  $V_{j+1}$ , il suffit d'opérer une contraction par deux du support de toutes les fonctions de  $V_j$ ,  $V_{j+1}$  est donc une réplique de  $V_j$  ( $V_{j+1}$  est l'image de  $V_j$ ) à une résolution double. Le paramètre  $j$  caractérise la finesse des détails d'une fonction de  $V_j$

- $V_{j+1} \supset V_j$

Les espaces  $V_j$  sont appelés espaces d'approximations de  $L^2(\mathbb{R})$ . Un signal  $f_{j+1}$  appartenant à  $V_{j+1}$  est susceptible de contenir des détails à toutes les résolutions inférieures.

- $\lim_{j \rightarrow -\infty} V_j = L^2(\mathbb{R})$

Une fonction quelconque de  $L^2(\mathbb{R})$ , avec des détails arbitrairement fins, est considérée comme ayant une résolution infinie. La projection d'un signal  $f$  de  $L^2(\mathbb{R})$  sur un espace  $V_j$  revient à calculer une approximation à la résolution  $j$ .

- $\lim_{j \rightarrow +\infty} V_j = \{0\}$

La fonction nulle est la seule fonction de résolution infiniment basse, elle ne contient aucun détail. Elle appartient par conséquent à tous les espaces d'approximations.

Les espaces d'approximation sont construits de telle sorte qu'il existe pour chaque sous-espace  $V_j$  une base orthonormée  $(\varphi_{jk})_{j,k \in \mathbb{Z}^2}$  telle que :

$$\varphi_{jk} = 2^{j/2} \varphi(2^j t - k) \quad (2.13)$$

L'approximation à la résolution  $j$  d'une fonction  $f_{j+1}$  de  $V_{j+1}$  se calcule en projetant  $f_{j+1}$  sur  $V_j$ , ce qui nous donne :

$$f_j(t) = \sum_{k=-\infty}^{+\infty} \langle f_{j+1}, \varphi_{jk} \rangle \varphi_{jk}(t) \quad (2.14)$$

L'erreur due à l'approximation est contenue dans un espace de détails  $W_j$  supplémentaire orthogonal de  $V_j$  dans  $V_{j+1}$  :

$$V_{j+1} = V_j \oplus W_j \quad (2.15)$$

Nous montrons que pour chaque espace  $W_j$ , il existe une famille d'ondelettes  $(\psi_{jk})_{k \in \mathbb{Z}}$  qui forment une base orthonormée. Ces ondelettes sont de la forme :

$$\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k) \quad (2.16)$$

Ainsi, la décomposition de  $f_{j+1}$  sur la base  $(\psi_{jk})$  représente les détails  $g_j$  que l'approximation à une échelle plus grossière n'a pas permis de conserver.

$$g_j(t) = \sum_{k=-\infty}^{+\infty} \langle f_{j+1}, \psi_{jk} \rangle \psi_{jk}(t) \quad (2.17)$$

$$f_{j+1}(t) = f_j(t) + g_j(t) \quad (2.18)$$

En itérant le processus décrit par la relation (2.15), nous obtiendrons :

$$V_n = \bigoplus_{j=-\infty}^{n-1} W_j \quad (2.19)$$

$$L^2(\mathfrak{R}) = \bigoplus_{j=-\infty}^{+\infty} W_j \quad (2.20)$$

Les bases d'ondelettes  $(\psi_{jk})_{k \in \mathbb{Z}}$  sont orthogonales entre elles, et leur réunion constitue une base orthonormée de  $L^2(\mathfrak{R})$ .

Les signaux numériques ont une résolution finie  $n$  correspondant au pas d'échantillonnage. Dans la pratique, le calcul de la transformée en ondelettes d'un signal  $f_n$  se fait par projections successives de  $f_n$  sur les espaces  $V_{n-1}, V_{n-2}, \dots, V_m$ , jusqu'à une résolution  $m$  aussi faible que souhaitée. Les signaux détails  $g_{n-1}, g_{n-2}, \dots, g_m$  calculés à chaque étape

représentent une partie de la décomposition en ondelettes de  $f_n$ . Pour reconstruire fidèlement le signal d'origine, il est nécessaire de conserver l'approximation  $f_m$  à la résolution  $m$  :

$$f_n(t) = f_m(t) + \sum_{j=m}^{n-1} g_j(t) \quad (2.21)$$

Un exemple de représentation multirésolution d'une image est donné par la figure 4.



**Fig. 4** Représentation multirésolution de l'image Lena à la résolution trois

#### 2.2.4. Choix d'ondelettes

Il n'y a pas une ondelette qui soit meilleure qu'une autre. Tout dépend de l'application utilisée. Dans certains cas, l'ondelette la plus simple (Haar) sera optimale. Pour d'autres applications, ce sera le pire des choix possibles. En pratique, il semblerait que l'élément le plus important soit le nombre de moments nuls. Pour la plupart des applications, il est désirable d'avoir le plus de coefficients d'ondelettes nuls et donc plus de moments nuls implique une meilleure transformation [Savaton, 1999 ; Baarir *et al*, 2004]. Cependant, les ondelettes ayant un plus grand nombre de moments ont aussi un support plus grand ce qui signifie, que si la fonction ou le signal a des discontinuités brusques, elle sera plus sujette aux phénomènes de Gibbs. De là, l'efficacité intrinsèque d'une ondelette  $\psi$  peut être défini par le nombre  $N$  de ces premiers moments nuls :

$$N = \max \left\{ n \in \mathbb{N} / \int_{-\infty}^{+\infty} t^n \psi(t) dt = 0 \right\} \quad (2.22)$$

Toutes les ondelettes ont au moins un moment nul d'ordre 1. En résumé, Plus le nombre de moments nuls d'une ondelette mère est grand, plus celle-ci est capable de représenter de manière compacte des signaux complexe. Ainsi, la détermination de l'ondelette permettant d'approximer au mieux une image repose sur deux critères :

- **Représentation creuse** : Les grands coefficients d'ondelettes apparaissent lorsqu'il y a une variation brutale du signal tel qu'un contour. Le nombre de coefficients de grande magnitude doit être le plus petit possible afin d'obtenir une représentation du signal la plus concise possible. Cette propriété est déterminée par la taille du support de l'ondelette qui doit être la plus petit possible. Par ailleurs, les coefficients d'ondelettes représentant les zones homogènes doivent avoir une faible magnitude. Cette propriété est obtenue lorsque l'ondelette admet un grand nombre de moments nuls (une ondelette  $\psi$  de moment  $N$  signifie que le produit scalaire entre  $\psi$  et un polynôme de degré  $n \leq N$  est nul).
- **Réduction des artefacts** : Les dégradations induites par la compression des coefficients d'ondelettes doivent avoir le minimum de répercussions (artefacts) dans le domaine spatial. La régularité de l'ondelette permet de réduire cet impact visuel. Enfin, parce qu'elles évitent l'apparition d'effets de bord ainsi que l'expansion des coefficients (nombre de coefficients de la transformée supérieur à celui du signal), les bases biorthogonales sont les mieux adaptées.

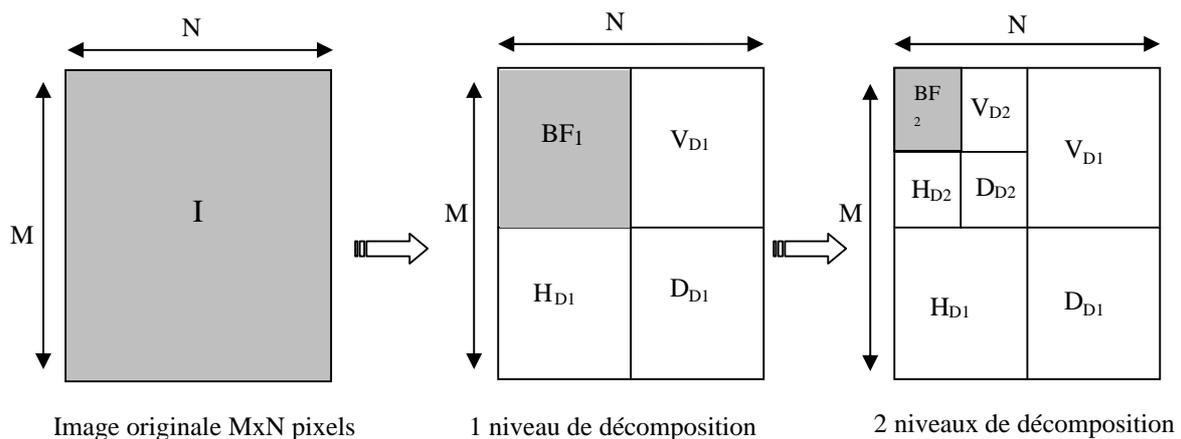
L'ondelette biorthogonale offrant le meilleur compromis entre moments nuls, support et régularité est réalisé par le filtre 9/7 de Daubechies [Antonini, 1999 ; Guillemot, 2004 ; Gaudau, 2006]. L'ondelette 9/7 (9 et 7 étant la longueur des filtres passe-bas et passe-haut) est aussi appelée (4,4) (les valeurs correspondant aux nombres de moments nuls des filtres passe-haut et passe-bas). Cette ondelette est quasiorthogonale et donc particulièrement appréciée dans le cas de la compression avec pertes, c'est d'ailleurs, l'ondelette par défaut pour la compression avec pertes de JPEG 2000 [Christophe, 2006]. Tandis que, l'ondelette 5/3 (5 et 3 étant les longueurs des filtres passe-bas et passe-haut) est aussi appelée (2,2) (les valeurs correspondant ici aux nombres de moments nuls des filtres passe-haut et passe-bas). Cette ondelette semble donner de meilleures performances pour la compression sans pertes,

c'est l'ondelette par défaut du standard JPEG 2000 pour la compression réversible [Averbuch et al, 2004 ; Christophe, 2006].

### 2.2.5. Application aux signaux bidimensionnels

La transformée en ondelettes a d'abord été étendue aux images 2D, en appliquant de façon séparée les filtres d'analyse et de synthèse sur les lignes et les colonnes [Antonini *et al*, 1992]. Par conséquent, l'analyse multirésolution appliquée à une image  $I$  conduit à quatre sous-images ou sous-bandes :

- Une sous-image notée BF représentant l'approximation de  $I$  à l'échelle  $2^{-1}$ .
- Trois sous-images  $V$ ,  $H$  et  $D$  correspondant au résultat du filtrage, respectivement, horizontal, vertical et diagonal (application des filtres verticaux et horizontaux). Ces sous-images contiennent les détails orthogonaux à la direction du filtrage.

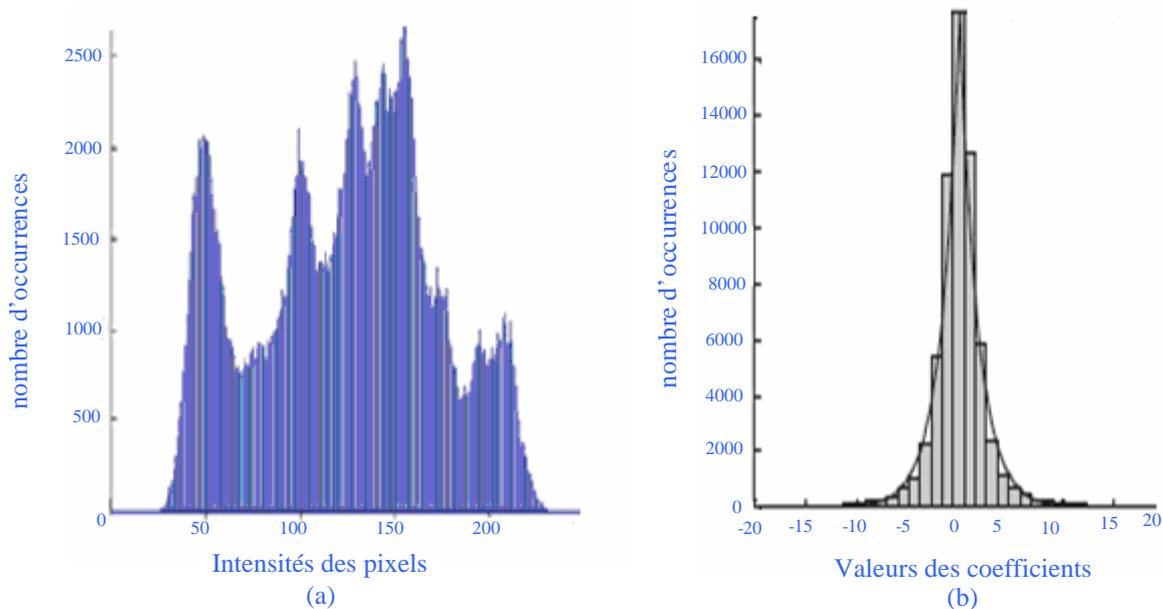


**Fig. 5** Analyse multirésolution à deux niveaux de décomposition d'une image  $I$

Nous obtenons ainsi une représentation de l'image au niveau de résolution 1. En répétant cette opération sur la sous-image BF, nous obtiendrons une représentation du signal à un deuxième niveau de résolution. Le nombre de résolution utilisé est généralement compris entre 3 et 5 dans les applications de compression. La figure 5 représente le résultat de l'analyse multirésolution d'une image sur deux niveaux de décomposition.

Le succès des ondelettes en traitement d'images, et particulièrement en compression, est essentiellement basé sur deux points [Guillemot, 2004 ; Gaudeau, 2006] :

1. Les ondelettes permettent de représenter, à une échelle donnée, les variations brutales d'une image sur un faible nombre de coefficients. Cette propriété permet de délivrer une représentation creuse du signal (Fig.6).
2. Les images peuvent présenter des propriétés d'autosimilarités à différents niveaux d'échelles qui sont préservées par la transformée en ondelettes. C'est ce que nous appelons le zoom multiéchelle [Mallat, 2000]. Il existe donc des dépendances entre les représentations de l'image à différentes échelles. Cette propriété est à la base des méthodes de compression de type arbres de zéro que nous décrirons un peu plus loin.



**Fig. 6** (a) Histogramme des intensités des pixels d'une image - (b) Histogramme des coefficients d'ondelettes d'une sous image (autre que la sous image basse fréquence).

A partir de ces propriétés, l'idée de la Transformée en ondelettes est de décomposer un signal quelconque  $f(t)$  en une superposition d'ondelettes. Inversement, nous pouvons reconstruire  $f$  à partir de sa fonction mère et des coefficients d'ondelettes obtenus lors de la décomposition. Dans un cadre d'analyse multirésolution, la transformée en ondelettes discrète peut être vue comme une décomposition du signal à différents niveaux de résolution, et peut être implantée en utilisant un couple de filtres discrets à travers un schéma classique d'analyse et de synthèse ( Fig.7).

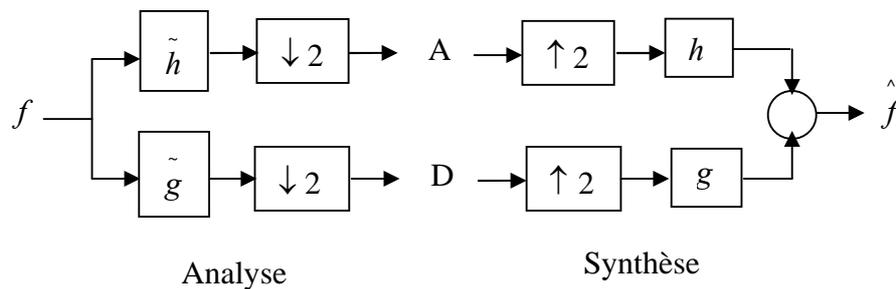


Fig. 7 Transformée en ondelettes à une décomposition : Analyse et Synthèse

Dans l'étape d'analyse, le signal discret  $f(t)$  est décomposé en 2 signaux, un signal d'approximations 'A' en utilisant un filtre passe-bas  $\tilde{h}$  et un signal de détails 'D' en se servant d'un filtre passe-haut  $\tilde{g}$ .

Cette décomposition est suivie par une décimation d'un facteur 2. Cela peut être représenté avec les équations suivantes [Mallat, 1989] :

$$A(t) = \sum_k \tilde{h}(2t - k) f(k) \quad (2.23)$$

$$D(t) = \sum_k \tilde{g}(2t - k) f(k) \quad (2.24)$$

Le nombre total de coefficients de  $A(t)$  et  $D(t)$  est égal au nombre de coefficients dans  $f(t)$ . Dans l'étape de synthèse,  $A(t)$  et  $D(t)$  sont sur-échantillonnés et filtrés avec  $h$  et  $g$ . La somme des sorties du filtre produit le signal reconstruit  $\hat{f}(t)$ .

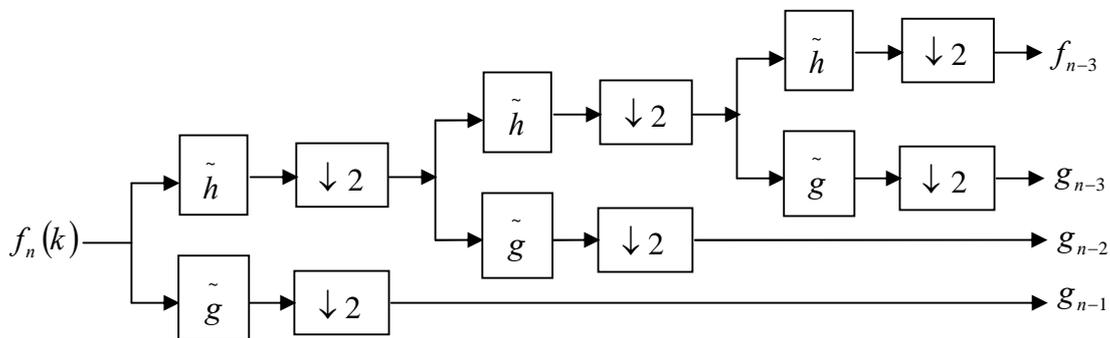
Afin de pouvoir reconstruire un signal à partir de sa transformée en ondelettes, les bancs de filtres de décomposition et de reconstitution doivent être liés par les relations qui suivent [Savaton, 1999] :

$$h(z)\tilde{h}(z^{-1}) + g(z)\tilde{g}(z^{-1}) = 2z^{-1} \quad (2.25)$$

$$h(z)\tilde{h}(-z^{-1}) + g(z)\tilde{g}(-z^{-1}) = 0 \quad (2.26)$$

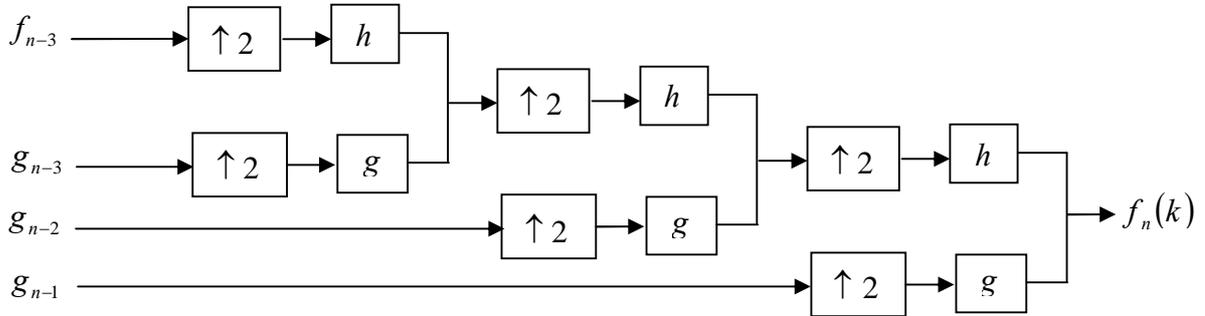
La condition (2.25) permet de s'affranchir du repliement du spectre produit par le sous-échantillonnage. L'équation (2.26), en revanche, assure la reconstitution parfaite du signal, à un décalage temporel près. Dans le but de simplifier les algorithmes de calculs des transformations en ondelettes, il est souhaitable de travailler avec des filtres à réponse impulsionnelle finie (RIF) [Savaton, 1999].

En conséquence, la décomposition en ondelettes se poursuit vers les résolutions les plus grossières en appliquant, pour chaque niveau, les filtres  $\tilde{h}$  et  $\tilde{g}$  à l'image d'approximation de la décomposition précédente. Le filtre  $\tilde{g}$  a pour objectif de conserver les composantes spectrales non retenus par  $\tilde{h}$ . Les deux filtres utilisés pour la décomposition sont fixes à toutes les échelles. Il faut donc, d'une échelle à la suivante, contracter le support des signaux analysés. Dans le cas d'un signal discret  $f_n(k)$ , cela se traduit par une décimation (sous échantillonnage) avec un pas de deux (Fig.8).



**Fig. 8** Décomposition par bancs de filtres sur trois niveaux

La transformation inverse s'effectue à l'aide d'un banc de filtres  $(h, g)$  dual du précédent en utilisant un sur échantillonnage d'un pas de deux (Fig.9).



**Fig. 9** Reconstitution par bancs de filtres sur trois niveaux

Notons bien, que nous avons vu précédemment que la fonction d'échelle et l'ondelette mère vérifient les propriétés de dilatation (les équations (2.11) et (2.12)). Ces relations régissent le passage d'une résolution quelconque à la résolution immédiatement inférieure [Savaton, 1999]. En posant dans l'équation (2.11) le changement de variable  $u = 2t$ , nous obtiendrons :

$$\frac{1}{2}\varphi\left(\frac{u}{2}\right) = \sum_{k=-\infty}^{+\infty} \tilde{h}_k \varphi(u-k) \quad (2.27)$$

Nous constatons clairement sur cette équation que son membre de droite est le produit de convolution de  $\varphi$  par la réponse impulsionnelle du filtre  $\tilde{h}$  ; tandis que, le membre de gauche reflète le changement d'échelle de la résolution la plus fine vers la résolution la plus grossière. La fonction (2.12) peut être matérialisée de la même manière par l'action sur  $\psi$  d'un filtre discret  $\tilde{g}$ .

### 2.3. La Transformée en ondelettes par schéma de lifting (LWT) :

Daubechies et Sweldens [Daubechies *et al*, 1998] ont utilisé un schéma appelé lifting pour calculer la transformée en ondelettes discrète. Ils ont montré que n'importe quelle transformée en ondelettes pouvait être calculée à partir de ce schéma. Celui-ci permet de réduire le coût calculatoire par rapport à l'implantation par convolution, car, il utilise les redondances

mathématiques entre le filtre passe-bas et le filtre passe-haut. Son second avantage est sa capacité à produire des coefficients d'ondelettes entières. Cette fonctionnalité est fondamentale pour pouvoir envisager des méthodes de compression sans perte, ou allant d'un schéma avec pertes vers du sans perte. La transformée par lifting consiste dans un premier temps à séparer les échantillons pairs et impairs dans 2 tableaux différents [Gaudeau, 2006]. Si  $f(n)$  est le signal d'entrée, ce découpage, appelé aussi transformée polyphase, se note :

$$s^{(0)}(n) = f(2n) \tag{2.28}$$

$$d^{(0)}(n) = f(2n + 1) \tag{2.29}$$

Nous appliquons ensuite une suite d'opérateurs de prédiction et de mise à jour pour obtenir les coefficients haute et basse fréquences de la transformée. Ces deux étapes forment un étage de lifting. Elles sont aussi appelées "pas de lifting dual" et "pas de lifting". Elles s'écrivent à l'itération :

$$d^{(i)}(n) = d^{(i-1)}(n) - \sum_k P^{(i)}(k) s^{(i-1)}(n - k) \tag{2.30}$$

$$s^{(i)}(n) = s^{(i-1)}(n) - \sum_k U^{(i)}(k) d^{(i)}(n - k) \tag{2.31}$$

où les coefficients  $P^{(i)}(k)$  et  $U^{(i)}(k)$  sont calculés en utilisant une factorisation Euclidienne de la matrice de polyphase [Savaton, 1999 ; Gaudeau, 2006].

