

M/621.783

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique  
Université 8Mai 1945 – Guelma  
Faculté des sciences et de la Technologie  
Département d'Electronique et Télécommunications



## Mémoire de fin d'étude

pour l'obtention du diplôme de Master Académique

Domaine : Sciences et Techniques  
Filière : Electronique  
Spécialité : Systèmes d'Electronique



---

# Compression d'images fixes par EZW (Embedded Zerotrees Wavelet)

---

Présenté par :

**Boukerche FAROUK et Amrani ABDRAHIM**

Sous la direction de :

**Dr. Doghmaine HAKIM**

JUIN 2013

# Remerciements

13/2947



*Au terme de ce mémoire nous tenons à remercier en premier lieu Dieu, le miséricordieux, qui nous a donné la force, volonté et courage tout au long de l'élaboration de ce travail.*

*Nos chaleureux remerciements et nos profondes gratitude vont à notre encadreur Mr. DOGHMAN HAKIM qui nous a orientés vers la bonne voix avec ses remarques pertinentes.*

*Nous tenons à remercier également les membres du jury qu'ils trouvent ici l'expression de notre reconnaissance et notre profond respect.*

*Nos vifs remerciements à tous mes professeurs de notre département d'électronique et télécommunications qui ont contribué à ma formation notre premier cycle d'étude jusqu'à la fin de notre cycle universitaire.*

*Je tiens tout particulièrement à remercier BELOUCIF TAKIY-EDDINE ami et frère pour son aide et ses conseils, il a toujours été présent, attentionné, patient.*

*Enfin, nous adressons nos remerciements à tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail.*

# Dédicace

*Je dédie ce modeste travail :*

- ✎ *A mes très chers parents qui sont la cause de m'être un jour un Cadre qui peut donner quelque chose à sa patrie.*
- ✎ *A Mes frères.*
- ✎ *A Mes sœurs FATMA et ZAHIA et toute la famille sans exception.*
- ✎ *A tous ceux qui m'ont aidé durant ma vie universitaire.*
- ✎ *A tous la promotion d'électronique.*

✎ FAROUK ✎

٢٠٠٠

*Je dédie ce modeste travail :*

- ✎ *A Mes très chères parents, pour leur sacrifice et leur dévouement pour mon bonheur,*
- ✎ *A Tous Mes amis,*
- ✎ *A ceux qui m'aiment.*

✎ ABDRAHIM ✎

# SOMMAIRE

Liste des figures.....	i
Liste des tableaux.....	ii
Introduction générale.....	1

## **Chapitre 01: Généralités sur l'imagerie.**

1. Introduction.....	3
2. Les notions de base sur l'imagerie.....	3
2.1. Définition d'une image.....	3
2.2. Pixels.....	6
2.3. L'image monochrome.....	6
2.4. L'image couleur.....	6
2.5. La résolution d'une image.....	6
2.6. Le contraste.....	7
2.7. Niveau de gris.....	7
2.8. Contours et textures.....	8
2.9. Le bruit.....	8
3. Conclusion.....	8

## **Chapitre 02: Etat de l'art sur les techniques de compression.**

1. Introduction.....	10
2. Les techniques de compression de donnée.....	10
2.1. Les techniques de compression sans pertes.....	10
2.1.1. Codage d'Huffman.....	11
2.1.2. Cordage RLE (Run Length Encoding).....	12
2.1.3. Cordage arithmétique.....	13

2.1.4. Propriétés des techniques de compression sans perte.....	14
2.2. Méthodes avec perte.....	14
2.2.1. Quantification vectorielle et Quantification scalaire.....	14
3. principe de compression .....	16
4. Codage Direct (Codage spatial).....	17
4.1. Le codage MIC (Modulation par Impulsion Codée).....	17
4.2. Le codage MICD (Modulation par Impulsion Codée Différentielle).....	18
4.3. Le codage prédictif .....	18
5. Le codage par transformée.....	19
5.1. La compression par DCT (Discret cosine transform).....	19
5.2. JPEG (Joint Photographic Experts Group).....	21
5.3. EZW (Embedded Zerotrees Wavelet) .....	21
5.4. JPEG2000 (Joint Photographic Experts Group).....	21
6. Conclusion.....	22

### **Chapitre 03: L'analyse multi-résolution (Ondelettes).**

1. Introduction.....	23
2. Transformé de Fourier .....	24
3. Transformée en ondelette.....	25
3.1. Définition d'une ondelette.....	26
3.2. Transformé en ondelette continue.....	26
3.3. Quelques propriétés de la transformée d'ondelette.....	27
3.4. Transformé en ondelette discrète .....	28
3.5. Analyse multi-résolution .....	28
3.6. Quelques ondelettes .....	30

a. Ondelettes de Haar.....	30
b. Ondelettes de Daubechies.....	30
3.7. Les Ondelettes en imagerie .....	31
4. Conclusion.....	31

**Chapitre 04 : La compression d’images par Embaded Zerotree Wavelet (EZW).**

1. Introduction.....	32
2. Les différentes méthodes de compression.....	32
2.1. JPEG 2000(Joint Photographic Experts Group).....	32
2.2. EBICOT (Embeded Block Coding With Optimized Truncation).....	33
2.3. EWZ (Embeded Zerotrees Wavelet).....	34
2.3.1. L’algorithme d’EZW.....	36
3. Résultats et discussions.....	39
3.1. La compression par DCT.....	40
3.2. La compression par JPEG.....	43
3.2. La compression par EZW.....	44
4. Conclusion.....	46
<b>Conclusion général.....</b>	<b>47</b>

# Liste Des Figures

## Chapitre 01

Figure 1.1: Processus de numérisation.....	4
Figure 1.2 : Images Matricielles.....	4
Figure 1.3 : Image vectorielle.....	5
Figure 1.4 : Exemples de résolutions sur une même surface.....	6
Figure 1.5 : Exemple d'image codée au niveau de gris.....	7

## Chapitre 02

Figure 2.1 : Principe du quantificateur vectoriel.....	16
Figure 2.2 : principe d'un CODEC de type DPCM.....	18
Figure 2.3 : Principe du codage prédictif à l'aide d'un filtre RIF à 4 coefficients.....	19
Figure 2.4 : Schéma synoptique d'un CODEC à base de DCT.....	20
Figure 2.5 : Principe de la troncature par zone.....	20
Figure 2.6 : Principe de codage utilisé par le groupe JPEG.....	21

## Chapitre 03

## Chapitre 04

Figure 4.1 : Illustration de la structure de donnée nommée quadtree.....	33
--	----

Figure 4.2 : Ordre de Morton.....	35
Figure 4.3 : Ordre de Morton.....	37
Figure 4.4: L'image originale de Cameraman.....	40
Figure 4.5: Résultats de la méthode DCT sur l'image Cameraman avec un masque carré.....	41
Figure 4.6: Résultats de la méthode DCT sur l'image Cameraman avec un masque diagonal.....	42
Figure 4.7: Résultats de la méthode JPEG sur l'image cameraman.....	43
Figure 4.8: Compression par EZW avec un bitrate =1.43bpp (ezw_encoding_threshold=10).....	44
Figure 4.9: Compression par EZW avec un bitrate =7.62bpp (ezw_encoding_threshold=1).....	45



# Liste Des Tableaux

Tableau (4.1) : Variation du RC et SNR en fonction de la taille du masque carré (P.40).

Tableau (4.2) : Variation du RC et SNR en fonction de la taille du masque diagonal (P.40).

# *Introduction Générale*

# Introduction

Depuis l'avènement de l'ère numérique, la quantité d'information échangée sur les réseaux du monde entier nécessite un traitement de l'information, visant à réduire la taille des données, tout en conservant leur intégrité. La théorie de l'information est née avec Claude Shannon (1916 – 2001) qui démontra que l'information pouvait être traitée de manière statistique, et non de manière déterministe. Nous nous plaçons au carrefour de l'audiovisuel (avec la vidéo à la demande, la télévision interactive...etc.), l'informatique (avec l'indexation vidéo, l'édition vidéo...etc.) et les télécommunications (avec la visiophonie, la vidéo conférence...etc.).

La formidable avancée technologique de ces dernières années, la baisse des coûts du matériel et l'accès à des plate-formistes pour le grand public incite les fournisseurs de contenus à proposer de plus en plus de choix dans le mode de transmission pour la vidéo. En outre, l'augmentation des débits dans les réseaux mondiaux a permis le développement d'application comme la télévision sur Internet ou la voix sur IP. Parallèlement, des efforts sont entrepris pour diminuer la taille des données classiques (vidéos, images, musiques, sons.....) que pour les données classiques (documents, pages web...). Les données les plus volumineuses restent à ce jour les images et les contenus vidéo, qu'il convient de les traiter en tenant compte de leurs spécificités.

Le principe général de toute compression est la réduction de la taille des données, en construisant des messages qui résumeront l'information initiale par élimination des redondances. La borne inférieure théorique pour une distorsion du signal donnée est fournie par le théorème de Shannon pour le codage de source :

<< L'information véhiculée par un signal dont le spectre est à support borné n'est pas modifiée par l'opération d'échantillonnage à condition que sa fréquence soit au moins deux fois plus grande que la plus grande fréquence contenue dans le signal. La reconstitution du signal original peut être effectuée par un filtre passe-bas idéal de fréquence de coupure égale à la moitié de la fréquence d'échantillonnage. >> Les

méthodes de compression tentent de trouver le meilleur compromis possible entre le débit final et la distorsion induite par le résumé de l'information.

C'est dans cette perspective que nous plaçons nos recherches, l'objectif étant de diminuer la taille des données des images transmises sur les réseaux. Nous proposons dans ce manuscrit un algorithme de codage et de permettre une adaptation aux possibilités qu'offrent les différents types de réseaux. Parallèlement, de nouvelles techniques de compression sont investiguées, toujours en tenant compte des spécificités des contenus. Les contributions de cet mémoire sont essentiellement tournées vers la création d'un codeur des images fixes. Nous contribuons également à l'élaboration des outils au système de codage lui-même, avec un contrôle débit-distorsion, un système de seuillage des coefficients Ondelettes est introduit.

Ce mémoire est organisé en quatre chapitres. Dans le premier chapitre, nous avons parlé sur les notions de base de traitement d'images.

Puis dans le chapitre 2, un état de l'art sur la compression d'images était présenté.

Dans le troisième chapitre une brève introduction sur la théorie des Ondelettes ainsi que leurs propriétés étaient présentées.

Dans le quatrième chapitre de ce mémoire l'algorithme de compression par EZW sera décrit ou cet algorithme est basée sur la transformée en ondelette.

Enfin, la dernière partie de ce mémoire est consacrée pour les résultats obtenus avec cette approche avec une conclusion générale.

## *Chapitre I*

# *Généralités sur l'imagerie*

# Généralités sur l'imagerie

## 1. Introduction:

Avec la parole, l'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre.

C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises. Donc, l'image était, depuis l'aube de l'humanité, un élément le plus significatif dans le domaine de l'échange de l'information, et elle est toujours d'ailleurs, sa puissance réside dans le fait qu'une image à elle seule, puisse remplacer des quantités considérables des mots.

Dans ce chapitre nous allons présenter des généralités et quelques notions fondamentales concernant le domaine d'imagerie.

## 2. Les notions de base sur l'imagerie :

### 2.1. Définition d'une image :

Une image est une représentation spatiale d'un objet, d'une scène bidimensionnelle ou tridimensionnelle. Dans la littérature, on distingue deux types d'images :

#### *a. L'image analogique :*

Elle est assimilable à une fonction continue  $I(x,y)$  de deux variables réelles  $x$  et  $y$ . Elle est définie sur une région rectangulaire bornée du plan. La valeur du pixel au point  $(x,y)$  est proportionnelle à l'énergie lumineuse reçue au niveau du capteur d'image, ainsi l'image est appelée aussi image d'intensités ou de luminances [1].

