

**Titre:** Transformées orthogonales de l'analyse spectrale pour le filtrage et  
Title: la compression des images

**Auteur:** Salam Benchikh  
Author:

**Date:** 2011

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Benchikh, S. (2011). Transformées orthogonales de l'analyse spectrale pour le  
filtrage et la compression des images [Mémoire de maîtrise, École Polytechnique  
de Montréal]. PolyPublie. <https://publications.polymtl.ca/572/>  
Citation:

## Document en libre accès dans PolyPublie

Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/572/>  
PolyPublie URL:

**Directeurs de  
recherche:** Michael J. Corinthios  
Advisors:

**Programme:** génie électrique  
Program:

UNIVERSITÉ DE MONTRÉAL

**TRANSFORMÉES ORTHOGONALES DE L'ANALYSE SPECTRALE  
POUR LE FILTRAGE ET LA COMPRESSION DES IMAGES**

SALAM BENCHIKH

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES (M.Sc.A)  
(GÉNIE ÉLECTRIQUE)

Avril 2011

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

**TRANSFORMÉES ORTHOGONALES DE L'ANALYSE SPECTRALE  
POUR LE FILTRAGE ET LA COMPRESSION DES IMAGES**

Présenté par : BENCHIKH Salam

en vue de l'obtention du diplôme de : Maitrise ès Sciences Appliquées

a été dûment accepté par le jury d'examen constitué de :

M. LEMIRE Michel, M.Ing., président

M. CORINTHIOS Michael J., Ph.D., membre et directeur de recherche

M. CONAN Jean, Ph.D., membre

## DÉDICACE

*À mon père et ma mère*

## REMERCIEMENTS

Je tiens à remercier très chaleureusement mon directeur de recherche Professeur Michael CORINTHIOS qui a accepté de diriger mes travaux de recherche et qui a joué un rôle important dans ma formation académique.

Je tiens aussi à remercier très chaleureusement les membres du jury Professeur Michel Lemire et Professeur Jean Conan d'avoir accepté d'examiner mon mémoire et tous les professeurs de l'École Polytechnique qui ont contribué à ma formation académique.

Je remercie également mes collègues du laboratoire de traitement du signal et tous mes amis qui de près ou de loin m'ont donné leurs encouragements.

Je voudrais aussi exprimer ma gratitude à deux personnes les plus précieuses au monde mon père et ma mère pour leur support inconditionnel malgré la distance et à qui je dédie ce travail. Aussi je sais gré à tous les membres de ma famille et surtout Fatima.

## RÉSUMÉ

Ces dernières années ont vu une demande croissante pour la transmission et le stockage de haute qualité des signaux numériques, tels que la voix, la musique, l'image, la vidéo et les données pour différentes applications, y compris le HDTV, le iPod, le téléphone cellulaire, la vidéo-conférence, la caméra numérique, le stockage des images biomédicales, l'archivage, etc. Pour satisfaire ces besoins, il y a eu des études très riches et beaucoup d'efforts de recherche dans le domaine de la compression des données.

Il existe deux types de compression: *Compression sans perte* où la reconstruction des données est identique à l'original mais le taux de compression est très faible et la *compression avec perte* où le taux de compression est très élevé mais avec perte d'information. Nous focalisons notre attention en particulier sur la compression avec perte. L'utilisation des transformées est recommandée pour ce genre de compression.

Pour une application donnée, le choix d'une transformée en particulier dépend de la quantité d'erreur tolérée dans la compression et de la complexité de calcul. Le but est de trouver une transformée qui ferait concentrer l'énergie du signal dans le plus petit nombre de paramètres et qui n'exigerait pas une complexité excessive de calcul. Les algorithmes dits rapides de ces transformées permettent de réduire significativement le nombre d'opérations arithmétiques pour le calcul de la transformée directe et de son inverse.

L'objectif de ce travail est d'étudier des systèmes de compression d'images à base de transformées satisfaisant un bon compromis entre le taux de compression et le rapport signal sur bruit pour une utilisation dans la transmission ou l'archivage. Les techniques de transformées choisies dans notre mémoire pour la compression d'images sont la transformée discrète en cosinus (DCT) et la transformée discrète en ondelettes (DWT). Ainsi, nous analyserons tout d'abord la compression par DCT, puis la compression par DWT. Nous nous pencherons sur les critères de performance de l'une et de l'autre. D'un autre côté, nous étudierons l'impact de la distribution spectrale des images sur la qualité de compression. Comme la qualité de la compression d'images par DWT dépend du type d'ondelette utilisée, plusieurs familles d'ondelettes à support compact sont étudiées et évaluées. Finalement une évaluation numérique de la compression par DCT et DWT est présentée.

## ABSTRACT

There have been increasing demands for high-quality transmission and data storage, such as voice, music, image, video and data for different applications, including HDTV, iPod, cellular phones, video conferencing, digital camera, storage of biomedical images, data archiving, etc. To meet those needs, there have been very rich studies and several research efforts in the field of data compression. There are two types of compression: 1) *lossless compression* in which reconstruction of the data is identical to the original one, but the compression ratio is very low and 2) *lossy compression* where the compression ratio is very high, but with data loss. In particular, we focus our attention on the lossy compression. The use of transform techniques is recommended for this type of compression.

For a given application, the choice of a particular transform technique is highly dependent on the amount of error that can be tolerated in the compression and computational complexity. The goal is to find a transform that is able to concentrate the signal energy in the smallest numbers of parameters and which does not require excessive computational complexity. The fast algorithms of these transforms can significantly reduce the number of arithmetic operations to evaluate the forward transform and its inverse.

The objective of this research is to study image compression systems based on transform techniques for data transmission or archive while providing a good tradeoff between the compression rate and signal to noise ratio. The transform techniques chosen in this thesis for image compression include discrete cosine transform (DCT) and discrete wavelets transform (DWT). We first review and analyze the compression technique using DCT and then using DWT. We place an emphasis on the performance criteria for DCT and DWT. In addition, we study the impact of the spectral distribution of the images on the quality of each compression technique. Since the quality of image compression by DWT is highly dependent on the type of wavelets, several families of compactly supported wavelets are studied and evaluated. Numerical evaluations for compression by DCT and by DWT are performed.

## TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS .....	IV
RÉSUMÉ.....	V
ABSTRACT .....	VI
TABLE DES MATIÈRES .....	VII
LISTE DES TABLEAUX.....	XII
LISTE DES FIGURES.....	XIII
LISTE DES SIGLES ET ABRÉVIATIONS .....	XVII
CHAPITRE 1 INTRODUCTION.....	1
1.1 Développement d’algorithmes rapides pour les transformées .....	2
1.2 Comparaison des performances des transformées .....	5
1.3 Standard JPEG.....	5
1.4 La transformée discrète en ondelettes .....	6
1.5 Objectif de la recherche.....	8
1.6 Contribution .....	9
1.7 Organisation du mémoire .....	9
CHAPITRE 2 CONCEPTS DE BASE.....	11
2.1 Concept de base de la compression d’images .....	11
2.1.1 Introduction .....	11
2.1.2 Compression des données .....	11
2.1.3 Méthodes réversibles.....	12
2.1.4 Méthodes non réversibles.....	13
2.1.5 Méthodologie de la compression d’images par les transformées.....	14

2.2	Concept de base des filtres .....	16
2.2.1	Introduction .....	16
2.2.2	Fonctions de transfert des filtres .....	16
2.2.3	Gabarit des filtres .....	17
2.2.4	Fonctions d'approximation .....	18
2.2.5	Technologie des filtres .....	18
CHAPITRE 3 LA TRANSFORMÉE DISCRÈTE EN COSINUS (DCT) .....		21
3.1	Introduction .....	21
3.2	La Transformée discrète en cosinus monodimensionnelle .....	21
3.2.1	Transformée discrète en cosinus type I (DCT I) .....	22
3.2.2	Transformée discrète en cosinus type II (DCT II) .....	22
3.2.3	Transformée discrète en cosinus type III (DCT III) .....	23
3.2.4	Transformée discrète en cosinus type IV (DCT IV) .....	24
3.3	La Transformée discrète en cosinus bidimensionnelle .....	25
3.4	Propriétés de la DCT .....	25
3.4.1	Décorrélation .....	25
3.4.2	Concentration de l'énergie .....	27
3.4.3	Séparabilité .....	29
3.4.4	Symétrie .....	30
3.4.5	Orthogonalité .....	30
3.5	Algorithme rapide de la DCT .....	31
3.5.1	DCT implémentée par la FFT .....	31
3.5.2	DCT implémentée par méthode directe .....	35
3.6	La norme de compression JPEG .....	37

3.6.1	Encodage séquentiel JPEG .....	37
3.6.2	Équation DCT d'une image de taille $8 \times 8$ .....	38
3.6.3	Matrice de forme de DCT .....	38
3.6.4	Prétraitement .....	39
3.6.5	Application de la DCT sur les blocs de taille $8 \times 8$ .....	39
3.6.6	Quantification .....	40
3.6.7	Codage .....	41
3.6.8	Reconstruction de l'image .....	42
3.6.9	Exemple .....	43
CHAPITRE 4 LA TRANSFORMÉE DISCRÈTE EN ONDELETTES .....		47
4.1	Introduction .....	47
4.2	Concepts de base .....	48
4.2.1	Décimation et interpolation .....	48
4.2.2	Bancs de filtres .....	49
4.2.3	Filtres à reconstruction parfaite et filtres miroirs conjugués .....	50
4.2.4	Condition de reconstruction parfaite du signal .....	51
4.3	La transformée en ondelettes .....	54
4.3.1	La transformée continue en ondelettes .....	54
4.3.2	La transformée discrète en ondelettes .....	55
4.3.3	Algorithme Rapide de décomposition multi-résolution .....	58
4.3.4	Implémentation de la transformée discrète en ondelettes .....	61
4.3.5	DWT dans la compression d'images .....	68
CHAPITRE 5 ÉVALUATION D'EFFICACITÉ DE QUELQUES ONDELETTES À SUPPORT COMPACT .....		74
5.1	Introduction .....	74

5.2	Choix des images de test .....	74
5.3	Choix des ondelettes.....	77
5.3.1	Compression par les ondelettes biorthogonales .....	78
5.3.2	Compression par les ondelettes Daubechies .....	81
5.3.3	Compression par les ondelettes Symlets .....	84
5.3.4	Compression par les ondelettes Coiflets .....	88
5.4	Discussion .....	90
5.5	Conclusion.....	96
CHAPITRE 6 COMPARAISON DE SYSTÈMES DE COMPRESSION À BASE DE DWT, DCT ET LA MÉTHODE HYBRIDE .....		97
DWT-AMÉLIORÉ-DCT .....		97
6.1	Introduction .....	97
6.2	Structure du travail .....	97
6.2.1	Images de test.....	97
6.2.2	Prétraitement .....	97
6.2.3	Quantification.....	98
6.2.4	Seuillage .....	99
6.2.5	Codage.....	100
6.2.6	Évaluation.....	100
6.3	Compression par DCT.....	101
6.3.1	Diagramme bloc .....	101
6.3.2	Algorithme à base de DCT séquentiel.....	101
6.3.3	Matrice de quantification .....	103
6.3.4	Exemple de compression d'images .....	105
6.4	Compression à base de DWT .....	108

6.4.1	Diagramme bloc .....	108
6.4.2	Algorithme à base de DWT .....	108
6.4.3	Seuillage .....	110
6.4.4	Exemple de compression d'images .....	112
6.5	Comparaison de la compression DWT et DCT .....	117
6.6	Combinaison des algorithmes de compression de DWT et DCT .....	119
6.6.1	Diagramme bloc .....	119
6.6.2	Algorithme à base de DWT et DCT .....	120
6.7	Conclusion .....	123
	CHAPITRE 7 CONCLUSION .....	124
	BIBLIOGRAPHIE .....	126
	ANNEXES .....	131

**LISTE DES TABLEAUX**

Tableau 4. 1 : Répartition de la longueur des filtres dans la famille des ondelettes .....	70
Tableau 5. 1 : PSNR et TC de trois images compressées par des ondelettes biorthogonales .....	79
Tableau 5. 2 : PSNR et TC de trois images compressées par des ondelettes Daubechies .....	82
Tableau 5. 3 : PSNR et TC de trois images compressées par des ondelettes Symlet .....	85
Tableau 5. 4: PSNR et TC de trois images compressées par des ondelettes Coiflets .....	88
Tableau 5. 5 : Comparaison de la compression par les différentes ondelettes de l'image échographie avec la compression des images <i>Rachis-cervical</i> , <i>Rami</i> et <i>Lena</i> .....	95
Tableau 6. 1 : Évaluation de l'algorithme de compression DCT pour deux images.....	103
Tableau 6. 2 : Évaluation de l'algorithme de compression DWT pour deux images .....	110

## LISTE DES FIGURES

Figure 2. 1 : Diagramme bloc de la compression et la décompression par les transformées.....	14
Figure 3.1 : a) Image <i>rachis-cervical</i> , b) Auto-corrélation normalisée de l'image <i>rachis-cervical</i> , c) DCT de l'auto-corrélation normalisée de l'image <i>rachis-cervical</i> .....	26
Figure 3. 2: a) Image <i>wbarb</i> , b) Auto-corrélation normalisée de <i>wbarb</i> , .....	27
Figure 3.3: a) Image <i>wbarb</i> , b) DCT de l'image <i>wbarb</i> . .....	27
Figure 3.4: a) Image <i>bust</i> , b) DCT de l'image <i>bust</i> .....	28
Figure 3.5 : Image <i>rachis_scoliose</i> , b) DCT de l'image <i>rachis_scoliose</i> .....	28
Figure 3.6: Codeur et décodeur JPEG.....	38
Figure 3.7 : Codage en zigzag pour chaque bloc .....	42
Figure 4.1: Opération de décimation.....	48
Figure 4.2: Opération d'interpolation.....	48
Figure 4.3: Phase d'analyse des bancs de filtres .....	49
Figure 4.4 : Phase de synthèse des bancs de filtres .....	49
Figure 4.5 : Banc de filtres à deux canaux, phase d'analyse.....	50
Figure 4.6 : Banc de filtres à deux canaux, phase de synthèse .....	51
Figure 4. 7 : Opération de décimation d'un facteur de 2.....	52
Figure 4. 8 : Opération d'interpolation d'un facteur de 2 .....	52
Figure 4. 9 : Opération de décimation et d'interpolation .....	52
Figure 4. 10 : Phase d'analyse et de synthèse .....	53
Figure 4. 11 : Décomposition du signal $x(n)$ .....	58
Figure 4. 12: Schéma bloc de la décomposition par ondelettes en trois itérations .....	60

Figure 4. 13 : Phase d'analyse à deux itérations .....	61
Figure 4.14 : Phase de synthèse à deux itérations .....	62
Figure 4.15 : Processus d'analyse multi-résolution à deux dimensions pour une itération .....	63
Figure 4. 16 : Décomposition d'une image en trois itérations .....	65
Figure 4. 17 : Analyse à trois niveaux de l'image <i>wbarb</i> . .....	66
Figure 4. 18 : Processus de synthèse multi-résolution à deux dimensions pour une itération .....	67
Figure 4. 19 : Fonctions d'ondelettes avec leurs fonctions d'échelles correspondantes utilisées dans la phase d'analyse et la phase de reconstruction des ondelettes Haar, Db2, Sym3, Bior2.2 et Coif1.....	72
Figure 5. 1 : a) image <i>Rami</i> b) DCT de l'image <i>Rami</i> .....	75
Figure 5. 2: a) image <i>Lena</i> , b) DCT de l'image <i>Lena</i> .....	75
Figure 5. 3: Image <i>rachis_cervical</i> , b) DCT de l'image <i>rachis_cervical</i> .....	76
Figure 5. 4: a) image <i>Écographie</i> b) DCT de l'image <i>Écographie</i> .....	76
Figure 5. 5: PSNR de la compression de trois images par des ondelettes biorthogonales.....	80
Figure 5. 6 : TC de la compression de trois images par des ondelettes biorthogonales.....	81
Figure 5. 7 : PSNR de la compression de trois images par des ondelettes de Daubechies .....	83
Figure 5. 8 : TC de la compression de trois images par des ondelettes de Daubechies .....	84
Figure 5. 9: PSNR de la compression de trois images par des ondelettes de Symlets .....	86
Figure 5. 10: TC de la compression de trois images par des ondelettes de Symlets.....	87
Figure 5. 11: PSNR de la compression de trois images par des ondelettes de Coiflets .....	89
Figure 5. 12: TC de la compression de trois images par des ondelettes de Coiflets .....	90
Figure 5. 13 : Comparaison de la compression des trois images par les ondelettes biorthogonales .....	91
Figure 5. 14: Comparaison de la compression des trois images par les ondelettes Daubechies....	92

Figure 5. 15: Comparaison de la compression des trois images par les ondelettes symlets .....	93
Figure 5. 16: Comparaison de la compression des trois images par les ondelettes Coiflets .....	94
Figure 6. 1: Deux images compressées avec DCT en utilisant l'équation (6.1) .....	99
Figure 6. 2 : Diagramme bloc de la compression et décompression par DCT .....	101
Figure 6. 3 : Comparaison de deux images compressées par DCT .....	104
Figure 6. 4 : Image originale de <i>wbarb</i> .....	105
Figure 6. 5: Image <i>wbarb</i> compressée par DCT avec coefficient non nuls 28.12% .....	106
Figure 6. 6: Image <i>wbarb</i> compressée par DCT avec coefficient non nuls 15.96% .....	106
Figure 6. 7: Image <i>wbarb</i> compressée par DCT avec coefficients non nuls 6.46% .....	107
Figure 6. 8: Image compressée par DCT avec coefficient non nuls 4.84% .....	107
Figure 6. 9 : Bloc diagramme de la compression et décompression par DWT .....	108
Figure 6. 10: Compression par DWT pour deux images.....	111
Figure 6. 11: Image <i>wbarb</i> compressée par DWT avec coefficients non nuls 52.47% .....	112
Figure 6. 12: Image <i>wbarb</i> compressée par DWT avec coefficients non nuls 20.21% .....	113
Figure 6. 13: Image <i>wbarb</i> compressée par DWT avec coefficients non nuls 12.51% .....	113
Figure 6. 14: Image <i>wbarb</i> compressée par DWT avec coefficients non nuls 8.33% .....	114
Figure 6. 15: Schéma détaillé de la compression de l'image <i>Lena</i> par <i>DWT-améliorée</i> . .....	115
Figure 6. 16: Image <i>wbarb</i> compressée par <i>DWT-amélioré</i> avec coefficients non nuls 3.7% .....	115
Figure 6. 17: Image <i>wbarb</i> compressée par <i>DWT-amélioré</i> avec coefficients non nuls 10.05% .....	116
Figure 6. 18: Comparaison de la compression DWT avec la compression <i>DWT-amélioré</i> .....	117
Figure 6. 19 : Comparaison de la compression par DWT et DCT pour l'image bust.....	118
Figure 6. 20: Comparaison de la compression par <i>DWT-amélioré</i> , DWT et DCT pour l'image <i>wbarb</i> .....	119
Figure 6. 21: Diagramme bloc de la combinaison de la compression DCT et <i>DWT-améliorée</i> ..	120

Figure 6. 22: Schéma bloc de la compression de l'image *Lena* par *DWT-améliorée-DCT* ..... 122

Figure 6. 23: Comparaison de la compression *DWT-amélioré* et de la compression *DWT-amélioré-DCT*..... 123

## LISTE DES SIGLES ET ABRÉVIATIONS

Bior	Bi-orthogonal ondelettes
CD	Compact Disc
Coif	Ondelettes Coiflet
Db	Ondelettes Daubechies
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
DVD	Digital Video Disc
DWT	Discrete Wavelet Transform
DWHT	Discrete Walsh Hadamard Transform
EZW	Embedded Zéro-tree coding of Wavelet
FFT	Fast Fourier Transform
GWT	Generalized Walsh Transform
HDTV	High Definition Television
iPod	Interent Portable Open Database
JPEG	Joint Picture Expert Group
KLT	Karhunen-Loeve Transform
MDCT	Modified Discrete Cosine Transform
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
RLE	Run length encoding
SPIHT	Set Partitionning In Hierarchical Trees
Sym	Symlet ondelettes
TC	Taux de Compression

## CHAPITRE 1 INTRODUCTION

Le domaine du traitement numérique du signal [1], [2] est la base fondamentale de plusieurs domaines de recherche de grande actualité. Dans toutes les applications où l'information est sous forme numérique ou gérée par un processeur numérique, le traitement numérique du signal est présent. On peut donner comme exemples: le traitement d'image [3], [4], [5], la compression de données [6], le traitement des signaux biomédicaux [7], la reconnaissance de la voix [8], [9], la sécurité des communications [10], la technologie de la communication [11], [12], etc.

Le traitement numérique du signal est la représentation du signal par des séquences de nombres et le traitement de ces séquences [13]. Le traitement peut avoir des buts variés et différents, à savoir, élimination des interférences, estimations des paramètres d'un signal, transformation de la forme d'un signal, élimination de bruit, compression de données, codage, restauration, etc.

Les outils ou les techniques utilisés pour traiter ces types de problèmes sont nombreux, on peut citer l'analyse spectrale, la corrélation, la convolution, le filtrage, la transformation, etc [1], [2], [13]-[16]. Ces techniques sont utilisées dans pratiquement tous les domaines de la technologie car elles sont applicables dans toutes les étapes de la chaîne d'un système: acquisition, analyse, transfert et restitution.

Comme un signal numérique contient souvent beaucoup de corrélations entre ses paramètres et si on exploite les outils du traitement numérique du signal, on pourrait compresser les données du signal et minimiser l'information redondante [4], [6]. En conséquence, le coût de l'archivage des données et la largeur de la bande passante de transmission sont réduits.

Il existe deux types de compression [6]: *Compression sans perte* où la reconstruction des données est identique à l'original mais le taux de compression est très faible et la *compression avec perte* où le taux de compression est très élevé mais avec perte de données. Dans le but d'atteindre des taux de compression élevés nous focalisons notre attention en particulier sur la compression avec perte. L'utilisation des transformées est recommandée pour ce genre de compression [2], [4], [6] et du moment que la transformée inverse est importante pour la reconstruction rapide des données alors les transformées orthogonales sont utilisées [6], [15], [16]. On peut citer la transformée discrète de Walsh [1], [15], [16], la transformée discrète de Walsh-Hadamard [1], [15], [16], la

transformée de Walsh généralisée [1], [17], la transformée discrète en cosinus [1], [15], [16], la transformée discrète de Fourier [1], [2], [15], [16], la transformée discrète de Mellin [1], [16], la transformée discrète de Hartley [1], [16], la transformée discrète en ondelettes [1], [2], [18], etc. Toutes ces transformées sont unitaires, symétriques, réversibles et la somme de leurs énergies avant et après transformation reste inchangée [1], [2], [15], [16].

Dans [19] des architectures cellulaires pour le traitement et l'analyse spectrale des signaux et des images, ainsi que des unités universelles de traitement UPE (Universal Processing Element) sont présentées. L'architecture à traitement parallèle et le design des réseaux cellulaires sont construits en utilisant ces unités UPE. L'approche d'architectures construites en 3D produit des vitesses de traitement beaucoup plus élevées que dans le cas usuel de propagation des résultats en 2D.

Aussi, pour une application donnée, le choix d'une transformée en particulier dépend de la quantité d'erreurs de reconstruction tolérée et de la complexité de l'implémentation de la transformée [3], [4]. Le but est de trouver une transformée qui ferait concentrer l'énergie du signal dans le plus petit nombre de coefficients et qui n'exigerait pas une complexité excessive de calcul. Les algorithmes dits rapides de ces transformées permettent de réduire significativement le nombre d'opérations arithmétiques pour le calcul de la transformée directe et de son inverse [1]-[6], [15], [16].

Pour placer ce travail de recherche dans son contexte, nous commencerons par une présentation des principaux algorithmes rapides des transformées, des comparaisons de ces transformées, du standard JPEG et en particulier la transformée en ondelettes.

## 1.1 Développement d'algorithmes rapides pour les transformées

Une revue de littérature des principales transformées utilisées dans la compression des données est présentée avec citations.

### Transformée de Fourier:

La transformée la plus utilisée dans le domaine de l'analyse spectrale est la transformée discrète de Fourier (DFT). Le calcul de cette transformée pour une séquence de  $N$  points nécessite  $N^2$  opérations. L'algorithme le plus commun qui permet de réduire la complexité de calcul à  $N \log_2 N$  opérations est l'algorithme de Cooley-Tukey [20], [21] (aussi connu sous le nom de Fast Fourier

Transform, FFT). L'idée de base de cet algorithme est de diviser la transformée en deux parties de taille  $N/2$  à chaque itération, c'est la transformée rapide Radix 2. Dans [22] un algorithme plus efficace est présenté par Corinthios pour le calcul de la DFT avec une base Radix4. D'autres algorithmes ont été introduits pour le design des processeurs spécialisés de la transformée discrète de Fourier. On peut citer, l'algorithme de [23] où Corinthios est le premier à avoir proposé une factorisation de la matrice de DFT en introduisant la multiplication de Kronecker. Cet algorithme produit un processeur câblé sans adressage. Ce processeur effectue l'ordonnancement des données pendant la phase du feedback et ainsi élimine la nécessité d'itérations antérieures pour la post-permutation des coefficients de la transformée. Aussi, l'algorithme de Corinthios [24] où un haut niveau de parallélisme est introduit et une base radix généralisée est utilisée, fut en effet construit. Ce processeur à base 4 fut cité comme étant le standard de cette classe de technologie des machines FFT du plus grand parallélisme dans [25].

Aussi pour réduire le nombre de mémoires utilisé pour le calcul de la FFT, une méthode est présentée dans [26]. La transformée de Fourier est la transformée qui touche à tous les domaines d'applications sans exception.

### **Transformée discrète de Hartley:**

La transformée discrète de Hartley (DHT) se distingue de la transformée DFT par le fait que lorsqu'on applique la transformée DHT sur des entrées réelles, on obtient des sorties réelles, et ceci sans l'intervention d'aucun calcul de valeur complexe [1]. La transformée continue de Hartley était introduite par Hartley [27] et sa forme discrète était introduite par Bracewell [28], et c'est Bracewell aussi qui présenta son algorithme rapide [29]. Cet algorithme nécessite  $N \log_2 N$  opérations arithmétiques. On peut voir dans [29], [30], [31] le développement d'algorithmes pour le calcul rapide de la transformée Discrète de Hartley (DHT).

### **Transformée Discrète de Walsh:**

La transformée de Walsh se présente sous plusieurs formes suivant l'ordonnancement des lignes de la matrice [1]. On peut noter que la transformée de Fourier est un cas particulier de la transformée de Walsh généralisée et que la transformée de Walsh-Hadamard est la transformée de Walsh à base 2 [1]. Parmi plusieurs aspects de l'article [32] présenté par Corinthios, la proposition de l'implantation parallèle de la transformée de Walsh généralisée (GWT) est l'une de ses principales contributions. Cette transformée permet de fournir des matrices pour tous les

ordres et pour n'importe quelle base. Dans [17] Corinthios a proposé l'approche par la méthode de factorisation pour aboutir à une configuration ayant un parallélisme ‘en masse’. De plus elle permet d'introduire un niveau arbitraire de parallélisme, par l'incorporation de  $M = p^m$  processeurs à base générale  $p$ , où  $m$  peut varier entre 0 et  $n-1$ . Cette approche a été appliquée pour la transformée de Walsh généralisée dans les ordres Naturel, Walsh-Paley et Walsh-Kaczmarz [17].

La transformée DWHT est très utilisée dans les applications où la diminution de la quantité de calcul est très importante [6] car elle ne nécessite que des additions et des soustractions du fait que les coefficients de la transformation sont des +1 ou des -1 [1], [6], [15].

### **Transformée Karhunen-Loève:**

La transformée continue de Karhunen-Loève a été développée par Karhunen [33] et Loève [34]. Cette transformée est une série d'analyse spectrale pour la représentation d'un signal aléatoire donné [20], dont les fonctions de base orthogonales sont obtenues par les vecteurs propres de la matrice d'auto-corrélation correspondante. Malgré que cette transformée soit connue pour être la transformée la plus performante [1]-[6], [15], [16], elle n'est pas généralement utilisée dans les applications de traitement du signal et ceci pour deux raisons : la transformée KLT dépend des données du signal et ne possède pas d'algorithme rapide pour son calcul [20].

La transformée Karhunen-Loève est utilisée comme un benchmark pour l'évaluation des transformées [20] et aussi, dans les applications où les statistiques varient faiblement et la taille de la transformée est petite [6], [35], [36].

### **Transformée en Cosinus:**

La transformée discrète en cosinus est très proche de la DFT. Elle peut être obtenue en utilisant la DFT, en miroitant la séquence originale de  $N$  points pour obtenir une séquence de  $2N$  points [3], [15]. Les algorithmes rapides [37], [38], [39] permettent le calcul de la transformée DCT en utilisant la factorisation de la matrice DCT. Aussi, la transformée DCT peut être implémentée en utilisant des transformées autre que DCT. Dans [15], l'algorithme rapide de DCT est présenté en utilisant la FFT. Aussi dans [20] l'algorithme rapide de la transformée DCT est implémenté en utilisant la transformée DWHT.

Il y a huit variantes de la transformée discrète en cosinus, mais les plus utilisées sont quatre [37], types I, II, III et IV (plus d'explications sont données dans le chapitre 3). La DCT type II de taille 8 est devenue la transformée de choix pour les standards suivants : JPEG, MPEG-1, MPEG-2, H.261, H.263, et MPEG-4 [4]. Aussi, une version modifiée, basée sur la DCT type IV, appelée transformée en cosinus discrète modifiée (MDCT) est utilisée dans les formats de compression de son avec perte: AAC, MP3, etc [6].

## 1.2 Comparaison des performances des transformées

La comparaison de la performance des transformées peut se faire de différentes manières: le calcul de la capacité d'empaquetage de la transformée, le calcul de l'erreur de reconstruction quadratique moyenne ou bien le calcul de la complexité de calcul de la transformée [3].

La comparaison de DCT, DFT et KLT dans [41], [42] montre que le gain du codeur de la transformée (mesure de l'empaquetage de l'énergie) de DCT est très proche de KLT.

La comparaison de DCT, KLT, DFT, DWHT et la transformée discrète de Haar dans [20] montre que l'erreur de reconstruction quadratique moyenne de DCT est très proche de KLT.

Dans [6], [43] il est montré que KLT minimise l'erreur de reconstruction quadratique moyenne pour n'importe quelle image et pour n'importe quel nombre de coefficients retenus. Mais comme la transformée KLT ne possède pas d'algorithme rapide, elle ne peut être utilisée pour une application pratique.

La comparaison de la compression d'une image par DCT, DFT et DWHT dans [3] montre que DCT est supérieure aux autres transformées du point de vue empaquetage d'énergie et que la transformée DWHT est la transformée la plus simple à implémenter [3], [4].

## 1.3 Standard JPEG

Durant les années 1990, le standard industriel JPEG qui utilise la DCT est devenu l'outil indispensable pour l'archivage et la transmission des images compressées [4], [44]. Alors, beaucoup de techniques de codage ont été proposées pour améliorer l'efficacité de la compression [44].

Le standard JPEG de base se sert d'un codeur de Huffman ou d'un codeur à longueur variable [6], [44]. Comme le codeur à longueur variable ne peut représenter un symbole par un code inférieur à 1 bit, le codeur RLE (run length encoding) est introduit dans JPEG [44]. Aussi, le codeur arithmétique adaptatif connu sous le nom de codeur QM a été introduit pour améliorer l'efficacité du codeur [4], [44].

Avec l'apparition de la transformée discrète en ondelettes (TDO ou DWT) et l'apparition des codeurs EZW [45] (Embedded Zero-tree coding of Wavelet) et SPIHT (Set Partitionning In Hierarchical Trees) [46], les chercheurs ont multiplié les efforts pour améliorer les codeurs du JPEG. Crouse [47] présente un codeur qui permet d'avoir un meilleur rapport signal sur bruit (PSNR) que le codeur EZW par l'amélioration du design de la matrice de quantification. Aussi, [48], introduit le codeur d'image EZW pour le calcul des coefficients DCT.

Mais, comme l'application de la transformée en cosinus donne une localisation en fréquence, mais non spatiale alors le partitionnement des données dans le cas de signaux non stationnaires est nécessaire. Ceci entraîne l'effet de blocs dans les données reconstruites pour des taux de compression très élevés [6], [49].

La description de la technique de compression par DCT est bien détaillée par Wallace [50].

## 1.4 La transformée discrète en ondelettes

L'introduction de la transformée en ondelettes (DWT) avait pour but de faire face aux problèmes de nonstationnarité des signaux. Lorsqu'on applique la DWT à une image, la transformation est appliquée sur l'ensemble de l'image et non pas par partitionnement, en conséquence absence d'effet de blocs lors de la reconstruction des données [44], [51]. Ceci permet une répartition uniforme de l'erreur de compression sur toute l'image. Donc, au lieu de représenter un signal comme la somme pondérée de sinusoïdes comme c'est le cas de la transformée de Fourier, nous avons la somme pondérée d'ondes translatées et dilatées dans le temps [18], [51], [52]. La sélection de l'ondelette mère détermine la représentation du signal ou de l'image. Cette transformée est la base de la norme JPEG2000 [4], [44], [53].

L'utilisation de la transformée en ondelettes dans le domaine de la recherche revient en grande partie aux travaux de recherche de Yves Meyer, Ingrid Daubechies [54] et Stéphane Mallat [55].

En 1990, Daubechies montra la connexion entre les ondelettes et la transformée des sous-bandes présentée par Mallat en 1989. Depuis, l'effort des chercheurs s'est multiplié pour améliorer les performances de l'algorithme de codage des coefficients DWT. Une nouvelle génération de codeur est apparue avec le codeur EZW de Shapiro [45].

L'algorithme EZW exploite l'existence de zéro-corrélation entre les images de sous-bande. Mais dans la pratique, EZW nécessite beaucoup de niveaux pour atteindre une meilleure performance du codage et pour obtenir une haute résolution de l'image reconstruite, la complexité de l'algorithme devient encore plus grande [56]. La réduction de la complexité de EZW a été proposée par Egger et al [57]. Le développement le plus important de EZW a été présenté par Said et Pearlman, SPIHT [46]. Ce codeur est supérieur à EZW du point de vue vitesse et efficacité de compression [44], [56]. Taubman [58] présenta l'algorithme EBCOT, à conception différente de EZW et de SPIHT. Cette technique permet de réduire le besoin de mémoire dans l'implémentation hardware et software [44].

D'un autre coté, le choix de l'ondelette utilisée dans la transformation est primordial dans la performance du codage de la compression d'images [51], [59], [60]. L'ondelette de base la mieux adaptée pour le codage dépend des spécificités des images à coder [4], [51]. Les algorithmes à base de DWT, proposés pour la compression d'images, choisissent l'ondelette mère ou de base à partir d'une librairie ou bien en effectuant une comparaison sur quelques ondelettes [52]. Dans [51], quatre familles d'ondelettes ont été utilisées pour l'étude, ondelettes biorthogonales, ondelette de Haar, ondelettes de Coiflets et ondelettes de Daubechies. Dans [59], les ondelettes Daubechies, Coiflets et d'autres ondelettes sont utilisées pour l'évaluation mais pas les ondelettes biorthogonales, de Symlet ou de Haar. Dans [60], uniquement deux ondelettes ont été évaluées, l'ondelette Coif1 de la famille Coiflet et l'ondelette Db3 de la famille Daubechies. Dans la norme JPEG2000, deux ondelettes sont utilisées, ondelette Daubechies (9,7) pour la compression avec perte et ondelette LeGall(5,3) pour la compression sans perte [44].

D'autre part, dans la littérature, il est dit que la transformée en ondelettes permet d'atteindre de meilleurs taux de compression que la transformation en cosinus [4], [44], [20] mais l'implémentation de DCT est moins couteuse que DWT. Un algorithme DCT pour une image de taille 8x8 nécessite 54 multiplications [52]. L'algorithme de DWT dépend de la longueur du filtre de l'ondelette utilisée, soit au moins une multiplication par coefficient [61].

Ainsi, nous constatons les points suivants:

- Les transformées les plus recommandées pour la compression d'images sont DCT et DWT,
- Les éléments les plus importants dans un système de compression sont le quantificateur et le codeur entropique,
- Le choix des images de test est très important dans l'évaluation des systèmes de compression d'images,
- Le choix des ondelettes influence la performance du codage.

## 1.5 Objectif de la recherche

L'objectif de ce travail est d'étudier des systèmes de compression d'images à base de transformée et de satisfaire un bon compromis entre le taux de compression <sup>(1)</sup> et le rapport signal sur bruit pour une utilisation dans la transmission ou l'archivage. Alors, les techniques de transformées étudiées dans notre mémoire sont la transformée discrète en cosinus (DCT) et la transformée discrète en ondelettes (DWT) qui sont les plus recommandées pour la compression d'images [2], [4], [61], [62]. En conséquence, les critères de choix des ondelettes dans le cas de la transformée discrète en ondelettes et la matrice de quantification dans le cas de la transformée discrète en cosinus sont étudiés pour atteindre un bon taux de compression et un rapport signal sur bruit acceptable.

Nous avons considéré quatre familles d'ondelettes, ondelettes Daubechies, ondelettes Coiflets, ondelettes Symlets et ondelettes biorthogonales. Elles sont toutes des ondelettes à support compact, ce qui correspond à des filtres à réponse impulsionnelle finie (RIF). Après la présentation de l'évaluation et du choix des images de test, nous avons procédé à la comparaison des ces ondelettes au sein de chaque famille.

<sup>(1)</sup> Taux de compression utilisé dans notre étude est défini à la page78.

## 1.6 Contribution

La contribution de ce mémoire se résume essentiellement aux points suivants:

- Amélioration du taux de compression par la transformée discrète en cosinus. Plusieurs matrices de quantification avec facteur de qualité variable sont nécessaires pour la comparaison, d'où une formule a été modifiée et proposée pour les construire. Les résultats numériques présentent une nette amélioration du taux de compression pour le même PSNR (Peak Signal to Noise Ratio). La matrice utilisée comme référence est la matrice de luminance standard utilisée dans la norme JPEG.
- Évaluation de la compression par la transformée discrète en ondelettes en utilisant 26 ondelettes appartenant à quatre familles d'ondelettes; *Daubechies*, *Symlets*, *Coiflets* et *biorthogonales*. Les résultats numériques montrent l'influence de l'activité spectrale des images sur l'efficacité de compression.
- Évaluation et comparaison numérique et graphique de la compression d'images par la transformée discrète en cosinus (DCT) et la transformée discrète en ondelettes (DWT), et la proposition de la transformée nommée DWT-*amélioré*.
- Proposition d'une méthode hybride nommée DWT-*amélioré*-DCT basée sur la combinaison de deux techniques de transformées DCT et DWT-*amélioré* pour la compression d'images et la comparaison des résultats avec la compression par DCT et la compression par DWT.