La figure 10 représente la transformation en ondelettes par schéma de lifting

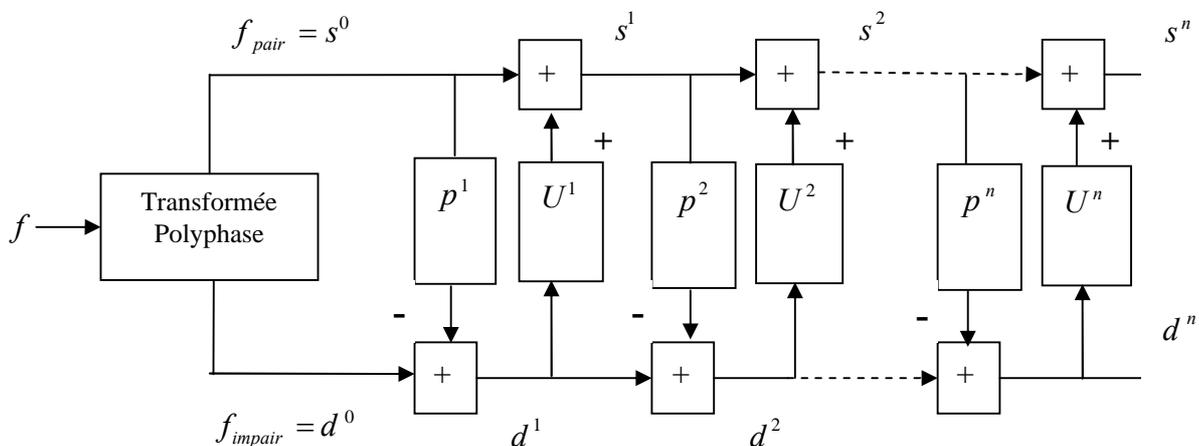


Fig. 10 Transformation en ondelettes par schéma de lifting

Nous constatons à partir de la figure 10 que le schéma de lifting, est constitué principalement de trois étapes essentielles :

### 2.3.1. La transformée polyphase :

La transformée polyphase en deux bandes est une opération très simple ; elle prend un signal d'entrée qu'elle sépare en deux composantes polyphases : paires et impaires. Plus exactement, l'application de cette transformée au signal original  $f$ , va simplement partitionner  $f$  en deux sous ensembles disjoints  $f_p$  (l'ensemble des échantillons appartenant à  $f$  d'indice pair) et  $f_i$  (l'ensemble des échantillons appartenant à  $f$  d'indice impair).

Transformée polyphase :  $f \rightarrow (f_p, f_i)$

### 2.3.2. Opération de prédiction :

La plupart des signaux en imagerie présente une structure de corrélation locale, à la fois, dans les domaines spatial et fréquentiel. En d'autres termes, les échantillons ou les fréquences d'un même voisinage seront fortement corrélés, tandis que, les échantillons n'appartenant pas au même voisinage ne présenteront qu'un faible taux de corrélation. Si le signal  $f$  présente une telle structure, alors, les deux sous signaux  $f_p$  et  $f_i$  seront fortement corrélés ; il en résulte la possibilité de prédire  $f_i$  à partir de  $f_p$  afin de prédire le sous ensemble  $f_i$ . La différence  $d$  entre  $f_i$  et sa prédiction constitue le détail de l'image, autrement dit les coefficients d'ondelettes.

$$d = f_i - P(f_p) \quad (2.32)$$

La transformée en ondelettes permet d'exploiter de façon similaire ce type de corrélation. L'étape de prédiction revient à appliquer un filtre passe haut au signal  $f$  et un sous échantillonnage. L'opération qui calcule la prédiction et à mettre en mémoire le détail, constitue un pas de lifting dual. Ainsi, la deuxième étape du schéma de lifting nous ramène à :

$$\text{Pas de lifting dual : } (f_p, f_i) \rightarrow (f_p, d) \quad (2.33)$$

### 2.3.3. Opération de mise à jour :

A ce stade des opérations, nous disposons de deux ensembles  $f_p$  et  $d$ . L'ensemble  $d$  obtenu par un procédé assimilable à un filtrage passe haut plus un sous échantillonnage, reproduit les hautes fréquences du signal original. L'ensemble  $f_p$  quant à lui, peut être vu comme une représentation du signal à un niveau de résolution plus grossière. Il forme alors les basses fréquences du signal original. Cette opération de mise à jour, repose sur les constats antérieurs. L'idée est d'établir une représentation faible en cherchant à conserver certaines propriétés globales du signal original.

$$s = f_p + U(d) \quad (2.34)$$

Cette opération est appelée : un pas de lifting. Il est possible d'associer plusieurs pas de prédictions  $P^{(i)}$  et de mise à jour  $U^{(i)}$ , comme le montre la figure 10. Nous obtiendrons ainsi, des signaux intermédiaires ; des signaux filtrés passe haut  $d^i$  et des signaux filtrés passe bas  $s^i$ . Le grand intérêt pratique de cette méthode réside dans la simplicité de l'inversion de la transformée qui consiste à changer les additions en soustractions :

$$f_i = d + P(f_p) \quad (2.35)$$

$$f_p = s - U(d) \quad (2.36)$$

### 2.3.4 Représentation polyphase des bancs de filtres

Soit un signal  $f(k)$  de  $N$  échantillons, la décomposition de ce signal à l'aide de deux bancs de filtres  $\left(\tilde{h}, \tilde{g}\right)$  donne naissance à un signal basse fréquence  $f^0(k)$ , issu du premier filtre et un signal haute fréquence  $f^1(k)$  issu de seconde filtre, comprenant chacun  $N$  échantillons. Après un sous échantillonnage,  $\frac{N}{2}$  coefficients d'approximations  $s(k)$  ( $s$  pour somme) et  $\frac{N}{2}$  coefficients de détails  $d(k)$  ( $d$  pour détails) sont conservés [Savaton, 1999]. Sur  $2N$  valeurs calculées,  $N$  seulement seront exploitées. La structure polyphase des bancs de filtres permet de n'effectuer que la quantité de calculs nécessaires.

1. Ecrivons tout d'abord le calcul des coefficients d'approximations du signal source pour un niveau de décomposition :

- L'application du filtrage passe-bas nous permet d'avoir :

$$f^0(k) = \sum_l \tilde{h}_l f(k-l) \quad (2.37)$$

- Ensuite le sous échantillonnage, pour avoir :

$$s(k) = f^0(2k) = \sum_l \tilde{h}_l f(2k-l) \quad (2.38)$$

2. A ce stade des opérations, nous allons séparer dans le calcul de  $s(k)$ , les coefficients de  $\tilde{h}$  portant sur les échantillons d'indices pairs et impairs de  $f(k)$  :

$$s(k) = \sum_l \tilde{h}_{2l} f(2k-2l) + \tilde{h}_{2l+1} f(2k-2l-1) \quad (2.39)$$

En posant :

$$\begin{cases} f_p(k) = f(2k) & (\text{pair}) \\ f_i(k) = f(2k-1) & (\text{impair}) \end{cases}$$

La relation (2.39) devient :

$$s(k) = \sum_l \tilde{h}_{2l} f_p(k-l) + \tilde{h}_{2l+1} f_i(k-l) \quad (2.40)$$

3. Le calcul de  $s(k)$  revient donc à appliquer séparément sur  $f_p$  et  $f_i$  deux filtres  $\tilde{h}_p$  et  $\tilde{h}_i$  dont les transformées en  $z$  s'écrivent :

$$\tilde{H}_p(z) = \sum_l \tilde{h}_{2l} z^{-l} \quad (2.41)$$

$$\tilde{H}_i(z) = \sum_l \tilde{h}_{2l+1} z^{-l} \quad (2.42)$$

De ces dernières équations (2.41) et (2.42), nous pouvons définir la représentation polyphase du filtre  $\tilde{h}$  :

$$\tilde{H}(z) = \tilde{H}_p(z^{-1}) + z^{-1} \tilde{H}_i(z^{-1}) \quad (2.43)$$

Avec le formalisme de la transformation en  $z$ , les coefficients d'approximations de  $f$  s'obtiennent de la manière suivante :

$$S(z) = \tilde{H}_p(z)F_p(z) + \tilde{H}_i(z)F_i(z) \quad (2.44)$$

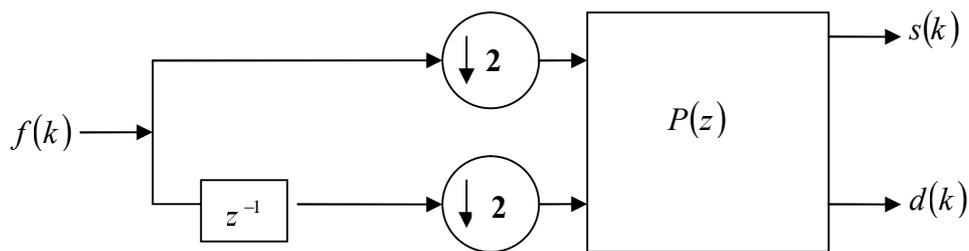
Nous définissons de même les filtres  $\tilde{G}_p$  et  $\tilde{G}_i$  pour les coefficients détails  $d$  :

$$D(z) = \tilde{G}_p(z)F_p(z) + \tilde{G}_i(z)F_i(z) \quad (2.45)$$

4. Réunissons enfin les équations (2.44) et (2.45) dans une relation matricielle :

$$\begin{bmatrix} S(z) \\ D(z) \end{bmatrix} = \begin{bmatrix} \tilde{H}_p(z) & \tilde{H}_i(z) \\ \tilde{G}_p(z) & \tilde{G}_i(z) \end{bmatrix} \cdot \begin{bmatrix} F_p(z) \\ F_i(z) \end{bmatrix} = P(z) \begin{bmatrix} F_p(z) \\ F_i(z) \end{bmatrix} \quad (2.46)$$

où  $P(z)$  est la matrice polyphase associée au banc de filtres  $(\tilde{h}, \tilde{g})$ . Cette relation est représentée sous la forme de schéma bloc sur la figure 11 :



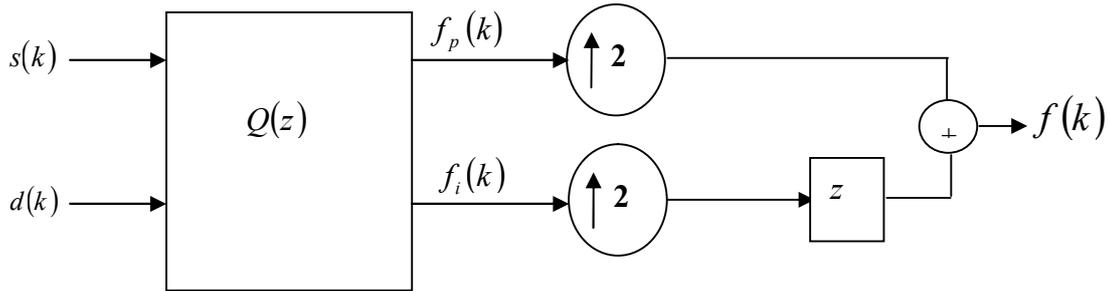
**Fig. 11** Représentation polyphasée du banc de filtres d'analyse

Notons que  $Q(z)$  est la matrice polyphase permettant la reconstruction du signal décomposé par  $P(z)$ . Nous obtiendrons pour le banc de filtres de synthèse le schéma de la figure 12.

La condition de reconstitution parfaite avec un retard de  $r$  échantillons s'écrit :

$$Q(z)P(z) = z^{-r}I \quad (2.47)$$

où  $I$  est la matrice identité  $2 \times 2$ .



**Fig. 12** Représentation polyphasée du banc de filtres de synthèse

### 2.3.5 Algorithme de factorisation de la matrice polyphase

Avant de donner les principales étapes qui permettent la factorisation de la matrice polyphase, nous allons définir une propriété très importante qui relie les deux filtres d'analyse et de synthèse. Sachant que, deux filtres sont complémentaires, si le déterminant de la matrice polyphase  $P(z)$  qui leur est associée est égal à 1.  $P(z)$  peut alors s'écrire comme suit :

$$P(z) = \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} \times \prod_{i=1}^m \begin{bmatrix} 1 & U^i(z) \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -P^i(z) & 1 \end{bmatrix} \quad (2.48)$$

où  $P^i(z)$  et  $U^i(z)$  sont les polynômes de LAURENT de la variable  $z$ ,  $K$  est une constante non nulle.

La matrice  $Q(z)$  s'en déduit simplement. Elle est de la forme :

$$Q(z) = \left\{ \prod_{i=m}^l \begin{bmatrix} 1 & 0 \\ P^i(z) & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -U^i(z) \\ 0 & 1 \end{bmatrix} \right\} \times \begin{bmatrix} \frac{1}{K} & 0 \\ 0 & K \end{bmatrix} \quad (2.49)$$

Les principales étapes de factorisation d'une matrice polyphase sont :

1. La première étape consiste à écrire la matrice polyphase en fonction des coefficients de  $\tilde{h}$  et  $\tilde{g}$  :

$$P(z) = \begin{bmatrix} \tilde{H}_p(z) & \tilde{H}_i(z) \\ \tilde{G}_p(z) & \tilde{G}_i(z) \end{bmatrix} \quad (2.50)$$

2. L'étape suivante consiste à factoriser  $P(z)$  en calculant le polynôme  $P^1(z)$  en écrivant :

$$P(z) = \begin{bmatrix} \tilde{R}_p(z) & \tilde{H}_i(z) \\ \tilde{R}'_p(z) & \tilde{G}_i(z) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -P^1(z) & 1 \end{bmatrix} = \begin{bmatrix} \tilde{R}_p(z) - \tilde{H}_i(z)P^1(z) & \tilde{H}_i(z) \\ \tilde{R}'_p(z) - \tilde{G}_i(z)P^1(z) & \tilde{G}_i(z) \end{bmatrix} \quad (2.51)$$

Par identification avec (2.50), nous obtiendrons :

$$\begin{cases} \tilde{H}_p(z) = \tilde{R}_p(z) - \tilde{H}_i(z)P^1(z) \\ \tilde{G}_p(z) = \tilde{R}'_p(z) - \tilde{G}_i(z)P^1(z) \end{cases} \quad (2.52)$$

$P^1(z)$  s'obtient par la division polynomiale de  $\tilde{H}_p(z)$  par  $-\tilde{H}_i(z)$  ou  $\tilde{G}_p(z)$  par  $-\tilde{G}_i(z)$ .  $\tilde{R}_p(z)$  et  $\tilde{R}'_p(z)$  sont respectivement les restes de ces deux divisions.

3. Après avoir calculé  $P^1(z)$ ,  $U^1(z)$  est calculé comme suit :

$$\begin{bmatrix} \tilde{R}_p(z) & \tilde{H}_i(z) \\ \tilde{R}'_p(z) & \tilde{G}_i(z) \end{bmatrix} = \begin{bmatrix} \tilde{R}_p(z) & \tilde{R}_i(z) \\ \tilde{R}'_p(z) & \tilde{R}'_i(z) \end{bmatrix} \cdot \begin{bmatrix} 1 & U^1(z) \\ 0 & 1 \end{bmatrix} \quad (2.53)$$

$U^1(z)$  est le quotient de la division polynomiale de  $\tilde{H}_i(z)$  par  $\tilde{R}_p(z)$  ou de  $\tilde{G}_i(z)$  par  $\tilde{R}'_p(z)$ .  $\tilde{R}_i(z)$  et  $\tilde{R}'_i(z)$  sont respectivement les restes de ces deux divisions.

4.  $P^2(z)$  et  $U^2(z)$  s'obtiennent en appliquant les deux étapes précédentes aux matrices

des restes  $\begin{bmatrix} \tilde{R}_p(z) & \tilde{R}_i(z) \\ \tilde{R}'_p(z) & \tilde{R}'_i(z) \end{bmatrix}$ . Les  $P^i(z)$  et  $U^i(z)$  suivants sont obtenus en itérant les

étapes 2 et 3 et en utilisant à chaque passage la matrice des restes obtenues à l'itération précédente.

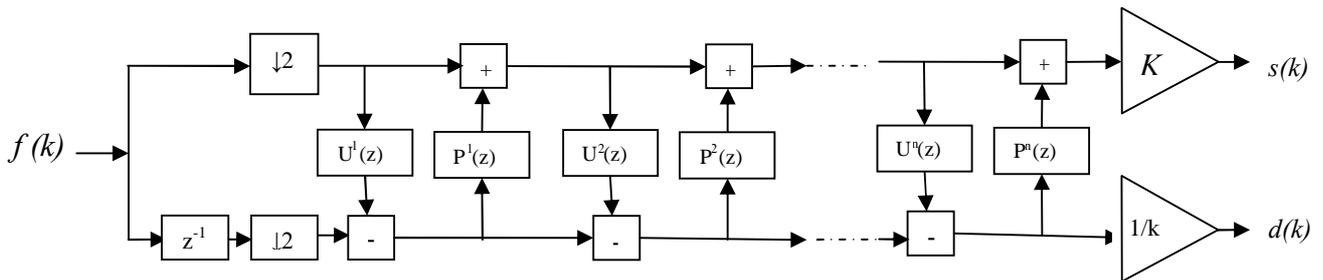


Fig. 13 Structure en échelle pour les bancs de filtres d'analyse

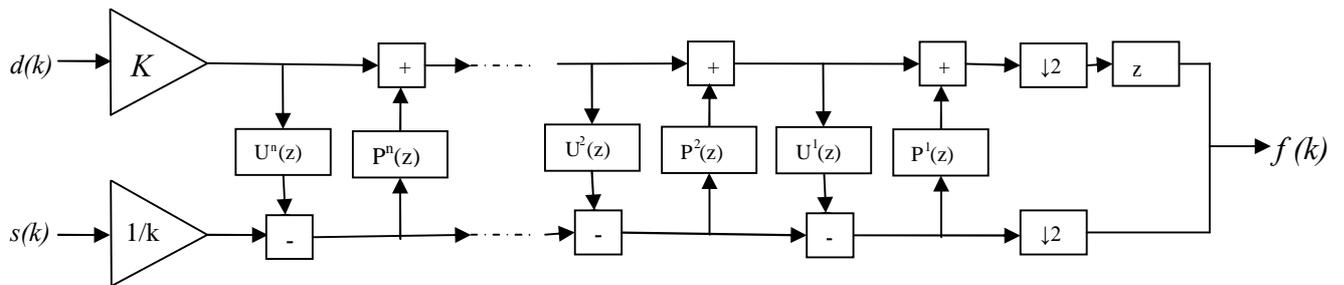


Fig. 14 Structure en échelle pour les bancs de filtres de synthèse

## 2.3.6 La transformée en ondelettes entières

### 2.3.6.a Principe

Dans la plupart des cas, la transformée en ondelettes produit des coefficients à valeurs réelles et bien que cela engendre en théorie une reconstruction parfaite de l'image originale, l'utilisation d'une précision arithmétique finie entraînera une reconstruction avec perte. Il est donc indispensable d'avoir une transformation produisant des entiers pour faire de la compression sans perte ou de la compression avec perte avec transmission progressive vers du sans perte. Traditionnellement ce type de transformation n'est pas facile à construire. Cependant, la construction devient très simple avec le lifting [Adams, 2002]. En effet, il suffit d'ajouter une opération d'arrondi après chaque pas de lifting. Ainsi le lifting et le lifting dual deviennent :

$$d^{(i)}(n) = d^{(i-1)}(n) - \left\lfloor \sum_k P^{(i)}(k) s^{(i-1)}(n-k) + \frac{1}{2} \right\rfloor \quad (2.54)$$

$$s^{(i)}(n) = s^{(i-1)}(n) - \left\lfloor \sum_k U^{(i)}(k) d^{(i)}(n-k) + \frac{1}{2} \right\rfloor \quad (2.55)$$

où  $\lfloor \cdot \rfloor$  désigne la partie entière inférieure.

Notons que bien que les équations (2.54) et (2.55) transforment des nombres entiers ( $d^{(i-1)}(n)$ ,  $s^{(i-1)}(n)$ ) en d'autres nombres entiers ( $d^{(i)}(n)$ ,  $s^{(i)}(n)$ ),

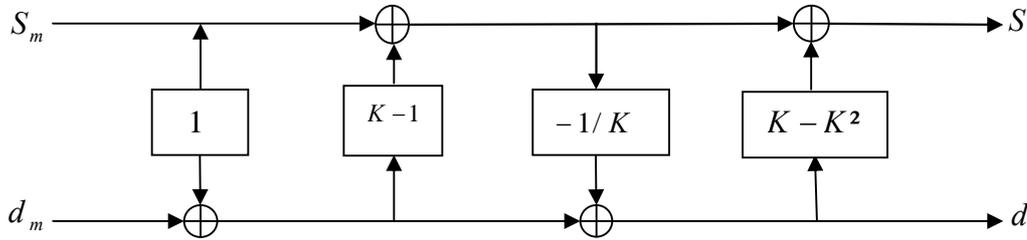
Il est très important de remarquer, que, cet arrondi n'introduit aucune perte d'information. Le signal (image) d'origine peut être reconstruit fidèlement en appliquant les opérations dans l'ordre inverse et en échangeant les signes « - » et « + ».

La présence de la matrice de normalisation  $\begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}$  ne garantit pas une décomposition

entière dans le cas où  $K \neq 1$ . Ce problème peut être résolu en remarquant que :

$$\begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} = \begin{bmatrix} 1 & K - K^2 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ -1/K & 1 \end{bmatrix} \times \begin{bmatrix} 1 & K-1 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (2.56)$$

Quatre étapes (deux maillons) de la transformation en ondelettes avec lifting décrivent rigoureusement cette opération. Le schéma devient réversible et se conforme au modèle des transformations en ondelettes entières.



**Fig. 15** Factorisation de l'opération de mise à l'échelle de  $s(n)$  et  $d(n)$ .

### 2.3.6.b Décomposition en ondelettes entière avec l'ondelette CDF 9/7

L'ondelette biorthogonale de Cohonen-Daubechies-Feauveau CDF9/7 [Daubechies et Sweldens, 1998] fait partie de la famille des ondelettes de type spline : la fonction d'échelle correspondant aux filtres de reconstruction est en effet une fonction B-Spline. La matrice polyphase associée au couple de filtres de l'ondelette CDF9/7 n'a pas un déterminant unitaire. Afin d'appliquer l'algorithme de factorisation, les coefficients des bancs de filtres doivent être multipliés par  $\sqrt{2}$ . Dans nos calculs, nous avons choisi les polynômes  $P^i(z)$  et  $U^i(z)$  sous la forme  $\omega(1+z)$  et  $\omega(1+z^{-1})$ , bien qu'il existe un grand nombre d'autres possibilités.