#### *b. Image numérique:*

En traitement de signal, une image est définie comme étant un signal bidimensionnel. Pour représenter une image par un outil informatique (ordinateur, appareils numérique,...etc.), l'image a besoin d'une numérisation qui consiste à diviser la surface de

l'image en éléments de taille fixe appelés cellules (pixels), ayant chacun comme caractéristique les densités de couleurs de bases codé sur n bits. [1]

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée dans un espace à deux dimensions. Le processus de numérisation se décompose en trois étapes (figure (1.1)).

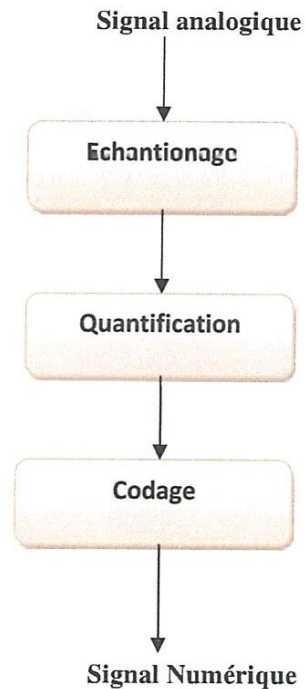


Figure (1.1): Processus de numérisation.

Il existe deux sortes des d'image numériques : les images matricielles et les images vectorielles :

- *Image matricielle* (bitmap) : Une image matricielle est formée d'un tableau de points ou pixels, auxquels on peut associer une couleur et une intensité comme il est représenté sur la figure 1.2.



Figure (1.2) : Images Matricielles.

Plus la densité des points sont élevée, plus le nombre d'informations est grand et plus la résolution de l'image est élevée.

Les images vues sur un écran de télévision ou une photographie sont des images matricielles. On obtient également des images matricielles à l'aide d'un appareil photo numérique, d'une caméra vidéo numérique ou d'un scanner.

Contrairement à une image vectorielle les images matricielles ne supporte pas les modifications.

- *Image vectorielle* : dans ce format les données sont représentées par des formes géométriques simples qui sont décrites d'un point de vue mathématique (cercle, rectangle ...etc.).

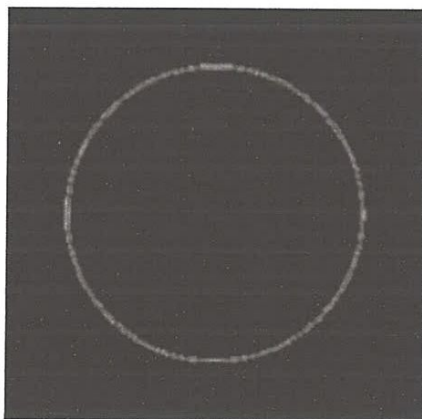


Figure (1.3) : Image vectorielle.

Puisque ces images sont représentés par des formules mathématiques (une courbe est défini par deux points, un cercle par un centre et un rayon). Donc c'est le processeur qui sera chargé de "traduire" ces formes en informations (interprétables par la carte graphique).

Ces images présentent deux avantages : elles occupent peu de place en mémoire et peuvent être redimensionnées sans perte d'information.

## 2.2. Pixels :

Le pixel (abréviation venant de l'anglais : Picture élément) est l'élément de base d'une image. Le pixel se caractérise par ses coordonnées et son intensité lumineuse (niveau du gris). Donc le pixel est le plus petit élément que peuvent manipuler le matériel et les logiciels d'affichage et d'impression [2].



### 2.3. L'image monochrome :

Dans le contexte des images monochromes noir et blanc, qui nous intéresse plus particulièrement, les valeurs d'intensités associées aux pixels sont appelées niveaux de gris. Le niveau de gris d'un pixel représente une mesure de l'accumulation de la lumière collectée dans la région du plan occupée par ce pixel [1].

### 2.4. L'image couleur :

Dans le contexte des images couleurs, la représentation fonctionnelle de l'image fait intervenir 3 fonctions d'intensités  $IR(u, v)$ ,  $IV(u, v)$ ,  $IB(u, v)$  relatives aux trois composantes primaires rouge, verte et bleue constitutives de la couleur [1].

### 2.5. La résolution d'une image :

La résolution, terme souvent confondu avec la « définition », détermine le nombre de points par unité de surface, exprimé en points par pouce (PPP, en anglais DPI pour Dots Per Inch). L'unité de longueur mondialement utilisée est le « pouce » (2,54 cm).

La résolution lie le nombre de pixels d'une image à ses dimensions réelles. Plus la résolution est élevée, plus les points nombreux, et plus l'image est fine.

#### Exemple :

Pour le premier cas on a une résolution de 3 dpi signifie donc 3 colonnes et 3 lignes de pixels sur un pouce carré ce qui donne donc 9 pixels sur un pouce carré.

Pour le deuxième cas on a une résolution de 8 dpi donc 64 pixels sur un pouce carré.

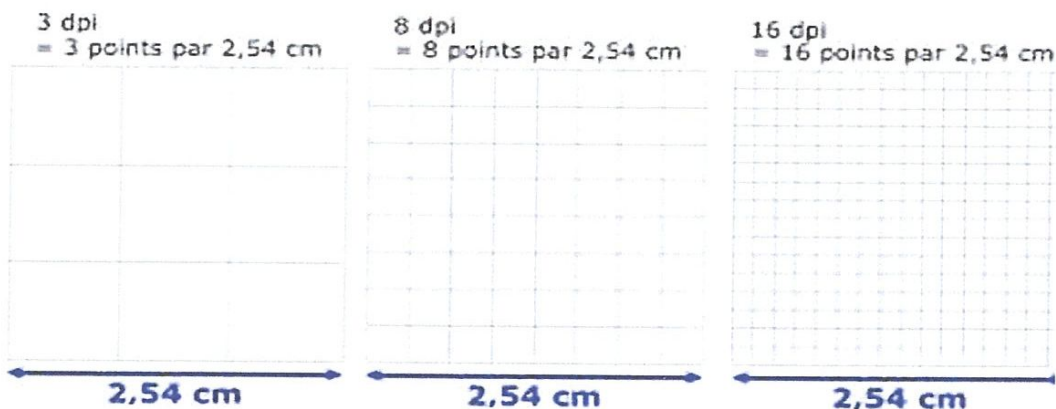


Figure (1.4) : Exemples de résolutions sur une même surface.

## 2.6. Le contraste :

Les capteurs ne perçoivent pas la couleur d'un même objet de la même manière. L'environnement à une forte influence sur cette perception. Il y a plusieurs types de contraste.

- ❖ **Contraste de luminosité:** c'est l'élément le plus perturbant pour la lecture des couleurs. Les capteurs s'adaptent à l'intensité moyenne d'une scène.
- ❖ **Contraste de saturation:** la vivacité d'une couleur est évaluée en fonction de la scène environnante. Une même couleur semble plus pâle si elle est entourée des couleurs soutenues.
- ❖ **Contraste de teinte:** la teinte est estimée également en fonction des teintes environnantes. Une même couleur apparaîtra froide si elle est entourée de couleurs chaudes.

## 2.7. Niveau de gris :

Dans une image en niveau de gris la couleur d'un pixel peut prendre des valeurs allant de 0 (Noir) jusqu'à 255 (blanc) en passant par un nombre fini de niveaux intermédiaires obtenus par dégradation du noir. Le pixel est ainsi codé sur un octet.

L'œil humain peut distinguer environ 40 niveaux de gris, c'est pour cette raison qu'il existe plusieurs opérations qui permettent la transformation des images en niveau de gris pour améliorer leur interprétation [3].

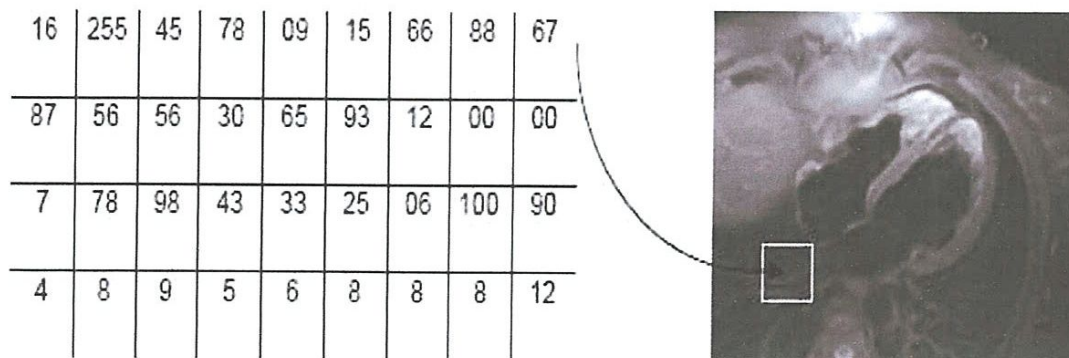


Figure (1.5) : Exemple d'image codée au niveau de gris.

## 2.8. Contours et textures :

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

## 2.9. Le bruit :

Le bruit est défini comme une variation arbitraire du niveau du gris ou comme de brusques variations de niveaux du gris. Le bruit provient soit de l'acquisition (caméra, scanner,..), soit de la scène elle même (poussière, condition atmosphériques..) [2].

Les dégradations subies par une image en termes d'effets et non plus de source peuvent être regroupé de la manière suivante :

- 👉 Dégradation ponctuelle (effet localisé en un pixel).
- 👉 Dégradation spatiale (effet sur un pixel et son voisin).
- 👉 Dégradation temporelle.
- 👉 Dégradation chromatique.

## 3. Conclusion:

La représentation des images fixes est l'un des éléments essentiels des applications multimédias, comme dans la plupart des systèmes de communication. La manipulation des images pose cependant des problèmes beaucoup plus complexes que celle du texte. En effet, l'image est un objet à deux dimensions, censé représenter un espace à trois dimensions, ce qui a deux conséquences majeures :

- 👉 Le volume des données à traiter est beaucoup plus important.
- 👉 La structure de ces données est nettement plus complexe.

Il en résulte que la manipulation, le stockage et la représentation de ces données se heurtent à certaines limitations.

Grâce au traitement d'images, ces contraintes sont levées ou contournées. En effet, ce domaine cherche à détecter la présence de certaines formes, certains contours ou certaines textures d'un modèle connu, c'est le problème de la détection. Un autre aspect de traitement d'image concerne l'analyse et la compréhension de l'image dans le but d'en extraire des informations utiles. Il cherche aussi parfois à compresser l'image, afin de

gagner en vitesse lors de la transmission de l'information, et en capacité de stockage, tout en dégradant le moins possible les images considérées.

# *Chapitre II*

*Etat de l'art sur*

*les différentes techniques de compression*

es

# Etat de l'art sur les techniques de compression

## 1. Introduction:

La compression de données est de réduire le nombre des bits utilisés pour stocker ou transmettre des informations. Il existe une grande variété de techniques de compression, logicielle et matérielles, les techniques de compression de données peuvent être divisées en deux grandes familles : conservatives et non conservatives. La deuxième méthode autorise une certaine perte de précision tandis que la compression conservative regroupe les techniques garanties de générer une copie exacte des données après un cycle compression/expansion.

Les méthodes de compression d'images visent à quantifier et à coder une information source (image) sur un nombre restreint de bits proche de l'entropie de cette source en exploitant la redondance qui peut exister dans une image. Cette redondance peut être spatiale ou spectrale (entre bandes de spectre, ou composante couleur), structurelle (entre formes ou régions d'une même image) ou temporelle (séquence vidéo). Les méthodes de compression exploitent ces différents types de redondance pour réduire le nombre de bits moyen nécessaire pour coder un symbole de la source.

