

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE



جامعة محمد بوضياف - المسيلة

UNIVERSITE MOHAMED BOUDIAF - M'SILA

كلية التكنولوجيا

FACULTE DE TECHNOLOGIE

قسم الإلكترونيك

DEPARTEMENT D'ELECTRONIQUE



## **MEMOIRE DE MASTER**

DOMAINE : SCIENCES ET TECHNOLOGIE

FILIERE : ELECTRONIQUE

OPTION : SYSTEMES DE TELECOMMUNICATIONS NUMERIQUES

### **THEME**

**Application de la technique SPIHT  
pour la compression des images médicales**

*Proposé et dirigé par :*

Dr. BOUKHENNOUFA N.

*Réalisé par :*

BOUZIDI Rima

*N° D'ordre: STN08*

**PROMOTION: JUIN 2016**

# **Application de la technique SPIHT pour la compression des images médicales**

**Présenté par :** Rima BOUZIDI

**Encadré par :** Dr. N. BOUKHENNOUFA

## **Résumé**

*L'imagerie médicale est un domaine en plein essor, du fait du développement des technologies numériques. Elle permet une investigation de plus en plus fine des organes humains grâce à la mise à disposition de systèmes de radiologie de plus en plus performants. La contrepartie réside dans une quantité de données générée considérable qui peut rapidement saturer les systèmes conventionnels de transmission et de stockage.*

*L'objectif de ce travail est de développer un algorithme de compression très efficace nommé "SPIHT" (Set Partitioning In Hierarchical Trees). Ce dernier est basée sur les transformées en ondelettes.*

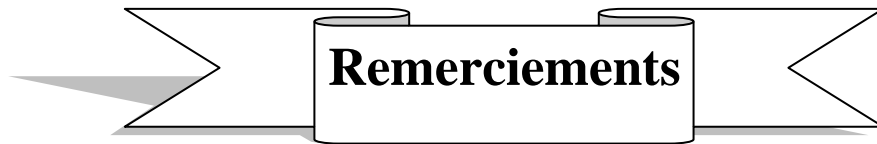
**Mots clés :** Compression, Images médicales, Transformées en ondelettes, SPIHT.

## **Abstract**

*Medical imaging is a growing field , due to the development of digital technologies. It allows investigation of increasingly fine human bodies through the provision of radiology systems more efficient . The consideration lies in a considerable amount of generated data that can quickly saturate conventional systems of transmission and storage.*

*The objective of this work is to develop a highly efficient compression algorithm called "SPIHT" (Set Partitioning In Hierarchical Trees ) . The latter is based on wavelet transforms.*

**Keywords :** Compression, Medical images, Wavelet transform, SPIHT.



# Remerciements

*Tout d'abord, je remercie le bon DIEU, de m'avoir donné le courage pour terminer ce travail.*

- ❖ *Mes vifs remerciements vont à mon encadreur le Dr. Nabil BOUKHENNOUFA pour son encadrement, son aide, sa patience et ses conseils très précieux tout au long de ce travail.*
- ❖ *Je voudrais exprimer mes remerciements aux membres du jury en acceptant de juger ce travail et d'y apporter leurs cautions.*
- ❖ *Mes parents, mes sœurs et toute ma famille qui m'a fourni un support tout au long de mes études.*
- ❖ *Je tiens aussi à remercier tous qui de près ou de loin m'ont aidé et encouragé pour achever travail.*

Bouzidi Rima

## *DEDICACES*

Je dédie ce modeste travail, en espérant qu'il soit à  
la hauteur de leurs attentes :

✓ A mes chers parents, A mes sœurs et mes neveux

Adam et Youcef.

✓ A ma nièce Farah.

✓ A toute ma famille.

✓ A toute mes amis, mes camarades et mes collègues  
du travail.

✓ À tous ceux qui m'ont soutenu et m'ont encouragé  
depuis mes premiers pas dans cette vie.

Rima BOUZIDI

# Table de matière

Liste des figures	
Liste des tableaux	
Introduction Générale.....	1

## Chapitre I : Généralités sur les techniques de compression

I.1.Introduction .....	4
I.2. Définition de la compression des images .....	4
I.3.l'objectif de compression.....	5
I.4. Schéma fonctionnel de la compression des images.....	5
I.4.1 .Transformation.....	5
I.4.2. Quantification.....	6
I.4.3.Codage.....	6
I.5 .Méthodes de compression sans perte.....	6
I.5.1.Entropie d'une source d'information.....	7
I.5.2. Redondance .....	7
I.5.3. Méthode de Shannon-Fano .....	7
I.5.4. Méthode de Huffman .....	8
I.5.5.Méthode de codage RLE.....	10
I.6. Méthodes de compression avec perte.....	12
I.6.1.Méthodes directes.....	13
I.6.1.1.Quantification.....	13
I.6.1.1.1.Quantification scalaire.....	13
I.6.1.2.2.Quantification vectorielle .....	14
I.6.2.Méthode par transformée.....	14
I.6.2.1.Transformée de Fourier Discrète DFT.....	14
I.6.2.2.Transformée en Cosinus Discrète DCT.....	15
I.7. Conclusion .....	17

## Chapitre II : Transformée en ondelettes

II.1 .Introduction .....	18
II.2. Fréquence locale .....	19
II.3. Représentations temps -fréquence et temps- échelle .....	21

II.3.1. Représentations temps-fréquence et temps-échelle .....	21
II.4. Généralités sur les ondelettes .....	23
II.4.1. Pourquoi les ondelettes ? .....	23
II.5. Définition d'une ondelette .....	24
II.5.1. Transformée en ondelette continue (CWT) .....	26
II.5.2. Transformée en ondelette discrète(DWT) .....	28
II.6. Extension de transformée en ondelettes pour les images .....	33
II.7. Coefficient d'ondelettes 2D .....	33
II.8. Compression d'images fixes par ondelettes classiques .....	34
II.9. Conclusion .....	36

### Chapitre III : Algorithme SPIHT

III.1. Introduction .....	37
III.2. Algorithme de codage SPIHT .....	37
III.3. Méthode de codage .....	40
III.3.1. Codage binaire .....	41
III.4. Arbres .....	41
III.4.1. Structure d'arbres .....	42
III.4.2. Arbres de Zéros .....	42
III.5. Initialisation .....	43
III.6. Exemple de SPIHT .....	45
III.7. Décodage SPIHT .....	47

### Chapitre IV : Résultats et Discussions

IV.1. Introduction .....	48
IV.2. principe de compression/décompression .....	48
IV.3. paramètres de validation .....	49
IV.3.1. taux de compression .....	49
IV.3.2. MSE .....	49
IV.3.3. PSNR .....	50
IV.4. Base de données des images médicales .....	50
IV.4.1. Images médicales en niveaux de gris .....	50
IV.4.1.1. Images de test de la base I .....	51

IV.4.1.1.Image de test de la base II.....	52
IV.5.Résultats et discussions.....	52
IV.5.1.choix de l'ondelette.....	52
IV.5.2.Niveau de décomposition.....	55
IV.5.3.Taille de l'image.....	57
IV.6.Conclusion.....	59
IV.6.Conclusion.....	59
Conclusion générale.....	60

# Liste des abréviations

**DFT:** Discrete Fourier Transform

**DCT:** Discrete Cosine Transform

**DWT:** Discrete Wavelet Transform

**CWT:** Continue Wavelet Transform

**JPEG:** Joint Photographic Experts Group

**JPEG 2000:** Joint Photographic Experts Group 2000

**PSNR:** Pick Signal to Noise Ratio (PRSB en français: Pick du Rapport Signal sur Bruit)

**SNR :** Signal to Noise Ratio

**FFT :** Fast Fourier Transform

**RLE:** Run-Length Encoding

**CR:** Compression Ratio

**EQM:** Erreur Quadratique Moyenne (MSE en anglais: Mean Square Error)

**RGB:** Red Green Blue

**SPIHT:** Set Partitioning In Hierarchical Trees

**EZW:** Embedded Zerotree Wavelet coding

**TFC:** Taille de fichier Compressé

**LPS:** La Liste des Coefficients Significatifs (LCS)

**LIP:** La Liste des Coefficients Non significatifs (LCN)

**LIS:** La Liste des Ensembles Non significatifs (LEN)

**IRM:** Imagerie par Résonance Magnétique

**QV:** Quantification Vectorielle

**QS:** Quantification Scalaire

**VHS:** Visuel Human System

# Liste des tableaux

**Tableau I .1** : Exemple de codage Shannon-Fano.

**Tableau .I.2:** Code de Huffman pour l'alphabet  $\Sigma$ .

**Tableau.III.1** : Tableau montrant étape par étape le déroulement du premier passage de l'algorithme SPIHT 2D.

**Tableau .IV.1** : compression de l'image appareil respiratoire taille 512 \*512 au niveau 1.

**Tableau .IV.2** : compression de l'image appareil respiratoire taille 512 \*512 au niveau 2.

**Tableau .IV.3** : compression de l'image appareil respiratoire taille 512 \*512 de niveau 3.

**Tableau .IV.4** : compression de l'image appareil respiratoire taille 512 \*512 de niveau 4.

**Tableau .IV.5** : compression de l'image appareil respiratoire taille 512 \*512 de niveau 5.

**Tableau .IV.6** : compression de l'image appareil respiratoire taille 512 \*512 de niveau 6.

**Tableau .IV.7** : compression de l'image genou taille 128 \*128 de niveau 6 .

**Tableau .IV.8** : compression de l'image appareil respiratoire taille512\*512 de niveau 6.

**Tableau .IV.9** : compression de l'image appareil respiratoire taille256\*256 de niveau 6.

**Tableau .IV.10** : compression de l'image appareil respiratoire taille128\*128 de niveau 6.

**Tableau .IV.11** : compression de l'image poitrine taille128\*128 de niveau 6.

# Liste des Figures

**Figure .I. 1 :** Diagramme de compression d'image pour la transmission ou stockage.

**Figure .I. 2:** Schéma fonctionnel de la compression des images.

**Figure .I. 3:** Algorithmes de compression sans perte les plus utilisés.

**Figure.I.4:** Arbre de Huffman pour l'alphabet  $\Sigma$ .

**Figure .I.5:** codage RLE.

**Figure .I.6:** Méthodes de compression avec perte.

**Figure .I.7 :** Distribution des fréquences de la DCT.

**Figure .I.8 :** Fonctions de base de la DCT-2D.

**Figure. II.1:** Pavages temps-fréquences associées aux représentations de Dirac et de Fourier.

**Figure. II.2:** Pavages du plan temps-fréquence pour la représentation temps-fréquence et la représentation temps-échelle.

**Figure .II. 3 :** *Ondelette simple « Dérivée seconde d'une Gaussienne ».*

**Figure .II.4. :** DCT 8x8 d'une image montrant : le coefficient DC, les 63 coefficients AC, les fréquences horizontales, verticales, et diagonales, ainsi que le balayage par ordre zig-zag (couleur bleu ciel).

**Figure .II. 5 :** Représentation des 64 composantes fréquentielles formant la base de décomposition en DCT 8x8.

**Figure .II. 6 :** Processus de décomposition de l'image selon la transformation en ondelette:(j+1) désigne le niveau de décomposition.

**Figure. II.7:** Représentation des coefficients d'approximation et de détails.

**Figure .II.8:** décomposition de l'image IRM "coupe sagittale" via une AMR, par la transformée en ondelettes de CDF9/7. de cerveau sur 3 niveaux de résolution.

**Figure .II.9 :** Détection des contours et singularités ponctuelles par les ondelettes 2D : (a) Image originale, (b) coefficients détails verticaux, (c) coefficients détails horizontaux, (d) coefficients détails diagonaux.

**Figure. II.10 :** Illustration de la directivité du transformée en ondelettes séparables 2D (décomposition d'une forme élémentaire).

**Figure .III.1:** Exemples de descendance parent-fils dans le cas SPIHT, le pixel désigné par (\*) n'accepte pas de descendants.

**Fig.III.2 :** Terminologie SPIHT pour les descendants.

**Figure .III3 :** Modèle de dépendances inter-bandes pour SPIHT.

**Fig.III.4 :** Exemple de SHAPIRO.

**Figure (IV.1): Schéma de** Principe d'un système de compression des images médicales.

**Figure. IV.2 :** Base de Données des images médicales.

**Figure. IV.3 :** Les images test de la base II de taille 128 x 128.

**Figure .IV.4 :** Les images test de la base I de taille 512 x 512.

Figure. IV.5.: image originale appareil respiratoire 512 x512.

Figure. IV.6.: image originale poitrine 512 x512.

**Figure. IV.7.:** image originale genou 128 x128.

**Figure. IV.8.:** transformée de l'ondelette haar pour le codeur SPIHT sur l'image Appareil respiratoire 512x512.

**Figure. IV.9.:** transformée de l'ondelette Db pour le codeur SPIHT sur l'image Appareil respiratoire 512x512.

**Figure. IV.10.:** transformée de l'ondelette Sym pour le codeur SPIHT sur l'image Appareil respiratoire 512x512.

**Figure. IV.11.:** transformée de l'ondelette Bior pour le codeur SPIHT sur l'image Appareil respiratoire 512x512.

**Figure. IV.12.:** Comparaison entre les quatre types d'ondelettes pour le codeur SPIHT sur l'image appareil respiratoire 512x512.

**Figure. IV.13 :** Résultats de l'image appareil respiratoire 512 \*512 reconstruite par SPIHT au niveau 6.

**Figure. IV.14 :** Résultats de l'image genou 128 \*128 reconstruite par SPIHT au niveau 6.

**Figure. IV.15 :** Résultats de l'image appareil respiratoire reconstruite par SPIHT à la taille 512.

**Figure. IV.16 :** Résultats de l'image poitrine reconstruite par SPIHT à la taille 512.

## Introduction générale

L'imagerie médicale est un domaine en plein essor, du fait du développement des technologies numériques. Elle permet une investigation de plus en plus fine des organes humains grâce à la mise à disposition de systèmes de radiologie de plus en plus performants.

La contrepartie réside dans une quantité de données générée considérable qui peut rapidement saturer les systèmes conventionnels de transmission et de stockage. L'augmentation croissante et continue des capacités de stockage apporte une réponse partielle à ce problème mais demeure la plupart du temps insuffisante.

La nécessité de compresser les images apparaît donc aujourd'hui incontournable. De plus, la compression présente un intérêt évident pour la transmission des images afin de réaliser un télédiagnostic qui est actuellement parmi les secteurs potentiels en télémédecine. L'utilisation massive des modalités numériques en imagerie médicale (IRM, Scanner X, médecine nucléaire, etc...) engendre aujourd'hui des volumes de données de plus en plus importants (Méga octets à Téra octets/ans).

Le problème devient encore plus critique avec la généralisation de séquences tridimensionnelles. Alors il est nécessaire d'utiliser des images compressées afin de limiter le volume de données à stocker et à transmettre. Il est fort utile de noter que la compression d'une image numérique permet de réduire le nombre de bits qu'elle occupe.

Pour atteindre l'objectif assigné, il faut assurer deux tâches : La réduction de la redondance n'introduisant pas de pertes, et la représentation approximative de l'information contenue dans l'image introduisant une perte mais nous verrons qu'il faudra minimiser cette perte sous peine de modifier le contenu informationnel des images compressées.

Actuellement, la compression dans un service de radiologie est toujours effectuée sans perte quand elle existe car elle constitue à ce jour le seul type de compression toléré par les médecins. En effet, la compression sans perte garantit l'intégrité des données et permet d'éviter les erreurs de diagnostic. Cependant, ce type de compression n'offre pas de réduction significative du volume de ces données. Dans ce contexte, la compression "avec pertes" maîtrisées peut être la

réponse la plus appropriée, à condition bien entendu que les pertes n'affectent pas la qualité des images pour l'usage régulier des praticiens. [1]

Hier encore inenvisageable, l'idée d'une compression avec pertes semble aujourd'hui de mieux en mieux acceptée par les médecins, comme en témoigne par exemple, l'American College of Radiology (ACR) qui estime que les techniques de compression avec pertes peuvent être utilisées à des taux raisonnables, sous la direction d'un praticien qualifié, sans aucune réduction significative de la qualité de l'image pour le diagnostic clinique. L'un des principaux enjeux de ce manuscrit est donc de proposer une méthode de compression d'images médicales avec pertes efficace et acceptable visuellement pour les médecins.

Les méthodes actuelles de compression pour les images médicales sans et avec pertes les plus efficaces pour les images médicales exploitent la corrélation des images afin d'améliorer la performance de compression. Plusieurs algorithmes de compression utilisant les ondelettes ont été proposés. L'une des applications les plus connues en compression d'image est l'algorithme SPIHT qui a amélioré le principe de l'EZW en proposant un codage plus efficace de l'information de signifiante qui code la position des coefficients énergétiques [2].

Ce mémoire nous nous intéressons à la compression d'images médicales en niveaux de gris, Notre méthode consiste à appliquer l'algorithme SPIHT pour cette compression tels que cette dernière basé sur la transformé en ondelettes.

Notre mémoire de Master est organisée selon quatre chapitres :

- **Le premier chapitre** : nous décrivons le principe général de la compression, (transformation, quantification et codage) ; ensuite, nous décrivons la classification des méthodes de codage qui peuvent être regroupées en deux sous-classes :
- Les méthodes sans pertes d'information : codage de shannon-fano, codage de Huffman et codage arithmétique.
- Les méthodes avec pertes d'information : quantification vectorielle, méthodes prédictives, méthodes hybrides, JIBG, codage par transformée et codage imbriqué.

- **Le deuxième chapitre** : L'objet de ce chapitre est de présenter les ondelettes que nous allons utiliser dans ce travail. Nous les situons dans le panorama plus large de l'analyse temps-fréquence, puis nous rappellerons leurs propriétés fondamentales.

- **Le troisième chapitre** : Dans ce chapitre nous commençons par présenter la méthode SPIHT. Ensuite nous mettons en œuvre cet algorithme et nous étudions ses performances à l'aide de plusieurs images testes. Les images que nous avons utilisées pour nos tests et pour l'évaluation de cet algorithme ont été choisies selon leurs caractéristiques. Ceci dans l'objectif d'évaluer les performances de cette méthode selon plusieurs cas de figures.

- **Le quatrième chapitre** : expérimentation et résultats trouvés.

Enfinement On termine notre mémoire par une conclusion générale et les perspectives envisagées concernant notre travail.

## I.1. Introduction

C'est avec l'apparition des ordinateurs et surtout avec Internet, que les images devenues prépondérantes. En effet, quoi de plus précis qu'une image ?

Il faudrait plusieurs pages de texte pour décrire une simple photographie, ou un schéma...etc. Cependant, si dans la rapidité d'information l'image détrône facilement le texte, celle-ci devient vite lourde de données, donc excessivement longue à transmettre et à traiter .voilà pourquoi depuis quelques années, les centres de recherche en informatique dépensent de nombreuses heures sur des algorithmes de compression.

Afin de limiter la taille ou le poids d'une image, nous devons la compresser, c-à-dire éliminer les informations inintéressantes ou redondantes.il existe de nos jours plus d'une vingtaine de formats de compression, spécifiquement dans la compression d'image (GIF, JPEG, BMP ...), ayant chacun leur propre méthode de codage, en commuant plusieurs algorithmes, mais tous sont complémentaires.

Presque chaque format de fichier incorpore l'une ou l'autre méthode de compression l'enjeu de la recherche sur compression d'image est de trouver un moyen de diminuer la taille d'une image, tout en essayant de limiter la dégradation due à la compression.

Nous allons le long de ce chapitre étudier les différentes méthodes de compression sans et avec perte. [3]

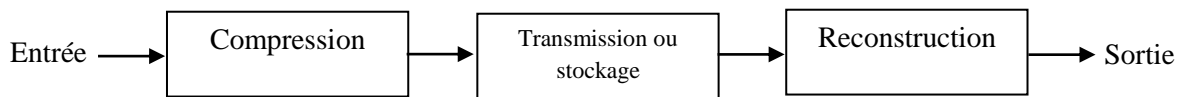
## I.2. Définition de la compression des images

La compression est l'action utilisée pour réduire la taille physique d'un bloc d'information, en compressant des données .on peut placer plus d'information dans la même espace de stockage ou utiliser moins de temps pour le transfert au travers d'un réseau téléinformatique .parce que généralement les images requièrent une place importante, la compression est devenue une part intégrante des données graphique .presque tous les formats des fichier graphiques utilisent l'une ou l'autre méthode de compression .on rencontre souvent la méthode de compression des données comme étant une partie de l'encodage de données au même titre que le cryptage des données (cryptographie) et la transmission de données .un compresseur ,comme son l'indique compresse les données alors qu'un décompresseur reconstruit les données originelles.[4]

Il est évident qu'un décompresseur ne peut opérer que connaissant l'algorithme de compression utilisé.la plupart des méthodes de compression visent à éliminer la redondance.