Après la factorisation, la matrice  $P(z)$  s'écrit :

$$P(z) = \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha(1+z) & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \quad (2.57)$$

Avec :

$$\begin{aligned} \alpha &= -1,586134342, & \beta &= -0,05298011854, & \gamma &= 0,8829110762, \\ \delta &= 0,4435068522, & \zeta &= 1,149604398 \end{aligned}$$

L'implantation de la décomposition en ondelettes entières avec l'ondelette biorthogonale CDF9/7 peut être faite de la manière suivante :

$$\begin{cases} S_l^{(0)} = f_{2l} \\ d_l^{(0)} = f_{2l+1} \end{cases}$$

$$\begin{cases} d_l^{(1)} = d_l^{(0)} - [\gamma(S_l^{(0)} + S_{l+1}^{(0)})] \\ S_l^{(1)} = S_l^{(0)} + \text{Int}(\delta(d_l^{(1)} + d_{l-1}^{(1)})) \end{cases}$$

$$\begin{cases} d_l^{(2)} = d_l^{(1)} - \text{Int}(\alpha(S_l^{(1)} + S_{l+1}^{(1)})) \\ S_l^{(2)} = S_l^{(1)} + \text{Int}(\beta(d_l^{(2)} + d_{l-1}^{(2)})) \end{cases}$$

$$\begin{cases} d_l^{(3)} = d_l^{(2)} + S_l^{(2)} \\ S_l^{(3)} = S_l^{(2)} + \text{Int}((\zeta - 1)d_l^{(3)}) \end{cases}$$

$$\begin{cases} d_l^{(4)} = d_l^{(3)} - \text{Int}((-1/\zeta)S_l^{(3)}) \\ S_l^{(4)} = S_l^{(3)} + \text{Int}((\zeta - \zeta^2)d_l^{(4)}) \end{cases}$$

$$\begin{cases} S_l = S_l^{(4)} \\ d_l = d_l^{(4)} \end{cases}$$

La reconstruction de signal décomposé peut être implantée de la manière suivante :

$$\begin{cases} S_l^{(4)} = S_l \\ d_l^{(4)} = d_l \end{cases}$$

$$\begin{cases} d_l^{(3)} = d_l^{(4)} + \text{Int}((-1/\zeta)S_l^{(3)}) \\ S_l^{(3)} = S_l^{(4)} + \text{Int}((\zeta - \zeta^2)d_l^{(4)}) \end{cases}$$

$$\begin{cases} d_l^{(2)} = d_l^{(3)} + S_l^{(2)} \\ S_l^{(2)} = S_l^{(3)} + \text{Int}((\zeta - 1)d_l^{(3)}) \end{cases}$$

$$\begin{cases} d_l^{(1)} = d_l^{(2)} + \text{Int}(\alpha(S_l^{(1)} + S_{l-1}^{(1)})) \\ S_l^{(1)} = S_l^{(2)} - \text{Int}(\beta(d_l^{(2)} + d_{l+1}^{(2)})) \end{cases}$$

$$\begin{cases} d_l^{(0)} = d_l^{(1)} + [\gamma(S_l^{(0)} + S_{l-1}^{(0)})] \\ S_l^{(0)} = S_l^{(1)} - \text{Int}(\delta(d_l^{(1)} + d_{l+1}^{(1)})) \end{cases}$$

$$\begin{cases} S_l^{(0)} = f_{2l} \\ d_l^{(0)} = f_{2l+1} \end{cases}$$

### 2.3.6.c Décomposition en ondelettes entières avec l'ondelette Gall 5/3

La matrice polyphase associée au couple de filtres de l'ondelette Gall 5/3 a un déterminant unitaire, alors, nous pouvons directement appliquer l'algorithme de factorisation. Après la factorisation, la matrice  $P(z)$  s'écrit :

$$P(z) = \begin{bmatrix} 1 & \beta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha(1+z) & 1 \end{bmatrix} \quad (2.58)$$

Avec :  $\alpha = -0,5$ ,  $\beta = 0,25$

L'implantation de la décomposition en ondelettes entières avec l'ondelette Gall 5/3 peut être faite de la manière suivante :

$$\begin{cases} S_l^{(0)} = f_{2l} \\ d_l^{(0)} = f_{2l+1} \end{cases}$$

$$\begin{cases} d_l^{(1)} = d_l^{(0)} - \text{Int}(\alpha(S_l^{(0)} + S_{l+1}^{(0)})) \\ S_l^{(1)} = S_l^{(0)} + \text{Int}(\beta(d_l^{(1)} + d_{l-1}^{(1)})) \end{cases}$$

$$\begin{cases} S_l = S_l^{(1)} \\ d_l = d_l^{(1)} \end{cases}$$

La reconstruction du signal décomposé peut être implantée de la manière suivante :

$$\begin{cases} S_l^{(1)} = S_l \\ d_l^{(1)} = d_l \end{cases}$$

$$\begin{cases} d_l^{(0)} = d_l^{(1)} + \text{Int}(\alpha(S_l^{(1)} + S_{l-1}^{(1)})) \\ S_l^{(0)} = S_l^{(1)} - \text{Int}(\beta(d_l^{(1)} + d_{l+1}^{(1)})) \end{cases}$$

$$\begin{cases} S_l^{(0)} = f_{2l} \\ d_l^{(0)} = f_{2l+1} \end{cases}$$

## **2.4 Discussion**

Dans ce chapitre, nous nous sommes intéressés à l'analyse des signaux monodimensionnels et bidimensionnels par transformations. Les transformations utilisées ont pour but de modifier la description d'une image en la décomposant en « éléments d'information » en général plus simples à interpréter. Ces éléments représentant l'image en sortie de la transformation (ou transformée) sont directement reliés à des coefficients sur lesquels nous porterons notre attention. Notre intérêt principal s'est porté, plus particulièrement, sur la transformée en ondelettes entières ayant plusieurs caractéristiques ; principalement, offrant une meilleurs localisation temporelle et fréquentielle à la fois.

# *Chapitre III*

## *Compression d'images par ondelettes entières et codeurs à arbres de zéros*

### **3.1 Préambule**

L'utilisation massive des modalités numériques en imagerie satellitaires engendre aujourd'hui des volumes de données de plus en plus importants. Le problème devient encore plus critique avec la généralisation des séquences hyperspectrales. Alors, il est nécessaire d'utiliser des images compressées afin de limiter le volume de données à stocker et à transmettre. Comme défini au chapitre 1, la compression d'une image numérique permet de réduire le nombre de bits qu'elle occupe, tout en exploitant les deux principaux points qui interviennent pour atteindre cet objectif ; à savoir la réduction de la redondance (n'introduisant pas de pertes), et la représentation approximative de l'information contenue dans l'image (introduisant une perte). De nombreux schémas de compression ont été proposés et standardisés, nous pouvons citer les normes JPEG pour les images, MPEG 1 et 2 pour la compression de vidéos. L'ensemble de ces standards reposent sur la transformée discrète en cosinus (DCT). Cependant, il y a quelques années une alternative intéressante a vu le jour : la transformée en ondelettes (la DWT ou bien la LWT), qui permet en particulier d'atteindre des taux de compression importants sans produire les artefacts (effets de bloc) observés sur les images compressées par la DCT. Ce chapitre a pour but de présenter un nouveau schéma de compression exploitant la corrélation des images afin d'améliorer la performance de la compression. Il s'appuie sur une transformée décorrélante : la transformée en ondelettes entières suivie du tri et du codage des coefficients résultants par les codeurs à arbres de zéros, à savoir le EZW (Embedded Zerotree Wavelet) [Shapiro, 1993] et le SPIHT (Set Partitioning In Hierarchical Trees) [Said et Pearlman, 1996], s'adaptant le mieux à la structure d'une

image décomposée par ce type d'ondelettes. Ces codeurs permettent d'obtenir d'excellents taux de compression et de faire aussi la transmission progressive d'images.

### 3.2. Méthode de compression décompression

#### 3.2.1. Principe

La figure 16 décrit la structure de l'algorithme que nous avons conçu, élaboré et appliqué à diverses images. Dans une première étape, l'image à compresser est décomposée par la transformation en ondelettes entières utilisant le schéma de lifting (LWT), implémentant deux types d'ondelettes : l'ondelette biorthogonale de Cohonen-Daubechies-Feauveau (CDF 9/7) et la Gall 5/3 prédéfinies au chapitre deux. L'image résultante est ensuite codée directement avec un codeur à arbres de zéros, EZW ou SPIHT, dans le but de générer une séquence binaire contenant l'information utile à sauvegarder ou à transmettre, c'est-à-dire, la séquence compressée. Afin de mettre en évidence, les performances de cet algorithme, une autre structure basé sur une transformation en ondelettes réelles (DWT), utilisant un quantificateur à zones mortes; associé, par la suite, aux codeurs à arbres de zéros sera adoptée pour une éventuelle comparaison. La décompression est alors réalisée en procédant par transformations inverses.

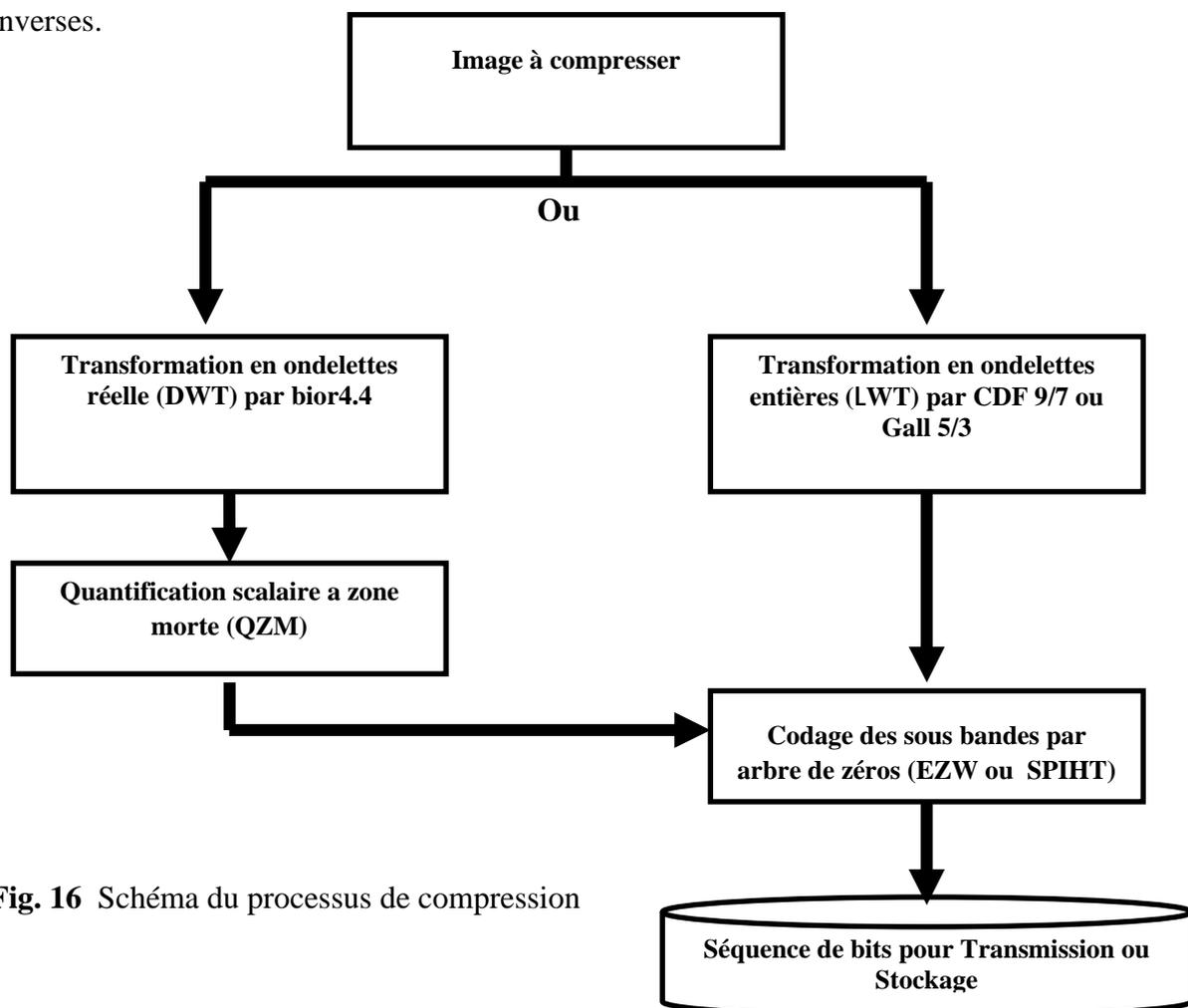
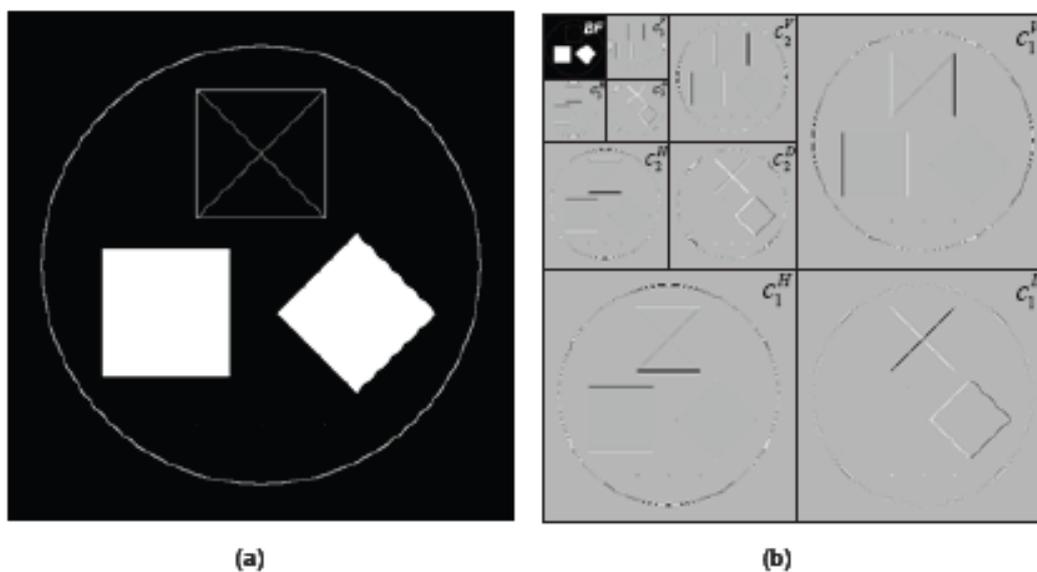


Fig. 16 Schéma du processus de compression

### 3.2.2 Transformation en ondelettes entières (LWT)

Il a été démontré dans le chapitre deux que n'importe quelle transformée en ondelettes pouvait être calculée à partir du schéma de lifting. Celui-ci permet de réduire le coût calculatoire par rapport à l'implantation par convolution, car, il utilise les redondances mathématiques entre le filtre passe-bas et le filtre passe-haut. Le second avantage de cette transformée est sa capacité à produire des coefficients d'ondelettes entiers. Cette fonctionnalité est fondamentalement utilisée dans notre schéma de compression.

Bien que la transformée en ondelettes soit utilisée pour décorréler les images, il est encore possible d'observer des structures dans les images transformées. La figure 17 (b), représente la transformée en ondelettes de l'image de la figure 17 (a). Sur cette figure, les coefficients d'ondelettes sont représentés en valeur absolue et la palette de couleurs a été ajustée de sorte que toute la gamme du gris soit utilisée dans chaque sous-bande. Ceci permet de faire apparaître les structures de l'image sur les différentes résolutions.



**Fig. 17** Illustration des dépendances inter échelles: (a) Image originale, (b) Image LWT

Des dépendances statistiques ont aussi été observées entre coefficients d'ondelettes dans un voisinage très local. Ces dépendances sont inter et intra échelles [Delaunay, 2008].

Par conséquent, après avoir appliqué la transformée en ondelettes entières sur des images tests, nous avons constaté que les images peuvent présenter des propriétés d'autosimilarités à différents niveaux d'échelles qui sont préservées par la transformation. Ces propriétés sont à la base des méthodes de compression utilisant les codeurs à arbres de zéro ; comme l'algorithme EZW et SPIHT que nous décrivons un peu plus loin.

La transformée en ondelettes entières retenue dans notre schéma de compression d'images est la transformée en ondelettes discrète biorthogonale CDF 9/7. C'est celle qui permet d'atteindre les meilleures performances en compression. C'est aussi la transformée en ondelettes utilisée dans la norme JPEG2000. Nous avons aussi implémenté l'ondelette de Gall 5/3. Les étapes de décompositions des ondelettes utilisées ont été détaillées dans le chapitre 2. Le nombre de niveaux de décomposition en ondelettes a été fixé en fonction de la dimension de l'image à compresser.

### 3.2.3 Etape de quantification

Dans un schéma de compression, l'étape de quantification est celle qui dégrade de manière irréversible l'image. Elle est cependant, d'une importance capitale dans la réduction du débit binaire. Cette étape a pour but de déterminer pour un débit cible, le dictionnaire qui minimisera la distorsion.

Il existe plusieurs types de stratégies et le quantificateur scalaire à zone morte est celui que nous avons appliqué aux coefficients issus de la transformée en ondelettes réelles. Ce quantificateur occupe une place centrale dans la compression par ondelettes car, il permet d'éliminer les coefficients non significatifs de faible magnitude au profit des coefficients significatifs qui sont alors quantifiés plus finement. Nous définissons ce type de quantificateur de la manière suivante :

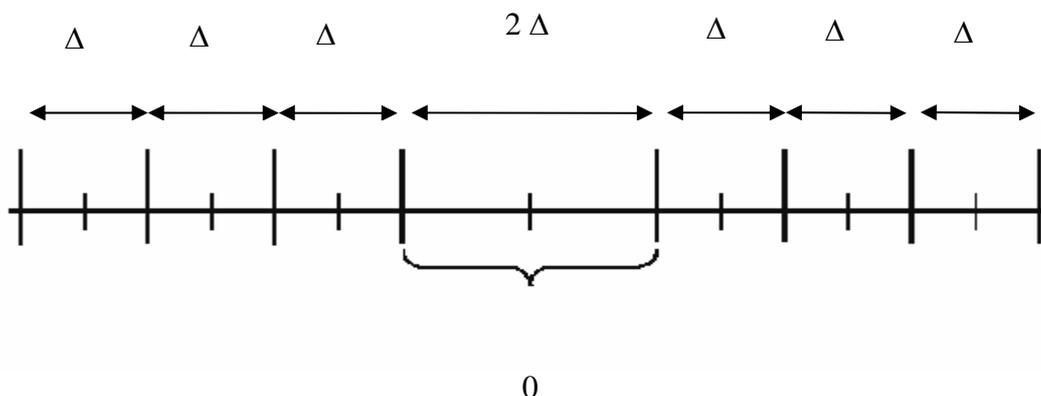
$$Q(x) = \left\{ \begin{array}{l} 0 \text{ si } |x| < \delta \\ \text{round} \left[ \frac{x - \text{signe de } (x) \times \left( \delta \div \frac{\Delta}{2} \right)}{\Delta} \right] \text{ si non} \end{array} \right\} \quad (3.1)$$

Avec  $\delta$  le seuil de la zone morte qui est défini par l'équation 3.2,  $\Delta$  est le pas de quantification et la fonction *round* est utilisée pour avoir une résultante entière.

$$\delta = \log_2 (\max (C_{i,j})) \quad (3.2)$$

Avec :  $\max (C_{i,j})$  est la valeur du coefficient d'ondelettes maximal de l'image transformée

Ce quantificateur est une légère modification du quantificateur uniforme. Pour lui, seul l'intervalle qui contient la valeur  $x = 0$  ( $a < 0 < b$ ) a une taille différente de  $\Delta$ , taille de tous les autres intervalles. La taille de l'intervalle contenant  $x = 0$  (la zone morte) est un multiple de  $\Delta$  (souvent  $2\Delta$ ). Cette modification permet d'améliorer la performance du système quand la distribution de la source est concentrée autour de zéro.

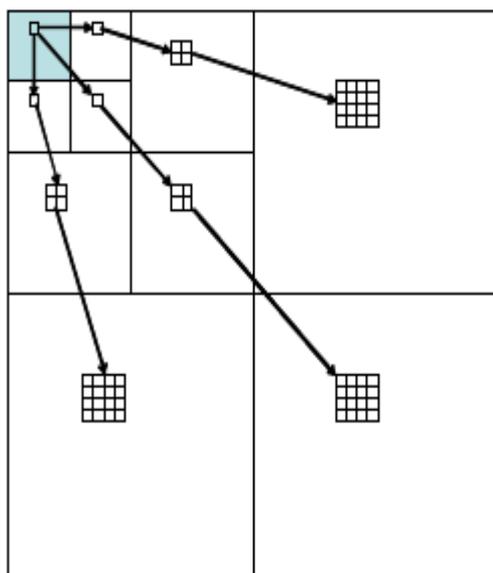


**Fig. 18** Définition des zones de quantification et mise en évidence de la zone morte

### 3.2.4 Etape de codage à arbre de zéro

Comme nous l'avons évoqué précédemment, une des raisons du succès de la transformée en ondelettes réside dans le fait qu'il existe d'importantes corrélations entre les différentes échelles concernant la position des coefficients de forte amplitude ou de manière équivalente de faible amplitude. Par conséquent, nous distinguons deux types d'approches pour les standards de compression actuels basés sur la transformée en ondelettes : l'approche inter bandes qui utilise les redondances inter échelles entre les sous bandes pour coder les coefficients d'ondelettes et celle intra bandes dans laquelle les sous bandes sont codées de façon indépendante. L'idée principale du codage inter-bandes, consiste à utiliser la position des coefficients significatifs à la plus basse résolution, pour prédire la position et la valeur des

coefficients significatifs aux échelles plus fines. Lorsqu'un coefficient est non significatif, les coefficients correspondant aux plus fines échelles ont une grande probabilité d'être non significatifs. Une structure en arbre est alors utilisée pour représenter les coefficients d'ondelettes. Un exemple d'une telle structure en arbre est donné sur la figure 19 : la position d'un coefficient d'ondelettes "père" engendre la localisation spatiale de quatre coefficients "enfant" à l'échelle supérieure. Les principales méthodes inter-bandes sont EZW et SPIHT.



**Fig. 19** Illustration de l'approche inter bandes : structure arborescente du EZW et du SPIHT

#### 3.2.4.1 Codeur EZW

La méthode de codage progressif connue sous le nom de Embedded Zerotree Wavelet coding (EZW), proposée par [Shapiro, 1993], est une méthode simple et très efficace de compression d'image par ondelettes. Elle a démontré sa puissance dans les deux formes de compression (avec et sans perte d'informations) depuis son élaboration en 1993.

En général, dans une représentation de l'image par coefficients d'ondelettes, l'image obtenue est organisée de façon à représenter les principaux traits de celle-ci dans les bandes de basses fréquences, puis, les détails dans les bandes de hautes fréquences. Le principe du codeur EZW s'appuie sur cette représentation, pour coder les coefficients d'une manière progressive. Ainsi,

nous commençons par les basses fréquences, ensuite nous codons les détails (hautes fréquences). Il est basé principalement sur deux points :

- Après la transformation en ondelettes d'une image, l'énergie dans les sous bandes diminue pendant que l'échelle diminue (la basse échelle signifie la haute résolution). Ainsi, les coefficients d'ondelettes seront plus petits en moyenne dans les sous bandes plus hautes que dans les sous bandes inférieures. Ceci prouve que le codage progressif est un choix très normal pour des images transformées par ondelettes, puisque les sous bandes plus hautes ajoutent seulement des détails fins.
- Les grands coefficients d'ondelettes sont plus importants que les plus petits.

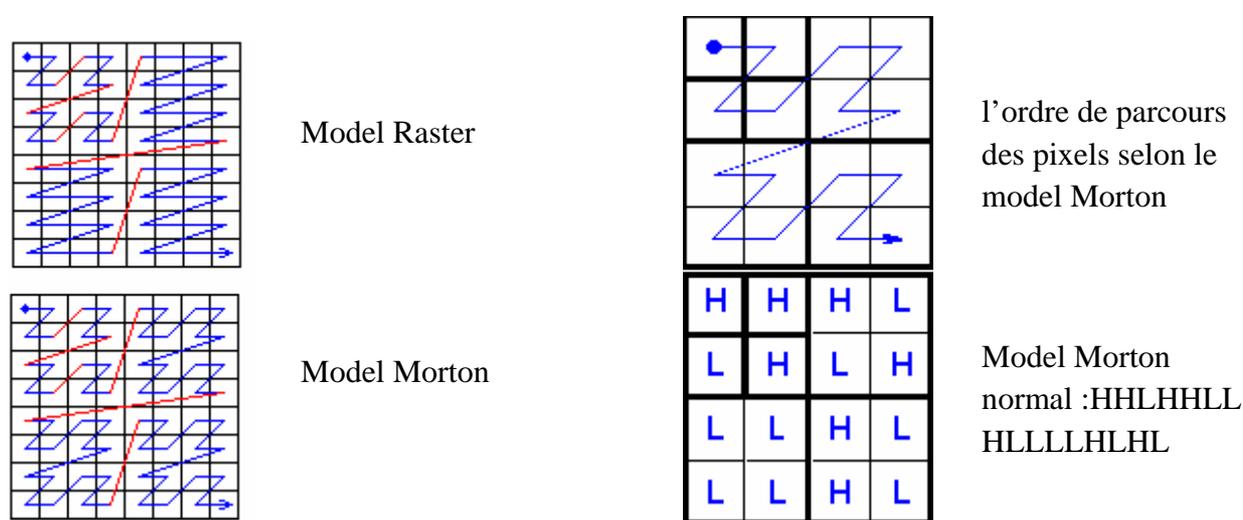
Ces deux points sont exploités en codant les coefficients d'ondelettes par ordre décroissant, dans plusieurs passages. Pour chaque passage, nous choisissons un seuil par rapport auquel tous les coefficients d'ondelettes sont comparés. Si, un coefficient d'ondelettes est supérieur au seuil, il est codé et sa valeur est remise à zéro dans l'image. Sinon, il est laissé pour le prochain passage. Quand tous les coefficients d'ondelettes ont été examinés, le seuil est abaissé et l'image est rebalayée pour ajouter plus de détails à l'image déjà codée. Ce processus est répété jusqu'à ce que tous les coefficients d'ondelettes soient complètement encodés ou qu'un autre critère soit satisfait (PSNR choisi soit atteint).