## 2. Les techniques de compression de donnée :

### 2.1. Les techniques de compression sans pertes :

La compression est dite sans perte lorsqu'il n'y a aucune perte de données sur l'information d'origine. Il y a autant d'information après la compression qu'avant, elle est seulement réécrite d'une manière plus concise (c'est par exemple le cas de la compression gzip pour n'importe quel type de données ou du format PNG pour des images synthétiques destinées au Web2). La compression sans perte est dite aussi compactage. L'information à compresser est vue comme la sortie d'une source de symboles qui produit des textes finis selon certaines règles. Le but est de réduire la taille moyenne des textes obtenus après la compression tout en ayant la possibilité de retrouver exactement le message d'origine (on

trouve aussi la dénomination codage de source en opposition au codage de canal qui désigne le codage correcteur d'erreurs).

Il n'existe pas de technique de compression de données sans perte universelle, qui pourrait compresser n'importe quel fichier : si une technique sans perte compresse au moins un fichier, alors elle en « grossit » également au moins un.

### 2.1.1. Codage d'Huffman :

Le but du code de Huffman est d'affecter un codage plus court pour les symboles les plus fréquents.

Le principe du codage de Huffman repose sur la création d'un arbre composé de nœuds. Supposons que la phrase à coder est « Wikipédia ». On recherche tout d'abord le nombre d'occurrences de chaque caractère (ici les caractères 'a', 'd', 'é', 'k', 'p' et 'w' sont représentés chacun une fois et le caractère 'i' trois fois). Chaque caractère constitue une des feuilles de l'arbre à laquelle on associe un poids valant son nombre d'occurrences. Puis l'arbre est créé suivant un principe simple : on associe à chaque fois les deux nœuds de plus faibles poids pour donner un nœud dont le poids équivaut à la somme des poids de ses fils jusqu'à n'en avoir plus qu'un, la racine. On associe ensuite par exemple le code 0 à la branche de gauche et le code 1 à la branche de droite.

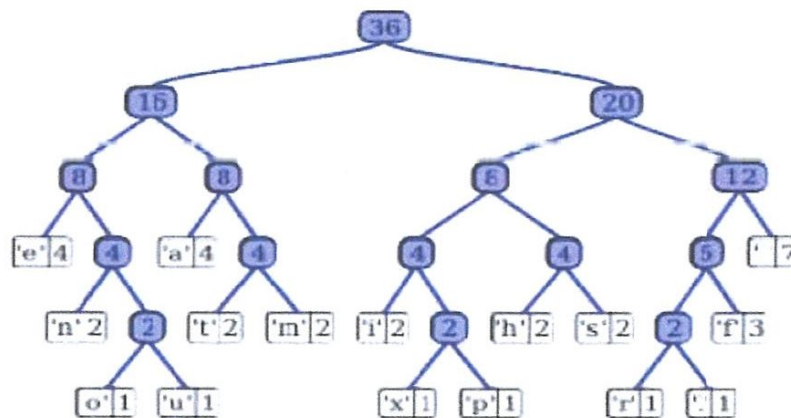
Pour obtenir le code binaire de chaque caractère, on remonte l'arbre à partir de la racine jusqu'aux feuilles en rajoutant à chaque fois au code un 0 ou un 1 selon la branche suivie. Il est en effet nécessaire de partir de la racine pour obtenir les codes binaires car lors de la décompression, partir des feuilles entraînerait une confusion lors du décodage. Ici, pour coder 'Wikipédia', nous obtenons donc en binaire : 101 11 011 11 100 010 001 11 000, soit 24 bits au lieu de 63 (9 caractères x 7 bits par caractère) en utilisant les codes ASCII (7 bits).

Il existe trois variantes de l'algorithme de Huffman, chacune d'elle définissant une méthode pour la création de l'arbre :

1. *Statique* : chaque octet a un code prédéfini par le logiciel. L'arbre n'a pas besoin d'être transmis, mais la compression ne peut s'effectuer que sur un seul type de fichier (ex: un texte en français, où les fréquences d'apparition du 'e' sont énormes ; celui-ci aura donc un code très court, rappelant l'alphabet morse).
2. *Semi-adaptatif* : le fichier est d'abord lu, de manière à calculer les occurrences de chaque octet, puis l'arbre est construit à partir des poids de chaque octet. Cet arbre restera

le même jusqu'à la fin de la compression. Il sera nécessaire pour la décompression de transmettre l'arbre.

3. *Adaptatif* : c'est la méthode qui offre a priori les meilleurs taux de compression car l'arbre est construit de manière dynamique au fur et à mesure de la compression du flux. Cette méthode représente cependant le gros désavantage de devoir modifier souvent l'arbre, ce qui implique un temps d'exécution plus long. Par contre la compression est toujours optimale et il n'est pas nécessaire que le fichier soit connu avant de compresser. En particulier, l'algorithme est capable de travailler sur des flux de données (streaming), car il n'est pas nécessaire de connaître les symboles à venir.



Figure

**2.1.2. Codage RLE (Run Length Encoding):**

Le RLE ou Run Length Encoding essay de tirer parti du fait que, souvent, le même symbole apparait plusieurs fois de manière consécutive dans le message. Par exemple symbole

AABBBBBBBBCCCC123

Est clairement un codage inefficace. On peut le remplacer par

2A8B4C123

qui est beaucoup plus compact. Dans ce cas, on a le problème de l'ambiguïté pour coder les chiffres, on ne saura pas s'il s'agit d'un symbole ou de l'indication du nombre de répétition du symbole qui suit. Pour éviter ce problème, on utilise un caractère d'échappement (@ par exemple).Ce caractère d'échappement signale qu'un *run* va suivre, la suite indiquant la longueur (length) de ce run et le symbole coder par le run.

AA@8B@4C123



Pour éviter de remplacer 2 caractères par 3, on ne déclenche le run-length que lorsque 3 caractères au moins sont consécutifs.

Il est intéressant d'utiliser le RLE juste après la quantification en choisissant un ordre de parcours des coefficients permettant de regrouper les valeurs consécutives semblables. C'est d'ailleurs ce qui est fait pour la compression JPEG.

### 2.1.3. Codage arithmétique :

Contrairement de l'algorithme de Huffman qui associe à des symboles des motifs binaires dont la taille dépend de leur distribution. Le codeur arithmétique traite le fichier dans son ensemble, en lui associant un unique nombre décimal rationnel.

Ce nombre compris entre 0 et 1, possède d'autant moins de chiffres après la virgule que le fichier dont il redondant. Ces chiffres décimaux dépendent non seulement des symboles du fichier dans l'ordre ou ils apparaissent, mais aussi leur distribution statistique.

#### Algorithme du codage arithmétique :

1. Calculer la probabilité associée à chaque symbole dans la chaîne à coder.
2. Associer à chaque symbole un sous intervalle proportionnel à sa probabilité, dans l'intervalle  $[0,1]$  (l'ordre de rangement des intervalles sera mémorisé car il est nécessaire au décodeur).
3. Initialiser la limite inférieure de l'intervalle de travail à la valeur 0 et la limite supérieure à la valeur 1.
4. Tant qu'il reste un symbole dans la chaîne à coder :
  - Largeur = limite supérieure – limite inférieure
  - Limite inférieure = limite inférieure + largeur \* (limite basse du sous intervalle du symbole)
  - Limite supérieure = limite inférieure + largeur \* (limite haute du sous intervalle du symbole)
5. La limite inférieure code la chaîne de manière unique.

Le plus simple est peut être d'expliquer le processus par l'exemple [8] :

Soit  $\{a, b\}$  l'alphabet utilisé par notre signal d'entrée  $(u, \dots, u)$ , et  $I$  l'intervalle  $[0,1]$ .

Notons  $p$  la probabilité d'apparition du symbole  $b$  est bien sur  $1-p$ .

A la lecture du  $K^{\text{ième}}$  symbole du signal, on subdivise l'intervalle courant  $I_K = [x_k \ y_k]$  en deux sous intervalles  $J_K = [X_K \ Z_K]$  et  $J'_K = [Z_K \ Y_K]$  où  $Z_K$  est le barycentre des bornes de  $I_K$  pondéré par  $p$ , et on le remplace par  $J_K$  si le symbole lu est un  $a$  et  $J'_K$  si c'est un  $b$ .

Une fois que le signal a été entièrement lu, on l'a donc codé par un couple de réels  $(X_N \ Y_N)$ , de manière unique puisque le décodeur (construit de manière semblable) reconstruit les intervalles  $I_N, I_{N-1}, \dots, I_0$ .

Les schémas des figures fig.3 et fig.4 explicitent sur un exemple le fonctionnement du codeur et du décodeur.

Intervalle dans lequel se trouve l'intervalle codé, puis on écrit le caractère correspondant et remplace courant par le sous-intervalle en question.

#### 2.1.4. Propriétés des techniques de compression sans perte :

Ces méthodes de compression sans pertes présentent en général des performances limitées. Pour respecter les contraintes posées par le spatial, il est nécessaire de recourir à des technique de compression avec pertes. Dans la plupart des cas, afin d'optimiser le débit au maximum, une compression sans pertes est utilisée à la suite de la compression avec pertes.

L'objectif sera de fournir au codeur sans pertes des données permettant de tirer parti au maximum de ses propriétés (concentration de l'énergie, regroupement des valeurs de même amplitude ensemble ...).

#### 2.2. Méthodes avec perte :

Ces méthodes permettent de retrouver une approximation de l'image numérique. Les pertes sont généralement indécélables à l'œil nu. Elles sont utilisées généralement pour la compression des données dont la qualité se limite aux perceptions humaines (images, vidéos, sons, ...).

##### 2.2.1. Quantification vectorielle et Quantification scalaire :

La quantification, dans son sens le plus général, consiste en l'approximation d'un signal d'amplitude continue par un signal d'amplitude discrète. Dans l'opposition qui est faite entre quantification scalaire (QS) et quantification vectorielle (QV), il convient d'entendre par entité « scalaire » de l'image, le pixel, et donc par « vecteur » de l'image, un vecteur de pixels (autrement dit un bloc de pixels).

### ➤ Quantification scalaire

Dans le cas de la quantification scalaire, on traite les échantillons de manière décorrélée les uns des autres. Chaque échantillon a une certaine amplitude, et il s'agit de définir des plages de quantification pour les coder. Puisqu'il y a des plages de quantification, il y a nécessairement une erreur de quantification qui provoque un bruit de quantification. Celui-ci dépend de la finesse des plages adoptées. On distingue plusieurs types de quantification scalaire ; la quantification uniforme et la quantification non uniforme.

### ➤ Quantification vectorielle

Pour améliorer encore les performances des codeurs, on ne peut plus se contenter de la quantification scalaire, qui ne traite les échantillons qu'indépendamment les uns des autres. Il convient d'utiliser la forte corrélation entre ces échantillons et de les regrouper afin de les quantifier ensemble. Il s'agira ensuite de régénérer des échantillons statistiquement proches ou similaires. On parle alors de quantification vectorielle, s'agissant littéralement de traiter des vecteurs scalaires.

La QV, développée par Gersho et Gray (1980) a pris une place très importante dans le domaine de la compression d'image que ce soit dans le but de transmission ou d'archivage.

### ➤ Principe de la quantification vectorielle

La quantification vectorielle dans son sens le plus général est l'approximation d'un signal d'amplitude discrète. Elle peut être vue comme une application  $Q$  associant à chaque vecteur d'entrée  $X$  de dimension  $K$  un vecteur  $Y=Q(X)$  de même dimension appartenant à un ensemble fini  $Y$  appelé Dictionnaire de taille finie  $N, Y=(Y_j, j=1 \dots N)$

Elle se décompose en deux applications : codeur, décodeur :

#### ➤ Codeur :

Le rôle du codeur consiste, pour tout vecteur  $X_i$  du signal en entrée à rechercher dans le dictionnaire  $Y$  le code vecteur  $Y_j$  le plus proche du vecteur source  $X$ . 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 [7].