Il ya plusieurs types de redondances :

- ❖ La redondance spatiale entre deux pixels ou blocs voisins dans une image.
- ❖ La redondance temporelle entre images successives dans une séquence vidéo. [5]



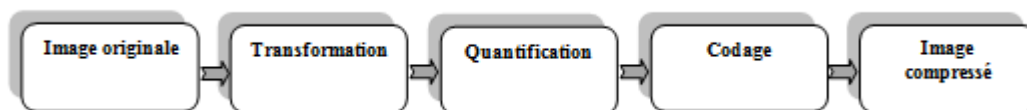
**Figure I.1 :** Diagramme de compression d'image pour la transmission ou stockage.

### I.3. Objectif de la compression

Le but de la compression des images est de réduire le nombre moyen de bits par pixel nécessaires à sa représentation. Il est possible dans une certaine limite de réduire ce nombre sans perte d'information. Au-delà, il est nécessaire d'élaborer des algorithmes de compression irréversibles (avec pertes) induisant une distorsion pas ou peu visible dans les conditions normales d'observation des images. [6]

### I.4. Schéma fonctionnel de la compression des images

Le schéma fonctionnel de la compression est présenté dans la figure I.2 ci-dessous :



**Figure I.2 :** Schéma fonctionnel de la compression des images.

#### I.4.1. Transformation

La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduisent une corrélation très forte sur l'image. On essaie donc de tirer parti de cette corrélation, pour réduire le volume d'information en effectuant une opération de décorrélation des pixels.

La décorrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible.

## I.4.2. Quantification

La quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. Elle représente une étape clé de la compression. Elle approxime chaque valeur d'un signal par un multiple entier d'une quantité  $q$ , appelée quantum élémentaire ou pas de quantification. Elle peut être scalaire ou vectorielle. Un des résultats fondamentaux des travaux de Shannon concernant la relation : (débit /distorsion) montrent que l'on obtient de meilleures performances en utilisant la quantification vectorielle.

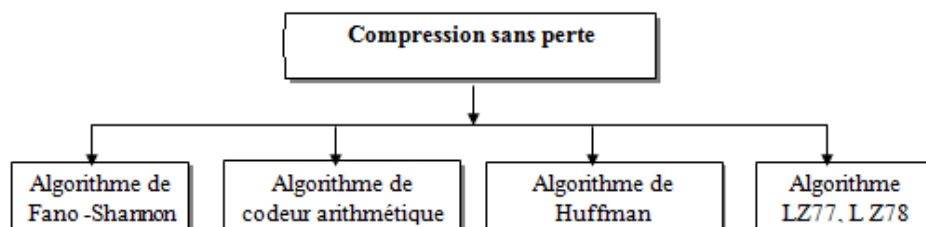
## I.4.3. Codage

Une fois les coefficients quantifiés, ils sont codés. Un codeur doit satisfaire a priori les deux conditions suivantes : Unicité : deux messages différents ne doivent pas être Codés de la même façon, Déchiffrable : deux mots de codes successifs doivent être distingués sans ambiguïté. A partir de ce schéma fonctionnel qui nous avons explicité, nous passons à la classification des méthodes de compressions d'image numériques. [7]

## I.5. Méthodes de compression sans perte

Cette catégorie des méthodes est caractérisée par la présentation exacte de l'information d'origine .soit une source d'information  $X = \{ x_1, x_2, \dots, x_n \}$ . Par application d'un algorithme de compression sans perte, on aboutira à un nouveau vecteur  $Y = \{ y_1, y_2, \dots, y_m \}$  de celle du vecteur composé  $Y$ .

La figure I.3 montre les compresseurs sans perte les plus utilisés en littérature spécialisée du domaine :



**Figure I.3 :** Algorithme de compression sans perte les plus utilisés.

### I.5.1. Entropie d'une source d'information

Une source d'information peut être un fichier de données financières, une image, un signal Electrocardiogramme, ...etc.

Cette source peut être modélisée par un vecteur  $X = \{x_1, x_2, x_3, \dots, x_n\}$  où  $x_i$  est une réalisation d'un code issu d'un alphabet  $\{c_1, c_2, c_3, \dots, c_M\}$ . L'entropie donnant la quantité d'information moyenne est définie par :

$$H(x) = - \sum_{i=1}^M P(x = C_i) \log_2 P(x = C_i) \quad (\text{I.1})$$

Où  $P(x = C_i)$  est la probabilité d'occurrence du code  $C_i$  dans la source d'information  $X$ .

Il est à noter que la source d'information dont les codes de l'alphabet sont équiprobables c'est-à-dire,  $P(x = c_1) = P(x = C_2) = P(x = C_3) = \dots = P(x = C_M) = 1/M$ , donne l'entropie maximale [8]. Il est notable que la compression sans perte de telle source d'information mène à l'échec.

Donc, il est préférable que les fréquences d'occurrences des différents codes  $c_i$  soient distinctes et même concentrées.

### I.5.2. Redondance [10]

Une image numérique présente la particularité de posséder des corrélations importantes entre les pixels voisins. Cette corrélation est vue comme une redondance des informations pertinentes. La redondance peut être de deux natures : **la redondance spatiale** qui apparaît directement entre les pixels voisins de l'image originale et **la redondance spectrale** qui est liée aux fréquences et qui est acquise avec les transformations de domaines. La redondance dans le domaine spatial n'est pas facilement identifiable et généralement ne fournit pas toujours un bon taux de compression. Il est donc nécessaire de faire une transformation pour obtenir une décorrélation de l'information spatiale et un groupement d'énergie fréquentielle.

### I.5.3. Méthode de Shannon-Fano

Le premier codage élaboré pour aboutir à des codes de tailles variables et menant à un codage moyen satisfaisant, a été celui de Shannon-Fano. Nous démarrons avec  $N$  symboles avec leurs probabilités (fréquence d'occurrences) connues, les symboles sont arrangés d'une façon descendante suivant leurs probabilités. L'ensemble des symboles est divisé en deux sous-ensembles qui ont la même (ou presque la même) probabilité. Tous les symboles appartenant à un sous-ensemble auront un code assigné 0 et les symboles de l'autre auront un code 1.

Chacun de ces deux sous-ensembles est subdivisé de la même manière récursivement à deux sous-ensembles et le bit assigné est déterminé de la même façon. Quand un sous-ensemble

contient juste deux symboles, leurs codes sont distingués par l'ajout d'un bit pour chacun (0 pour le premier symbole et 1 pour le deuxième symbole).

Un exemple d'illustration est donné par le tableau (I.1). La sous-d'information (ou le fichier) contient 7 codes de probabilités {0.05, 0.10, 0.10, 0.15, 0.15, 0.20, 0.25}.

Probabilité d'occurrence	étapes	Résultat de codage final (bit)
1. 0.25	1 1	11
2. 0.20	1 0	10
3. 0.15	0 1 1	011
4. 0.15	0 1 0	010
5. 0.10	0 0 1	001
6. 0.10	0 0 0 1	0001
7. 0.05	0 0 0 0	0000

**Tableau I.1** : Exemple de codage Shannon-Fano

Le codage moyen de cet exemple est 2.7 bits /symbole. C'est un résultat satisfaisant proche de celui de l'entropie donnant un codage de 2.67 bits /symbole. [11]

#### I.5.4. Méthode de Huffman

Introduit en 1952 par David A. Huffman dans l'article 'A Method for the Construction of Minimum-Redondance Codes [10], le codage de Huffman est utilisé intensivement en compression de données. Le codage de Huffman produit un code pré-optimal. La construction d'un code de Huffman est assez simple. Bien qu'il existe des techniques plus efficaces pour construire un code de Huffman, la technique présentée se veut simple et est souvent utilisée dans la littérature.

La technique utilise les arbres binaires et deux listes de données. Dans les arbres binaires, les feuilles représentent les symboles d'un alphabet  $\Sigma$  et chacun des nœuds a une probabilité  $p$  qui lui est associée. Pour les feuilles,  $p$  est simplement la fréquence d'apparition du symbole représenté. Pour les nœuds internes,  $p$  est égal à la somme des probabilités des nœuds  $l_1$  :  $p = p_1 + p_2$ , où  $p_1$  est la probabilité du  $l_1$  gauche et  $p_2$ , la probabilité du  $l_1$  droit. Deux listes de données sont utilisées pour ne pas avoir à retrier les nœuds à chaque fois. La première liste,  $F_1$ , est utilisée pour les feuilles et la deuxième liste,  $F_2$ , pour les nœuds internes.

L'algorithme pour construire un code de Huffman pour un alphabet  $\Sigma$  est le suivant :

1. Créer une feuille pour chacun des symboles de l'alphabet
2. Ajouter chacune des feuilles, en ordre croissant de probabilité  $p_2$ , a  $F_1$
3. Tant que  $F_1 + F_2 > 1$ 
  - Retirer de  $F_1$  et/ou  $F_2$  les deux nœuds ayant les plus petits  $p$
  - Créer un nouveau nœud ayant comme ls les deux nœuds qui viennent d'être retirés ; la probabilité du nœud est égale à la somme des probabilités de ses deux ls :  $p = p_1 + p_2$ , où  $p_1$  est la probabilité du ls gauche et  $p_2$ , la probabilité du ls droit
  - Ajouter le nouveau nœud a la n de  $F_2$
4. Le nœud restant est le nœud racine de l'arbre (arbre de Huffman).

Une fois l'arbre construit, nous sommes en mesure d'obtenir le mot de code d chacun des symboles de l'alphabet. Pour ce faire, il s'agit simplement d'accumuler les bits dans l'ordre racine feuille (chacune des feuilles représente un symbole). On peut donc retrouver le mot de code associé un symbole en temps linéaire au nombre de bits qu'il contient.

- Le codage de Huffman est optimal parmi les codes préfixes qui assignent des mots de code de longueurs entières aux symboles.
- Il faut assurer un accord entre le codeur et le décodeur quant à la façon de gérer les nœuds.

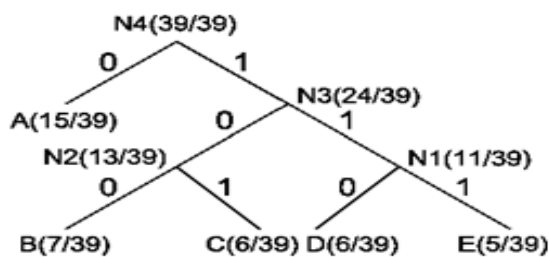


Figure I.4 : Arbre de Huffman pour l'alphabet  $\Sigma$

Symbole	Mot de code	$-\log(p)$
A	0	1.38
B	100	2.48

C	101	2.70
D	110	2.70
E	111	2.96

**Tableau I.2 :** Code de Huffman pour l'alphabet  $\Sigma$

La figure I.4. Illustre l'arbre de Huffman pour l'alphabet = A B C D E (tableau I.2). Les nœuds internes sont étiquetés N1, N2, N3 et N4. La probabilité associée à chacun des nœuds est entre parenthèses. Une branche porte l'étiquette 0 ou 1 servant à former les mots de code des symboles qui sont représentés par les feuilles de l'arbre. Le tableau 4.3 present le code de Huffman pour l'alphabet  $\Sigma$ .

Le codage de Huffman est, quant à lui, un code préfix optimal ; optimal si les mots de code qui sont produits sont composés d'un nombre entier de bits et qu'on n'encode qu'un seul symbole à la fois. En combinant les symboles en courtes séquences, il y a moyen d'atténuer l'effet néfaste de l'intégralité des bits dans les mots de code qui sont produits sont composés d'un nombre entier de bits et qu'on n'encode qu'un seul symbole à la fois. En combinant les symboles en courtes séquences, il y a moyen d'atténuer l'effet néfaste de l'intégralité des bits dans les mots de code. [12]

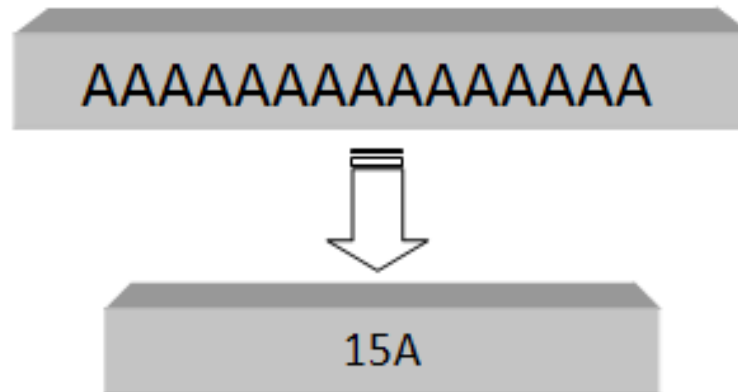
### I.5.5. Méthode de codage RLE

Cette méthode est destinée principalement aux sources d'information contenant fréquemment des successions de symboles similaires. Les images naturelles sont un bon exemple.

Dans une image les pixels voisins sont fortement corrélés entre eux. Souvent on observe que les pixels consécutifs dans une région douce d'une image sont identiques où la variation entre les pixels voisins est très petite. L'aspect des codes identiques est particulièrement vrai pour des images binaires où habituellement l'image qui se compose des 0 ou des 1. Même si les pixels consécutifs dans des images de niveaux de gris ou de couleur ne sont pas exactement identiques mais lentement changeants, ils peuvent souvent être prétraités et les valeurs traitées consécutives de pixels deviennent identiques.

Le codage RLE est une approche simple pour coder une chaîne constituée d'une succession d'un même code, par exemple la donnée  $d=55555555\ 19\ 19\ 19\ 19\ 19\ 19\ 19\ 19\ 19\ 19\ 19\ 19\ 19\ 0\ 0\ 0\ 0\ 0\ 0\ 8\ 23\ 23\ 23\ 23\ 23\ 23$  contient des chaînes de 5, 19, 0, 23 et 8. Les données peuvent être

représentées d'une manière compacte en indiquant simplement la valeur du symbole et la longueur de sa succession (voir Figure I.5).



**Figure I.5 :** codage RLE

De cette manière, la donnée  $d$  peut être codée par le codage RLE comme suit (5 7) (19 12) (0 8) (23 6). Pour la facilité de compréhension, nous avons représenté le code sous forme des paires. Ici la première valeur représente le code, alors que la deuxième indique le nombre de répétition de ce code.

Dans certains cas, l'aspect des chaînes de symboles peut ne pas être très évident. Mais les données peuvent probablement être prétraitées afin de faciliter le codage RLE.

Considérons la donnée  $d = 26 \ 29 \ 32 \ 35 \ 38 \ 41 \ 44 \ 50 \ 56 \ 62 \ 68 \ 78 \ 88 \ 98 \ 108 \ 118 \ 116 \ 114 \ 112 \ 110 \ 108 \ 106 \ 104 \ 102 \ 100 \ 98 \ 96$ . nous pouvons simplement prétraiter cette donnée, en prenant la différence d'échantillon  $e(i) = d(i) - d(i-1)$ , pour produire les données traitées  $t = 26 \ 3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 6 \ 6 \ 6 \ 6 \ 10 \ 10 \ 10 \ 10 \ 10 \ -2 \ -2 \ -2 \ -2 \ -2 \ -2 \ -2 \ -2 \ -2 \ -2$ . cette donnée prétraitée peut maintenant être facilement codée par RLE: (26 1) (3 6) (6 4) (10 5)

(-2 11). [11]

Une variante de cette technique est appliquée dans la norme JPEG. D'autre part, les images binaires (en noir et blanc), où la valeur de chaque pixels est 0 ou 1 peuvent être aussi subir un codage RLE similaire.

**Exemple :**

Pour :  $d=0000000001111111111100000000000000011100000000000010011111111111$ , le codage RLE est :  $c(d) = (0,9,11,15,3,13,1,2,10)$ . le premier 0 indique la valeur du premier bit, la valeur 9 indique le nombre de sa succession .

La valeur 11 indique le nombre de succession de 1 et ainsi de suite, nous aurons une suite alternée de nombre de succession de « 1 ». Tandis que la donnée binaire originale  $d$  exige 65 bits pour le stockage, sa représentation compacte  $c(d)$  exige 36 bits seulement dans la prétention que chaque code est représenté par 4 bits. Il est clair qu'il y a un problème d'allocation de bits pour coder les nombres de succession. [11] [13]

## I.6. Méthodes de compression avec perte

Dans la compression sans perte le taux de compression n'est pas toujours suffisant pour la transmission et le stockage de données de grandes tailles, et pour descendre au-dessous de l'entropie on agit sur la source d'information avec une méthode de compression avec perte mais ça entraîne le dilemme (taux de compression-distorsion). Pour améliorer, il va falloir accepter quelque dégradation . ceci permet d'atteindre des taux de compression arbitrairement grand au prix d'une dégradation toujours plus importante.

L'objectif des algorithmes de compression avec perte est minimiser cette dégradation de qualité pour un taux de compression donnée. Donc la clé de la compression avec perte est une modification non réversible de la source permettant d'obtenir une nouvelle source dont l'entropie est plus faible. il existe deux catégories principales de compression avec perte :

- ❖ Les méthodes directes
- ❖ Les méthodes par transformées

La figure (I.6) présente un schéma synoptique des méthodes de compression avec perte : [14]

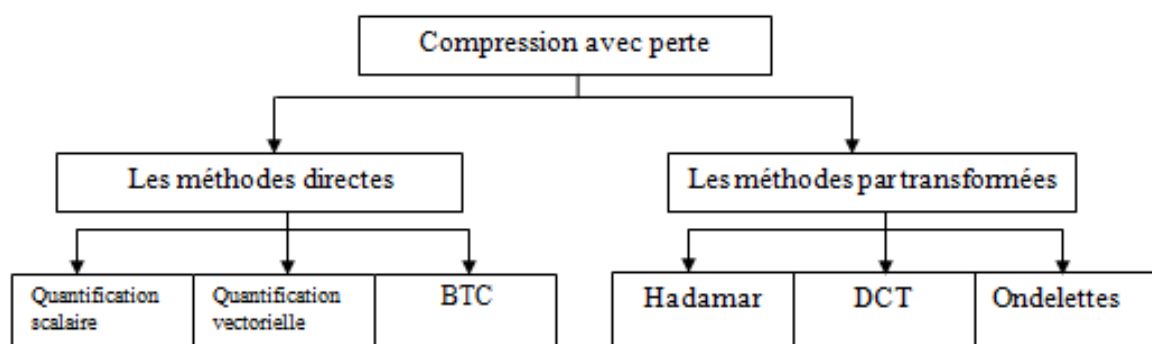


Figure I.6 : Méthodes de compression avec perte.

### I.6.1. Quantification [15]

La quantification désigne, dans sa forme la plus simple, le nombre de bits nécessaires pour stocker une image. C'est une application surjective ce qui veut dire que plusieurs points possèdent la même image (niveau quantifié) et c'est cet aspect qui la rend non réversible. La quantification peut être appliquée sur un seul élément (quantification scalaire), ou sur un group (vecteur) d'élément (c'est la quantification vectorielle).

Mathématiquement la quantification peut être formulée ainsi :

Soit « E » un espace vectoriel où « d » est la dimension de « E » et R une relation équivalence sur E.

Soit  $A = \{y_1, y_2, \dots, y_{N-1}\}$  un ensemble de tous les représentants  $y_i$  des classes d'équivalence  $S_i$  de E/R.

On distingue deux types de quantification : la quantification scalaire et la quantification vectorielle.

#### I.6.1.1. Quantification scalaire

C'est le cas où « d » définie précédemment est égale à « 1 » d'où Q(x) est appelée quantification scalaire où QS et  $y_i$  sont appelées niveaux de quantification et les ' $S_i$ ' sont

Appelées intervalles de quantification .la QS consiste à quantifier les niveaux de gris des pixels les uns après les autres .le principe de construction du dictionnaire est comme suit :

Le dictionnaire est construit et optimisé par l'algorithme de Llyod , on le définit par

- ❖ La forme et les dimensions de pixels.
- ❖ La taille du dictionnaire.