Dans la représentation de « l'image ondelettes », chaque coefficient peut être considéré en tant qu'ayant quatre descendants dans la prochaine plus haute sous bande (Fig. 19). Pour les quatre descendants, chacun à également quatre fils dans la prochaine plus haute sous bande ; ainsi, nous voyons un arbre à quatre descendants émerger ; chaque racine a quatre branches, ce processus se poursuit jusqu'aux fréquences les plus hautes. C'est cette dépendance existante entre les coefficients d'ondelettes à travers différentes échelles qu'utilise le codeur EZW pour coder efficacement les grandes parties de l'image qui sont en dessous du seuil actuel.

Nous pouvons donc, définir un zerotree comme étant un arbre quadruple dont tous les noeuds enfants sont égaux ou plus petits que les noeuds parents, tout en sachant que les racines (les noeuds parents) dans ce cas, doivent être plus petites que le seuil par rapport auquel les coefficients d'ondelettes sont comparés. L'arbre est codé avec un symbole unique et reconstruit par le décodeur comme un arbre quadruple rempli de zéros. Le codeur EZW exploite le fait qu'il y' ait une probabilité très élevée que tous les

coefficients dans un arbre quadruple soient plus petits qu'un certain seuil si, la racine de cet arbre est plus petite que ce seuil. Ceci entraîne alors un seul code zerotree pour tout l'arbre.

Ceci dit, en balayant toute la représentation des coefficients d'ondelettes, selon le modèle choisi (Fig. 20), des basses aux hautes fréquences, nous aurons automatiquement beaucoup de Zerotree, ce qui constituera un gain considérable au niveau de la compression.



**Fig. 20** Modèles de balayage des coefficients d'ondelettes

Ces modèles de balayage ont été choisis de manière à préserver l'ordre d'importance des coefficients traités, ainsi, pour les deux types, nous commençons par parcourir les coefficients de basses fréquences, tout en avançant graduellement vers les détails (hautes fréquences), la différence entre eux est la façon avec laquelle est parcourue une même sous bande.

#### Algorithme de base :

L'algorithme peut être divisé en deux étapes : une étape principale (étape de tri) qui constitue le fond de tâche du codeur. Une étape secondaire ou subalterne, qui est plus pour le raffinement de la reconstruction de l'image pendant le décodage.

Le premier point de cette l'algorithmme est de déterminer le seuil initial  $T_0$  qui sera défini comme suit :

$$T_0 = 2^n \quad (3.3)$$

Avec :  $n = \log_2 [\text{Max}(|C_{i,j}|)]$

Max (.) est la valeur du coefficient d'ondelettes  $C_{i,j}$  maximal de l'image transformée, de la ligne i et de la colonne j.

Ce seuil, comme le montre la définition, est un multiple de deux, en fait c'est la puissance de deux la plus proche (inférieure ou égale) de la valeur maximale de tous les coefficients de l'image. Cette puissance fera partie par la suite du paquet d'informations transmis pour initialiser le décodage.

Par la suite, et suivant un balayage précis, chacun des coefficients parcourus est comparé (en valeur absolue) au seuil  $T_0$ . Si le coefficient est supérieur au seuil, il est codé 'Significatif Positif ' ou ' Significatif Négatif ', sinon il est : soit ' Zéro isole ' ou bien ' Zerotree'. Nous nous ramenons ainsi à X symboles (selon les niveaux de gris, résolution de l'image...) à coder à un dictionnaire de quatre symboles :

- Significatif Positif (P) : indique que la valeur absolue du coefficient traité est supérieure au seuil et que son signe est positif.
- Significatif Négatif (N) : indique que la valeur absolue du coefficient traité est supérieure au seuil et que son signe est négatif.
- Zéro isolé (IZ) : indique que la valeur absolue du coefficient traité est inférieure au seuil et qu'il existe parmi ses descendants (selon l'arborescence présentée dans la Figure 19) ceux qui sont significatifs (c'est à dire supérieurs au seuil).
- Zerotree (ZTR) : indique que la valeur absolue du coefficient traité est insignifiante par rapport au seuil considéré ainsi que tous les coefficients qui lui succèdent dans l'arbre de descendance.

Chaque coefficient significatif est ensuite mis à zéro dans la transformée en ondelettes afin que sa position ne soit plus encodée, et sa valeur absolue est placée dans une liste pour être

codée dans la seconde étape. Après le balayage de tous les coefficients, le seuil est divisé par deux, et l'opération est réitérée selon le nouveau seuil, cette méthode est appelée « quantification par approximations successives » et peut être renouvelée tant que le seuil est supérieur ou égal à 1, sachant que :

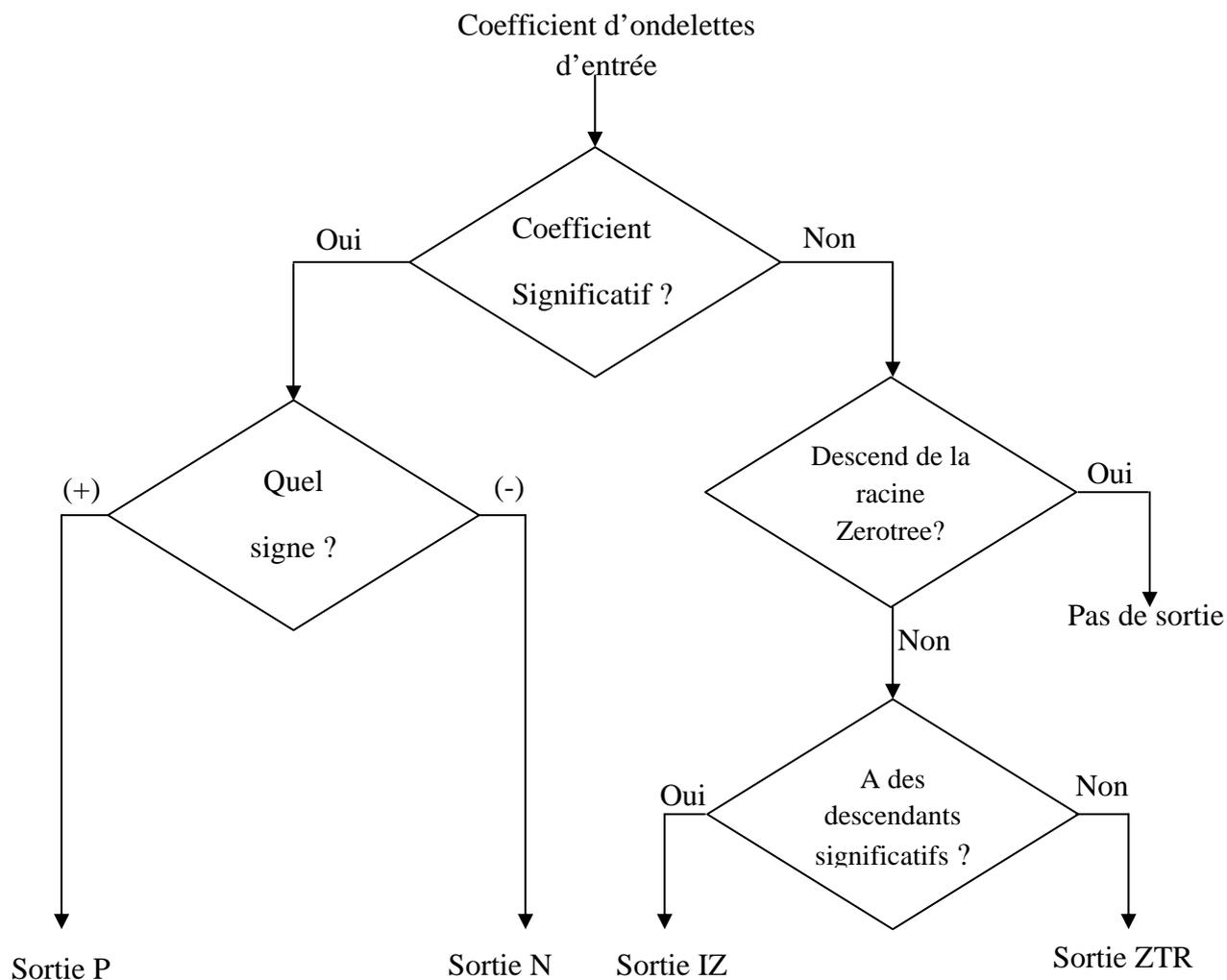
$$T_{n+1} = \frac{T_n}{2} \quad (3.4)$$

Le traitement secondaire (étape de raffinement) sert à faire du raffinement sur les valeurs des coefficients déjà classés, ainsi, après chaque étape de tri, chaque coefficient codé ' Positif ou Négatif ' subit une autre comparaison ; si  $T_n$  est le seuil courant, alors les coefficients évalués dans l'étape précédente ont leurs valeurs absolues dans l'intervalle  $[T_n, 2 T_n[$ , cet intervalle est alors divisé en deux :  $[T_n, \frac{3 T_n}{2}[$  et  $[\frac{3 T_n}{2}, 2 T_n[$ .

Les coefficients dont la valeur absolue se trouve dans le premier intervalle sont codés par le symbole '0', alors que ceux se trouvant dans le second intervalle nous leur associons le symbole '1'. Lorsque la seconde étape est finie, l'algorithme reprend le processus et génère la liste suivante dont le nouveau seuil étant  $T_{n+1}$ , déduit de l'équation 3.4.

L'étape de raffinement permet au décodeur dans le cas d'une compression avec pertes d'informations, d'avoir plus de précision sur la valeur du coefficient codé, certe, la reconstruction ne sera pas parfaite, mais, elle sera de loin meilleure que si on ne codait qu'avec l'étape de tri.

Le principe général de l'algorithme est illustré par l'organigramme de la figure 21.



**Fig. 21** Test de signification des coefficients d'ondelettes

Pour que le codage soit parfait, un protocole entre codeur et décodeur est établi, ainsi, le décodeur doit connaître le dictionnaire de codage (dans ce cas les 4 symboles utilisés au codage), et le type de balayage des coefficients effectués (Raster ou Morton). Pour sa part, le codeur doit transmettre au moins le seuil de départ, de préférence la puissance associée à ce seuil, et le nombre de niveaux de décomposition de l'ondelette.

### 3.2.4.2 Codeur SPIHT

L'algorithme SPIHT a été proposé par Saïd et Pearlman en 1996 pour la compression avec et sans perte [Saïd et Pearlman, 1996]. Il repose sur la même idée que celle de Shapiro (EZW) pour caractériser les dépendances entre les coefficients d'ondelettes. Cependant, il est à la fois plus complexe et plus efficace pour coder les listes de signification.

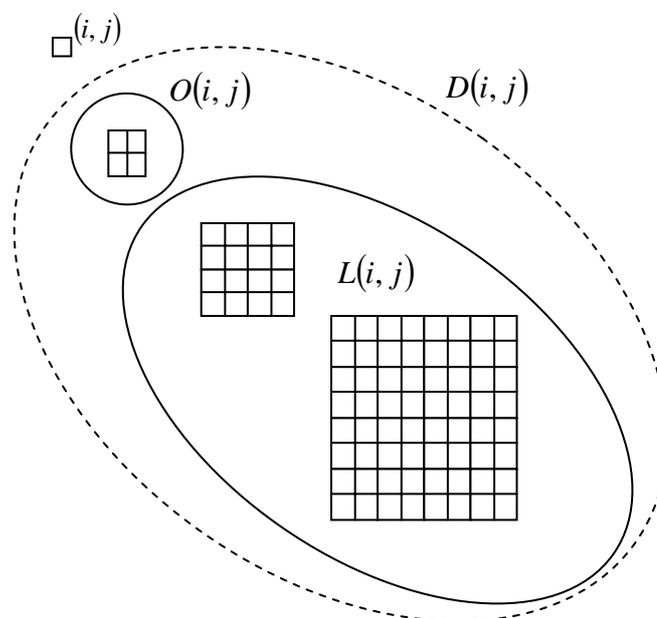
Les principes de base de l'algorithme SPIHT est un rangement partiel par amplitude des coefficients d'ondelettes, ensuite, un partitionnement dans des arbres hiérarchiques est effectué (à chaque seuil appliqué, les arbres sont triés sur la base de leur signification en deux catégories d'arbres) et l'amplitude de chaque coefficient significatif est progressivement raffinée [Christophe, 2006 ; Saïd et Pearlman, 1996]. Sa première implantation est basée sur des arbres à orientation spatio-temporelle équilibrée. Par conséquent, le même nombre de décompositions récursives d'ondelettes est requis pour les deux directions spatiales (x et y). Si cela n'est pas respecté, plusieurs noeuds de l'arbre ne sont pas rattachés ou sont liés avec la même localisation spatiale et par conséquent, les dépendances entre les noeuds de l'arbre sont détruites et ainsi la performance de compression.

La première différence à noter par rapport à EZW est que toute sortie est binaire. La seconde différence est que l'ordre de traitement des coefficients est dépendant des données. Alors que dans EZW, les coefficients sont traités en zigzag dans chaque sous bande. Le système de listes de SPIHT laisse l'ordre entièrement dépendant des données, et les coefficients sont traités selon leur ordre dans les listes. Un autre point de différence à noter, est la façon dont les coefficients des arbres sont construits, triés et découpés. Ainsi, la structure même des arbres de zéros est différente. Dans EZW, un arbre de zéros est défini par un coefficient racine, et ses descendants ont tous la valeur zéro à l'intérieur d'un plan de bits. Tandis que, SPIHT utilise deux types d'arbres de zéros. Le premier (type A) consiste en une simple racine ayant tous ses descendants à 0 pour un plan de bits donné. Cela diffère un peu des arbres de zéros de EZW du fait que la racine, elle-même, n'a pas besoin d'être non significative. En fait, bien que l'arbre de zéros soit spécifié par les coordonnées de la racine, la racine n'est pas incluse dans l'arbre. Le second type d'arbre (type B) est similaire mais exclut les huit enfants de la racine. Les arbres de type B contiennent uniquement les petits-enfants, arrière-petits-enfants ... de la racine. De plus, dans SPIHT, les arbres sont définis de telle façon, que chaque, noeud ne possède aucun descendant (les feuilles) ou bien 4 descendants qui forment un groupe adjacent de  $2 \times 2$ . Les coefficients de la sous bande basses

fréquences correspondant aux racines de l'arbre sont également groupés en coefficients  $2 \times 2$  adjacents. Cependant, la relation parent enfant pour un coefficient du noeud racine est altérée par rapport à EZW. Dans chaque groupe de  $2 \times 2$  de la sous bande basses fréquences, un des coefficients n'a pas de descendants. Ainsi, tous les coefficients  $C_{i,j}$  qui possèdent deux coordonnées impaires  $(i, j)$  n'ont pas de descendant.

Les ensembles de coordonnées suivants, (Fig. 22), sont utilisés dans l'algorithme de codage complet:

- $O(i, j)$  : Ensemble des coordonnées de tous les enfants du noeud  $(i, j)$ . Par exemple,  $O(i, j) = \{ (2i, 2j), (2i, 2j+1), (2i+1, 2j), (2i+1, 2j+1) \}$
- $D(i, j)$  : Ensemble des coordonnées de tous les descendants du noeud  $(i, j)$  (arbre de zéro de type A).
- $L(i, j) = D(i, j) - O(i, j)$  (arbre de zéro de type B).
- $H$  = ensemble des coordonnées de toutes les racines d'arbre à orientation spatiale = les coordonnées des parents de la sous bande basses fréquences



**Fig. 22** Terminologie SPIHT pour les descendants

Les règles à respecter pour le partitionnement sont :

- La partition initiale est formée des ensembles  $\{(i, j)\}$  et  $D(i, j)$ , pour tout  $(i, j) \in H$  (à la sous bande basses fréquences qui ont un descendant).
- Si  $D(i, j)$  est significatif alors il est découpé en  $L(i, j)$  plus 4 ensembles d'un seul élément avec  $(l, m) \in O(i, j)$ .
- Si  $L(i, j)$  est significatif alors il est partitionné en 4 sous-ensembles  $D(l, m)$  avec  $(l, m) \in O(i, j)$ .

### Principe de l'algorithme de codage

Pour réaliser pratiquement son codage imbriqué, SPIHT stocke l'information de signification dans 3 listes ordonnées :

- la Liste des Coefficients Significatifs (LCS),
- la Liste des Coefficients Non significatifs (LCN),
- la Liste des Ensembles Non significatifs (LEN).

Dans chaque liste, l'entrée est représentée par les coordonnées  $(i, j)$  qui représentent dans LCS et LCN des coefficients individuels et dans LEN soit l'ensemble  $D(i, j)$  soit  $L(i, j)$ . Pendant l'étape de signification (étape de tri), les coefficients dans LCN, qui étaient non significatifs dans la passe précédente sont testés. Ceux qui deviennent significatifs sont mis dans LCS. Similairement, les ensembles de LEN sont évalués dans leur ordre d'entrée, et quand un ensemble est trouvé significatif il est supprimé de cette liste puis partitionné. Les nouveaux ensembles avec plus d'un élément sont ajoutés à la fin de LEN (avec le type A ou B), alors que les simples coefficients sont ajoutés à la fin de LCS ou LCN suivant leur signification. La liste LCS contient les coordonnées des coefficients qui seront visités dans la prochaine étape de raffinement.

L'algorithme de codage SPIHT est défini comme suit, tout en représentant son principe général dans la figure 23 :

### A. Initialisation

- Sortir  $n = \lceil \log_2 [\max |C_{i,j}|] \rceil$  où  $\max |C_{i,j}|$  désigne la valeur absolue du coefficient d'ondelettes le plus élevé, le seuil  $T_n = 2^n$ ,
- LCS est vide,
- LCN =  $\{(i, j)\}$  cet ensemble représente les nœuds de la basse fréquence.
- LEN contient les mêmes coefficients que LCN, excepté, ceux qui n'ont pas de descendants.

Pour déterminer la signification d'un sous-ensemble E pour un seuil donné  $T_n$ , nous utiliserons l'opérateur de signification  $S_{T_n}$  tel que :

$$S_{T_n}(E) = \begin{cases} 1 & \text{si } \exists C \in E : |C| \geq T_n \\ 0 & \text{si } \forall C \in E : |C| < T_n \end{cases} \quad (3.5)$$

### B. Etape de signification (étape de tri)

1. Pour chaque  $(i, j) \in \text{LCN}$  faire :

1.1 Sortir  $S_{T_n}(i, j)$

1.2 Si  $S_{T_n}(i, j) = 1$  alors mettre  $(i, j)$  dans LCS et coder le signe de  $C(i, j)$

2. Pour chaque  $(i, j) \in \text{LEN}$  faire :

2.1 Si l'entrée est de type A

a- Sortie  $S_{T_n}(D(i, j))$

b- si  $S_{T_n}(D(i, j)) = 1$  alors

- Pour chaque  $(l, m) \in O(i, j)$  faire :

- Sortir  $S_{T_n}(l, m)$

- Si  $S_{T_n}(l, m) = 1$  alors mettre  $(l, m)$  dans LCS et coder le signe de  $C(l, m)$

- Si  $S_{T_n}(l, m) = 0$  alors mettre  $(l, m)$  à la fin de LCN

- Si  $L(i, j) \neq \emptyset$  alors mettre  $(l, m)$  à la fin de LEN comme une entrée de type

B et allez au point 2.2.

2.2 Si l'entrée est de type B, donc

a- Sortir  $S_{T_n}(L(i, j))$

b- Si  $S_{T_n}(L(i, j)) = 1$  alors

- mettre  $(l, m)$  à la fin de LEN comme une entrée de type A

- supprimer  $(i, j)$  de LEN

### C. Etape de raffinement

Pour chaque coefficient  $(i, j) \in \text{LCS}$ , à l'exception, de ceux inclus dans la même étape de signification (c.à.d. pour le même  $n$ ). Sortir le nième bit le plus significatif de  $|C(i, j)|$

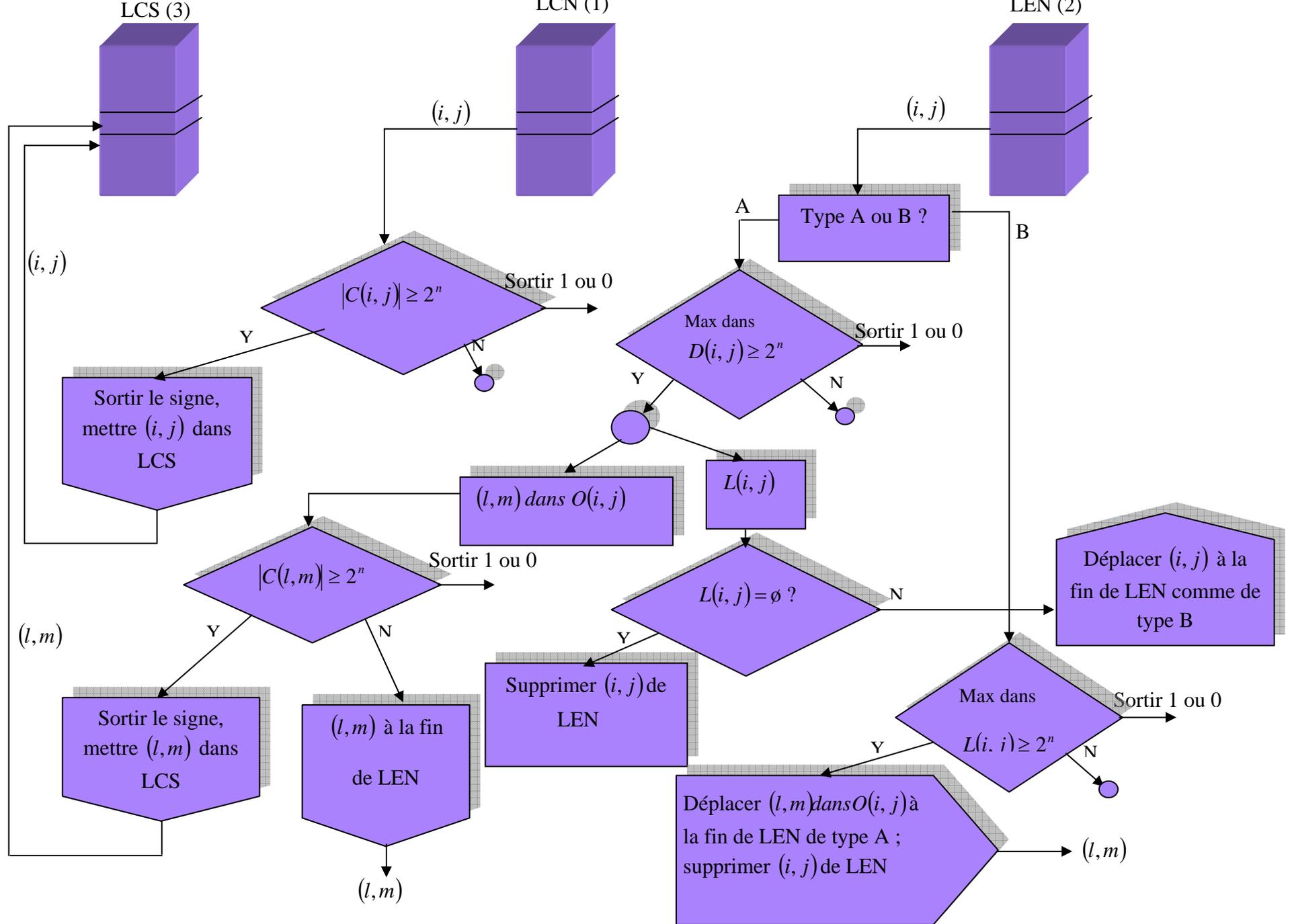
Ensuite, modifiez le seuil  $T_n$  en décrémentant  $n$  de 1 et renouveler l'étape de signification.