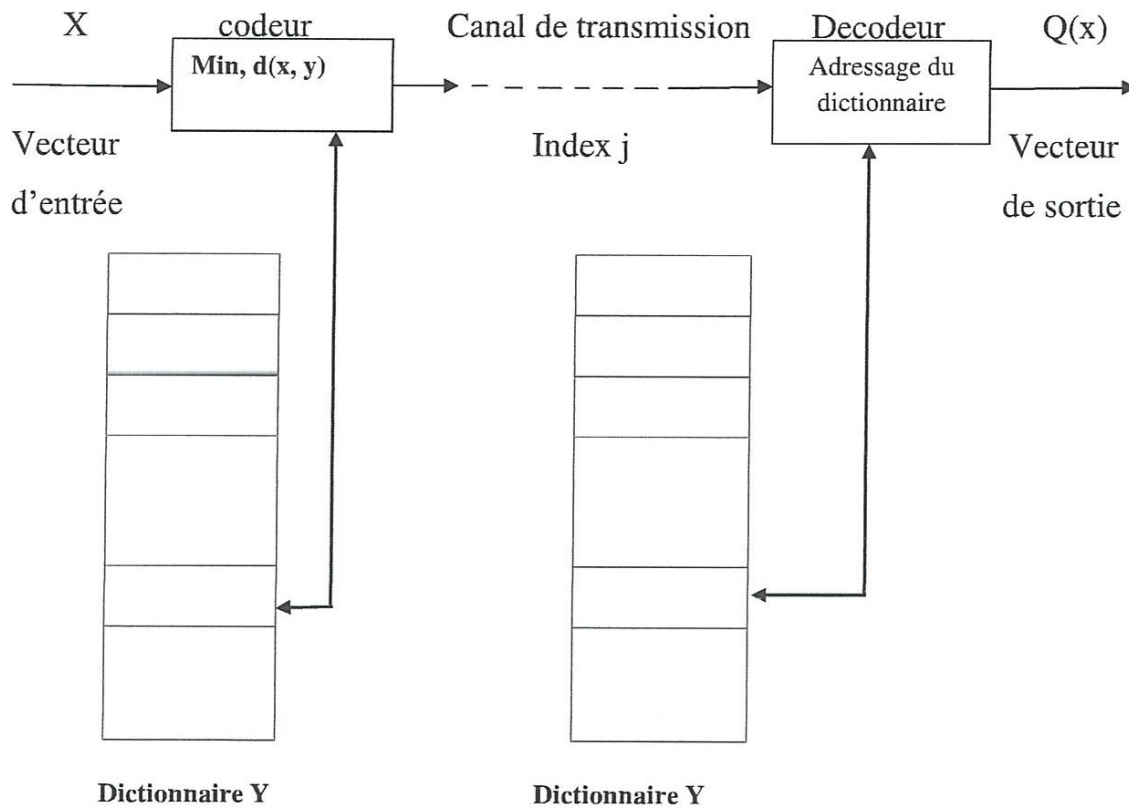


Figure 2.1 : Principe du quantificateur vectoriel.

### 3. principe de compression :

Les méthodes de compression d'images visent à quantifier et à coder une information source (image) sur un nombre restreint de bits proche de l'entropie de cette source en exploitant la redondance qui peut exister dans une image [14].

Cette redondance peut être spatiale ou spectrale (entre bandes de spectre, ou composante couleur), structurelle (entre formes ou régions d'une même image) ou temporelle (séquence vidéo).

Les méthodes de compression exploitent ces différents types de redondance pour réduire le nombre de bits moyen nécessaire pour coder un symbole de la source.

- ✓ La première étape consiste à détecter et à réduire la redondance spatiale, spectrale et temporelle de l'information source. Les pixels originaux sont analysés et transformés sous forme de coefficients décorrélés. Il existe de nombreuses approches qu'on peut les regrouper en deux catégories : Méthodes spatiales et

méthodes par transformées. Les méthodes spatiales manipulent directement les pixels (exp : DPCM, PCM, ...etc.) tandis que les méthodes par transformées conduisent à un changement de l'espace de représentation et manipulent ensuite les coefficients décorrélés obtenus via ce changement (exp : JPEG, JPEG2000, QMF,...etc.).

- ✓ La deuxième étape est la quantification de l'information décorrélée qui consiste à réduire le nombre des symboles nécessaires et suffisant pour obtenir une représentation fidèle de l'information originale. Cette étape introduit une perte d'information donc une dégradation de restitution. C'est ce qui fait la nature irréversible d'un tel processus de compression.
- ✓ Dans l'étape de codage, les symboles issus de l'étape de quantification sont codés par une technique similaire à celle utilisée pour la compression non destructif comme le codage entropique de type Huffman.

#### 4. Codage Direct (Codage spatial) :

Dans ce type de compression on agit directement sur les échantillons (pixels) de l'image. Pour cela on s'appuie sur la force de corrélation existante entre les pixels voisins (redondance spatiale). La redondance spatiale part du principe que les pixels voisins d'une image ont des valeurs de luminance et de chrominance très proches. En fonction des techniques utilisées, le codage direct peut être déterministe (MIC, MICD), ou bien non déterministe (Codage prédictif).

##### 4.1. Le codage MIC (Modulation par Impulsion Codée) :

Dans le codage MIC, encore appelé PCM, le code binaire correspondant à chaque niveau d'un signal  $X(t)$  est transmis vers le récepteur chargé de restituer ces niveaux (étape de décodage).

Soit une image de taille  $N \times M$ , quantifiée sur  $2^k$  niveaux, le nombre de bits  $B$  à transmettre est donnée par :

$$B = N * M * K \quad (2.1)$$

Pour réduire le nombre de bits transmis on procède à un codage entropique, qui consiste à transmettre des codes de longueurs variables entre l'émetteur et le récepteur [4][5][6].

## 4.2. Le codage MICD (Modulation par Impulsion Codée Différentielle) :

Le codage MICD (DPCM) utilise le même principe de quantification que le codage MIC. Cependant, il est appliqué sur un signal différence obtenu à partir d'opérations de soustraction entre chaque pair d'échantillons voisins. En effet, compte tenu de la redondance, souvent importante, entre les échantillons voisins on est sûr qu'une différence entre ces échantillons conduira à des échantillons résultats possédant des amplitudes relativement faibles. Ainsi, le code binaire affecté à ces échantillons résultats comportera moins de bits d'où l'idée de cette méthode de compression.

La figure suivante présente le schéma synoptique d'un système de transmission, appliqué sur un signal unidimensionnel, basé sur la méthode MICD :

$$d(n) = x(n) - x(n-1)$$

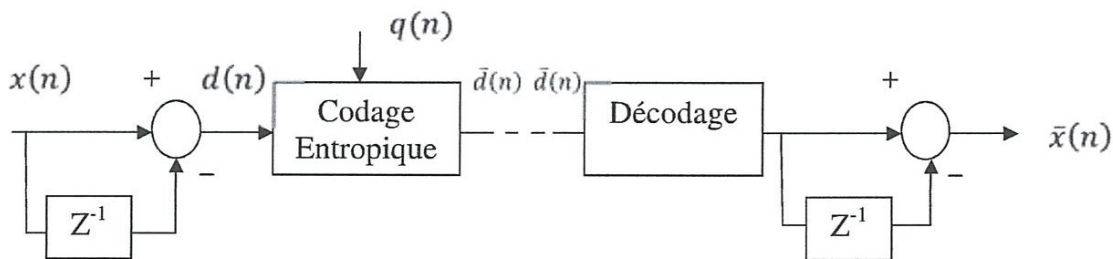


Figure 2.2 : principe d'un CODEC de type DPCM.

Dans la pratique, cette technique donne des taux de compression relativement faibles essentiellement dans le cas des images.

## 4.3. Le codage prédictif :

Le codage prédictif est une méthode de compression non déterministe. Elle s'inspire du codage DPCM. Seulement la quantification est effectuée sur la différence entre l'échantillon présent  $x(i)$  et sa prédiction  $x_e(i)$ .

$$d(i) = x(i) - x_e(i)$$

$d(i)$  : L'erreur de prédiction.

$x_e(i)$  est la valeur estimée de  $x(i)$  obtenue par un prédicteur linéaire d'ordre  $n$  (exemple du type RIF à 04 coefficients (figure 2.2)).

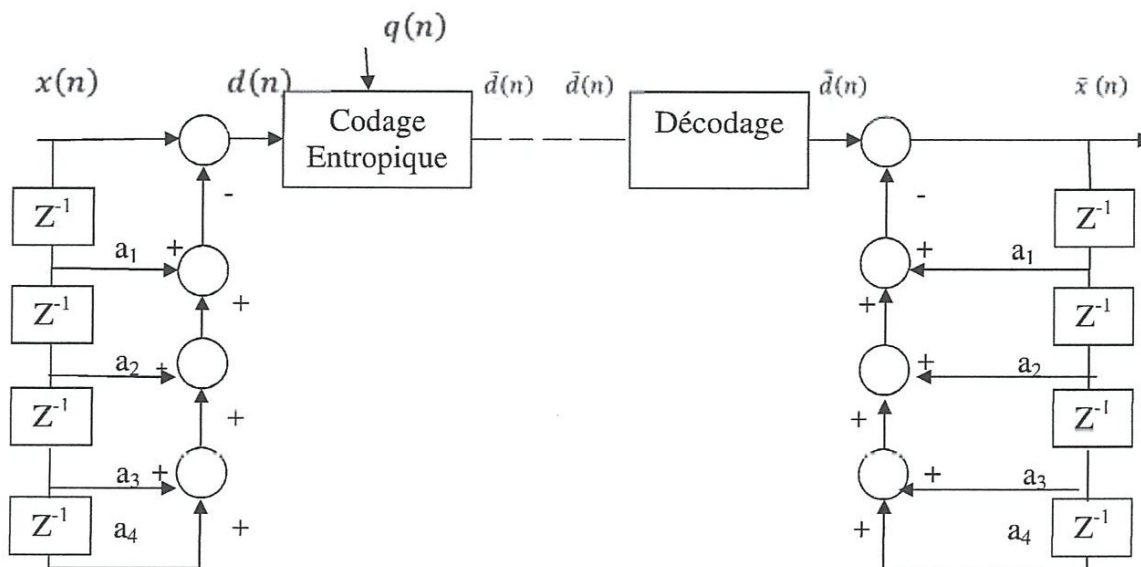


Figure 2.3 : Principe du codage prédictif à l'aide d'un filtre RIF à 4 coefficients.

Cette méthode permet d'avoir des rapports de compression jusqu'à 4 pour des coefficients de prédiction bien choisis, avec une qualité du signal restauré acceptable [4].

## 5. Le codage par transformée :

### 5.1. La compression par DCT (Discret cosine transform) :

La DCT est la transformation la plus utilisée pour la compression des données. Elle assure une concentration de la quasi-totalité de l'énergie dans la gamme des basses fréquences.

Cette transformation orthogonale est très proche de celle de **Karkhunen-loeve** (KLT).

Cependant, son principal avantage est la simplicité de sa mise en œuvre ainsi que l'existence d'un algorithme rapide permettant l'obtention de ses coefficients (dédit à partir de l'algorithme FFT).

La compression par DCT s'effectue en deux temps : la transformation suivit par la quantification (le codage). La figure suivante expose le principe de fonctionnement d'un système de transmission à codage par DCT.