Le principe du codage est attribuer à chaque amplitude d'entrée la plus proche valeur du dictionnaire scalaire .l'importance de la quantification scalaire (QS), du point de vue de compression, tient à plusieurs raisons parmi lesquelles on peut citer :

- ❖ L'intérêt général de la quantification, dont une application particulière est la conversion analogique numérique.
- ❖ La simplicité qui permet des traitements en temps réel des applications difficiles.la QS est généralement à la base d'autres méthodes.

#### I.6.1.2. Quantification vectorielle

C'est le cas où « d » défini précédemment est ( $d > 1$ ) d'où Q(x) est appelé quantification vectorielle où 'QV' et  $y_i$  sont appelées représentants et les  $S_i$  sont appelées zones ou régions.

Quantifier un phénomène physique consiste à le remplacer par un autre de même nature. La QV consiste à remplacer un bloc complet de pixels par la représentation binaire du numéro du bloc le plus ressemblant établi appelé dictionnaire. Ce qui sera stockée ou transférée est la liste des étiquettes. Le bloc déclaré le plus ressemblant au bloc original est celui qui parmi tous les blocs du dictionnaire conduit à la plus petite erreur.

La quantification vectorielle se décompose en deux applications : codeur, décodeur.

## Codeur

Le rôle du codeur consiste pour tout vecteur  $X$  du signal en entrée à rechercher dans le dictionnaire  $Y$  le code vecteur  $y_j$  ainsi sélectionnée qui sera transmise ou stocké. C'est à ce niveau donc que s'effectue la compression.

## Décodeur

Il dispose d'une réplique du dictionnaire et consulte celui-ci pour fournir le code vecteur d'indice correspondant à l'adresse reçue. Le décodeur réalise l'opération de décompression.

## I.6.2. Méthode par transformée [16]

La transformation des données d'entrée est faite afin de mieux compacter l'énergie de la transformée d'image sur un nombre faible de coefficients.

La transformation a pour objet de décorrélérer les pixels d'une image. On opère la transformation sur un bloc unitaire ou directement sur l'image entière.

### I.6.2.1. Transformée de Fourier Discrète DFT

La DFT permet de passer du domaine spatial au domaine fréquentiel. La DFT d'un signal discret  $X_n$  s'exprime par :

$$X_k = \sum_{n=0}^{N-1} x_n \exp\left\{i \frac{2\pi nk}{N}\right\} \quad (\text{I. 3})$$

La fonction inverse permettant de remonter au signal original  $X_n$  connaissant sa transformée  $X_k$  est :

$$x_n = \sum_{k=0}^{N-1} X_k \exp\left\{+i \frac{2\pi nk}{N}\right\} \quad (\text{I. 4})$$

La DFT présente un coût de calcul très élevé. Ainsi une variante de transformée de Fourier dite rapide (FFT) a été développée pour assouplir ce coût.

### I.6.2.2. Transformée en Cosinus Discrète DCT [17]

Ahmed, Natarajan et Rao [17] ont été les premiers à mettre en équation la DCT en 1974. C'est une transformation mathématique qui transforme un ensemble de données d'un domaine spatial en un spectre de fréquence et inversement par l'IDCT (*Inverse Discrete Cosine Transform*).

#### \* DCT unidimensionnelle

La DCT unidimensionnelle (DCT-1D)  $F(k)$  d'une séquence  $f(i)$  de  $N$  échantillons est donnée par l'équation (I.5) :

$$F(k) = \sum_{i=0}^{N-1} f(i) \cos \frac{\pi}{2N} (2i+1)k \quad (I.5)$$

Avec  $k = 0, 1, 2, \dots, N-1$ , et  $a(k)$  est défini comme :

$$c(a) = \begin{cases} \frac{1}{\sqrt{n}}, & \text{pour } (k = 0) \\ \frac{2}{\sqrt{n}}, & \text{pour } (k \neq 0) \end{cases} \quad (I.6)$$

La transformée inverse de la DCT-1D a pour équation :

$$f(i) = \sum_{k=0}^{N-1} a(k) F(k) \cos \frac{\pi}{2N} (2i+1)k \quad (I.7)$$

Il est clair d'après (I.6) et (I.7) que pour  $k=0$

$$F(k=0) = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} f(i) \quad (I.8)$$

Ainsi, le premier coefficient de la transformée est la valeur moyenne de la séquence d'échantillons (pixels). Dans la littérature, cette valeur est désignée sous le nom du coefficient DC (pour *Direct Component*) ou coefficient continu. Tous les autres coefficients de la transformée sont nommés les coefficients AC (*Alternative Component*) qui représentent les amplitudes des fréquences spatiales.

Le dessin de  $\sum_{i=0}^{N-1} \cos \frac{\pi}{2N} (2i+1)k$ , pour  $N=8$  et en variant la valeur de  $i$  Selon l'observation précédente, la première forme d'onde au-dessus-gauche ( $k=0$ ) rend une valeur constante (DC),

tandis que, toutes les autres formes d'onde ( $k = 1.2\dots, 7$ ) ont des fréquences progressivement croissantes [18].

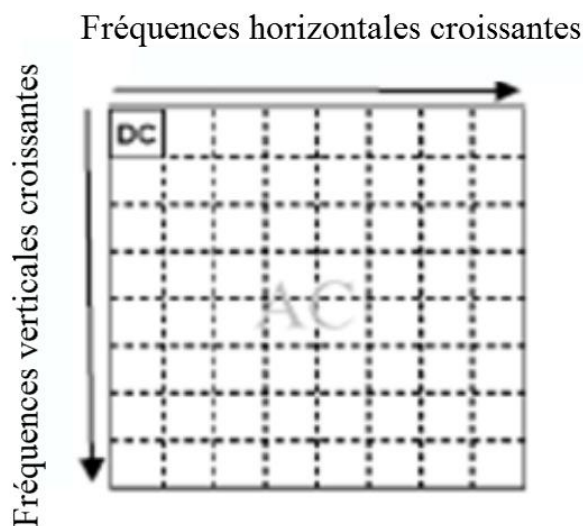
**\* DCT bidimensionnelle**

La DCT-1D est utilisée en traitement des signaux unidimensionnels tels que les signaux de la parole. Pour l'analyse d'un signal bidimensionnel (2D) comme les images, on a besoin d'une version 2D de la DCT.

La DCT-2D est effectuée sur une matrice carrée  $N \times N$  de pixels et donne une matrice carrée  $N \times N$  de coefficients fréquentiels. Comme pour la DCT-1D, l'élément (0,0) est appelé la composante DC et les autres éléments sont les composantes AC.

Par convention, les 64 valeurs transformées (de chaque bloc de  $8 \times 8$ ) sont positionnées d'une certaine manière, ainsi la valeur moyenne de tous ces coefficients est placée en haut à gauche de ce bloc. Plus on s'éloigne des coefficients continus plus leurs grandeurs tendent à diminuer, ce qui signifie que la DCT concentre l'énergie d'image en haut à gauche de la matrice transformée, les coefficients en bas et à droite de cette matrice contiennent moins d'information utile.

On peut représenter la distribution des fréquences de la DCT d'une matrice de  $8 \times 8$  éléments par la figure I.7.



**Figure I.7 :** Distribution des fréquences de la DCT

Les deux équations qui suivent donnent, respectivement, la DCT-2D directe et inverse [17]

$$F(k, m) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos\left(\frac{\pi(2i+1)k}{2N}\right) \cos\left(\frac{\pi(2j+1)m}{2N}\right) \tag{I.9}$$

Où  $k, m = 0, 1, 2, 3, \dots, N-1$  sont des indices des fréquences spatiales dans la direction horizontale et verticale de l'image, et  $a(k)$  et  $a(m)$  sont donnés par l'expression (I.6).

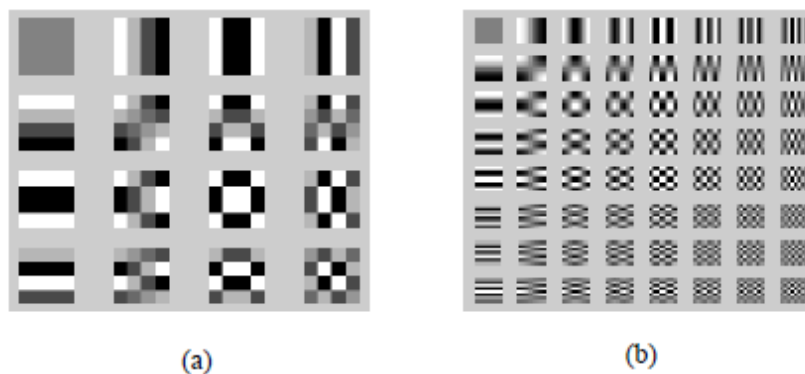
- $f(i, j)$  représente la valeur d'un pixel de l'image initiale pour  $i$  et  $j$  donnés.
- $F(k, m)$  représente les coefficients de la DCT.
- $N$  représente la taille du bloc.

La transformée en inverse est donnée comme :

$$f(i, j) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} a(k)a(m)F(k, m) \cos\left(\frac{\pi(2i+1)k}{2N}\right) \cos\left(\frac{\pi(2j+1)m}{2N}\right) \tag{I.10}$$

Où  $i, j = 0, 1, 2, 3, \dots, N-1$ .

Dans le cas d'un bloc de taille 4x4 et 8x8 les différentes fonctions de bases qui représentent l'espace transformé sont, respectivement, au nombre de 16 et 64. Ces fonctions sont représentées sur la figure (I.8). Les fonctions situées en haut à gauche représentent les basses fréquences de la transformée, les fréquences spatiales augmentent au fur et à mesure que l'on se déplace vers le coin inférieur droit du bloc.



(a) : Pour  $N=4$ , (b) : Pour  $N=8$

**Figure I.8** : Fonctions de base de la DCT-2D

### I.7. Conclusion

Ce chapitre présente l'état de l'art de la compression des images médicales et décrit les différents Techniques réversibles utilisées actuellement dans le domaine de la compression des images numériques. Une attention particulière a été réservée aux méthodes de compression basées sur l'algorithme SPIHT et cette dernière basées aussi sur la transformée en ondelettes.

La prochaine chapitre sera réservé au développement de la technique de la compression par la transformée en ondelettes.

## II.1. Introduction

La représentation efficace de l'information visuelle est au cœur de nombreux problèmes en traitement d'images incluant la compression, le filtrage, et l'extraction de primitives pertinentes. Par représentation efficace on entend la description compacte et précise, par un nombre restreint d'échantillons, de l'information significative contenue dans une image [19].

En effet, une transformée efficace peut capturer l'essence d'un signal donné par un nombre minimal de fonctions de base. En pratique, cette description doit être générée par des transformations structurelles et des algorithmes rapides [20]. La transformée en ondelettes [21] récemment intégrée dans le standard de codage d'images JPEG2000, constitue un exemple typique pour ces représentations. Leur application à l'imagerie médicale date de 1992. Depuis leur introduction il y a deux décennies, les ondelettes ont fait l'objet de beaucoup d'attention dont elles y ont démontré leur force dans des domaines aussi diversifiés que le débruitage, la compression, le codage, l'imagerie médicale ou satellitaire... Récemment, il est apparu évident que les transformations usuelles de type séparable, telles que la transformée en ondelettes, semblent former une catégorie restreinte et limitée des possibilités de représentations des images et de signaux multidimensionnels. Cette limitation est due en premier lieu au fait que de telles représentations ne prennent pas en compte la régularité des structures géométriques d'une image. Comprendre comment représenter les structures géométriques complexes est devenu donc un enjeu important pour améliorer l'état de l'art du traitement d'images.

Le besoin de disposer d'outils permettant l'identification de la structure géométrique d'une image se fait ainsi sentir. Par ailleurs, les communautés de chercheurs en traitement d'images et en vision par ordinateur ont plutôt tendance à privilégier les représentations d'images exhibant une sensibilité directionnelle au spectre fréquentiel. Cette caractéristique tant désirée trouve sa justification dans les aspects orientationnels du système de perception visuelle humain. Basés sur cette observation, des travaux récents montrent qu'il est possible de définir des cadres théoriques de représentations multi-échelles plus vastes donnant naissance à de nouvelles transformées directionnelles dites ondelettes de secondes générations plus adaptées à la structures géométriques de l'image .

L'objet de ce chapitre est de présenter les ondelettes que nous allons utiliser dans ce travail. Nous les situerons dans le panorama plus large de l'analyse temps-fréquence, puis nous rappellerons leurs propriétés fondamentales.

## II.2. Fréquence locale

Les ondelettes ont été introduites il y a presque 50 ans pour combler une lacune entre deux modes extrêmes de représentation du signal: la représentation d'une fonction par son graphe classique de  $\mathbb{R}$  dans  $\mathbb{R}$ , ce qui correspond à une décomposition sur la base continue des distributions de Dirac :

$$f(t) = \int_{\mathbb{R}} f(u) \delta(t - u) du \quad (\text{II.1})$$

et la représentation dans la base de Fourier :

$$f(t) = \int_{\mathbb{R}} f(\omega) e^{i\omega t} d\omega \quad (\text{II.2})$$

La première représentation donne une information précise en temps: la valeur  $f(t)$  indique l'intensité du signal au temps  $t$ . En revanche l'information fréquentielle est nulle. La valeur en un point de  $f$  ne donne aucune information sur le (contenu fréquentiel) du signal  $f$ . à l'inverse, la représentation de Fourier donne une information très précise en fréquence, mais ne donne aucune information temporelle. Pour prendre une analogie musicale, supposons que la courbe  $f$  représente un son.

La représentation temporelle  $f$  permet de lire la localisation temporelle du signal, c'est-à-dire ici on entend un son, la on n'entend rien, sans que l'on puisse savoir quelle est la hauteur du son ou bien les notes qui composent un accord. En revanche, la représentation fréquentielle permet de dire: cet air de musique contient un la, un si, un ré et pas de sol, mais cette représentation ne permet de dire quand ces notes (de durée finie) sont jouées.

Naturellement, chacune de ces représentations contient les informations de l'autre puisque la transformation de Fourier permet de passer de l'une à l'autre. Néanmoins à chaque fois un seul type d'information est explicitée.

On peut reformuler les observations ci-dessus en disant que les distributions  $\delta(t - \tau)$  sont très localisées en espace et très peu en fréquence, et que l'inverse les  $e^{j\omega t}$  ont des résolutions spatiale et fréquentielles respectivement nulle et infinie. La question qui vient naturellement est la suivante: existe-t-il une représentation dans laquelle on puisse lire une information mixte, comme « à tel instant, on entend un la et un do » ? Morlet et Gabor ont cherché concevoir des fonctions de base qui se situent à mi-chemin entre ces extrêmes, c'est-à-dire qui ont à la fois une bonne localisation fréquentielle et une bonne localisation spatiale.

Une limite théorique dans cette perspective est bien connue: c'est l'inégalité de Heisenberg. Soit une fonction de base  $f$ , qu'on suppose de norme  $L_2$  égale à 1:

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = 1 \quad (\text{II.3})$$

On définit le centre  $c(f)$  et la largeur  $D(f)$  d'une telle fonction par :

$$c(f) = \int_{-\infty}^{\infty} t |f(t)|^2 dt \quad (\text{II.4})$$

$$D(f) = \sqrt{\int_{-\infty}^{\infty} (t - c(f))^2 |f(t)|^2 dt} \quad (\text{II.5})$$

L'inégalité de Heisenberg est une inégalité fondamentale qui s'écrit pour toute fonction  $f$  de norme 1 :

$$D(f) D(f) \geq \frac{1}{2} \quad (\text{II.6})$$

Si on appelle largeur fréquentielle d'une fonction  $f$  la largeur de sa transformée de Fourier  $\hat{f}$  cette inégalité interdit donc d'avoir une fonction avec des largeurs temporelle On connaît également les fonctions qui réalisent le minimum de cette limite théorique ce sont les fonctions gaussiennes translatées et modulées :

$$A e^{-\frac{(t-t_0)^2}{2\Delta t^2}} e^{j\omega_0 t} \quad (\text{II.7})$$

Où  $A$  est un coefficient de normalisation tel que la fonction ait une norme  $L_2$  égale à 1, qui dépend de  $\Delta t$ . Pour de telles fonctions, et seulement pour ces fonctions - 1 à l'inégalité (H) devient une égalité. Ces fonctions ont été appelées ondelettes de Gabor. [22]

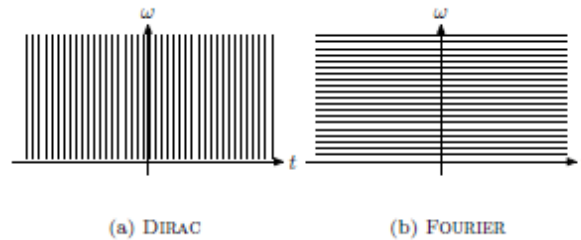
## II.3. Représentations temps -fréquence et temps- échelle

A une telle fonction, on associe un pavé temps-fréquence, c'est-à-dire un rectangle dans le plan  $(t, \omega)$  centre en  $(c(f), c(\hat{f}))$  et de dimensions  $D(f)$  et  $D(\hat{f})$ . Ce pavé est une représentation.

### II.3.1. Représentations temps-fréquence et temps-échelle

Intuitive de la couverture en temps et en fréquence d'une fonction. On associe également à une base un pavage du plan temps -fréquence, qui est un recouvrement du plan  $(t, \omega)$  par des rectangles de couverture des fonctions de base. Si le centre des boîtes de Heisenberg est fixé comme le point des centres spatiaux et fréquentiels de la fonction, leurs dimensions sont en général choisies de telle manière que les boîtes forment une partition du plan temps fréquence. Cette représentation a un aspect arbitraire, d'autant qu'aucun résultat ne lie le fait qu'une famille soit une base au fait que les pavés temps-fréquence de la famille recouvrent le plan.

Le pavage temps-fréquence correspondant par exemple aux bases de fonctions de Dirac et de Fourier sont des pavages par des rectangles infiniment fins et allongés qui sont représentées schématiquement dans la figure II.1 :



**Figure II.1 :** Pavages temps-fréquences associés aux représentations de Dirac et de Fourier.

Pour représenter un signal comme combinaison de telles fonctions, il est très redondant de laisser varier indépendamment les trois paramètres  $t_0$ ,  $w_0$  et  $\Delta t$ . Deux approches différentes ont prévalu :

- la première approche est l'approche temps-fréquence, dans laquelle la largeur spatiale  $\Delta t$  des fonctions  $g$  est indépendante de la fréquence. Les fonctions s'écrivent sous la forme :

$$g_{t_0, w_0}(t) = e^{jw_0 t} g_0(t - t_0) \quad (\text{II.8})$$

Où  $g_0(t) = A_0 e^{-t^2/2\Delta t^2}$ . Ce mode d'approximation est également appelé analyse de Fourier à fenêtre.

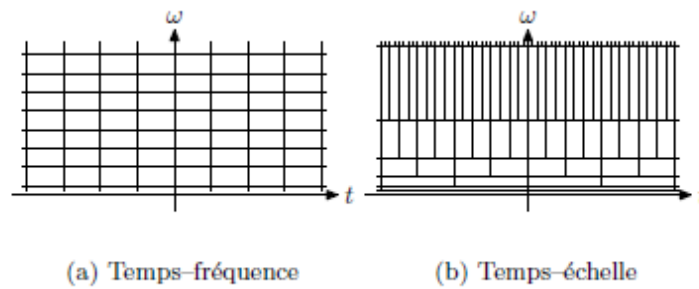
- Une deuxième approche est l'approche temps-échelle, dans laquelle la largeur spatiale des fonctions  $g$  est inversement à la fréquence (le produit  $w_0 \Delta t$  est constant et égal à  $c$ ). On obtient alors, à un facteur de module 1 près, une forme relativement simple pour les différentes ondelettes :

$$g_{t_0, D t}(t) = \frac{1}{\sqrt{D t}} g_0 \left( \frac{t - t_0}{D t} \right) \quad (\text{II.9})$$

Où  $g_0(t) = A_0 e^{-t^2/2\Delta t^2} e^{jct}$ .

Les pavages du plan temps-fréquence donnent une idée de la différence des deux approches.

Dans le cas de l'analyse temps-fréquence, le pavage obtenu est un pavage par des domaines rectangulaires qui se déduisent les uns des autres par translation dans le plan temps-fréquence. Dans le cas de l'analyse temps-échelle, les domaines sont également de surface constante, mais ont une résolution fréquentielle relative  $Dw/w_0$  constante.