Le codeur doit transmettre progressivement, au décodeur, au moins le seuil de départ,  $S_{T_n}$ , le code des signes des coefficients significatifs ainsi que le bit le plus significatif de ces derniers. Pour obtenir, l'algorithme de décodage, il suffit simplement de remplacer le mot Sortir par Entrée dans l'algorithme précédent.

Un exemple des algorithmes EZW et SPIHT est détaillé dans l'annexe B.

### 3.3. Discussion

Nous avons expliqué dans ce chapitre notre schéma de compression/décompression par les codeurs à arbre de zéros (EZW et SPIHT) adaptés au domaine des ondelettes entières. L'application de cette méthode aux différents types d'images tests et MSG ainsi que les résultats obtenus feront l'objet du chapitre suivant.



**Fig. 23** Principe général de l'algorithme SPIHT

# *Chapitre IV*

## *Résultats et discussion*

### **4.1 Préambule**

Il est important de souligner que la comparaison des performances fournies par toutes les méthodes de compression nécessite une base de données dite de référence pour les tester. Ainsi, un ensemble d'images références très utilisées en compression d'images sera défini pour évaluer notre algorithme. Comparer les algorithmes de compression en imagerie est un problème très complexe. Nous avons évoqué, précédemment, les difficultés pour évaluer les pertes, la qualité de l'image reconstruite, la précision du diagnostic. Dans un même temps, nous avons décrit brièvement un certain nombre d'outils statistiques et subjectifs permettant d'évaluer les performances des méthodes. Le rapport signal sur bruit demeure le critère essentiel d'évaluation. Nous nous intéressons dans ce chapitre à présenter les résultats obtenus par application de notre schéma de compression utilisant les ondelettes entières CDF9/7 et Gall5/3 associées aux codeurs EZW (Embedded Zerotree Wavelet) et le SPIHT (Set Partitioning In Hierarchical Trees). Pour mieux apprécier la pertinence de nos résultats, deux grandeurs ont été considérées. Ce sont le taux de compression en bit par pixel (bpp) et le rapport signal sur bruit (PSNR).

### **4.2. Présentation des données**

Afin d'évaluer les performances de la méthode décrite dans le chapitre précédent, deux sortes d'images ont été utilisées :

- Une série d'images tests très utilisées en compression d'images. Ces images sont Lena et Goldhill.
- Une série d'images Météosat seconde génération (MSG), collectées toutes les quinze minutes durant l'année 2007.

### 4.2.1. Image Tests

Pour valider la méthode que nous avons mise en œuvre et montrer son efficacité, nous l'avons appliqué aux images tests Lena et Goldhill (Fig.24). Celles-ci sont formées de 512x512 pixels codés sur 256 niveaux de gris. Elles représentent respectivement un portrait et un paysage que l'on peut rencontrer de manière classique dans une base de données multimédia et qui sont composées de diverses textures, dont, celles très fines.



a) Image Lena



b) Image Goldhill

**Fig. 24** Images Tests

### 4.2.2 Images Météosat Seconde Génération (MSG)

Les images utilisées ont été prises par le satellite Météosat de seconde génération (MSG) qui est un satellite géostationnaire doté de douze canaux différents; du spectre visible et infrarouge : Deux dans le visible, deux dans le proche infrarouge, deux dans le vapeur d'eau, cinq dans l'infrarouge et le dernier le visible haute résolution (HRVIS). A titre indicatif, les principales caractéristiques de ces canaux sont données dans le tableau 2. La figure 25 donne un exemple d'images MSG que nous avons traité. Ces images sont de taille 512\*512\*8 bits. Elles ont été prises dans le canal visible 0.6 (Fig. 25.a) et le canal infrarouge 10.8 (Fig. 25.b) au dessus de l'Afrique du Nord le 29 octobre 2007 à 13H GMT. Ces deux canaux sont utilisés pour la détection et le suivi des nuages, l'analyse des séries d'images d'une même

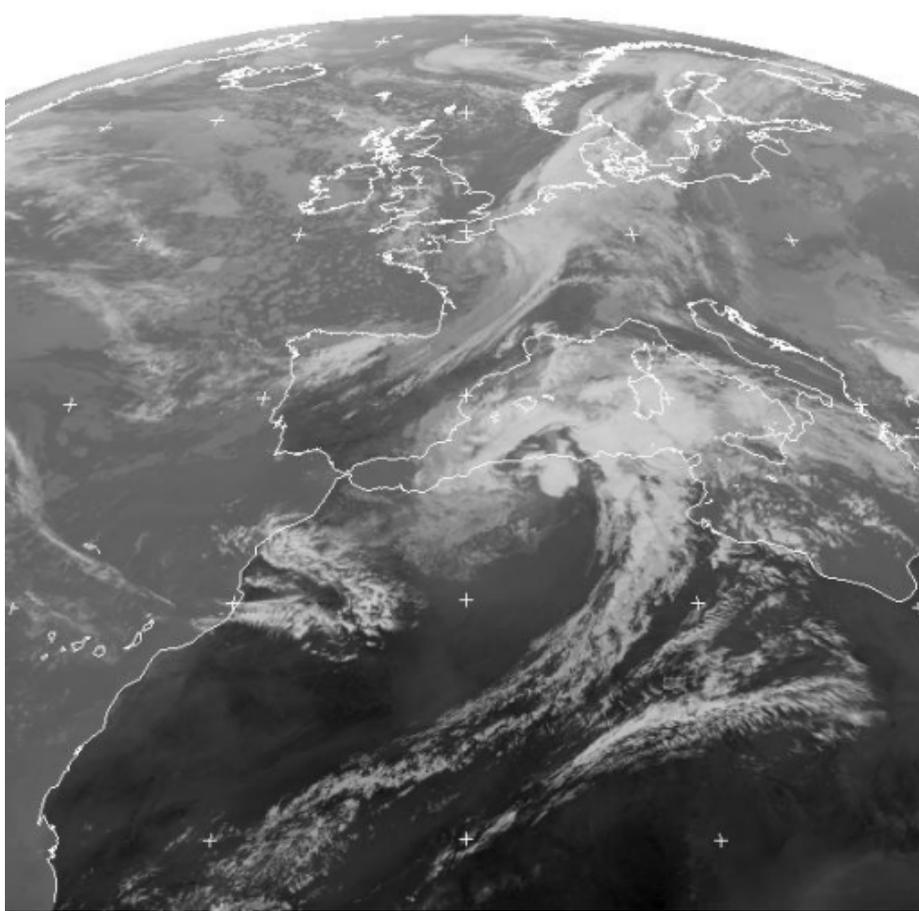
zone (par exemple de la végétation), la détection des aérosols (surveillance des poussières et des agents polluants) et la navigation dans les images (repères géographiques).

Canal	Bande spectrale ( $\mu\text{m}$ )
Visible : HRV	0,56 à 0,71
canal 0.6	0,56 à 0,71
canal 0.8	0,74 à 0,88
Proche Infrarouge : canal 1.6	1,5 à 1,78
Infrarouge moyen : canal 3.9	3,48 à 4,36
Vapeur d'eau : canal 6.2	5,35 à 7,15
canal 7.3	6,85 à 7,85
Infrarouge : canal 8.7	8,3 à 9,1
canal 9.7	9,38 à 9,94
canal 10.8	9,8 à 11,8
canal 12.0	11, à 13
canal 13.4	12,4 à 14,4

**Tableau 2** Bandes spectrales des douze canaux du satellite MSG



**Fig. 25.a** Image Visible 0.6 prise par le satellite MSG le 29 Octobre 2007 à 13 H GMT.



**Fig. 25.b** Image Infrarouge 10.8 prise par le satellite MSG le 29 Octobre 2007 à 13 H GMT.

### 4.3. Résultats expérimentaux et interprétations

L'algorithme de compression que nous avons développé dans le chapitre précédent, a été mis en œuvre en utilisant le langage de programmation MATLAB 2007b, sous l'environnement WindowsXP installé sur un microordinateur Dual Core de fréquence 1.8 GHz, possédant une mémoire centrale (RAM) de 1 Go.

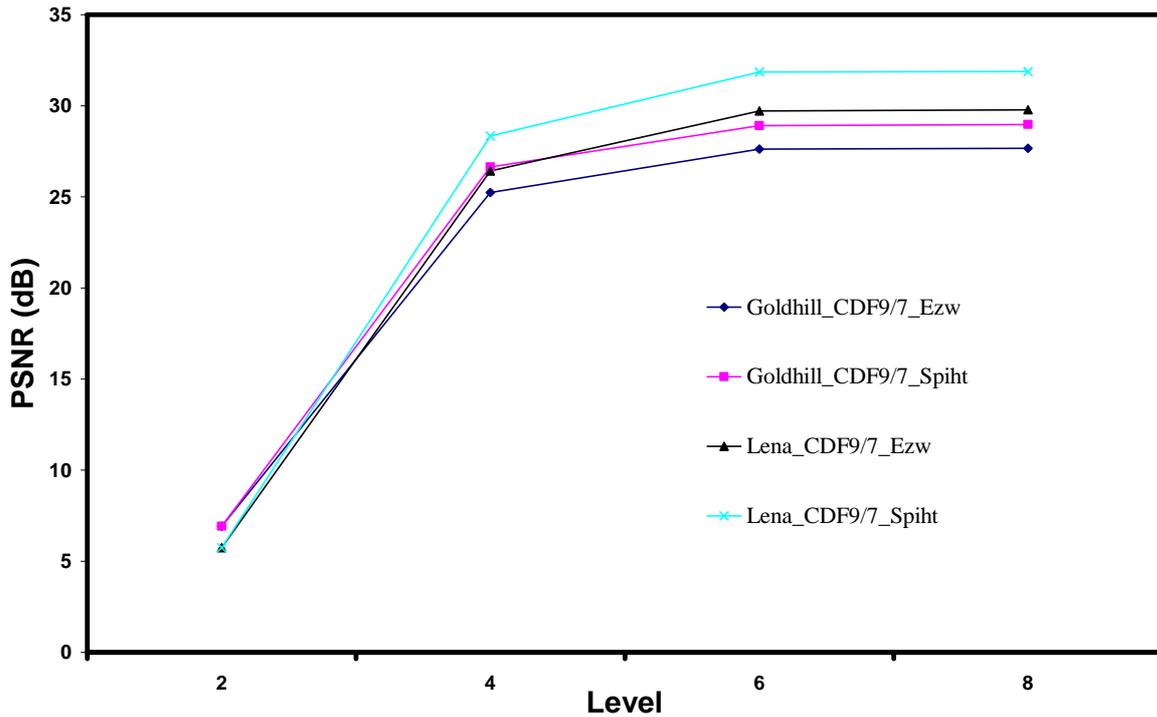
### 4.3.1 Application aux images tests

Pour évaluer ses performances, notre schéma de compression a été appliqué, aux images Lena et Goldhill. Comme indiqué dans le chapitre 3, notre méthode de compression repose sur les codeurs EZW et SPIHT appliqués dans le domaine des ondelettes entières utilisant deux types de filtres. À savoir l'ondelette biorthogonale CDF9/7 ainsi que l'ondelette de Gall5/3 en variant le niveau de décomposition (level).

Pour fixer le niveau de décomposition qui sera utilisé lors de la compression, nous avons, dans un premier temps, analysé les variations du PSNR qui représente la qualité des images reconstruites en fonction du niveau de décomposition. Les résultats obtenus pour un taux de compression de 0,2 bits/pixels, tout en utilisant les deux types d'ondelettes, ainsi que, nos deux types de codage, sont illustrés dans le tableau 3. La figure 26 illustre l'évolution du PSNR en fonction du niveau de décomposition (level).

level	Image Goldhill				Image Lena			
	CDF9/7 EZW	CDF 9 /7 SPIHT	Gall5/3 EZW	Gall5/3 SPIHT	CDF9/7 EZW	CDF 9 /7 SPIHT	Gall5/3 EZW	Gall5/3 SPIHT
2	6,92	6,92	11,07	11,07	5,75	5,75	12,26	12,26
4	25,23	26,64	25,05	26,10	26,43	28,34	26,07	27,59
6	27,61	28,91	24,08	25,45	29,72	31,86	25,38	26,28
8	27,66	28,97	22,09	24,10	29,78	31,88	23,68	25,14

**Tableau 3** Résultats des PSNR en fonction du niveau de décomposition (level).



**Fig. 26** Evolution du PSNR en fonction du niveau de décomposition level.

Nous constatons, à partir de ces résultats, que le PSNR augmente avec le niveau de décomposition pour les différentes ondelettes utilisées sur les deux images tests. Par conséquent, nous pouvons conclure que les performances des codeurs à arbres de zéros (EZW et SPIHT) sont maximales pour de grandes résolutions. Du coup, le nombre de niveaux de décomposition en ondelettes est fixé en fonction de la dimension de l'image à compresser ( $\text{level} = \log_2(\text{ht})$  avec ht représente la hauteur de l'image). Ce nombre est un compromis entre performances en compression et complexité calculatoire. De plus, un faible niveau de décomposition réduit le nombre de pixels à évaluer par les codeurs.

Par la suite, et après avoir fixé le niveau de décomposition (level), vu précédemment, nous avons effectué d'autres tests sur nos images références en faisant varier, cette fois-ci, le taux de compression en bits par pixel (bpp). Les résultats obtenus sont répertoriés respectivement dans les tableaux 4 et 5. Ils interprètent les différentes valeurs du PSNR ainsi que celles des temps de compression selon l'ondelette utilisée (respectivement CDF9/7 et Gall5/3), le codeur associé (EZW et SPIHT) mais aussi l'image test sur laquelle est appliquée notre méthode.

bits par pixel (bpp)	Temps de compression (s)				PSNR (dB)			
	Image Goldhill		Image Lena		Image Goldhill		Image Lena	
	EZW	SPIHT	EZW	SPIHT	EZW	SPIHT	EZW	SPIHT
0,1	1,75	5,73	1,61	5,08	25,99	26,87	26,67	28,71
0,2	4,49	12,15	3,71	9,15	27,67	28,98	29,79	31,87
0,3	5,67	20,06	4,47	19,61	28,36	30,05	31,08	33,04
0,4	9,99	37,16	9,15	29,36	29,65	30,76	32,54	34,98
0,5	20,79	61,91	13,73	42,25	30,31	31,34	33,46	35,69
0,6	36,24	82,54	15,30	60,44	30,94	32,74	33,96	36,16
0,7	38,79	102,75	17,07	95,58	31,43	33,40	34,85	36,47
0,8	41,59	147,96	27,50	127,36	31,93	33,88	35,41	37,23
0,9	43,77	193,99	47,13	167,82	32,27	34,30	35,91	37,94
1	45,37	250,99	64,28	215,68	32,76	34,65	36,20	38,29

**Tableau 4** Résultats des PSNR et des temps de compression en fonction du taux de compression (bpp) pour l'ondelette CDF9/7.

bits par pixel (bpp)	Temps de compression (s)				PSNR (dB)			
	Image Goldhill		Image Lena		Image Goldhill		Image Lena	
	EZW	SPIHT	EZW	SPIHT	EZW	SPIHT	EZW	SPIHT
0,1	1,93	5,99	1,70	5,30	22,18	18,31	18,46	15,77
0,2	2,83	10,73	2,66	10,74	22,51	24,55	20,23	21,62
0,3	8,08	25,54	6,42	21,09	25,45	24,91	25,25	24,85
0,4	18,91	44,33	11,19	27,18	25,82	25,16	25,49	26,38
0,5	29,04	74,08	12,66	44,30	26,00	27,14	25,60	26,54
0,6	30,75	94,05	14,26	68,10	26,21	28,19	30,08	26,61
0,7	33,01	113,23	24,13	112,95	26,33	28,47	30,47	26,67
0,8	33,87	150,70	43,97	161,58	27,75	28,65	30,69	29,73
0,9	44,19	197,69	62,95	188,61	30,35	28,79	30,78	32,24
1	71,25	268,24	67,88	230,09	30,59	28,89	30,93	32,66

**Tableau 5** Résultats des Psnr et du temps de compression en fonction du taux de compression (bpp) pour l'ondelette Gall5/3.

Nous donnons ci-après, les courbes des différentes évolutions, selon les différents paramètres de notre schéma de compression, à savoir le type de l'ondelette, le codeur utilisé tout en variant le taux de bits par pixel (bpp). Les figures 27 et 28 illustrent respectivement l'évolution du PSNR en fonction du taux en bpp des deux images Lena et Goldhill ; en variant le type d'ondelette pour les codeurs EZW et SPIHT.

Nous constatons, que l'ondelette CDF9/7 est celle qui permet d'obtenir de bons résultats car elle s'adapte mieux aux variations des différentes textures des images références, permettant ainsi, aux codeurs EZW et SPIHT de coder efficacement les différents coefficients d'ondelettes déduits de la transformation. De plus, nous remarquons que l'image Goldhill est très difficile à compresser à cause de sa complexité et le codeur SPIHT est celui qui permet d'atteindre les meilleurs PSNR.

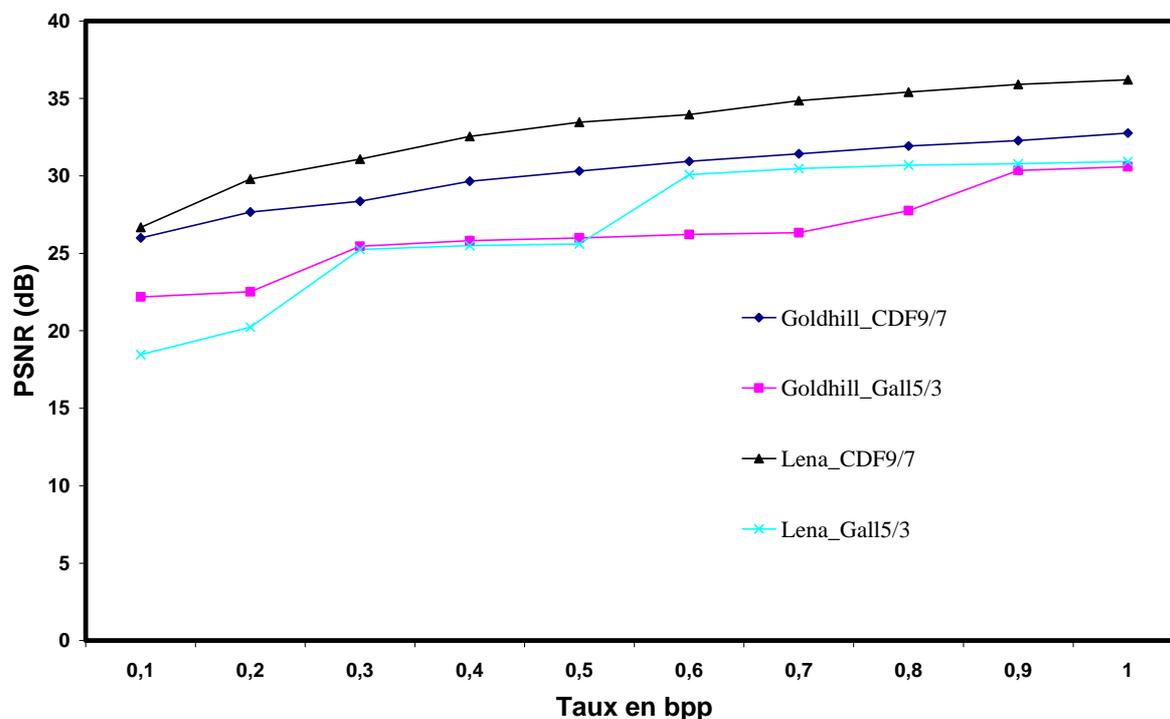
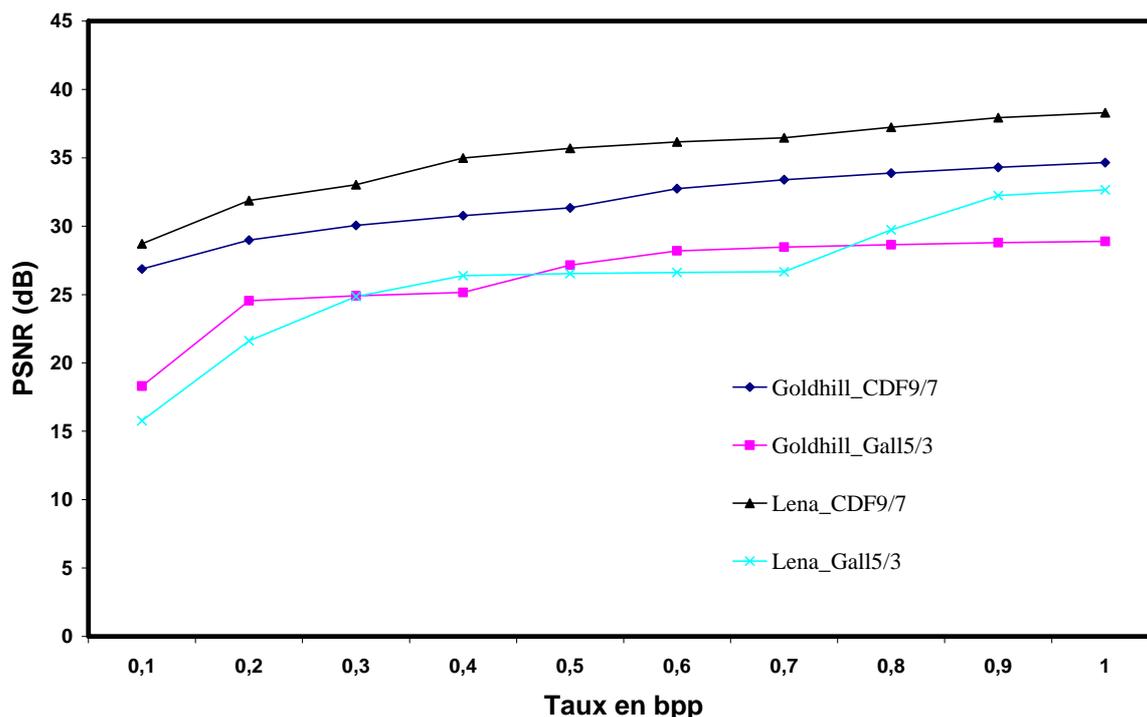


Fig. 27 Rapport signal sur bruit (PSNR) en fonction du taux en bpp avec le codeur EZW.