## 1.7 Organisation du mémoire

Ce mémoire contient sept chapitres. Le chapitre 2 est un rappel des concepts de base du domaine de la compression des données et du filtrage. Dans le chapitre 3 nous nous penchons sur la technique de la transformée discrète en cosinus (DCT), en énumérant les propriétés les plus importantes. L'algorithme rapide de DCT à implémentation directe et indirecte est exposé ainsi qu'un exemple détaillé de la compression DCT. La transformée discrète en ondelettes est présentée dans le chapitre 4. Après, un rappel des concepts de base reliés à la transformée en

ondelettes où nous décrivons les bancs de filtres et les conditions sur les filtres pour avoir une reconstruction parfaite du signal, la transformée en ondelettes est introduite. Aussi l'algorithme rapide de la décomposition multi-résolution est présenté. Ce chapitre est finalisé par la description des principaux éléments à considérer lors de la conception d'un système de compression à base de DWT. Dans le chapitre 5, une évaluation d'efficacité de vingt-six ondelettes à support compact est présentée. Ces ondelettes appartiennent à quatres familles d'ondelettes. Chaque famille est évaluée avec trois images de test: on determine le rapport signal sur bruit (PSNR) en fonction de l'ordre du filtre, le taux de compression en fonction de l'ordre du filtre et le PSNR en fonction du taux de compression. Au chapitre 6, les résultats de compression de deux images par la transformée discrète en cosinus, par la transformée discrète en ondelettes et par une combinaison des deux transformées, sont présentés.

Le mémoire se termine par une conclusion générale au chapitre 7 qui résume notre travail ainsi qu'une vision des travaux futurs.

## CHAPITRE 2 CONCEPTS DE BASE

Ce chapitre est divisé en deux parties. La première partie où les concepts de base du domaine de la compression sont présentés, comporte une brève introduction suivie d'une présentation des différents types de compression ainsi qu'une description des méthodes de compression réversible et non réversible. Étant donné que le filtrage permet aussi la compression des données, la deuxième partie du chapitre expose les concepts de base des filtres.

### 2.1 Concept de base de la compression d'images

#### 2.1.1 Introduction

La compression d'images est la réduction d'un certain nombre des données de l'information utilisées pour représenter l'image. L'utilité de la compression des données surgit surtout dans la transmission et le stockage des données par la réduction de la bande passante et la réduction de la mémoire de stockage des ordinateurs. Le but est d'atteindre le meilleur taux de fidélité possible pour une capacité de stockage ou de transmission disponible. On trouve l'application de la compression d'image dans le broadcast, le téléphone cellulaire, l'iPod, les images biomédicales, la téléconférence, l'archivage documentaire qu'il soit administratif, éducatif ou autre, etc. Le besoin de l'accès à ces données nécessite des espaces de stockage énormes et leurs transmissions exigent une bande passante plus large. D'un autre côté et de nos jours, la puissance des processeurs augmente plus rapidement que les capacités de stockage ou les infrastructures de transmission. Pour pallier à ce problème, il est plus judicieux d'exploiter la puissance des microprocesseurs et de réduire la taille des données. En exploitant la corrélation des données pour minimiser la redondance de l'information, la compression des données permet de réduire le coup de l'archivage et la taille de la bande passante.

#### 2.1.2 Compression des données

Le terme de compression des données fait référence au processus de réduction de la quantité des données nécessaires pour la représentation de l'information. La base de cette compression est l'élimination des données redondantes [3].

## Types de compressions

La compression des données peut être classée de différentes manières:

- *Compression symétrique* est une méthode dont le processus de compression et celui de la décompression sont identiques. Par contre, si le processus utilisé dans la compression est différent de celui de la décompression, on parle d'une *compression asymétrique*. Cette dernière est utilisée dans l'archivage des données contrairement à la compression symétrique qui est généralement utilisée dans les transmissions.
- *Compression réversible* est une méthode dont la reconstruction des données de l'information est fidèle mais le taux de compression est faible. Lorsque les données de l'information peuvent tolérer un certain niveau de dégradation, on parle d'une *compression non réversible*, comme dans le cas de l'audio et de la vidéo où les capteurs sensoriels humains (l'œil et l'oreille) ne discernent pas certaines fréquences et la reconstruction de l'information est acceptable avec un taux de compression élevé.

### 2.1.3 Méthodes réversibles

Dans cette section, nous allons présenter quelques méthodes de compression réversible les plus utilisées dans la compression des données.

#### 2.1.3.1 Compression RLE (Run length encoding):

La compression RLE est basée sur la répétition des données consécutives. Au lieu de coder les symboles consécutifs de la chaîne, on code le nombre des symboles consécutifs de la même valeur [4], [6]. Cette méthode est utilisée depuis l'arrivée de la théorie de l'information [6].

#### 2.1.3.2 Codage Huffman

Le codage Huffman a été proposé par David Huffman en 1952 [63]. C'est une méthode statistique basée sur l'attribution d'un mot de code binaire pour chaque symbole de la chaîne à compresser. La longueur des mots du code des symboles est variable. Le symbole le plus fréquent a une longueur du mot de code la plus petite et ainsi de suite. Le codeur de Huffman est un arbre binaire ordonné par tous les symboles et par leurs fréquences d'apparition [3], [4], [6]. Les deux symboles les moins fréquents de la chaîne sont reliés par leurs 'Parents' en faisant la

somme de leurs fréquences. Les symboles prennent alors les valeurs 0 et 1. Le processus est répété sur l'ensemble des symboles jusqu'à ce qu'il ne reste qu'un seul symbole parent formant la racine de l'arbre. L'opération inverse est utilisée pour le décodage.

#### 2.1.3.3 Codage LZW

Le compresseur LZ77, utilisé pour l'archivage, et le compresseur LZ78 utilisé dans la compression d'images, ont été inventés par Abraham Lempel et Jacob Ziv en 1977 et 1978 respectivement [66]. Welch modifia le compresseur pour l'utiliser dans les contrôleurs de disques durs en 1984 [67]. Alors le codeur LZW est né. C'est un algorithme utilisé dans la compression et la décompression. Il est basé sur la multiplicité des occurrences de séquences de caractères dans la séquence à encoder. Son principe est de substituer des motifs par un code d'affectation en construisant au fur et à mesure un dictionnaire. Il est rapide en compression et en décompression et ne nécessite pas de virgule flottante [4], [6].

#### 2.1.3.4 Codage EZW

Le codeur EZW est le codeur proposé par Shapiro en 1993. Il permet la transmission des coefficients de la transformation dans l'ordre décroissant de leurs valeurs absolues. La caractéristique de cet algorithme est la transmission progressive de l'image. Il présente l'avantage de ne nécessiter ni phase d'apprentissage, ni dictionnaire, ni l'information sur l'image [45].

### 2.1.4 Méthodes non réversibles

Parmi les méthodes non réversibles nous avons des méthodes qui font appel à la technique des transformées. Beaucoup de techniques des transformées ont été proposées pour la compression du signal qu'il soit à une dimension ou multidimensionnel car la transformation permet de fournir des informations facilement compressibles.

La famille des transformées orthogonales est fortement recommandée pour ce domaine d'application du moment que la transformée inverse est nécessaire pour la reproduction du signal original et que l'inverse d'une matrice orthogonale est sa transposée [3]-[6]

La transformée en ondelettes et la transformée en cosinus sont deux exemples de la famille des transformées orthogonales. Plus de détail sur ces deux transformées est exposé dans les chapitres suivants.

La différence entre la méthode de compression réversible et la méthode de compression non réversible est la présence d'un quantificateur dans la chaîne du système de compression [65].

### 2.1.5 Méthodologie de la compression d'images par les transformées

La compression d'images par la méthode des transformées nécessite trois étapes de base. La première étape est la transformation des données de l'image pour obtenir des données moins corrélées. La deuxième étape est la quantification ou le seuillage, c'est dans cette étape que la perte d'information se produit. Finalement le codage, les données quantifiées sont réduites par codage pour fin de transmission ou d'archivage.

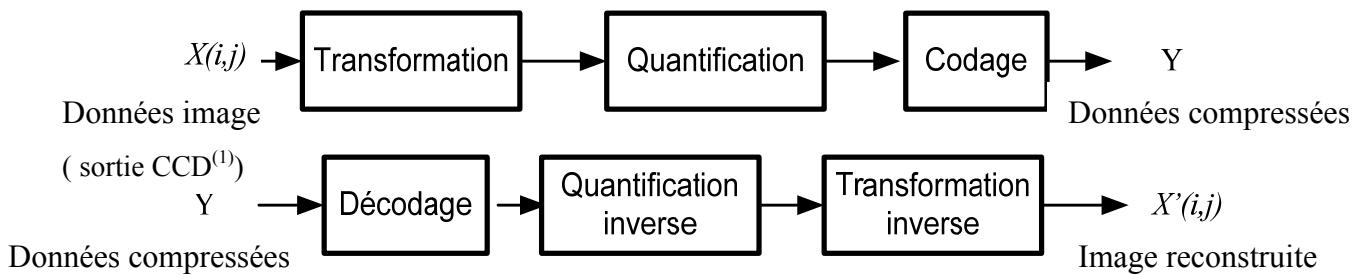


Figure 2. 1 : Diagramme bloc de la compression et la décompression par les transformées

Dans la phase de décompression ou de reconstruction de l'image, trois étapes sont nécessaires : les données transmises ou archivées sont décodées. La quantification inverse est appliquée au résultat du décodage. Puis la transformation inverse est appliquée aux données du bloc Quantification inverse comme présenté dans figure 2.1.

#### 2.1.5.1 Transformation

Le choix de la méthode de transformation doit permettre une réduction de la corrélation des données d'entrée. La sélection de la transformation est un facteur important dans la compression des données. Les transformées les plus connues dans le domaine de la compression sont: La transformée discrète de Fourier (DFT), la transformée discrète en cosinus (DCT), la transformée

<sup>(1)</sup> CCD : Charge-Coupled Device, un capteur photographique.

discrète de Hadamard (DHT), la transformée de Karhunen Loève (KLT), la transformée de Walsh généralisée (WGT), la transformée discrète en ondelettes (DWT). Aussi, DCT et DWT sont les plus utilisées dans la compression des données et en particulier dans la compression d'images [3], [4], [6], [53].

### 2.1.5.2 Quantification ou seuillage

*La quantification* permet de représenter numériquement chaque échantillon par un facteur de quantification (à partir d'une table) et *le seuillage* est l'élimination d'échantillons quand leur valeur est inférieure à la valeur du seuil [6].

Par la quantification et le seuillage, on introduit une dégradation de la qualité et une réduction de la précision. Cette dégradation est basée sur le choix du facteur de quantification et la valeur du seuil [3], [6]. Si la valeur du seuil est élevée, la dégradation sera considérable mais si la valeur du seuil est faible, alors la valeur de la dégradation est négligeable ou faible avec un taux de compression aussi faible.

Le choix du facteur de quantification ou du seuil doit tenir compte du système visuel humain pour atteindre une meilleure qualité de compression et un haut taux de compression [3], [6], [51]. À savoir, le système visuel humain est moins sensible aux fréquences hautes mais sensible aux fréquences basses.

Nous avons deux types de seuillage [59]:

**Hard thresholding ou seuillage dur** : si la valeur du pixel est inférieure à la valeur du seuil<sup>(1)</sup>, on remplacera la valeur du pixel par zéro sinon elle restera la même.

**Soft thresholding ou seuillage mou** : si la valeur du pixel est inférieure à la valeur du seuil, on remplacera la valeur du pixel par zéro, mais si elle est supérieure au seuil, on la divise par la valeur du seuil et le résultat remplacera la valeur originale du pixel.

### 2.1.5.3 Codage

Après quantification ou seuillage, la matrice obtenue est disponible pour le codage. Pour la transmission ou l'archivage, il est très avantageux de minimiser le nombre de bits du signal.

<sup>(1)</sup> Le choix du seuil pour la compression DWT est donné à la page 108.

Les méthodes utilisées sont les méthodes de compression réversibles. On peut citer comme exemple : le codage Huffman, le codage EZW etc.

## 2.2 Concept de base des filtres

Comme mentionné ci-dessus, le système auditif et le système oculaire ne peuvent discerner certaines fréquences et l'élimination de ces dernières produira à la reconstruction de l'information un produit similaire à l'original. L'élimination d'une bande de fréquence ou de certaines fréquences est une opération de filtrage. Les filtres sont utilisés dans la compression des données de l'information et font partie des méthodes de compression irréversible. Dans ce qui suit nous allons présenter les notions de base des filtres.

### 2.2.1 Introduction

Un filtre est un circuit électronique permettant l'atténuation de certaines composantes du signal et laissant passer d'autres. Cette modification est effectuée dans les deux domaines : temporel et fréquentiel. Ainsi, le filtre permet d'apporter une modification de la phase et ou de l'amplitude des composantes du signal considéré [1].

Les filtres sont principalement utilisés pour éliminer le bruit additionné au signal utile. Cette opération est appelée filtrage, c'est une opération très commune en traitement du signal. Le but du filtrage est de [1]:

- modifier les composantes spectrales d'un signal,
- retarder un signal,
- isoler une information utile contenue dans une bande passante déterminée par rapport aux bruits et aux autres informations existantes hors de cette bande.

### 2.2.2 Fonctions de transfert des filtres

Un filtre analogique est caractérisé par sa fonction de transfert dans le plan de Laplace.

$$H(j\Omega) = H(s)_{s \rightarrow j\Omega} = \frac{\sum_{i=0}^M a_i s^i}{\sum_{i=0}^N b_i s^i} \quad \text{avec } M \leq N$$

Avec,  $a_i$  les coefficients du signal de sortie et  $b_i$  les coefficients du signal d'entrée.

La fonction de transfert caractérise complètement un système linéaire. En effet, la fonction de transfert n'a pas de sens pour un système non linéaire. De plus, elle est indépendante de l'entrée du système.

$$H(j\Omega) = |H(j\Omega)|e^{j\angle H(j\Omega)}$$

Où  $|H(j\Omega)|$  est le gain et  $\angle H(j\Omega)$  est le déphasage.

La représentation graphique de  $H(j\Omega)$  se fait par le diagramme de Bode.

Caractéristiques:

- La fonction de transfert d'un système réalisable a le degré du dénominateur supérieur au degré du numérateur,
- L'ordre d'un filtre est donné par le degré du dénominateur de la fonction de transfert,
- Un filtre causal est stable si tous ses pôles sont à parties réelles négatives,
- Les fonctions de transfert peuvent être décomposées comme le produit de fonctions de transfert du premier et du second ordre.

### 2.2.3 Gabarit des filtres

Les filtres analogiques présentent les imperfections suivantes [1]:

- Transition progressive entre la bande passante et la bande coupée,
- Irrégularité du gain dans la bande passante et la bande coupée (ondulation),
- Affaiblissement dans la bande coupée,
- Irrégularité du temps de propagation.

Avec le gabarit du filtre, on peut fixer les limites de la fonction de transfert du filtre à réaliser et ainsi faire un compromis sur ces différentes imperfections.

Il suffit que la courbe de la fonction de transfert passe par les limites du gabarit du filtre. Donc pour chaque application un gabarit est défini.

## 2.2.4 Fonctions d'approximation

Pour la réalisation d'un filtre, on utilise les fonctions d'approximation. Les paramètres des valeurs de composantes des filtres passifs et des valeurs de la fonction de transfert des filtres actifs, sont tabulés à l'avance pour des prototypes passe-bas normalisés.

Les fonctions d'approximation les plus utilisées sont [1]:

- BUTTERWORTH
- CHEBYCHEV
- BESSEL ou THOMSON
- CAUER ou ELLIPTIQUE

Le choix de la fonction d'approximation dépend des valeurs des coefficients de l'équation caractéristique ou des pôles du système. Le choix de la fonction d'approximation va influencer la réponse en régime transitoire.

Les fonctions modèles utilisées pour la synthèse des filtres sont soit la réponse impulsionnelle soit la réponse fréquentielle de filtre analogique connu.

## 2.2.5 Technologie des filtres

On distingue essentiellement trois types de technologies des filtres. Seul le filtre numérique sera considéré dans la suite.

### 2.2.5.1 Filtre passif

Les filtres passifs sont réalisés principalement de composants passifs (résistances, condensateurs, bobines couplées ou non). Par conséquent, leur gain (rapport de puissance entre la sortie et l'entrée) ne peut excéder 1. Autrement dit, ils ne peuvent qu'atténuer en partie des signaux, mais pas les amplifier car cela nécessiterait un apport d'énergie.

### 2.2.5.2 Filtre actif

Les filtres actifs sont réalisés par l'usage d'au moins un composant actif (par exemple transistor, amplificateur opérationnel, ou autre circuit intégré, ...). Le gain de ces filtres peut être supérieur à 1 donc ils peuvent amplifier. Mais, contrairement aux filtres passifs, ils nécessitent une alimentation.

### 2.2.5.3 Filtre numérique

Un filtre numérique se caractérise par le traitement entièrement numérique du signal. Au préalable, le signal analogique est filtré par un filtre anti-repliement puis il est numérisé par un convertisseur analogique-numérique qui généralement intègre un échantillonneur-bloqueur, c'est-à-dire qu'à intervalles réguliers (séparés par la période d'échantillonnage), l'amplitude instantanée du signal est échantillonnée et maintenue puis quantifiée [1].

Le filtre numérique est caractérisé par l'équation suivante:

$$y[n] = \sum_{i=0}^M b_i x[n-i] + \sum_{j=1}^N a_j y[n-j] \quad (2.1)$$

Si les coefficients  $a_j$  sont nuls alors :

$$y[n] = \sum_{i=0}^M b_i x[n-i] \quad (2.2)$$

C'est l'équation des filtres à réponse impulsionnelle finie FIR ou filtre non récursif. L'échantillon  $y[n]$  ne dépend que d'un nombre fini d'échantillons. Sa fonction de transfert est :

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^M b_i z^{-i} = \sum_{n=0}^M h[n] z^{-i} \quad (2.3)$$

On remarque que  $h[n]$  est égal à  $b_n$  et que  $h[n]$  est égal à zéro pour  $n > M$  .

Les filtres FIR sont toujours stables, la sortie ne dépend pas de la valeur précédente.

Si les coefficients  $a_j$  sont non nuls alors :

$$y[n] = \sum_{i=0}^M b_i x[n-i] + \sum_{j=1}^N a_j y[n-j] \quad (2.4)$$

La transformée en  $Z$  de  $y[n]$  est donnée comme suit :

$$Y(z) = \sum_{n=0}^{\infty} y[n] z^{-n} = \sum_{n=0}^{\infty} \sum_{i=0}^M (b_i x[n-i]) z^{-n} + \sum_{n=0}^{\infty} \sum_{j=1}^N (a_j y[n-j]) z^{-n} \quad (2.5)$$

$$Y(z) = \sum_{i=0}^M b_i \sum_{n=0}^{\infty} x[n-i] z^{-n} + \sum_{j=1}^N a_j \sum_{n=0}^{\infty} y[n-j] z^{-n} \quad (2.6)$$

D'où

$$Y(z) = X(z) \sum_{i=0}^M b_i z^{-i} + Y(z) \sum_{j=1}^N a_j z^{-j} \quad (2.7)$$

Alors:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^M b_i z^{-i}}{1 - \sum_{j=1}^N a_j z^{-j}} \quad (2.8)$$

La réponse impulsionnelle de ces filtres est infinie. Cette équation est la réponse impulsionnelle des filtres RII ou filtres récursifs.

Un système numérique est stable si le module des pôles dans le domaine complexe est inférieur à l'unité. De même, un système analogique est stable si les parties réelles des pôles dans le domaine de Laplace sont inférieures à zéro.

## CHAPITRE 3 LA TRANSFORMÉE DISCRÈTE EN COSINUS (DCT)

La transformée discrète en cosinus a prouvé son utilité dans de nombreux domaines du traitement numérique du signal et de l'image [3]-[7], [20], [44], [53]. En ce qui suit, nous allons présenter les formes directes et inverses de la transformée de DCT les plus utilisées ainsi que quelques propriétés de la DCT. Nous exposons les deux types d'algorithme rapide de DCT : Algorithmes rapides implémentés par une méthode de factorisation et algorithmes rapides implémentés par d'autres transformées. On finalise ce chapitre en introduisant la norme JPEG et ses différentes étapes pour la compression d'une image avec un exemple pour visualiser la qualité de la compression.

### 3.1 Introduction

La transformée discrète en cosinus est l'une des techniques permettant la conversion du signal temporel en composantes fréquentielles. Cette technique est largement utilisée dans la compression des données et de l'image. Les standards, JPEG de Joint Picture Expert Group, MPEG de Moving Picture Expert group, H.261, H.263 et H.264 de l'Union International de Télécommunication utilisent la technique de base DCT qui est très proche de la DFT [4]

La transformation DCT est la projection de l'image sur une base de fonctions cosinus. L'intérêt de cette méthode réside dans le fait qu'uniquement les hautes fréquences spatiales, auxquelles le système visuel humain est le moins sensible, sont comprises dans les coefficients d'ordre supérieur. L'élimination des coefficients à partir d'un certain ordre revient alors à un filtrage passe-bas. Plus le nombre de coefficients retenus sera faible, plus la compression sera importante, et moins de détails seront conservés.

### 3.2 La Transformée discrète en cosinus monodimensionnelle

Il y a huit variantes de la transformée discrète en cosinus, mais les plus utilisées sont quatre. La plus courante parmi ces quatre est la DCT type II et est appelée DCT et son inverse est la DCT type III et est appelée IDCT [20]. Nous allons présenter ces quatre variantes DCT I, DCT II, DCT III et DCT IV dans ce qui suit.

### 3.2.1 Transformée discrète en cosinus type I (DCT I)

Les formules des transformations directe et inverse de DCT I sont présentées ci-dessous :

Transformation directe:

$$X_c[k] = \sqrt{\frac{2}{N}} \alpha_k \sum_{n=0}^N \alpha_n x[n] \cos \left[ \frac{\pi n k}{N} \right] \quad k = 0, \dots, N \quad (3.1)$$

Transformée inverse:

$$x[n] = \sqrt{\frac{2}{N}} \alpha_n \sum_{k=0}^N \alpha_k X_c[k] \cos \left[ \frac{\pi n k}{N} \right] \quad n = 0, \dots, N \quad (3.2)$$

Avec,  $\alpha_k$  constante définie comme suit :

$$\alpha_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } k = 0 \text{ ou } N \\ 1 & \text{autrement} \end{cases} \quad (3.3)$$

Cette forme de DCT I est définie pour les valeurs  $N$  entières supérieures à 2.

La forme orthonormée matricielle de la transformation directe DCT type I peut être exprimée comme suit

$$[C_{N+1}^I]_{k,n} = \sqrt{\frac{2}{N}} \left[ \alpha_k \alpha_n \cos \frac{\pi k n}{N} \right] \quad k, n = 0, 1, \dots, N-1 \quad (3.4)$$

et celle de sa transformée inverse peut être exprimée comme  $[C_{N+1}^I]^{-1} = [C_{N+1}^I]^T = C_{N+1}^I$ , où  $[.]^{-1}$  représente la matrice inverse et  $[.]^T$  représente la matrice transposée.

### 3.2.2 Transformée discrète en cosinus type II (DCT II)

Les formules des transformations directe et inverse de DCT II sont présentées ci-dessous :

Transformation directe:

$$X_c[k] = \sqrt{\frac{2}{N}} \alpha_k \sum_{n=0}^{N-1} x[n] \cos \left[ \frac{\pi(2n+1)k}{2N} \right] \quad k = 0, \dots, N-1 \quad (3.5)$$

Transformée inverse :

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha_k X_c[k] \cos \left[ \frac{\pi(2n+1)k}{2N} \right] \quad n = 0, \dots, N-1 \quad (3.6)$$

Cette forme est définie pour toutes les valeurs  $N$  positives. Elle est généralement désignée par DCT c'est la forme la plus utilisée.

La forme orthonormée matricielle de la transformation directe DCT type II peut être exprimée comme suit,

$$[C_N^{II}]_{k,n} = \sqrt{\frac{2}{N}} \left[ \alpha_k \cos \frac{\pi k(2n+1)}{2N} \right] \quad k, n = 0, 1, \dots, N-1 \quad (3.7)$$

Et celle de sa transformée inverse peut être exprimée comme  $[C_N^{II}]^{-1} = [C_N^{II}]^T = C_N^{III}$ , où  $[.]^{-1}$  représente la matrice inverse et  $[.]^T$  représente la matrice transposée.

### 3.2.3 Transformée discrète en cosinus type III (DCT III)

Les formules des transformations directe et inverse de DCT III sont présentées ci-dessous :

Transformation directe:

$$X_c[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \alpha_n x[n] \cos \left[ \frac{\pi n(2k+1)}{2N} \right] \quad k = 0, \dots, N-1 \quad (3.8)$$

Transformée inverse:

$$x[n] = \sqrt{\frac{2}{N}} \alpha_n \sum_{k=0}^{N-1} X_c[k] \cos \left[ \frac{\pi n(2k+1)}{2N} \right] \quad n = 0, \dots, N-1 \quad (3.9)$$

Cette forme est définie pour toutes les valeurs de  $N$  positives.

La forme orthonormée matricielle de la transformation directe DCT type III peut être exprimée comme suit,

$$[C_N^{III}]_{k,n} = \sqrt{\frac{2}{N}} \left[ \alpha_n \cos \frac{\pi k(2n+1)}{2N} \right] \quad k, n = 0, 1, \dots, N-1 \quad (3.10)$$

et celle de sa transformée inverse peut être exprimée comme  $[C_N^{III}]^{-1} = [C_N^{III}]^T = C_N^{II}$ .

### 3.2.4 Transformée discrète en cosinus type IV (DCT IV)

Les formules des transformations directe et inverse sont présentées ci-dessous :

Transformation directe:

$$X_c[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x[n] \cos \left[ \frac{\pi(2n+1)(2k+1)}{4N} \right] \quad k = 0, \dots, N-1 \quad (3.11)$$

Transformée inverse:

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X_c[k] \cos \left[ \frac{\pi(2n+1)(2k+1)}{4N} \right] \quad n = 0, \dots, N-1 \quad (3.12)$$

La forme orthonormée matricielle de la transformation directe DCT type IV peut être exprimée comme suit,

$$C_N^{IV} = \sqrt{\frac{2}{N}} \left[ \cos \frac{\pi(2n+1)(2k+1)}{4N} \right] \quad k, n = 0, 1, \dots, N-1 \quad (3.13)$$

et celle de sa transformée inverse peut être exprimée comme  $[C_N^{IV}]^{-1} = [C_N^{IV}]^T = C_N^{IV}$

L'analyse de ces formes montre les points suivants:

- Les transformées en cosinus de type I et de type IV sont représentables par des matrices involutives dont les inverses sont des matrices de la transformation directe elle-même,
- La transformée inverse de type II est similaire à la transformée directe de type III.

La DCT II est très utilisée en traitement du signal et de l'image et spécialement en compression [20], [50].

**Remarque :** Si la séquence du signal est supérieure à  $N$ , il est possible de diviser le signal en sous-séquences et ainsi la DCT peut être appliquée indépendamment sur toutes les séquences. On note que la valeur du point de la fonction de base ne change pas mais seulement la valeur de  $x(n)$  dans chaque sous-séquence. Donc, il est souhaitable de calculer les fonctions de base qui se répètent et les faire multiplier par les sous-séquences. Ceci permet la réduction des opérations mathématiques [68].

### 3.3 La Transformée discrète en cosinus bidimensionnelle

La transformée discrète bidimensionnelle est une extension directe de la DCT monodimensionnelle. Pour la transformation directe bidimensionnelle, la DCT II est utilisée et pour la transformation inverse bidimensionnelle, la DCT III est utilisée. Elles sont données comme suit :

Transformation directe:

$$X_c[k_1, k_2] = \frac{2}{\sqrt{N_1 N_2}} \alpha_{k_1} \alpha_{k_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x[n_1, n_2] \cos \left[ \frac{\pi(2n_1 + 1)k_1}{2N_1} \right] \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N_2} \right] \quad (3.14)$$

Transformation inverse :

$$x[n_1, n_2] = \frac{2}{\sqrt{N_1 N_2}} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \alpha_{k_1} \alpha_{k_2} X_c[k_1, k_2] \cos \left[ \frac{\pi(2n_1 + 1)k_1}{2N_1} \right] \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N_2} \right] \quad (3.15)$$

Avec,  $(n_1, n_2) = 0, 1, \dots, N-1$ ,  $(k_1, k_2) = 0, 1, \dots, N-1$  et  $\alpha_p = \begin{cases} \frac{1}{\sqrt{2}} & p = 0 \\ 1 & \text{autrement.} \end{cases}$

### 3.4 Propriétés de la DCT

Dans cette section nous soulignons quelques propriétés de la transformée discrète en cosinus et en particulier celles qui sont importantes pour la compression des signaux bidimensionnels.

#### 3.4.1 Décorrélation

La DCT-II bidimensionnelle est la transformée de décorrélation utilisée dans le standard de codage international de l'image et de la vidéo [20]. Le but de la compression est de minimiser la redondance de l'information ceci revient à minimiser la corrélation des coefficients du signal. La corrélation d'une image dénotée  $R_{ff}(\cdot, \cdot)$  est définie par:

$$R_{ff}(k_1, k_2) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) f(k_1 + m, k_2 + n) \quad (3.16)$$

où,  $0 \leq k_1 < 2(M-1)$ ,  $0 \leq k_2 < 2(N-1)$  et  $(\cdot)^*$  représente le conjugué de la fonction.

Pour visualiser l'effet de la décorrélation sur une image, nous allons prendre comme exemple deux images et on calcule l'auto-corrélation de ces images, puis on applique la transformation DCT sur le résultat de l'auto-corrélation normalisée de chaque image.

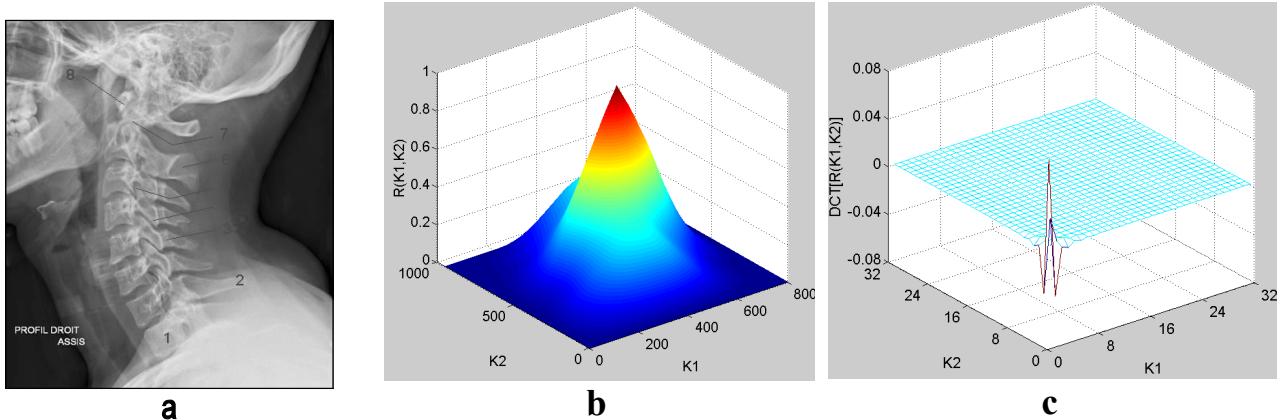


Figure 3.1 : a) Image *rachis-cervical*, b) Auto-corrélation normalisée de l'image *rachis-cervical*, c) DCT de l'auto-corrélation normalisée de l'image *rachis-cervical*.

La figure 3.1.b représente l'auto-correlation normalisée de l'image *rachis-cervical* et la figure 3.1.c représente la DCT de la figure 3.1.b.

Aussi, la figure 3.2.b représente l'auto-correlation normalisée de l'image *wbarb* et la figure 3.2.c représente la DCT de la figure 3.2.b.

Il faut noter que les données de l'image sont représentées par une matrice et que toute opération de traitement est appliquée sur les éléments de cette matrice. Aussi, l'application de la transformation DCT sur une image produit une matrice dont les éléments sont appelés coefficients DCT. Dans ce qui suit nous allons utiliser ces notions.

Alors, comme montré dans figure3.1 et figure3.2, en normalisant les éléments de la matrice de l'autocorrelation des images, on remarque que la DCT permet de réduire l'amplitude de ces coefficients et aussi de les réorganiser de telles sortes que les premiers coefficients de DCT soient à haute amplitude et les derniers coefficients soient de très faible amplitude, très proche de zéro. figure3.1.c et figure3.2.c montrent bien que la transformation DCT permet une excellente décorrélation.

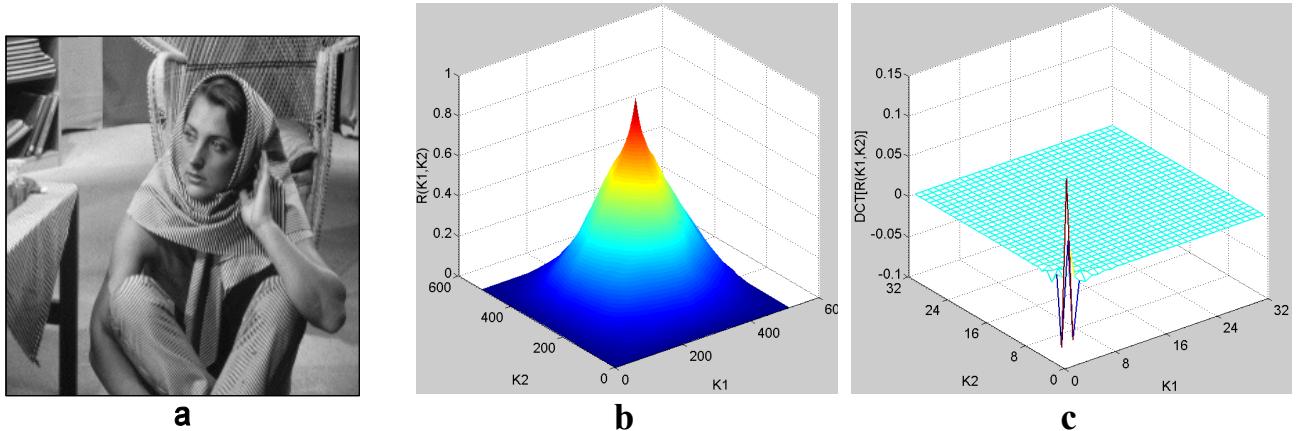


Figure 3. 2: a) Image *wbarb*, b) Auto-corrélation normalisée de *wbarb*,  
c) DCT de l'auto-corrélation normalisée de l'image *wbarb*

### 3.4.2 Concentration de l'énergie

L'efficacité d'une transformée est mesurée par sa capacité de concentrer les données d'entrée dans le plus petit ensemble de coefficients possible. La DCT est capable de concentrer l'énergie du signal dans les premiers coefficients DCT donc dans les basses fréquences. Ceci permet au quantificateur d'éliminer les coefficients de haute fréquence et à faible amplitude sans ou en introduisant de très faible distorsion lors de la reconstruction de l'image non visible à l'œil nu (en utilisant un simple filtrage passe-bas).

Les figures ci-dessous montrent la concentration de l'énergie pour différentes images :

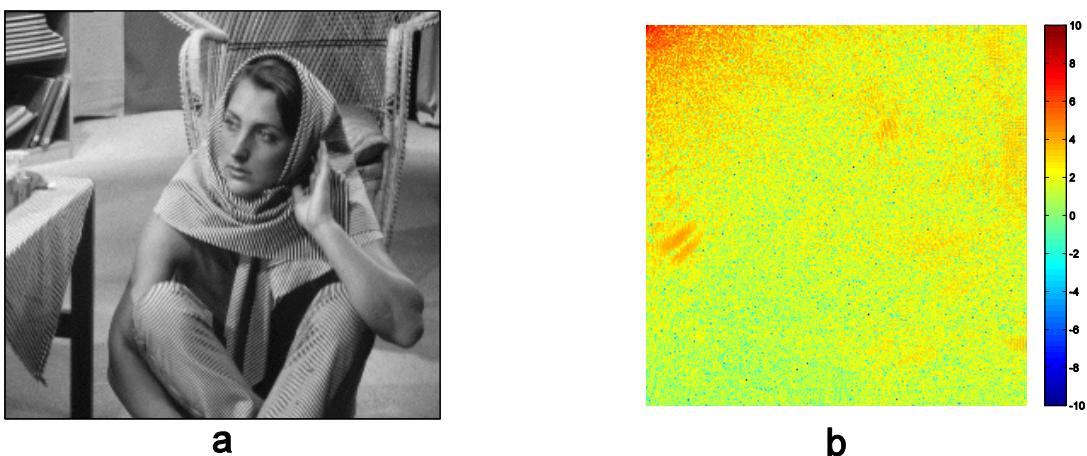


Figure 3.3: a) Image *wbarb*, b) DCT de l'image *wbarb*.

Les coefficients DCT à haute amplitude sont représentés par des points rouges, les coefficients DCT à très faible amplitude sont représentés par des points verts et les coefficients DCT à haute amplitude mais de signe négative sont représentés par des points bleus.

Dans la figure 3.3.b, la variation de l'intensité est assez importante sur toute la surface et le détail spatial est plus important que dans l'image *bust*. L'énergie de l'image est concentrée principalement dans les basses fréquences et peu dans les hautes fréquences.

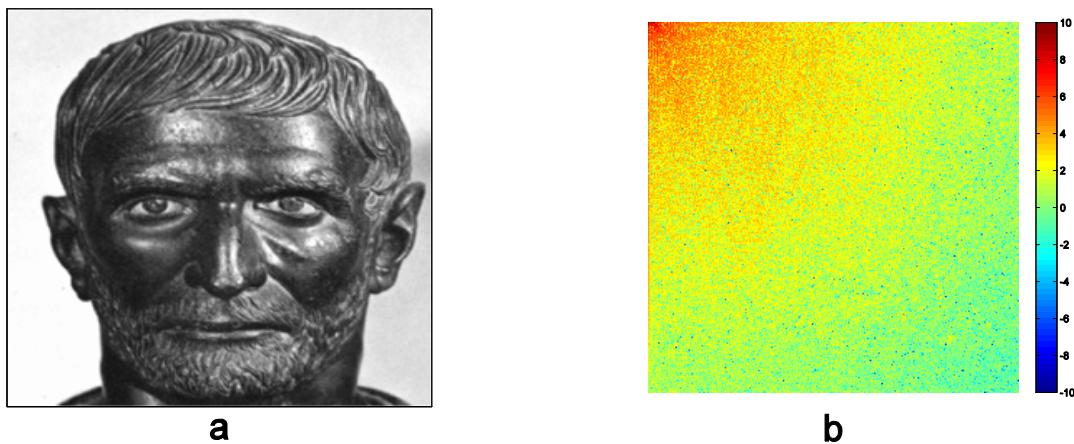


Figure 3.4: a) Image *bust*, b) DCT de l'image *bust*

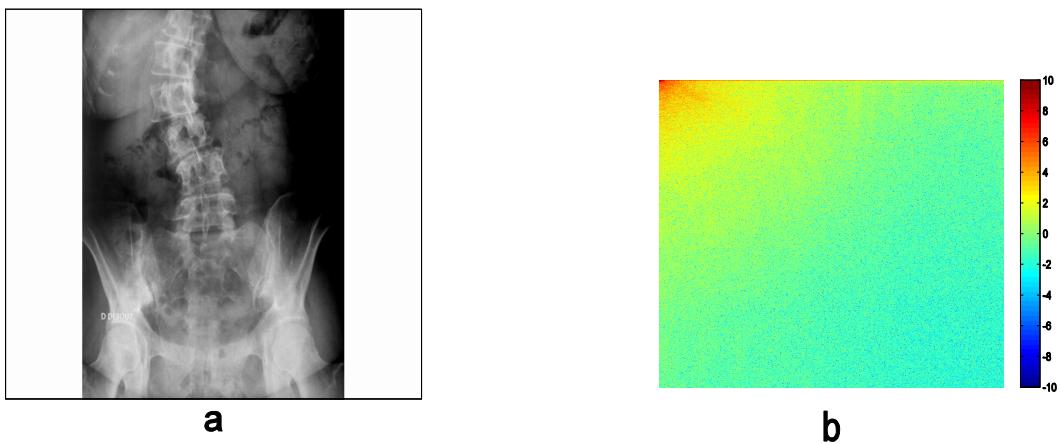


Figure 3.5 : Image *rachis\_scoliose*, b) DCT de l'image *rachis\_scoliose*

On voit bien que l'énergie de l'image *rachis\_scoliose* (figure 3.5.b) est concentrée en haut de l'image sur le côté gauche. Nous avons une variation faible de l'intensité sur une large surface et le détail spatial est assez faible. Aussi, dans l'image *bust* (figure 3.4.b) la variation de l'intensité est assez importante sur plus de la moitié de la surface de l'image, de plus le détail spatial est assez faible. L'énergie de l'image *bust* est concentrée dans les faibles fréquences et peu dans les hautes fréquences. L'énergie de l'image *Rachis\_scoliose* est concentrée dans les faibles fréquences.