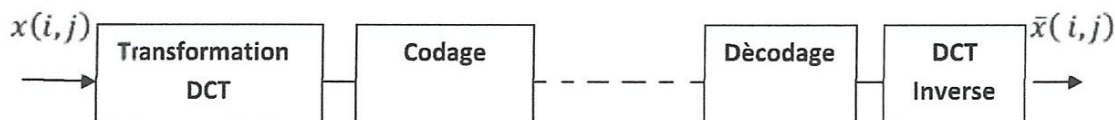
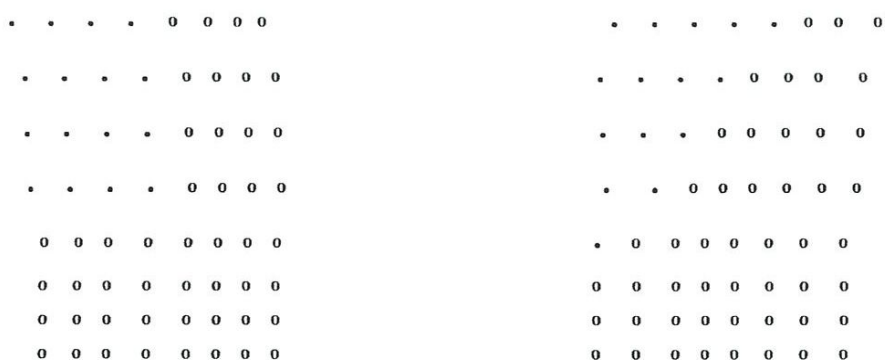


Figure 2.4 : Schéma synoptique d'un CODEC à base de DCT.

La DCT est une transformation conservatrice donnant à la sortie une matrice de coefficients représentant l'énergie totale répartie sur les différentes fréquences du signal source [13]. La quantification est l'étape non conservative de la méthode .Elle consiste soit à réaliser une opération de scuillage sur les coefficients de la DCT pour ne garder ceux qui sont représentatifs (par exemple les coefficients hauts fréquences) d'où l'idée de la compression .En effet, la propriété essentielle de la DCT est la concentration des coefficients dans les basses fréquences. Dans ce sens plusieurs méthodes de quantification ont été proposes : - La norme JPEG, qui consiste à réduire de plus en plus la précision des coefficients haute fréquences. La troncature des blocs (Block Troncation Coding), pouvant être effectue de deux manières différentes :

- Troncature par la méthode de seuil (**Threshold Coding**) : Seul les coefficients dont l'amplitude est plus importance est plus que ce seuil seront conservés.
- Troncature par zone (**Zonal Coding**) : Cette technique consiste à utiliser un masque qui couvre la région d'intérêt et élimine les coefficients restants. La figure suivante donne deux types de masque :



a: masque rectangulaire  
de taille 4\*4.

b : masque diagonal  
à 5 lignes.

Figure 2.5 : Principe de la troncature par zone.



## 5.2. JPEG (Joint Photographic Experts Group):

JPEG est l'acronyme de Joint Photographic Experts Group. Il s'agit d'un comité d'experts qui édite des normes de compression pour l'image fixe.

Elle est destinée à la compression des images couleur et à niveau de gris en vue de leur stockage sur les supports numériques et leur transmission à travers un réseau à faible débit.

Les techniques définies par la norme JPEG se divisent en deux classes : les méthodes de compression sans pertes utilisant le codage MICD suivi d'un codage entropique. La seconde classe, concerne les méthodes de compression avec pertes, basé sur la transformation DCT suivie d'une quantification et d'un codage entropique. Evidemment, l'image compressée sera de qualité moindre que l'originale. Cependant, il est possible de régler la qualité en fonction du choix de l'utilisateur. On notera que le rapport moyen de compression entre le JPEG et le GIF est de 1 pour 4. L'algorithme de compression JPEG se compose en trois parties :

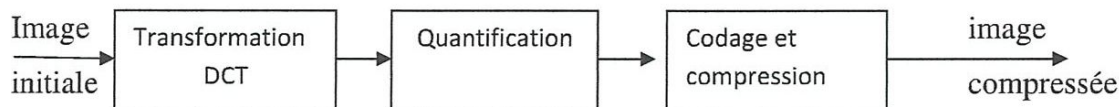


Figure 2.6 : Principe de codage utilisé par le groupe JPEG.

L'image est avant tout scindée en blocs de taille 8\*8, pour éviter le nombre très élevé des opérations si on prenait la DCT de l'image tout entière. En effet, le calcul de chaque élément de la matrice DCT dépend de la matrice utilisée. Les blocs 8\*8 permettent de calculer la DCT plus rapidement et avec un minimum de pertes.

## 5.3. EZW (Embedded Zerotrees Wavelet) :

L'algorithme d'EZW tire fortement profit de la structure de la décomposition en ondelette afin de réaliser une compression efficace de l'image. La compression provient principalement de deux processus. Le premier consiste en l'encodage itératif des coefficients dont les valeurs sont fixées par rapport au seuil lors de l'itération considérée. Le second processus de compression réside dans l'utilisation d'arbre de zéro. Lors qu'une racine d'arbre est identifiée, tous les coefficients descendants ne sont plus encodés.

#### 5.4. JPEG2000 (Joint Photographic Experts Group):

Le travail sur la norme JPEG 2000 a commencé peu après la publication de la norme JPEG. Ce travail a été motivé principalement pour apporter de souplesse ainsi que des taux de compression plus importants. Il y a notamment moins de limitations concernant la taille des images, le nombre de bandes.

JPEG2000 est le standard le plus récent de compression d'image. Il utilise une transformée en ondelettes et effectue un codage optimisé de façon à satisfaire principalement deux objectifs :

- Produire un codage progressif. Ceci signifie que l'on compresse l'image une seule fois et que l'on peut la décompresser plus ou le moins selon la qualité que l'on souhaite obtenir

Réduire la redondance qui existe dans les coefficients d'ondelettes. Pour améliorer la distorsion par rapport à un simple codage entropique des coefficients quantifiés, JPEG2000 utilise un codeur adaptatif [7].

# *Chapitre III*

## *L'analyse multi-résolution (Ondelettes)*

## L'analyse multi-résolution (Ondelettes)

### 1. Introduction:

En 1975 le géophysicien *Jean Morlet* a créé des petites fonctions mathématiques ayant certaines propriétés intéressantes, ceci afin de sonder les différentes couches géologique et détecter des nappes de pétroles. Ces fonctions sont les ondelettes, la terminologie du mot « ondelette » vient de mot « onde ». Donc les ondelettes sont, avant tout, un puissant outil d'analyse mathématique. La véritable révolution due aux ondelettes s'est opérée dans le très vaste domaine du traitement du signal.

En 1980 l'analyse par ondelettes a été officiellement introduite dans le domaine du traitement de signal (l'analyse des signaux sismiques), Il s'agissait de donner une représentation des signaux permettant de faire apparaître simultanément des informations temporelles (localisation dans le temps, durée) et fréquentielles, facilitant par là l'identification des caractéristiques physiques de la source du signal.

L'analyse par ondelettes a été formalisée plus tard par Grossmann et Morlet (1984) et Goupillaud. C'est au sein de ces dernières communautés que des développements théoriques et <sup>applicatifs</sup> appliqués majeurs ont eu lieu ces dernières années. Des avancées significatives ont notamment été faites par Meyer, Mallat, Daubechies, Chui, Wornell et Holschneider. Ces avancées ont alors influencé d'autres domaines de recherche. Ces interactions entre développement et application favorisent encore aujourd'hui l'évolution rapide de l'outil "ondelettes" [3].

Aujourd'hui l'analyse par ondelette remplace ou complète l'analyse de Fourier dans la plupart des domaines, une des premières causes de ceci, c'est que l'analyse par ondelette à bien éliminer les limitations de l'analyse de Fourier. La transformée en ondelettes fournit une représentation d'un signal dans le plan temps-fréquence, ce qui n'est pas le cas pour la transformé de Fourier.

Dans ce chapitre on va présenter brièvement la théorie d'ondelette sans entrer beaucoup dans les détails, d'abord on va parler sur la transformé de Fourier, son utilité et ces limitations. Ensuite on parle sur la transformé en ondelette (continue et discrète), on explique comment cette transformation peut compléter ou remplacer la transformé de Fourier. Un autre outil, la multi-résolution qui lorsqu'on l'utilise avec les ondelettes elle donne de bons résultats. Enfin on cite quelques exemples d'ondelettes les plus utilisées et des implémentations les plus connues pour la décomposition.

## 2. Transformé de Fourier :

La Transformée de Fourier (TF) est une transformation mathématique qui permet de décomposer un signal quelconque dans une somme des sinusoides en multipliant chacune par un coefficient. (Cette somme est infinie et contiendra toutes les fréquences si notre signal étudié n'est pas périodique). La transformation de Fourier d'une fonction en temps sera une fonction en fréquence qui à chaque fréquence associe un coefficient.

La transformé de Fourier est définie comme suit :

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t} dt \quad (3.1)$$

La fonction  $\hat{f}$  peut être vue comme le spectre en fréquences de la fonction  $f$  (fonction étudié).

Pour la transformé inverse on a :

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega)e^{j\omega t} d\omega \quad (3.2)$$

L'information qui nous intéresse dans un signal quelconque est souvent cachée dans les fréquences, c'est à dire qu'on veut souvent savoir les fréquences principales qui constituent le signal. (Ceci est très difficile, voir impossible à partir de la représentation temporelle) La Transformée de Fourier nous fournit exactement ça. Si on regarde le spectre du signal, les fréquences qui correspondent aux coefficients maximaux sont les fréquences principales.

Les transformé de Fourier est, dans certains cas, limitées.

- La TF a du mal à représenter les discontinuités. Le Phénomène de Gibbs est un bon exemple (Ceci vient du fait que les fonctions sinus et cosinus sont des fonctions

continues et infinies donc leur somme doit être une fonction continue infinie ce qui n'est pas le cas des fonctions réelles qu'on essaie d'étudier).

- Le deuxième c'est que la Transformée de Fourier d'écrit parfaitement le domaine fréquentiel par contre on n'a aucune information sur le domaine temporel. C'est à dire que si notre signal ne change pas au cours de temps, la TF est un outil parfait par contre si notre signal varie au cours de temps, on sera incapable de déterminer le lieu temporel de chaque composant. Donc, elle n'est pas applicable pour les signaux non stationnaires.

L'une des solutions de ces défauts est la TF à fenêtre. La TF à fenêtre est une méthode qui essaie de fournir des informations à la fois sur le domaine temporelle et fréquentielle. On décompose le signal en plusieurs parties qui peuvent être considérées comme stationnaires. Ensuite on applique la TF à chacun des morceaux. Avec cette solution Il y a par contre un problème de résolution. On ne peut pas savoir exactement les composants fréquentielles qui existent à chaque instant. Ceci est dû au principe de « Heisenberg » appliqué au traitement de signal. « Si mes morceaux ou ma fenêtre est très petit j'ai une bonne résolution temporelle mais j'ai une résolution faible en fréquence et inversement si ma fenêtre est très large je gagne de la résolution fréquentielle mais je perds en temporel » (Si on utilise une fenêtre étroite on aura une mauvaise résolution surtout pour les fréquences dont la période est trop grande pour rentrer dans la fenêtre qu'on utilise). Donc la question est quelle largeur de fenêtre à utiliser. Bien sur la meilleure fenêtre est toujours celle qui est le mieux adapté au signal et à ce qu'on aimera savoir.

### 3. Transformée en ondelette :

Pour comprendre la Transformée en Ondelettes il est nécessaire de connaître sa relation avec la Transformée de Fourier qui se résume dans ces deux remarques:

- a. La Transformée en Ondelettes a été développé à cause des limitations de la Transformée de Fourier.
- b. La Transformée en Ondelettes et la Transformée de Fourier ont des propriétés en commun.

Donc la transformé en ondelette à été développé à partir des propriétés de TF et en éliminant les limitations qu'on a cité auparavant.