**Figure II.2 :** Pavages du plan temps-fréquence pour la représentation temps-fréquence et la représentation temps-échelle.

Les ondelettes que nous allons utiliser dans la suite correspondent à l'analyse temps-échelle. Les avantages de ce mode de décomposition sont multiples. On dispose de moyens efficaces pour construire des bases discrètes pour lesquelles les calculs de transformation sont très rapides. L'analyse multi-échelles repose sur une forme de fonction unique. Enfin, dans l'analyse temps-échelle, les fonctions de base ont une taille de support proportionnelle à la résolution spatiale liée à leur fréquence par l'inégalité de Heisenberg. [23]

## II.4. Généralités sur les ondelettes

### II.4.1. Pourquoi les ondelettes ?

Que sont les ondelettes ? L'origine du mot ondelette n'est, en soi pas très compliqué.

Les ondelettes utilisées par Jean Morlet avaient, en effet, une allure de *petites ondes*. D'où cette appellation. Vers 1975, ce géophysicien, a créé des petites fonctions mathématiques ayant certaines propriétés intéressantes ; ceci afin de sonder les différentes couches géologiques.

Morlet est donc un des artisans majeurs des ondelettes. Cependant, d'autres chercheurs travaillant dans des domaines de recherches différents, utilisaient à cette époque des outils forts semblables aux *petites ondes* de Morlet. Seulement, ces méthodes étaient souvent expérimentales et ne possédaient aucun formalisme rigoureux. Ce formalisme émergera durant les années 80 des travaux de scientifiques tels que ceux d'Yves Meyer, de Stéphane Mallat, d'Ingrid Daubechies et de bien d'autres [24] [25].

Les ondelettes sont, avant tout, un puissant outil d'analyse mathématique, mais avant de parler d'analyse en ondelettes commençant par parler de l'analyse de Fourier. L'analyse de Fourier est un outil de base en traitement du signal, indispensable dans de nombreux domaines

de recherche, mais elle montre vite des limites justifiées dès lors que l'on sort du cadre rigoureux de sa définition : le domaine des signaux stationnaires d'énergie finie.

Dans l'analyse de Fourier tous les aspects temporels (début, fin, durée d'un événement), bien que présents dans la phase, deviennent illisibles dans le spectre. En particulier, la transformée de Fourier (TF) d'un morceau de musique qui ne permet pas de retrouver le rythme joué, mais simplement les notes présentes. Le spectre seul ne permet pas de dissocier deux partitions différentes ayant les mêmes notes. Or, on souhaiterait pourtant parfois réaliser à la fois une analyse en temps et en fréquence, pour retrouver la « portée musicale » associée à ces signaux non stationnaires.

L'étude des signaux non stationnaires nécessite donc le développement de méthodes Spécifiques. La première solution, mise en place intuitivement au milieu du siècle, correspond aux **analyses de Fourier à fenêtre glissante** (FFG) introduites dès 1945 par D. Gabor avec l'idée d'un plan temps-fréquence où le temps deviendrait un paramètre complémentaire de la fréquence. À ces approches, s'est ajoutée la **transformée en ondelettes**, existant à l'état latent aussi bien en mathématiques qu'en traitement du signal, mais dont le véritable essor a commencé au début des années 1980.

L'analyse en fréquence ne nous informe pas sur la localisation temporelle du changement du régime dans le signal.

## II.5. Définition d'une ondelette

Les ondelettes sont des fonctions  $\phi_{a,b}$  obtenues en dilatant d'un facteur  $a$  et en translatant d'un facteur  $b$  une fonction mère  $\phi$  monodimensionnelle, de nature oscillante, pour analyser des signaux par multi-résolution, les manipuler et les reconstruire :

$$\phi_{a,b}(t) = \frac{1}{\sqrt{a}} \phi\left(\frac{t-b}{a}\right) \text{ où } \int \phi(t) dt = 0 \quad (\text{II.2})$$

Le point de vue des ondelettes consiste à considérer un signal quelconque comme superposition des signaux élémentaires (les ondelettes) oscillants mais localisée dans le temps, à la différence des ondes éternelles de Fourier. Tous ces signaux élémentaires ont même forme et ne diffèrent que par leur instant d'apparition et leur durée.

Une ondelette ou valeur admissible est une fonction complexe de carré intégrable sur le plan, qui en plus satisfait la condition d'admissibilité :

$$\int \frac{|\hat{\phi}(k)|^2}{|k|^2} d^2k < \infty \quad (\text{II.3})$$

Où  $\hat{\phi}$  est la transformée de Fourier de  $\phi$  et  $|k| = (k_1^2 + k_2^2)^{1/2}$

Si  $\phi$  est la transformée **régulière**, l'admissibilité signifie simplement que l'ondelette est de **moyenne nulle** :

$$\phi(0) = 0 \text{ C.à.d. } \int \hat{\phi}(k) d^2K = 0 \quad (\text{II.4})$$

Une fonction  $f(x)$  de  $L^2(R)$  peut être analysée grâce à sa projection sur un ensemble de fonction de base  $\{\psi_{a,b}(t)\}$  de  $L^2(R)$ , appelée famille d'ondelettes définie comme :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right); a, b \in R, a \neq 0 \quad (\text{II.5})$$

Il s'agit d'un ensemble continu de translation et de dilatation de l'ondelette mère  $\psi(t) = \psi_{1,0}(t)$ . La transformation (4) définit donc une correspondance entre la variable spatiale  $t$  et les variables de dilatation et de translation  $a$  et  $b$ , accroissant du même coup la redondance dans la nouvelle représentation du signal :

$$c(a,b) = \langle f(t), \psi_{a,b}(t) \rangle \quad (\text{II.6})$$

Dans ce qui la notion  $\langle f, g \rangle$  ou  $\langle f |, g \rangle$  désigne le produit scalaire de  $f$  par  $g$ .

Les propriétés du signal que nous verrons ainsi apparaître seront donc fonction de l'ondelette mère  $\psi(t)$ .

D'une manière tout à fait générale, nous pouvons reconstruire La fonction originale en faisant appel à une seconde base d'ondelette de  $L^2(R)$  :

$$f(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(a,b) \tilde{\psi}_{a,b}(t) db \frac{da}{a^2} \quad (\text{II.7})$$

Pour assurer la convergence de cette dernière intégrale, nous devons imposer ce qu'on appelle la condition d'admissibilité :

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 = \int_{-\infty}^{\infty} \tilde{\psi}(t) dt \quad (\text{II.8})$$

La fonction  $\psi(t)$  sera donc oscillant et sa transformée de Fourier  $\psi(\omega)$  aura un comportement passe-bande. Ainsi, si  $\psi(t)$  est localisée dans l'espace autour de 0 et en fréquence autour

de  $\omega_0$  alors l'ondelette  $\psi_{a,b}(t)$  sera localisé dans l'espace autour de  $b$  et en fréquence de  $\frac{\omega_0}{a}$ .

Insistons sur le fait que le qualificatif continu porte sur la nature des variables  $a$  et  $b$ . En outre, la variable  $t$  appartient au même support  $\mathbb{R}$ .

Exemple d'ondelette mère : [26]

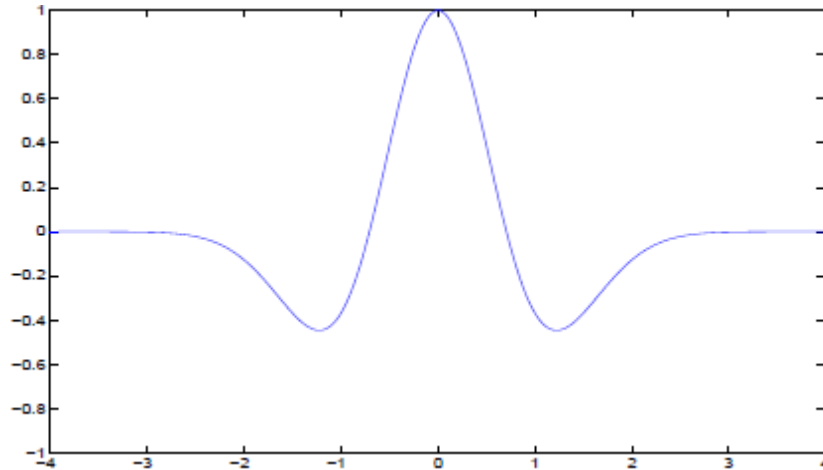


Figure II.3 : Ondelette simple « Dérivée seconde d'une Gaussienne »

### II.5.1. Transformée en ondelette continue(CWT)

On appelle transformée en ondelette du signal à temps continue  $x(t)$  l'expression :

$$s(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt \quad (\text{II.9})$$

Où  $\psi(t)$  est une fonction donnée, généralement à support compact. Cette expression est définie quel que soit  $a \in \mathbb{R}^+$  et  $b \in \mathbb{R}$ .

Le paramètre  $b$  caractérise l'emplacement sur l'axe des temps de la fonction sur laquelle on projette le signal à analyser  $x(t)$ .

L'idée sous-jacente à cette transformation est de considérer **une fenêtre que l'on dilate et translate**. son originalité est qu'elle permet d'analyser simultanément des événements dont le temps caractéristique est très différent.

Une autre manière d'introduire la transformation TOC ( $a, b$ ) consiste à définir les ondelettes comme fonctions de base. Le TOC( $a, b$ ) mesure la « similitude » entre le signal et les fonctions

des base  $\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$  appelées ondelettes . Celles-ci sont des versions dilatées ou contractées  $\left(\psi\left(\frac{t}{a}\right)\right)$  et translatées  $\psi(t-b)$  de l'ondelette prototype de base  $\psi(t)$  .

L'analyse ondelettes conduit à un ensemble de coefficients ondelettes indiquant de combien un signal est proche à une fonction de base particulière. Comme pour la transformée de Fourier, l'idée de départ est de décomposer sur une base orthogonale  $\{ C_{a,b} \}$  un signal  $s$  en une superposition (infinie en théorie, mais limitée numériquement) d'ondelettes :

$$s(x) = \sum_{i,j} O_{i,j} \cdot \phi_{i,j}(x) \quad \text{Où} \quad \int \phi_{i,j}(x) \cdot s(x) dx \quad (\text{II.10})$$

La décomposition est faite au moyen d'une ondelette d'analyse, alors que la reconstruction se fait au moyen d'une ondelette duale de synthèse .La forme d'onde originale est synthétisée par addition de blocs élémentaires de forme constante mais de taille et d'amplitude de différentes.

Les ondelettes  $\psi_{a,b}(t)$  se comportent comme une base orthogonale sur laquelle on projette le signal. La synthèse consiste à sommer les projections orthogonales du signal sur les ondelettes.

$$x(t) = c \iint_{a \gg} TOC(a,b) \psi_{a,b}(t) \frac{dadb}{2} \quad (\text{II.11})$$

Dans un but de compression, il est nécessaire de représenter de façon discrète les paramètres  $a$  et  $b$  .On choisit, par exemple,  $a = 2^j$  et  $b = k2^j$  On obtient alors des **coefficients d'ondelettes** :

$$c_j(k) = 2^{\frac{-j}{2}} \int_{-\infty}^{+\infty} x(t) \psi(2^{-j}t - k) dt \quad (\text{II.12})$$

$$x(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} c_j(k) \phi(2^{-j}t - k) \quad (\text{II.13})$$

C'est le problème général de la décomposition d'un signal suivant des fonctions de base.

On parle d'**ondelettes orthogonales** si  $\psi(t) = \phi^*(t)$  mais il ne s'agit que d'un cas particulier. A la différence des méthodes temps-«fréquence», la transformation en ondelette ne fournit pas directement une «fréquence instantanée» mais elle renseigne sur l'importance des « fluctuations instantanées ».est l'unique méthode d'analyse permettant d'accéder aux propriétés instantanées de **régularité** (exposants exemple) d'une fonction, ce qui explique qu'elle soit l'outil privilégié pour analyser les structures fractales ou la turbulence.

Ceci signifie que  $S$  est déterminée par sa restriction sur toute droite  $a=\text{constante}$ . Si  $\alpha$  est réel ceci signifie que la phase de  $S$  est constante sur toute droite qui pointe sur  $(0,0)$  c'est-à-dire telle que  $t/a=\text{constante}$  ; et sur de telles droites le module se comporte comme  $a^{-\alpha(1/2)}$ .

➤ Elle conserve l'énergie : ceci est crucial car elle assure qu'il n'y a pas de perte d'information en allant de  $s$  à  $S$  et que le signal peut-être reconstruit à partir de sa transformée. Nous avons :

$$\int |s(t)|^2 dt = \int |S(a,t)|^2 \frac{da}{a^2} dt \quad (\text{II.14})$$

Ceci entraîne que  $s(t)$  peut-être reconstruit à partir de  $s(a, t)$  :

$$s(t) = \int S(a,t) f_{a,t}(t) \frac{da}{a^2} dt \quad (\text{II.15})$$

On peut aussi reconstruire  $s$  à partir de la formule unidimensionnelle :

$$s(t) = C^{te} \int S(a,t) \frac{da}{a^{3/2}} \quad (\text{II.16})$$

$\Phi$  vérifiant certaines conditions.

Cette intégrale converge très rapidement si bien que l'on peut reconstruire le signal  $s$  avec une bonne précision en sommant seulement quelques coefficients (le nombre de coefficients à sommer dépend du signal considéré).

## II.5.2. Transformée en ondelette discrète (DWT) [27]

La transformation en ondelettes DWT (Gonzalez and Wood 2008) est une méthode consistant à transformer l'image du domaine spatial au domaine spatiaux-fréquentiel. La DWT se base sur des opérations de filtrage passe-haut et passe-bas et d'échantillonnage appliquées selon un algorithme arborescent et récursif. Le filtrage consiste à décomposer l'image en sous-bandes et l'échantillonnage vise à réduire la résolution de chaque sous-bande au minimum requis. Pour une décomposition à un seul niveau de résolution, la DWT représente l'image sous forme de quatre sous-bandes de résolution inférieure, une représente l'image d'approximation et les trois autres montrent les détails de l'image à orientations horizontales, verticales, et diagonales tel que montré dans les Figures (II.3) et (II.4). La transformée en ondelettes inverse DWT-1 permet une reconstruction parfaite de l'image à partir de ses sous-bandes ondelettes.

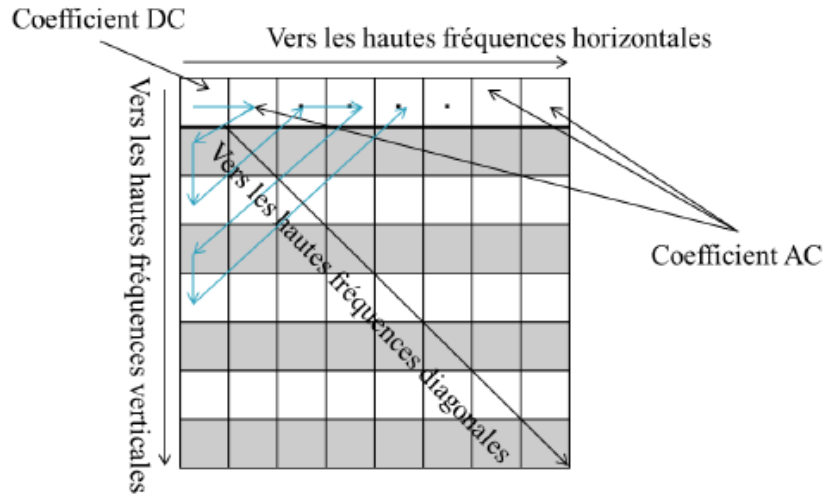
$$B(i, j) = a_i a_j \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m, n) \cos\left(\frac{p(2m+1)i}{2M}\right) \cos\left(\frac{p(2n+1)j}{2N}\right) \quad (\text{II.17})$$

Avec  $M$  et  $N$  désignant le nombre de lignes et de colonnes de l'image  $I$

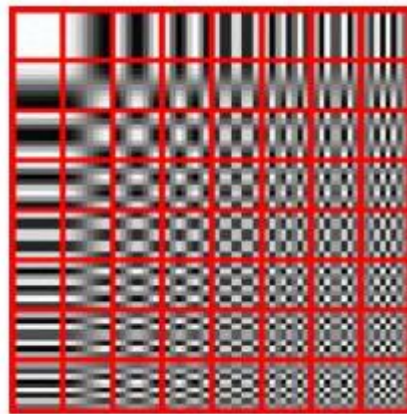
$$0 \leq i \leq M-1$$

$$0 \leq j \leq N-1$$

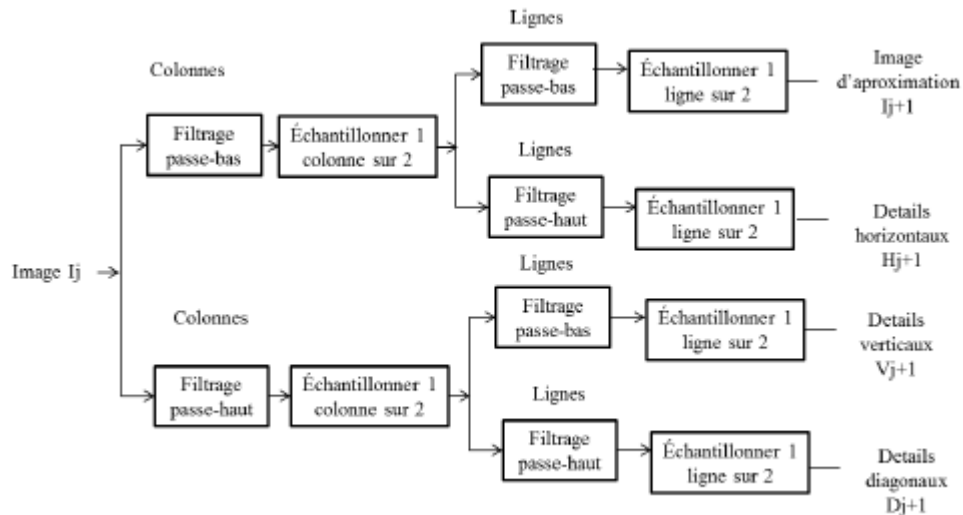
$$a_i = \begin{cases} \frac{1}{\sqrt{M}}, & i = 0 \\ \frac{\sqrt{2}}{\sqrt{M}}, & 1 \leq i \leq M - 1 \end{cases}, a_j = \begin{cases} \frac{1}{\sqrt{N}}, & j = 0 \\ \frac{\sqrt{2}}{\sqrt{N}}, & 1 \leq j \leq N - 1 \end{cases} \quad (\text{II.18})$$



**Figure II.4 :** DCT 8x8 d’une image montrant : le coefficient DC, les 63 coefficients AC, les fréquences horizontales, verticales, et diagonales, ainsi que le balayage par ordre zig-zag (couleur bleu ciel).



**Figure II. 5 :** Représentation des 64 composantes fréquentielles formant la base de décomposition en DCT 8x8.



**Figure II. 6 :** Processus de décomposition de l'image selon la transformation en ondelette :  
( $j+1$ ) désigne le niveau de décomposition.

Dans le cas bidimensionnel (2D), les coefficients d'ondelettes sont obtenus par application de la transformée en ondelettes (1D) sur les deux dimensions du signal à chaque niveau de décomposition (lignes puis colonnes pour les images numériques).

Alors, après  $k$  niveaux de décomposition on obtient  $3k$  parties (sous bandes) de coefficients hauts fréquences ou détails (à chaque niveau trois détails sont obtenus ; horizontales, verticales et diagonales) et une partie de basses fréquences appelée approximation ou bande de base.

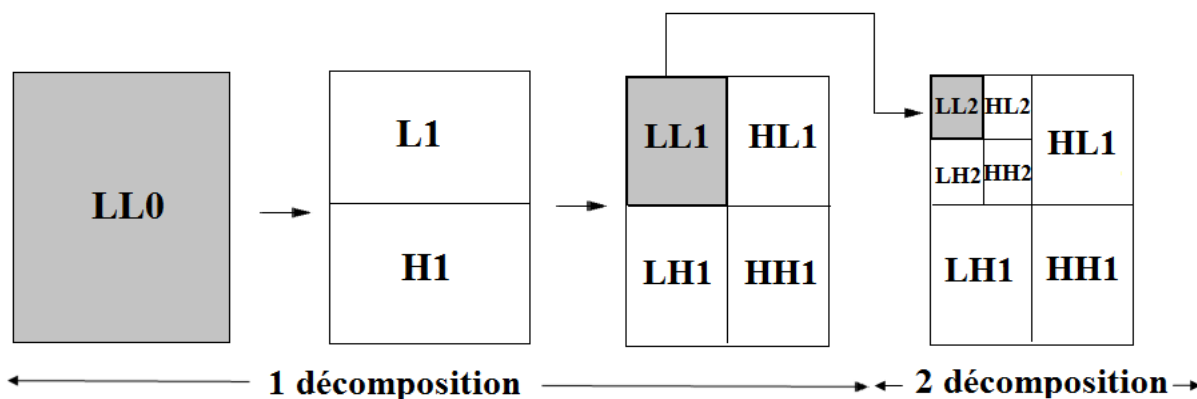
C'est ce que l'on appelle habituellement l'analyse multi résolution. Il s'agit de l'application de l'algorithme de Mallat.