### 3.4.3 Séparabilité

Une transformée est séparable si les conditions suivantes sont satisfaites:

$$X(k_1, k_2, n_1, n_2) = X_1(k_1, n_1)X_2(k_2, n_2) \quad (3.17)$$

et

$$x(k_1, k_2, n_1, n_2) = x_1(k_1, n_1)x_2(k_2, n_2) \quad (3.18)$$

Avec  $x$  étant la transformée inverse de  $X$  et  $x_I$  étant la transformée inverse de  $X_I$ .

Aussi, l'équation (3.14) de la transformée discrète en cosinus peut aussi s'écrire de la façon suivante:

$$X[k_1, k_2] = \sqrt{\frac{2}{N_1 N_2}} \alpha_{k_1} \alpha_{k_2} \sum_{n_1=0}^{N_1-1} \cos \left[ \frac{\pi(2n_1 + 1)k_1}{2N_1} \right] \sum_{n_2=0}^{N_2-1} x[n_1, n_2] \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N_2} \right] \quad (3.19)$$

Donc, la transformée discrète en cosinus est séparable.

Cette propriété a pour avantage de calculer  $X_c$  en deux séquences à une dimension, transformation des lignes puis transformation des colonnes successivement.

L'équation (3.15) de la transformée discrète en cosinus inverse peut aussi être calculée de la même manière:

$$x[n_1, n_2] = \sqrt{\frac{2}{N_1 N_2}} \sum_{k_1=0}^{N_1-1} \alpha_{k_1} \cos \left[ \frac{\pi(2n_1 + 1)k_1}{2N_1} \right] \sum_{k_2=0}^{N_2-1} \alpha_{k_2} X[k_1, k_2] \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N_2} \right] \quad (3.20)$$

La transformée discrète en cosinus inverse est séparable.

### 3.4.4 Symétrie

La transformation est symétrique si elle est séparable et satisfait la condition suivante:

$$X_1(k_2, n_2) = X_2(k_2, n_2) \quad (3.21)$$

C'est-à-dire, la fonction  $X_1$  est équivalente à la fonction  $X_2$ .

La relation entre les propriétés de symétrie et de séparabilité peut être donnée par la forme:

$$\mathbf{T} = \mathbf{A}^T \mathbf{X} \mathbf{A} \quad (3.22)$$

Où  $\mathbf{A}$  étant la matrice de transformation symétrique de  $N \times N$  avec l'entrée  $a(i, j)$  est donnée par :

$$a(i, j) = \alpha_i \sum_{j=0}^{N-1} \cos \left[ \frac{\pi(2j+1)i}{2N} \right] \quad (3.23)$$

Et  $\mathbf{X}$  est la matrice d'image de  $N \times N$  et  $(.)^T$  est la transposée.

Cette propriété permet de calculer au préalable, la matrice de la transformée en offline, puis de l'appliquer à l'image, ce qui produirait une amélioration de l'efficacité de calcul.

### 3.4.5 Orthogonalité

La transformée est orthogonale, si et seulement si la transformée de la matrice est orthogonale :

$$\mathbf{A}^T \mathbf{A} = \mathbf{I} \quad (3.24)$$

où  $\mathbf{I}$  est la matrice identité.

Cette propriété aussi permet de réduire la complexité de calcul de la transformée.

### 3.5 Algorithme rapide de la DCT

Un aspect important de l'application d'une transformée est l'existence d'un algorithme rapide pour la calculer. La transformée discrète en cosinus DCT a l'avantage de pouvoir être implémentée par l'algorithme de la transformée de Fourier rapide (TFR ou FFT). La DCT d'une séquence de  $N$  points peut être obtenue par la transformée discrète de Fourier (DFT) d'une séquence de  $2N$  points. D'autres algorithmes ont été développés par les chercheurs et sont généralement classés en deux catégories : Algorithmes à calcul indirect (utilise d'autres transformées orthogonales) et algorithmes à calcul direct (factorisation de la matrice).

#### 3.5.1 DCT implémentée par la FFT

En suivant l'approche de [68] qui décrit l'implémentation de la transformée rapide DCT à une dimension par FFT, nous présentons ci-dessous l'implémentation de la transformée rapide DCT à deux dimensions par FFT. Pour calculer la séquence à deux dimensions  $f(n_1, n_2)$ , on doit diviser la séquence en deux séquences à une dimension (la DCT est séparable). On calcule la transformation des lignes puis la transformation des colonnes.

*Transformation des lignes* : on fixe  $n_2$  et on exprime la séquence  $f(n_1, n_2)$  de longueur  $N$  en deux séquences, paire  $f_p(n_1, n_2)$  et impaire  $f_i(n_1, n_2)$ .

On définit une nouvelle séquence  $y(n_1, n_2)$  de même longueur que  $f(n_1, n_2)$  soit,

$$\begin{cases} y[n_1, n_2] \triangleq f[2n_1, n_2] \\ y[N - 1 - n_1, n_2] \triangleq f[2n_1 + 1, n_2] \end{cases} \quad \text{pour } n_1 = 0, \dots, \frac{N}{2} - 1 \quad (3.25)$$

En utilisant l'expression (3.16), la transformée en cosinus à deux dimensions de  $f(n_1, n_2)$  peut être écrite comme suit :

$$\begin{aligned} F[k_1, k_2] = & \frac{\sqrt{2}}{N} \alpha_{k_2} \sum_{n_2=0}^{N-1} \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N} \right] \left[ \sum_{n_1=0}^{N/2-1} \alpha_{k_1} f[2n_1, n_2] \cos \left[ \frac{\pi(4n_1 + 1)k_1}{2N} \right] \right. \\ & \left. + \sum_{n_1=0}^{N/2-1} \alpha_{k_1} f[2n_1 + 1, n_2] \cos \left[ \frac{\pi(4n_1 + 3)k_1}{2N} \right] \right] \end{aligned} \quad (3.26)$$

Avec  $\alpha_k$  constante définie comme suit  $\alpha_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } k = 0 \text{ ou } N \\ 1 & \text{autrement.} \end{cases}$

En substituant (3.25) dans (3.26) on obtient :

$$\begin{aligned}
F[k_1, k_2] = & \frac{\sqrt{2}}{N} \alpha_{k_2} \sum_{n_2=0}^{N-1} \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N} \right] \left[ \sum_{n_1=0}^{N/2-1} \alpha_{k_1} y[n_1, n_2] \cos \left[ \frac{\pi(4n_1 + 1)k_1}{2N} \right] \right. \\
& \left. + \sum_{n_1=0}^{N/2-1} \alpha_{k_1} y[N-1-n_1, n_2] \cos \left[ \frac{\pi(4n_1 + 3)k_1}{2N} \right] \right]
\end{aligned} \tag{3.27}$$

Posons  $m = N - 1 - n_1$  dans la deuxième sommation. La limite de sommation de 0 et  $N/2-1$  pour  $n_1$  devient  $N-1$  et  $N/2$  pour  $m$  et ainsi l'expression (3.27) s'écrira:

$$\begin{aligned}
F[k_1, k_2] = & \frac{\sqrt{2}}{N} \alpha_{k_2} \sum_{n_2=0}^{N-1} \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N} \right] \left[ \sum_{n_1=0}^{N/2-1} \alpha_{k_1} y[n_1, n_2] \cos \left[ \frac{\pi(4n_1 + 1)k_1}{2N} \right] \right. \\
& \left. + \sum_{m=N/2}^{N-1} \alpha_{k_1} y[m, n_2] \cos \left[ 2k_1 \pi - \frac{\pi(4m + 1)k_1}{2N} \right] \right]
\end{aligned} \tag{3.28}$$

Puisque,  $\cos \left[ 2k_1 \pi - \frac{\pi(4m + 1)k_1}{2N} \right] = \cos \left[ \frac{\pi(4m + 1)k_1}{2N} \right]$ , la combinaison des deux sommations de l'expression (3.28) donnera :

$$F[k_1, k_2] = \frac{\sqrt{2}}{N} \alpha_{k_2} \sum_{n_2=0}^{N-1} \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N} \right] \sum_{n_1=0}^{N-1} \alpha_{k_1} y[n_1, n_2] \cos \left[ \frac{\pi(4n_1 + 1)k_1}{2N} \right] \tag{3.29}$$

Posons :

$$g(n_1, n_2) = \sum_{n_1=0}^{N-1} \alpha_{k_1} y[n_1, n_2] \cos \left[ \frac{\pi(4n_1 + 1)k_1}{2N} \right] \tag{3.30}$$

Puisque, la DFT de  $y[n_1, n_2]$  est donnée par :

$$\text{DFT}\{y[n_1, n_2]\}_N = \sum_{n_1=0}^{N-1} y[n_1, n_2] e^{-\frac{j2\pi n_1 k_1}{N}} \tag{3.31}$$

alors, si on multiplie les deux cotés de l'expression (3.31) par  $e^{-\frac{j\pi k_1}{2N}}$ , et on considère la partie réelle ( $Re [.]$ ) on aura :

$$\begin{aligned} Re \left[ e^{-\frac{j\pi k_1}{2N}} \text{DFT} \{y[n_1, n_2]\}_N \right] &= Re \left[ e^{-\frac{j\pi k_1}{2N}} \sum_{n_1=0}^{N-1} y[n_1, n_2] e^{-\frac{j2\pi n_1 k_1}{N}} \right] \\ &= \sum_{n_1=0}^{N-1} y[n_1, n_2] \cos \left[ \frac{\pi(4n_1 + 1)k_1}{2N} \right] \end{aligned} \quad (3.32)$$

En remplaçant ce résultat dans l'expression (3.30) on aura :

$$g(n_1, n_2) = Re \left[ \alpha_{k_1} W_{2N}^{k_1/2} \text{DFT} \{y(n_1, n_2)\}_N \right] \quad (3.33)$$

avec  $W_{2N}^{k_1/2} = e^{-\frac{j\pi k_1}{2N}}$

En remplaçant l'équation (3.33) dans l'équation (3.29) on obtient :

$$F[k_1, k_2] = \frac{\sqrt{2}}{N} \alpha_{k_2} \sum_{n_2=0}^{N-1} \cos \left[ \frac{\pi(2n_2 + 1)k_2}{2N} \right] g(n_1, n_2) \quad (3.34)$$

*Transformation des colonnes:* maintenant on considère  $n_2$  variable et  $n_1$  fixe, puis on exprime la séquence  $g(n_1, n_2)$  en deux séquences, l'une à coefficients pairs  $g_p(n_1, n_2)$  et l'autre à coefficients impairs  $g_i(n_1, n_2)$ .

On définit aussi une nouvelle séquence  $z(n_1, n_2)$  de même longueur que  $g(n_1, n_2)$  :

$$\begin{cases} z(n_1, n_2) \triangleq g(n_1, 2n_2) = g_p(n_1, n_2) \\ z(n_1, N-1-n_2) \triangleq g(n_1, 2n_2+1) = g_i(n_1, n_2) \end{cases} \quad \text{pour } n_1 = 0, \dots, \frac{N}{2}-1 \quad (3.35)$$

En substituant (3.35) dans (3.34) on obtient ce qui suit :

$$\begin{aligned} F[k_1, k_2] &= \frac{\sqrt{2}}{N} \alpha_{k_2} \left[ \sum_{n_2=0}^{N/2-1} g(n_1, 2n_2) \cos \left[ \frac{\pi(4n_2 + 1)k_2}{2N} \right] \right. \\ &\quad \left. + \sum_{n_2=0}^{N/2-1} g(n_1, 2n_2 + 1) \cos \left[ \frac{\pi(4n_2 + 3)k_2}{2N} \right] \right] \end{aligned} \quad (3.36)$$

D'où,

$$F[k_1, k_2] = \frac{\sqrt{2}}{N} \alpha_{k_2} \left[ \sum_{n_2=0}^{N/2-1} z(n_1, n_2) \cos \left[ \frac{\pi(4n_2 + 1)k_2}{2N} \right] \right. \\ \left. + \sum_{n_2=0}^{N/2-1} z(n_1, N-1-n_2) \cos \left[ \frac{\pi(4n_2 + 3)k_2}{2N} \right] \right] \quad (3.37)$$

En effectuant le changement de variable,  $m = N - 1 - n_2$  dans la deuxième sommation, la limite de sommation de 0 et  $N/2-1$  pour  $n_2$  devient  $N-1$  et  $N/2$  pour  $m$  et ainsi l'expression (3.37) s'écrira :

$$F[k_1, k_2] = \frac{\sqrt{2}}{N} \alpha_{k_2} \left[ \sum_{n_2=0}^{N/2-1} z(n_1, n_2) \cos \left[ \frac{\pi(4n_2 + 1)k_2}{2N} \right] \right. \\ \left. + \sum_{m=N/2}^{N-1} z(n_1, m) \cos \left[ 2k_2 \pi - \frac{\pi(4m + 1)k_2}{2N} \right] \right] \quad (3.38)$$

La combinaison des deux sommations donne :

$$F[k_1, k_2] = \frac{\sqrt{2}}{N} \alpha_{k_2} \sum_{n_2=0}^{N-1} z(n_1, n_2) \cos \left[ \frac{\pi(4n_2 + 1)k_2}{2N} \right] \quad (3.39)$$

En suivant les mêmes étapes de (3.31) à (3.32) utilisées dans la transformation des lignes on obtient :

$$Re \left[ e^{-\frac{j\pi k_2}{2N}} \text{DFT}\{z[n_1, n_2]\}_N \right] = \sum_{n_2=0}^{N-1} z[n_1, n_2] \cos \left[ \frac{\pi(4n_2 + 1)k_2}{2N} \right] \quad (3.40)$$

D'où,

$$F[k_1, k_2] = \frac{\sqrt{2}}{N} \alpha_{k_2} Re \left[ W_{2N}^{k_2/2} \text{DFT}\{z(n_1, n_2)\}_N \right] \quad (3.41)$$

À partir de cette expression, on remarque que le calcul rapide de la DCT donné par  $F[k_1, k_2]$  est obtenu par le calcul rapide de la DFT de  $z(n_1, n_2)$ .

### 3.5.2 DCT implémentée par méthode directe

La méthode directe appliquée pour le calcul de la DCT est basée sur la factorisation de la matrice de DCT-II. Cette méthode produit le même résultat mais avec moins d'opérations arithmétiques [20]. Le premier algorithme rapide à méthode directe a été reproduit dans [39]. La matrice de la transformée est définie comme suit :

$$C_N^{II} = B_N \begin{pmatrix} \hat{C}_{\frac{N}{2}}^{II} & 0 \\ 0 & \hat{C}_{\frac{N}{2}}^{IV} J_{\frac{N}{2}} \end{pmatrix} \begin{pmatrix} I_{\frac{N}{2}} & J_{\frac{N}{2}} \\ J_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{pmatrix} \quad (3.42)$$

Où  $B_N$  est la matrice de permutation. Cette matrice permute les données de la séquence transformée d'un ordre de bit inversé à un ordre naturel, c'est à dire ordonner les lignes paires en sens croissant et les lignes impaires en sens décroissant.  $J_{\frac{N}{2}}$  est la matrice d'identité croisée à  $N/2$

points et  $I_{\frac{N}{2}}$  est la matrice d'identité à  $N/2$  points.  $\hat{C}_{\frac{N}{2}}^{II}$  est la matrice de la transformée discrète en

cosinus Type II à  $N/2$  points donnée par  $\hat{C}_{\frac{N}{2}}^{II} = \frac{2}{\sqrt{N}} \left[ \cos \frac{\pi k (2n+1)}{N} \right]$  avec  $k, n = 0, 1, \dots, N-1$ , et

$\hat{C}_{\frac{N}{2}}^{IV}$  est la matrice de la transformée discrète en cosinus Type IV à  $N/2$  points donnée par  $\hat{C}_{\frac{N}{2}}^{IV} =$

$\frac{2}{\sqrt{N}} \left[ \cos \frac{\pi (2n+1)(2k+1)}{2N} \right]$  avec  $k, n = 0, 1, \dots, N-1$ . Pour bien assimiler cette transformation, un

exemple ci-dessous est présenté pour  $N=4$ .

Exemple pour  $N=4$  :

En utilisant l'équation (3.7) on obtient

$$C_4^{II} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{8} & \sin \frac{\pi}{8} & -\sin \frac{\pi}{8} & -\cos \frac{\pi}{8} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \sin \frac{\pi}{8} & -\cos \frac{\pi}{8} & \cos \frac{\pi}{8} & -\sin \frac{\pi}{8} \end{bmatrix} \quad (3.43)$$

À partir de cette matrice on doit trouver l'expression donnée dans l'équation (3.42) alors si on

pose  $C_8^k = \cos \frac{k\pi}{8}$  pour  $k = 1, 2, 3, 5, 6, 7$  on aura :

$$C_4^{II} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix} \quad (3.44)$$

En utilisant la matrice de permutation  $B_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ , et en effectuant les transformations suivantes  $C_8^2 = C_4^1$ ,  $C_8^6 = C_4^3$ ,  $C_8^7 = -C_8^1$ ,  $C_8^5 = -C_8^3$  on obtient facilement :

$$C_4^{II} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_4^1 & C_4^3 & C_4^3 & C_4^1 \\ C_8^3 & -C_8^1 & C_8^1 & C_8^3 \\ C_8^1 & C_8^3 & -C_8^3 & C_8^1 \end{bmatrix} \quad (3.45)$$

Maintenant on factorise la deuxième matrice de l'équation (3.45) en introduisant la matrice

$\begin{pmatrix} I_{\frac{4}{2}} & J_{\frac{4}{2}} \\ J_{\frac{4}{2}} & -I_{\frac{4}{2}} \end{pmatrix}$ , on a ce qui suit :

$$C_4^{II} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ C_4^1 & C_4^3 & 0 & 0 \\ 0 & 0 & -C_8^1 & C_8^3 \\ 0 & 0 & C_8^3 & C_8^1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix} \quad (3.46)$$

À partir de cette équation on peut observer que  $C_4^{II}$  donné dans (3.43) peut être exprimé sous la

$$\text{forme } C_4^{II} = B_N \begin{pmatrix} \hat{C}_4^{II} & 0 \\ 0 & \hat{C}_4^{IV} J_{\frac{4}{2}} \end{pmatrix} \begin{pmatrix} I_{\frac{4}{2}} & J_{\frac{4}{2}} \\ J_{\frac{4}{2}} & -I_{\frac{4}{2}} \end{pmatrix}.$$

On peut conclure que la fonction DCT type II à  $N$  points est décomposée récursivement en deux fonctions, DCT type II paire à  $N/2$  points et DCT type IV impaire à  $N/2$  points. Les autres algorithmes rapides de DCT type II diffèrent par la modification et l'amélioration de la matrice de factorisation DCT type IV [20]. Cette matrice de factorisation est devenue une base de développement d'algorithme rapide de DCT type II.

Pour plus d'information sur les algorithmes rapides développés pour la DCT type II, veuillez retourner à la section 4.4.3 de [20] et [37]-[40].

## 3.6 La norme de compression JPEG

JPEG est l'acronyme de Joint Photographic Expert Group. Ce comité a donné son nom à la norme de compression d'image numérique. La norme JPEG fait référence à la norme ISO/CEI 10918-1 ou UIT-T recommandation T.81. Cette norme est basée sur la technique de transformation orthogonale DCT en découplant l'image en blocs de  $16 \times 16$ ,  $8 \times 8$ ,  $4 \times 4$ ,  $4 \times 8$  ou  $8 \times 4$ .

Il y a quatre modes de compression par JPEG [4], [44], [50]:

- Encodage séquentiel
- Encodage progressif
- Encodage sans perte
- Encodage hiérarchique

Nous présentons dans ce qui suit le principe général de l'encodage séquentiel qui est la base des autres modes de compression [4].

### 3.6.1 Encodage séquentiel JPEG

Les différentes étapes de l'encodage séquentiel JPEG sont comme suit :

- Un prétraitement est appliqué sur l'image à compresser
- L'image est coupée en bloc de taille  $8 \times 8$
- La DCT est appliquée sur chaque bloc de gauche à droite et de haut en bas.
- Chaque bloc est compressé par quantification
- Codage de la matrice quantifiée en zigzag
- Compression par l'algorithme de Huffman

Le diagramme bloc de compression et de décompression JPEG est donné à la figure 3.6:

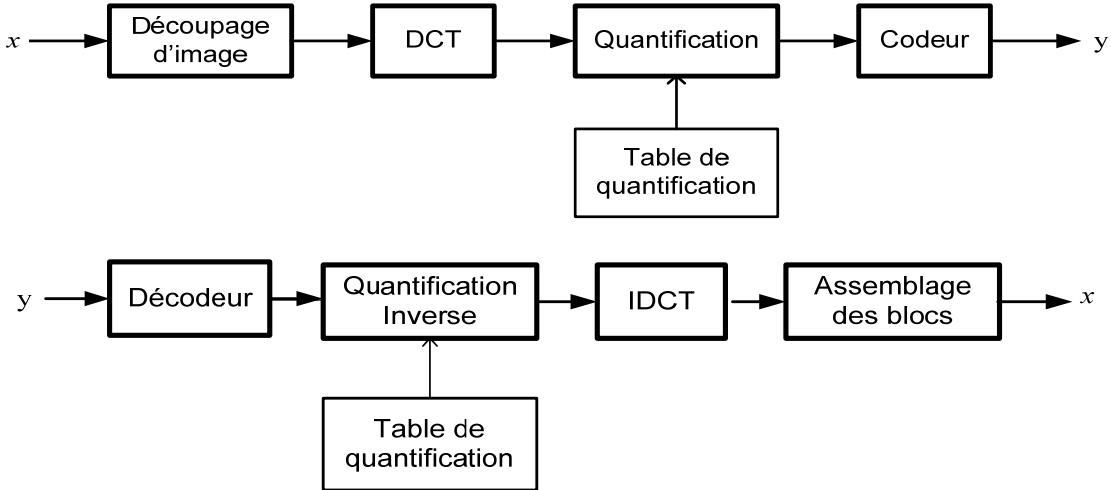


Figure 3.6: Codeur et décodeur JPEG

### 3.6.2 Équation DCT d'une image de taille $8 \times 8$

Soit une image à deux dimensions  $x(n_1, n_2)$  de taille  $8 \times 8$  ( $N_1 = N_2 = 8$ ). La transformation de cette image vers le domaine fréquentiel se fait en lui appliquant l'équation de la transformation directe en cosinus. L'expression (3.14) deviendra comme suit:

$$X_c[k_1, k_2] = \frac{1}{4} \alpha_{k_1} \alpha_{k_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x[n_1, n_2] \cos \left[ \frac{\pi(2n_1 + 1)k_1}{16} \right] \cos \left[ \frac{\pi(2n_2 + 1)k_2}{16} \right] \quad (3.47)$$

Aussi, l'équation de transformation inverse en cosinus de l'expression (2.11) sera:

$$x[n_1, n_2] = \frac{1}{4} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \alpha_{k_1} \alpha_{k_2} X_c[k_1, k_2] \cos \left[ \frac{\pi(2n_1 + 1)k_1}{16} \right] \cos \left[ \frac{\pi(2n_2 + 1)k_2}{16} \right] \quad (3.48)$$

### 3.6.3 Matrice de forme de DCT

Pour le calcul rapide on utilise la matrice de forme dont les éléments sont donnés par la formule (3.38):

$$\mathbf{T}(k_1, k_2) = \begin{cases} \frac{1}{\sqrt{N}} & \text{si } k_1 = 0 \\ \sqrt{\frac{2}{N}} \cos \left[ \frac{(2k_2 + 1)k_1 \pi}{2N} \right] & \text{si } k_1 > 0 \end{cases} \quad (3.49)$$

Pour un bloc de  $8 \times 8$  le développement de l'équation T donne:

$$\mathbf{T} = \begin{pmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{pmatrix}$$

### 3.6.4 Prétraitement

Pour une image en niveau de gris, les valeurs des pixels sont rangées de 0 à 255 avec un pas de 1. Un noir pur est représenté par zéro et un blanc pur est représenté par 255. Or, la DCT est applicable sur des pixels qui sont rangés de -128 à 127 [53], alors le bloc original est nivelé par la soustraction de 128 pour chaque coefficient de la matrice d'entrée.

Pour une image RVB [3], les pixels sont représentés par trois couleurs, le rouge, le vert et le bleu. Chaque couleur est rangée de 0 à 255 avec un pas de 1. Comme la plus grande valeur que peut prendre un pixel d'une image en niveau de gris est codée en 8 bits ( $2^8 = 256$ ) alors le pixel d'une image RVB est codée en 24 bits ( $3 \times 8 = 24$ ). Pour la compression, il est préférable de procéder à la transformation des composantes RVB (R : rouge, V : vert, B : bleu) dans l'espace YCbCr (Y : luminance, Cb et Cr : chrominance) car la compression des composantes Cb et Cr se fait sans perte visible de détails.

La transformation s'écrit:

$$Y = 0.299R + 0.587V + 0.114B \quad (3.50)$$

$$C_r = 0.169R + 0.331V + 0.5B + 128 \quad (3.51)$$

$$C_b = 0.5R + 0.419V + 0.081B + 128 \quad (3.52)$$

### 3.6.5 Application de la DCT sur les blocs de taille $8 \times 8$

La DCT va être appliquée sur chaque bloc après prétraitement par la fonction :

$$\mathbf{D} = \mathbf{T} \mathbf{M} \mathbf{T}^T \quad (3.53)$$

**M** est la matrice nivélée,

**T** est la matrice de forme orthogonale et  $\mathbf{T}^T$  sa transposée,

**D** est la forme matricielle équivalente de l'équation (3.14).

La matrice **M** est multipliée par **T** du coté gauche, (**TM**). Alors nous avons la transformation des lignes puis le résultat est multiplié par la  $\mathbf{T}^T$  pour réaliser la transformation des colonnes.

### 3.6.6 Quantification

Comme l'œil humain est moins sensible aux hautes fréquences, les éléments de la matrice de quantification sont représentés par un coefficient de poids correspondant à leur importance dans l'image. La matrice de quantification **Q** n'est pas unique; elle dépend du facteur de qualité choisi. Ce choix nous informe de la qualité et du niveau de compression désiré.

Si le facteur de qualité de la matrice de quantification est très petit alors la qualité de l'image reconstruite est médiocre et son taux de compression est élevé. Par contre, si ce facteur de qualité est très élevé alors la qualité de l'image reconstruite est très bonne mais avec un taux de compression faible.

Des expériences subjectives utilisant le système visuel humain ont fait jaillir la matrice de quantification standard de JPEG [3]. La matrice ci-dessous donne un taux de compression très élevé et une qualité de décompression d'image excellente [3].

$$\mathbf{Q} = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}.$$

La référence [72] présente une autre façon de calculer les coefficients  $Q(i,j)$  de la matrice de quantification en tenant compte des critères psycho-visuels et sans avoir recours à la matrice de quantification du standard JPEG ci-dessus. Ses coefficients sont calculés par la formule suivante:

$$Q(i,j) = 1 + (1 + i + j) \times f_q \quad (3.54)$$

Avec  $f_q$ : facteur de qualité, varie de 1 à 25 et  $i,j$  : dimensions de la matrice, varient de 0 à 7.

C'est dans l'étape de quantification que l'information se perd et la précision de l'image reconstruite est déterminée. Chaque coefficient de l'image va être divisé par le coefficient correspondant dans la matrice de quantification et le résultat sera arrondi à une valeur entière.

Le nombre de zéros contenus dans la matrice résultante après quantification varie suivant le choix du facteur de qualité.

### 3.6.7 Codage

Après quantification, les 64 coefficients de chaque bloc sont traités pour l'encodage entropique. L'encodage entropique utilise des statistiques sur la source d'entrée pour former un code. Les propriétés de ce code dépendent de la source. L'encodage Huffmann et l'encodage arithmétique sont deux principaux types de l'encodage entropique.

Le traitement avant encodage est présenté dans figure 3.7 et les différentes étapes de ce traitement sont données comme suit:

Dans chaque bloc nous avons 64 coefficients; le premier coefficient  $C_{0,0}$  est la composante DC du bloc et les 63 autres coefficients sont les composantes AC ( $AC_{i,j} = C_{i,j}$ ,  $j = 1, \dots, 7$  et  $i = 0, 1, \dots, 7$ ) du bloc. Les coefficients DC et les coefficients AC sont codés différemment. Si on prend une image  $N \times N$  avec  $N=256$  on aura 32 blocs de  $8 \times 8$  donc 32 coefficients DC. Pour l'encodage des coefficients DC, on code la différence des coefficients DC des différents blocs successivement suivant la formule suivante  $Diff_b = DC_b - DC_{b-1}$  avec  $b = 1, \dots, N/8$ . Ainsi, le coefficient DC précédent prédit le coefficient suivant. Pour l'encodage des coefficients AC on utilise pour chaque bloc le codage en zigzag. Cette méthode permet d'augmenter la taille des suites de zéro.

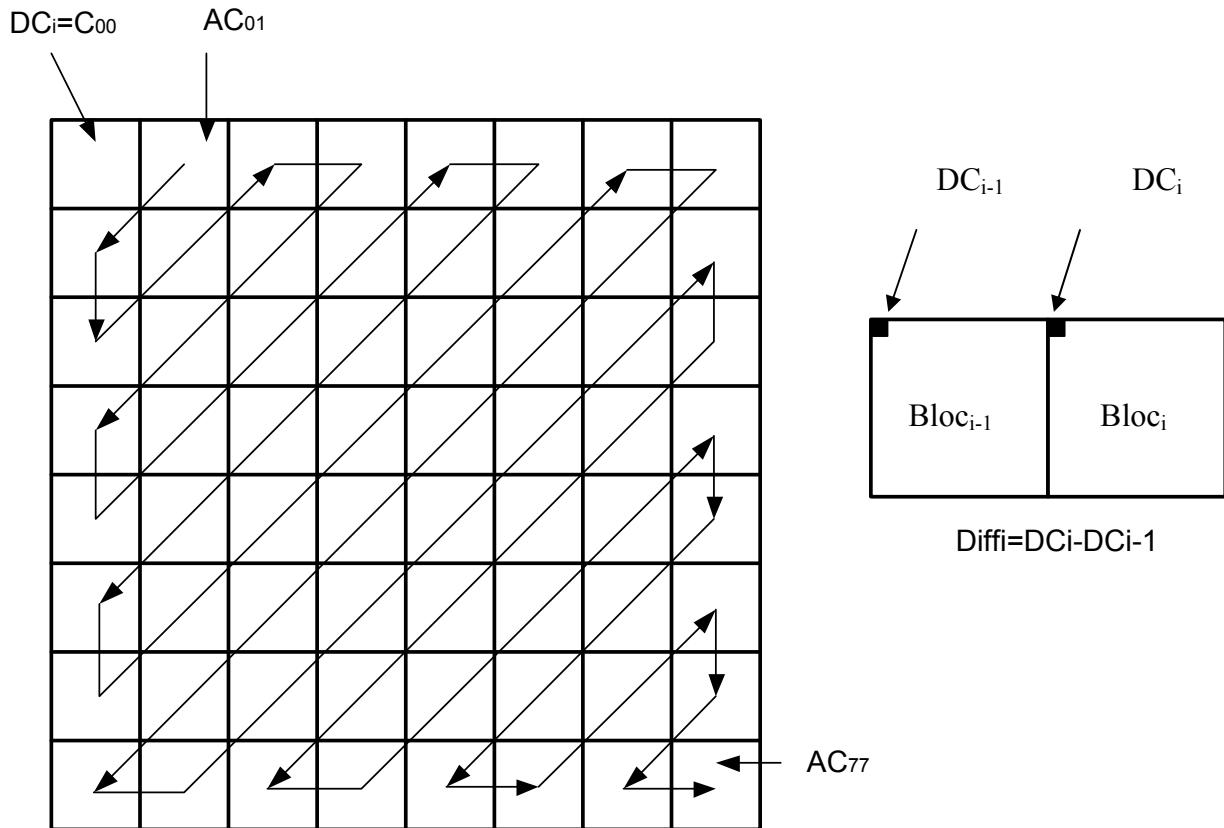


Figure 3.7 : Codage en zigzag pour chaque bloc

### 3.6.8 Reconstruction de l'image

La phase de reconstruction de l'image suit le chemin inverse de la phase de compression. Après le décodage des données de l'image compressée, chaque élément de la matrice obtenue est multiplié par le coefficient qui lui correspond dans la matrice de quantification, précédemment utilisée. C'est uniquement la partie entière de chaque coefficient qui est considérée en sortie. La transformée discrète en cosinus inverse est appliquée sur la sortie du quantificateur inverse. Du moment que nous avons appliqué un prétraitement au début de la compression en appliquant la soustraction du nombre 128 pour chaque coefficient, alors un ajout du nombre 128 est appliqué pour chaque coefficient de la matrice de transformation inverse.

### 3.6.9 Exemple

Pour bien visualiser les différentes étapes de la compression par DCT, nous avons pris le premier bloc  $8 \times 8$  de l'image *wbarb* ( $\mathbf{M}$ ) et nous avons appliqué le processus de compression au complet.

*La matrice  $8 \times 8$  à compresser:*

$$\mathbf{M} = \begin{pmatrix} 17 & 23 & 17 & 23 & 27 & 23 & 23 & 30 \\ 56 & 55 & 59 & 63 & 59 & 60 & 60 & 65 \\ 65 & 63 & 69 & 65 & 69 & 69 & 72 & 72 \\ 69 & 72 & 76 & 76 & 82 & 84 & 95 & 108 \\ 105 & 112 & 122 & 134 & 142 & 153 & 159 & 160 \\ 149 & 146 & 136 & 131 & 116 & 108 & 90 & 80 \\ 64 & 53 & 47 & 36 & 31 & 29 & 33 & 39 \\ 45 & 57 & 72 & 79 & 96 & 103 & 118 & 126 \end{pmatrix}$$

*Prétraitement:*

On retranche le nombre 128 à la matrice  $8 \times 8$  et on obtient :

$$\mathbf{M}' = \mathbf{M} - 128 = \begin{pmatrix} -111 & -105 & -111 & -105 & -101 & -105 & -105 & -98 \\ -72 & -73 & -69 & -65 & -69 & -68 & -68 & -63 \\ -63 & -65 & -59 & -63 & -59 & -59 & -56 & -56 \\ -59 & -56 & -52 & -52 & -46 & -44 & -33 & -20 \\ -23 & -16 & -6 & 6 & 14 & 25 & 31 & 32 \\ 21 & 18 & 8 & 3 & -12 & -20 & -23 & -48 \\ -64 & -75 & -81 & -92 & -97 & -99 & -95 & -89 \\ -83 & -71 & -56 & -49 & -32 & -25 & -10 & -2 \end{pmatrix}$$

#### Remarque

Si on applique la valeur absolue sur les coefficients de la matrice après prétraitement, on observe que les coefficients les plus élevés sont présents sur la première ligne de la matrice et que le coefficient le plus élevé correspond au premier coefficient de la matrice.

*Application de la DCT:*

La transformation DCT est appliquée à  $\mathbf{M}'$ , le résultat est la matrice  $\mathbf{D}$

$$\mathbf{D} = \begin{pmatrix} -406.875 & -32.7756 & 3.7003 & -6.9407 & 2.625 & -2.114 & 0.7674 & 1.0272 \\ -122.7516 & 1.1753 & -2.9122 & -0.1802 & 3.3924 & -1.7106 & -0.9811 & -2.008 \\ -188.8345 & -5.8142 & 2.1062 & -1.8838 & 2.2092 & -3.0997 & -1.638 & 0.9699 \\ 32.7689 & 60.4568 & -7.0628 & 8.0392 & 0.5468 & -0.2609 & -2.0338 & -7.2928 \\ 39.875 & -88.2675 & -2.156 & -6.0612 & -0.125 & -4.9191 & -3.5718 & 1.532 \\ -152.2 & 37.4349 & 17.8185 & -0.9156 & 3.2796 & -2.9053 & 0.0242 & 2.3619 \\ 53.9993 & 18.7591 & -13.388 & 4.6601 & -3.4858 & -1.8305 & -1.6062 & -0.0365 \\ -18.8039 & -27.1627 & 2.02516 & 2.3055 & -1.5409 & 1.6776 & 0.3056 & 2.1908 \end{pmatrix}$$

On observant cette matrice, on remarque les points suivants:

- Les coefficients à amplitude la plus élevée en valeur absolue sont situés en haut de la matrice et du coté gauche
- L'amplitude la plus élevée de la matrice est représentée par le premier coefficient de la matrice
- L'amplitude du premier coefficient de la matrice est presque la somme de tous les autres coefficients de la matrice

*Application de la quantification:*

La matrice de quantification  $\mathbf{Q}$  utilisée est la matrice de luminance standard de JPEG.

$$\mathbf{Q} = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

Chaque coefficient DCT ( $D(i,j)$ ) est divisé par le coefficient de quantification  $Q(i,j)$  correspondant dans la matrice et le résultat sera arrondi à la valeur entière.

$$\mathbf{A} = \text{Arrondi}\left(\frac{D(i,j)}{Q(i,j)}\right) = \begin{pmatrix} -25 & -3 & 0 & 0 & 0 & 0 & 0 & 0 \\ -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Nous avons beaucoup de zéros comme prévu et la valeur la plus élevée en valeur absolue est localisée dans le premier coefficient de la matrice.

Le résultat de la matrice  $\mathbf{A}$  est codé en zigzag puis encodé par le codeur Huffman pour réaliser le codage JPEG.

*Décompression:*

Pour la décompression de la matrice quantifiée  $\mathbf{A}$ , on va multiplier chaque coefficient  $A(i,j)$  de la matrice quantifiée par le pas de quantification de la table de quantification utilisée en phase de quantification et le résultat sera arrondi à la valeur entière:

$$\mathbf{N} = \text{Arrondi}(A(i,j) \times Q(i,j))$$

La transformation inverse en cosinus est appliquée à la matrice résultante, puis la valeur 128 est ajoutée au résultat.

$$\mathbf{B} = \text{IDCT}(\mathbf{N}) + 128 = \begin{pmatrix} 23 & 23 & 24 & 25 & 26 & 27 & 27 & 28 \\ 57 & 58 & 58 & 59 & 60 & 61 & 61 & 62 \\ 67 & 68 & 69 & 71 & 73 & 75 & 77 & 77 \\ 58 & 61 & 68 & 76 & 85 & 94 & 100 & 104 \\ 118 & 121 & 126 & 133 & 141 & 148 & 153 & 156 \\ 140 & 136 & 129 & 120 & 111 & 102 & 95 & 91 \\ 68 & 65 & 59 & 51 & 43 & 35 & 29 & 26 \\ 47 & 53 & 64 & 79 & 95 & 109 & 121 & 127 \end{pmatrix}$$

Ainsi,  $\mathbf{B}$  représente notre matrice après décompression. Pour vérifier la précision ou bien l'écart qui existe entre l'image originale et l'image reconstruite, on calcule la différence des deux matrices.

$$\mathbf{M} - \mathbf{B} = \begin{pmatrix} -6 & 0 & -7 & -2 & 1 & -4 & -4 & 2 \\ -1 & -3 & 1 & 4 & -1 & -1 & -1 & 3 \\ -2 & -5 & 0 & -6 & -4 & -8 & -5 & -5 \\ 10 & 9 & 8 & 0 & -3 & -10 & -5 & 4 \\ -13 & -9 & -4 & 1 & 1 & 5 & 6 & -4 \\ 9 & 10 & 7 & 11 & 5 & 6 & -5 & 11 \\ -4 & -12 & -12 & -15 & -12 & -6 & 4 & 13 \\ -2 & 4 & 8 & 0 & 1 & -6 & -3 & -1 \end{pmatrix}$$

On remarque que plus de 70% des valeurs de la matrice décompressée sont très proches des valeurs de la matrice originale **M**.