### 3.4. Transformée en ondelette discrète :

La transformée en ondelettes discrète (DWT) est une implémentation de la transformée en ondelettes continue. En utilisant un ensemble discret des échelles d'ondelettes, en d'autres termes cette transformée décompose le signal en système mutuellement orthogonaux d'ondelettes, dans la pratique, on restreint les paramètres de dilatation-translation  $(a, b)$  à un sous ensemble discret.

On choisit :

$$a_m = a^{-m}, b_n = n \cdot b \cdot a^{-m}, (a > 1, b \neq 0)$$

Avec :

$$\Psi_{a_n, b_n} = a^{m/2} \Psi(a^m x - nb) \quad (3.9)$$

Compte tenu de la redondance d'informations données par les coefficients  $Cf$  sur la «base» des  $\psi_{a_n, b_n}$ , Y. Meyer a montré qu'on peut choisir l'ondelette  $\Psi \in \mathcal{L}_2(\mathbb{R})$ , régulière, telle que  $(\Psi_j)$  soit une base orthonormée de  $\mathcal{L}_2(\mathbb{R})$ .

Ces ondelettes particulières sont appelées ondelettes orthogonales sur lesquelles on peut décomposer tout signal  $f \in \mathcal{L}_2(\mathbb{R})$  :

$$f(x) = \sum_{j,k} \langle f, \Psi_{j,k} \rangle \Psi_{j,k}(x) \quad (3.10)$$

$$\text{Avec : } \Psi_{j,k} = 2^{-j/2} \Psi(2^j x - k)$$

Ces coefficients sont indépendants les uns des autres et on les obtient comme suite :

$$C_{j,k} = 2^{-j/2} \int f(x) \Psi(2^j x - k) dx \quad (3.11)$$

Le réseau des paramètres de dilatation-translation considéré dans ce cas est la collection de tous les intervalles dynamiques.

$$I(j, k) = [k2^{-j}, (k+1)2^{-j}] \quad j, k \in \mathbb{Z}^2 \quad (3.12)$$

Où les paramètres de dilatation-translation sont :  $(a, b) = (2^j, 2^{-j}k)$

### 3.5. Analyse multi-résolution :

Le concept de la multi-résolution est d'utiliser plusieurs fois la transformation en ondelette afin de décomposer le signal de départ en plusieurs signaux contenant des informations différentes. D'une manière très simple, à la première échelle de la DWT va extraire les détails des plus fins du signal. À la deuxième échelle, on aura les détails un peu plus grossiers et ainsi de suite, jusqu'à obtenir un signal complètement lisse [10].

La notion de multi-résolution veut dire qu'on utilise différentes résolutions pour étudier différentes composantes spectrales du signal. Les signaux réels qu'on essaie d'étudier ont souvent les caractéristiques suivantes :

Les composants de hautes fréquences sont présents localement dans le signal comme des petites variations, et les composants de basses fréquences sont présents tout au long, en traçant l'allure globale du signal. Il semble logique donc de choisir une bonne résolution temporelle (qui impose une mauvaise résolution fréquentielle) pour étudier les hautes fréquences car ils apparaissent localement, et une bonne résolution fréquentielle (qui inversement impose une mauvaise résolution temporelle) pour étudier les basses fréquences qui sont longtemps présents.

Contrairement à la Transformée de Fourier à fenêtre, où la résolution reste constante au cours de la transformation, avec la transformée en Ondelettes on change les échelles de l'ondelette mère, donc la résolution varie. Les hautes fréquences (petites variations) sont captées par des ondelettes étroites, et les basses fréquences (l'allure globale) par les ondelettes larges.

Y.Meyer et S.Mallat ont introduit l'analyse multi résolution de  $\mathcal{L}^2(\mathbb{R})$  et s'est révélée particulièrement féconde pour la construction des bases orthonormées d'ondelettes, ainsi que le fondement mathématiques pour approcher une fonction « $f$ » par ses approximations successives  $f_j$ ,  $j \in \mathbb{Z}$  contenant de plus en plus d'informations dans des espaces emboîtés. L'idée de bases est alors de mesurer les changements entre l'approximation  $f_j$  de « $f$ » à l'échelle  $2^{-j}$  et l'approximation  $f_{j+1}$  de  $f$  à l'échelle  $2^{-(j+1)}$ .

L'analyse multi-résolution de  $\mathcal{L}^2(\mathbb{R})$  est la donnée d'une suite croissante  $V_j$ , de sous espaces fermés de  $\mathcal{L}^2(\mathbb{R})$  ayant les propriétés suivantes :

$$\downarrow \forall j \in \mathbb{Z}, f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$$

$\downarrow V_0$  est invariant par translations entières de la variable.

$$f \in V_0 \Leftrightarrow \forall K \in \mathbb{Z} \quad f(x - K) \in V_0$$

$$\downarrow \bigcup_{j \geq 1} V_j = V_0 \quad \text{et} \quad \bigcap_{j \geq 1} V_j = \{0\}.$$

$\downarrow$  il existe une fonction  $g$  dans  $V_0$  telle que  $\{(x - k)\}$  représente une base de  $V_0$ .

L'espace de toutes les approximations possibles est interprété par  $V_j$  à l'échelle  $2^{-j}$ .



- La condition  $V_j \subset V_{j+1}$  traduit le fait que toute information accessible à l'échelle  $2^{-j}$  l'est aussi à l'échelle  $2^{-(j+1)}$  : l'information contenue dans le signal est dégradée lorsque  $j$  décroît.
- La condition (1) est la propriété d'invariance par dilatation.
- La condition (3) précise que les  $V_j$  correspondent aux différents espaces d'approximation.

D'après ces quatre conditions, on montre que l'analyse multi-résolution d'un signal à la résolution  $2^j$  est la projection orthogonale sur l'espace  $V_j$  et pour que cette analyse puisse être effectuée, il faut trouver une base orthogonale pour l'espace  $V_j$ .

### 3.6. Quelques ondelettes :

#### a. Ondelettes de Haar :

C'est une fonction classique orthogonale sa fonction d'échelle  $\varphi(x)$  [10], ainsi que son ondelette  $\Psi(x)$  ont la forme suivante respectivement :

$$\varphi_{j,K}(x) = \begin{cases} 2^{-j/2} & \text{pour } 2^j(K-1) \leq x < 2^j K \\ 0 & \text{ailleurs} \end{cases} \quad (3.13)$$

$$\Psi_{j,K}(x) = \begin{cases} 2^{-j/2} & \text{pour } 2^j(K-1) \leq x < 2^j(K-1/2) \\ 2^{-j/2} & \text{pour } 2^j(K-1/2) \leq x < 2^j(K+1) \\ 0 & \text{ailleurs} \end{cases} \quad (3.14)$$

#### b. Ondelettes de Daubechies :

De nombreuses recherches ont été faites dans le domaine de la synthèse des filtres. Daubechies a proposé en 1988 une autre forme d'ondelettes orthogonales à support compact tout comme les ondelettes de Haar (remarque que daubechies1 n'est que l'ondelette de Haar). L'ondelette de Daubechies possède  $N$  moments nuls, ce qui fait que sa régularité augmente avec  $N$  [11]. Cette ondelette est notée  $D_n$ , où  $n$  est le nombre de moments nuls de l'ondelette  $\Psi$ .

$$\varphi(x) = \sqrt{2} * \sum_{K=0}^{2N-1} h_{K+1} * \varphi(2X - K) \quad (3.15)$$

$$\Psi(x) = \sqrt{2} * \sum_{K=0}^{2N-1} g_{K+1} * \varphi(2X - K) \quad (3.16)$$

Où  $g_K = (-1)^K h_{K+1}$  pour  $K=1,2,\dots,2N$ .

### 3.7. Les Ondelettes en imagerie :

L'idée est simple, comme tout filtre appliqué à une image, il suffit de multiplier les coefficients de l'ondelette (après l'avoir discrétisée) par la valeur des pixels de l'image à traiter. On notera que l'ondelette peut alors être représentée sous la forme d'un tableau (de dimension 1) de valeurs décimales. En réalité il y a 2 tableaux qui représentent l'ondelette car une ondelette se compose de 2 filtres : un passe-bas et un passe-haut [12].

### 4. Conclusion :

Les ondelettes sont l'un des outils les plus efficaces actuellement dans le domaine de traitement des signaux. La transformée en ondelettes permet de visualiser de façon particulièrement intéressante les différentes fréquences intervenant dans un signal ainsi que leurs instants d'apparition. Ses applications devraient donc s'élargir progressivement et toucher de nombreux domaines, signaux biomédicaux (électrocardiogrammes, ...etc.), sismographie, traitement d'images, . . . etc.

La détection et la segmentation basées sur les décompositions en ondelettes apparaissent comme des techniques très performantes. Elles sont suffisamment robustes pour s'adapter à de nombreux type de données et permettent une extraction des structures en fonction de leur morphologie. La transformé en ondelettes nous permet d'isoler les détails d'un signal, donc de supprimer n'importe qu'elle détail jugé comme un bruit, ou non nécessaire pour le problème traité.

# *Chapitre IV*

## *La compression d'images par Embedded Zerotree Wavelet (EZW)*

## La compression d'images par Embaded Zerotree Wavelet (EZW)

### 1. Introduction:

La transformée en ondelette permet, comme décrit dans le chapitre précédent, de représenter les images sous forme de coefficients ordonnés en bandes de fréquences. Pour la compression d'images, la transformée représente le premier maillon de la chaîne, afin de compresser l'information, il faut compléter le cycle par la quantification et le codage. Afin de garder le côté progressif de la transformée en ondelettes, une méthode d'organisation et de quantification des coefficients s'impose pour la méthode de compression par EZW.

Cette méthode de compression (EZW) a été développée par Jérôme M. Shapiro. Cet algorithme exploite les redondances dans les différentes sous-bandes afin de compresser au maximum l'image. Il utilise en plus une transformée en ondelettes de préférence complète (au contraire d'EBICOT qui se base sur une transformée en ondelettes incomplète comme on le verra). Pour cela, elle s'appuie sur une structure de données particulièrement adaptée à la transformée en ondelettes : les quadtree.

Dans ce chapitre nous allons présenter quelques notions fondamentales concernant les concepts et les caractéristiques d'EZW.

### 2. Les différentes méthodes de compression :

#### 2.1. JPEG 2000(Joint Photographic Experts Group):

JPEG 2000 est donc un nouveau système de codage d'images qui utilise les connaissances actuelles dans le domaine de la technologie des ondelettes.

Précisons, au cœur de JPEG il y a la transformée de Fourier discrète. Au cœur de JPEG 2000, il y a une décomposition en ondelettes. Le principe même du codage est différent. JPEG 2000 n'est pas une amélioration de JPEG, il constitue une autre manière d'analyser, de décomposer l'image pour la compresser avec ou sans pertes en apportant des fonctionnalités complètement nouvelles, en particulier pour les transmissions à faible bande passante.