**Fig. 28** Rapport signal sur bruit (PSNR) en fonction du taux en bpp avec le codeur SPIHT.

Le critère le plus déterminant pour l'estimation d'un résultat est le critère visuel, car seul l'œil et l'analyse que fait le cerveau du signal capté peuvent juger la qualité des images. Les figures 29 et 30 présentent les images reconstruites après application de notre algorithme de compression sur les images Lena et Goldhill avec un débit de 0,5 bpp, c'est-à-dire, un taux de compression de 93,75 %. Ces résultats montrent clairement que c'est le filtre CDF9/7 suivi du codeur SPIHT qui donne le meilleur PSNR, qui vaut 31,34 dB pour l'image Goldhill et 35,69 dB pour l'image Lena. Ces images reconstruites sont de bonnes qualités visuelles, car, la majorité des détails des deux images sont restituées. Cependant, nous remarquons l'apparition d'artefacts très minimes qui s'accroissent au niveau des contours, dans le cas de l'application de l'ondelette Gall5/3.



a) Gall5/3\_EZW  
PSNR=26,00 dB



b) Gall5/3\_SPIHT  
PSNR=27,14 dB



c) Gall5/3\_EZW  
PSNR=25,60 dB



d) Gall5/3\_SPIHT  
PSNR=26,54 dB

**Fig. 29** Images reconstruites avec EZW et SPIHT associé au filtre Gall5/3 pour un taux de 0,5 bit/pixel



e) CDF9/7\_EZW  
PSNR=30,31 dB



f) CDF9/7\_SPIHT  
PSNR=31,34 dB



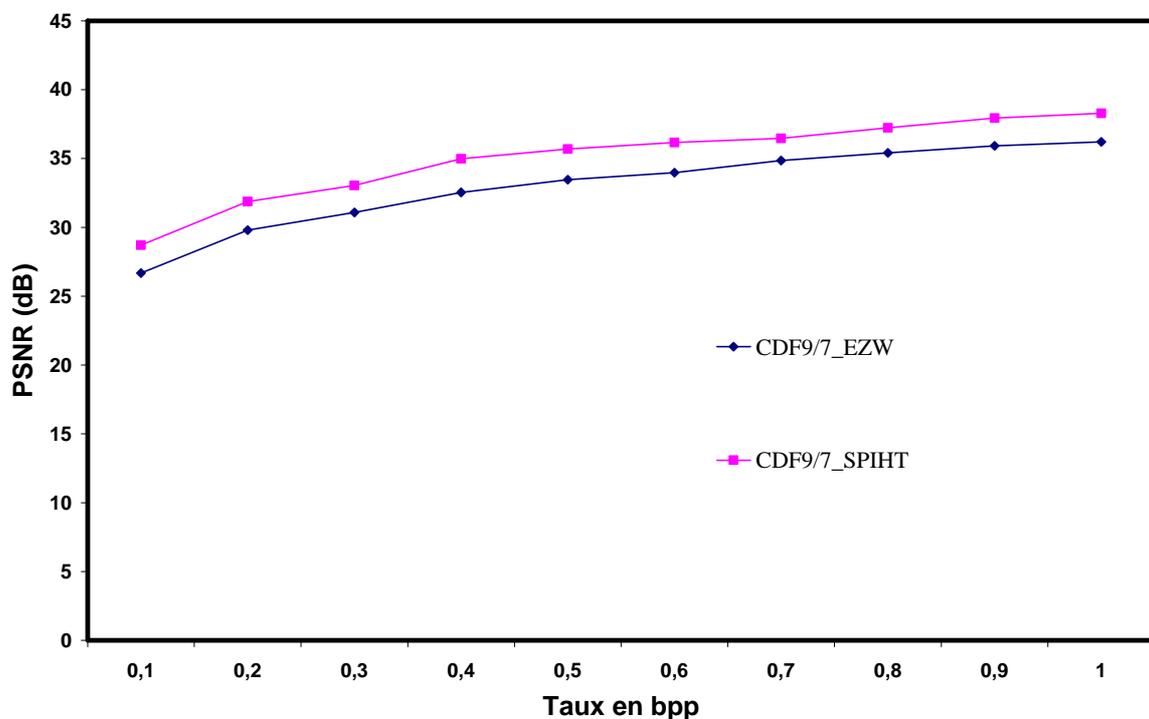
g) CDF9/7\_EZW  
PSNR=33,46 dB



h) CDF9/7\_SPIHT  
PSNR=35,69 dB

**Fig. 30** Image reconstruites avec EZW et SPIHT associé au filtre CDF9/7 pour un taux de 0.5 bits/pixel.

La figure 31 illustre l'évolution du PSNR en fonction du taux en bits par pixel dans le cas de l'image Lena. Dans cette figure, nous remarquons que le codeur SPIHT compresse efficacement et délivre les meilleures performances en terme de PSNR que EZW. Les images reconstruites pour un taux de 1 bit par pixel qui sont représentées dans la figure 32, accentuent la performance du codeur SPIHT appliqué à l'ondelette CDF9/7. En effet, le PSNR atteint 34,65 décibels et que l'image Goldhill reconstruite est de meilleur aspect visuel. Notons aussi, que la complexité de ces codeurs nécessite un temps d'exécution important pour de forts taux de compression (Fig.33).



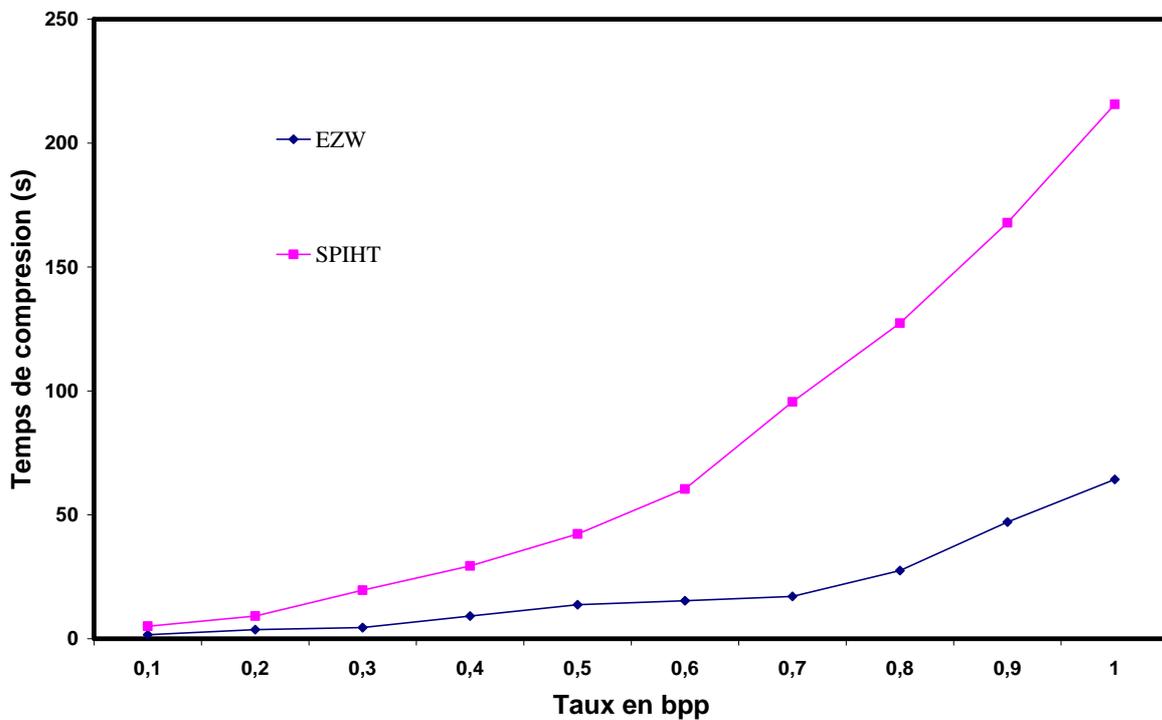
**Fig. 31** Rapport signal sur bruit (PSNR) en fonction du taux en bpp de l'image Lena en utilisant l'ondelette CDF9/7.



a) CDF9/7\_EZW  
PSNR=32,76 dB

b) CDF9/7\_SPIHT  
PSNR=34,65 dB

**Fig. 32** Images Godhill reconstruites avec EZW et SPIHT associé au filtre CDF9/7 pour un taux de 1 bits par pixel.

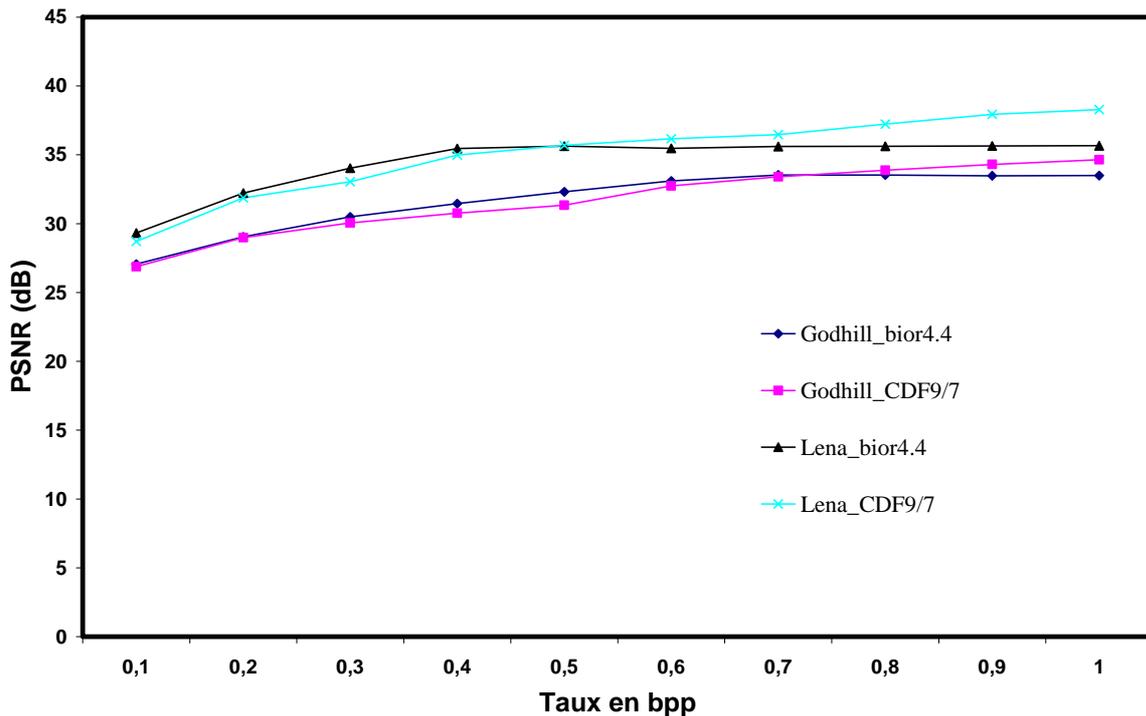


**Fig. 33** Temps de compression en fonction du taux en bpp dans le cas de l'image Lena.

Afin de mieux, localiser le degré de performance de l'ondelette entière CDF9/7, nous avons effectué une comparaison avec l'ondelette biorthogonale réelle « bior4.4 » en utilisant le codeur SPIHT. Les différents résultats obtenus par l'ondelette bior4.4 sont illustrés dans le tableau 6. La figure 34 montre l'évolution des PSNR en fonction du débit en bits par pixel pour les deux types d'ondelettes. Nous constatons que l'ondelette entière CDF9/7 prend le dessus à partir de 0,5 bits par pixel par rapport à l'ondelette réelle bior4.4.

bits par pixel (bpp)	Temps de compression (s)				PSNR (dB)			
	Image Goldhill		Image Lena		Image Goldhill		Image Lena	
	EZW	SPIHT	EZW	SPIHT	EZW	SPIHT	EZW	SPIHT
0,1	2,05	13,97	5,04	14,29	25,38	27,06	26,48	29,32
0,2	2,97	31,49	6,56	24,45	27,25	29,04	29,13	32,22
0,3	5,32	41,70	11,18	35,76	28,32	30,49	30,79	34,02
0,4	7,45	73,38	15,07	54,71	29,40	31,46	32,26	35,46
0,5	8,88	97,74	16,84	58,30	29,98	32,31	32,96	35,63
0,6	10,24	143,06	25,81	55,39	30,48	33,10	33,75	35,47
0,7	19,96	146,78	39,84	59,94	31,03	33,53	34,64	35,60
0,8	29,68	149,15	41,87	62,59	31,61	33,53	35,60	35,62
0,9	41,53	151,53	17,98	65,96	32,09	33,48	35,53	35,64
1	43,40	154,48	19,92	69,79	32,59	33,49	35,64	35,65

**Tableau 6** Résultats des PSNR et du temps de compression en fonction du taux en bpp pour l'ondelette bior4.4.



**Fig. 34** Rapport signal sur bruit (PSNR) en fonction du taux en bpp du codeur CDF9/7\_SPIHT.

Dans le but de valider notre méthode de compression, les résultats obtenus avec l'image Lena ont été comparés à ceux de diverses méthodes publiées dans la littérature [Jacquin, 1993; Lepsoy *et al.*, 1993 ; Thomas and Deravi, 1995 ; Davis, 1998 ; Chung-Su, 1998 ; Charier *et al.*, 1999 ; Ameer *et al.*, 2002 ; Lahdir *et al.*, 2006 ; Sadoun *et al.*, 2006]. De telles méthodes sont généralement basées sur les ondelettes, la DCT et les fractales. Celles que nous avons choisies pour la comparaison, sont répertoriées dans le tableau 7.

Le but de cette comparaison est d'identifier les méthodes qui conduisent aux critères de performances les plus grands possibles (le taux de compression et le PSNR). Pour faciliter cette comparaison, nous avons utilisé le facteur de performance (Q) prédéfini dans le travail de [Lahdir *et al.*, 2006] en compression d'images. Ce facteur (Q) est obtenu en effectuant le produit du taux de compression et du PSNR convertis en rapports sans dimensions. D'après le tableau 7, notre algorithme est le plus performant du point de vue de la compression et de la restitution d'une image. En résumé, les résultats représentés dans ce tableau montrent que notre algorithme de compression est celui qui réalise le meilleur compromis entre le taux de compression et la qualité de restitution de l'image.

Types d'algorithme	PSNR (dB)	Taux de compression C (%)	Facteur de performance Q
<b>Ondelettes entières CDF9/7 et SPIHT</b>	<b>35,69</b>	<b>93,75</b>	<b>3475,13176</b>
Ondelettes et fractales sans itération [Lahdir <i>et al.</i> , 2006]	32,90	93,75	1827,97931
DCT et QV en sous bandes [Ameur <i>et al.</i> , 2002]	32,20	93,75	1555,86273
JPEG2000 [Charier <i>et al.</i> , 1999]	32,22	93,75	1563,04426
Ondelettes et fractales multirésolutions [Sadoun <i>et al.</i> , 2006]	30,08	72,36	737,131766

**Tableau 7** Comparaison des résultats obtenus sur l'image Lena avec ceux publiés dans la littérature

#### 4.3.2 Application aux images MSG

Les tests effectués précédemment nous ont permis de fixer les différents paramètres de performances de notre algorithme de compression, à savoir, le type de l'ondelette entière qui est la biorthogonale CDF9/7, le niveau de décomposition (level) qui dépendra de la dimension de l'image à compresser, ainsi que, du codeur SPIHT à utiliser. Ces différents paramètres ont été choisis pour compresser les images MSG dans les gammes visible 0.6 et infrarouge 10.8 qui sont représentés sur les figures 25.a et 25.b. Le tableau 8 regroupe les différents résultats obtenus par application de l'algorithme CDF9/7\_SPIHT en variant le taux de bits par pixel (bpp).

bits par pixel (bpp)	PSNR (dB)		Temps de compression (seconde)	
	Image visible 0.6	Image infrarouge 10.8	Image visible 0.6	Image infrarouge 10.8
	CDF9/7_SPIHT	CDF9/7_SPIHT	CDF9/7_SPIHT	CDF9/7_SPIHT
0,1	22,83	25,23	5,80	6,31
0,2	25,30	27,12	12,05	14,53
0,3	27,11	28,88	17,35	27,36
0,4	28,80	30,31	28,18	40,98
0,5	30,33	31,14	38,35	56,86
0,6	31,18	31,93	52,74	88,93
0,7	31,95	33,58	76,49	106,55
0,8	33,34	34,44	90,27	134,96
0,9	34,23	35,09	117,78	164,63
1	34,85	35,65	149,20	212,74

**Tableau 8** Résultats obtenus par CDF9/7\_SPIHT sur les images MSG

A partir des valeurs regroupées dans le tableau 8, nous obtenons les courbes de la figure 35. Ces diagrammes donnent les valeurs du rapport signal sur bruit (PSNR) obtenues pour différentes valeurs du débit de compression en bpp. Elles montrent que le PSNR croît lentement avec le taux en bpp selon une loi qui, est approximativement logarithmique. Constatant bien que pour des débits très faibles, les PSNR sont de valeurs très faibles; par conséquent, visuellement des artéfacts apparaissent du fait que notre algorithme de codage n'arrive plus à discerner les vrais contours des textures fines.

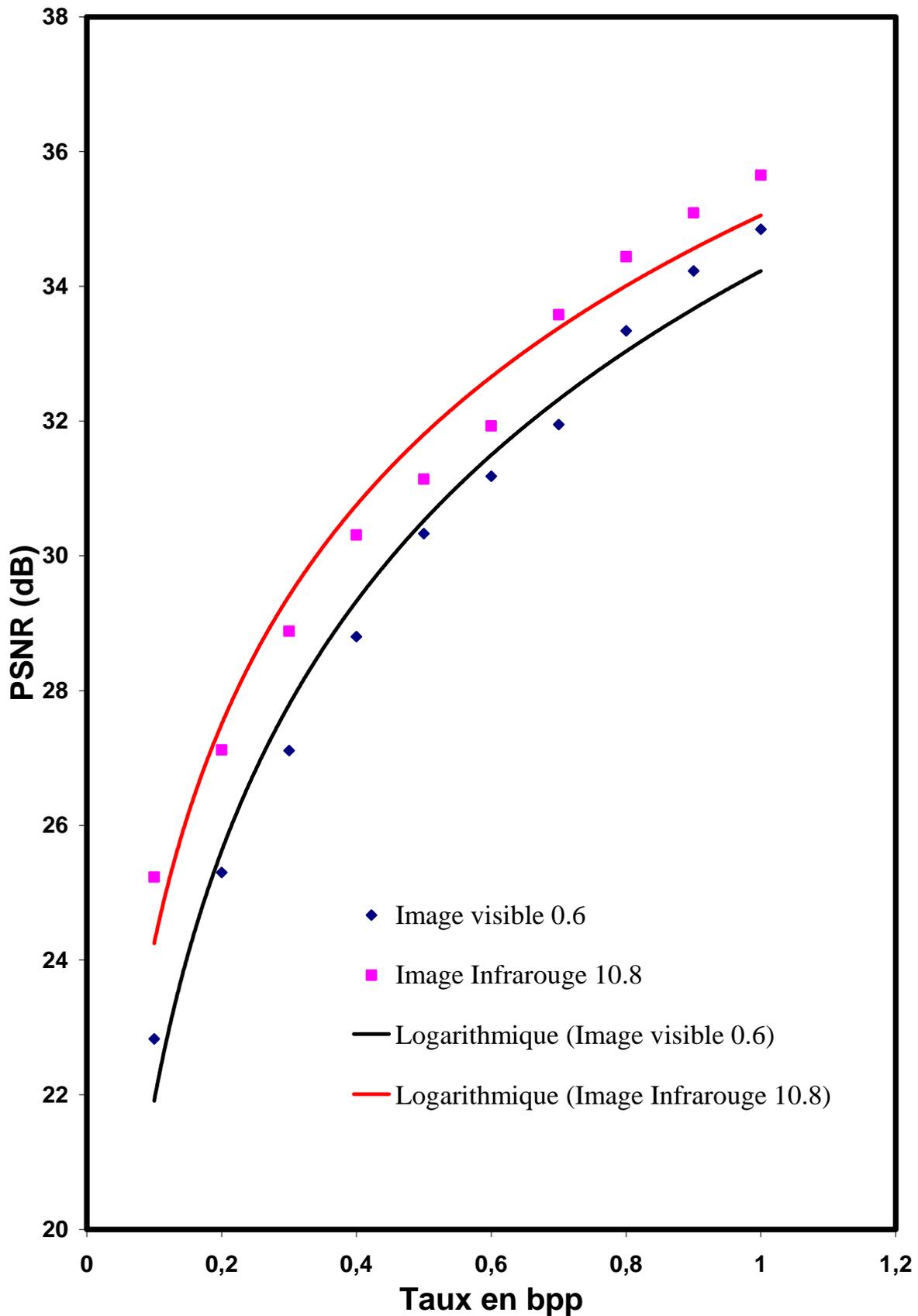
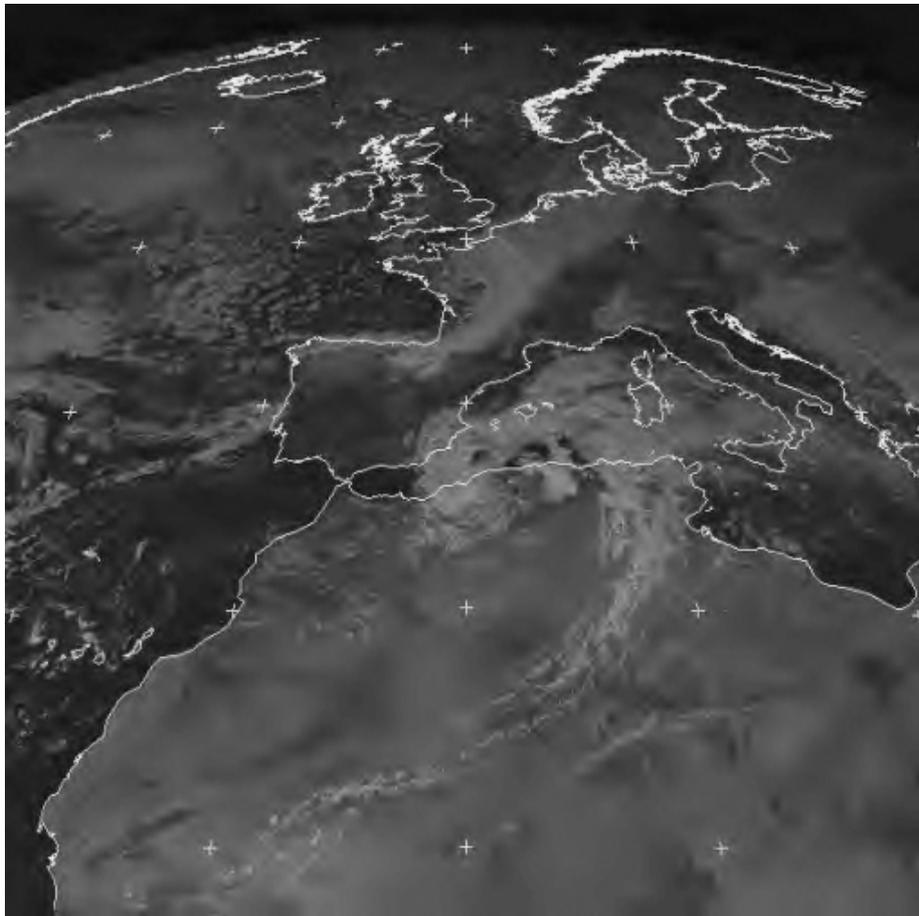
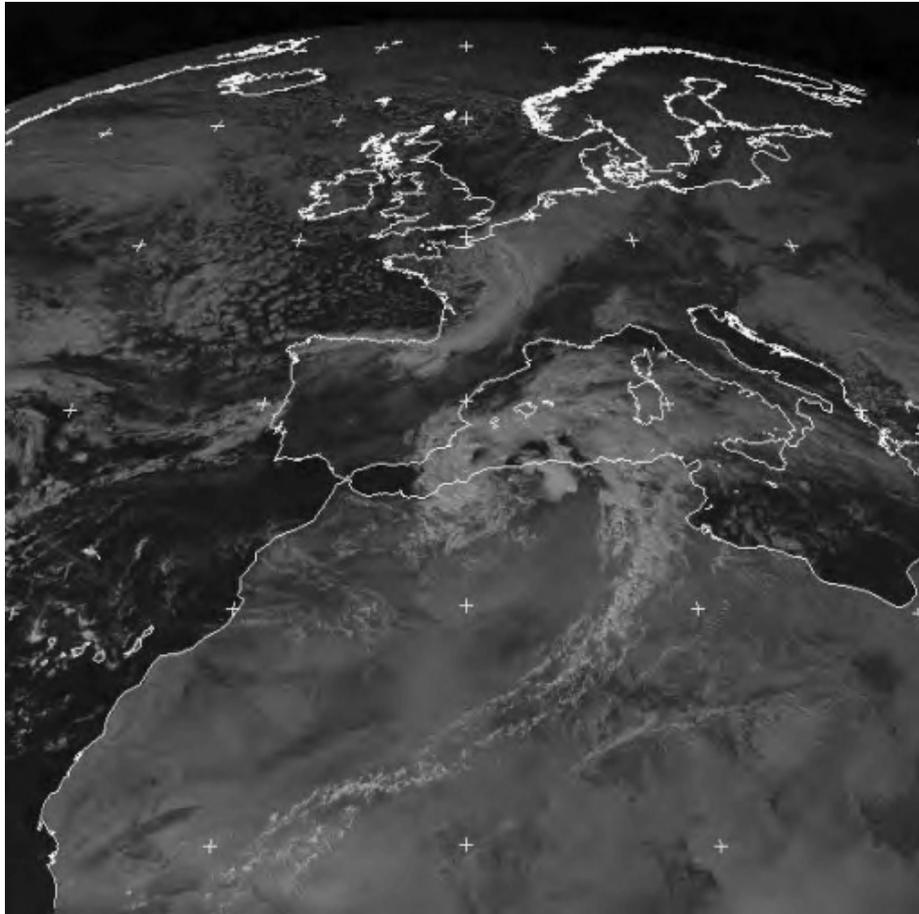


Fig. 35 Rapport signal sur bruit (PSNR) en fonction du taux de bits par pixel (bpp) pour les images MSG

Les images indiquées par les figures 36 et 38 résultent respectivement de la compression, puis de la décompression des images MSG, visible et infrarouge, avec un débit de compression de 0,5 bits par pixel, c'est-à-dire, un taux de compression de 93,75 %. Sur ces images, des artéfacts minimes apparaissent tout en conservant leurs propriétés initiales. Pour un débit de 1 bits/pixel, les images des figures 37 et 39 sont convenablement reconstruites. En effet, les zones claires de l'image visible 0.6 et les détails plus fins liés à la texture des nuages ou à celle des zones d'ombre ont été bien reconstitués. Les nuages, les types de sol, la mer Méditerranée et l'océan Atlantique apparaissant sur l'image originale sont bien reproduits et les contours bien définis. En ce qui concerne l'image infrarouge 10.8, les nuances de gris des régions froides et celles des régions chaudes sont convenablement reproduites.