Dans le chapitre 6, nous allons voir avec des exemples d'images, le résultat de la compression par la transformée DCT.

## CHAPITRE 4 LA TRANSFORMÉE DISCRÈTE EN ONDELETTES

Pour présenter la technique de la transformée discrète en ondelettes, nous débutons ce chapitre par une concise introduction suivie d'un rappel des concepts de base reliés à la transformée en ondelettes; les bancs de filtres et les conditions sur les filtres pour avoir une reconstruction parfaite du signal sont décrits. Par la suite, les bases d'ondelettes, les fonctions d'échelles, l'algorithme rapide de la décomposition multi-résolution et une description de l'implémentation de la transformée discrète en ondelettes pour des signaux monodimensionnel et bidimensionnel sont exposés. Finalement, les paramètres à considérer lors de la conception d'une architecture à base de la DWT utilisée pour la compression d'images sont présentés.

### 4.1 Introduction

La transformée de Fourier est la technique la plus utilisée dans le traitement du signal. Elle permet le passage du domaine temporel au domaine fréquentiel. Dans ce passage, l'information temps est perdue pour les signaux non stationnaires [1], [2], [5], [6], [44], [49], d'où l'introduction de la technique des fenêtres. Grâce à cette technique, il est devenu possible de connaître à quel moment un événement du signal c'est produit mais avec une largeur de la fenêtre réglée d'avance [49].

La transformée en ondelettes ou de Wavelet est une solution, elle permet d'avoir la taille de la fenêtre variable. Au lieu de représenter un signal comme la somme pondérée de sinusoïdes, nous avons la somme pondérée d'ondes translatées et dilatées dans le temps.

La transformée discrète en ondelettes (DWT) permet la représentation des coefficients du signal ou de l'image partiellement localisée en temps et en fréquence, non redondant et réversible [18], [49], [51].

## 4.2 Concepts de base

### 4.2.1 Décimation et interpolation

La décimation d'un signal  $x(n)$  par un facteur  $k$ , est le prélèvement d'un échantillon sur  $k$  [1], en suivant la formule suivante:

$$x_d(m) = x(n) \quad \text{pour } n = m/k \text{ et } m \text{ entier} \quad (4.1)$$

La décimation est symbolisée par la figure 4.1 :

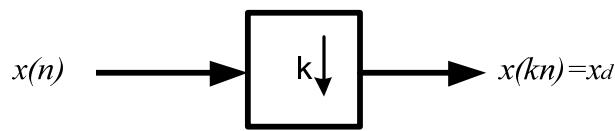


Figure 4.1: Opération de décimation

L'interpolation d'un signal  $x(n)$  par un facteur  $k$ , est l'insertion de zéros par un facteur de  $k$  [1], en suivant la formule suivante:

$$x_u(m) = \begin{cases} x(n) & \text{pour } n = km \\ 0 & \text{ailleurs} \end{cases} \quad (4.2)$$

L'interpolation est symbolisée par la figure 4.2

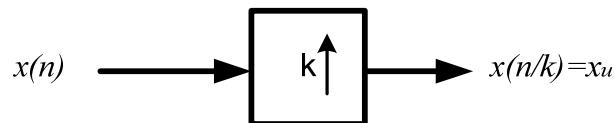


Figure 4.2: Opération d'interpolation

## 4.2.2 Bancs de filtres

Un banc de filtres est un ensemble de filtres avec une entrée ou une sortie [68]. Le système de la figure 4.3 est appelé banc de filtres d'analyse,  $h_i(n)$  sont des filtres d'analyse et  $k$  le facteur de décimation.

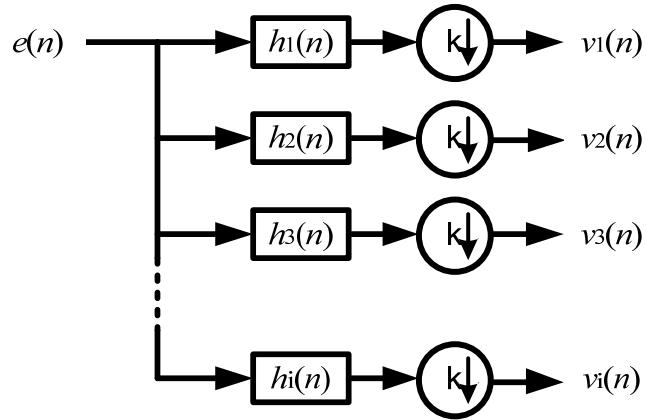


Figure 4.3: Phase d'analyse des bancs de filtres

De même le système de la figure 4.4 est appelé banc de filtres de synthèse,  $g_i(n)$  sont des filtres de synthèse et  $k$  le facteur d'interpolation.

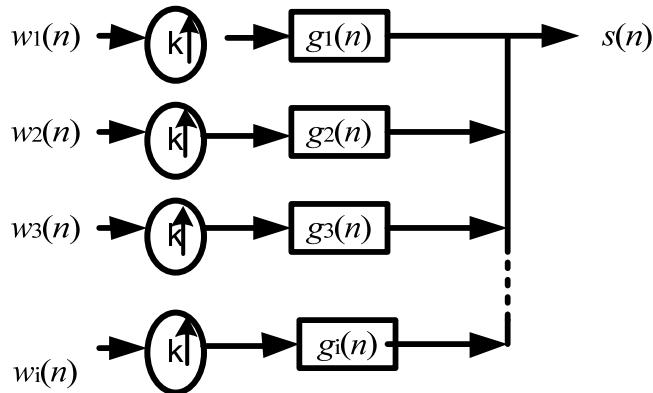


Figure 4.4 : Phase de synthèse des bancs de filtres

### 4.2.3 Filtres à reconstruction parfaite et filtres miroirs conjugués

Lorsque, la sortie du banc de filtres de synthèse est égale à l'entrée du banc de filtres d'analyse sans avoir effectué aucun traitement alors on parlera de filtres à reconstruction parfaite.

#### *Phase d'analyse :*

La figure 4.5 est un exemple d'un banc de filtres discrets à deux canaux, le signal d'entrée est convolué avec le filtre passe-bas  $h_1(n)$  suivi par une décimation (ou sous échantillonnage) d'un facteur de 2 à la sortie. De même, ce même signal d'entrée est convolué avec le filtre passe-haut  $h_2(n)$  suivi par une décimation d'un facteur de 2 à la sortie.

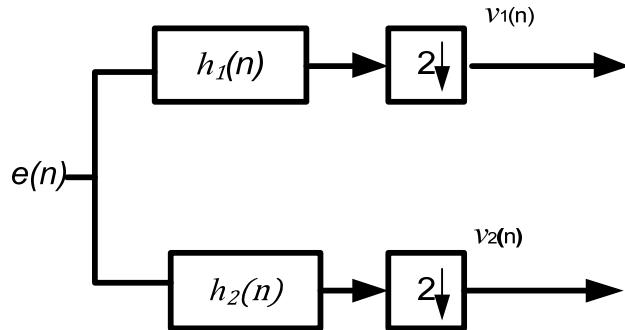


Figure 4.5 : Banc de filtres à deux canaux, phase d'analyse

La phase d'analyse est définie comme suit :

$$v_1(n) = e(n) * h_1(n) \downarrow 2 \quad (4.3)$$

$$v_2(n) = e(n) * h_2(n) \downarrow 2 \quad (4.4)$$

\* : opération de convolution

#### *Phase de synthèse:*

La phase de synthèse est la phase de reconstruction du signal décomposé. Après interpolation d'un facteur de 2 (insertion de zéros) des deux signaux de sorties de la phase d'analyse  $v_1(n)$  et  $v_2(n)$  et la convolution de ces signaux avec les filtres passe-bas  $g_1(n)$  et passe-haut  $g_2(n)$

respectivement, les résultats des deux sorties des filtres seront sommés pour obtenir le signal de sortie  $y(n)$ , voir figure 4.6.

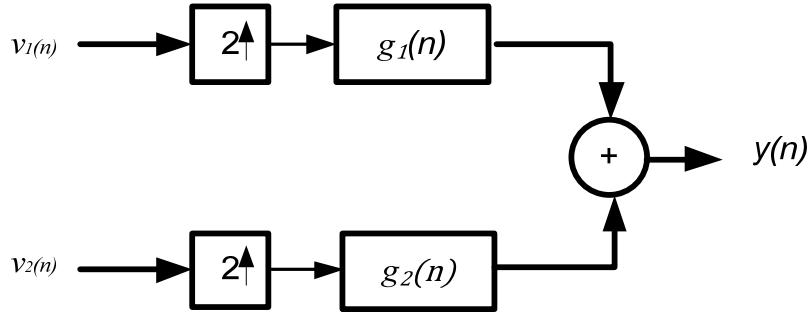


Figure 4.6 : Banc de filtres à deux canaux, phase de synthèse

La phase de synthèse est définie comme suit :

$$y(n) = v_1(n) \uparrow 2 * g_1(n) + v_2(n) \uparrow 2 * g_2(n) \quad (4.5)$$

Les bancs de filtres sont à reconstruction parfaite si le signal de sortie est égal au signal d'entrée [68]. Si en plus, les filtres d'analyse sont égaux aux filtres de synthèse deux à deux,  $h_1(n) = g_1(n)$  et  $h_2(n) = g_2(n)$  alors, on parle de filtres miroirs conjugués [68].

#### 4.2.4 Condition de reconstruction parfaite du signal

Pour voir les conditions de reconstruction parfaite du signal, travaillons dans le domaine de la transformée en  $z$  qui est le cas général de la transformée de Fourier.

Soit la séquence  $x[n]$  dont la transformée en  $Z$  est:

$$X(z) = \sum_{n=-\infty}^{+\infty} x[n]z^{-n} \quad (4.6)$$

Où  $z$  est la variable complexe.

L'interpolation et la décimation de  $x(n)$  dans le domaine  $z$  sont comme suit:

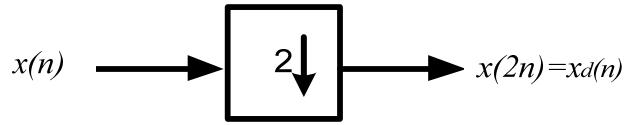


Figure 4. 7 : Opération de décimation d'un facteur de 2

$$x_d(n) = x(2n) \Leftrightarrow X_d(z) = \frac{1}{2} [X(z^{1/2}) + X(-z^{1/2})] \quad (4.7)$$

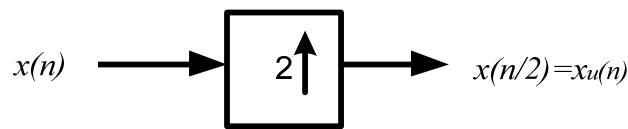


Figure 4. 8 : Opération d'interpolation d'un facteur de 2

$$x_u(n) = \begin{cases} x\left(\frac{n}{2}\right) & n = 0, 2, 4, \dots \\ 0 & \text{autrement} \end{cases} \Leftrightarrow X_u(z) = X(z^2) \quad (4.8)$$

Avec,  $x_d(n)$  le signal décimé et

$x_u(n)$  le signal interpolé.

Si la séquence  $x(n)$  est décimée puis interpolée, on aura le signal de sortie  $y(n)$  comme présenté dans figure. 4.9 :

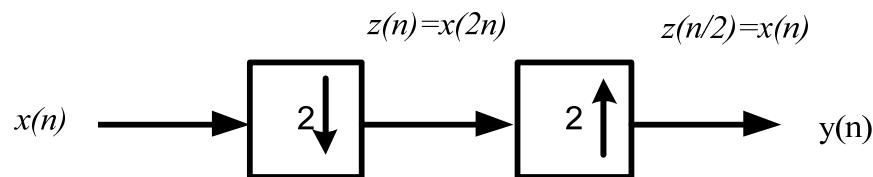


Figure 4. 9 : Opération de décimation et d'interpolation

Alors,

$$Y(z) = X_d(z^2) = \frac{1}{2} \left[ X \left( \left( z^{\frac{1}{2}} \right)^2 \right) + X \left( -\left( z^{\frac{1}{2}} \right)^2 \right) \right] \quad (4.9)$$

D'où,

$$Y(z) = \frac{1}{2} [X(z) + X(-z)] \quad (4.10)$$

Appliquons le résultat de l'expression (4.10) à la Figure 4.10

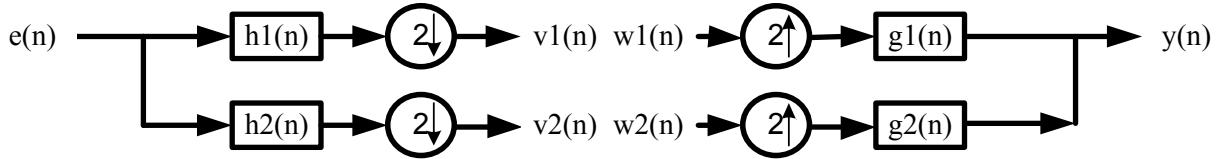


Figure 4. 10 : Phase d'analyse et de synthèse

Avec  $E(z)$ ,  $H_1(z)$ ,  $H_2(z)$ ,  $G_1(z)$ ,  $G_2(z)$ ,  $V_1(z)$ , et  $V_2(z)$  sont les transformées en  $z$  de  $e(n)$ ,  $h_1(n)$ ,  $h_2(n)$ ,  $g_1(n)$ ,  $g_2(n)$   $v_1(n)$  et  $v_2(n)$  , respectivement; on aura,

La transformée en  $z$  de la sortie de la sous bande  $v_1(n)$  est :

$$V_1(z) = \frac{1}{2} \left[ E \left( z^{(1/2)} \right) H_1 \left( z^{(1/2)} \right) + E \left( -z^{(1/2)} \right) H_1 \left( -z^{(1/2)} \right) \right] \quad (4.11)$$

La transformée en  $Z$  de la sortie de la sous bande  $v_2(n)$  est :

$$V_2(z) = \frac{1}{2} \left[ E \left( z^{(1/2)} \right) H_2 \left( z^{(1/2)} \right) + E \left( -z^{(1/2)} \right) H_2 \left( -z^{(1/2)} \right) \right] \quad (4.12)$$

La transformée de  $Z$  de la sortie  $y(n)$  est :

$$Y(z) = [W_1(z^2)G_1(z) + W_2(z^2)G_2(z)] \quad (413)$$

Comme,  $V_1(z)$  est égal à  $W_1(z)$  et  $V_2(z)$  est égal à  $W_2(z)$  alors :

$$\begin{aligned} Y(z) = & \left[ \frac{1}{2} \left[ E \left( \left( z^{\frac{1}{2}} \right)^2 \right) H_1 \left( \left( z^{\frac{1}{2}} \right)^2 \right) + E \left( -\left( z^{\frac{1}{2}} \right)^2 \right) H_1 \left( -\left( z^{\frac{1}{2}} \right)^2 \right) \right] G_1(z) \right. \\ & \left. + \frac{1}{2} \left[ E \left( \left( z^{\frac{1}{2}} \right)^2 \right) H_2 \left( \left( z^{\frac{1}{2}} \right)^2 \right) + E \left( -\left( z^{\frac{1}{2}} \right)^2 \right) H_2 \left( -\left( z^{\frac{1}{2}} \right)^2 \right) \right] G_2(z) \right] \end{aligned} \quad (4.14)$$

$$Y(z) = \left[ \frac{1}{2} [E(z)H_1(z) + E(-z)H_1(-z)]G_1(z) + \frac{1}{2} [E(z)H_2(z) + E(-z)H_2(-z)]G_2(z) \right] \quad (4.15)$$

$$Y(z) = \frac{1}{2} G_1(z) [E(z)H_1(z) + E(-z)H_1(-z)] + \frac{1}{2} G_2(z) [E(z)H_2(z) + E(-z)H_2(-z)] \quad (4.16)$$

$$Y(z) = \frac{1}{2} E(z) [G_1(z)H_1(z) + G_2(z)H_2(z)] + \frac{1}{2} E(-z) [G_1(z)H_1(-z) + G_2(z)H_2(-z)] \quad (4.17)$$

Pour une reconstruction sans erreur  $Y(z) = E(z)$  on doit avoir :

$$G_1(z)H_1(z) + G_2(z)H_2(z) = 2 \quad (4.18)$$

et

$$G_1(z)H_1(-z) + G_2(z)H_2(-z) = 0 \quad (4.19)$$

$G_2(z)$  et  $G_1(z)$  sont déterminées à partir des fonctions  $H_1(z)$  et  $H_2(z)$ . Pour plus de détail voir chapitre 7 de [3].

Les bancs de filtres à reconstruction parfaite sont aussi appelés bancs de filtres biorthogonaux. Ils sont étroitement liés à la théorie des ondelettes [68].

## 4.3 La transformée en ondelettes

### 4.3.1 La transformée continue en ondelettes

La transformée en ondelettes d'un signal  $x(\cdot)$  est définie comme la corrélation de  $x(\cdot)$  avec l'ondelette correspondante [74] :

$$A(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi^* \left( \frac{t-b}{a} \right) dt \quad (4.20)$$

Où,  $x(t)$  : le signal à analyser,

$A(a, b)$  : les coefficients de la transformée en ondelettes,

$\psi(t)$  : l'ondelette mère,

$(\cdot)^*$  : Opérateur conjugué,

$b$  : la localisation de l'ondelette dans le temps, le coefficient de translation,

$a$  : la largeur de l'ondelette ou de la fenêtre, le facteur d'échelle pour la dilatation.

La transformée inverse en ondelettes est définie par les coefficients de la transformée des ondelettes :

$$x(t) = \iint A(a, b)\psi_{a,b}(t)dbda \quad (4.21)$$

Où,  $x(t)$  : le signal à analyser,

$A(a, b)$  : les coefficients de la transformée en ondelettes,

$\psi(t)$  : l'ondelette mère,

$\psi_{a,b}(t)$  : est la dilatation et translation de l'ondelette mère  $\psi(t)$

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \quad (4.22)$$

### 4.3.2 La transformée discrète en ondelettes

Les coefficients de la DWT représentent la projection du signal par les fonctions de base générées par la translation et la dilatation de l'ondelette mère. La sélection de cette dernière détermine la représentation du signal. Les différentes ondelettes mères présentent des performances différentes dépendamment des caractéristiques du signal [49], [51].

#### 4.3.2.1 Bases d'ondelettes orthogonales

Pour numériser un signal continu  $x(t)$  les valeurs de dilatation et de translation  $a$  et  $b$  de l'ondelette même doivent être discrètes et sont représentées comme suit,

$$a = a_0^j \quad \text{et} \quad b = kb_0a_0^j \quad (4.23)$$

Avec,  $a_0 > 1$  et  $b_0 > 0$  et  $j = 0, 1, 2, \dots$ , et  $k$  constante.

En remplaçant (4.23) dans (4.22), l'ondelette mère dilatée et translatée s'écrit,

$$\psi_{j,k}(t) = \frac{1}{\sqrt{a_0^j}}\psi\left(\frac{t-kb_0a_0^j}{a_0^j}\right) \quad (4.24)$$

Ainsi, la transformée discrète en ondelettes est [49],

$$A_{j,k} = \frac{1}{\sqrt{a_0^j}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-kb_0a_0^j}{a_0^j}\right)dt \quad (4.25)$$

### 4.3.2.2 Transformée des ondelettes orthonormées et l'échelle dyadique

Le choix des paramètres  $a_0$  et  $b_0$  respectivement 2 et 1 est le choix commun pour l'ondelette discrète [49], [74]. Il est aussi connu sous le nom d'échelle dyadique.

Une ondelette avec l'échelle dyadique de l'équation (4.24) s'écrira:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - k2^j}{2^j}\right) \quad (4.26)$$

Ces ondelettes sont orthogonales et normalisées pour avoir une énergie unitaire.

$$\int_{-\infty}^{+\infty} \psi_{j,k}(t) \psi_{j',k'}(t) dt = \begin{cases} 1 & \text{si } j = j' \text{ et } k = k' \\ 0 & \text{ailleurs} \end{cases} \quad (4.27)$$

La transformée discrète en ondelettes d'un signal  $x(t)$  est :

$$A_{j,k} = \int_{-\infty}^{+\infty} x(t) \psi_{j,k}(t) dt \quad (4.28)$$

La reconstruction du signal original  $x'(t)$  suite au choix d'une base d'ondelette orthonormée est définie par,

$$x'(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} A_{j,k} \psi_{j,k}(t) \quad (4.29)$$

### 4.3.2.3 Fonction d'échelle et ondelette à support compact

La fonction d'échelle a la même forme que la fonction d'ondelette. Elle est définie par,

$$\varphi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \varphi\left(\frac{t - k2^j}{2^j}\right) \quad (4.30)$$

Avec la propriété :

$$\int_{-\infty}^{+\infty} \varphi_{0,0}(t) dt = 1 \quad (4.31)$$

La dilatation de la fonction d'échelle est donnée comme suit :

$$\varphi(t) = \sum_{u=-\infty}^{+\infty} c_u \varphi(2t - u) \quad (4.32)$$

Avec,  $\varphi(2t - u)$  la version contractée de  $\varphi(t)$

$$\sum_{u=-\infty}^{+\infty} c_u = 2 \quad (4.33)$$

$$\sum_{u=-\infty}^{+\infty} c_u c_{u+2v} = \begin{cases} 2 & \text{si } v = 0 \\ 0 & \text{ailleurs} \end{cases} \quad (4.34)$$

À l'échelle  $j+1$ , l'équation (4.30) deviendra,

$$\begin{aligned} \varphi_{j+1,k}(t) &= \frac{1}{\sqrt{2^{j+1}}} \varphi\left(\frac{t - k2^{j+1}}{2^{j+1}}\right) \\ &= \frac{1}{\sqrt{2^j}} \frac{1}{\sqrt{2}} \sum_{u=-\infty}^{+\infty} c_u \varphi\left(\frac{2t}{2^{j+1}} - 2k - u\right) \\ &= \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2^j}} \sum_{u=-\infty}^{+\infty} c_u \varphi\left(\frac{t}{2^j} - (2k + u)\right) \\ &= \frac{1}{\sqrt{2}} \sum_{u=-\infty}^{+\infty} c_u \varphi_{j,2k+u}(t) \end{aligned} \quad (4.35)$$

Cette équation montre que pour construire la fonction d'échelle  $\varphi_{m,n}(t)$ , il suffit de connaître la fonction d'échelle à  $\varphi_{m-1,2n+u}(t)$ .

La relation entre la fonction d'échelle et la fonction d'ondelettes est donnée par:

$$\psi(t) = \sum_{u=-\infty}^{+\infty} (-1)^u c_{N-1-u} \varphi(2t - u) \quad (4.36)$$

$N$  est le nombre des coefficients d'échelle.

$$\text{Si on pose : } b_u = (-1)^u c_{N-1-u} \quad (4.37)$$

L'équation (4.36) sera,

$$\psi(t) = \sum_{u=-\infty}^{+\infty} b_u \varphi(2t - u) \quad (4.38)$$

Avec,

$$\sum_{u=-\infty}^{+\infty} b_u = 0 \quad (4.39)$$

À partir de l'équation (4.26) et en suivant les mêmes étapes de l'équation (4.35), on obtient l'expression suivante :

$$\psi_{j+1,k}(t) = \frac{1}{\sqrt{2}} \sum_{u=-\infty}^{+\infty} b_u \psi_{j,2k+u}(t). \quad (4.40)$$

De cette expression, on peut aussi dire que pour trouver la fonction ondelette  $\psi_{m,n}(t)$  il suffit de connaître la fonction d'ondelette à  $\psi_{m-1,2n+u}(t)$ .

### 4.3.3 Algorithme Rapide de décomposition multi-résolution

L'algorithme rapide de calcul de la transformée en ondelettes a été introduit par Mallat en 1989 [55]. Cet algorithme est connu sous le nom de codeur à sous bande à deux canaux et il utilise des filtres miroirs quadratiques.

La décomposition du signal  $x(n)$  produit deux types de coefficients : *coefficients d'approximation* et *coefficients de détail*. Ces vecteurs sont obtenus par la convolution du signal d'entrée avec les filtres passe-bas et passe-haut respectivement suivis par une décimation, voir figure ci-dessous :

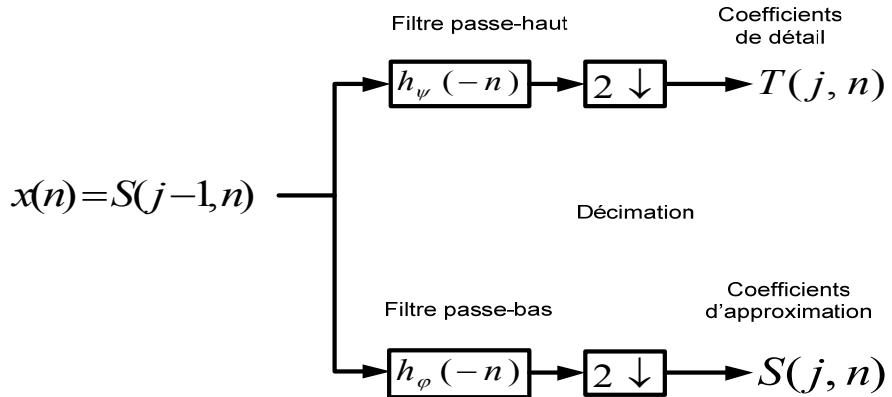


Figure 4. 11 : Décomposition du signal  $x(n)$

#### *Coefficients d'approximation :*

Les coefficients d'approximation  $S(j, k)$  sont déterminés à partir de la fonction d'échelle  $\varphi_{j,k}(t)$  et du signal  $x(t)$  suivant la formule suivante [49],

$$S(j, k) = \int_{-\infty}^{+\infty} x(t) \varphi_{j,k}(t) dt \quad (4.41)$$

En remplaçant  $\varphi_{j,k}(t)$  par l'équation (4.35), on aura :

$$S(j, k) = \int_{-\infty}^{+\infty} x(t) \frac{1}{\sqrt{2}} \sum_{u=-\infty}^{+\infty} c_u \varphi_{j-1, 2k+u}(t) dt \quad (4.42)$$

$$S(j, k) = \frac{1}{\sqrt{2}} \sum_{u=-\infty}^{+\infty} c_u \int_{-\infty}^{+\infty} x(t) \varphi_{j-1, 2k+u}(t) dt \quad (4.43)$$

$$S(j, k) = \frac{1}{\sqrt{2}} \sum_{u=-\infty}^{+\infty} c_u S(j-1, 2k+u) \quad (4.44)$$

Si on pose,  $2k+u = v$ , alors :

$$S(j, k) = \frac{1}{\sqrt{2}} \sum_{v=-\infty}^{+\infty} c_{v-2k} S(j-1, v) \quad (4.45)$$

Les coefficients d'approximation de l'itération  $j$  sont dépendants des coefficients d'approximation de l'itération précédente  $j-1$ .

### **Coefficients de détail :**

Aussi, les coefficients de détail  $T(j, k)$  sont déterminés à partir de la fonction d'ondelettes  $\psi_{j,k}(t)$  et du signal  $x(t)$  suivant la formule suivante, [49]

$$T(j, k) = \int_{-\infty}^{+\infty} x(t) \psi_{j,k}(t) dt \quad (4.46)$$

En remplaçant  $\psi_{j,k}(t)$  par l'équation (4.40), et en suivant les mêmes étapes de l'équation (4.42) à (4.45), on aura :

$$T(j, k) = \frac{1}{\sqrt{2}} \sum_{v=-\infty}^{+\infty} b_{v-2k} S(j-1, v) \quad (4.47)$$

Ici aussi, les coefficients de détail de l'itération  $j$  sont dépendants des coefficients de détail de l'itération précédente  $j-1$ .

À partir de ces deux équations (4.45) et (4.47), on voit bien que si les coefficients de l'approximation  $S(j, k)$  sont connus à l'instant de décomposition  $j=0$ , alors l'itération des équations ci-dessus de  $S(j, k)$  et  $T(j, k)$  généreront les coefficients d'ondelette d'approximation et

de détail à tous les instants de décomposition supérieures à  $j=0$ . Le calcul itératif de  $S(j,k)$  et de  $T(j,k)$  représente respectivement le filtrage passe-haut et le filtrage passe-bas du signal d'entrée.

Aussi, les coefficients  $\frac{1}{\sqrt{2}}c_u$  et  $\frac{1}{\sqrt{2}}b_u$  sont les coefficients des filtres passe-bas et passe-haut respectivement.

La figure 4.12 représente la décomposition en ondelettes du signal  $x(n)$  avec une itération égale à 3. On dénote le vecteur d'entrée  $X_0$ , les vecteurs d'approximation  $S_j$ , et les vecteurs de détails  $T_j$ .

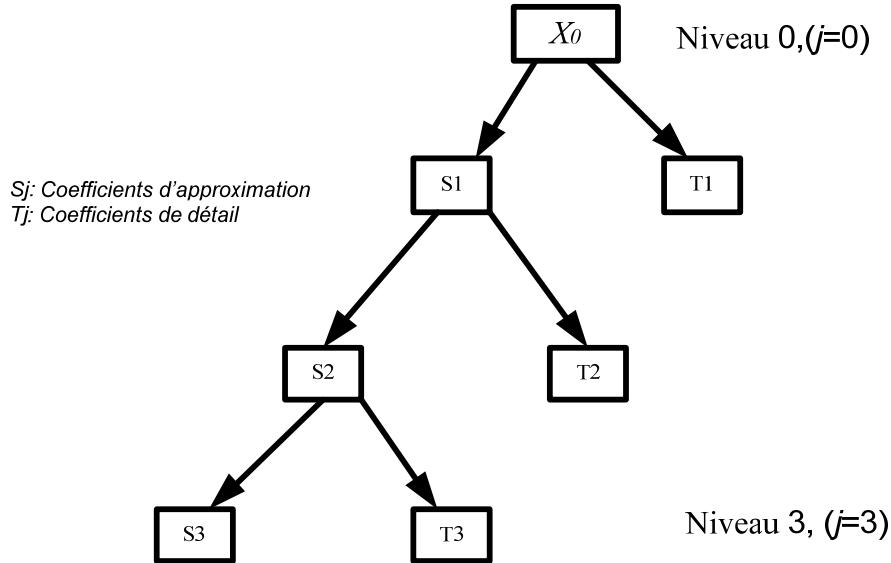


Figure 4. 12: Schéma bloc de la décomposition par ondelettes en trois itérations

Pour la reconstruction du signal, on procède de manière inverse, on additionne les coefficients d'approximation et de détail de niveau  $j$  ( $j=0,1,2,3$ ) puis on additionne les coefficients d'approximation de niveau  $j-1$  et ainsi de suite, comme montré ci-dessous :

$$S_3 + T_3 = S_2 \quad (4.48)$$

$$S_2 + T_2 = S_1 \quad (4.49)$$

$$S_1 + T_1 = X_0 \quad (4.50)$$

$$S_3 + T_3 + T_2 + T_1 = X_0 \quad (4.51)$$

L'opération de reconstruction du signal s'effectue en insérant des zéros puis on appliquant la convolution avec les filtres de reconstruction correspondants.

#### 4.3.4 Implémentation de la transformée discrète en ondelettes

L'implémentation de la transformée en ondelettes est équivalente à l'implémentation d'un filtre dont la réponse impulsionnelle est la fonction d'ondelette désirée. Donc la transformée en ondelettes peut être implémentée par des bancs de filtres [3], [4], [18], [44].

##### Phase d'analyse

La fonction  $h_\psi(n)$  et la fonction  $h_\varphi(n)$  de la figure 4.13 représentent le filtre passe-haut et le filtre passe-bas, respectivement. La phase d'analyse est le passage de la séquence  $x(n)$  par un filtre passe-bas et un filtre passe-haut suivi par une décimation d'un facteur de 2. La convolution du signal d'entrée avec  $h_\varphi(n)$  donne le signal d'approximation,

$$s_1(n) = [x(n) * h_\varphi(n)] \downarrow 2 \quad (4.52)$$

et la convolution du signal d'entrée avec  $h_\psi(n)$  donne le signal de détail,

$$t_1(n) = [x(n) * h_\psi(n)] \downarrow 2 \quad (4.53)$$

Si  $N$  est la longueur du signal  $x(n)$  et  $M$  la longueur du filtre  $h_1(n)$  et  $h_2(n)$  alors la longueur du signal  $s_1(n)$  et du signal  $t_1(n)$  est [53]:

$$L = \frac{N + M - 1}{2} \quad (4.54)$$

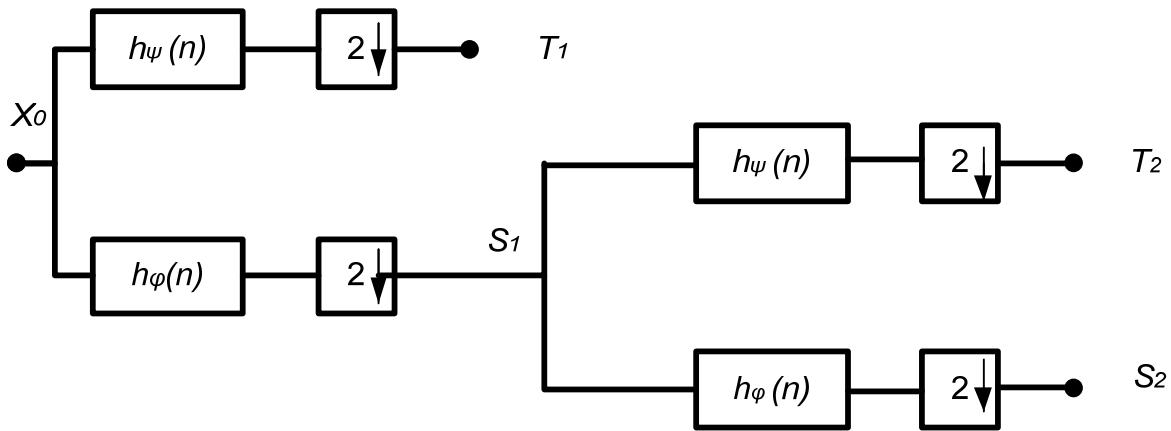


Figure 4. 13 : Phase d'analyse à deux itérations

$X_0$ ,  $S_j$  et  $T_j$  dénotent respectivement, le vecteur d'entrée, les vecteurs d'approximation, et les vecteurs de détails.

Si le nombre d'itérations est supérieur à un, le vecteur d'approximation  $S_1$  traversera un autre banc de deux filtres  $h_\psi(n)$  et  $h_\phi(n)$  similaire au premier banc de filtres pour produire les vecteurs  $S_2$  et  $T_2$  mais, le vecteur de détail  $T_1$  restera inchangé. Si nous avons une troisième itération, c'est aussi le vecteur d'approximation  $S_2$  qui traversera le banc de filtres  $h_\psi(n)$  et  $h_\phi(n)$  pour produire les vecteurs  $S_3$  et  $T_3$  alors que le vecteur  $S_2$  restera inchangé, et ainsi de suite.

### Phase de synthèse :

Les vecteurs  $S_2$ ,  $T_2$  et  $T_1$  de la figure 4.14 sont des vecteurs obtenus après la phase d'analyse (de décomposition) de la séquence  $x(n)$  de la figure 4.13. Une interpolation est appliquée sur chacun de ces vecteurs suivie par une convolution avec le filtre correspondant. Les vecteurs d'approximation et de détails obtenus au niveau  $j$  sont sommés pour former le vecteur d'approximation de niveau  $j-1$  et à son tour ce vecteur va être sommé avec le vecteur de détail de niveau  $j-1$  pour donner le vecteur d'approximation de niveau  $j-2$  et ainsi de suite jusqu'à l'obtention du vecteur de niveau 0, le vecteur  $Y$ .

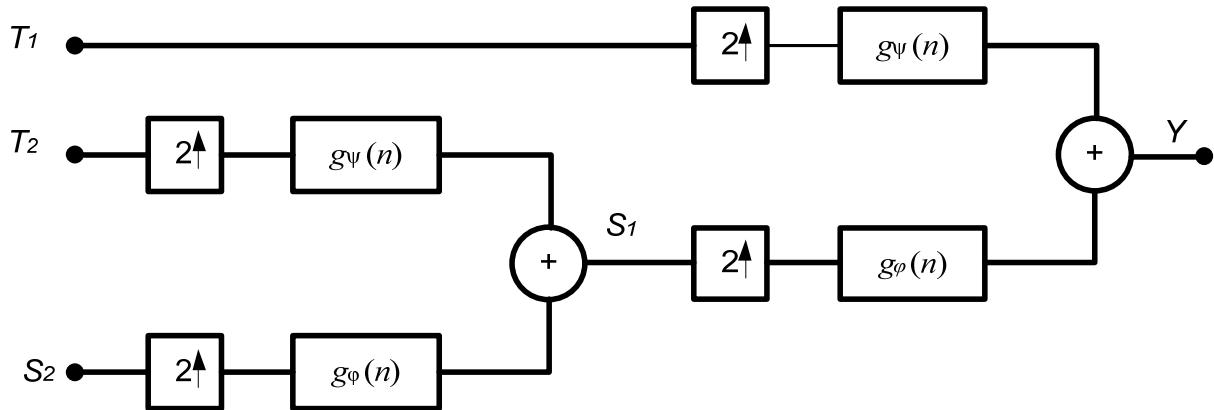


Figure 4.14 : Phase de synthèse à deux itérations

La sortie du signal  $y(n)$  peut être formulée comme suit

$$y(n) = [[s_1(n) \uparrow 2] * g_\phi(n)] + [[t_1(n) \uparrow 2] * g_\psi(n)] \quad (4.55)$$

Si le nombre d'itérations est égal à 2 comme dans figure 4.14, alors l'équation (4.55) devient

$$\begin{aligned}
 y(n) = & \left[ ([s_2(n) \uparrow 2] * g_\varphi(n)) + ([t_2(n) \uparrow 2] * g_\psi(n)) \right] \uparrow 2 * g_\varphi(n) \\
 & + \left[ [t_1(n) \uparrow 2] * g_\psi(n) \right]
 \end{aligned} \tag{4.56}$$

### Signal bidimensionnel :

Pour réaliser la transformation en ondelettes sur une image, on pourrait appliquer le même algorithme de décomposition multirésolution de Mallat, utilisé pour un signal monodimensionnel. Le calcul de la transformée discrète en ondelettes d'une image est la transformée discrète à une dimension des lignes suivie par la transformée discrète à une dimension des colonnes.

La figure 4.15 visualise la décomposition d'une image en quatre images, trois images de détails et une image d'approximation.