L'algorithme utilisé par JPEG 2000, en fait une succession d'algorithmes, peut se décomposer, en première analyse, comme le montre le schéma suivant :

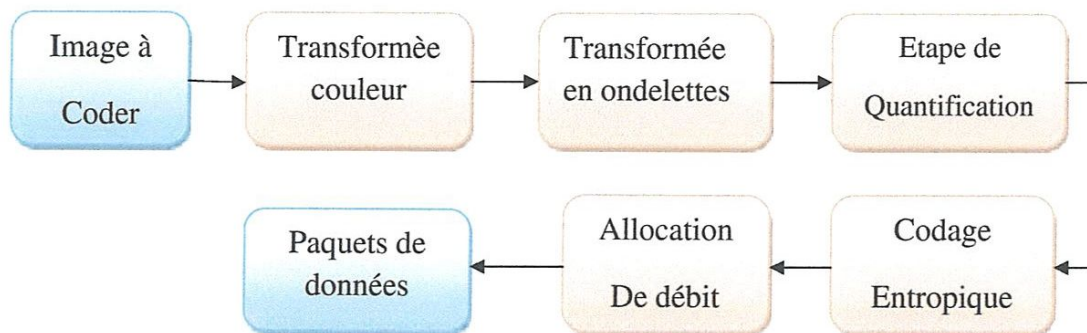


Figure (4.1) : Les étapes de l'algorithme de JPEG2000

## 2.2. EBICOT (Embedded Block Coding With Optimized Truncation):

L'algorithme EBICOT, bien qu'il est basé sur une transformée en ondelettes comme l'algorithme EZW. Cet algorithme est considéré comme un des plus performants algorithmes de compression d'image aussi bien en termes de compression pure qu'en termes de fonctionnalités. C'est pourquoi, que cet algorithme été choisi pour devenir le cœur du format JPEG2000. Il a été conçu par David Taubman, un chercheur australien.

L'algorithme EBICOT commence, comme EZW par une décomposition en ondelettes, mais cette fois, celle-ci doit être incomplète. Ensuite, on effectue une quantification des coefficients obtenus sachant que le taux de quantification dépend de la sous-bande. Ensuite, on divise l'image ainsi obtenu en plusieurs code-blocs (codeblock en anglais). Le code-bloc est le concept central d'EBICOT (un peu comme le quad-tree pour EZW). Un code-bloc est, en fait, une sous-matrice carré de la matrice image et qui est de taille 32x32 ou 64x64. De plus, un code-bloc doit faire partie d'une sous-bande, donc, un code-bloc étant de taille fixe(32x32 ou 64x64), si la transformée en ondelette serait complète, alors  $LL_n$  serait réduit à un simple point, et donc un code-bloc contiendrait plusieurs sous-bande, alors que justement, un code-bloc doit faire parti d'une sous-bande. On comprend alors pourquoi la transformation en ondelette ne peut être complète, le niveau  $m$  maximum de la transformation est celui pour lequel la sous-bande  $LL_m$  a exactement la même taille qu'un code-bloc. Une fois la division en sous-blocs effectué, on entre dans le cœur de l'algorithme.

### 2.3. EWZ (Embedded Zerotrees Wavelet):

L'algorithme, est basé sur une structure de donnée nommée quadtree. Mais qu'est-ce qu'un quadtree ? Comme son nom l'indique, c'est un arbre dont les nœuds ont 4 fils. Mais en quoi cette structure de données est utile pour cet algorithme en particulier et pour la compression d'image basée sur une transformée en ondelettes en général ? La réponse est simple : si on se souvient de la manière dont on obtient une sous-bande, on remarque qu'une sous-bande  $LH_k$  (ou  $HL_k$  ou  $HH_k$ ) a quatre fois plus de points qu'une sous-bande  $LH_{k+1}$  (ou  $HL_{k+1}$  ou  $HH_{k+1}$ ). Vu que chaque sous-bande est une "transformée" de l'image complète, à quatre points d'une sous-bandes  $LH_k$  correspond un et un seul point d'une sous-bande  $LH_{k+1}$  : en effet un point d'une bande  $LH_{k+1}$  représente la même zone qu'un carré de 4 points de  $LH_k$  et même qu'un carré de 16 points d'une sous-bande  $LH_{k-1}$ . Ainsi, on va organiser les points de la matrice comme un arbre : chaque point d'une certaine bande  $LH_k$  (respectivement  $HL_k$  ou  $HH_k$ ), pour  $k \neq 1$ , aura 4 fils : les quatres points qui correspondent à la même zone de l'image dans la sous-bande  $LH_{k+1}$ , (respectivement  $HL_{k+1}$ , ou  $HH_{k+1}$ ). Les feuilles de l'arbre seront donc les points des sous-bandes  $LH_1$ ,  $HL_1$ , et  $HH_1$  (car il n'y a pas de sous-bande  $LH_0$ ,  $HL_0$  et  $HH_0$ ). En réalité seul les points dans la sous-bande  $LL_n$  (où  $n$  est le niveau de la décomposition) n'ont pas quatres fils, mais d'une part,  $LL_n$  n'est réduit qu'à un unique point car la transformation est de préférence complète, et d'autre part, il suffit de considérer que c'est le seul point qui n'aura que 3 fils qui seront les points qui constituent sous-bandes  $LH_n$ ,  $HL_n$  et  $HH_n$ . Ce point  $LL_n$  devient donc la racine de l'arbre tout en étant le seul nœud n'ayant que trois fils. Même si la transformation n'est pas complète, on peut tout de même utiliser l'algorithme : il suffit de représenter la matrice image non pas en un arbre mais en plusieurs arbres : un arbre pour chaque points de  $LL_n$ , mais je ne m'étendrai pas sur ce cas, car il complique inutilement la compréhension de l'algorithme EZW. On a donc créé un arbre dont tous les nœuds (excepté la racine) ont 4 fils : c'est cela un quadtree. Le schéma suivant est extrait de l'article de Shapiro et résume très bien la structure de donnée :

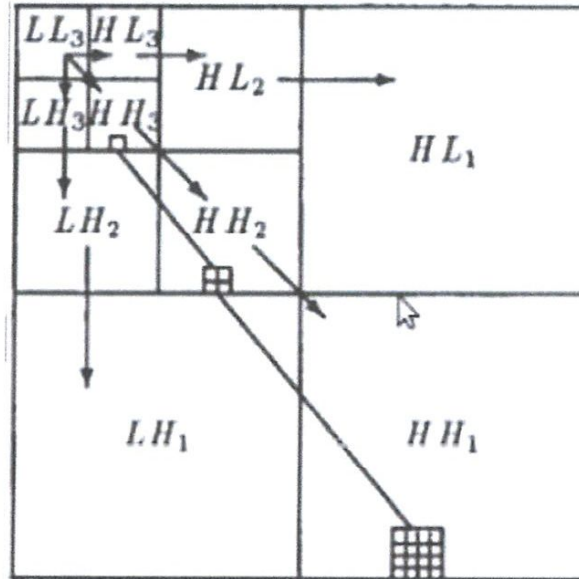


Figure (4.2) : Illustration de la structure de quadtree

Maintenant comment peut-on utiliser cette structure pour exploiter les redondances entre les sous-bandes ? Et bien, on sait que les hautes fréquences d'une image sont en général plus faibles que les basses fréquences. (Cette propriété est d'ailleurs utilisée par JPEG qui conserve plus souvent les basses fréquences d'un bloc 8x8 que les hautes fréquences). Or, comme on l'a dit dans la première partie, les points en haut à gauche, c'est-à-dire correspondantes à des sous-bandes ayant un coefficient élevé correspondent aux basses fréquences alors que celle qui se situent aux autres extrémités correspondent aux hautes fréquences. Donc, vu la manière dont a été construit le quad-tree, on en déduit que plus un nœud est "profond" dans l'arbre, plus il correspond à un point représentant une haute fréquence. Ainsi, si un point de l'image a une valeur faible (au regard d'un certain seuil), alors il y a de très fortes chances pour que tous les fils (ainsi que les petits-fils et les petit-petit-fils, ...etc.) soit aient aux-aussi une faible intensité.

Donc le principe de l'algorithme est le suivant : on choisit un seuil, ensuite et on code l'arbre nœud après nœud, mais dès qu'on trouve un nœud (donc un point) dont l'intensité est faible, on vérifie que toute sa descendance a, elle aussi une faible intensité, et si c'est le cas (et c'est souvent le cas, d'après la propriété évoqué précédemment), on élague l'arbre du nœud (qui est alors appelé un "zerotree") et de toute sa descendance, ainsi on ne code pas une grosse partie de l'image, qui n'est pas significative. Et donc on compresse. Donc,

elle suppose que si une zone d'une image est peu signifiante, elle sera faible pour toutes les sous-bandes correspondantes à des fréquences supérieures.

### 2.3.1. L'algorithme d'EZW :

Après avoir expliqué le principe général de l'algorithme, entrons dans le détail de l'algorithme. L'algorithme se déroule en un nombre fini d'étapes chacune composé de deux passes. On choisit un seuil initial  $T_{init}$  qui est la plus grande puissance de 2 inférieur à la plus grande valeur absolu de l'intensité des points, et on pose  $T = T_{init}$ . Ensuite on effectue un certain nombre de phases chacune composé de deux passes que l'on va décrire :

#### ➤ Première passe : la passe dominante :

On parcourt l'arbre dans un ordre qui est tel que les parents soit toujours parcourus avant leurs fils. Quatre cas sont possibles :

- ❖ Soit le nœud-point considéré à une intensité qui est supérieur au seuil  $T$ , on code alors le point par un symbole nommé **POS** (pour positif).
- ❖ Soit le nœud-point considéré à une intensité qui est inférieur à  $(-T)$ , on code alors le point par le symbole **NEG** (pour négatif).
- ❖ Soit le nœud-point a une intensité qui est inférieur, en valeur absolue, au seuil  $T$ , mais il existe un nœud de sa descendance dont l'intensité est supérieure au seuil  $T$ , on code alors ce point par le symbole **IZ** (pour zéro isolé).
- ❖ Soit le nœud-point a une intensité qui est inférieur, en valeur absolue, au seuil  $T$ , et toute sa descendance est dans le même cas que lui, alors on code alors ce point par le symbole **ZTR** (pour arbre de zéro) et on ne codera pas les nœuds de sa descendance lors du codage des symboles de cette passe.

#### ➤ Deuxième passe : la passe de raffinage :

On ne code dans cette passe que les points qui ont été codés par **POS** ou **NEG** lors d'une des précédentes passes dominantes. Pour chaque point codé, on code 0 si le bit de l'écriture binaire de l'intensité correspondant à  $T/2$  vaut 0, et on code 1 sinon. Autrement dit, on code 0 pour chaque point qui est signifiant vis-à-vis du seuil  $T$ , le résultat de l'opération "ET binaire" entre  $T/2$  et la valeur absolue de l'intensité est nulle et 1 si non.

Ensuite, on recommence en mettant à jour le seuil  $T = T/2$ . Sauf que pendant la passe dominante, on considérera les points qui ont déjà été codés comme valant 0 (car on sait déjà qu'ils sont supérieures à  $T$ , et donc, à fortiori, à  $T/2$ ). Cela permet de larguer plus de



nœud. Et ainsi de suite, jusqu'à qu'on ait la précision voulu ou qu'on a atteint la taille maximale que l'on s'était fixé.