**Fig. 36** Reconstitution de l'image visible 0.6 (taux de 0,5 bits/pixel et PSNR = 30,33 dB)



**Fig. 37** Reconstitution de l'image visible 0.6 (taux de 1 bit/pixel et PSNR = 34,85 dB)



**Fig. 38** Reconstitution de l'image Infrarouge 10.8 (taux: 0,5 bits/pixel et PSNR=31,14 dB)



**Fig. 39** Reconstitution de l'image Infrarouge 10.8 (taux : 1 bit/pixel et PSNR= 35,65 dB)

#### 4.4. Discussion

L'algorithme de compression et de décompression des images, ainsi, développé est capable de compresser toute image avec de forts taux de compression et de la restituer avec une qualité appréciable. Ainsi, les tests effectués sur l'image Lena et Godhill prises pour références, et sur des images météosat (visible et infrarouge) collectées par le satellite MSG ont montré la supériorité de cet algorithme vis à vis d'autres méthodes de compression, telle que : celles basées sur les ondelettes. En effet, notre algorithme réalise un très bon compromis entre le taux de compression, la qualité de restitution d'une image et la limitation du nombre

d'altérations. De telles performances ont pu être atteintes grâce à la combinaison de l'ondelette entière biorthogonale CDF9/7 et le codeur SPIHT que nous avons adopté pour réaliser le processus de compression et de décompression des images, sachant que, le PSNR a atteint 35,69 dB pour un débit de 0.5 bits/pixel dans le cas de l'image référence Lena. Concernant les images MSG, notre algorithme de compression a pu atteindre, pour un débit de 0,5 bits/pixel, un PSNR de 30,33 décibels pour l'image visible 0.6 et un autre de 31,14 décibels pour l'image infrarouge 10.8. Visuellement, nous avons constaté que les images reconstruites sont d'un aspect visuel appréciable car pratiquement, tous les détails des images originales sont bien restitués.

Par conséquent, notre algorithme appliqué à l'ensemble des images MSG enregistrées durant l'année 2007 a permis de réduire de près de 90 % le volume des données satellitaires à sauvegarder, tout en gardant leurs propriétés initiales. Il représente, donc, un outil puissant qui rend possible l'archivage d'images météorologiques sur de longues périodes, et par conséquent, l'étude des changements climatiques à long terme.

# Conclusion

Dans le contexte actuel de développement des communications numériques à l'échelle planétaire, la compression des données échangées ou stockées est une opération inévitable afin de remplir les fonctionnalités d'archivage et de transmission rapide.

Dans ce mémoire, nous avons exposé, le fait que la compression dite "sans perte" ne permettait pas une réduction significative du volume de ces données. Nous avons ensuite investigué la compression avec pertes maîtrisées, à savoir, des pertes n'affectant pas la qualité des images. Une nouvelle méthode basée sur une transformée en ondelettes, qui possède des propriétés intéressantes du point de vue de la compression, a été élaborée et implémentée. En particulier, nous avons utilisé une transformation en ondelettes par schéma de lifting implémentant deux types de filtres : CDF 9/7 et Gall 5/3. Une fois la décomposition effectuée, nous nous sommes intéressés au codage des coefficients d'ondelettes. Sur cette décomposition, plusieurs structures d'arbres peuvent être déterminées. Ainsi, différents algorithmes de compression basés sur cette transformation et les structures d'arbres sont définis. La première méthode est une adaptation de l'algorithme EZW, la seconde est celle de SPIHT. Leurs résultats sont comparés à d'autres algorithmes, à savoir, le standard le plus renommé en compression : JPEG 2000.

Sur l'ensemble des images références, l'algorithme de compression, basé sur l'ondelette CDF 9/7 et le codeur SPIHT, proposé, permet d'atteindre des PSNR de l'ordre de 35,69 dB pour des débits de 0,5 bpp. Tandis que notre application sur les images MSG, nous a fait atteindre des PSNR de 30,33 dB pour l'image visible et un PSNR de 31,14 dB pour l'image infrarouge.

Cette étude montre l'intérêt des codeurs à arbres de zéros (Zerotree) adaptés sur une décomposition en ondelette entière pour la compression des images. Avec une complexité d'implémentation, SPIHT présente des performances nettement supérieures à celles de JPEG 2000, tout en produisant un train binaire emboîté. Notons bien que SPIHT atteint ces performances sans codeur arithmétique. Ces propriétés sont particulièrement intéressantes

dans le contexte de la compression des images satellitaires. Des améliorations sont toujours possibles, particulièrement dans le cas de SPIHT si nous arrivons à trouver un moyen de tirer partie de la structure d'arbre 3D dans le but d'une application aux images hyperspectrales, ou bien encore lui faire associer le codeur arithmétique pour améliorer ces performances.

# Annexe A

## Les ondelettes biorthogonales

### A.1 Les ondelettes biorthogonales

Les bases d'ondelettes biorthogonales sont une généralisation des bases orthogonales qui permet, en s'affranchissant de la contrainte d'orthogonalité, d'obtenir des bases d'ondelettes symétriques, de forte régularité et à support fini dont les filtres associés sont à phase linéaire et à reconstruction exacte. Les ondelettes biorthogonales ont notamment été utilisées pour le débruitage des signaux et des images, la reconnaissance d'empreintes digitales et la résolution de certains problèmes mathématiques (calcul matriciel et différentiel). Pour les applications en traitement d'images, il est important que les filtres associés aux fonctions échelle et ondelette soient à reconstruction exacte et à phase linéaire ou nulle pour préserver les contours de l'image et éviter les effets de bords. Ces filtres doivent être à supports courts pour diminuer le nombre d'opérations à effectuer lors des calculs et permettre ainsi, une exécution rapide des traitements. De plus, un ordre élevé de régularité est souhaitable, en particulier en compression d'images. Cependant, ces critères ne peuvent être satisfaits simultanément dans le cadre des bases orthogonales. En effet, parmi les filtres RIF orthogonaux, symétriques et à reconstruction exacte, seul le filtre de Haar (dont l'ordre de régularité est faible) peut satisfaire ces critères [Cohen *et al.*, 1992].

Une analyse Multirésolution biorthogonale de  $L^2(\mathbb{R})$  est définie par deux suites de sous espaces  $\{V_j\}_{j \in \mathbb{Z}}$  et  $\{\tilde{V}_j\}_{j \in \mathbb{Z}}$  ayant les propriétés suivantes:

- \*  $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$  et  $\bigcap_{j \in \mathbb{Z}} \tilde{V}_j = \{0\}$
- \*  $\bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R}^n)$  et  $\bigcup_{j \in \mathbb{Z}} \tilde{V}_j = L^2(\mathbb{R}^n)$
- \*  $V_j \subset V_{j+1}$  et  $\tilde{V}_j \subset \tilde{V}_{j+1}$

et de deux fonctions  $\phi(x)$  et  $\tilde{\phi}(x)$  de norme 1 dans  $L^2(\mathbb{R})$  engendrant des bases non orthogonales de  $V_j$  et  $\tilde{V}_j$  respectivement, tel que la base  $(\phi_{jk}, \tilde{\phi}_{jk})$  soit biorthogonale :

$$\langle \phi_{jk}, \tilde{\phi}_{jl} \rangle = \delta_{kl} \quad (\text{A.1})$$

On peut définir alors deux suites de carrés sommables  $h_k$  et  $\tilde{h}_k$  telles que:

$$\phi(x) = \sum_{k=-\infty}^{+\infty} h(k) \phi(2x-k) \quad (\text{A.2})$$

$$\tilde{\phi}(x) = \sum_{k=-\infty}^{+\infty} \tilde{h}(k) \tilde{\phi}(2x-k) \quad (\text{A.3})$$

On introduit deux fonctions ondelettes  $\psi$  et  $\tilde{\psi}$  définies par:

$$\psi(x) = \sum_k g_k \phi(2x-k) \quad , \quad g_k = (-1)^{k-1} \tilde{h}(1-k) \quad (\text{A.4})$$

$$\tilde{\psi}(x) = \sum_k \tilde{g}_k \tilde{\phi}(2x-k) \quad , \quad \tilde{g}_k = (-1)^{k-1} h(1-k) \quad (\text{A.5})$$

Le système  $(\psi_{jk}, \tilde{\psi}_{jk})$  est appelé base biorthogonale de  $L^2(R)$ .

Les ondelettes biorthogonales permettent d'avoir des ordres de régularité plus grands que ceux des ondelettes orthogonales pour des longueurs de filtres équivalentes. Elles permettent aussi d'avoir un grand nombre de moments nuls pour un support réduit. Ces qualités sont très appréciées dans diverses applications et notamment en compression d'images. Les ondelettes biorthogonales comptent plusieurs familles (pseudo-coiflets, filtres récursifs...), mais la plus usuelle est la famille des ondelettes splines.

## A.2 Choix d'une ondelette en compression d'images

Etant donné la grande variété d'ondelettes, la question fondamentale à se poser lors de l'utilisation de ces fonctions est, quelle est la meilleure ondelette pour cette application ? La question ainsi formulée est très vaste et n'a pas de réponse absolue. Une approche de proposition de solution est l'étude des propriétés spécifiques des ondelettes et d'analyser leurs influences sur le traitement envisagé. La théorie des ondelettes permet d'associer aux fonctions d'échelle et d'ondelette un certain nombre de propriétés intéressantes pour l'analyse et la synthèse des signaux. Les principales propriétés de la fonction d'ondelettes sont :

- **Support** : Si la fonction d'échelle  $\Phi$  et l'ondelette  $\Psi$  sont à support compact, les filtres  $H$  et  $G$  sont à réponse impulsionnelle finie (FIR), et sont propices à une implantation informatique. Dans le cas contraire, une décroissance rapide des filtres permet tout de même de les approcher raisonnablement par des filtres FIR.
- **Coefficients rationnels** : Dans une implantation informatique, il est souhaitable de réaliser la transformation le plus rapidement possible avec le moins d'occupation de l'espace mémoire. Il sera donc intéressant de ne pas effectuer les calculs en réel. Les filtres ayant des coefficients  $h_k$  et  $g_k$  rationnels (ou même dyadiques) sont donc très recherchés. Certaines précautions permettent même d'effectuer les transformations directement avec des entiers.
- **Orthogonalité** : Si les ondelettes sont orthogonales, la transformée en ondelettes est stable et réversible. Les opérateurs de projections dans les différents sous-espaces  $V_j$  et  $W_j$  conduisent à des approximations optimales au sens de  $L^2$ .
- **Symétrie** : Si les fonctions d'échelle et d'ondelette sont (anti-)symétriques, les filtres sont alors à phase linéaire. On observe une distorsion de la phase à la reconstruction lorsque cette propriété est absente. En fait, il n'existe pas d'ondelettes orthogonales qui soient à la fois symétriques et à support compact. C'est pourquoi, les ondelettes biorthogonales ont été introduites et sont abondamment utilisées.
- **Régularité** : La régularité de l'ondelette est importante en compression. Les compressions correspondent souvent à des mises à zéro de certains coefficients d'ondelettes, ce qui équivaut à négliger les composantes  $d_k^j = \langle f, \Psi_{j,k} \rangle$  de la fonction originale. La régularité de l'ondelette permet de masquer les erreurs de reconstruction. De plus, une plus grande régularité entraîne une meilleure localisation fréquentielle des filtres.
- **Moments nuls** : Ceci se traduit graphiquement, par plusieurs passages par zéro de l'ondelette. Soit  $N$  ( $N \geq 1$ ) le nombre de moments nuls, on définit les moments d'ordre  $p$  ( $p \geq 0$ ) de la fonction d'ondelette  $\Psi$  par :

$$N_p^\Psi = \int_{-\infty}^{+\infty} \psi(x) x^p dx$$

$N \Psi_p=0$  pour  $0 \leq p < N$ , cela signifie que la projection de tout polynôme de degré inférieur à  $N$  est nulle. Par conséquent, les coefficients d'ondelettes  $\langle f, \Psi_{j,k} \rangle$  seront nuls. Cette observation est très intéressante pour la compression des signaux car plus  $\psi$  aura de moments nuls, plus l'énergie (c'est-à-dire l'information) sera concentrée sur quelques coefficients. Le nombre de moments nul, est également important pour la détection des singularités dans un signal. La valeur de  $N$  joue également un rôle sur la régularité de  $\Phi$  qui est inférieure à  $(N-1)$ .

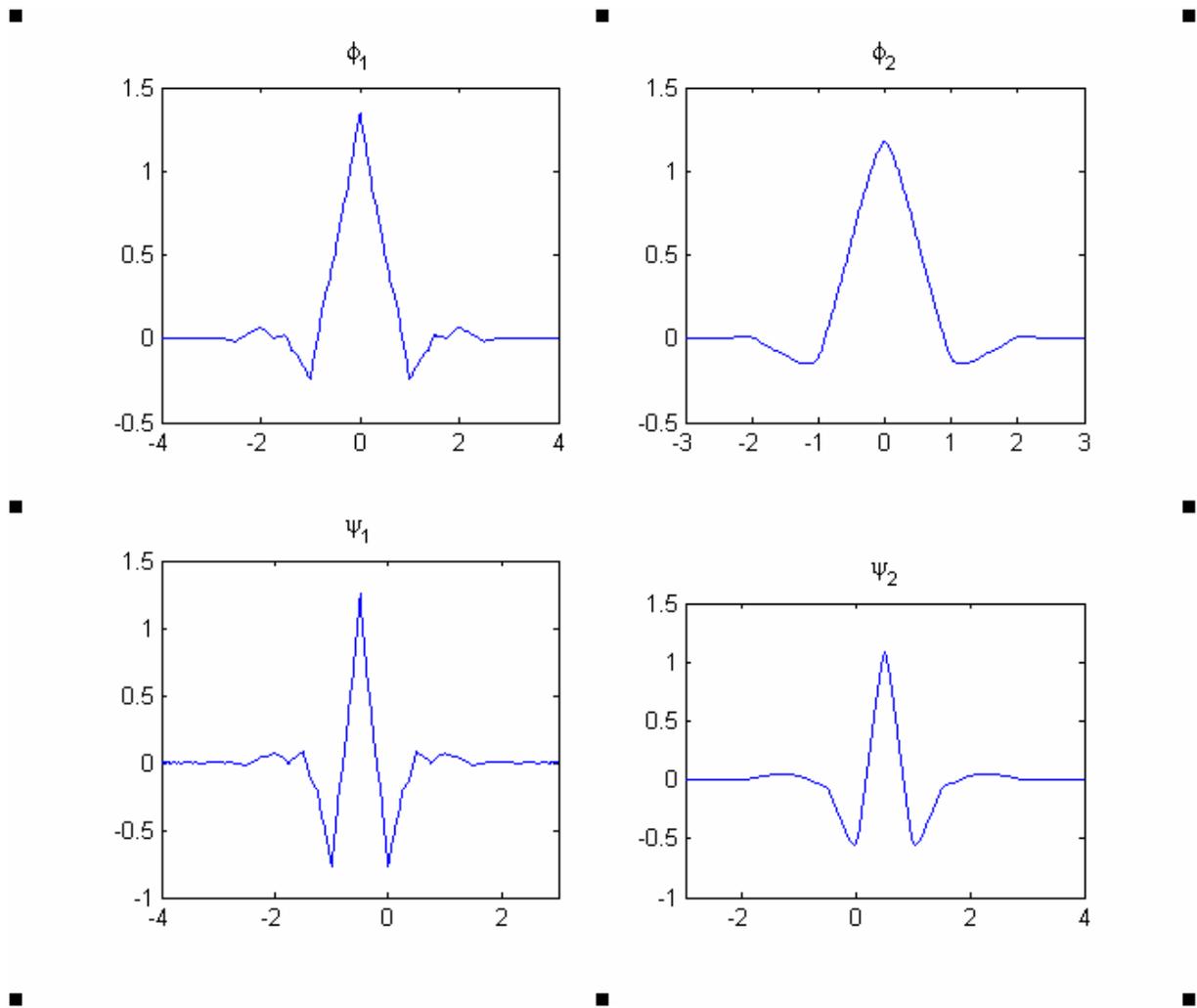
- **Localisation** : Elle est obtenue en faisant varier deux paramètres, l'un d'échelle et l'autre de translation. L'ajustement de ces deux paramètres ' $b$ ' (translation), ' $a$ ' (dilatation) de l'ondelette lors de l'analyse du signal, fait agir une ondelette comme un zoom à effet de réduction ou d'agrandissement. Cette propriété est très intéressante dans le cas de l'imagerie, étant donné qu'il faut pouvoir caractériser assez précisément les phénomènes locaux, ou les singularités en général, dans des applications comme la segmentation.

### A.3 L'ondelette de Cohen-Daubechies-Feauveau (CDF 9/7)

L'ondelette biorthogonale CDF9/7 (9 et 7 étant la longueur des filtres passe-bas et passe-haut) est aussi appelée (4,4) (les valeurs correspondant aux nombres de moments nuls des filtres passe-haut et passe-bas). Cette ondelette est quasi orthogonale et donc particulièrement appréciée dans le cas de la compression avec pertes, c'est d'ailleurs l'ondelette par défaut pour la compression avec pertes de JPEG2000. Le nombre d'étages en lifting est de 4. Les valeurs numériques des coefficients des filtres sont présentées dans le tableau A.1.

	0	$\pm 1$	$\pm 2$	$\pm 3$	$\pm 4$
$h_a^0$	0.85269868	0.37740286	-0.11062440	-0.023849465	0.037828456
$h_a^1$	0.78848562	-0.41809227	-0.04068942	0.064538883	
$h_s^0$	0.78848562	0.41809227	-0.04068942	-0.064538883	
$h_s^1$	0.85269868	-0.37740286	-0.11062440	0.023849465	0.037828456

**Tableau A.1.** Filtres utilisés pour l'ondelette CDF9/7.



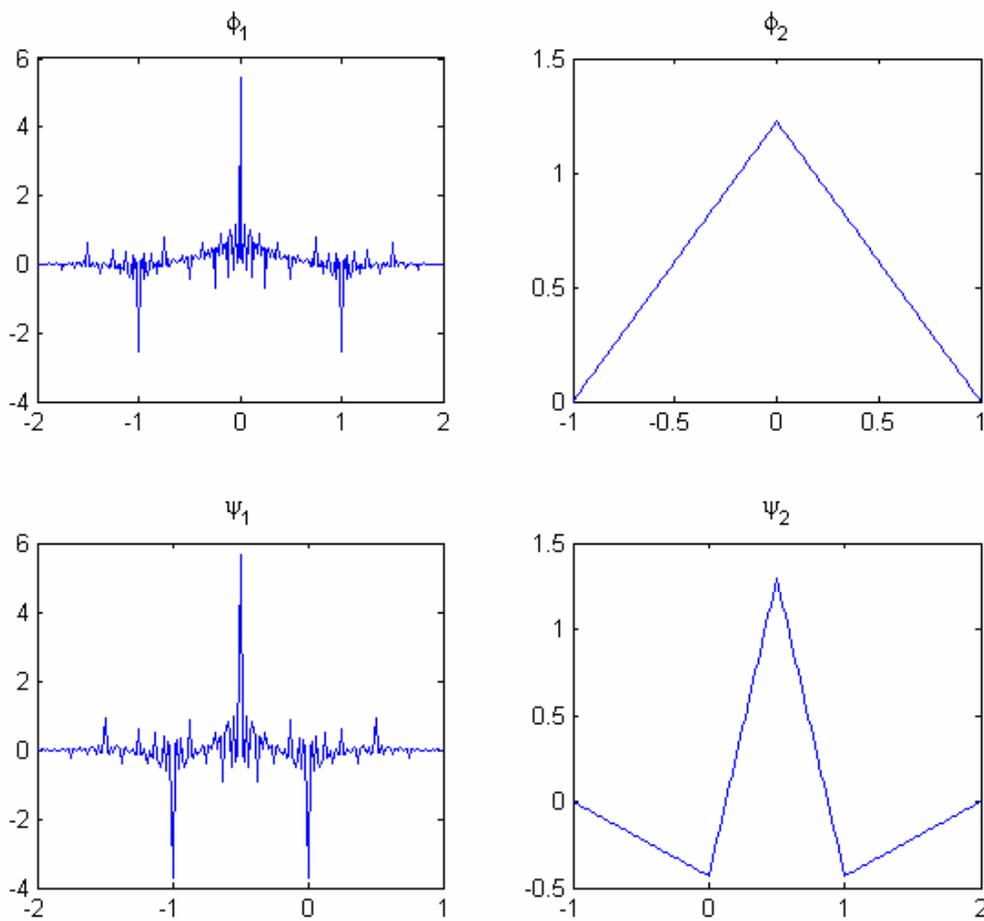
**Fig. A.1** La représentation de l'ondelette CDF9/7.