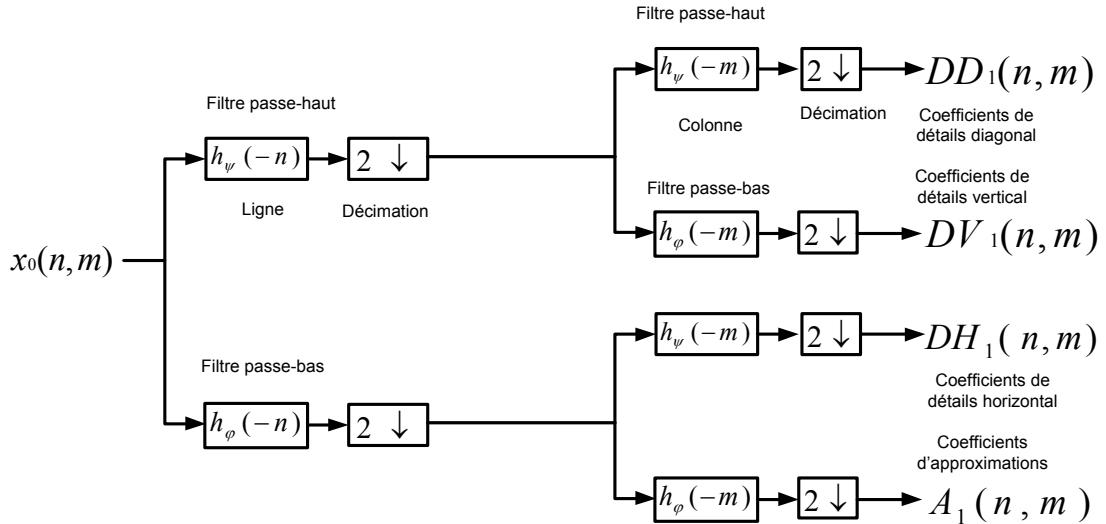


Figure 4.15 : Processus d'analyse multi-résolution à deux dimensions pour une itération

### Phase d'analyse

La séquence d'entrée de l'image est décrite sous forme matricielle  $\mathbf{S}_0$ . Nous avons besoin de trois bancs de filtres pour effectuer l'analyse de notre image à une itération. Le premier banc de filtres est utilisé pour la transformation niveau ligne et les deux autres bancs de filtres sont utilisés pour la transformation niveau colonne. Le filtre passe-bas  $h_\varphi(n)$  et le filtre passe-haut  $h_\psi(n)$  sont convolus avec la matrice d'entrée niveau ligne puis chaque sortie des deux filtres est suivie d'une décimation d'un facteur de 2. Les deux séquences résultantes niveaux colonnes vont aussi

être convoluées avec les deux autres bancs de filtres identiques au premier banc de filtres pour donner trois séquences de détails diagonal (DD<sub>1</sub>), vertical (DV<sub>1</sub>) et horizontal (DH<sub>1</sub>), et une séquence d'approximation (A<sub>1</sub>). C'est la transformation séparable niveau ligne colonne.

La convolution de la matrice niveau ligne avec  $h_\varphi(-n)$  engendre la séquence suivante,

$$[x_0(n, m) * h_\varphi(-n)] \downarrow 2 \quad (4.57)$$

La convolution de (4.57) niveau colonne avec  $h_\varphi(-n)$  engendre la séquence d'approximation

$$A1 = \left[ \left[ [x_0(n, m) * h_\varphi(-n)] \downarrow 2 \right] * h_\varphi(-m) \right] \downarrow 2 \quad (4.58)$$

La convolution de (4.57) niveau colonne avec  $h_\psi(-n)$  engendre la séquence de détail horizontal

$$DH1 = \left[ \left[ [x_0(n, m) * h_\varphi(-n)] \downarrow 2 \right] * h_\psi(-m) \right] \downarrow 2 \quad (4.59)$$

La convolution de la matrice niveau ligne avec  $h_\psi(-n)$  engendre la séquence suivante,

$$[x_0(n, m) * h_\psi(-n)] \downarrow 2 \quad (4.60)$$

La convolution de (4.60) niveau colonne avec  $h_\varphi(-n)$  engendre la séquence de détail vertical

$$DV1 = \left[ \left[ [x_0(n, m) * h_\psi(-n)] \downarrow 2 \right] * h_\varphi(-m) \right] \downarrow 2 \quad (4.61)$$

La convolution de (4.60) niveau colonne avec  $h_\psi(-n)$  engendre la séquence de détail diagonal

$$DD1 = \left[ \left[ [x_0(n, m) * h_\psi(-n)] \downarrow 2 \right] * h_\psi(-m) \right] \downarrow 2 \quad (4.62)$$

Pour une deuxième, troisième ou quatrième itération c'est uniquement les signaux d'approximation à deux dimensions qui sont décomposés en quatre autres signaux.

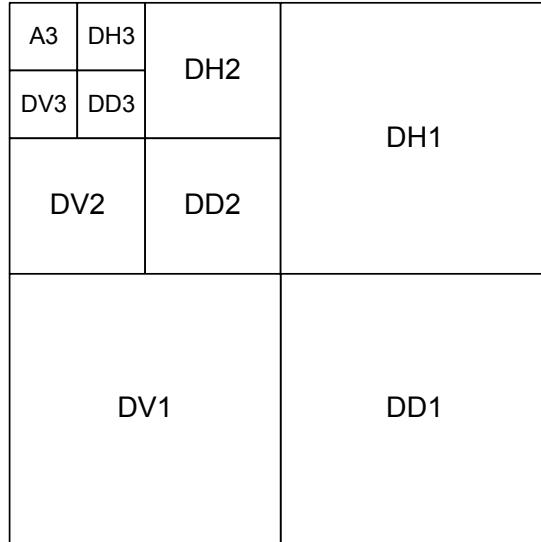


Figure 4. 16 : Décomposition d'une image en trois itérations

La figure 4.16 présente un schéma bloc de la décomposition d'une image en trois itérations. La première itération de l'image originale produit l'image DD1, DV1, DH1 et A1. La deuxième itération permet la décomposition de l'image A1 en DD2, DV2, DH2 et A2. Enfin la troisième itération permet la décomposition de l'image A2 en DD3, DV3, DH3 et A3.

#### **Exemple d'analyse par DWT :**

Dans la figure 4.17, nous avons trois itérations. À la première itération, l'image originale est décomposée en une image d'approximation  $A_1$  et trois images de détails. À la deuxième itération, l'image d'approximation  $A_1$  est décomposée en donnant l'image d'approximation  $A_2$  et trois images de détails. Enfin, l'image d'approximation  $A_2$  est décomposée pour donner l'image d'approximation  $A_3$  et trois images de détails d'où la décomposition de l'image en trois itérations produit une image d'approximation  $A_3$  et neuf images de détails.

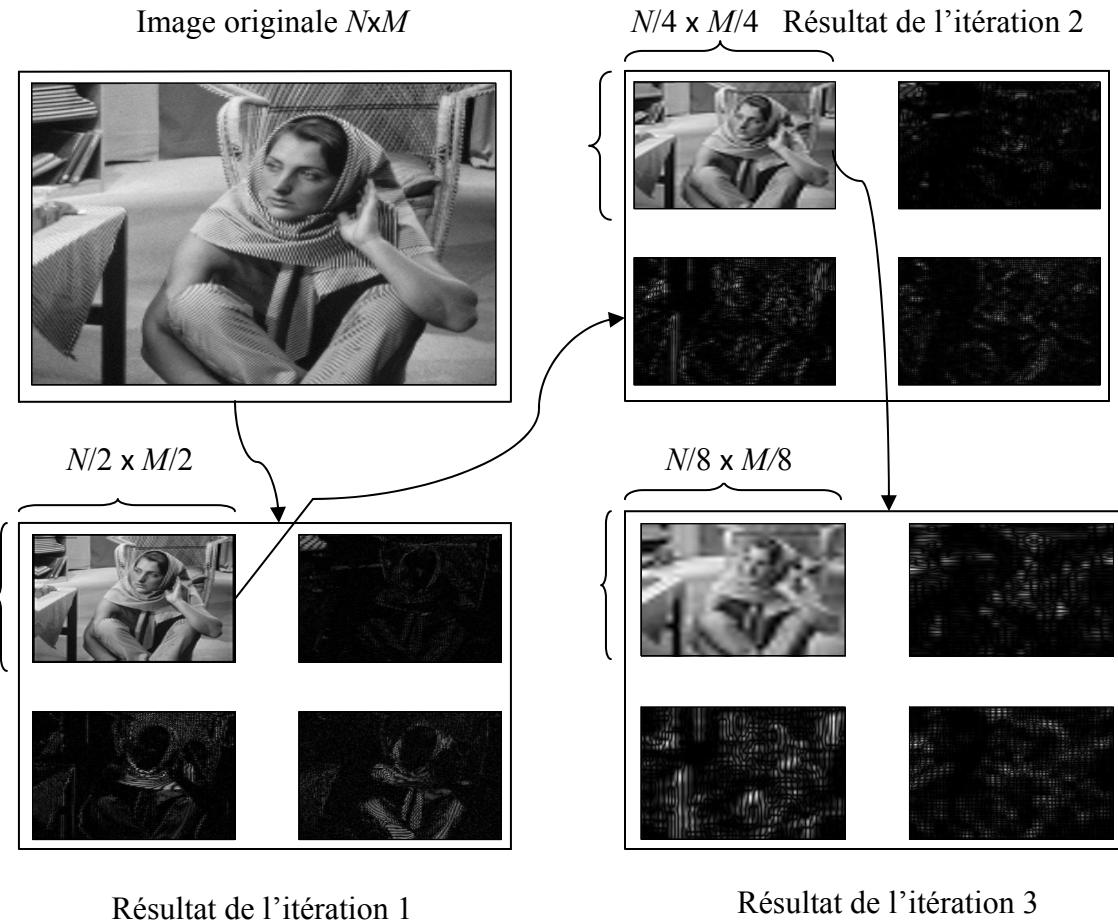


Figure 4. 17 : Analyse à trois niveaux de l'image *wbarb*.

### Phase de synthèse

La transformation inverse ou la phase de synthèse pour une itération nécessite trois bancs de filtres et trois sommateurs, comme présenté dans figure 4.18. Cette transformation est aussi séparable niveau ligne colonne.

On peut voir dans figure 4.18 que les quatre séquences de sortie de la phase d'analyse (la séquence d'approximation et les trois séquences de détails) sont interpolées d'un facteur de 2 puis convoluées aux filtres correspondant en parallèle. Les sorties des filtres sont sommées deux à deux, la séquence d'approximation avec la séquence de détail horizontal et la séquence de détail vertical avec la séquence de détail diagonal. À ce niveau, les deux séquences résultantes vont aussi être interpolées et convoluées aux filtres correspondants en parallèle et finalement sommées pour donner jour à l'image reconstruite  $Y(n,m)$ .

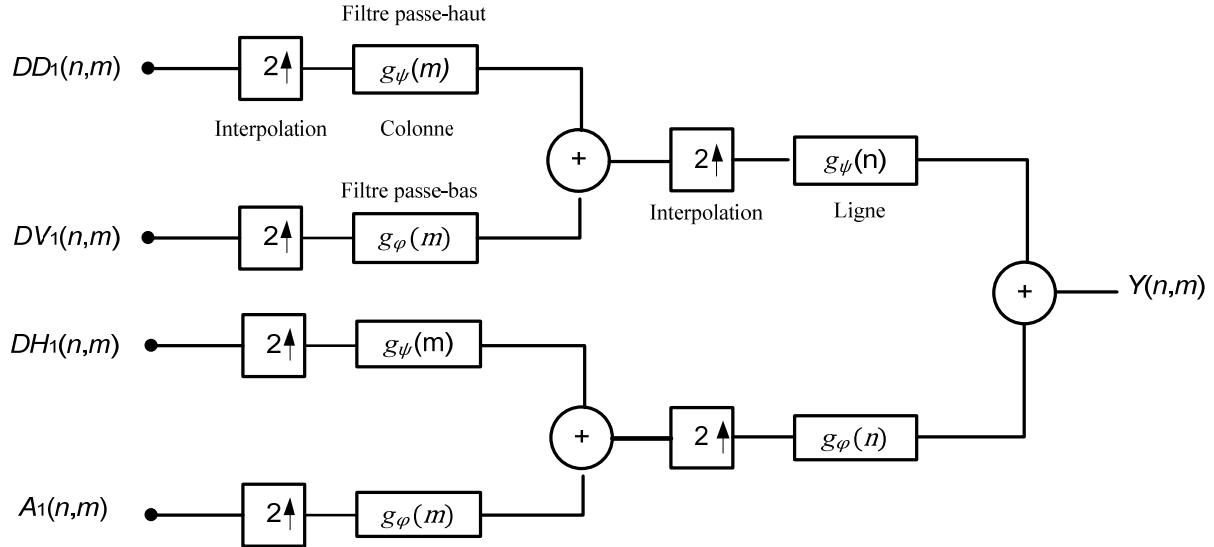


Figure 4. 18 : Processus de synthèse multi-résolution à deux dimensions pour une itération

La convolution de la séquence d'approximation niveau colonne avec le filtre  $g_\varphi(m)$  donne :

$$A1 \uparrow 2 * g_\varphi(m) \quad (4.63)$$

La convolution de la séquence de détail horizontal niveau colonne avec le filtre  $g_\psi(m)$  donne :

$$DH1 \uparrow 2 * g_\psi(m) \quad (4.64)$$

La convolution de la somme de (4.63) et (4.64) niveau ligne avec le filtre  $g_\varphi(n)$  donne :

$$[(A1 \uparrow 2 * g_\varphi(m)) + (DH1 \uparrow 2 * g_\psi(m))] \uparrow 2 * g_\varphi(n) \quad (4.65)$$

La convolution de la séquence de détail vertical niveau colonne avec le filtre  $g_\varphi(m)$  donne :

$$DV1 \uparrow 2 * g_\varphi(m) \quad (4.66)$$

La convolution de la séquence de détail diagonal niveau colonne avec le filtre  $g_\psi(m)$  donne :

$$DD1 \uparrow 2 * g_\psi(m) \quad (4.67)$$

La convolution de la somme de (4.66) et (4.67) niveau ligne avec le filtre  $g_\psi(n)$  donne :

$$[(DV1 \uparrow 2 * g_\varphi(m)) + (DD1 \uparrow 2 * g_\psi(m))] \uparrow 2 * g_\psi(n) \quad (4.68)$$

La somme de (4.65) et (4.68) produit le signal reconstruit  $s(n,m)$  :

$$s(n,m) = \left[ \left[ [A1 \uparrow 2 * g_\varphi(m)] + [DH1 \uparrow 2 * g_\psi(m)] \right] \uparrow 2 * g_\varphi(n) \right] \\ + \left[ \left[ [DV1 \uparrow 2 * g_\varphi(m)] + [DD1 \uparrow 2 * g_\psi(m)] \right] \uparrow 2 * g_\psi(n) \right] \quad (4.69)$$

Pour une itération supérieure à 1, les séquences d'approximation et de détails (horizontal, vertical et diagonal) de niveau  $j$  vont subir tout le processus de reconstruction pour une itération afin de donner le jour à la séquence d'approximation de niveau  $j-1$  qui va à son tour subir avec les signaux de détails de même niveau le même processus de reconstruction d'une itération jusqu'à épuisement des signaux de détails. L'image obtenue est l'image  $s(n,m)$ .

### 4.3.5 DWT dans la compression d'images

Pour la conception d'un système de compression d'images à base de la transformée discrète en ondelettes on doit prendre en considération les points suivants :

- *Image de test*
- *Fonction ondelette*
- *Ordre et longueur du filtre*
- *Nombre de décomposition*
- *Complexité de calcul*

#### 4.3.5.1 Type d'image pour le test

La qualité des systèmes de compression d'images est influencée par le contenu de l'image ou bien son activité spectrale [51]. Le choix d'une image pour les tests de compression d'images est un problème fondamental [51], [59], [60]. Les images doivent être classées selon leurs caractéristiques fréquentielles et spatiales.

### ***Mesures dans le domaine spatial (spatial frequency measure (SFM))***

Cette mesure permet d'avoir une idée globale de l'activité spectrale de l'image [70]:

$$SFM = \sqrt{R^2 + C^2} \quad (4.70)$$

Où,

$$R = \sqrt{\frac{1}{MN} \sum_{n=1}^N \sum_{m=2}^M [x(n, m) - x(n, m-1)]^2} \quad (4.71)$$

et

$$C = \sqrt{\frac{1}{MN} \sum_{m=1}^M \sum_{n=2}^N [x(n, m) - x(n-1, m)]^2} \quad (4.72)$$

Avec  $R$  Fréquences des lignes,  $C$  Fréquences des colonnes,  $M$  Nombre de lignes,  $N$  Nombre de colonnes, et  $x(n, m)$  Coefficient de la matrice d'image.

### ***Mesure dans le domaine fréquentiel (spectral activity measure (SAM))***

Cette mesure permet de voir la quantité de l'activité spectrale dans l'image [51]

$$SAM = \frac{A}{B} \quad (4.73)$$

Avec

$$A = \frac{1}{MN} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} |F(j, k)|^2 \quad (4.74)$$

$$B = \left[ \prod_{j=0}^{M-1} \prod_{k=0}^{N-1} |F(j, k)|^2 \right]^{\frac{1}{MN}} \quad (4.75)$$

$F(j, k)$  est la transformée de Fourier discrète de l'image.

Si  $SFM$  est grand alors l'image contient beaucoup de composantes de haute fréquence et si  $SAM$  est petit alors l'image est à très grande activité spectrale. Les images avec beaucoup d'activités spectrales sont difficiles à compresser.

La transformée discrète en cosinus permet aussi de visualiser la distribution de l'activité spectrale et sa fréquence. Dans le chapitre suivant nous allons utiliser la DCT pour le choix des images de test.

#### 4.3.5.2 Choix de la fonction ondelette

Le choix de l'ondelette pour la transformation doit être ajusté au contenu de l'image [51], [59], [60]. Une image avec une haute distribution spectrale est insensible aux choix des ondelettes.

Les propriétés de choix de la fonction de base ondelette sont :

- Support compact, pour avoir une implémentation efficace,
- Symétrie, pour éviter le déphasage dans la compression,
- Orthogonalité, pour avoir un algorithme rapide,
- Régularité et degré de linéarité, en relation avec l'ordre et la longueur du filtre.

#### 4.3.5.3 Ordre et longueur du filtre

La longueur du filtre dépend de l'ordre du filtre, mais la relation entre l'ordre et la longueur du filtre dépend de la famille des ondelettes. Nous avons cinq familles d'ondelettes à support compact, les ondelettes de Haar, les ondelettes de Daubechies, les ondelettes Coiflet, les ondelettes Symlet et les ondelettes biorthogonales.

Tableau 4. 1 : Répartition de la longueur des filtres dans la famille des ondelettes

Filtre/Ondelettes	Haar	Daubechies	Coiflet	biorthogonal	Symlet
Ordre	1	$N$	$N$	$N_d N_r$	$N$
Longueur	2	$2N$	$6N$	X	$2N$

Le tableau 4.1 présente la relation entre l'ordre et la longueur du filtre pour les cinq familles des ondelettes. On peut voir que l'ondelette de Haar est un cas spécial des ondelettes de Daubechies; pour un ordre du filtre  $N=1$ , la longueur du filtre est  $L=2$ . Pour la famille Daubechies et la famille Symlet, si l'ordre du filtre est  $N$  alors la longueur du filtre est  $2N$ . Pour la famille Coiflet, si l'ordre du filtre est  $N$  alors la longueur du filtre est  $L=6N$ . Pour la famille biortogonale, les ondelettes utilisées dans l'analyse du signal sont différentes des ondelettes utilisées dans la synthèse du même signal.

Aussi, on distingue les ondelettes par un nom qui représente la famille des ondelettes et un numéro qui indique l'ordre du filtre.

*Exemple :*

Coif3 : est l'ondelette de Coiflet avec l'ordre du filtre  $N=3$  donc sa longueur est  $L=18$ ,

Db2 : est l'ondelette de Daubechies avec l'ordre du filtre  $N=2$  donc sa longueur est  $L=4$ .

Pour la famille des ondelettes biorthogonales, le numéro qui accompagne le nom de l'ondelette indique l'ordre du filtre utilisé dans la décomposition du signal ( $N_d$ ) et l'ordre du filtre utilisé lors de la reconstruction ( $N_r$ ) respectivement.

*Exemple :*

Bior2.2 implique  $N_d=2$ , ( $L(\text{FPB})=5$  et  $L(\text{FPH})=3$ ),  $N_r=2$ , ( $L(\text{FPB})=3$  et  $L(\text{FPH})=5$ ).

Comme on peut le remarquer dans les fonctions des ondelettes biorthogonales, la longueur des filtres passe-bas (FPB) et la longueur des filtres passe-haut (FPH) utilisées dans la décomposition et la reconstruction sont différentes. Elles doivent être déterminées pour chaque type de filtre.

Un ordre de filtre élevé permet d'avoir une fonction large dans le domaine temporel et un degré de régularité élevé. Aussi ce type de filtre permet d'avoir une bonne localisation en fréquence avec une bonne régularité de l'ondelette. Par contre, un filtre avec un ordre faible permet d'avoir une très bonne localisation en temps et permet aussi de préserver une bonne quantité de l'information des contours [51].

La régularité étant importante dans la compression d'images par ondelettes, la conception de ces filtres nécessite un ordre élevé, donc une longueur élevée. Mais, ceci fera augmenter la complexité de calcul de DWT, alors un compromis est nécessaire entre la longueur du filtre, le degré de régularité et la complexité de calcul.

Dans la figure 4.19 on peut voir cinq exemples de fonctions d'échelles et d'ondelettes utilisées dans les phases d'analyses et de synthèses de la transformation en ondelettes.

La première colonne représente les fonctions d'échelles utilisées en phase d'analyse, la deuxième colonne représente les fonctions d'échelles utilisées en phase de synthèse, la troisième colonne représente les fonctions ondelettes utilisées en phase d'analyse et la quatrième colonne représente les fonctions d'ondelettes utilisées en phase de synthèse.

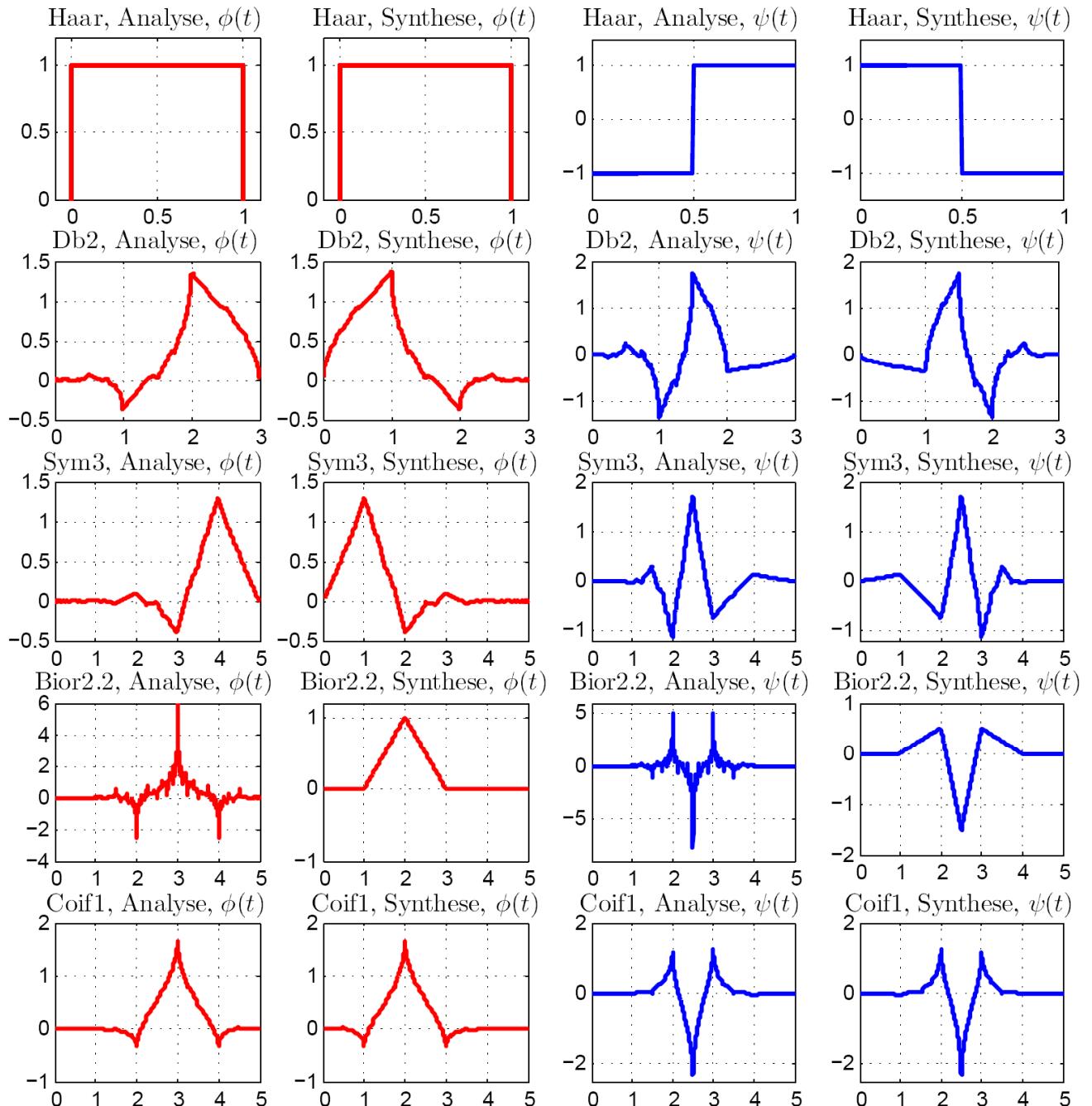


Figure 4.19 : Fonctions d'ondelettes avec leurs fonctions d'échelles correspondantes utilisées dans la phase d'analyse et la phase de reconstruction des ondelettes Haar, Db2, Sym3, Bior2.2 et Coif1.

#### 4.3.5.4 Nombre de décompositions

La qualité de compression est déterminée par le nombre de décompositions de l'image. Le nombre optimal de décompositions de l'image permet d'avoir une valeur du PSNR la plus élevée dans une large bande de taux de compression pour un certain ordre de filtre [51]. Après avoir décomposé l'image et récupéré les coefficients de la transformation en ondelettes, la compression peut être achevée par l'élimination des coefficients inférieurs à un seuil donné.

Aussi, plus le nombre d'itérations est grand plus le nombre d'opérations arithmétiques est grand. Dans notre travail nous avons limité le nombre d'itérations à 3.

#### 4.3.5.5 Complexité de calcul

Pour une image de  $N \times N$  en décomposition dyadique, la complexité de calcul est approximativement [59]:

$$C = 16 \times N^2 \times L \times \frac{(1 - 4^{-j})}{3} \quad (4.76)$$

Où  $L$  la longueur du filtre

$j$  le nombre de décompositions.

## CHAPITRE 5 ÉVALUATION D’EFFICACITÉ DE QUELQUES ONDELETTES À SUPPORT COMPACT

### 5.1 Introduction

Le but de notre travail est d’effectuer la compression de données d’images avec la technique de la transformée en ondelettes (DWT) tout en gardant la qualité essentielle de l’image originale, c'est-à-dire, avoir un bon compromis entre PSNR et le taux de compression. Pour cela, nous procéderons à l’évaluation de l’image de test en premier lieu puis, nous entamerons le processus du choix de l’ondelette. Le choix de ces étapes est dicté par le fait que l’ondelette est primordial dans la performance du codage de la compression d’images [51], [60] et que l’ondelette de base la mieux recommandée pour le codage dépend des spécificités des images à coder [60].

Le squelette de notre algorithme est décrit à la page 109.

### 5.2 Choix des images de test

Le choix des images de test est un problème fondamental lors de l’évaluation des systèmes de compression [60]. Comme ce choix est basé sur la variation de l’activité spectrale, la transformée discrète en cosinus (DCT) est utilisée pour l’évaluation de ces images. Nous vous rappelons que les points rouges et jaunes de la DCT d’une image représentent les coefficients de la DCT, que l’augmentation des fréquences spatiales horizontales est montrée par l’augmentation des points rouges et jaunes du haut de la DCT de l’image de gauche à droite et que l’augmentation des fréquences spatiales verticales est montrée par l’augmentation des points rouges et jaunes de haut au bas de la DCT de l’image. Les points rouges indiquent que l’intensité des coefficients est au maximum et les points jaunes indiquent le détail spatial.

Quatre images sont prises pour fins d’évaluation et sont, *Rami*, *Rachis-cervical*, *Lena* et *Écographie*.

La figure 5.1, nous permet de voir la DCT de l’image *Rami*. On voit en haut, sur le côté gauche de l’image un champ de points rouges et jaunes, ce sont les coefficients de la transformation DCT. Les points rouges sont bien concentrés dans les basses fréquences de l’image, c’est la

concentration de l'énergie de l'image. La figure 5.1.a nous montre une variation faible de l'intensité sur une très large surface, aussi le détail spatial est faible.

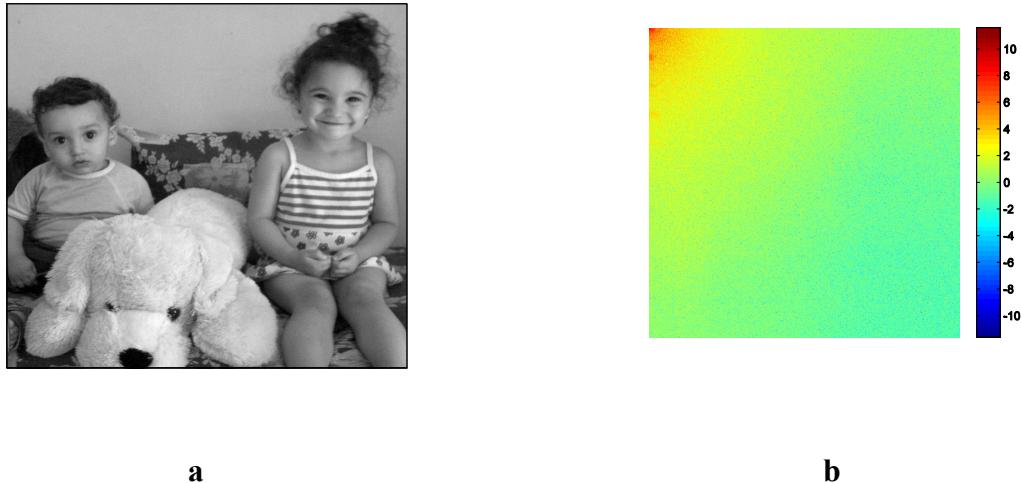


Figure 5. 1 : a) image *Rami* b) DCT de l'image *Rami*

La DCT de l'image *Lena* est visualisée dans la figure 5.2. On remarque un champ de points rouges et jaunes plus important que dans l'image *Rami*, toujours localisé en haut de l'image et sur le côté gauche. La variation de l'intensité est faible sur une large surface, elle est plus importante que dans l'image *Rami*. On peut dire que le détail spatial est assez faible, l'énergie est concentrée dans les basses fréquences et très peu dans les hautes fréquences.

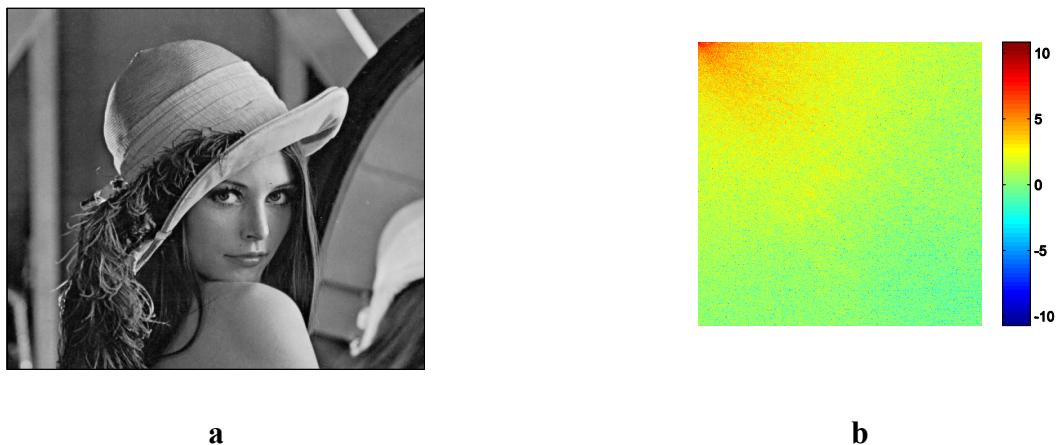


Figure 5. 2: a) image *Lena*, b) DCT de l'image *Lena*

La DCT de l'image *Rachis-cervical* est représentée par la figure 5.3. La variation de l'intensité est faible sur une large surface de l'image et présente une variation de l'intensité du côté des

fréquences spatiales horizontales et verticales, en comparaison avec la figure 5.2. En résumé, le détail spatial est assez faible et la concentration de l'énergie est localisée dans les basses fréquences et peu dans les hautes fréquences.

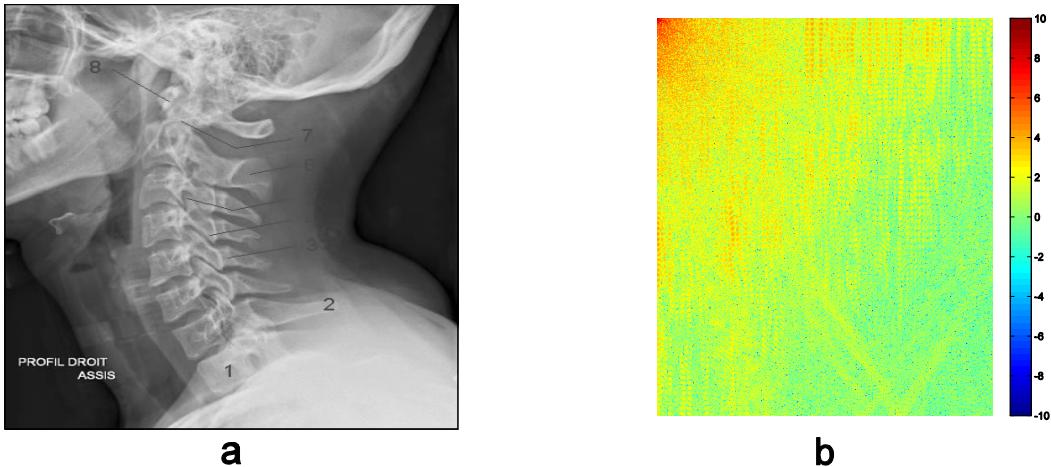


Figure 5. 3: Image *rachis\_cervical*, b) DCT de l'image *rachis\_cervical*

La DCT de l'image *Écographie* qu'on peut voir dans figure. 5.4, montre une variation de l'intensité plus importante que dans la figure 5.2 et la figure 5.3. On observe aussi, plus de coefficients de DCT du côté des hautes fréquences spatiales verticales et horizontales. Le détail spatial est assez important.

L'énergie de l'image est concentrée dans les basses fréquences mais elle est importante dans les hautes fréquences et précisément dans les hautes fréquences verticales.

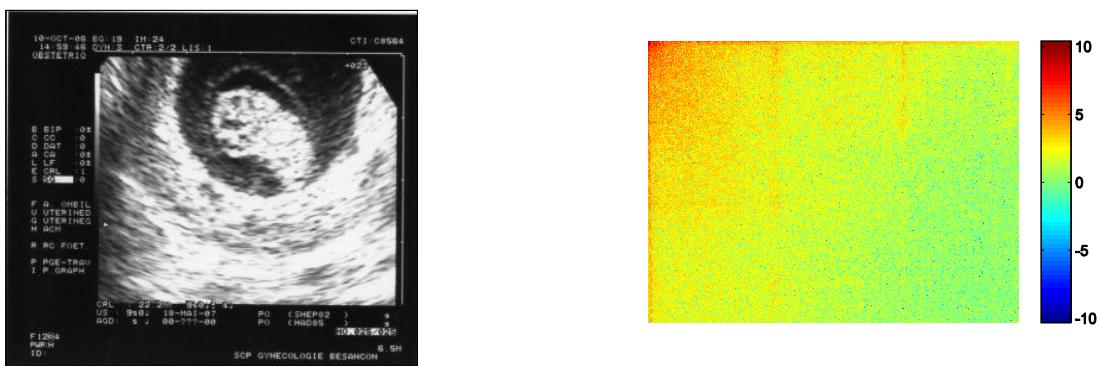


Figure 5. 4: a) image *Écographie* b) DCT de l'image *Écographie*

Selon [51], le meilleur choix de la fonction d'ondelette s'établit en utilisant des images avec des variations spectrales modérées; alors les images Rami, Lena et Rachis cervicale sont choisies pour nos tests. Ces trois images sont de tailles différentes à savoir 841x841, 512x512 et 400x474 respectivement.

### 5.3 Choix des ondelettes

Les trois images choisies pour le test sont :

- *Rami* et *Rachis-cervical*, sont des images en composantes RVB. Chacune des composantes de ces images est codée sur 8 bits. Les versions niveaux de gris sont générées en calculant la moyenne des trois composantes. Ces images sont de dimensions respectives 841x841 et 400x474.
- *Lena* est une image en niveau de gris de dimension 512x512. La composante de cette image est codée sur 8 bits.

Les ondelettes choisies pour le test sont classées par famille. Chaque famille a un ensemble d'ondelettes dont le numéro qui suit le nom, indique l'ordre des filtres utilisé pour leurs conceptions.

Les 26 ondelettes utilisées dans le test sont :

Ondelettes biorthogonales: Bior2.2, Bior3.3, Bior4.4, Bior5.5 et Bior6.8

Ondelettes de Daubechies: Db2, Db3, Db4, Db5, Db6, Db7, Db8, Db9 et Db10

Ondelettes de Symlets: Sym2, Sym3, Sym4, Sym5, Sym6, Sym7 et Sym8

Ondelettes de Coiflets : Coif1, Coif2, Coif3, Coif4 et Coif5

Ces ondelettes sont à support compact ce qui correspond à des filtres à réponse impulsionnelle finie (RIF).

Pour l'évaluation de la qualité des images après compression, on mesure la qualité de reconstruction de l'image par rapport à l'image originale (PSNR : rapport signal sur bruit) en fonction du taux de compression (TC).

Le PSNR permet de fournir une mesure objective de la distorsion introduite par la compression. Soient  $x(i,j)$  et  $y(i,j)$  les images originales et reconstruites de même taille  $M \times N$ . Le calcul du PSNR est défini par le calcul de l'erreur quadratique moyenne (*MSE* : Mean Square Error) :

$$MSE = \frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M [x(i,j) - y(i,j)]^2 \quad (5.1)$$

et

$$PSNR = 10 \log_{10} \left( \frac{Pic^2}{MSE} \right) \quad (5.2)$$

Le *Pic* est la valeur maximale possible du pixel d'image. Comme les pixels des images sont codés sur 8 bits, alors  $Pic = 2^8 - 1 = 255$ . L'unité du PSNR est le décibel (dB).

Plus le PSNR est important plus les images sont semblables. Aussi, pour une bonne reconstruction de l'image le PSNR doit être égal ou supérieur à 30 dB [44].

Notre méthode consiste à fixer le taux de compression (TC) et à chercher le seuil qui permet d'atteindre ce taux puis on calcule le PSNR. Dans notre travail nous avons choisi un seuil global où les coefficients inférieurs à ce dernier sont mis à zéro pour atteindre notre taux de compression souhaité.

Le TC représente le pourcentage du nombre de coefficients de l'image originale non utilisé dans la reconstruction de l'image. Il est calculé en évaluant le nombre de zéros après transformation et quantification.