Le principe est de décomposer un signal en une approximation, appelée également la bande de base ou basse fréquence (L en anglais de "low"), et en ses détails ou composantes hautes fréquences. La même opération est répétée, toujours sur l'approximation uniquement, jusqu'à l'obtention de la résolution souhaitée. S'il s'agit d'une image nous aurons un signal 2D. Dans ce cas chaque niveau de décomposition sera obtenu en deux étapes.

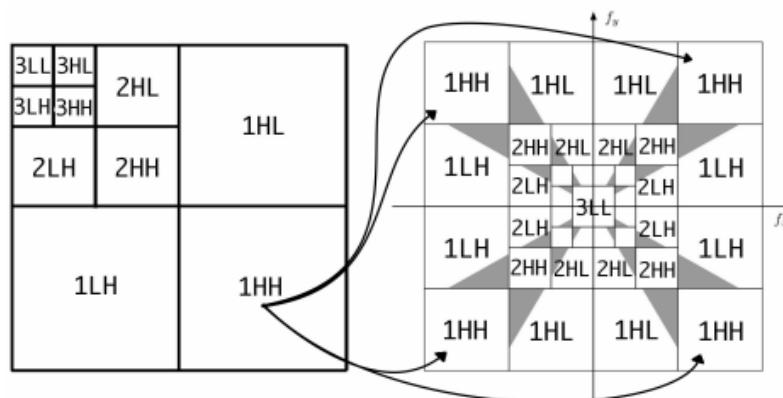
La première étape concerne les lignes et la seconde étape les colonnes. Un niveau de décomposition sur une image nous donnera une approximation (Bande de base ou LL1 pour low-Low niveau1) et trois détails respectivement LH1, HL1 et HH1 (pour Low-High niveau1, High Low niveau1 et High- High niveau1) :

- **LH1** : une décomposition basse fréquence selon lignes suivie par une décomposition haute fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition.

- **HL1** : une décomposition haute fréquence selon lignes suivie par une décomposition basse fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition.
- **HH1** : une décomposition haute fréquence selon lignes suivie par une décomposition haute fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition . [28] [29].



**Figure II.7** : Représentation des coefficients d'approximation et de détails



**Figure II.7** : À gauche : décomposition ondelette classique en sous-bandes représentées dans un espace spatio-fréquentiel. À droite : empiètement horizontal et vertical sur les sous-bandes diagonales dans un espace fréquentiel, avec correspondance des sous-bandes de la figure de gauche.

### II.6. Extension de transformée en ondelettes pour les images [30]

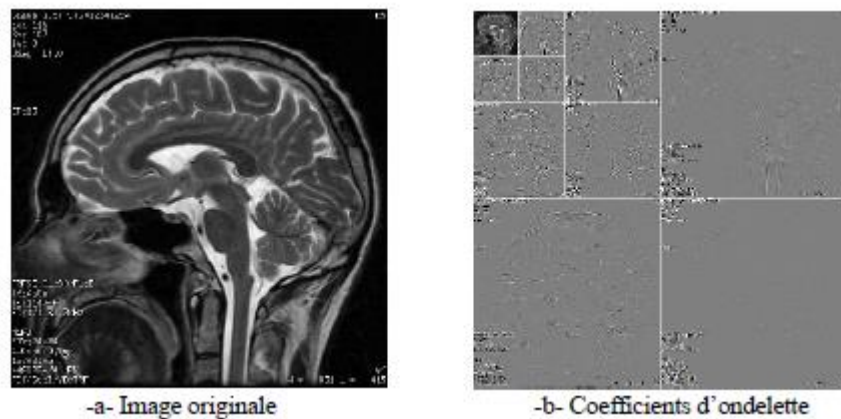
Il existe plusieurs façons pour étendre la transformation en ondelettes pour un signal 2D, tel que l'image. Le procédé classique consiste à obtenir la fonction d'échelle 2D $\phi(x, y)$  et la fonction d'ondelette  $\psi(x, y)$  par le produit tensoriel des fonctions d'ondelettes 1D $\psi(x)$ ,  $\psi(y)$  et les fonctions d'échelle  $\phi(x)$ ,  $\phi(y)$ .

La fonction d'ondelette mère 2D correspondante permet d'obtenir les détails dans trois directions, horizontale, verticale et diagonale :

$$y^v(x, y) = j(x)y(y) \quad (\text{II.17})$$

$$y^H(x, y) = y(x)j(y) \quad (\text{II.18})$$

$$y^D(x, y) = y(x)y(y) \quad (\text{II.18})$$

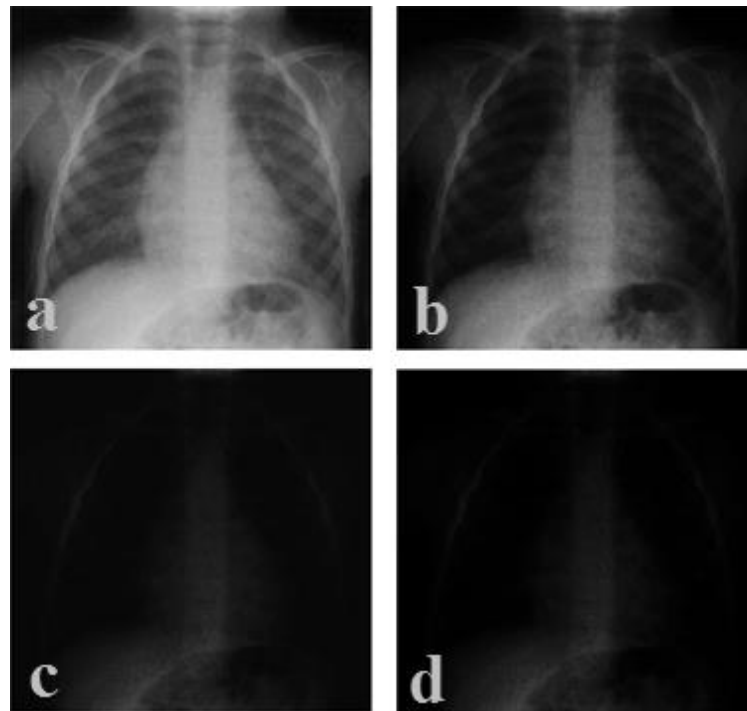


**Figure II.8 :** Décomposition de l'image IRM "coupe sagittale" via une AMR, par la transformée en ondelettes de CDF9/7. de cerveau sur 3 niveaux de résolution.

## II.7. Coefficient d'ondelettes 2D

L'ensemble des coefficients d'échelle et de détails en 2D est de même taille que le nombre de pixels de l'image originale et peut se représenter sous la forme d'une image organisée de façon hiérarchique (voir Figure II.8) dont les pixels les plus blancs représentent les coefficients de grandes amplitudes, alors que les pixels noirs représentent les coefficients d'ondelettes les moins significatifs.

La Figure II.9 donne un exemple de décomposition en ondelettes d'une image à niveaux de gris de taille  $512 \times 512$  pixels. On peut constater qu'aux petites échelles, les coefficients d'ondelettes de grande amplitude se concentrent au voisinage des points où l'intensité des pixels des images change rapidement. Ils correspondent soit aux contours horizontaux, verticaux ou diagonaux de l'image.



(a) Image originale, (b) coefficients détails verticaux,  
(c) coefficients détails horizontaux, (d) coefficients détails diagonaux.

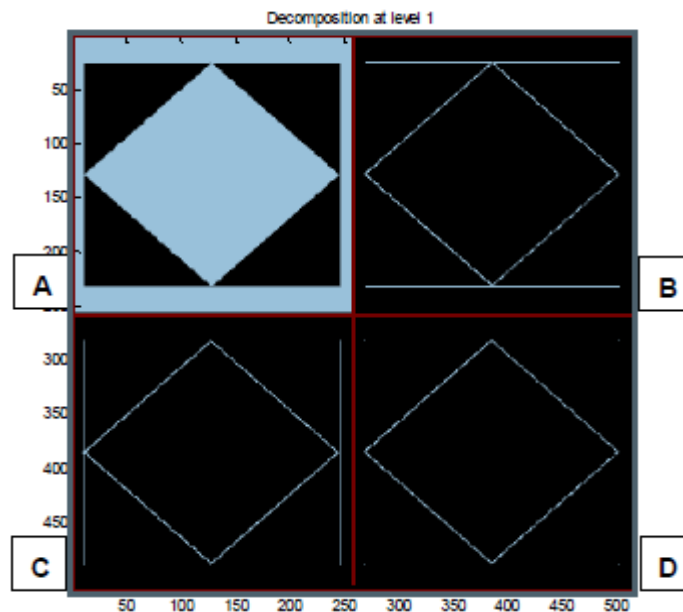
**Figure II.9 :** Détection des contours et singularités ponctuelles par les ondelettes 2D :

## II.8. Compression d'images fixes par ondelettes classiques

La popularité des ondelettes dans divers domaines scientifiques et techniques est due à son efficacité dans la représentation des singularités ponctuelles dans les signaux unidimensionnels, à cause du support compact des fonctions de bases de décomposition, ce qui implique que si on a un signal 1-D dont l'énergie est concentrée en un point (discontinuité ponctuelle), nous n'aurons que quelques coefficients d'ondelettes significatifs à chaque échelle.

Pour les signaux 2D tels que les images, le problème devient beaucoup plus complexe, à cause de la présence d'autre type de régularités le long des bords et contours des objets.

Les ondelettes séparables 2-D obtenues par le produit tensoriel de deux ondelettes 1D horizontal et vertical, ne peuvent pas détecter les régularités géométriques, qui ne sont pas le long des directions horizontales, verticales ou diagonales. Ainsi, les coefficients obtenus présentent certaines corrélations inter ou intra-échelles. La Figure II.9 illustre la décomposition d'une forme élémentaire 2-D, suivant une base d'ondelettes séparables.



(A) Image originale, (B) coefficients détails horizontaux,  
(C) coefficients détails verticaux, (D) coefficients détails diagonaux.

**Figure II.10 :** Illustration de la directivité du transformé en ondelettes séparables 2D  
(décomposition d'une forme élémentaire)

L'efficacité de la représentation est un facteur important dans la transformée en ondelettes 2D. Elle peut se référer à la capacité de capturer les informations importantes à propos d'un objet avec le moins possible de coefficients non nuls. Ce facteur est de grande utilité dans la compression d'images. De ce fait, un grand nombre de méthodes de compression d'images parmi les plus récentes et les plus efficaces, prennent appui sur les ondelettes. La plupart de ces méthodes sont basées sur l'exploitation des redondances entre les coefficients obtenus par une transformation en ondelettes.

On peut distinguer les méthodes qui essaient d'exploiter les redondances entre les différentes échelles de décomposition en ondelettes telle que les méthodes de type arbre de zéros (Zero-Tree), et les méthodes qui exploitent les redondances intra-échelle, comme le codage par paquets d'ondelettes ou l'EBCOT utilisé dans le standard JPEG 2000.

## II.9. Conclusion

La compression d'image par ondelettes est maintenant bien développée pour les algorithmes de compression d'images. Ces algorithmes ont prouvé leur capacité à détecter les singularités horizontales, verticales, et diagonales. Cependant, les ondelettes ne sont pas en mesure de bien représenter les régularités géométriques le long des bords des objets. Ceci étant dû au fait que les

bases de transformée en ondelettes bidimensionnelle sont obtenues par le produit tensoriel de deux ondelettes unidimensionnelles appliquées suivant les lignes et les colonnes de l'image. Pour remédier à ce problème, il est recommandé d'utiliser des algorithmes capables d'exploiter efficacement les régularités géométriques. Le chapitre suivant présente un type parmi ces algorithmes, il a été nommé SPIHT.

### III.1. Introduction

L'algorithme SPIHT [30] reprend les principes évoqués dans EZW tout en proposant de partitionner récursivement les arbres de coefficients. Ainsi, là où EZW codait un coefficient non significatif isolé ('ZI'), SPIHT effectue un partitionnement récursif de l'arbre de manière à déterminer la position des coefficients significatifs dans la descendance du coefficient considéré. Les coefficients significatifs sont codés de manière similaire à EZW : leur signe est envoyé dès qu'ils sont identifiés comme étant signifiants et ils sont ajoutés à la liste des coefficients à raffiner. Cet algorithme fonctionne également par plans de bits. Il offre des performances remarquables, atteignant celles d'EZW sans codage entropique. En ajoutant un codage entropique de l'information de signifiante, un gain supplémentaire entre 0.3 et 0.6 dB peut être obtenu [33].

Les bits envoyés lors de la passe de signifiante correspondent au programme exécuté à l'encodeur lors de l'exécution de l'algorithme de classement en coefficients significatifs et non significatifs. En suivant le même programme, le décodeur reste synchrone avec les décisions de l'encodeur et retrouve la même classification. Cet algorithme repose sur la gestion de trois listes, de coefficients signifiants (LSP), de coefficients insignifiants (LIP) et d'ensembles insignifiants (LIS). Moyennant un seuil de signifiante divisé par deux à chaque itération, et dont la valeur initiale est transmise au décodeur.

L'algorithme se déroule de la manière suivante : la liste des coefficients signifiants est initialement vide, tandis que la liste de coefficients insignifiants contient les racines de chaque arbre (coefficients de la bande basse) et la liste d'ensembles insignifiants contient l'ensemble des descendants de chaque arbre. Cette partition initiale est segmentée récursivement au moyen de deux règles. Si un ensemble de descendants d'un nœud est signifiant, il est séparé en quatre coefficients fils directs de ce nœud, et l'ensemble des autres descendants. Les fils directs sont ajoutés à la LIP ou la LSP en fonction de leur signifiante. Si au moins un élément de l'ensemble des autres descendants est signifiant, cet ensemble est séparé en quatre ensembles insignifiants ajoutés à la LIS. Le fait de traiter les coefficients par groupes de quatre permet d'effectuer un codage entropique efficace par la suite. Comme dans EZW, la passe de raffinement consiste à coder progressivement les bits de poids faibles des coefficients significatifs [33] [34].

### III.2. Algorithme de codage SPIHT [36]

L'algorithme SPIHT a été proposé par Saïd et Pearlman en 1996 pour la compression d'image 2D avec et sans perte. Cet algorithme repose sur la même idée que celle de Shapiro (EZW) pour caractériser les dépendances entre les coefficients d'ondelettes. Cependant, il utilise les trois

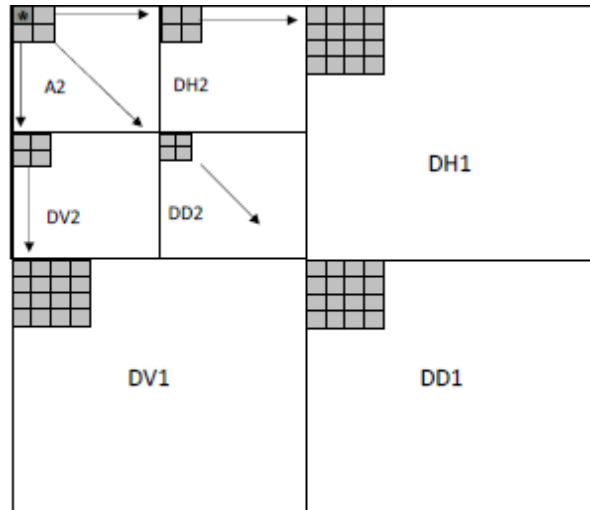
principes de base suivant : un rangement partiel par amplitude des coefficients d'ondelettes de la TO 2D (résultant de la quantification par approximations successives), un partitionnement dans des arbres hiérarchiques (à chaque seuil appliqué les arbres sont triés sur la base de leur signification en deux catégories d'arbre) et un ordonnancement de la transmission des bits de raffinement (l'amplitude de chaque coefficient significatif est progressivement raffinée).

La différence essentielle entre EZW et SPIHT est la façon dont les coefficients des arbres sont construits, triés et découpés. Ainsi la structure même des arbres de zéros est différente. Dans l'EZW, un arbre de zéros est défini par un coefficient racine et ses descendants ont tous la valeur zéro à l'intérieur d'un plan de bits. SPIHT utilise lui deux types d'arbres de zéros. Le premier (type A) consiste en une simple racine ayant tous ses descendants à 0 pour un plan de bits donné. Cela diffère un peu des arbres de zéros de l'EZW du fait que la racine elle-même n'a pas besoin d'être non significative.

En fait, bien que l'arbre de zéros soit spécifié par les coordonnées de la racine, la racine n'est pas incluse dans l'arbre. Le second type d'arbre (type B) est similaire mais exclut les quatre enfants de la racine. Les arbres de type B contiennent uniquement les petits-enfants, arrières petits-enfants, ..... de la racine.

Une autre différence est l'ordre de traitement des coefficients qui est dépendante des données. Alors que les coefficients sont traités en zigzag dans chaque sous-bande pour l'EZW. Le système de liste du SPIHT laisse l'ordre entièrement dépendant des données. Les coefficients sont traités selon leur position dans les listes. La définition des arbres de zéros comme on vient de le voir est sensiblement différente car SPIHT considère deux types d'arbres de zéros : le type A (arbre de degré 1) et le type B (arbre de degré 2). Il est à noter que dans les deux cas, rien n'est dit sur la valeur du coefficient à la racine qui peut être significatif. [36]

Les coefficients de la sous-bande de plus basse fréquence (A2 en figure III.1) sont regroupés par quatre, et pour chaque groupe, la descendance est comme suit : Un des quatre coefficients (marqué par '\*' en figure III.1) n'admet pas des descendants, alors que les trois autres (en gris en figure III.1) ont quatre descendants chacun. Pour les autres sous-bandes, comme dans le cas de l'EZW, chaque coefficient admet quatre descendants. [29]



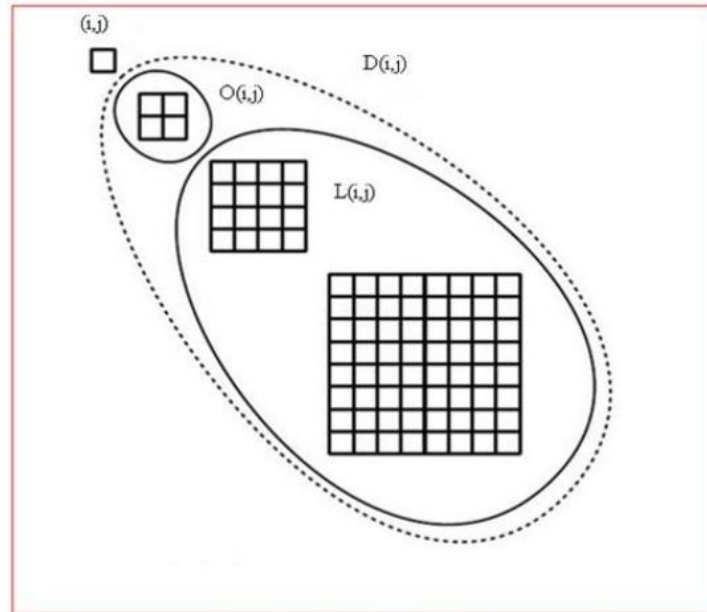
**Figure III.1:** Exemples de descendances parent-fils dans le cas SPIHT, le pixel désigné par (\*) n'accepte pas de descendants. [29]

Les principes de base de SPIHT sont les suivants [32]:

- un rangement partiel par amplitude des coefficients d'ondelettes résultant de la quantification par approximations successives.
- un partitionnement dans des arbres hiérarchiques à chaque seuil appliqué. Les arbres sont triés sur la base de leur signification en deux catégories d'arbres et un ordonnancement de la transmission des bits de raffinement (l'amplitude de chaque coefficient significatif est progressivement raffinée).