#### A.4 L'ondelette biorthogonale de Gall 5/3

L'ondelette Gall5/3 (5 et 3 étant la longueur des filtres passe-bas et passe-haut) est aussi appelée spline (2,2) (les valeurs correspondant ici aux nombres de moments nuls des filtres passe-haut et passe-bas). Cette ondelette semble donner de meilleures performances pour la compression sans pertes, c'est l'ondelette par défaut du standard JPEG2000 pour la compression sans pertes. Le nombre d'étages en lifting est de 2. Les valeurs numériques des coefficients des filtres sont présentées dans le tableau A.2

	0	$\pm 1$	$\pm 2$
$h_a^0$	6/8	2/8	-1/8
$h_a^1$	1	-1/2	
$h_s^0$	1	1/2	
$h_s^1$	6/8	-2/8	-1/8

**Tableau A.2** Filtrés utilisés pour l'ondelette Gall5/3.



**Fig. A.2** La représentation de l'ondelette Gall5/3.

# Annexe B

## Exemples pour EZW et SPIHT

Les algorithmes EZW et SPIHT sont populaires et utilisées dans beaucoup de travaux, néanmoins, pour en comprendre les subtilités, il est utile d'appliquer à la main ces algorithmes sur des données simples. Pour permettre de mieux comprendre ces deux algorithmes (décrits dans le chapitre 3 ainsi que dans [Shapiro, 1993] et [Said and Pearlman, 1996]) et mettre en évidence leurs différences, on applique ici la première étape de ces deux algorithmes sur un exemple concret.

### B.1 Exemple de codage EZW

Pour EZW, rappelons qu'un coefficient est déclaré significatif (symbole P ou N suivant son signe) si sa valeur absolue est supérieure au seuil  $T_n$  du plan de bits. Sinon, soit il s'agit d'un zéro isolé (IZ) si au moins un de ses descendants est significatif, soit d'un arbre de zéros (ZTR) si aucun de ses descendants n'est significatif.

Nous prendrons un exemple de 4x4 coefficients d'ondelettes d'une image :

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Fig. B.1 Exemple de coefficients issus de la transformée en ondelettes

Sous bandes	Valeurs des coefficients	Symbole	
LL3	63 (0,0)	P	(1)
HL3	-34 (1,0)	N	
LH3	-31 (0,1)	IZ	(2)
HH3	23 (1,1)	ZTR	(3)
HL2	49 (2,0)	P	
HL2	10 (3,0)	ZTR	
HL2	14 (2,1)	ZTR	
HL2	-13 (3,1)	ZTR	
LH2	15 (0,2)	ZTR	
LH2	14 (1,2)	IZ	
LH2	-9 (0,3)	ZTR	
LH2	-7 (1,3)	ZTR	
HL1	7 (4,0)	Z	(4)
HL1	13 (5,0)	Z	
HL1	3 (4,1)	Z	
HL1	4 (5,1)	Z	
HL1	-1 (2,4)	Z	
HL1	47 (3,4)	P	(5)
HL1	-3 (2,5)	Z	
HL1	-2 (3,5)	Z	

**Tableau B.1** Déroulement de la première étape de l'algorithme EZW sur les données de la figure B.1

Quelques précisions sont données pour certaines étapes, les numéros correspondants sont donnés dans la dernière colonne du tableau B.1 :

- Le premier coefficient a la valeur 63, il est supérieur au seuil courant initial est 32, le symbole P est émis.
- -31 est non significatif, mais il possède un descendant qui est significatif (47) c'est un zéro isolé, IZ est émis.
- 23 est non significatif et tous ses descendants sont non significatifs, il s'agit donc d'un arbre de zéros, ZTR est émis.
- 7 est non significatif et comme la sous-bande HL1 n'a pas de descendant, les symboles IZ et ZTR peuvent être regroupés en un seul symbole Z.
- 47 est significatif, le symbole P est émis. Comme ce symbole sera traité pendant l'étape de raffinement, il pourra être considéré comme étant 0 pour les étapes suivantes (le symbole 14 de la LH2 sera alors une racine d'un arbre de zéros).

En supposant que EZW utilise 2 bits pour coder les symboles P, N, ZTR et IZ et 1 bit pour coder le symbole Z, nous obtiendrons un débit de  $26+7=33$  bits.

## B.2 Exemple de codage SPIHT

L'algorithme SPIHT maintient trois listes de coefficients :

la liste des coefficients significatifs (LCS= LIP),

la liste des coefficients non significatifs (LCN= LSP),

Et la liste des ensembles non significatifs (LEN= LIS).

$O(x, y)$  est l'ensemble des enfants de  $(x, y)$  (un seul niveau de descendance),

$D(x, y)$  est l'ensemble de tous les descendants

$L(x, y) = D(x, y) - O(x, y)$  est l'ensemble des descendants à l'exception des enfants.

La fonction  $S_{T_n}$  est égale à 0 si tous les descendants de  $(x, y)$  sont en dessous du seuil  $T_n$  (arbre de zéros), et 1 dans le cas contraire. SPIHT considère deux types d'arbres de zéros : le type A où tous les descendants ne sont pas significatifs (arbre de degré 1) et le type B où tous les descendants, à l'exception d'au moins un des enfants, ne sont pas significatifs (arbre de degré 2). Le déroulement de l'algorithme SPIHT sur les données de la figure B.2 est décrit comme suit :

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

**Fig. B.2** Exemple de coefficients issus de la transformée en ondelettes

Initialisation

<u>LIP</u>	
(0,0)	→ 26
(0,1)	→ 6
(1,0)	→ -7
(1,1)	→ 7

<u>LSP</u>	

$$n = \lfloor \log_2(26) \rfloor = 4$$

<u>LIS</u>	
(0,1) <i>D</i>	→ {13, 10, 6, 4}
(1,0) <i>D</i>	→ {4, -4, 2, -2}
(1,1) <i>D</i>	→ {4, -3, -2, 0}

T=16

Première étape de tri

<u>LIP</u>	
(0,1)	→ 6
(1,0)	→ -7
(1,1)	→ 7

<u>LSP</u>	
Empty	
-----	
(0,0)	→ 26

pas de raffinement

<u>LIS</u>	
(0,1) <i>D</i>	→ {13, 10, 6, 4}
(1,0) <i>D</i>	→ {4, -4, 2, -2}
(1,1) <i>D</i>	→ {4, -3, -2, 0}

1 1	0 0 0
Sig./+	Insig.

0 0 0
D sont insign.

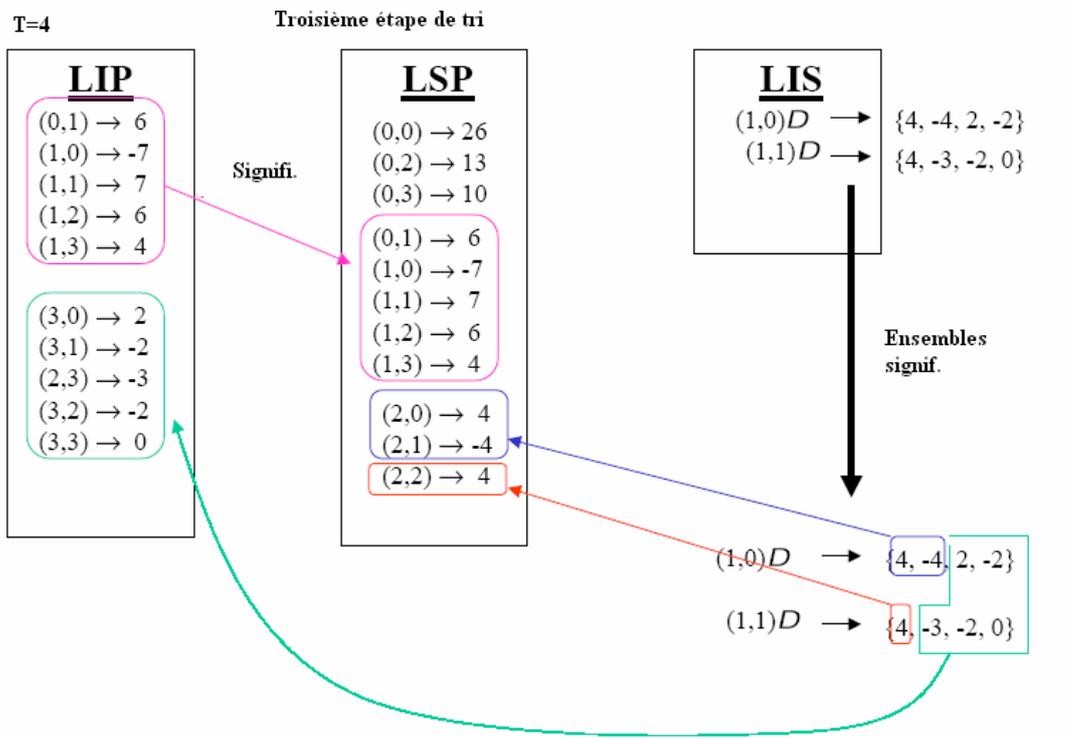
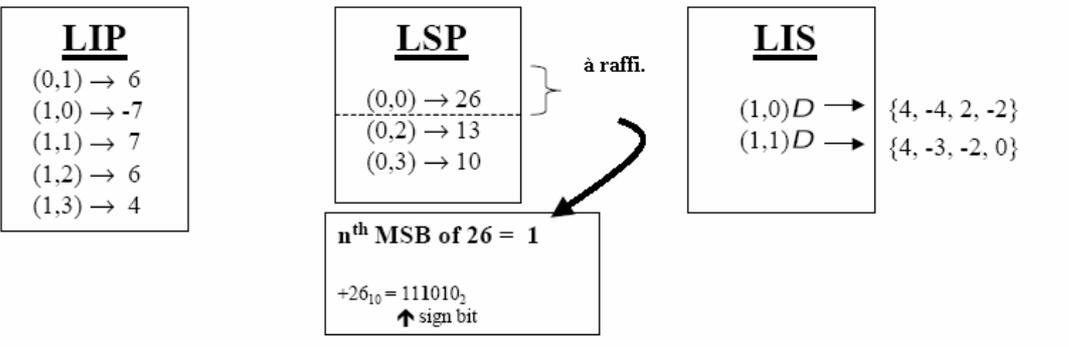
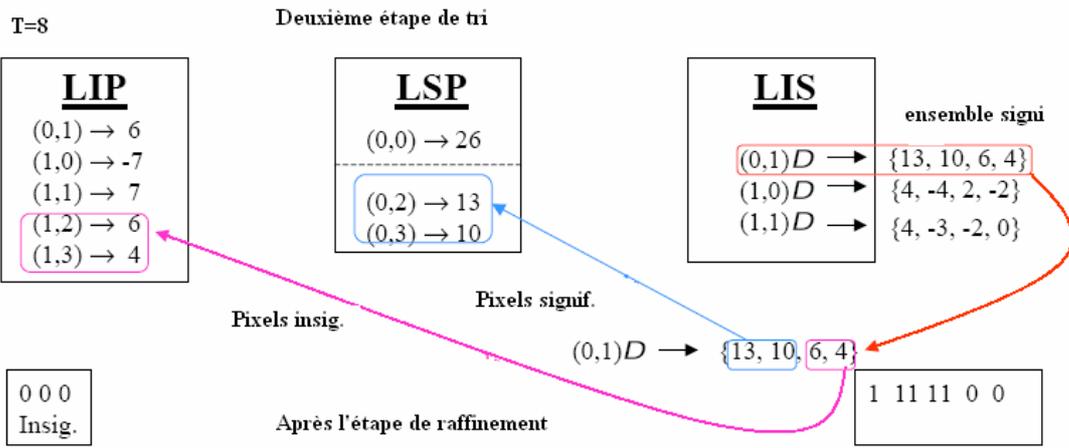
Après l'étape de raffinement

<u>LIP</u>	
(0,1)	→ 6
(1,0)	→ -7
(1,1)	→ 7

<u>LSP</u>	
(0,0)	→ 26

<u>LIS</u>	
(0,1) <i>D</i>	→ {13, 10, 6, 4}
(1,0) <i>D</i>	→ {4, -4, 2, -2}
(1,1) <i>D</i>	→ {4, -3, -2, 0}

Significatif



T=4

Après la troisième étape de tri

**LIP**

(3,0) → 2  
(3,1) → -2  
(2,3) → -3  
(3,2) → -2  
(3,3) → 0

**LSP**

(0,0) → 26  
(0,2) → 13  
(0,3) → 10  
(0,1) → 6  
(1,0) → -7  
(1,1) → 7  
(1,2) → 6  
(1,3) → 4  
  
(2,0) → 4  
(2,1) → -4  
(2,2) → 4

**LIS**

## *Références*

- Adams, M.D., and Kossentini, F. (1999) Evaluation of reversible integer-to-integer wavelet transforms for image compression. In proc of IEEE International Conference on Image Processing, volume 3, pages 541-545, Kobe, Japan, Octobre 1999
- Adams, M.D., (2002) Reversible integer to integer Wavelet Transforms for Image coding, Thèse de doctorat, Département of Electrical and Computer Engineering, Philosophy speciality, The University of British Columbia, 109 p.
- Akrour, L.** (2003) Compression d'images numériques par fractales dans le domaine DCT, mémoire d'ingénieur d'état en Electronique, Option Contrôle, Université de Mouloud Mammeri de Tizi-Ouzou, 114 p.
- Ammar, M. (2002) Optimisation d'un schéma de codage d'image à base d'une TCD. Application à un codeur JPEG pour l'enregistrement numérique à bas débit. Thèse de doctorat, Ecole nationale supérieure des télécommunications, signal et images, Paris, France, 143 p.
- Ameur, S., Adane, A. H. et Lahdir, M. (2002) Compression d'images Météosat en sous bandes par transformation discrète en cosinus et quantification vectorielle. Télédétection, vol. 2, no. 4, p. 255-266.
- Antonini, M., Barlaud, M., Mathieu, P. and Daubechies, L. (1992) Image coding using wavelet transform. IEEE Transactions on Image Processing, vol. 1, no. 2, p. 205- 220.
- Averbuch, A., Valery, A. and Zheludev, A. (2004) New Family of Spline-Based Biorthogonal Wavelet Transforms and Their Application to Image Compression. IEEE Transactions on Image Processing, vol. 13, no. 7, p. 993-1007.

- Averbuch, A., Lazar, D. and Israeli, M. (1996) Image compression using wavelet transform and multiresolution decomposition. *IEEE Transactions on Image Processing*, vol. 5, no. 1, p. 4-15.
- Ayad, M. et Chikouche, D. (2007) Application de la transformée en ondelette à l'analyse des signaux vibratoires d'un système d'engrenage en vue d'un diagnostic précoce, 4th International Conference on Computer Integrated Manufacturing CIP'2007.
- Baarir, Z.E.; Ouafi, A. (2004) Etude de la transformée en ondelettes dans la compression d'images fixes, *Courrier du Savoir*, N°5, p 69-74, 2004.
- Borko, F. (1995) A Survey of multimedia compression techniques and standards. Part I JPEG standard. *Real-Time Imaging*, no. 1, p. 49–67.
- Briellmann, C. (2006) Introduction au codage de l'information, Haute école Technique et Informatique HTI.
- Chariar, M., Santa Cruz, D. and Larsson, MC. (1999) JPEG2000, the next millennium compression standard for still images. In *Proc. of the IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, vol. 1, p. 131-132.
- Christophe, E. (2006) compression d'images hyperspectrales et son impact sur la qualité des données. Thèse de doctorat, Informatique et télécommunications Spécialité: Signal et image, université de Toulouse, France.
- Chaux,C. (2006) Analyse en ondelettes M-bandes en arbre dual ; Application à la restauration d'images, Thèse de doctorat, Spécialité : Traitement du signal, Université de Marne- la-Vallée, 177 p.

- Cziho, A. (1999) Quantification vectorielle et compression d'image, application à l'imagerie médicale. Thèse de doctorat, école doctorale en informatique, traitement du signal et télécommunications, université de Rennes 1, France, 160 p.
- Daubechies, I., Cohen, A. and Feauveau, J. (1992) Biorthogonal Bases of compactly supported Wavelets, *Comm. Pure. Appl. Math.*, vol. XLV, p. 485–560.
- Daubechies, I. (1988) Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, vol. 12, p. 909-996.
- Davoine, F.(1995)Compression des images fixes par fractale basées sur la triangulation de Delaunay, Thèse de doctorat, Institut national Polytechnique de grenoble, France.
- Daubechies, I. and Sweldens, W. (1988) Factoring wavelet transform into lifting steps. *Fourier Anal.Appl.* , vol. 41, N°3, p.247-269, 1998.
- Delaunay, X. (2008) Compression d'images satellite par post-transformées dans le domaine ondelettes. Thèse de doctorat, de l'université de Toulouse Délivré par l'Institut National Polytechnique de Toulouse, Discipline ou spécialité : Signal, Image, Acoustique et Optimisation, France.
- Delgorge, C. (2005) Proposition et Evaluation de techniques de compression d'images ultrasonores dans le cadre d'une télé échographie robotisée. Thèse de doctorat, Sciences et technologies industrielles université d'Orléans, France.
- Elkefi, A. et Antonini, M. (2003) Compression de maillage 3D multirésolutions, Transformée en ondelettes 2<sup>ème</sup> génération, mémoire d'ingénieur, spécialité Informatique, Université de Nice Sofia Antipolis, 82 p.

- Gaudeau, Y. (2006) Contributions en compression d'images médicales 3D et d'images naturelles 2D. Thèse de doctorat de l'Université Henri Poincaré, Nancy 1, Spécialité Automatique, Traitement du Signal et Génie Informatique, France.
- Guillemot, L. (2004) Une approche vectorielle pour exploiter le contenu de l'image en compression et tatouage. Thèse de doctorat de l'Université Henri Poincaré, Nancy 1, Spécialité Automatique, Traitement du Signal et Génie Informatique, France, 152 p.
- Huffman, D. A. (1952) A method for the construction of minimum redundancy codes, IRE Proc., vol. 40, p. 1098-1101.
- Isodore, P.A.B, Barret, M., Dinh-Tuan, P. (2007) Transformations optimales à haut débit pour le codage d'image multi-composantes selon la norme JPEG2000. Colloque GRETSI, 11 au 14 septembre 2007, Troyes, p. 489- 492.
- Jeannic, G., Ricorlel, V., Barba, D. (2007) Représentation structurelle d'images fixes par transformée en ondelettes orientées. Colloque GRETSI, 11 – 14 septembre 2007 ; Troyes, p. 461 – 464.
- Kubasov, D. (2008) Codage de sources distribuées: nouveaux outils et application à la compression vidéo. Thèse de doctorat de l'Université de Rennes1, spécialité informatique, France, 200 p.
- Langlet, R. (2004) Transformée en ondelettes : Application aux propriétés diélectriques et mécanique de nanostructures carbonées, Thèse de doctorat, Spécialité Physique, Université de Franche Comté, 173 p.
- Lahdir, M. (2008) Nouvelle approche de compression d'images basée sur les ondelettes et les fractales : Application aux images Météosat, Thèse de doctotat de l'Université de Mouloud Mammeri de Tizi-Ouzou, Spécialité Electronique, Option Télédétection, 128 p.

Lahdir, M., Abdat, S., Ameer, S., (2008) Application des algorithmes génétiques pour optimiser la compression, Colloque internationale sur l'optimisation et les systèmes d'information COSI08; Juin 08-10, 2008 ; Tizi-Ouzou, Algérie

Lahdir, M., Ameer, S. et Adane, A H. (2006) Algorithme non itératif, basé sur les ondelettes biorthogonales et les fractales, pour la compression d'images satellitaires. Télédétection, vol. 6, no. 4, p. 345-358.

Lahdir, M., Ameer, S. et **Akrouf, L.** (2006) Compression d'images par un codage fractal non itératif Implémenté dans le domaine DCT. 4ème Conférence Internationale JTEA 2006, 12-14 Mai 2006, Tunisie.

Lahdir, M., Ameer, S. et **Akrouf, L** (2005) Compression d'images numériques par fractal dans le domaine DCT. Internationale Conférence: Sciences of Electronic, Technologies of Information and Telecommunication SETIT2005, 27-31 Mars 2005, Sousse, Tunisie.

Lahdir, M., et Ameer, S. (2004) Compression d'images en sous bandes par transformation discrète en cosinus et quantification vectorielle. Internationale Conférence: Sciences of Electronic, Technologies of Information and Telecommunication SETIT2004, 15-20 Mars 2004, Sousse, Tunisie.

Lempel, A and Ziv, J. (1986) Compression of two-dimensional data, IEEE transactions on information theory, vol. 32, no. 1, p. 2-8.

Mallat, S.G. (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 7, p. 674-693.

Mallat, S.G. (2000) Une exploration des signaux en ondelettes. Les éditions de l'école polytechnique, 2000.

Marsault, X. (1995) Compression et cryptage des données multimédias. Hermes, Paris, 239 p.

Mercier, G., Roux, C., Martineau, G. (2003) Technologies du Multimédia, ENST Bretagne, Département ITI, Brest, France, 117 p.

Meyer, Y. (1992) Les ondelettes : Algorithmes et applications, Armand Collin.

Pearlman, W. A et B.J. Kim (1997). An Embedded Wavelet Video Coder Using Three Dimensional Set Partitioning in Hierarchical Trees (SPIHT). In IEEE Data Compression Conference DCC'97, p. 221-260.

Pointet, A. (2003) Compression des images de télédétection, Télédétection 2003, Vol 3, N° 2-3-4, p. 323 – 335.

Rodrigues, J. M. M. (2006) Transfert sécurisé d'images par combinaison de techniques de compression, cryptage et marquage. Thèse de doctorat, sciences et techniques du Languedoc, université Montpellier II, France, 143 p.

Sadoun, N., Lahdir, M. et Ameer, S. (2006) Compression d'images médicales par une approche hybride fractal quantification en cosinus sous bandes. 3 International Symposium on Image Video Communications over fixed and mobile networks ISIVC 06, 13-15 Septembre 2006, Yasmine Hammamet, Tunisie

Said, A. and Pearlman, W. A.(1996) A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. IEEE Trans. Circuits Syst. Video Technol., vol. 6, p. 243-250.

Shapiro, J. M. (1993) Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. On signal processing, vol. 41, no 12, p. 3445-3462.

Savaton, G. (1999) Compression réversible et irréversible d'images médicales à base d'ondelettes entières. Mémoire DEA. Université d'Angers.

Shannon, C. E. (1948), A mathematical theory of communication, Bell System Tech. J., vol. 27, p. 379-423 and p. 623-656.

Taubman, D. S. (2002) Subband transforms and the DWT, Signal Processing 2, University of New South Wales Sydney, Australia.

Taubman, D. and Marcellin, MW. (2002) JPEG2000 image compression: fundamentals, standards and practice. Dordrecht: Kluwer Academic Publishers.

Ziv, J. and Lempel, A. (1977) A universal algorithm for sequential data compression, IEEE transactions on information theory, vol. 23, no. 3, p 337-343.

## *Résumé :*

La question fondamentale abordée dans ce mémoire est celle de l'élaboration de méthodes de compression pour faciliter l'archivage d'images satellitaires de grande taille avec de forts taux de compression et le minimum de distorsions. Nous présentons une nouvelle méthode de compression que nous avons élaboré pour faciliter l'archivage d'images MSG avec de forts taux de compression et le minimum de distorsions. Notre méthode est basée sur les ondelettes entières biorthogonales en utilisant le schéma de lifting avec l'ondelette CDF9/7 et les codeurs à arbres de zéros (Zerotree). Ces codeurs sont EZW (Embedded Zerotree Wavelet coefficients) et SPIHT (Set Partitioning In Hierarchical Trees). L'algorithme ainsi développé a été appliquée à des images tests (Lena et Goldhill) et à une série d'images Météosat (MSG) prises dans les gammes du visible et de l'infrarouge au dessus de l'Afrique du Nord et du sud de l'Europe. Comparer aux méthodes se basant sur les ondelettes, JPEG2000 par exemple, notre méthode nous a permis d'atteindre des taux de compression de 97 % et un PSNR de 31,87 dB. Cette méthode a été employée pour archiver des images Météosat seconde génération (MSG) collectées toutes les quinze minutes en 2007. Après décompression, les images ont été reproduites fidèlement et l'espace mémoire requis pour leur stockage, a été réduit de plus de 90 %.

## *Mots-clés :*

Compression d'image, ondelettes entières, schéma de lifting, facteur de mérite, Zerotree, SPIHT, EZW, quantification à zones mortes (QZM), CDF9/7, Gall5/3, MSG, télédétection.