### 5.3.1 Compression par les ondelettes biorthogonales

Le tableau 5.1 représente les résultats de compression de trois images par la transformée en ondelettes, en suivant l'algorithme de compression de la page 108. Les ondelettes utilisées pour la compression d'images sont des ondelettes de la famille biorthogonale.

Pour différents ordres d'ondelettes biorthogonales, nous avons différents PSNR. Dans le tableau 5.1, la compression de l'image *Rami* par les différentes ondelettes biorthogonales, visualise une variation du PSNR de 45.8365 dB à 47.9637 dB; alors que la compression de l'image *Lena* par les mêmes ondelettes donne une variation du PSNR de 34.2455 dB à

39.1683dB. L'image *Lena* présente un PSNR très faible en comparaison avec le PSNR de l'image *Rami*. Ceci est dû au fait que l'image *Lena* présente plus d'activité spectrale en comparaison avec l'image *Rami*. La variation spectrale de ces images est montrée aux figures 5.1 et 5.2. On peut conclure que plus l'activité spectrale est élevée plus le PSNR est faible.

Tableau 5. 1 : PSNR et TC de trois images compressées par des ondelettes biorthogonales

Wavelet	Rachis-cervical		Rami		Lena	
	PSNR	TC%	PSNR	TC%	PSNR	TC%
<u>Bior2.2</u>	36.4276	93.01	45.8365	93.41	34.4254	93.12
<u>Bior3.3</u>	34.8484	92.70	45.9596	93.23	34.2455	92.86
<u>Bior3.9</u>	34.8641	90.99	46.4059	92.35	34.4674	91.41
<u>Bior4.4</u>	40.7106	91.94	47.9637	92.68	39.1683	91.42
Bior5.5	38.88	91.99	46.9684	92.87	35.8285	92.28
Bior6.8	46.2444	87.37	47.6206	92.20	35.8069	91.57

On remarque aussi dans la figure 5.5 que lorsque l'ordre de l'ondelette augmente, les courbes des deux images ont presque la même forme. Ceci s'explique par l'absence de la variation des fréquences horizontales et verticales dans les deux images.

La courbe de compression de l'image *Rachis-cervical* par les différentes ondelettes biorthogonales est située entre les deux courbes de compression des images *Rami* et *Lena*. L'activité spectrale en diagonale de l'image *Rachis-cervical* est moindre que celle de l'image *Lena* mais plus importante que celle de l'image *Rami*. De plus, nous avons quelques variations de l'activité spectrale au niveau des fréquences verticales et horizontales. Ces variations de l'activité spectrale du coté des fréquences horizontales et verticales ne sont pas observées dans les images de *Rami* et de *Lena*, comme on peut le voir dans les figures 5.1 à 5.3.

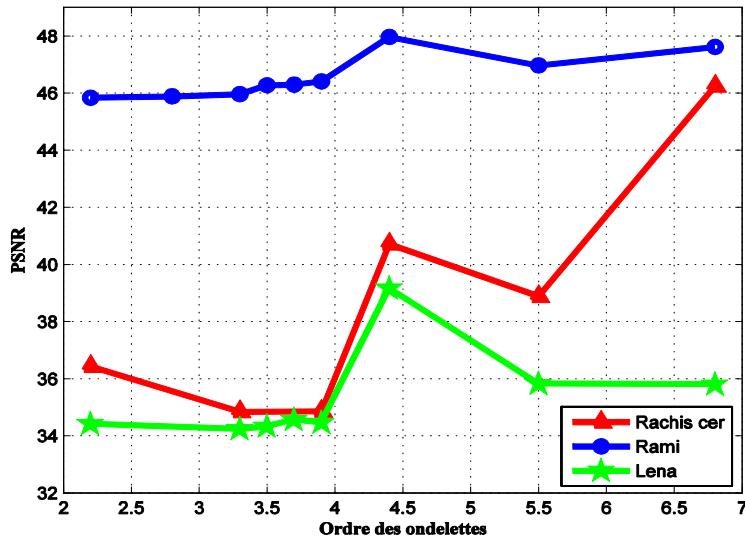


Figure 5. 5: PSNR de la compression de trois images par des ondelettes biorthogonales

Aussi, on remarque que le meilleur PSNR pour les deux images *Rami* et *Lena* est obtenu par l'ondelette Bior4.4, alors que pour l'image *Rachis-cervical* c'est l'ondelette Bior6.8 qui donne le meilleur PSNR suivi par l'ondelette Bior4.4 en deuxième position.

La figure 5.6, montre que pour les différents ordres des ondelettes biorthogonales, la courbe de l'image *Rami* présente un taux de compression meilleur que celui donné par les courbes des images *Lena* et *Rachis-cervical*. La variation du taux de compression en fonction de l'ordre des filtres des ondelettes biorthogonales est presque de même forme pour les trois images.

Le taux de compression de l'image *Rami* par les différentes ondelettes de la famille biorthogonales varie de 92.2% à 93.42%, ce qui représente un très bon taux de compression. L'image *Lena* aussi présente un très bon taux de compression qui varie de 91.57% à 93.12%. L'image *Rami* présente un meilleur taux de compression dans l'ensemble puisqu'elle présente peu d'activité spectrale en comparaison avec les autres images.

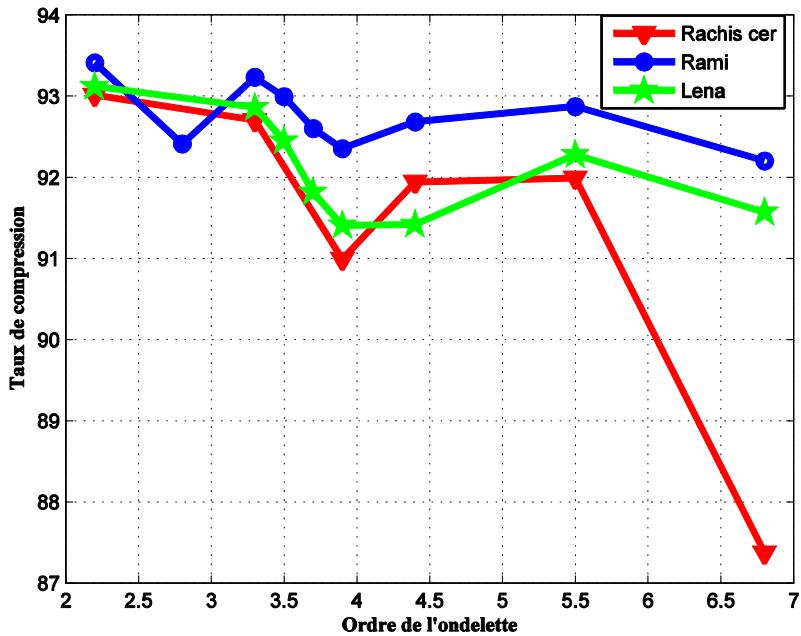


Figure 5. 6 : TC de la compression de trois images par des ondelettes biorthogonales

Le taux de compression de l'image *Rachis-cervical* en fonction de l'ordre des ondelettes biorthogonales varie de 87.37% à 93.01%. Le taux de compression de l'image *Rachis-cervical* est inférieur au taux de compression de l'image *Lena* sauf pour l'ondelette Bior4.4. La figure 5.6 montre clairement que le taux de compression le plus élevé des trois images est donné par l'ondelette Bior2.2.

### 5.3.2 Compression par les ondelettes Daubechies

Le tableau 5.2 présente l'évaluation du PSNR et du TC de la compression de trois images en fonction de différents ordres d'ondelette de Daubechies. On peut observer dans le tableau que la valeur du PSNR la plus élevée de la compression des trois images est toujours donnée par l'ondelette Db2. Les PSNR de la compression par l'ondelette Db2 des images *Rami*, *Rachis-cervical* et *Lena* sont 51.1979 dB, 44.03 dB et 43.9693 dB, respectivement.

Tableau 5. 2 : PSNR et TC de trois images compressées par des ondelettes Daubechies

Wavelet	Rachis-cervical		Rami		Lena	
	PSNR	TC%	PSNR	TC%	PSNR	TC%
<u>Db2</u>	44.03	90.07	51.1979	87.89	43.9693	84.99
Db3	41.7115	91.98	45.9744	93.40	38.0298	92.12
Db4	36.6691	92.64	46.1701	93.22	41.5221	89.33
Db5	38.3203	92.29	46.2474	93.04	36.2357	92.47
Db6	37.4857	92.11	46.3813	92.96	36.0077	92.24
Db7	38.7175	91.62	46.4468	92.78	34.6953	92.09
Db10	38.6479	90.78	46.5050	92.33	34.5815	91.37

Le PSNR de la compression de l'image *Rami* par les différentes ondelettes de Daubechies autres que Db2 varie de 45.9744 dB à 46.50 dB. Le PSNR pour les différentes ondelettes utilisées est intéressant, cette variation est presque uniforme. On peut expliquer ceci par le fait que l'image *Rami* ne présente pas beaucoup de variations de l'activité spectrale.

Comme l'activité spectrale de l'image *Lena* est plus importante que celle de l'image *Rami*, alors il est logique que pour les différentes ondelettes de Daubechies, les valeurs du PSNR de l'image *Lena* soient inférieures à ceux de l'image *Rami*. Le PSNR de l'image *Lena* varie de 34.5815 dB à 43.9693 dB et le PSNR de l'image *Rachis-cervical* varie de 36.6691 dB à 44.03 dB.

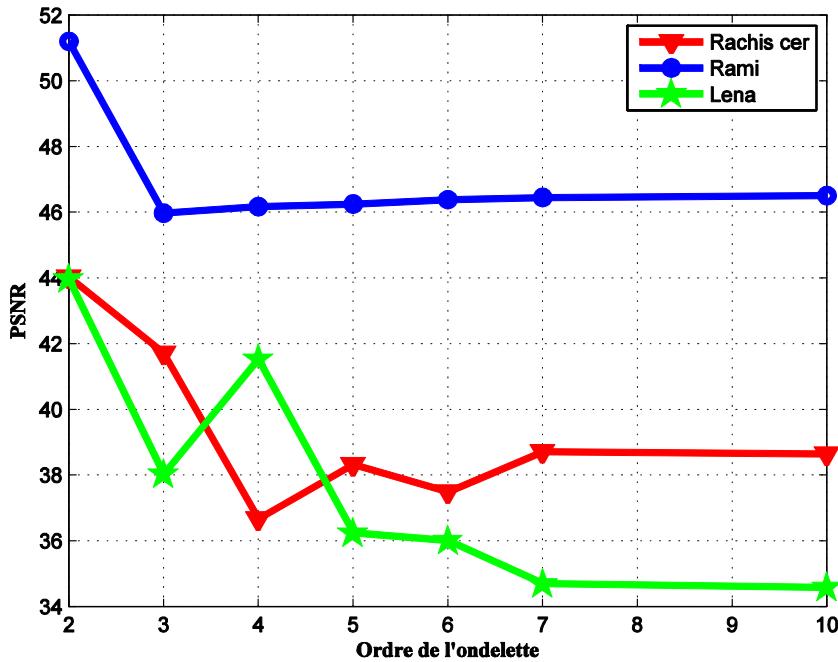


Figure 5. 7 : PSNR de la compression de trois images par des ondelettes de Daubechies

La figure 5.8 montre clairement que l'ondelette Db2 donne le plus bas taux de compression pour les trois images. Le taux de compression le plus élevé pour les images *Rami*, *Lena* et *Rachis-cervical* est donné par les ondelettes Db3, Db5 et Db4, respectivement. On ne peut pas conclure laquelle des ondelettes de Daubechies donne le meilleur taux de compression.

La courbe de l'image *Rami* montre sa supériorité aux deux autres courbes. Le phénomène est toujours justifié par le fait que l'image *Rami* ne présente pas beaucoup d'activités spectrales. Si on exclut l'ondelette Db2, le taux de compression en fonction des autres ondelettes pour les trois images est presque uniforme. Pour les différentes ondelettes autres que l'ondelette Db2, le taux de compression de l'image *Rami* varie de 92.33% à 93.40%, le taux de compression de l'image *Rachis-cervical* varie de 90.78% à 92.64% et le taux de compression de l'image *Lena* varie de 89.33% à 92.47%. Dans l'ensemble, les trois images présentent un très bon taux de compression pour les différentes ondelettes de Daubechies.

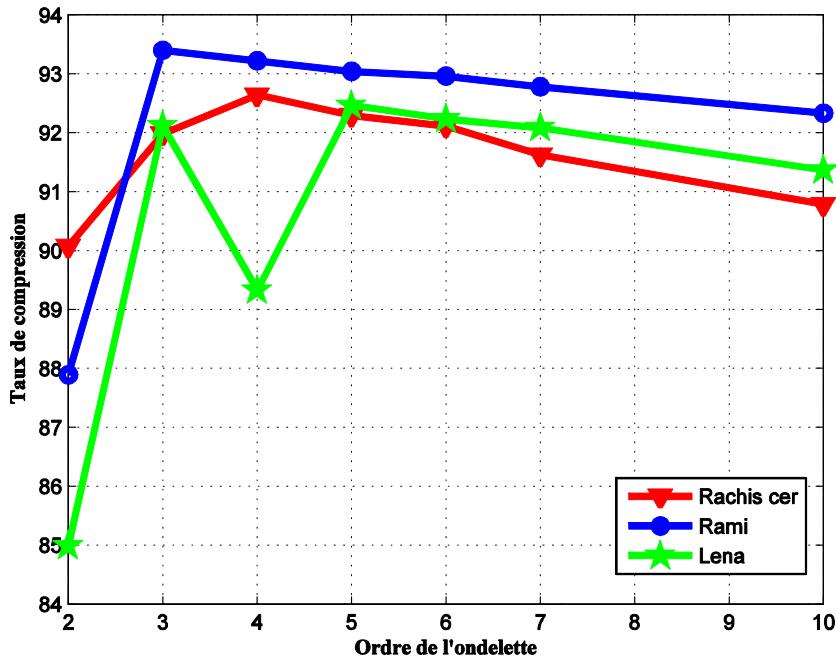


Figure 5. 8 : TC de la compression de trois images par des ondelettes de Daubechies.

### 5.3.3 Compression par les ondelettes Symlets

On observant le tableau 5.2 et le tableau 5.3, on remarque que les valeurs du PSNR et de TC obtenues par la compression des trois images par les ondelettes Db2 et Sym2 sont identiques. Aussi, la compression de ces images par les ondelettes Db3 et Sym3 produit des valeurs de PSNR et de TC identiques. Donc, les ondelettes Db2, Sym2, Db3 et Sym3 sont identiques deux à deux.

Dans le tableau 5.3, le PSNR le plus élevé pour les images *Rami* et *Lena* est donné par l'ondelette Sym2. Concernant l'image *Rachis-cervical*, le PSNR le plus élevé est donné par l'ondelette Sym7. Si on exclut l'ondelette Sym2, le PSNR de l'image *Rami* varie de 45.9744 dB à 46.6518dB et le PSNR de l'image *Lena* varie de 34.7696 dB à 41.2303 dB. Comme le PSNR le plus élevé de l'image *Rachis-cervical* est donné par l'ondelette Sym7 soit 53.7991 dB, alors le PSNR de cette image compressée par les ondelettes de Symlet autres que Sym7, varie de 38.2581dB à 45.8073 dB.

Tableau 5. 3 : PSNR et TC de trois images compressées par des ondelettes Symlet

Wavelet	Rachis-cervical		Rami		Lena	
	PSNR	TC%	PSNR	TC%	PSNR	TC%
<u>Sym2</u>	44.03	90.07	51.1979	87.89	43.9693	84.99
<u>Sym3</u>	41.7115	91.98	45.9744	93.40	38.0298	92.12
<u>Sym4</u>	40.3677	92.18	46.6518	93.16	41.2303	89.63
<u>Sym5</u>	45.8073	88.53	46.2101	93.04	34.7696	92.69
<u>Sym6</u>	38.2581	92.06	46.4751	92.95	36.4984	92.14
<u>Sym7</u>	53.7991	74.80	46.4001	92.79	34.7732	92.09
<u>Sym8</u>	37.6197	91.38	46.4786	92.59	40.2555	89.81

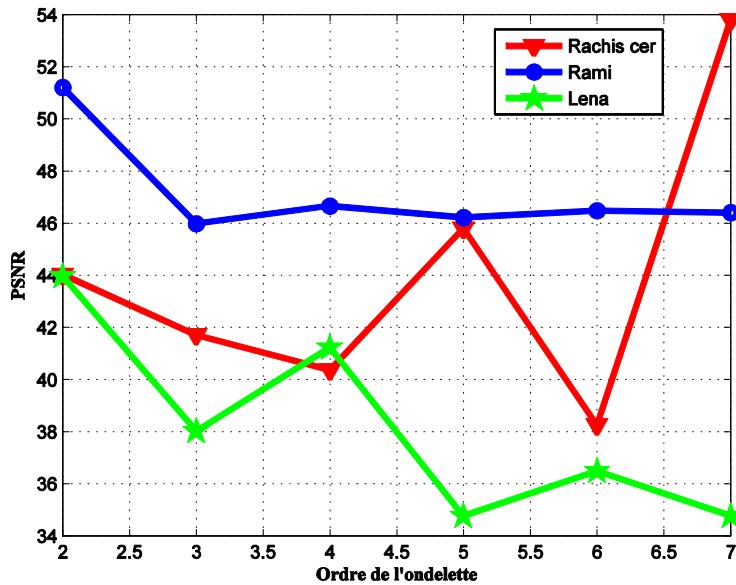


Figure 5.9: PSNR de la compression de trois images par des ondelettes de Symlets

Dans la figure 5.9, on voit toujours que le PSNR de la courbe *Rami* est supérieur aux PSNR des deux autres courbes, sauf pour l'ondelette Sym7 et que le PSNR de la courbe *Rachis-cervical* est supérieur au PSNR de la courbe *Lena* sauf pour l'ondelette Sym4.

La figure 5.10 présente le taux de compression des trois images par les ondelettes Symlets. Le taux de compression de l'image *Rami* suit presque une ligne droite, il varie de 87.89 % à 93.4 %. Le taux le plus élevé est donné par l'ondelette Sym3. Le taux de compression de l'image *Rachis-cervical* varie de 88.53% à 92.18% pour les ondelettes Sym2 à Sym6. Pour l'ondelette Sym7 le taux de compression est 74.80%. De même pour l'image *Lena*, en excluant la valeur de l'ondelette Sym2, le taux de compression varie de 89.81% à 92.69%. On remarque alors, que la courbe *Lena* a presque la même forme que la courbe *Rami* mais avec un taux de compression plus bas. Le taux de compression pour l'ondelette Sym2 de l'image *Lena* est de 84.99%. Dans l'ensemble le taux de compression par les ondelettes Symlets pour les trois images est bon.

Le meilleur taux de compression des images *Rami*, *Rachis-cervical* et *Lena* est donné par les ondelettes Sym3, Sym4 et Sym5 successivement. On ne peut déduire l'ondelette qui donnera le meilleur taux de compression.

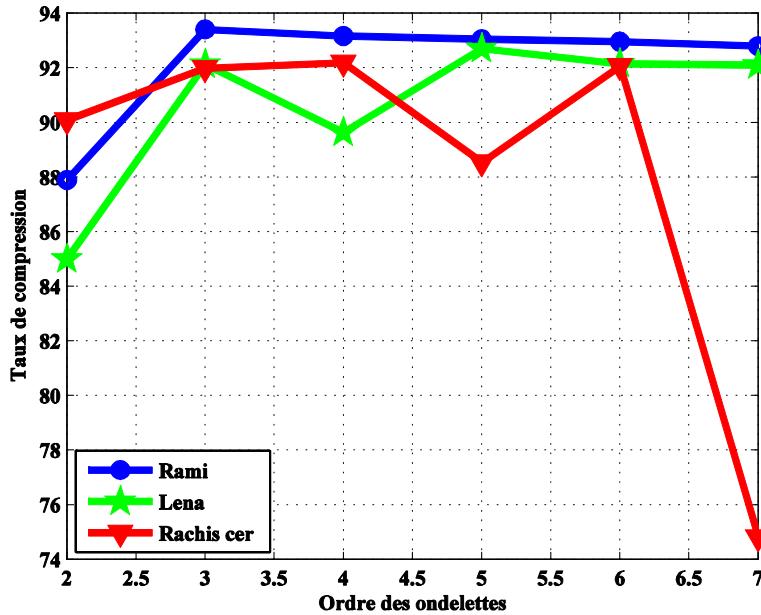


Figure 5. 10: TC de la compression de trois images par des ondelettes de Symlets

En comparant la figure. 5.7 et la figure. 5.9, on remarque que les courbes *Rami* et *Lena* suivent presque la même forme de courbure et donnent presque pour les ondelettes de même ordre, les mêmes valeurs du PSNR dans les deux figures. La courbe *Rachis-cervical* présente la même forme de courbure dans les deux figures mais, produit des valeurs différentes du PSNR pour les ondelettes d'ordre 4, 5 et 7. De même, la figure 5.8 et la figure 5.10 montrent la ressemblance de la courbure des tracés et la ressemblance des valeurs du taux de compression sauf pour les ondelettes d'ordres 5 et 7 de la courbe *Rachis-cervical*.

### 5.3.4 Compression par les ondelettes Coiflets

Le tableau 5.4 présente les valeurs de l'évaluation de la compression des trois images, *Rami*, *Lena* et *Rachis-cervical* par les ondelettes Coiflets.

Tableau 5.4: PSNR et TC de trois images compressées par des ondelettes Coiflets

Wavelet	Rachis-cervical		Rami		Lena	
	PSNR	TC%	PSNR	TC%	PSNR	TC%
<u>Coif1</u>	58.5566	60.61	55.4437	77.14	42.1980	87.64
<u>Coif2</u>	41.3902	91.36	49.5264	91.35	40.6449	89.87
<u>Coif3</u>	60.6623	57.99	48.3857	91.84	39.3921	90.26
<u>Coif4</u>	40.8005	89.66	48.8631	91.13	38.7011	89.78
<u>Coif5</u>	43.0173	87.87	46.6195	91.51	39.0349	88.92

Dans la figure 5.11 les courbes *Rami* et *Lena* suivent presque une ligne décroissante. Le PSNR de la courbe *Rami* varie de 46.6195 dB à 55.4437 dB. Et le PSNR de la courbe *Lena* varie de 38.7011 dB à 42.198 dB.

Toujours le PSNR de l'image *Rami* est supérieur au PSNR de l'image *Lena* pour toutes les ondelettes coiflets. Le PSNR de la courbe *Rachis-cervical* varie de 40.8005 dB à 60.6623 dB. Ce qui donne une différence d'environ 20 dB, la courbe est en forme de zigzag.

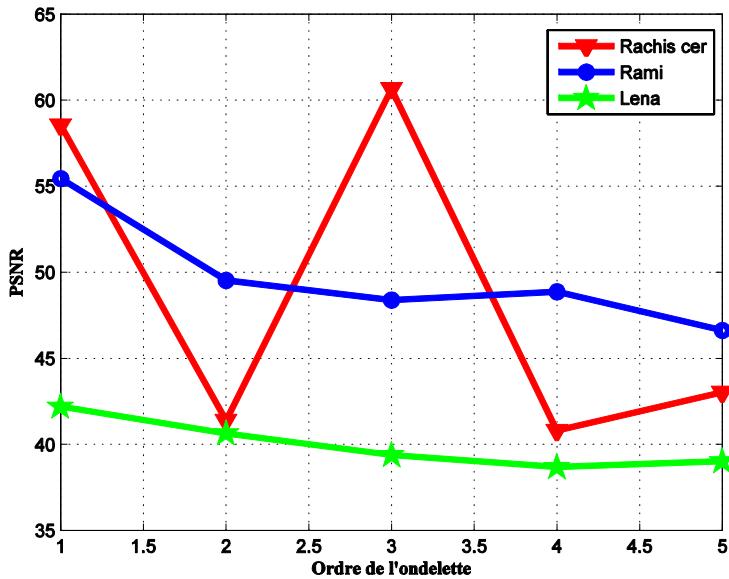


Figure 5. 11: PSNR de la compression de trois images par des ondelettes de Coiflets

La figure 5.12 exprime le taux de compression des images *Lena*, *Rami* et *Rachis-cervical* en fonction de l'ordre des ondelettes Coiflets. Le taux de compression des courbes *Rami* et *Lena* suit presque une ligne droite, ce qui n'est pas le cas de la courbe *Rachis-cervical* qui suit une ligne en zigzag. Le taux de compression le plus bas pour les deux courbes *Rami* et *Lena* est donné par l'ondelette Coif1, soit 77.14% et 87.64%. Le taux de compression le plus bas de la courbe *Rachis-cervical* est donné par l'ondelette Coif3 suivi en deuxième position par l'ondelette Coif1. Le taux de compression le plus haut est donné par l'ondelette Coif2. Si on exclut l'ondelette Coif1, le taux de compression des courbes *Rami* et *Lena* varie entre 91.13% à 91.84% et 88.92% à 90.26% respectivement. Donc, le taux de compression est presque stable.

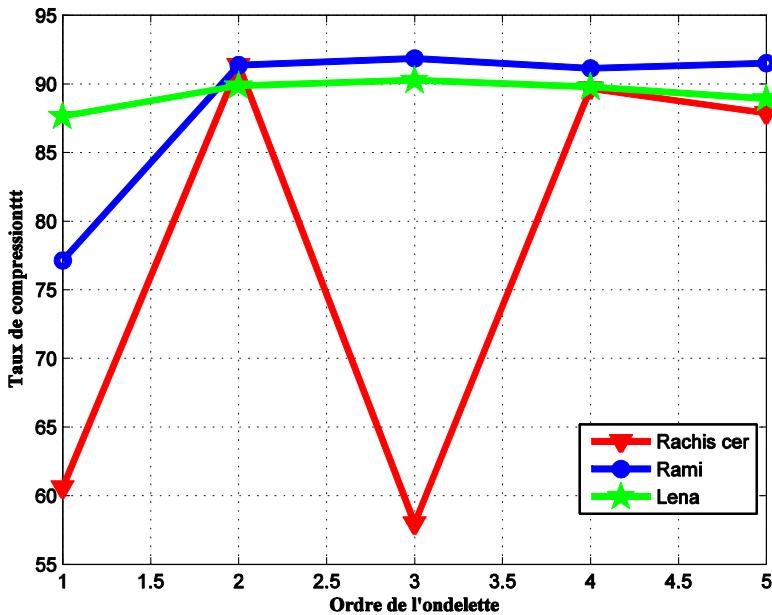


Figure 5. 12: TC de la compression de trois images par des ondelettes de Coiflets

On peut dire que lorsque l'image ne présente pas de variations de l'activité spectrale pour les fréquences horizontales et verticales, le TC de l'image est bon pour presque toutes les ondelettes Coiflets sinon le TC est variable.

## 5.4 Discussion

Les figures ci-dessous représentent une comparaison de la compression des images *Rami*, *Rachis-cervical* et *Lena* par les ondelettes biorthogonales, Daubechies, Symlets et Coiflets. La figure 5.13 décrit la compression des trois images par les ondelettes biorthogonales. Cette figure nous permet de voir que le meilleur PSNR des trois courbes est donné par la courbe *Rami* par l'ondelette Bior4.4 et avec un taux de compression supérieur à 92%. Cette ondelette Bior4.4 présente aussi le meilleur PSNR pour la courbe *Lena* avec un taux de compression supérieur à 91%. Pour la courbe *Rachis-cervical*, l'ondelette Bior6.8 donne le meilleur PSNR mais avec un taux de compression le plus faible des trois courbes soit 87.37%. En deuxième position, l'ondelette Bior4.4 présente un PSNR égal à 40.7106 dB et un taux de compression de 91.94%.

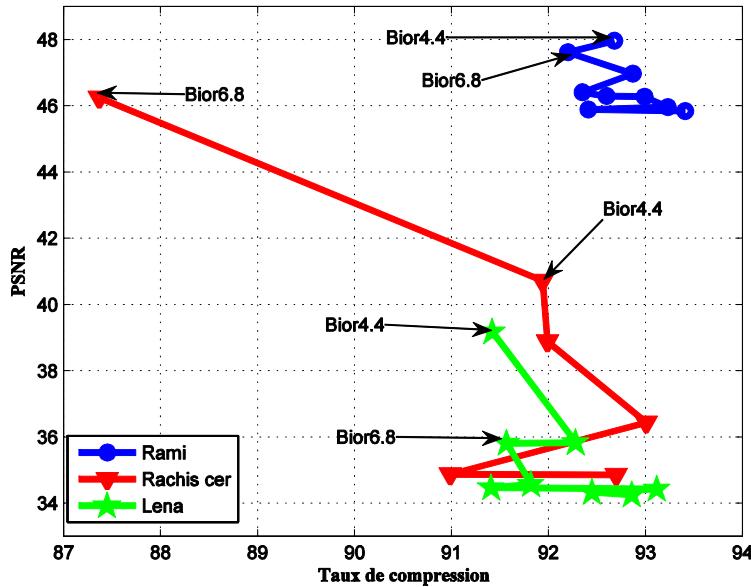


Figure 5. 13 : Comparaison de la compression des trois images par les ondelettes biorthogonales

Le deuxième meilleur PSNR des courbes *Rami* et *Lena* est donné par l'ondelette Bior6.8 avec un taux de compression supérieur à 91%. Dans la courbe *Rami*, l'ondelette Bior5.5 est légèrement supérieure à l'ondelette Bior4.4 en taux de compression. Pour l'image *Lena*, l'ondelette Bior5.5 donne un PSNR égal à 35.82 dB et l'ondelette Bior6.8 donne un PSNR égal à 35.80 dB. Le taux de compression de l'ondelette Bior5.5 est meilleur que la Bior4.4 et la Bior6.8 mais pas considérablement, soit 91.99%.

Comme la qualité de la compression prime sur le taux de compression, sauf pour certaines applications, alors on peut dire que pour la compression d'un ensemble d'images avec des ondelettes biorthogonales, il est préférable d'utiliser l'ondelette Bior4.4.

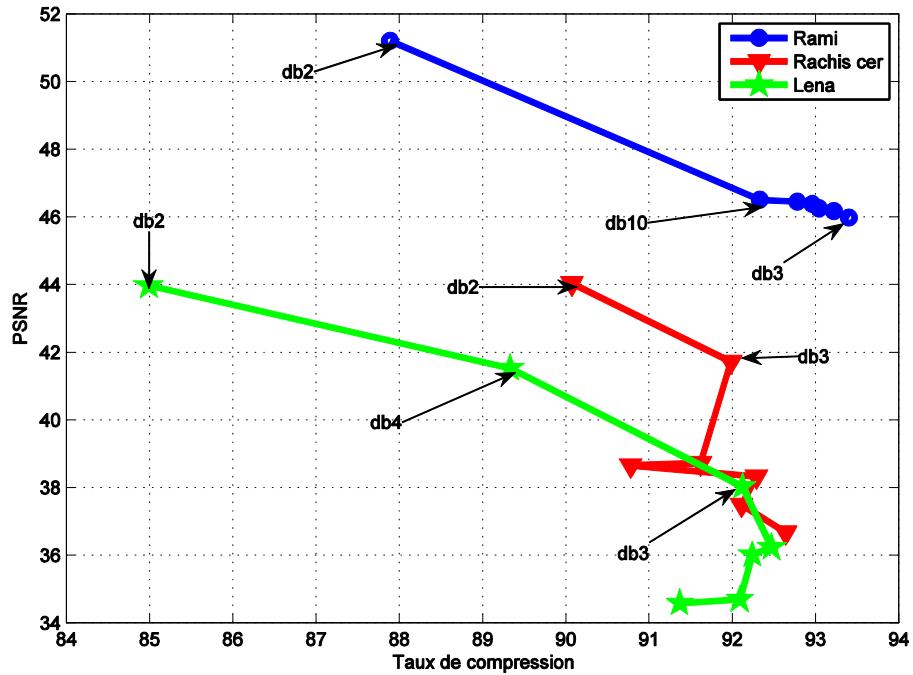


Figure 5. 14: Comparaison de la compression des trois images par les ondelettes Daubechies

Dans la figure 5.14, on voit clairement que l'ondelette Db2 donne le meilleur PSNR dans les trois courbes mais avec un taux de compression variant de 84.99% à 90.07%. Le taux de compression donné par l'ondelette Db2 est le taux le plus faible pour les trois images. On remarque également à partir du tableau 5.2 que l'ondelette Db3 donne un PSNR de 38.0298 dB avec un taux de compression de 92.12% pour l'image *Lena*, un PSNR de 41.7115 dB avec un taux de compression de 91.98% pour l'image *Rachis-cervical* et un PSNR de 45.9744 dB avec un taux de compression de 93.40% pour l'image *Rami*. Dans la famille des ondelettes Daubechies, l'ondelette Db3 donne un très bon taux de compression pour les trois images.

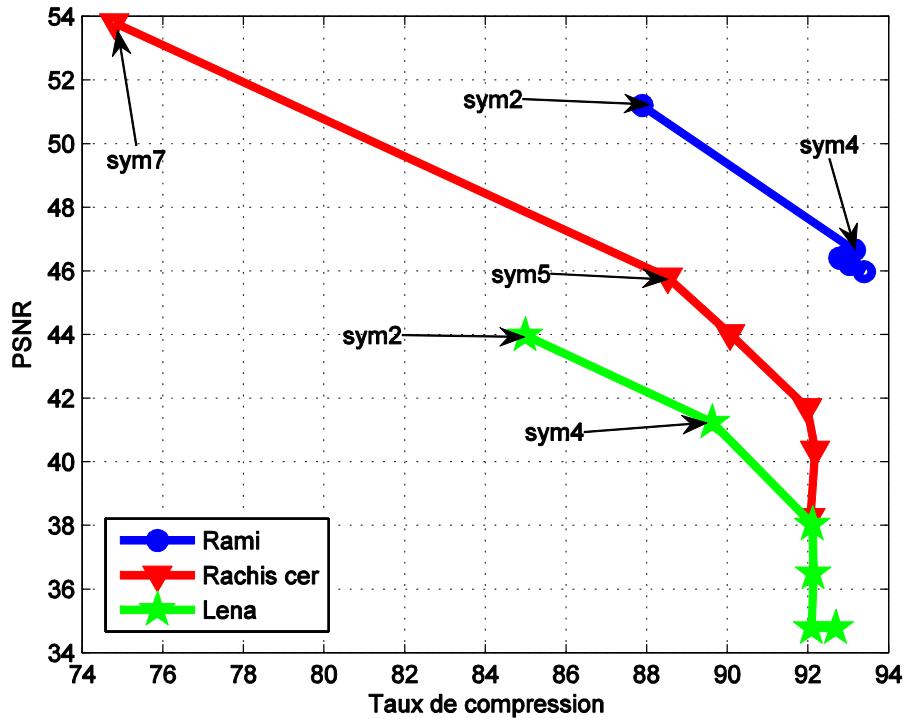


Figure 5.15: Comparaison de la compression des trois images par les ondelettes symlets

La figure 5.15 représente la compression des images *Rami*, *Lena* et *Rachis-cervical* par des ondelettes de Symlets. D'après la figure, on peut dire que l'ondelette Sym2 peut donner un très bon PSNR mais avec presque un taux de compression le plus bas pour les trois images. Pour les trois courbes, l'ondelette Sym4 donne un PSNR supérieur à 40 dB et un taux de compression supérieur à 89.63%. Sym3 et Sym4 donnent un bon compromis PSNR taux de compression pour les trois images.

La comparaison de la compression des trois images *Rami*, *Lena* et *Rachis-cervical* par les ondelettes Coiflets est résumée dans la figure 5.16. Les meilleurs PSNR de la courbe *Rami* et de la courbe *Lena* sont donnés par l'ondelette Coif1, soit 55.4437 dB et 42.1980 dB respectivement. Pour la courbe *Rachis-cervical*, le meilleur PSNR donné par l'ondelette Coif1 vient en deuxième position après Coif3, soit 58.5566dB et 60.6623dB respectivement. L'ondelette Coif2 donne le meilleur compromis PSNR TC pour les trois images. On remarque aussi que lorsque le PSNR est grand le taux de compression est bas et lorsque le PSNR est bas le taux de compression est haut. Le PSNR est inversement proportionnel au TC.

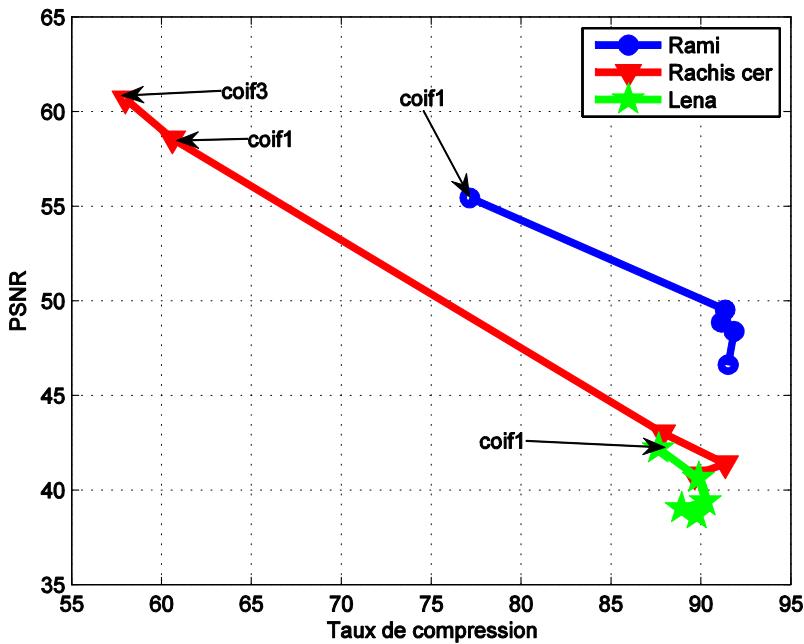


Figure 5.16: Comparaison de la compression des trois images par les ondelettes Coiflets

Afin de compléter nos résultats, nous avons appliqué les deux meilleures ondelettes de chaque famille pour la compression de l'image *Échographie*. À savoir, l'image *Échographie* présente plus d'activités spectrales que les trois images vues ci-haut, *Rami*, *Lena* et *Rachis-cervical*. Voir figure 5.4.

Le tableau 5.5, présente la comparaison des résultats de la compression de l'image *Échographie* avec les résultats des autres images citées ci-haut. L'ondelette Bior4.4 donne bien un bon compromis entre un PSNR et un taux de compression acceptable dans la famille des ondelettes biorthogonales. L'ondelette Db2 donne un bon PSNR, alors que l'ondelette Db3 donne un bon taux de compression dans la famille des ondelettes Daubechies. L'ondelette Sym2 donne aussi un bon PSNR suivi de l'ondelette Sym4 avec un taux de compression plus ou moins acceptable dans la famille des ondelettes de Symlets. L'ondelette Coif2 peut donner un meilleur compromis entre un PSNR et un taux de compression acceptable dans la famille des ondelettes Coiflets.

Tableau 5. 5 : Comparaison de la compression par les différentes ondelettes de l'image échographie avec la compression des images *Rachis-cervical*, *Rami* et *Lena*.