Les descriptions complètes des deux algorithmes de codage et décodage SPIHT sont données ci-dessous. Commençons par définir les différents ensembles utilisés par cet algorithme :

- ❖  $O(i, j)$  : Ensemble des coordonnées de tous les **enfants** du nœud  $(i, j)$ .
- ❖  $D(i, j)$  : Ensemble des coordonnées de tous les **descendants** du nœud  $(i, j)$ .
- ❖  $L(i, j) = D(i, j) - O(i, j)$  : L'ensemble des **descendants à l'exception des enfants** (type B d'arbre de zéros).



**Figure III.2 :** Terminologie SPIHT pour les descendants.

Les règles de partitions sont les suivantes [30] [31] [33] [34] :

- a)** La partition initiale est formée des ensembles  $\{(i, j)\}$  et  $D(i, j)$ , pour tous  $(i, j) \in LL_n$  qui ont un descendant. Dans chaque groupe de coefficients  $2 \times 2$  dans la  $LL_n$  dernière sous bande  $LL_n$ , un des coefficients n'a pas de descendances.
- b)** Si  $D(i, j)$  est significatif alors il est découpé en  $L(i, j)$  plus 4 ensembles  $D(l, m)$  d'un seul élément avec  $(l, m) \in O(i, j)$

### III.3. Méthode de codage

L'algorithme est cependant plus sophistiqué : contrairement à l'algorithme EZW qui n'utilise qu'un seul ensemble décrivant la signifiante des coefficients, Pour réaliser pratiquement un codage emboîté, l'algorithme SPIHT stocke l'information significative dans 3 listes ordonnées :

- LIS ou (LEN): Liste des ensembles non significatifs.
- LIP ou (LCN): Liste des coefficients non significatifs.
- LSP ou (LCS) Liste des coefficients significatifs.

Dans chaque liste, l'entrée, de coordonnées  $(i, j)$ , représente dans LSP et LIP un coefficient individuel et dans LIS elle représente soit l'ensemble  $D(i, j)$  soit l'ensemble  $L(i, j)$ .

Pendant la passe de signification, les coefficients dans LIP, qui étaient non significatifs dans la passe précédente, sont de nouveau testés. Ceux qui deviennent significatifs sont placés dans LSP. Similairement, les ensembles de LIS sont évalués selon leur ordre d'entrée. Si un ensemble

est trouvé significatif il est supprimé de cette liste puis partitionné. Les nouveaux ensembles, constitués de plus d'un élément, sont ajoutés à la fin de LIS avec le type (**A** ou **B**). Tandis que les simples coefficients sont ajoutés à la fin de LSP ou de LIP suivant leur signification.

La liste LSP contient les coordonnées des coefficients qui seront traités dans la prochaine passe de raffinement .

Pour préciser la relation entre la comparaison des amplitudes à un seuil net le message en bits envoyé, on utilise la fonction :

$$s_n(\Gamma) = \begin{cases} 1, & \text{si } \max_{(i,j) \in \Gamma} |c_{i,j}| \geq \lambda \\ 0, & \text{sinon} \end{cases} \quad (III.1)$$

Avec  $c_{i,j}$  qui représente le coefficient de coordonnées (i, j).

$s_n(\Gamma)$  : indique la signifiante de l'ensemble de coordonnées  $\Gamma$ .

Même principe que l'EZW, on utilise une passe pour la description des coefficients significatifs et une passe de raffinement. L'algorithme SPIHT se déroule comme suit [30] :

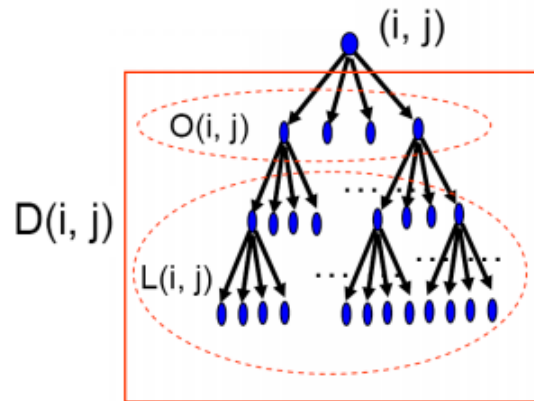
### III.3.1. Codage binaire [32]

Trois listes sont maintenues :

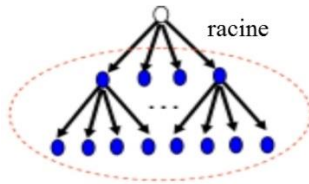
- LSP : (List of Significant Pixels) : Les coordonnées de tous les coefficients de la sous-bande de plus haut fréquence
- LIP = (List of Insignificant Pixels) : Les coordonnées de tous les coefficients de la sous-bande de plus basse fréquence.
- LIS = (List of Insignificant Sets) : Les coordonnées de sous-bandes de plus basse fréquence admettant des descendants.

### III.4. Arbres

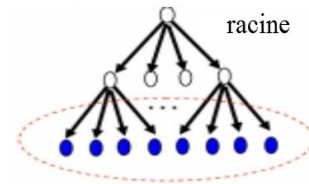
Se compose des descendants enfants et petits enfants :



Type **A** :  $D(i, j)$   
n'est pas significatif




Type **B** :  $L(i, j)$   
n'est pas significatif



### III.4.1. Structure d'arbres

Chaque nœud a 4 enfants, Sauf LL (impair, impair) = 0

 : Sans enfant

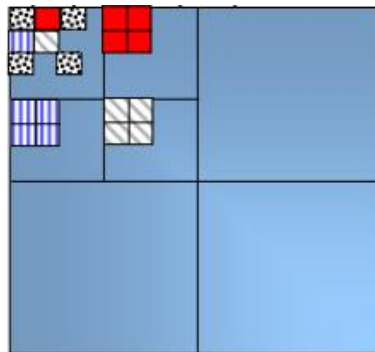


Figure III.3 : Modèle de dépendances inter-bandes pour SPIHT

### III.4.2. Arbres de Zéros

- ❖ Dans EZW : la racine et tous les descendants sont non significatifs.
- ❖ Dans SPIHT : la racine non incluse dans l'arbre.
- Lorsqu'un arbre est significatif :
  - vérifier si les 4 enfants directs sont significatifs et les passer en LIP ou en LSP

- vérifier si les petits-enfants (type B) sont significatifs, si oui, diviser en 4 sous arbres.

➤ LIP et LSP :

- passer un coef de la LIP à la LSP s'il devient significatif raffiner les coefs de la LSP à chaque passe.

### III.5. Initialisation [37]

Sortie  $n = \lceil \log_2(w_{\max}) \rceil$  ;  $Tn = 2^{\lceil \log_2(w_{\max}) \rceil}$  ; LSP =  $\emptyset$  ; LIP =  $\{(i, j) \text{ LL}\}$ .

LIS contient les mêmes coefficients que LIP excepté ceux qui n'ont pas de descendants.

#### 1. Passe de signification :

2.1. Pour chaque  $(i, j)$  LIP faire :

2.1.1. Sortie  $\sigma Tn(i, j)$

2.1.2. Si  $\sigma Tn(i, j) = 1$  alors mettre  $(i, j)$  dans LSP et coder le signe de  $w(i, j)$

2.2. Pour chaque  $(i, j)$  LIS faire :

2.2.1. Si l'entrée est de type **A**

a) Sortie  $\sigma Tn(D(i, j))$

b) si  $\sigma Tn(D(i, j)) = 1$  alors :

➤ Pour chaque  $(l, m)$  O  $(i, j)$  faire :

▪ Sortie  $\sigma Tn(l, m)$

▪ si  $\sigma Tn(l, m) = 1$  alors mettre  $(l, m)$  dans LSP et coder le signe de  $w(l, m)$

▪ Si  $\sigma Tn(l, m) = 0$  alors mettre  $(l, m)$  à la fin de LIP

➤ Si  $L(i, j) \neq \emptyset$  alors mettre  $(i, j)$  à la fin de LIS comme une entrée de type B et aller à

2.2.2 sinon supprimer  $(i, j)$  de LIS

2.2.2. Si l'entrée est de type **B**

a) Sortie  $\sigma Tn(L(i, j))$

b) si  $\sigma Tn(L(i, j)) = 1$  alors :

▪ Mettre chaque  $(l, m)$  O  $(i, j)$  à la fin de LIS comme une entrée de type **A**

▪ supprimer  $(i, j)$  de LIS.

#### 2. Passe de raffinement :

Pour chaque coefficient  $(i, j)$  LCS à l'exception de ceux ajoutés au cours de la dernière passe. Ecrire le nième bit le plus significatif de  $|w(i, j)|$ .

### **3. Modification du pas de quantification : $T_n \leftarrow T_n/2$ et aller à l'étape 2.**

Pour obtenir l'algorithme de décodage, il suffit simplement de remplacer le mot Sortie par Entrée dans l'algorithme précédent. De plus, le décodeur exécute une tâche supplémentaire en modifiant l'image reconstruite. Pour un seuil  $T_n$  donné, quand un coefficient est déplacé dans la LSP, il est évident que  $T_n < w(i, j) < 2T_n = 2^{n+1}$ . Ainsi, le décodeur utilise cette information plus le bit de signe juste après l'insertion dans la LCS pour mettre  $\hat{w}(i, j) = \pm 1.5 T_n$ . De manière identique, pendant la passe de raffinement le décodeur ajoute ou soustrait  $T_n/2$  à  $\hat{w}(i, j)$  quand on reçoit les bits de la représentation binaire de  $|w(i, j)|$ . De cette manière, la distorsion baisse à la fois pendant les 2 passes.

Enfin, on notera que contrairement à l'EZW, SPIHT produit directement des symboles binaires. Ainsi, un codeur arithmétique n'est pas obligatoire même s'il est souvent implanté pour améliorer les performances de codage. L'ensemble des améliorations proposées dans SPIHT par rapport à l'EZW, en fait la référence des méthodes de codage inter-bande (c.à.d. celles qui utilisent les redondances inter-échelles entre les sous-bandes pour coder les coefficients d'ondelettes).

## **III.6. Exemple**

Nous allons maintenant voir le déroulement du SPIHT sur la même matrice de décomposition en TO 2D donnée par SHAPIRO :

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

**Fig.III.4** : Exemple de SHAPIRO.

Seuils successifs correspondants aux plans de bits :

$T_5 = 32$  ,  $T_4 = 16$  ,  $T_3 = 8$  ,  $T_2 = 4$  ,  $T_1 = 2$  ,  $T_0 = 1$

Comme on l'a vu précédemment l'algorithme SPIHT maintient trois listes de coefficients : la liste des coefficients significatifs LSP, la liste des coefficients non significatifs LIP et la liste des ensembles non significatifs LIS. La fonction  $\sigma_{T_n}(i, j)$  est égale à 0 si tous les descendants de  $(i, j)$  sont en dessous du seuil  $T_n$  (arbre de zéros) et 1 dans le cas contraire. SPIHT considère deux types d'arbres de zéros : le type A ou tous les descendants ne sont pas significatifs et le type B ou tous les descendants, à l'exception d'au moins un des enfants, n'est pas significatif.

Le déroulement de l'algorithme SPIHT sur les données de la figure III.4 est décrit dans le tableau III.1. Quelques précisions sont données pour différentes étapes :

1. Initialisation des listes : la LSP est vide, la LIP contient les 4 coefficients de plus basse fréquence et la LIS contient la même chose que la LIP à part le coefficient  $(0,0)$  (valeur 63) qui n'a pas de descendants, le seuil de départ est  $T_n=32$ .
2.  $(0,0)$  qui a la valeur 63 est significatif, il est mis dans la LSP. Le signe de cette valeur est également émis, et noté  $+$ . De même pour  $(1,0)$  (valeur-34) qui est significatif et son signe est noté  $-$ . Pour  $(0,1)$  et  $(1,1)$  ils ne sont pas significatifs donc on les laisse dans LIP.
3. et 4. Un des descendants de  $(1,0)$  est significatif (49) on teste donc les 4 enfants de  $(1,0)$  et on classe chacun soit dans LSP ou dans LIP et on émit leur signes. Comme un des enfants de  $(1,0)$  est significatif, il change de type (B) et mis à la fin de la LIS. On fait de même pour  $(0,1)$ .
5. Tous les descendants de  $(1,1)$  sont en dessous du seuil, on ne fait donc rien de particulier, on passe à l'élément suivant dans la LIS.
6.  $(1,0)$  est de type B, on regarde donc  $L(1,0)$  qui n'est pas significatif, on ne fait rien et on passe à l'élément suivant dans la LIS.
7.  $(0,1)$  est de type B, on regarde donc  $L(0,1)$  qui est significatif (présence de la valeur 47), on partage l'ensemble et on ajoute  $(0,2)$ ,  $(1,2)$ ,  $(0,3)$ ,  $(1,3)$  à la fin de la LIS comme élément de type A et on enlève  $(0,1)$  de la LIS.
8. On teste donc les 4 ensembles type A obtenus. Pour  $(1,2)$  il a un des descendants qui est significatif on teste alors les 4 enfants de  $(1,2)$  et on classe chacun soit dans LSP ou dans LIP et on émit son signe. Il n'a pas de petits-enfants, alors on le retire de la LIS.
9. Au final on obtient les listes LIP, LSP et LIS et ces listes constituent les listes d'entrées pour le passage suivant ou le seuil sera de 16.

Coeff. testé	Sortie	Action	Liste	
			LIS = {(1,0)A,(0,1)A, (1,1)A} LIP= {(0,0),(1,0),(0,1),(1,1)} LSP = ∅	(1)
(0,0)	1+	(0,0) dans LSP	LIP = {(1,0),(0,1),(1,1)} LSP = {(0,0)}	(2)
(1,0)	1-	(1,0) dans LSP	LIP = {(0,1),(1,1)} LSP = {(0,0),(1,0)}	(2)
(0,1)	0	Rien		(2)
(1,1)	0	Rien		(2)
D(1,0)	1	Tester descendants	LSP = {(0,0),(1,0),(2,0)}	(3)
(2,0)	1+	(2,0) dans LSP	LIP = {(0,1),(1,1),(3,0)}	(4)
(3,0)	0	(3,0) dans LIP	LIP = {(0,1),(1,1),(3,0),(2,1)}	
(2,1)	0	(2,1) dans LIP	LIP = {(0,1),(1,1),(3,0),(2,1),(3,1)}	
(3,1)		(3,1) dans LIP Change de type	LIS = {(0,1)A, (1,1)A, (1,0)B}	
D(0,1)	1	Tester descendant	LIP = {(0,1),(1,1),(3,0),(2,1),(3,1),(0,2)}	
(0,2)	0	(0,2) dans LIP	LIP = {(0,1),(1,1),(3,0),(2,1),(3,1),(0,2),(1,2)}	
(1,2)	0	(1,2) dans LIP	LIP={ (0,1),(1,1),(3,0),(2,1),(3,1),(0,2),(1,2),(0,3)}	
(0,3)	0	(0,3) dans LIP	LIP={ (0,1),(1,1),(3,0),(2,1),(3,1),(0,2),(1,2),(0,3),(1,3)}	
(1,3)	0	(1,3) dans LIP Change de type	LIS = {(1,1)A, (1,0)B, (0,1)B}	
D(1,1)	0	Rien		(5)
L(1,0)	0	Rien		(6)
L(0,1)	1	Ajouter des ensembles	LIS = {(1,1)A, (1,0)B, (0,2)A, (1,2)A, (0,3)A, (1,3)A}	(7)
D(0,2)	0	Rien		(8)
D(1,2)	1	Tester descendant	LIP={ (0,1),(1,1),(3,0),(2,1),(3,1),(0,2),(1,2),(0,3),(1,3),(2,4)}	(8)
(2,4)	0	(2,4) dans LIP	LSP = {(0,0),(1,0),(2,0),(3,4)}	
(3,4)	1+	(3,4) dans LSP	LIP={ (0,1),(1,1),(3,0),(2,1),(3,1),(0,2),(1,2),(0,3),(1,3),(2,4),(2,5)}	
(2,5)	0	(2,5) dans LIP	LIP={ (0,1),(1,1),(3,0),(2,1),(3,1),(0,2),(1,2),(0,3),(1,3),(2,4),(2,5),(3,5)}	
(3,5)	0	(3,5) dans LIP	LIS = {(1,1)A, (1,0)B, (0,2)A, (0,3)A, (1,3)A}	
L(1,2)= ∅		enlever (1,2)		
D(0,3)	0	Rien		(8)
D(1,3)	0	Rien		
			LIS = {(1,1)A, (1,0)B, (0,2)A, (0,3)A, (1,3)A} LIP = {(0,1),(1,1),(3,0),(2,1),(3,1),(0,2),(1,2),(0,3),(1,3),(2,4),(2,5),(3,5)} LSP = {(0,0),(1,0),(2,0),(3,4)}	(9)

**Tableau III.1 :** Tableau montrant étape par étape le déroulement du premier passage de l’algorithme SPIHT 2D.

### III.7. Décodage SPIHT

Pour obtenir, l’algorithme de décodage, il suffit simplement de remplacer le mot Sortie par Entrée dans l’algorithme précédent. Donc, les deux algorithmes ont la même complexité algorithmique. [33]

#### Plan de bits suivant [31] :

LIS = {(2,2)A, (1,2)B, (3,1)A, (4,1)A, (4,2)A} LSP = {(1,1),(1,2),(1,3),(5,4)}

LIP = {(2,1),(2,2),(1,4),(2,3),(2,4),(3,1),(3,2),(4,1),(4,2),(5,3),(6,3),(6,4) }

Reçoit 1-1+0000000000 0 0 0 0 0 1 0 1 0

➤ la LIP contient 12 éléments : reçoit 1- 1+ 0000000000

LSP = {(1,1),(1,2),(1,3),(5,4),(2,1),(2,2)}

LIP = {(1,4),(2,3),(2,4),(3,1),(3,2),(4,1),(4,2),

(5,3),(6,3),(6,4) }

#### la LIS

Pour (2,2)A, reçoit 0 : D non significatif

Pour (1,2)B, reçoit 0 : L non significatif

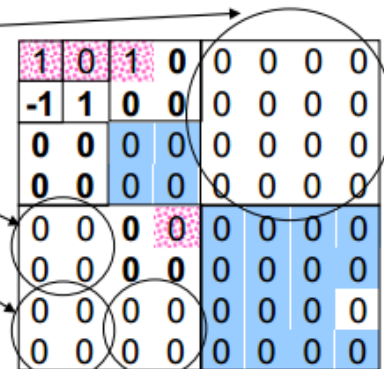
Pour (3,1)A, reçoit 0 : D non significatif

Pour (4,1)A, reçoit 0 : D non significatif

Pour (4,2)A, reçoit 0 : D non significatif

les bits suivants sont de raffinement

Voir LSP : 1010 bits de ces noeuds



Enfin, on notera que l’algorithme SPIHT produit directement des symboles binaires. Ainsi, un codeur arithmétique n’est pas nécessaire même s’il est souvent implanté pour améliorer les performances du codage. [30]

## IV.1. Introduction

Dans ce mémoire, nous nous sommes intéressés aux méthodes de compression avec pertes basées sur la transformée en ondelettes 2D. Cette dernière possède des propriétés intéressantes. En effet, la transformée en ondelettes 2D associe de bonnes localisations spatiales et fréquentielles. Comme nous travaillons sur des images médicales la localisation spatiale et fréquentielle sont importantes. Dans notre projet, nous avons appliqué la transformées en ondelettes suivie de la technique appelée SPIHT (Set Partitioning in Hierarchical Trees) pour la compression de différentes images médicales en niveau de gris pour différentes modalités de type BMP. [38]

L'algorithme SPIHT exploite les corrélations entre sous-bandes aussi bien entre deux niveaux consécutifs de résolution (relations père-fils) qu'entre sous-bandes plus éloignées (relations de petits-enfants, arrières petits-enfants...). [39]

## IV.2. Principe de la compression/décompression

Notre simulation est basée sur le schéma bloc de la figure IV.1. Les étapes de construction sont : acquisition de l'image, application de la transformée en ondelette (DWT), codage par l'approche SPIHT. Les étapes de reconstruction englobent : décodage SPIHT, transformation en ondelette inverse (IDWT), enregistrement de l'image reconstituée.