Wavelet	<i>Rachis-cervical</i>		<i>Rami</i>		<i>Lena</i>		<i>Écographie</i>	
	PSNR	TC%	PSNR	TC%	PSNR	TC%	PSNR	TC%
<u>Bior4.4</u>	40.7106	91.94	47.9637	92.68	39.1683	91.42	37.1562	82.33
Bior6.8	46.2444	87.37	47.6206	92.20	35.8069	91.57	25.9717	90.43
<u>Db2</u>	<b>44.03</b>	<b>90.07</b>	<b>51.1979</b>	<b>87.89</b>	<b>43.9693</b>	<b>84.99</b>	<b>43.4569</b>	<b>70.66</b>
Db3	41.7115	91.98	45.9744	93.40	38.0298	92.12	27.1471	92.28
<u>Sym2</u>	<b>44.03</b>	<b>90.07</b>	<b>51.1979</b>	<b>87.89</b>	<b>43.9693</b>	<b>84.99</b>	<b>43.4569</b>	<b>70.66</b>
Sym4	40.3677	92.18	46.6518	93.16	41.2303	89.63	33.9834	85.96
<u>Coif1</u>	58.5566	60.61	55.4437	77.14	42.1980	87.64	35.0848	84.11
<u>Coif2</u>	41.3902	91.36	49.5264	91.35	40.6449	89.87	68.0014	30.54
Coif3	60.6623	57.99	48.3857	91.84	39.3921	90.26	25.7166	90.51

## 5.5 Conclusion

Afin de résumer le travail de ce chapitre, nous pouvons conclure qu'en se basant sur nos images de test, nous sommes arrivés à montrer les points suivants :

- Ondelette Bior4.4 donne un bon compromis entre le PSNR et le taux de compression dans la famille des ondelettes biorthogonales.
- Ondelette Db2 donne un bon PSNR, alors que l'ondelette Db3 donne un bon taux de compression dans la famille des ondelettes Daubechies.
- Ondelette Sym2 donne un bon PSNR, meilleur que l'ondelette Sym4. L'ondelette Sym4 donne un taux de compression plus ou moins acceptable dans la famille des ondelettes de Symlets.
- Ondelette Coif2 peut donner un meilleur compromis entre le PSNR et le taux de compression dans la famille des ondelettes Coiflets et l'ondelette Coif1 donne le meilleur PSNR mais avec un taux de compression variable.
- Ondelettes Sym2 et Sym3 sont équivalentes aux ondelettes Db2 et Db3 respectivement, mais du point de vue complexité de calcul, Db2 et Db3 sont moins complexes.

Afin d'avoir un bon compromis PSNR-TC, il faut choisir l'ondelette Bior4.4 ou bien l'ondelette Coif1. Si on avait à choisir entre les deux ondelettes Bior4.4 et Coif1, nous prendrions Bior4.4 puisque les ondelettes de Coiflets sont symétriques. Ceci peut causer des déphasages dans le traitement d'image [51].

Si on voulait obtenir un très bon rapport signal sur bruit sans tenir compte du taux de compression alors on choisirait l'ondelette Db2 ou l'ondelette Coif2.

# CHAPITRE 6 COMPARAISON DE SYSTÈMES DE COMPRESSION À BASE DE DWT, DCT ET LA MÉTHODE HYBRIDE DWT-AMÉLIORÉ-DCT

## 6.1 Introduction

Après avoir effectué le choix des images de test et après avoir évalué les différentes ondelettes sur nos images de test, nous procérons dans ce chapitre à l'évaluation de notre algorithme de compression à base de la transformée discrète en ondelettes. Dans la littérature [2], [4]-[10], DCT et DWT sont recommandées pour la compression de données et spécialement pour la compression d'images. Conséquemment, un algorithme à base de DCT suivant la norme JPEG et un algorithme à base de DWT sont réalisés pour une comparaison des résultats.

## 6.2 Structure du travail

Dans cette section nous présentons les procédés de compression communs aux deux techniques DWT et DCT.

### 6.2.1 Images de test

Pour le test de nos algorithmes de compression nous avons utilisé deux images. La première image est *bust* (figure.3.4.a) et la deuxième image est *wbarb* (figure.3.3.a). Les deux images sont des images à niveaux de gris et de taille  $256 \times 256$ . L'activité spectrale est assez importante dans les deux images. On peut voir l'activité spectrale de ces deux images dans figure 3.4.b et figure 3.3.b, respectivement.

### 6.2.2 Prétraitement

Comme mentionné précédemment, les valeurs des pixels d'une image à niveaux de gris sont rangées de 0 à 255 avec un pas de 1. Un noir pur est représenté par zéro et un blanc pur est représenté par 255. Comme la transformée DCT et la transformée DWT sont applicables sur des pixels qui sont rangés de -128 à 127 [53], alors le bloc original est nivéé par la soustraction de 128 pour chaque entrée.

### 6.2.3 Quantification

Comme expliqué dans le chapitre 3, le choix de la matrice de quantification détermine la précision de l'image reconstruite et une bonne partie de l'information est perdue dans cette étape. Pour débuter la compression, on représente l'image en forme de matrice. La transformation DCT est appliquée pour chaque bloc de taille 8x8 de cette matrice, de gauche à droite et de haut au bas. Chaque coefficient de cette transformation DCT est divisé par le coefficient correspondant dans la matrice de quantification et le résultat est arrondi à la valeur entière. Le nombre de zéros contenu dans la matrice résultante après quantification varie selon le choix du facteur de qualité utilisé dans la matrice de quantification.

Pour pouvoir comparer la technique de compression DCT à la technique de compression DWT, il nous faut plusieurs matrices de quantification avec un facteur de qualité variable. Dans la compression JPEG, on ne trouve pas une formule pour la génération des matrices de quantification, mais la matrice de quantification standard JPEG de luminance [3] est recommandée. Alors, nous avons apporté une modification sur la formule de l'équation (3.43), la formule qui génère les matrices de quantification. Nous avons modifié la formule pour avoir :

$$Q(i, j) = (1 + (1 + i + j) \times f_q) / 4 \quad (6.1)$$

Avec,  $f_q$  : facteur de qualité varie de 1 à 100

$i, j$  : ligne, colonne de la matrice varie de 0 à 7.

La figure 6.1 permet de visualiser la variation du nombre de coefficients non nul utilisés dans la reconstruction de l'image en fonction du rapport signal sur bruit (PSNR). Le nombre de coefficients non nul est exprimé en pourcent (%) et le PSNR en dB. La compression de l'image *bust* et l'image *wbarb* par DCT montre que lorsque le nombre de coefficients DCT pour la représentation de l'image diminue, la valeur du PSNR diminue aussi.

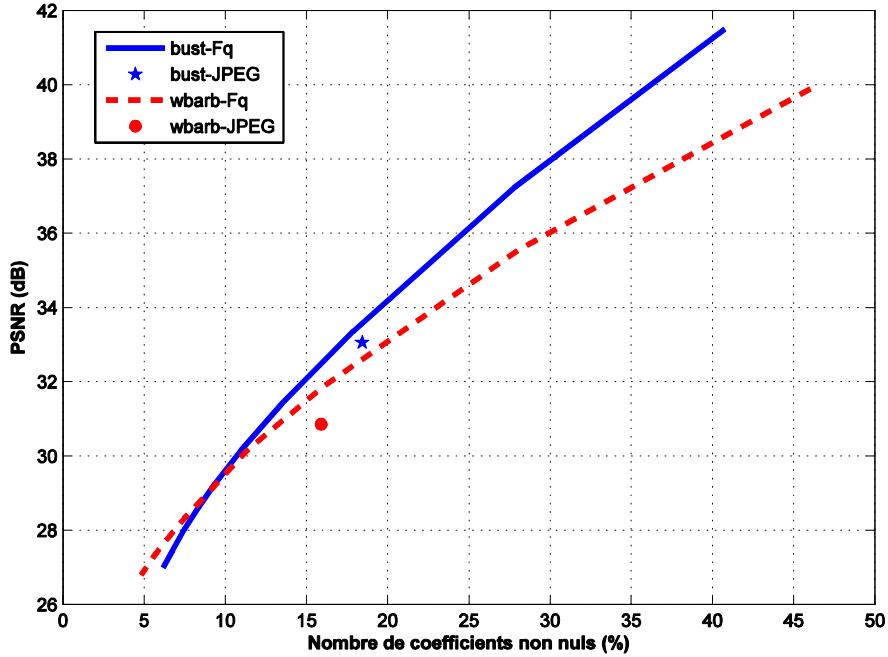


Figure 6.1: Deux images compressées avec DCT en utilisant l'équation (6.1) et la matrice de quantification JPEG

D'autre part, les deux points qui se trouvent sur la figure 6.1 représentent la compression de l'image *bust* (étoile bleu) et l'image *wbarb* (point rouge) utilisant la matrice de quantification standard de luminance de JPEG. Ces deux points sont utilisés comme références. On remarque bien que les courbes générées par l'utilisation de l'équation (6.1) comme matrice de quantification des deux images *bust* et *wbarb* présentent un meilleur PSNR que la matrice de référence (matrice de quantification standard de luminance JPEG).

#### 6.2.4 Seuillage

Le choix du seuil doit tenir compte du système visuel humain, afin d'atteindre une meilleure qualité de compression et un haut taux de compression [3], [6], [51]. Le système visuel humain est moins sensible aux fréquences hautes mais sensible aux fréquences basses. Comme décrit dans le chapitre 1, nous avons deux types de seuillages, le seuillage dur et le seuillage doux. Pour la compression à base de la transformée discrète en ondelettes, nous avons choisi le seuillage dur.

### 6.2.5 Codage

Pour la transmission et l'archivage, il est important de minimiser le nombre de bits du signal ou de l'image à transmettre. Alors, après quantification ou seuillage le codage est effectué. Les méthodes utilisées pour le codage sont en général des méthodes de compression réversibles.

Rappelons que l'objectif de notre recherche est la comparaison des deux techniques DCT et DWT sans l'influence des codeurs. Alors, le bloc codeur n'est pas considéré.

### 6.2.6 Évaluation

Noté que pour fin de complémentarité, nous allons répéter les explications données dans le chapitre 5 sur l'évaluation.

L'évaluation de la qualité d'une image après compression est une étape indispensable. L'appréciation visuelle est subjective, mais elle est toujours présente dans l'évaluation.

Le calcul du rapport signal bruit (PSNR) et le pourcentage du nombre de coefficients utilisé pour la reconstruction de l'image à partir de l'image originale sont les mesures objectives dans notre étude.

Le PSNR permet de fournir une mesure objective de la distorsion introduite par la compression. Soient  $x(i,j)$  et  $y(i,j)$  l'image originale et l'image reconstruite, de même taille  $M \times N$ . Le calcul du PSNR est défini par le calcul de l'erreur quadratique moyenne ( $MSE$  : Mean Square Error).

$$MSE = \frac{1}{n \cdot m} \sum_{i=1}^n \sum_{j=1}^m [x(i,j) - y(i,j)]^2 \quad (6.2)$$

et le PSNR est défini comme suit :

$$PSNR = 10 \log_{10} \left( \frac{Pic^2}{MSE} \right) \quad (6.3)$$

Avec  $Pic$  est le maximum possible de la valeur du pixel d'image. Comme les pixels des images sont codés sur 8 bits, alors  $Pic = 2^8 - 1 = 255$ . L'unité du PSNR est le décibel (dB). Plus le PSNR est important plus les images sont semblables, aussi pour une bonne reconstruction de l'image le PSNR doit être égale ou supérieur à 30 dB [44].

## 6.3 Compression par DCT

### 6.3.1 Diagramme bloc

Comme on le voit dans la figure 6.2 du diagramme bloc, après la phase de découpage d'image en blocs de  $8 \times 8$ , trois opérations sont nécessaires pour la compression de l'image  $X(i,j)$  par DCT: la transformation, la quantification et le codage. Concernant la reconstruction de l'image, on applique les opérations de décodage, de quantification inverse, de transformation inverse puis on assemble les blocs résultants.

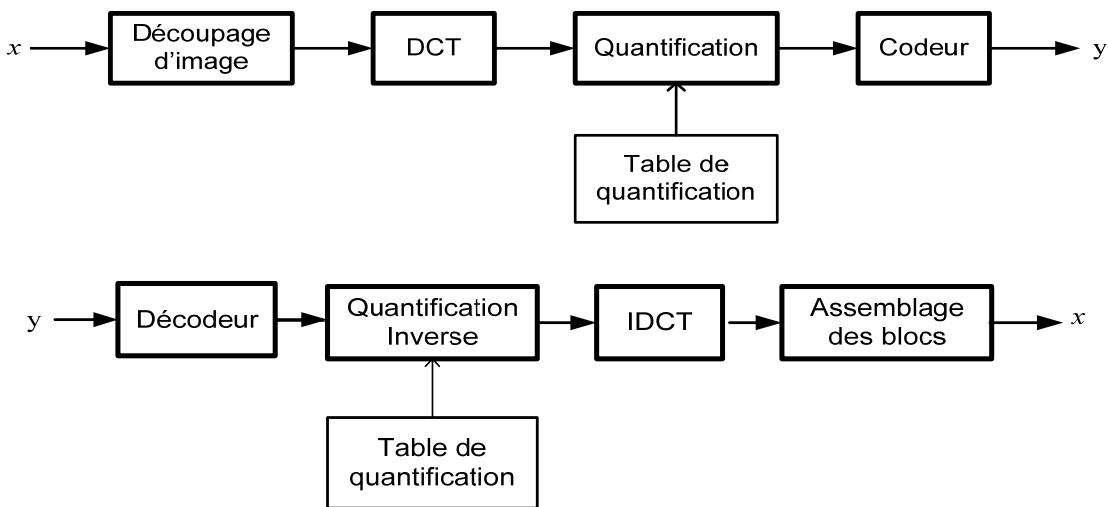


Figure 6. 2 : Diagramme bloc de la compression et décompression par DCT

L'algorithme de compression à base de DCT modélisé pour notre travail, suit le mode de base DCT séquentiel présenté dans le chapitre 3.

### 6.3.2 Algorithme à base de DCT séquentiel

Pour réduire la complexité des calculs, l'image est divisée en blocs de  $8 \times 8$ . La DCT à deux dimensions est appliquée sur chaque bloc de haut en bas et de gauche à droite. Les éléments de la matrice obtenue sont appelés coefficients DCT. Les coefficients basses-fréquences sont localisés dans le coin supérieur gauche de la matrice et les coefficients hautes-fréquences sont localisés dans le coin inférieur à droite de la matrice. Une matrice de quantification est appliquée suivant le facteur de qualité choisi. Après codage en zigzag, nous allons avoir une longue chaîne de zéro pour chaque bloc de  $8 \times 8$ . Pour la reconstruction de l'image, la matrice décodée est multipliée par

la matrice de quantification, puis la transformée inverse en cosinus à deux dimensions est appliquée pour donner des blocs de  $8 \times 8$ . La valeur 128 est ajoutée au résultat. Ces différents détails sont donnés dans l'algorithme ci-dessous:

**Algorithme :**

**Étape 1 : Prétraitement de l'image**

*Avoir niveaux de gris pour les images à compresser*

*Calcul de la matrice  $\mathbf{M}$ , le bloc original est nivélé par la soustraction de 128*

**Étape 2 : Décomposition de l'image**

*Choisir des blocs de  $8 \times 8$*

*Calcul de la matrice de forme orthogonale  $\mathbf{T}$*

*Calcul de la DCT,  $\mathbf{D} = \mathbf{T} \mathbf{M} \mathbf{T}^T$*

**Étape 3 : Quantification**

*Choisir la matrice de quantification*

*Diviser les blocs  $8 \times 8$  de la matrice transformée par la matrice de quantification*

**Étape 4 : Codage**

*Codage en zigzag*

*Encodage arithmétique ou Huffman*

**Étape 5 : Reconstruction de l'image**

*Décodage*

*Dé-quantification :*

*Multiplier les blocs de  $8 \times 8$  de la matrice décodée par la matrice de quantification inverse*

*Transformée inverse :*

*Calcul de la IDCT,  $\mathbf{D}^T = \mathbf{T}^T \mathbf{M} \mathbf{T}$*

*Ajout de 128 à la matrice  $\mathbf{D}^T$*

**Fin**

### 6.3.3 Matrice de quantification

Les matrices de quantification utilisées pour l'évaluation sont générées de l'équation (6.1). Le tableau 6.1 permet de visualiser la variation du nombre de coefficients non nuls en fonction de la variation du facteur de qualité. Plus le facteur de qualité est grand plus le nombre de coefficients nuls est petit. Donc, le facteur de qualité et le nombre de coefficients nuls après quantification sont inversement proportionnels, mais la variation du PSNR et le facteur de qualité sont proportionnels. Lorsque le facteur de qualité augmente, le PSNR augmente et le nombre de coefficients nuls diminue.

Tableau 6. 1 : Évaluation de l'algorithme de compression DCT pour deux images

Facteur de qualité : Fq	DCT Image 1: bust		DCT Image 2: wbarb	
	Nb de coef non nul(%)	PSNR (dB)	Nb de coef non nul(%)	PSNR (dB)
10	6.18	27.0029	4.84	26.7970
20	6.72	27.4389	5.26	27.0760
30	7.39	27.9533	5.76	27.3918
40	8.25	28.5451	6.46	27.7862
50	9.35	29.2540	7.46	28.3034
60	10.95	30.1535		
70	13.48	31.4066	11.28	30.1256
80	17.88	33.3629	15.96	31.8454
90	27.76	37.2238	28.12	35.5675
95	40.74	41.4938	45.70	39.8848

La figure 6.3 montre la variation du nombre de coefficients non nuls de DCT utilisée pour la reconstruction de l'image en fonction du PSNR. Les courbes *wbarb* et *bust* montrent clairement l'influence de l'activité spectrale sur la qualité de l'image. Comme l'image *bust* présente moins d'activité spectrale que l'image *wbarb*, alors le PSNR de l'image *bust* est supérieur à celui de l'image *wbarb* pour une même valeur du nombre de coefficients DCT non nuls utilisé pour la reconstruction de l'image.

D'autre part, on remarque que plus le nombre de coefficients DCT non nuls utilisé pour la reconstruction de l'image diminue, plus l'écart des deux courbes diminue. Pour un nombre de coefficients DCT égal à 38%, le PSNR de la courbe *wbarb* est de 38dB et le PSNR de l'image *bust* est environ 41.5 dB, donc un écart de 2.5 dB. Alors, que pour un nombre de coefficients DCT égal à 9% la valeur du PSNR de l'image *bust* et de l'image *wbarb* est environ 29 dB, donc l'écart est nul.

On peut observer aussi dans la figure 6.3 que pour un nombre de coefficients DCT inférieur ou égal à 9 %, le PSNR de l'image *wbarb* est supérieur à celui de l'image *bust*. L'écart entre les deux courbes n'est pas significatif. On peut conclure qu'à partir d'un certain seuil, la réduction de la qualité de l'image diminue plus rapidement et que l'image ayant plus d'activité spectrale prend le dessus.

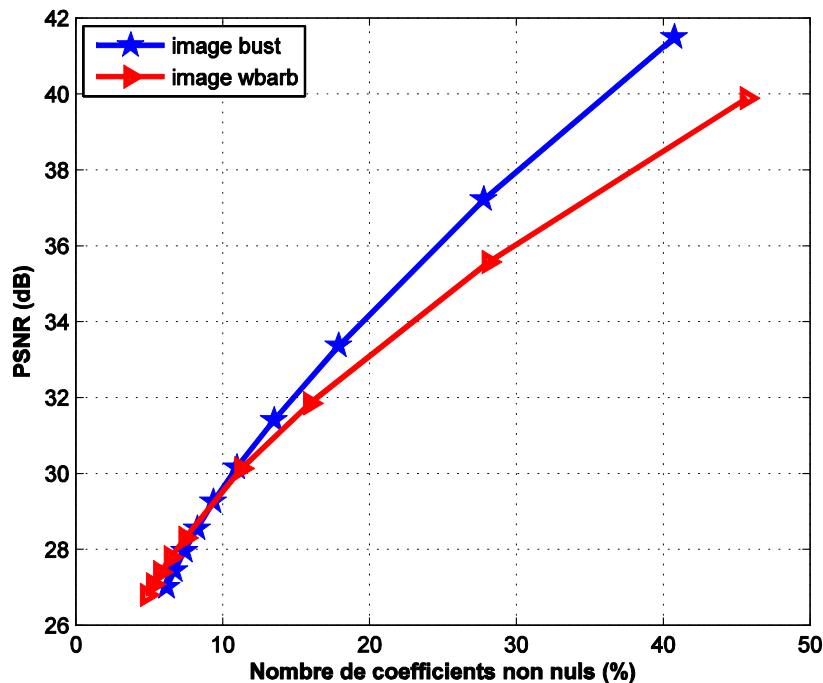


Figure 6.3 : Comparaison de deux images compressées par DCT

### 6.3.4 Exemple de compression d'images

La figure 6.4 représente l'image *wbarb* originale utilisée pour l'évaluation de la compression. Après application de la transformation DCT sur cette image, nous avons pris juste 28.12 % des coefficients DCT, le résultat est l'image de la figure 6.5. Le PSNR de cette image est égal à 35.5675 dB. On visualisant cette image on ne distingue presque pas de différence avec l'image originale de la figure 6.4.

Dans la figure 6.6 l'image est reconstruite à partir de juste 15.96 % des coefficients DCT, son PSNR est égal à 31.8454 dB. La comparaison de cette image avec l'image originale présente une différence mais cette différence est acceptable. La figure 6.7 et la figure 6.8 montrent des dégradations dans les images reconstruites très apparentes. La reconstruction de l'image avec juste 6.46% des coefficients DCT est visualisée dans la figure 6.7, son PSNR est égal à 27.7862 dB. La reconstruction de l'image avec 4.84 % des coefficients DCT est visualisée dans la figure 6.8 avec un PSNR égal à 26.7970 dB. La dégradation de ces deux dernières images est appelée effet de blocs ou mosaïque. Elle est due à l'étape de découpage de l'image.



Figure 6. 4 : Image originale de *wbarb*

Image compressée par DCT (coefficients: 28.12%)



Figure 6. 5: Image *wbarb* compressée par DCT avec coefficient non nuls 28.12%

Image compressée par DCT (coefficients: 15.96%)



Figure 6. 6: Image *wbarb* compressée par DCT avec coefficient non nuls 15.96%



Figure 6. 7: Image *wbarb* compressée par DCT avec coefficients non nuls 6.46%



Figure 6. 8: Image compressée par DCT avec coefficient non nuls 4.84%

L'effet de blocs est bien visualisé sur l'image de la figure 6.8.

## 6.4 Compression à base de DWT

Dans cette section nous présentons la méthode de compression d'images à base de DWT avec des exemples d'images en montrant l'influence de l'activité spectrale sur le résultat de compression et vers la fin nous avons proposé une méthode appelée DWT-amélioré qui permet l'amélioration des résultats de compression. L'ondelette utilisée pour la compression est l'ondelette bio4.4 de la famille biorthogonale.

### 6.4.1 Diagramme bloc

La figure 6.9 présente le diagramme bloc du codeur et du décodeur. Trois opérations sont appliquées sur l'image à compresser: la transformation DWT, la quantification ou seuillage et le codage. Pour la reconstruction de l'image, on applique les opérations suivantes : le décodage, la quantification inverse puis la transformation inverse IDWT. La transformation c'est la phase de décomposition de l'image et la transformation inverse c'est la phase de reconstruction de celle-ci.

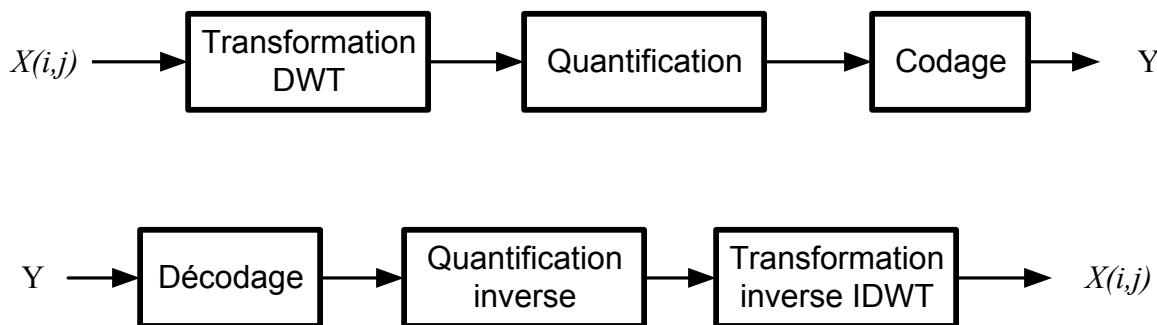


Figure 6. 9 : Bloc diagramme de la compression et décompression par DWT

### 6.4.2 Algorithme à base de DWT

Pour éviter l'effet de bloc, la transformation discrète en ondelettes est appliquée sur l'image entière. Pour la décomposition de l'image, le nombre d'itérations est choisi égal à 3. Alors, pour chaque itération l'image d'approximation traverse un filtre passe-bas et un filtre passe-haut suivi d'une décimation d'un facteur de 2, comme présenté dans le chapitre 4. Nous avons choisi le coefficient minimal de l'image d'approximation de niveau 3 comme un seuil global à ne pas dépasser. Donc, pour tout coefficient des deux autres niveaux (niveau 1 et niveau 2) inférieur ou égal au seuil global est mis à zéro. Le résultat est enregistré dans deux vecteurs, l'un pour les

coefficients et l'autre pour la taille de l'image originale et les images de niveau 1, 2 et 3, selon un certain ordre. Pour la reconstruction de l'image nous avons besoin de ces deux vecteurs et des filtres de reconstruction. Les différentes étapes de notre algorithme à base de DWT sont décrites ci-dessous:

**Algorithme:**

**Étape1 : Prétraitement de l'image**

*Avoir niveaux de gris pour les images à compresser*

*Calcul de  $M$ , le bloc original est nivéle par la soustraction de 128*

**Étape2 : Décomposition de l'image**

*Choisir la fonction ondelette (bior4.4)*

*Choisir le nombre d'itérations*

*Calcul des coefficients d'approximation*

*Calcul des coefficients de détails*

**Étape3 : Quantification ou seuillage**

*Choisir le seuil (inférieur ou égal au  $\min(cAi)$ )*

*Choisir les niveaux de détail sur lesquels la compression sera appliquée*

*Pour tout coefficient inférieur au seuil, remplacer par zéro*

**Étape4 : Codage**

*Codage en zigzag*

*Encodage arithmétique ou Huffman*

**Étape5: Reconstruction de l'image**

*Décodage:*

*Les vecteurs quantifiés*

*Transformée inverse:*

*La fonction ondelette*

*Ajout de 128 pour le nivelement*

**Fin.**

### 6.4.3 Seuillage

Après décomposition de l'image en trois itérations, nous allons avoir trois images de niveau 1, trois images de niveau 2 (images diagonale verticale et horizontale) et quatre images de niveau 3 (image d'approximation, image diagonale, image verticale et image horizontale). Pour la compression, nous avons décidé de ne pas modifier les quatre images de niveau 3 et de prendre comme seuil de compression pour les six autres images la plus petite valeur du coefficient de l'image d'approximation. Le seuillage utilisé est le seuillage dur. La variation du seuil pour l'évaluation se fait en divisant notre seuil à chaque fois par la valeur 2.

Le tableau 6.2 montre bien cette variation. À chaque fois que la valeur du seuil est divisée par la valeur 2, le nombre de coefficients DWT non nuls augmente et la valeur du PSNR augmente.

Tableau 6. 2 : Évaluation de l'algorithme de compression DWT pour deux images

DWT de l'image bust			DWT de l'image wbarb		
seuil	Nb de coef non nuls(%)	PSNR (dB)	seuil	Nb de coef non nuls(%)	PSNR (dB)
Seuil 0	8.33	24.7447	Seuil 1	8.33	25.8702
Seuil 0/2	8.66	25.8747	Seuil 1/2	8.37	26.0400
Seuil 0/4	9.84	28.0138	Seuil 1/4	9.02	27.2794
Seuil 0/8	13.70	31.5310	Seuil 1/8	12.51	30.3013
Seuil 0/16	21.30	35.9071	Seuil 1/16	20.21	34.4108
Seuil 0/32	33.39	41.1103	Seuil 1/32	32.49	38.9832
Seuil 0/64	48.66	46.8347	Seuil 1/64	52.47	44.9369

La figure 6.10 présente la variation du nombre de coefficients utilisé pour la reconstruction de l'image en fonction du PSNR de deux images, image *bust* et image *wbarb*. La compression de

ces deux images par la transformée DWT montre toujours la supériorité de l'image *bust*. L'image *bust* a moins d'activité spectrale que l'image *wbarb*. On remarque que pour un nombre de coefficients non nuls égal à 27% le PSNR de l'image *bust* est supérieur au PSNR de l'image *wbarb* de plus de 1 dB. Pour un nombre de coefficients non nuls égal à 11%, le PSNR de l'image *bust* et le PSNR de l'image *wbarb* présentent une différence presque nulle. On observe aussi, qu'à partir d'un nombre de coefficients non nuls égal ou inférieur à 10%, la courbe de l'image *wbarb* prend le dessus et présente un PSNR supérieur à celui de l'image *bust*. On peut conclure que la différence du PSNR des deux images diminue avec la diminution du nombre de coefficients DWT utilisé pour la reconstruction de l'image, mais toujours en faveur de l'image ayant moins d'activité spectrale que l'autre. À partir d'un certain nombre de coefficients cette différence s'annule puis augmente mais pas considérablement, en faveur de l'image ayant le plus d'activité spectrale.

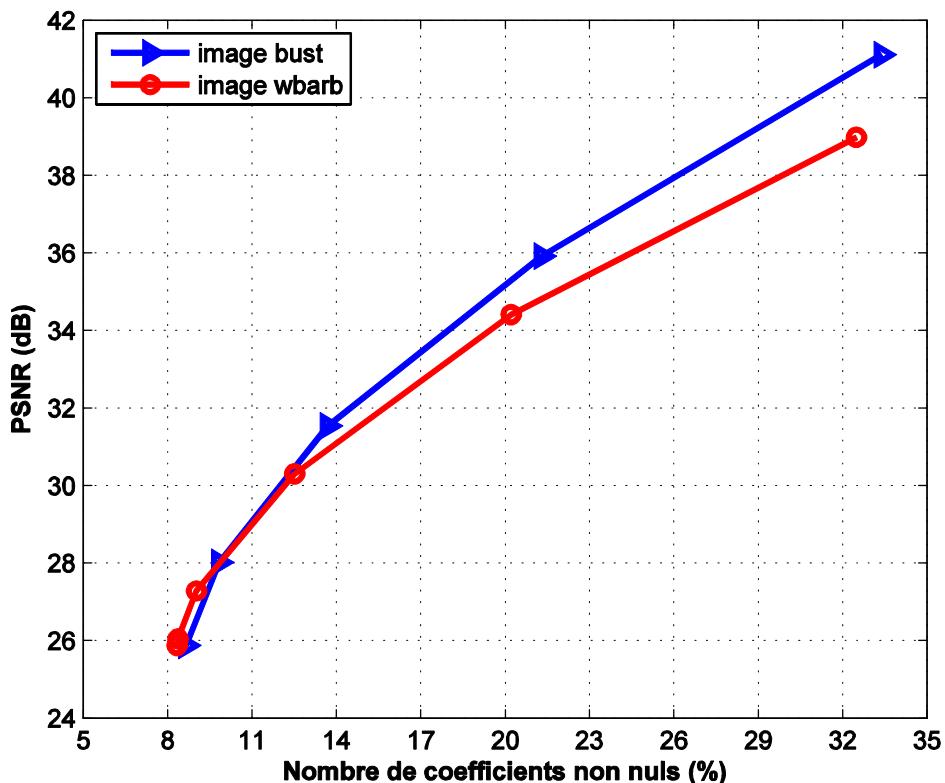


Figure 6. 10: Compression par DWT pour deux images

#### 6.4.4 Exemple de compression d'images

La figure 6.11 montre l'image *wbarb* compressée par la transformée DWT en utilisant approximativement la moitié des coefficients DWT de l'image (52.47 %). L'image reconstruite est indifférentiable de l'image originale (figure 6.4), le PSNR de cette image est de 44.9369 dB.



Figure 6. 11: Image *wbarb* compressée par DWT avec coefficients non nuls 52.47%

La figure 6.12 est une image reconstruite avec un nombre de coefficients DWT de 20.21%, son PSNR est de 34.4108 dB. Cette image est très proche de l'image originale de la figure 6.4.

Avec un nombre de coefficients DWT non nuls, égal à 12.51% et un PSNR égal à 30.3013 dB, nous avons l'image de la figure 6.13. Cette image présente le phénomène de flou au lieu de l'effet mosaïque généré par la compression DCT. Ce phénomène de flou est beaucoup plus accentué dans l'image de la figure 6.14 où le nombre de coefficients DWT utilisé dans l'image reconstruite est égal à 8.33% et son PSNR est égal à 25.8702 dB.



Figure 6. 12: Image *wbarb* compressée par DWT avec coefficients non nuls 20.21%



Figure 6. 13: Image *wbarb* compressée par DWT avec coefficients non nuls 12.51%



Figure 6. 14: Image *wbarb* compressée par DWT avec coefficients non nuls 8.33%

**DWT-amélioré :**

Afin d'améliorer la qualité de notre algorithme de compression, nous avons apporté un léger changement sur la partie seuillage. Alors au lieu d'appliquer le seuil choisi sur uniquement les six images des niveaux d'itérations 1 et 2, nous l'avons appliqué sur la totalité des images, donc sur les neuf images de détails (détail horizontal, détail vertical et détail diagonal) des trois niveaux d'itérations. Mais le seuil choisi pour les images d'itération 1 et d'itération 2 doit être différent du seuil d'itération 3. Cette méthode nous l'avons appelée *DWT-amélioré*. La figure 6.15 présente le détail de la compression par la méthode *DWT-améliorée*.

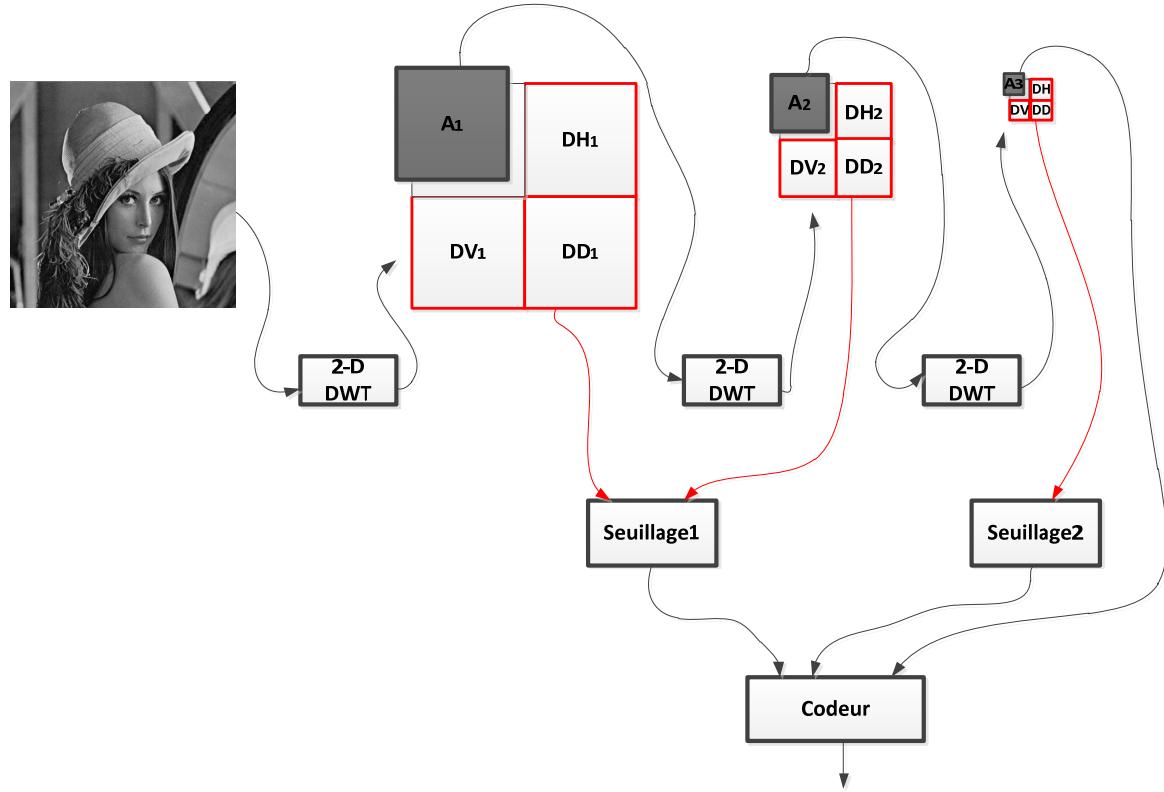


Figure 6. 15: Schéma détaillé de la compression de l'image *Lena* par *DWT-améliorée*.



Figure 6. 16: Image *wbarb* compressée par *DWT-amélioré* avec coefficients non nuls 3.7%

Dans la figure 6.16 on voit bien que cette méthode a permis de diminuer encore plus le nombre de coefficients utilisé dans la reconstruction de l'image (coefficients non nuls égal à 3.7% et PSNR égal à 24.5032 dB), c'est ce qui explique la dégradation de la qualité de l'image. Nous avons aussi appliqué cette méthode à la figure 6.17 où le nombre de coefficients de DWT utilisé pour la reconstruction de l'image est égal à 10.05 % et le PSNR correspondant est égal à 30.2762 dB. Le phénomène de flou est toujours présent mais l'image est plus au moins acceptable en comparaison avec l'image de la figure 6.13 où le PSNR est égal à 30.3013 dB et le nombre de coefficients DWT utilisé pour la reconstruction est égal à 12.51%.

**Image compressée par bior4.4 (coefficients: 10.05%)**



Figure 6. 17: Image *wbarb* compressée par *DWT-amélioré* avec coefficients non nuls 10.05%

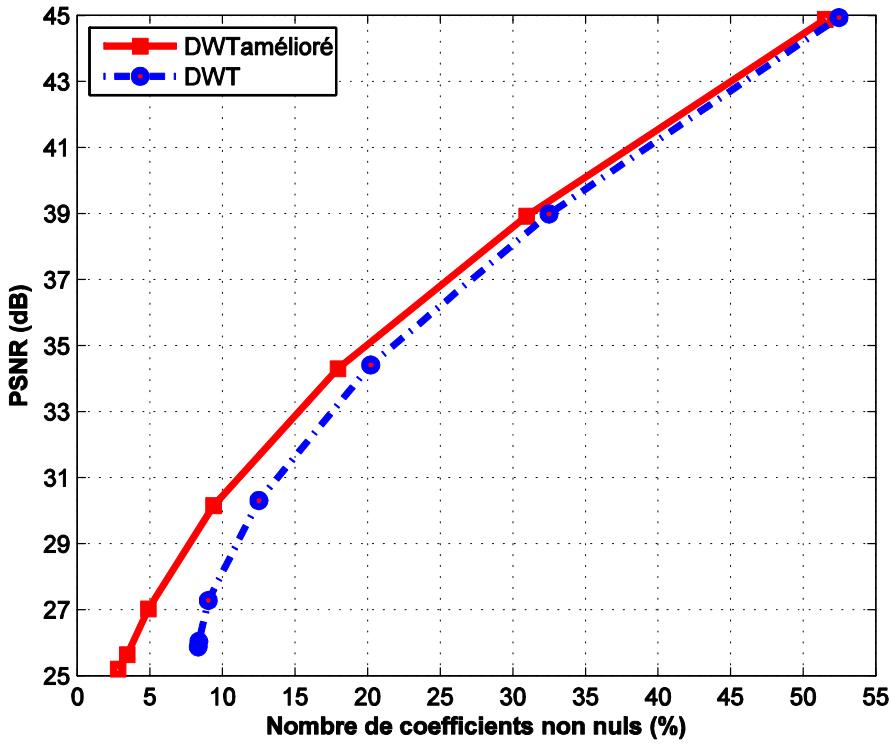


Figure 6.18: Comparaison de la compression DWT avec la compression *DWT-amélioré*

La figure 6.18 présente la supériorité de la compression par *DWT-amélioré* précisément lorsque le nombre de coefficients de la transformée non nuls diminue.

## 6.5 Comparaison de la compression DWT et DCT

La figure 6.19 présente la comparaison de la compression à base de DWT et DCT de l'image *bust*. Le seuil choisi pour la compression à base de DWT est appliqué uniquement sur les images de l'itération 1 et de l'itération 2. Cette comparaison est basée sur l'évolution du PSNR en fonction du nombre de coefficients de la transformée utilisé pour la reconstruction. Pour un nombre de coefficients non nuls de la transformée égal à 33%, le PSNR de l'image compressée par DCT est légèrement inférieur à 39 dB et le PSNR de cette même image compressée par DWT est légèrement supérieur à 41 dB (nous avons appliqué un zoom). Donc nous avons une différence d'environ 2 dB en faveur de la compression à base de DWT. Cette différence diminue au fur et à mesure que le nombre de coefficients utilisés pour la reconstruction diminue. Si le nombre de coefficients de la transformée augmente, la différence augmente aussi. Cette différence s'annule pour un nombre de coefficients de la transformée proche de 14 %. Pour un

nombre de coefficients de la transformée inférieur à 14%, cette différence est inversée en faveur de la compression à base de DCT. Pour un nombre de coefficient égal à 9%, le PSNR de l'image compressée par DWT est égal à 26.5 dB et le PSNR de l'image compressée à base de DCT est égal à 29 dB. Donc une différence égale à 2.5 dB en faveur de la compression à base de la DCT (nous avons appliqué un zoom sur la figure).

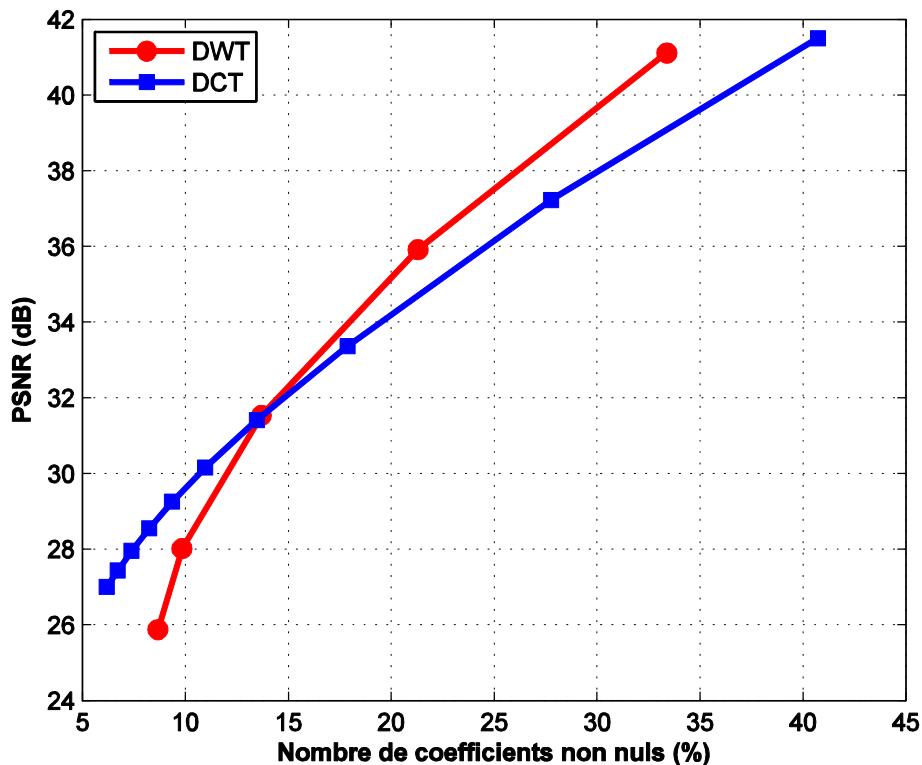


Figure 6. 19 : Comparaison de la compression par DWT et DCT pour l'image bust

La comparaison de la compression de l'image *wbarb* par DWT-amélioré, DWT et DCT est présentée dans la figure 6.20. On observe que les remarques faites concernant la compression de l'image *bust* par DWT et DCT sont toujours valables pour la compression de l'image *wbarb* par ces deux techniques. Pour n'importe quel nombre de coefficients non nuls utilisé pour la reconstruction de l'image, la DWT-amélioré donne des PSNR supérieurs à ceux donnés par DCT ou DWT. Lorsque, la différence entre la compression par DCT et DWT est nulle, la DWT-amélioré présente une différence avec ces deux techniques supérieure à 1 dB.

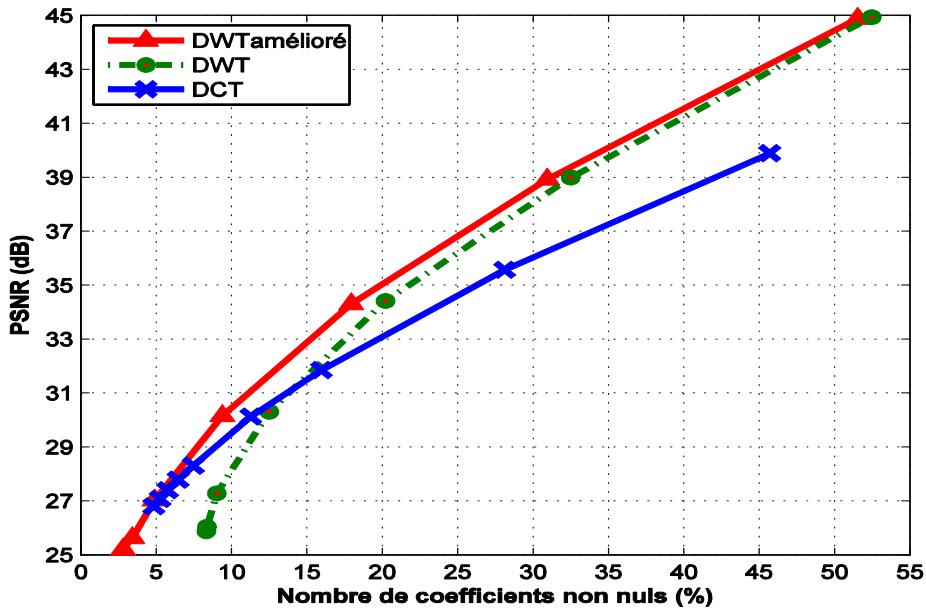


Figure 6. 20: Comparaison de la compression par DWT-amélioré, DWT et DCT pour l'image *wbarb*.

## 6.6 Combinaison des algorithmes de compression de DWT et DCT

Dans cette partie nous voulons voir si on peut atteindre des résultats de compression d'images meilleurs que ceux obtenus dans les sections 6.3 et 6.4. On propose la combinaison des deux techniques DCT et *DWT-amélioré* du fait que la combinaison de deux transformées peut pallier aux inconvénients de l'une et de l'autre [71]. Cette méthode est nommée *DWT-amélioré-DCT*.

### 6.6.1 Diagramme bloc

Le bloc diagramme de l'algorithme de compression *DWT-amélioré-DCT* est présenté dans la figure 6.21. Le choix de notre architecture est dicté par les points suivants:

- Effet de mosaïque à fort taux de compression par DCT,
- DWT permet une analyse globale de l'image,
- Effet de flou à fort taux de compression par DWT et DWT-amélioré,
- La compression par DWT est meilleure que la compression par DCT pour une large bande,
- La compression par DCT est meilleure que la compression DWT pour une petite bande,
- La compression DWT-amélioré donne de meilleur résultat que la compression DCT et la compression DWT.

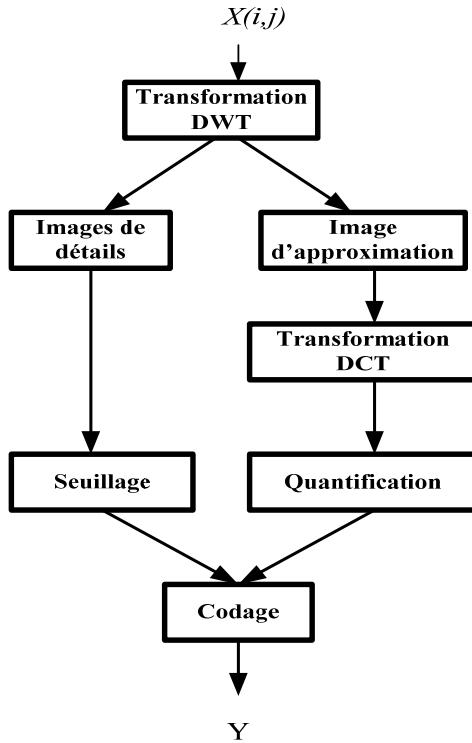


Figure 6. 21: Diagramme bloc de la combinaison de la compression DCT et *DWT-améliorée*.

### 6.6.2 Algorithme à base de DWT et DCT

L'image à compresser est décomposée en sous-bandes par l'algorithme de Mallat. Le nombre d'itérations est égal à 3 et la fonction ondelette est la Bior4.4. Les images de détail des itérations 1, 2 et 3 vont subir le processus de la compression DWT-amélioré et l'image d'approximation de niveau 3 subira la compression DCT avec un facteur de qualité égal à 80.

#### Algorithme:

##### Étape1 : Prétraitement de l'image

*Avoir niveaux de gris pour les images à compresser*

*Calcul de M, le bloc original est nivélé par la soustraction de 128*

##### Étape2: Décomposition de l'image

*Choisir la fonction ondelette (bior4.4)*

*Choisir le nombre d'itérations*

*Calcul des coefficients d'approximation*

*Calcul des coefficients de détails*

**Étape3 : Quantification ou seuillage**

*Pour les images de détails:*

*Choisir le seuil (inférieur ou égal au  $\min(cAi)$ ) pour les images de détail*

*Choisir les niveaux de détail sur lesquels la compression sera appliquée*

*Pour tout coefficient inférieur au seuil, remplacer par zéro*

*Pour l'image d'approximation :*

*Choisir la matrice de quantification*

*Calcul de la DCT de l'image d'approximation*

*Diviser les blocs  $8 \times 8$  de la matrice transformée par la matrice de quantification*

**Étape4 : Codage**

*Codage en zigzag*

*Encodage arithmétique ou Huffman*

**Étape5 : Reconstruction de l'image**

*Décodage:*

*Les vecteurs quantifiés*

*Pour l'image d'approximation:*

*Dé-quantification:*

*Multiplier les blocs de  $8 \times 8$  de la matrice décodée par la matrice de quantification inverse*

*Transformée inverse:*

*Calcul de la IDCT,  $\mathbf{D}^T = \mathbf{T}^T \mathbf{M} \mathbf{T}$*

*Pour les images de détails et l'image d'approximation:*

*Transformée inverse:*

*La fonction ondelette*

*Ajout de 128 pour le nivelement*

**Fin.**

La figure 6.22 est un schéma explicatif du diagramme bloc de la figure 6.21. On voit bien que le seuil 1 est appliqué sur les six images de détail de niveau 1 et de niveau 2 et que le seuil 2 est appliqué sur les autres trois images de détail de niveau 3. Aussi, un découpage en blocs de  $8 \times 8$  est appliqué sur l'image d'approximation de niveau 3 avant de subir la transformation DCT et l'opération de quantification.

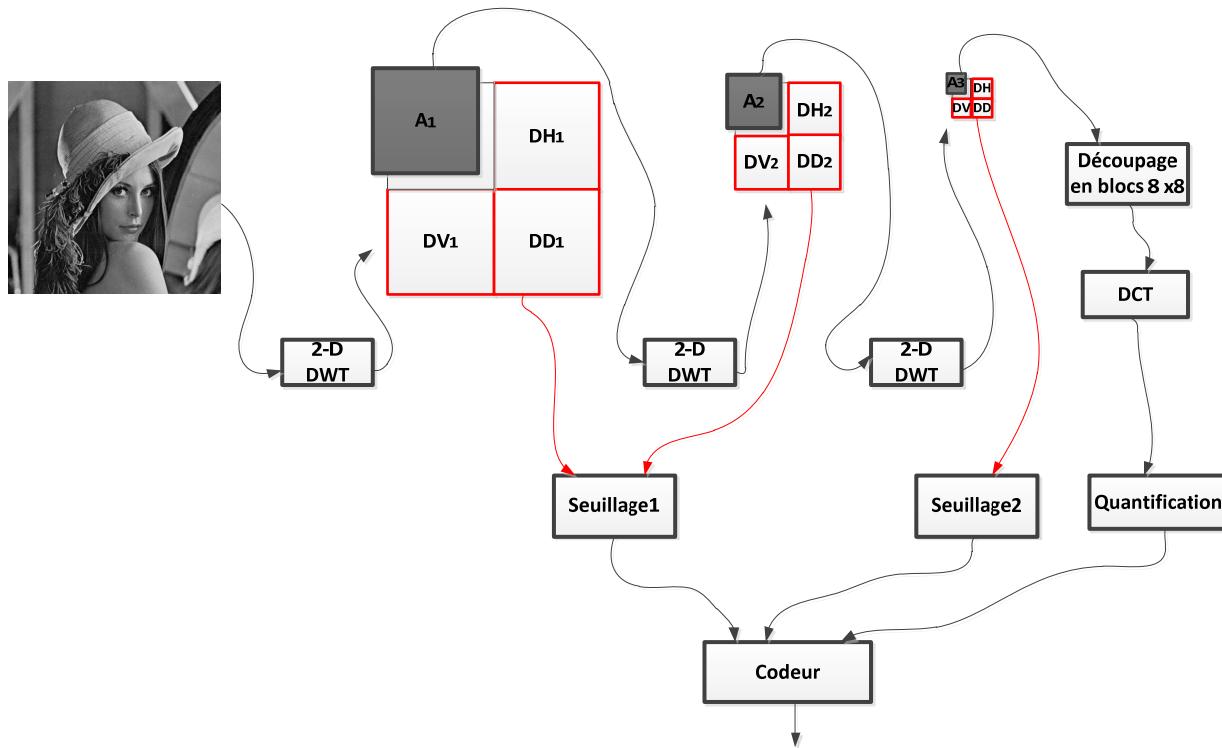


Figure 6. 22: Schéma bloc de la compression de l'image *Lena* par *DWT-améliorée-DCT*.

La figure 6.23 présente l'évaluation de la compression de l'image *wbarb* par les deux techniques: *DWT-amélioré* et *DWT-amélioré-DCT*. Les compressions de *wbarb* par *DWT-amélioré* et par *DWT-amélioré-DCT* sont visualisées par la courbe en ligne continue et en trait discontinu respectivement. Il est bien montré dans la figure que la compression par *DWT-amélioré* présente une meilleure performance pour un nombre de coefficients non nuls supérieur à 18% mais au-dessous de cette valeur la compression par la méthode hybride *DWT-amélioré-DCT* présente une amélioration de la performance supérieure à 0.5 dB.

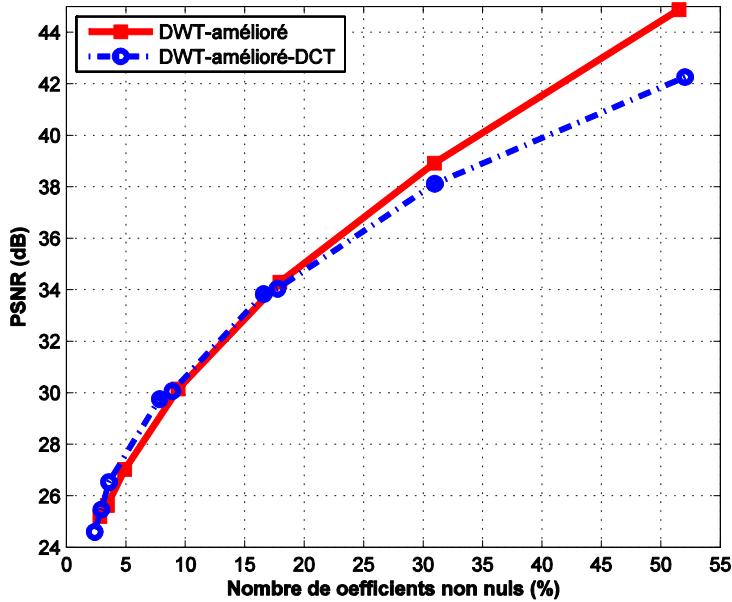


Figure 6. 23: Comparaison de la compression *DWT-amélioré* et de la compression *DWT-amélioré-DCT*

## 6.7 Conclusion

On peut conclure dans ce chapitre qu'en se basant sur nos images de test, nous avons présenté les points suivants :

L'activité spectrale de l'image influence la qualité de l'image compressée par DCT ou DWT. Lorsque le nombre de coefficients de la transformée utilisé pour la reconstruction est très faible, l'image présentant le plus d'activité spectrale aura le PSNR le plus grand (ceci étant dû aux particularités des images de test utilisées), mais si le nombre de coefficients de la transformée utilisé pour la reconstruction est grand, alors l'image présentant le moins d'activité spectrale aura le PSNR le plus grand.

Pour l'amélioration de la qualité de compression par DWT, nous avons appliqué deux seuils. Nous sommes arrivés à montrer des améliorations considérables surtout lorsque le nombre de coefficients utilisé pour la reconstruction de l'image est très faible, nous avons nommé cette méthode *DWT-amélioré*.

Finalement, une combinaison de deux techniques DCT et *DWT-amélioré* a été considérée et nous l'avons nommée *DWT-amélioré-DCT*. Les résultats montrent que la méthode hybride *DWT-amélioré-DCT* présente un PSNR supérieur à celui de la compression par *DWT-amélioré* pour un nombre de coefficients de la transformée inférieur à 18%.

## CHAPITRE 7 CONCLUSION

Le but de notre mémoire est d'aboutir à une compression de données d'images tout en gardant la qualité essentielle de l'image originale, c'est-à-dire, avoir un bon compromis entre PSNR et taux de compression.

Du moment que la méthode des transformées orthogonales est recommandée pour la compression de données et d'images en particulier, alors nous avons analysé la transformée discrète en ondelettes et la transformée discrète en cosinus qui sont les plus utilisées dans la compression d'images.

Comme la qualité de la compression d'images par DWT dépend du type d'ondelette utilisé et le type d'ondelette utilisé à son tour dépend des spécifications de l'image à compresser, pour cela quatre familles des ondelettes à support compact ont été étudiées : les ondelettes biorthogonales, les ondelettes Symlets, les ondelettes Coiflets et les ondelettes Daubechies. Ces ondelettes sont facilement implémentables par filtres RIF.

On se basant sur nos images de test nous avons montré que l'ondelette Bior4.4 donne bien un bon compromis entre un PSNR et un taux de compression acceptable dans la famille des ondelettes biorthogonales. Aussi, l'ondelette Db2 donne un bon PSNR alors que l'ondelette Db3 donne un bon taux de compression dans la famille des ondelettes Daubechies. De même, l'ondelette Sym2 donne un bon PSNR et l'ondelette Sym4 donne un bon taux de compression avec un PSNR acceptable dans la famille des ondelettes de symlets. Enfin, l'ondelette Coif1 donne un meilleur compromis entre un PSNR et un taux de compression acceptable, dans la famille des ondelettes coiflets.

Nous avons prouvé que les ondelettes Sym2 et Sym3 sont équivalentes aux ondelettes Db2 et Db3 respectivement du point de vue PSNR, mais du point de vue complexité de calcul, Db2 et Db3 sont moins complexes.

Pour la comparaison de la compression DWT et la compression DCT, plusieurs matrices de quantification avec un facteur de qualité variable sont nécessaires. Une formule a été modifiée et présentée, nous avons montré une amélioration du taux de compression par la compression

DCT. La matrice utilisée comme référence pour la comparaison est la matrice de luminance standard utilisée dans la norme JPEG.

Pour l'amélioration de la qualité de compression par DWT, nous avons appliqué deux seuils. Nous sommes arrivés à montrer des améliorations considérables surtout lorsque le nombre de coefficients utilisé pour la reconstruction de l'image est très faible, nous avons nommé cette méthode DWT-amélioré.

Finalement, une combinaison de deux techniques DCT et DWT-amélioré a été considérée et nous l'avons appelée DWT-amélioré-DCT. Les résultats montrent pour nos images de test que DWT-amélioré-DCT présente un PSNR supérieur à celui de la compression par DWT-amélioré pour un nombre de coefficients de la transformée inférieur à 18%.

### **Travaux futurs :**

Les résultats produits dans ce mémoire représentent une base fondamentale pour des travaux futurs dans l'architecture hardware. On peut citer :

- Les algorithmes de nos simulations peuvent être réalisés en implémentation hardware pour des travaux futurs.
- Les résultats de la compression peuvent être améliorés par l'ajout d'un codeur EBCOT.
- L'exploration de l'influence de l'ordre et du nombre d'itérations dans les résultats de compression d'image et de la vidéo.

## BIBLIOGRAPHIE

- [1] M. Corinthios, ``*Signals, systems, transforms and digital signal processing with Matlab(c)*,`` Taylor and Francis CRC press, 2009.
- [2] K. V. Madisetti, ``*The digital signal processing handbook. Digital signal processing fundamentals*,`` 2<sup>nd</sup> Ed., Boca Raton, FL: CRC Press, 2010.
- [3] R. C. Gonzalez, R. E. Woods, ``*Digital image processing*,`` 2<sup>nd</sup> Ed, prentice Hall, New jersey, 2002.
- [4] Y. Q. SHi, H. Sun, ``*Image and video compression for multimedia engineering*,`` 2<sup>nd</sup> Ed, CRC press, Taylor & Francis group, 2008.
- [5] R. N. Bracewell, ``*Fourier Analysis and imaging*,`` Kluwer Academic, Plenum Publishers, New York, 2003.
- [6] K. Sayood, ``*Introduction to data compression*,`` 3<sup>rd</sup> Ed, Morgan Kaufmann,2006
- [7] L..Brechet, M.-F. Lucas, C. Doncarli et D. Farina, ``Compression of biomedical signals with mother wavelet optimization and best-basis wavelet packet selection,`` *IEEE Trans on biomedical engineering*, Vol. 54, No.12, pp.2186-2192, Dec 2007
- [8] D. Jurafsky, J. H. Martin, ``*Speech and Language Processing*,`` 2<sup>nd</sup> Ed Pearson. Prentice. Hall, New Jersey, Jan. 2009.
- [9] R. L. Rabiner and R.W. Schafer, ``*Digital Processing of speech signals*,`` Englewood Cliffs, N.J.: Prentice Hall, 1978.
- [10] P. Stavroulakis, M. Stamp, ``*Handbook of Information and Communication Security*,`` Springer Verlag 2010.
- [11] T. S. Rappaport, ``*Wireless Communications: Principles and Practice*,`` Prentice Hall, 1996.
- [12] J. Proakis, ``*Digital communications*,`` McGraw-Hill, 2001.
- [13] J. G. Proakis, D. G. Manolakis, ``*Digital Signal Processing Principles, Algorithms, and Applications*,`` 4<sup>th</sup> Ed, Upper Saddle River, N.J. : Pearson Prentice Hall, 2007.
- [14] A. V. Oppenheim and R. W. Schafer, ``*Discrete time signal processing*,`` 2<sup>nd</sup> Ed, Upper Saddle River, N.J.: Prentice Hall, 1999.
- [15] N. Ahmed and K. R. Rao, ``*Orthogonal transforms for Digital Signal Processing*,`` Springer Verlag 1975.
- [16] A. D. Poularikas, ``*Transform and Applications Handbooks*,`` 3<sup>rd</sup> Ed Boca Raton Fl, CRC Press 2010.
- [17] M. J. Corinthios, ``Generalised Transform Factorization for massive parallelism,`` *IEEE proc, Vis. Image signal process*, Vol. 151, No.3, pp. 153-163, june 2004.

- [18] S. Mallat, ``*Une Exploration des Signaux en Ondelettes*,`` Édition de l'école polytechnique, France, 2000.
- [19] M. J. Corinthios, ``*3-D cellular arrays for parallel/cascade image/signal processing*,`` in Karpovsky, M. (Ed): ``*Spectral techniques and Fault detection*,`` Academic press, New York, USA, 1985.
- [20] V. Bristanak, P. Yip, K. R.Rao, ``*Discrete cosine and sine transforms*`` Elsevier, Amsterdam, 2007.
- [21] J.W. Cooley and J.W Tukey, ``An Algorithm for the Machine Calculation of Complex Fourier Series,`` *Math of comput*, Vol. 19, pp. 297-301, 1965.
- [22] M. J. Corinthios, ``A Fast Fourier Transform for High Speed signal Processing,`` *IEEE Trans, on computer*, Vol. C-20, No.8, pp. 843-846 , August 1971.
- [23] M. J. Corinthios, ``A time series-analyser,`` *Proc. Symp. Comput. Processing in com*, Vol. 19, Brooklyn, April 1969.
- [24] M. J. Corinthios, K. C. Smith, J. L. Yen ``A parallel Radix 4 fast Fourier Transform computer,`` *IEEE Trans. on comput*, Vol. C-24, No. 1, pp. 80-92, Jan 1975.
- [25] P.M. Grant, J.H. Collins, ``Introduction to electronic warfare,`` *IEE PROC.*, Vol. 129, Pt, F, No. 3, pp. 435-442, June 1982.
- [26] Y. Wang, Y. Tang, Y. Jiang, Jin-Gyun Chung, Sang-Seob Song, Myoung-Seob Lim, ``Novel Memory Reference Reduction Methods for FFT Implementations on DSP Processors,`` *IEEE Tran. On signal process*. Mai 2007.
- [27] R. V. L. Hartley, ``A More Symetrical Fourier Analysis Applied to Transmission Problems,`` *Proc. IRE*, Boston, 1990.
- [28] R. N. Bracewell, ``Discrete Hartley Transform,`` *J. Opt. Soc. Am*, Vol. 73, No. 12, pp. 1832-1835, 1983.
- [29] R. N. Bracewell, ``The Fast Hartley Transform,`` *Proc. IEEE*, Vol. 72, No. 8, pp. 1010-1018, 1984.
- [30] J. G. Liu, F. H. Y. Chan, F. K. Lam, H. F. Li, George S. K. Fung, ``Moment-based fast Discrete Hartley transform,`` *Elsevier, signal processing*, No. 83, pp. 1749-1757, 2003.
- [31] H.S. Hou, ``The fast Hartley transform algorithme,`` *IEEE Trans. Comput*, No. 36, pp. 147-156, Feb. 1987.
- [32] M. J. Corinthios, ``Optimal Parallel and pipelined Processing through a new class of Matrices with application to generalized spectral analysis,`` *IEEE Trans. on comput*, Vol. 43, No. 4, pp. 443-459, April 1994.

- [33] K. Karhunen, ``Über Lineare Methoden in der Wahrscheinlichkeitsrechnung,`` *Annales Academiae Scientiarum Fennicae*, Ser. A137, 1947.
- [34] M. Loève, ``Fonctions Aléatoires de Second Ordre,`` Processus stochastiques et Mouvement Brownien, P. Lévy, Ed. Hermann, Paris, 1948.
- [35] P.A. Chou, T. Lookabaugh, and R. M. Gray. Khaled Sayood, ``Optimal Pruning with application to tree-structured source coding and modeling,`` *IEEE Trans. on information theory*, No. 35: pp. 31-42, Janv 1989.
- [36] J.A. Saghri, A.J. Tescher, and J.T. Reagan, ``Terrain Adaptive Transform Coding of Multispectral Data,`` *In Proceedings International Conference on Geosciences and Remote sensing* (IGARSS '94), pp. 314-316. IEEE, 1994.
- [37] N. Ahmed, N. Natarajan, K. R. Rao, ``Discrete cosine transform,`` *IEEE Trans. Comput.*, Vol.23, pp. 90-92, 1974.
- [38] Z. Wang, ``Pruning the fast discrete cosine transform``*, IEEE Trans on communications*, Vol. 39, No.5, pp.640-643, 1991.
- [39] W. H. Chen, C. H. Smith, S. C. Fralick, ``A fast computational algorithm for the discrete cosine transform``*, IEEE Trans. on communications*, Vol. Com-25, pp. 1004-1009, 1977.
- [40] Z. Wang, ``Reconsideration of a fast computational algorithm for the discrete cosine transform``*, IEEE Trans. on communications*, Vol. Com-31, pp. 121-123, 1983.
- [41] R. Zelinski and P. Noll, ``Adaptive Block quantization of speech signals``*, Technical Report* No. 181, Heinrich Hertz Institut, Berlin, 1975.
- [42] N.S. Jayant and P. Noll, ``Digital Coding for Waveforms``*, Prentice Hall, Englewood Cliffs, NJ, 1984.*
- [43] H. P. Kramer, M. V. Mathews, ``A linear Coding for Transmitting a Set of Correlated Signals,`` *IEEE Trans. Info. Theory*, Vol. IT-2, pp. 41-46, 1956.
- [44] D. S. Taubman, M. W. Marcellin, ``JPEG2000: Image compression fundamentals, standards and practice,``*Kluwer Academic Publishers, Boston,2002.*
- [45] J. M. Shapiro, ``Embedded Image Coding Using Zerotrees of Wavelet coefficients,`` *IEEE Trans. Signal processing*, Vol.41, pp. 3445-3462, Dec. 1993.
- [46] A. Said and W. A. Pearlman, ``A new fast and efficient image codec based on set partitioning in hierarchical trees,`` *IEEE Trans. Circuits Syst. Video technol*, Vol. 6, pp. 243-250, june 1996.
- [47] M. Crouse et K.Ramchandran, ``Joint thresholding and quantization selection for transform image coding: Entropy-constrained analysis and application to JPEG,`` *IEEE Trans. Image Processing*, Vol.6, pp.285-297, Feb.1997.

- [48] Z. Xiong, O. Guleryuz et M. T. Orchard, ``A DCT-based embedded image coder,`` *IEEE Signal Processing lett*, Vol. 3, pp. 289-290, Nov. 1996.
- [49] P. S. Addison, ``*The Illustrated Wavelet Transform Handbook*,`` Institute of physics publishing, Bristol and Philadelphia, UK, 2002.
- [50] G. K. Wallace, ``The JPEG still Picture compression Standard,`` *Communication of the ACM*, 34(4): pp. 30-44, 1991.
- [51] S. Grgic, M. Grgic, B. zoko-cihlar, ``performance analysis of image compression using wavelets,`` *IEEE Trans. industrial electronics*, Vol. 48, No.3, June 2001.
- [52] L.Brechet, M.-F. Lucas, Christian Doncarli et Dario Farina, ``Compression of biomedical signals with mother wavelet optimization and best-basis wavelet packet selection,`` *IEEE Trans on biomedical engineering*, Vol. 54, No.12, pp.2186-2192, Dec 2007
- [53] M. Rabbani, R. Joshi, ``An overview of the JPEG 2000 still image compression standard`` , *Elsevier, Signal processing: Image communication*, pp-3-48, 2002.
- [54] I.Daubechies, ``The wavelet transform, time frequency localization and signal analysis,`` *IEEE trans. On inform theory*, Vol. 36, No.5, pp. 961-1005, Sep 1990.
- [55] S. Mallat, ``A theory for multiresolution signal decomposition: the wavelet decomposition`` , *IEEE Trans. On pattern analys and machine intell*, Vol.11, No.7, pp.674-693, July 1989.
- [56] U. S. Mohammed, W. M. Abdelhafiez, ``Image coding sheme based on object extraction and hybrid transformation technique,`` *International journal of engineering science and technology*, Vol. 2 No. 5, pp. 1375-1383, 2010.
- [57] O. Egger, A. Nicoulin, W. Li, ``Embedded zerotree based image coding with low decoding complexity using linear and morphological filter bank,`` *IEEE int conf on acoustics, speech and signal proc, NY*, pp. 2237-2240, 1995.
- [58] D.S. Taubman, `` EBCOT: Embedded block coding with optimized truncation,`` *technical report N1020R*, ISO/IEC JTC1/Sc29/WG1, October 1998.
- [59] G. K. Kharate, V. H. Patill, N. L. Bhale, ``Selection of mother wavelet for image compression on basis of nature of image`` , *journal of multimedia*, Vol. 2, No. 6, 2007.
- [60] M. K. Mandal, S. Panchanathan, T. Aboulnasr, ``Choice of wavelets for image compression`` , *lecture note in computer science*, Vol. 1133, pp. 239-249, 1996.
- [61] Z. Xiong, K. Ramchandran, M. T.Orchard and Y. Zhang, ``a comparative study of DCT and Wavelet based image coding,`` *IEEE Trans. Circuits and systems for video technology*, Vol. 9, No. 5, Aug. 1999.
- [62] A.N. Akansu, W. A. Serdijn, I. W. Selesnick, ``Emerging applications of wavelets: A review, `` *Elsevier Physical Communication*, Vol. 3, No. 1, pp 1-18, Mars 2010.

- [63] D.A. Huffman, ``A Method for the Construction of Minimum redundancy Codes, ``*Proc. IRE*, No. 40, pp. 1098-1101, 1952.
- [64] R.N. Bracewell, ``*The Fourier Transform and its application*, `` McGraw Hill, New York, 2000.
- [65] J. G. Proakis, Editor, ``*Wiley Encyclopedia of Telecommunication*, `` Vol2, Wiley-interscience, A John Wiley and sons publication, N.J, 2003.
- [66] J.Ziv, A. Lempel, ``A universal algorithm for sequential data compression, `` *IEEE Trans. Inform. Theory*. No. IT-23, pp. 337-343, 1977.
- [67] R. B. Welch, ``A technique for high-Performance data compression, `` *IEEE Trans. Comput.* No. 17, pp. 8-19, 1984.
- [68] S. A. Khayam, ``The discrete Cosine Transform (DCT): theorie and application, `` *Information théory and codage, ECE*. 802-602 séminaire1-CEE, Mars 2003.
- [69] H. Bekkouche, ``*Synthese de Banc de Filters Adaptés, application à la Compression des Images* ``, Thèse de doctorat, Paris, Dec 2008.
- [70] A. M. Eskicioglu, P.S. Fasher, ``Image quality measures and their performance, `` *IEEE Trans on communications*, Vol. 43, No. 12, pp 2959-2965, Dec 1995.
- [71] A. Al-haj, ``Combined DWT-DCT Digital image Watermarking, `` *journal of computer science* 3(9): 740-746, 2007.
- [72] <http://www.techno-science.net/?onglet=articles&article=9&page=8>
- [73] T. Acharya, A. K. Ray, ``*Image processing: principles and applications* ``, Wiley-interscience, A John Wiley and sons publication, N.J, 2005.
- [74] G.A. Ramash, G. Haitao, Burrus, C. Sidney, ``*Introduction to wavelets and wavelet transforms* ``, Upper Saddle River, N.J. : Prentice Hall, c1998.

## ANNEXE 1 – Programme de compression

**Compression DWT de l'image *rami* avec l'ondelette Sym3 :**

```
%Décomposition de l'image
Load rami
smap=size(map,1);
sx=size(X);
%%% Coefficients d'approximation et de détails niveau 1
[cA1,cH1,cV1,cD1]=dwt2(X,'sym3');
% construction de l'approximation niveau 1
A1=upcoef2('a',cA1,'sym3',1);
% construction des coefficients niveau 2 et 3.
[C,S]=wavedec2(X,3,'sym3');
%%%Coefficients d'approximations de niveau2 et 3
cA2=appcoef2(C,S,'sym3',2);
cA3=appcoef2(C,S,'sym3',3);
%%%Coefficients de détails niveau 2 et 3
[cH2,cV2,cD2]=detcoef2('all',C,S,2);
[cH3,cV3,cD3]=detcoef2('all',C,S,3);
% Approximations et détails
A2=wrcoef2('a',C,S,'sym3',2);
H2=wrcoef2('h',C,S,'sym3',2);
V2=wrcoef2('v',C,S,'sym3',2);
D2=wrcoef2('d',C,S,'sym3',2);
H1=wrcoef2('h',C,S,'sym3',1);
V1=wrcoef2('v',C,S,'sym3',1);
D1=wrcoef2('d',C,S,'sym3',1);
A3=wrcoef2('a',C,S,'sym3',3);
H3=wrcoef2('h',C,S,'sym3',3);
V3=wrcoef2('v',C,S,'sym3',3);
D3=wrcoef2('d',C,S,'sym3',3);
% tracer des images
figure, colormap(map),
subplot(3,4,1);image(wcodemat(A1,smap));
title('Approximation A1'),
subplot(3,4,2);image(wcodemat(H1,smap));
title('Horizontal detail H1'),
subplot(3,4,3);image(wcodemat(V1,smap));
title('Vertical detail V1'),
subplot(3,4,4);image(wcodemat(D1,smap));
title('diagonal detail D1'),
subplot(3,4,5);image(wcodemat(A2,smap));
title('Approximation A2'),
subplot(3,4,6);image(wcodemat(H2,smap));
title('Horizontal detail H2'),
subplot(3,4,7);image(wcodemat(V2,smap));
title('Vertical detail V2'),
subplot(3,4,8);image(wcodemat(D2,smap));
title('diagonal detail D2'),
subplot(3,4,9);image(wcodemat(A3,smap));
title('Approximation A3'),
subplot(3,4,10);image(wcodemat(H3,smap));
title('Horizontal detail H3'),
subplot(3,4,11);image(wcodemat(V3,smap));
title('Vertical detail V3'),
subplot(3,4,12);image(wcodemat(D3,smap));
title('diagonal detail D3'),
%%%%%%%%% PHASE DE RECONSTRUCTION %%%%%%%%%%%%%%
```

```

R1(1,:)=cA3(:);
R2(1,:)=cH3(:);
R3(1,:)=cV3(:);
R4(1,:)=cD3(:);
R5(1,:)=cH2(:);
R6(1,:)=cV2(:);
R7(1,:)=cD2(:);
R8(1,:)=cH1(:);
R9(1,:)=cV1(:);
R10(1,:)=cD1(:);
TY=min(min(cA3));
sz=size(R8(:));
for i=1:sz
    if abs(R8(i))<abs(TY)
        R8(i)=0;
    else R8(i)=R8(i);
    end
    if abs(R9(i))<abs(TY)
        R9(i)=0;
    else R9(i)=R9(i);
    end
    if abs(R10(i))<abs(TY)
        R10(i)=0;
    else R10(i)=R10(i);
    end
end
sz1=size(R5(:));
for i=1:sz1
    if abs(R5(i))<abs(TY)
        R5(i)=0;
    else R5(i)=R5(i);
    end
    if abs(R6(i))<abs(TY)
        R6(i)=0;
    else R6(i)=R6(i);
    end
    if abs(R7(i))<abs(TY)
        R7(i)=0;
    else R7(i)=R7(i);
    end
end
%%% J1 sont les images de niveau3 ou itération3
J1=[R1 R2 R3 R4];
%%% J12 sont les images de niveau2 ou itération2
J2=[R5 R6 R7];
%%% J3 sont les images de niveau1 ou itération1
J3=[R8 R9 R10]; % Diagonal 3 est mise à zéro
%%% Codage en zigzag
J=[J1 J2 J3];
%J=[R1 R2 R3 R4 R5 R6 R7 R8 R9 zeros(size(R10)) ];
%%% Reconstruction de l'image avec le nouveau vecteur ( sans diagonal 3)
X0=waverec2(J,S,'sym3');
figure,
image(X), colormap(map),
title('Image originale'),
figure
image(X0), colormap(map),
title('Image compressée à % avec sym3'),

```