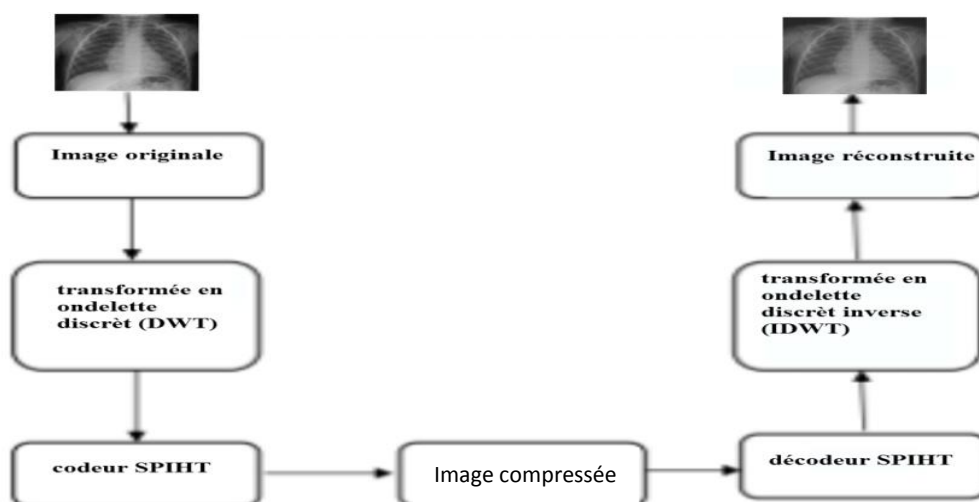


Figure IV.1 : Schéma de Principe de la compression.

Dans ce chapitre, nous allons résumer les résultats de tests pour la validation de l'algorithme méthode SPIHT.

Différents types d'images médicales ont été utilisées avec différentes tailles 512\*512, 256\*256 et 128\*128. Les images utilisées sont codées sur 8 bits pour chaque pixel. Les paramètres de tests sont le PSNR et le taux de compression (CR).

Nous avons aussi cherché à montrer l'intérêt du bon Choix de l'ondelette dans l'étape de transformation, pour cela nous allons effectuer une comparaison entre les ondelettes mères db4, bior4.4, Haar et sym4.

Un autre paramètre très important est le nombre de niveau de décomposition. Par la suite nous allons tirer conclusion du meilleur niveau.

### IV.3. Paramètres de validation

Les méthodes d'évaluations objectives de la qualité utilisent des métriques pour calculer mathématiquement la qualité de l'image : Le PSNR (PSNR : *Peak Signal to Noise Ratio*) ou rapport signal à bruit crête est une métrique qui se base sur le calcul de l'erreur quadratique moyenne EQM (MSE on anglais) entre les pixels de l'image de référence  $I$  et l'image distordue  $I'$ .

#### IV.3.1. Taux de compression

Le taux de compression (CR) est défini comme le rapport entre le nombre total de bits nécessaires pour représenter l'information originale et le nombre total de bits du fichier binaire à stocker qui résulte de la méthode de compression :

$$CR = \frac{\text{Taille de l'information compressée}}{\text{Taille de l'information originale}} \quad (\text{IV.1})$$

Ils évident que l'algorithme de compression le plus performant est celui qui atteint un CR le plus petit possible.

#### IV.3.2. MSE (Mean Square Error)

Le MSE représente l'erreur quadratique moyenne (EQM) entre l'image codée et l'image originale. Le MSE est défini par l'expression IV.2.

$$MSE = \frac{\sum_{n=1}^n \sum_{m=1}^m (I_{ij} - I_{ij}^*)^2}{mn} \quad (\text{IV.2})$$

Tels que  $I$  et  $I^*$  sont respectivement l'image originale et l'image codée de taille  $m' \times n$  où  $I_{ij}$  et  $I^*$  sont leurs composantes.

### IV.3.3. PSNR (*Peak Signal to Noise Ratio*)

Le PSNR est calculé par la relation suivante :

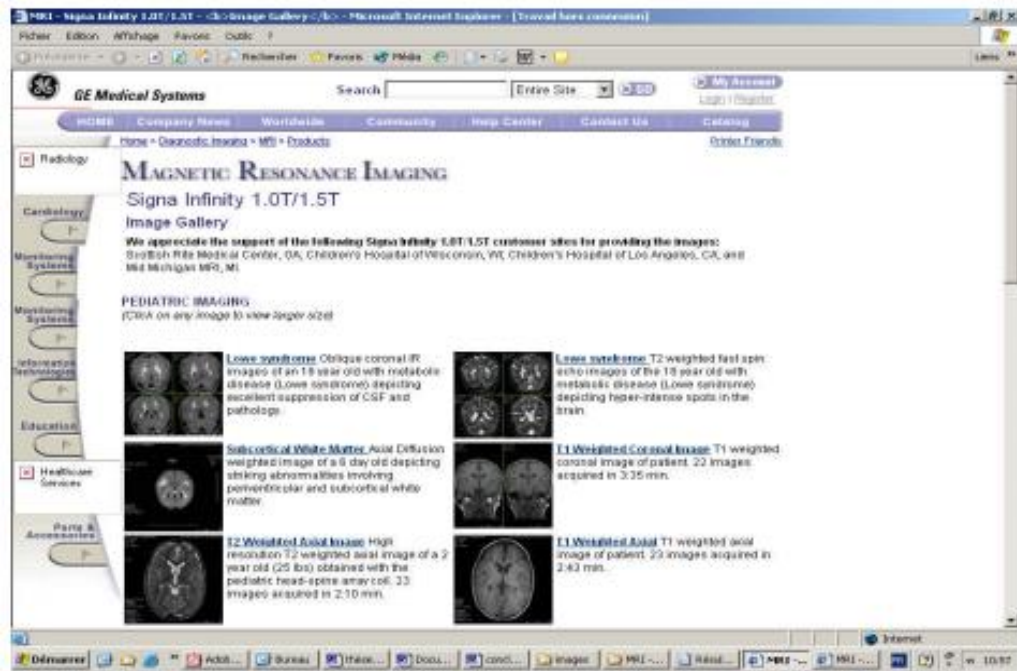
$$PSNR_{db} = 10 \log_{10} \frac{MN(2^n - 1)^2}{\sum_{i=1}^M \sum_{j=1}^N (I(i, j) - I'(i, j))^2} \quad (\text{IV.3})$$

Où  $M$  et  $N$  étant le nombre de lignes et de colonnes des images  $I$  et  $I'$  respectivement. Une valeur de PSNR supérieure à  $40 \text{ dB}$  est en général un indicateur d'une très bonne qualité d'image (i.e. la dégradation subie est en général imperceptible). [31]

## IV.4. Base de données des images médicales

### IV.4.1. Images médicales en niveaux de gris

L'ensemble des images utilisées pour les tests en compression au cours de ce mémoire sont des images médicales de différentes modalités : IRM, CT, Mammographie, Echographie. Elles sont acquises de la base de données GE-SYSTEM présentées à l'adresse de la référence [40].



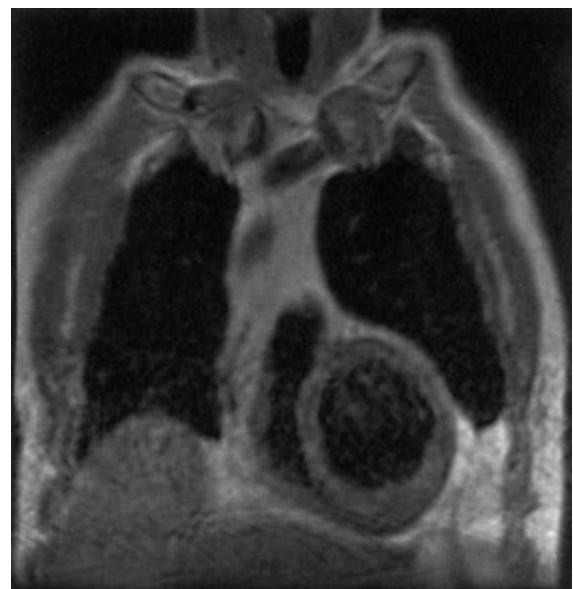
**Figure IV.2 :** Base de données des images médicales.

#### IV.4.1.1. Images de test de la base I

Les images de la première base sont en niveau de gris et de tailles  $512 \times 512$ . La figure IV.3 montre deux types d'images IRM de tailles  $512 \times 512$ .



(a) Poitrine



(b) appareil respiratoire

**Figure IV.3 :** Images test de la base I de taille  $512 \times 512$ .

### IV.4.1.2. Image de test de la Base II

Les images de la base II sont regroupées par type d'examen, à savoir : radiographies conventionnelle, coupes tomographiques et images par résonance magnétique. Exceptées "Angiot", "colon", et "us". L'image de la figure IV.4 est issue de la base de données [41]. La taille de cette image est 128 \* 128.



**Figure IV.4 :** Genou de la base II de taille 128 x 128.

Notre étude dans s'est concentrée que sur les trois images de test cités {poitrine, appareil respiratoire et genou} de taille 512\*512 ,256\*256.et 128\*128.

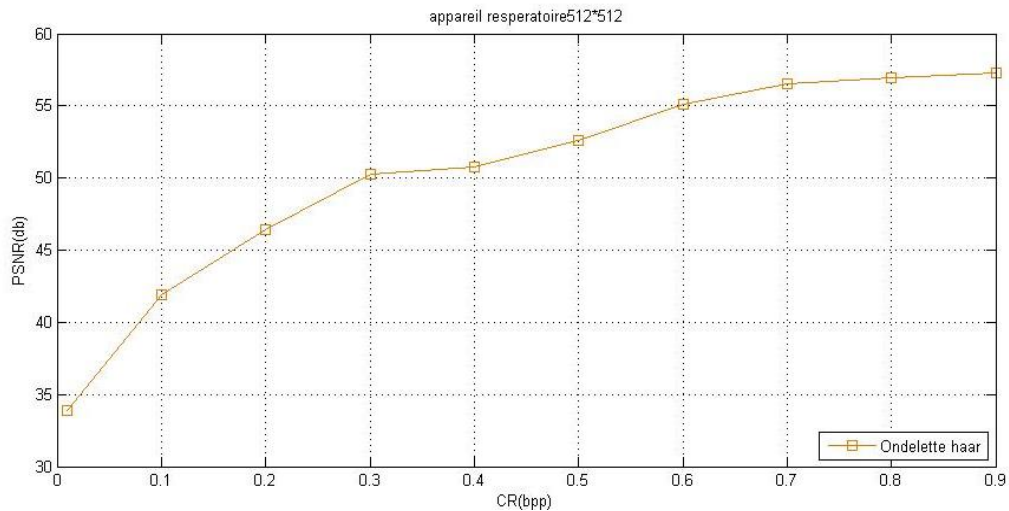
## IV.5 .Résultats et discussions

### IV.5.1. Choix de l'ondelette

Comme nous l'avons mentionné dans le chapitre II, nous allons revenir sur le protocole de choix de nos ondelettes et des principales propriétés pour avoir le meilleur résultat. L'ondelette mère peut être choisie de diverses manières en fonction des propriétés désirées sur la base de décomposition.

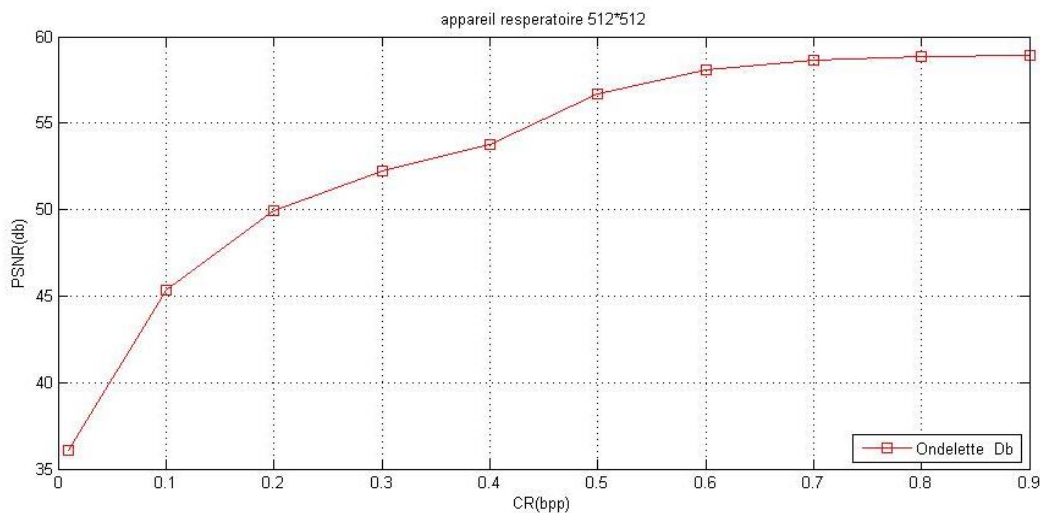
L'utilisation de l'ondelette Haar nous montre dans la courbe de la figure IV.5 que le PSNR augmente en fonction du taux de compression. Avec un algorithme de compression/décompression, on cherche toujours à avoir un CR soit petite. Au même temps la

valeur du PSNR correspondant à cette valeur de CR soit élevée et soit entre 27 et 40 db. La figure IV.5 montre qu'avec un CR de 0.01 bpp, le PSNR est égale à 33.86 db.



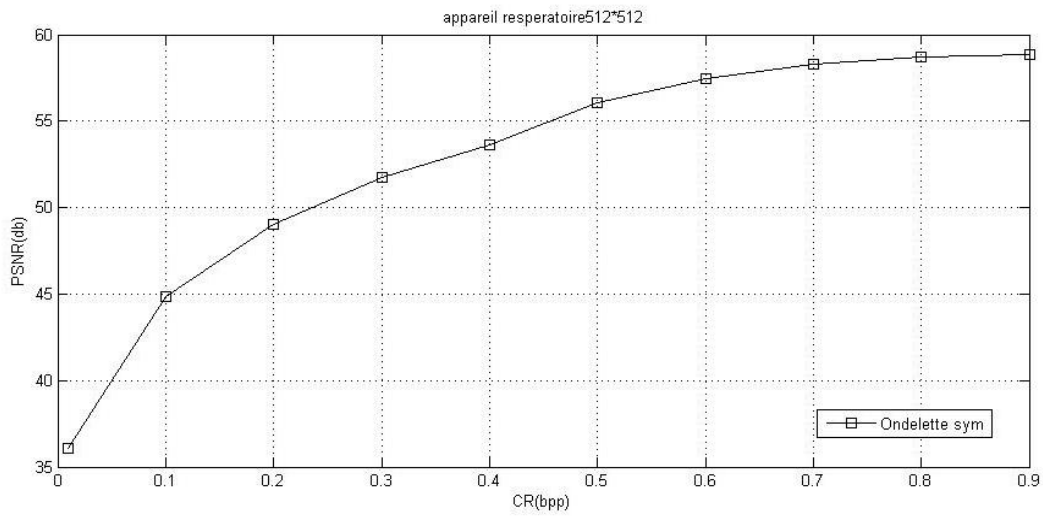
**Figure IV.5 :** Compression SPIHT utilisant l'ondelette Haar de l'image appareil respiratoire.

Nous avons changé une nouvelle ondelette s'appelle Daubechies 4 (Db4). Les résultats ont été améliorés de point de vue augmentation du PSNR. L'intervalle du PSNR est entre 36 db et 59 db pour des CR allant de 0.01 et 0.9 (voir figure IV.6).



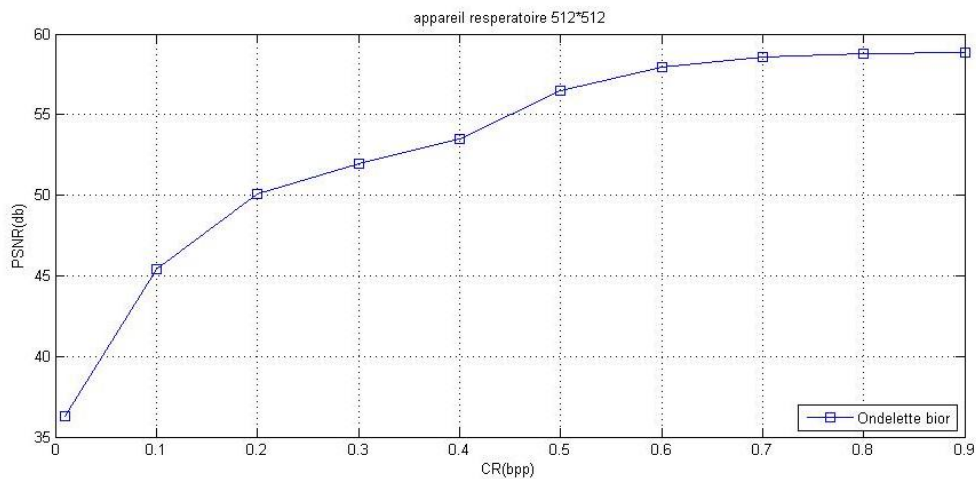
**Figure IV.6 :** Compression SPIHT utilisant l'ondelette DB4 de l'image appareil respiratoire.

L'utilisation de l'ondelette Symlet 4 (Sym 4) a aussi donné une autre amélioration du PSNR par rapport à l'ondelette Haar. Les résultats obtenus par l'ondelette Sym 4 sont plus ou moins proches de ceux obtenus par db4 (voir figure IV.7).



**Figure IV.7 :** Compression SPIHT utilisant l'ondelette Sym 4 de l'image appareil respiratoire.

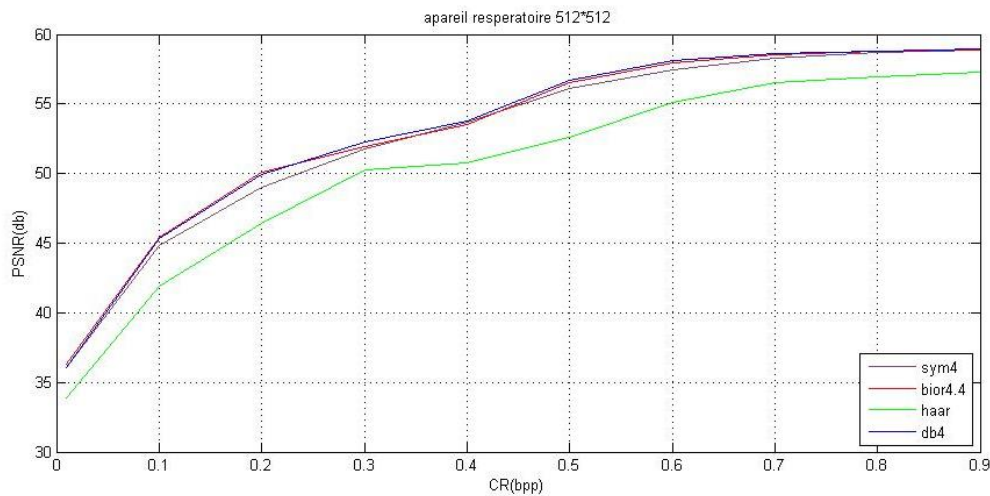
Presque les mêmes résultats ont été obtenus utilisant l'ondelette Biorthogonale 4.4 (bior 4.4). Pour un CR de 0.9, le PSNR est 59 db et pour un CR de 0.01 le PSNR est 36 db (voir figure IV.8).



**Figure IV.8 :** Compression SPIHT utilisant l'ondelette Bior 4.4 de l'image appareil respiratoire.

D'après les courbes précédentes nous avons extrait le meilleur type d'ondelette que nous allons l'utiliser dans les prochains tests. Nous avons eu de très bons résultats de compression SPIHT avec l'ondelette bior 4.4. Avec une telle ondelette mère, la figure IV.9 montre que la courbe du PSNR

en fonction du CR dépasse les autres courbes. Le PSNR égal à 36,3168 db pour un CR=0.01 et pour un CR égal à 0.9 le PSNR était 58.93 db.



**Figure IV.9 :** Comparaison entre les quatre types d'ondelettes pour la compression SPIHT de l'image appareil respiratoire

#### IV.5.2. Niveau de décomposition

Un autre critère très important dans la compression est le niveau de décomposition de la transformée en ondelettes. Dans notre travail on a choisi seulement quelques résolutions pour justifier le meilleur choix afin d'avoir une efficacité élevée.

Les tableaux IV.1 résument les résultats obtenus pour les niveaux de décomposition 1, 2, 3, 4, 5, 6 et 7. L'image de test est l'appareil respiratoire 512x512. L'ondelette utilisée est bior 4.4.

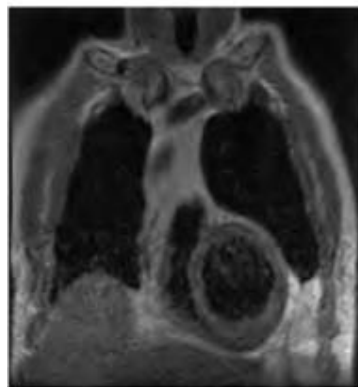
CR	PSNR (db)						
	Niveau 1	Niveau 2	Niveau 3	Niveau 4	Niveau 5	Niveau 6	Niveau 7
0.01	8.04	12.77	12.80	23.58	35.30	36.29	34.65
0.1	12.84	18.32	38.16	44.53	45.54	45.73	42.92
0.2	12.84	28.09	46.80	49.87	50.60	50.71	47.30
0.3	18.28	37.20	51.54	53.88	54.73	54.85	51.41
0.4	18.28	42.45	55.81	60.20	61.49	61.70	55.64
0.5	22.93	47.76	63.44	72.41	76.81	77.36	63.95

**Tableau IV.1 :** Résultats de compression de l'image appareil respiratoire taille 512 \* 512 pour différents niveaux de décomposition.

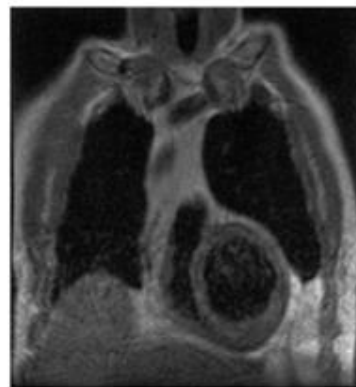
Pour chaque valeur de CR dans le tableau IV.1, tout en augmentant le niveau de décomposition jusqu'au niveau 6 le PSNR augmente. Pour le niveau 7, on remarque que le PSNR a été affaibli par rapport au niveau 6. On peut dire que le niveau de décomposition 6 donne les meilleurs résultats par rapport aux autres niveaux 1, 2, 3, 4, 5 et 7. cela est du d'un coté, à l'exploitation des caractéristiques des transformées en ondelettes appliqués aux images et d'un autre coté à la relation pères, fils et descendants.

Pour des CR fixés à 0.01, 0.1, 0.2, 0.3, 0.4 et 0.5, les PSNR calculés pour le niveau 6 varient entre 36.29 db et 77.36 db. Par contre les autres niveaux les PSNR sont plus faibles.

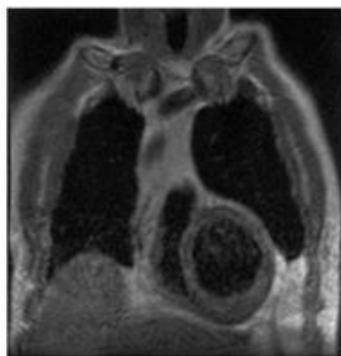
L'image appareil respiratoire de taille 512 \* 512 originale et les images resituées pour des CR 0.01, 0.1 et 0.4 sont affichées dans la figure IV.10. La qualité obtenue dans le cas le plus défavorable est celle pour un CR égal à 0.01. L'image reconstituée est d'une qualité très acceptable (figure IV.13).



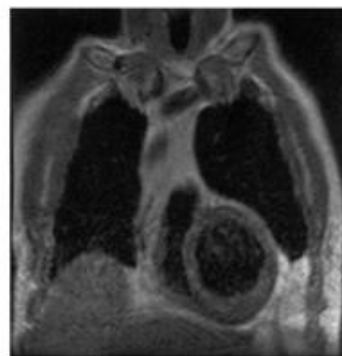
(a) Image originale



(b) PSNR=36.29 db, CR=0.01



(c) PSNR=45.73 db, CR=0.1



(d) PSNR=61.70 db, CR=0.4

**Figure IV.10 :** Résultats de l'image appareil respiratoire 512 \*512.

### IV.5.3. Taille de l'image

Une image médicale se compose d'une matrice de pixels contenant les valeurs de luminosité. Nous avons utilisé pour nos tests les tailles suivantes : 512 x512 , 256 x256 et 128 x128. Ceci a pour objectif de montrer l'influence de la taille de l'image sur les résultats.

L'image appareil respiratoire originale est de taille 512\*512. Par Matlab, nous avons réduit la taille de cette image afin d'obtenir les tailles 256\*256 et 128 \*128. Le tableau IV.2 donne les résultats obtenus pour les trois tailles utilisés de l'image respiratoire. L'ondelette utilisée est bior4.4 avec six niveaux de décomposition.

CR	PSNR (db)		
	512*512	256*256	128*128
0.01	36.29	30.20	25.52
0.1	45.73	39.87	35.89
0.2	50.71	43.82	40.57
0.3	54.85	47.65	44.53
0.4	61.70	51.35	48.40
0.5	77.36	55.71	52.53

**Tableau IV.2** : Résultats de compression de l'image appareil respiratoire pour différentes tailles.

On constate que le tableau IV.2 montre que pour une taille 512\*512 et un CR égal à 0.01, le PSNR est 36.29 contre 30.20 et 25.52 pour les autres tailles. On peut dire que les résultats pour une image de taille plus grande sont améliorés par rapport aux autres tailles. Aussi, nous avons besoin de la compression surtout pour les images de tailles plus grandes.

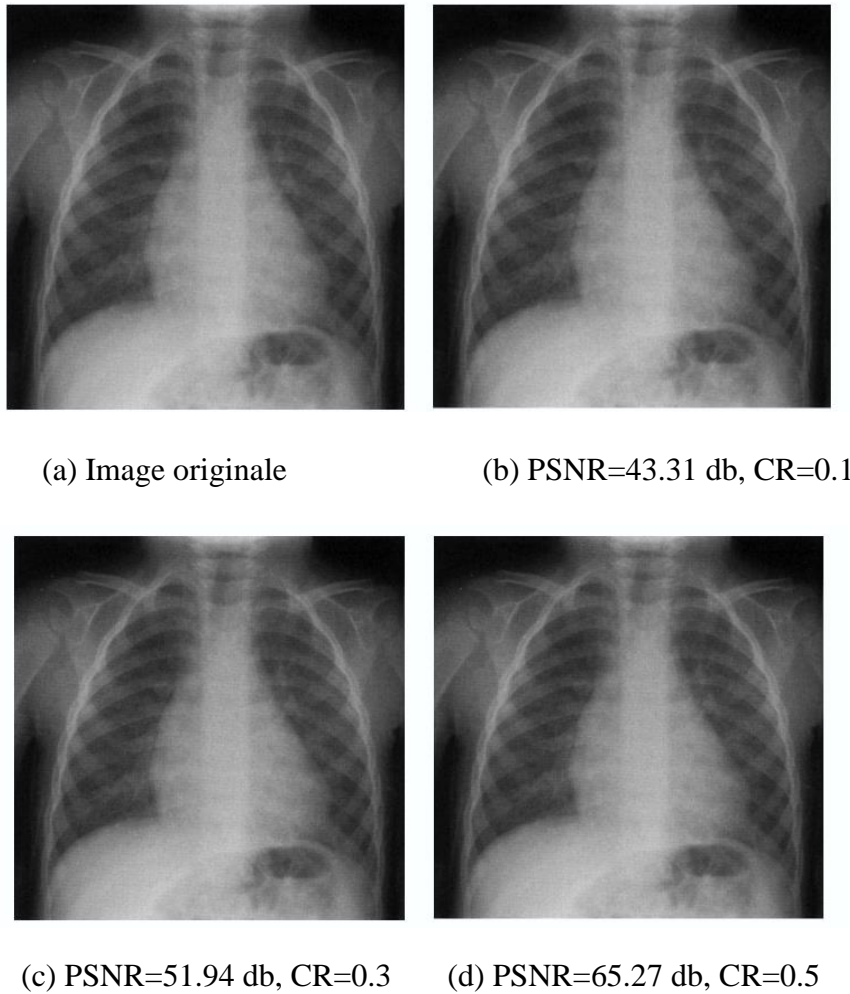
D'autres tests ont été effectués sur d'autres types d'images avec de différentes tailles. Le tableau IV.3 donne les résultats obtenus pour l'image poitrine de taille 512\*512. Les mêmes paramètres ont été fixés pour le type d'ondelettes bior4.4 et le niveau de décomposition 6.

CR	0.01	0.1	0.2	0.3	0.4	0.5
PSNR (db)	36.31	43.31	47.86	51.94	56.42	65.27

**Tableau IV.3** : Résultats de compression de l'image poitrine de taille 512\*512.

La qualité visuelle de l'image poitrine est affichée dans la figure IV.11. Les trois figures reconstituées (b), (c) et (d) ont de très bonnes qualités de reconstruction pour les rapports de compression 0.1, 0.3 et 0.5.

Donc ce tableau montre bien que si la taille de l'image est grande, nous obtenons une grande valeur du PSNR due à une très bonne amélioration de compression.



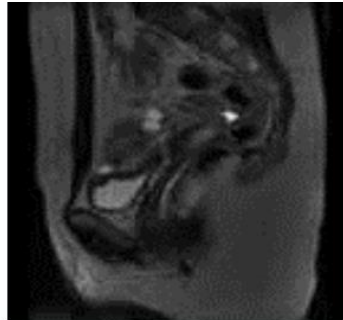
**Figure. IV.11** : Résultats de l'image poitrine 512 \*512.

Une autre image de test "genou" a été compressé par la technique SPIHT. Toujours les mêmes paramètres ont été conservés. C'est une image d'origine carré et de taille 128\*128. On obtient les mêmes commentaires que pour les images appareil respiratoire et poitrine. Ce qui confirme l'efficacité de la méthode SPIHT.

CR	0.01	0.1	0.2	0.3	0.4	0.5
PSNR (db)	36.37	45.03	50.30	54.54	61.16	76.65

**Tableau IV.4** : Résultats de compression de l'image genou de taille 128\*128.

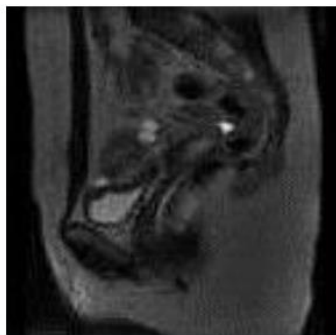
La figure IV.12 illustre les images restituées pour des CR allant de 0.01 jusqu'à 0.5. Les qualités obtenus sont acceptables surtout pour les CR 0.3 et 0.5.



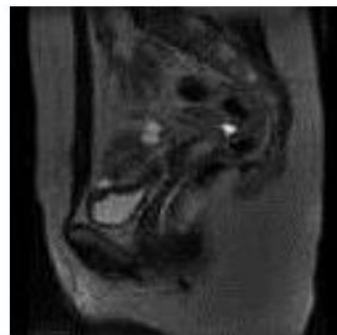
(a) Image originale



(b) PSNR=36.37 db, CR=0.01



(c) PSNR=61.16 db, CR=0.4



(d) PSNR=76.65 db, CR=0.5

**Figure. IV.12 :** Résultats de l'image genou 128 \*128 reconstruite par SPIHT au niveau 6

## IV.6. Conclusion

A travers ce chapitre, nous avons effectué plusieurs tests pour évaluer la technique SPIHT et pour vérifier l'influence de plusieurs paramètres à savoir: changement du type d'ondelette, niveau de décomposition, type d'image et taille de l'image. Les critères utilisés sont le taux de compression (CR) et le rapport signal sur bruit (PSNR).

Nous avons constaté que jusqu'à 10% de la taille de l'image médicale, SPIHT reste très performante (bonnes qualités de reconstruction).

## Conclusion générale :

La compression des données et particulièrement des images date déjà de plusieurs décennies. Cependant, elle est toujours d'actualité compte des exigences technique et technologique liées aux multimédias d'une manière générale et aussi aux moyens de communications comme les réseaux informatiques. De nombreux algorithmes ont vu le jour et certains sont devenus des normes et des standards. Les techniques de compression avec pertes assurent généralement des taux de compression relativement élevés mais avec un certain degré de dégradation. Les objectifs des nouvelles techniques de compression est donc de réduire plus et d'altérer les moins possible.

Dans ce mémoire , nous avons exposé la compression dite "avec perte". Nous avons ensuite investigué, à savoir les méthodes de compression qui reposent sur la transformée en ondelettes et par codages (Technique SPIHT),. Ces méthodes offrent la possibilité de diminuer considérablement les taux de compression à qualité image équivalente. L'idée est d'élaborer un outil numérique capable de compresser une image médicale avec les principales contraintes suivantes:

- ❖ Taux de compression .
- ❖ PSNR

L'objectif de ce travail est l'évaluation de la technique pour l'amélioration de la compression des images médicales en niveau de gris .

Dans la première partie de notre travail, nous avons passé en revue certaines de ces nouvelles transformées, dites de secondes génération, et leurs applications dans la compression d'image. Ce sont des décompositions multi-échelles, qui opèrent selon une multitude d'orientations fréquentielles et qui offrent un bon compromis entre la représentation éparse (ou compacte) de l'information visuelle et la qualité perceptuelle de l'image reconstruite. Nous avons opté dans ce mémoire particulièrement, la transformée en d'ondelette.

En effet, en deuxième lieu nous avons appliqué la technique de compression qui est basée sur la DWT et opérant sous forme progressive à savoir l'approche SPIHT. Il s'agit d'une amélioration d'une autre technique de la même espèce en l'occurrence EZW. Nous avons donc mis au point cet algorithme et nous l'avons testé et évalué sur plusieurs images de tests. Les résultats sont très concordants avec les travaux d'autres chercheurs dans le même domaine.

D'autre part, ces résultats prouvent bien que cet algorithme est très performant essentiellement pour certains types d'images.

A travers le protocole de validation que nous avons suivi dans le chapitre 4, nous avons prouvé que l'algorithme SPIHT est très performant du point de vue rapport taux de compression/qualité d'image.

## Bibliographie

- [1] Boukli Hacene Ismail, « Compression d'images médicales par ondelettes de seconde génération » Thèse de Doctorat En Electronique biomédicale, Setembre 2014.
- [2] Ouafi Abdelkrim : « Compression d'images avec pertes par codages imbriqués, Proposition d'une optimisation de l'algorithme EZW » Thèse de Doctorat en Sciences en Electronique, 2008/2009.
- [3] A.W. Wong, R.K. Taira, and H.K. Huang, "Implementation of a digital archive system for aradiology department", in Proc. SPIE Conf. on Medical Imaging VI: PACS Design and Evaluation 1645, pp. 182-190, 1992.
- [4] Bradley J.Erickson, "Irreversible Compression of Medical Images", Society for computer applications in radiology, 2000.
- [5] P. Raf y, Y. Gaudeau, D. P. Miller, et J-M. Moureaux, "Computer Aided Detection (CAD) of Solid Lung Nodules in Lossy Compressed MDCT Chest Exams", ECR, Vienne, mars 2006.
- [6] S .Benabdelkader, « compression des images satellites monochromatique via la transformation en ondelettes », thèse de magister, univ de Constantine, 1996.
- [7] R .J .Clarke, « transform coding of images »,Academic press INC ,London 1985 .
- [8] Sahir Mourad, « Compression des images numériques par la technique des ondelettes», thèse de Magister, univ. de Ferhat Abbas-Setif,2011.
- [9] E. Incerti, « Compression d'images, algorithmes et standards», Edition Vuibert, Paris, 2003.
- [10] Khalid sayood, «lossless compression Handbook», academic press series in communication, networking,and multimedia,Elsevier Science, New York,2003.

- [11] Tinku Acharya, Ping-Sing Tsai, «JPEG 2000 standard for image compression concepts, Algorithms and VLSI architectures »,John Wiley & Sons , Inc, New Jersey 2005.
- [12] José Marconi M.Rodrigues, «Transfert sécurisé d'images par combinaison de technique de compression cryptage et marquage»,Thèse doctorat de l'univ Montpellier 2006.
- [13] David Salomon, «Data compression » the complete reference 3<sup>eme</sup> edition, Edition Spring, 2004.
- [14] D. A. Human. « method for the construction of minimum-redundancy codes». In Proceedings of the Institute of Radio Engineers, volume 40, sep 1952.
- [15] Jacques CLAVIER « théorie de l'information –Aspects mathématiques»,techniques de l'ingénieur , traité électronique, E3082-1.
- [16] Redha Benzid, « ondelette et statistiques d'ordre supérieur Appliquées aux signaux uni et bidimensionnels»,Thèse de doctorat univ. de Batna ,2005 .
- [17] Cheniti Ilyes, M .Bouderouaz Tarek, « les ondelettes dans la compression des images », Mémoire de fin d'études d'Ingéniorat en génie électronique; univ. de M'sila Juin 2006.
- [18] Théodore Totozafiny «compression d'images couleur pour application à la télésurveillance routière par transmission vidéo à trèsbas débit», Thèse Doctorat de l'univ. de pau et des pays de l'adour, 3juillet 2007.
- [19] M. N. Do and M. Vetterli, "The finite ridgelet transform for image representation,"IEEE Trans.Image Processing, vol. 12, no. 1, pp. 16–28, Jan. 2003.
- [20] Minh. N. Do, "Directional multiresolution image representations," D. Phil.Dissertation, Department of Communication Systems, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, October 2001.

- [21] S. Mallat, A wavelet tour of signal processing, Academic Press, San Diego, 2nd edition, 1999.
- [22] S. A. Khayam, «The discrete cosine transform (DCT): theory and application, » Department of Electrical & Computer Engineering, Michigan State University, March, 2003.
- [23] livre «technique de compression d'images », Boukelif Aoued, page 156
- [24] David F. Walnut (2001). An Introduction to Wavelet Analysis. Springer, 450p.
- [25] Frédéric Truchetet (1998) Ondelettes pour le signal numérique. Editions Hermes.
- [26] Rene Alt, « La transformation en ondelettes », Université Pierre et Marie Curie, Paris.
- [27] Barhoumi Khaled, « Approche par marquage pour l'évaluation de la qualité d'image dans les applications multimédias », mémoire pour l'obtention du grade de Maîtrise en informatique, Univ. de Quebec en Ottawa, Novembre 2012.
- [28] M.L. Hilton, «Wavelet and Wavelet Packet Compression of Electrocardiograms », IEEE Trans. Biomed. Eng., vol. 44, pp. 394-402, May 1997.
- [29] A. Boukaache «Compression d'images hautes définitions en utilisant une transformation hybride» Thèse de doctorat 2012 Guelma algérie.
- [30] Ouafi, Abdelkrim. Compression d'images fixes biomédicales par les transformées en ondelettes associées aux algorithmes de quantification vectorielle et de codage entropique. Mémoire de Magister. s.l. : Université Mohamed Khider de Biskra, 2001.
- [31] S. G. Mallat. A Wavelet Tour of Signal Processing. Academic Press, 1997.
- [32] [www.enseeiht.fr/~mailhes](http://www.enseeiht.fr/~mailhes)
- [33] V. Chappelier "Codage progressif d'images par ondelettes orientées" Thèse de Doctorat. Université de Rennes 1, 2005.

- [34] A. Boukaache ‘‘Compression d’images hautes définitions en utilisant une transformation hybride’’ Thèse de doctorat 2012 Guelma algérie
- [35] S. Chang & L. Carin, "A Modified SPIHT Algorithm for Image Coding With a Joint MSE and Classification Distortion Measure," *IEEE Trans. Image Processing*, 15(3), pp. 713-725, 2006.
- [36] S.G.Mallat. (1989). "A Theory for Multiresolution Signal Decomposition : the Wavelet Representation", *IEEE PAMI*, Vol.11(7), pp.674–693.
- [37] Pearlman, W. A et B.J. Kim (1997). "An Embedded Wavelet Video Coder Using Three Dimensional Set Partitioning in Hierarchical Trees (SPIHT). In *IEEE Data Compression Conference DCC'97*, p 221-260.
- [38] BELADGHAM.M, « Construction d’une technique d’aide au diagnostic en Imagerie médicale. Application à la Compression d’images», thèse de doctorat en électronique, université de Tlemcen. Décembre 2012 .
- [39] R.W. BUCCIGROSSI, E.P. SIMONCELLI, "Image compression via joint statistical characterization in the wavelet domain", *IEEE Trans. Image processing*, Vol. 8, pp.1688–1701, December 1999.
- [40] <http://www.gemedicalsystems.com>
- [41] <http://www.cis.rit.edu/htbooks/mri/inside.htm>