L'ordre dans lequel sont codés les points est un ordre spécial appelé ordre de Morton, qui permet de coder d'abord les sous-bandes  $LH_k$ ,  $HL_k$  et  $HH_k$  avant de commencer de coder les sous-bandes  $LH_{k-1}$ ,  $HL_{k-1}$  et  $HH_{k-1}$ , de telle sorte qu'on est sûr de coder les parents avant les enfants. Voici un schéma de l'ordre de codage :

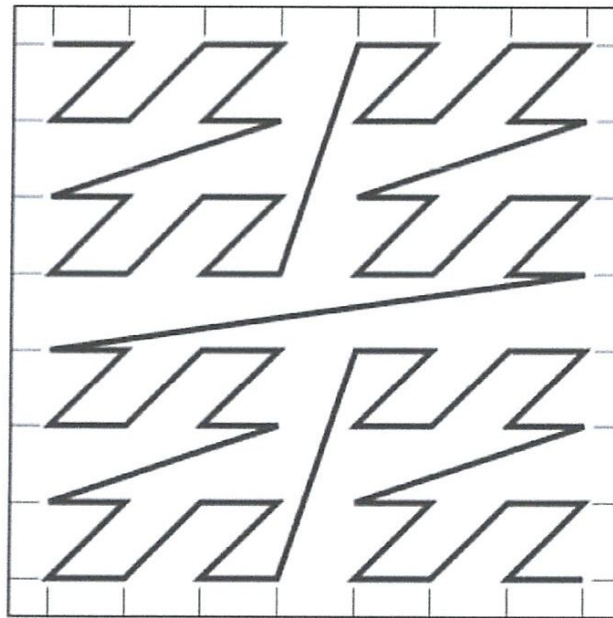


Figure (4.3) : Ordre de Morton

### Exemple :

Prenons un exemple : considérons la matrice suivante qui est le résultat d'une transformée en ondelettes complètes :

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

63 étant la plus grande intensité en valeur absolue, on choisit comme seuil 32 (qui est la plus grande puissance de 2 inférieures à 63). On a alors les symboles suivants :

POS	NEG	POS	ZTR	ZTR	ZTR		
IZ	ZTR	ZTR	ZTR	ZTR	ZTR		
ZTR	IZ						
ZTR	ZTR						
		ZTR	POS				
		ZTR	ZTR				

En suivant l'ordre de Morton, on code donc :

POS NEG IZ ZTR POS ZTR ZTR ZTR ZTR IZ ZTR ZTR ZTR ZTR ZTR ZTR POS  
ZTR ZTR

Puis on passe à l'ordre de raffinement, puisque le résultat du ET binaire entre 63 et 16 n'est pas nul (il vaut 16 car  $63 =_2 111111$  ET  $16 =_2 001000$  donc  $63 \text{ ET } 16 =_2 111111 \text{ ET } 001000 = 001000$ ) donc on code 1. A contrario,  $34 \text{ ET } 16 =_2 100010 \text{ ET } 001000 = 000000 = 10 \text{ 0}$ , donc on code 0. On code donc, au final pour cette passe :

Puis on passe à la seconde phase avec un seuil 16. Cette fois, on a les symboles suivants :

*	*	*					
NEG	POS						
ZTR	ZTR	ZTR	ZTR				
ZTR	ZTR	ZTR	ZTR				
			*				

Donc on le code :

IZ ZTR POS NEG ZTR ZTR ZTR ZTR ZTR ZTR ZTR ZTR

Enn les coefficients déjà codés étant (dans l'ordre où ils ont été codés : 63, -34, 49, 47, -31, et 23, la passe de raffinement donne : 100110

Et on continue, avec les autres passes.

### 3. Résultats et discussions :

Pour comparer les résultats des différents algorithmes, nous avons choisi l'image «cameraman » avec une résolution de 256x256 points (fig(4.4)). Les échantillons de l'image sont codés sur 8 bits, soit 256 niveaux de gris. On peut mesurer le degré de compression soit par le calcul du rapport de compression (RC), soit par le taux de compression avec :

$$RC = \frac{\text{taille de l'image initiale}}{\text{taille de l'image finale}}$$

Le taux de compression est donné par :

$$\text{Taux (\%)} = 100 * \frac{(RC - 1)}{RC}$$

Et le degré d'altération des images (exprimé en terme du rapport signal sur bruit SNR) est :

$$SNR(dB) = 10 * \log_{10} \left( \frac{255^2}{MSE} \right)$$

Où :

$$MSE = \frac{1}{N.M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} e^2(i,j)$$

D'après les résultats obtenus, on peut remarquer que la compression par DCT, en utilisant un masque diagonal donne un rapport de compression supérieur à celle utilisant un masque carré. La qualité de l'image diminue considérablement au-delà du rapport de compression égale à 7 (masque carré) et au-delà de 10 (masque diagonal).

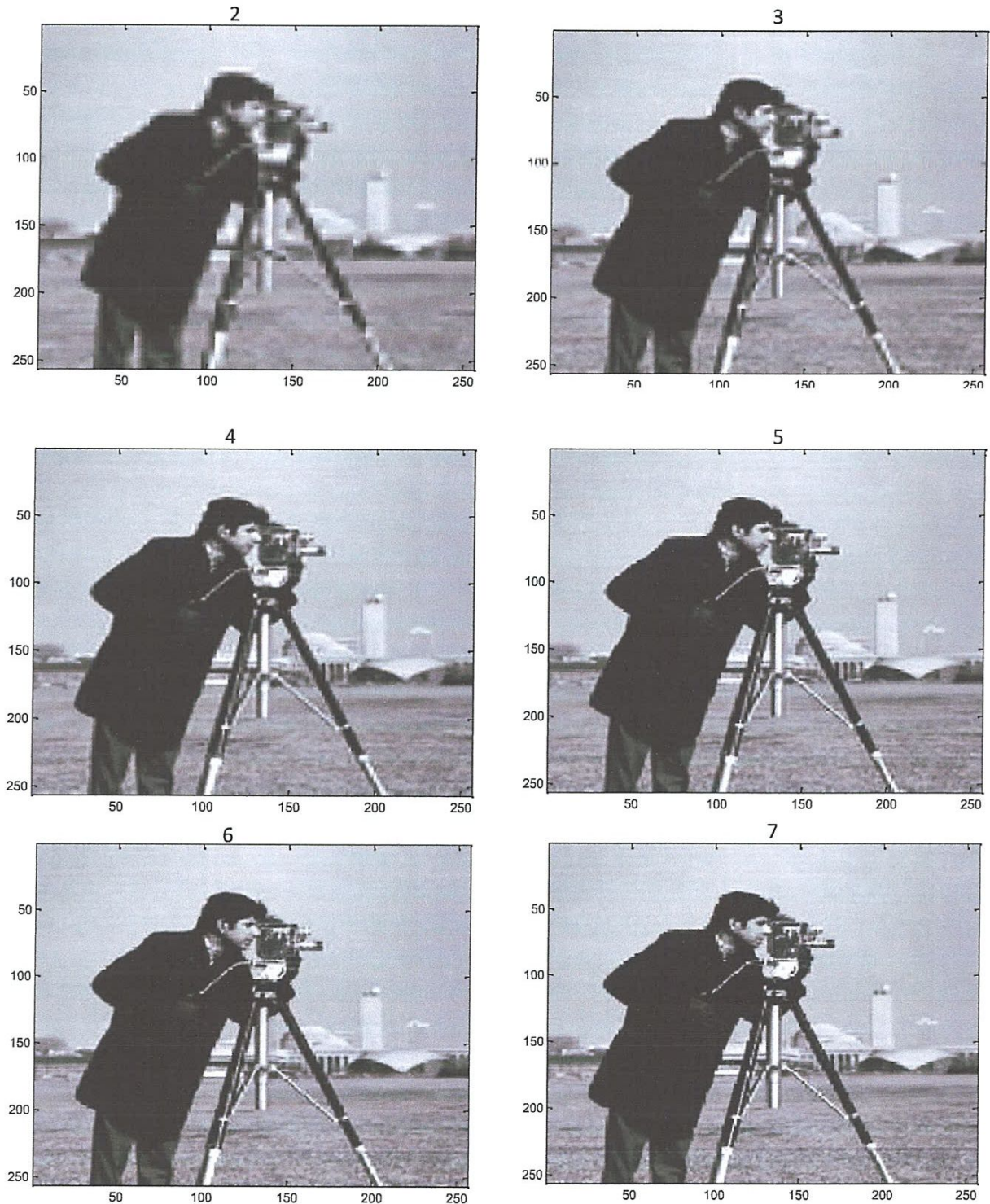


Figure (4.5) : Résultats de la méthode DCT sur l'image Cameraman avec un masque carré

### 3.2. La compression par JPEG :

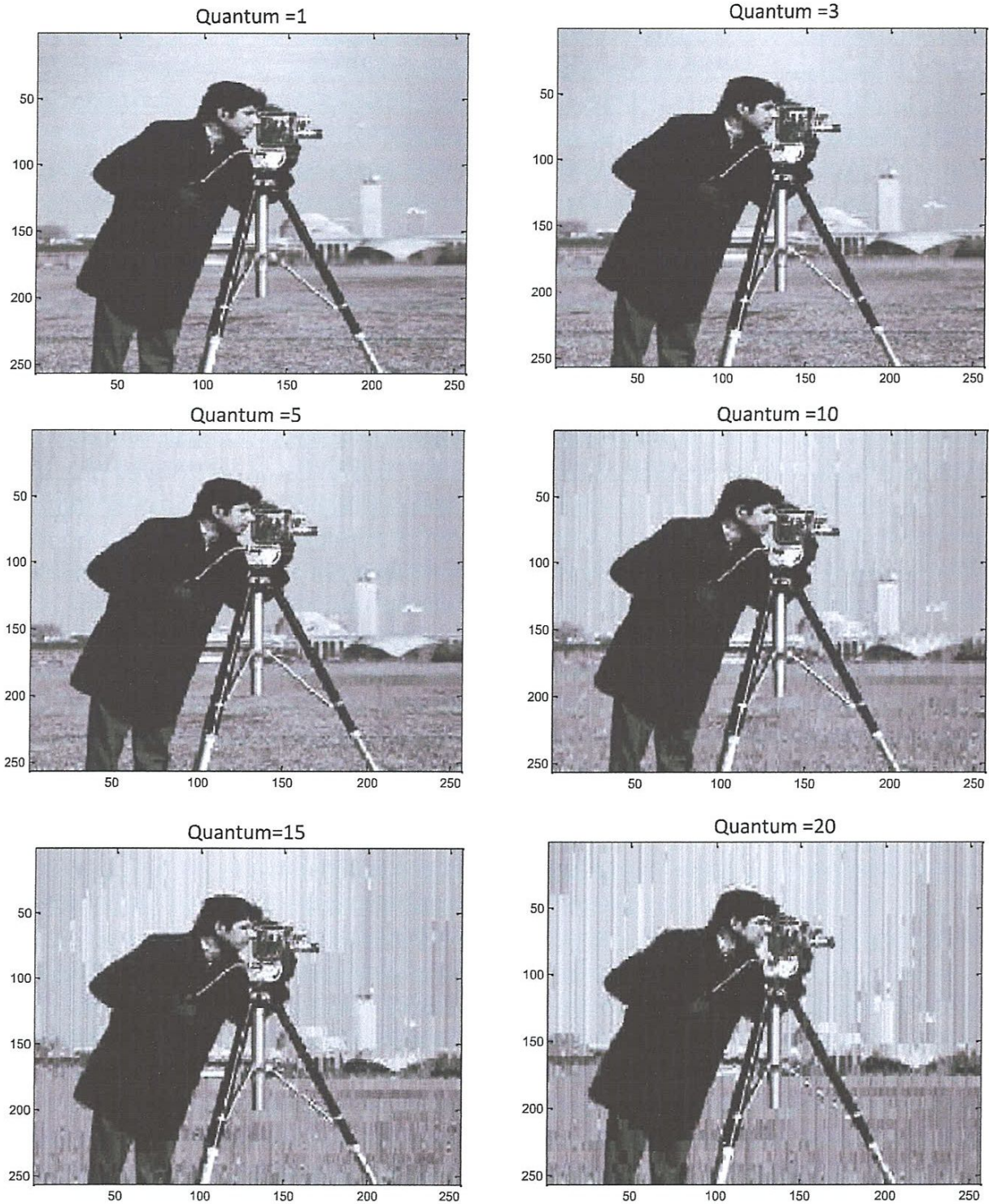


Figure (4.7) : Résultats de la méthode JPEG sur l'image cameraman

D'après les résultats obtenus par la méthode de compression JPEG, on remarque que la qualité de l'image diminue considérablement au-delà du quantum=5 (on voit clairement

l'effet de bloc sur la compression. Tandis que pour des valeurs de quantum  $< 5$ , on obtient des images après un cycle compression/décompression de qualité acceptable.

### 3.2. La compression par EZW :

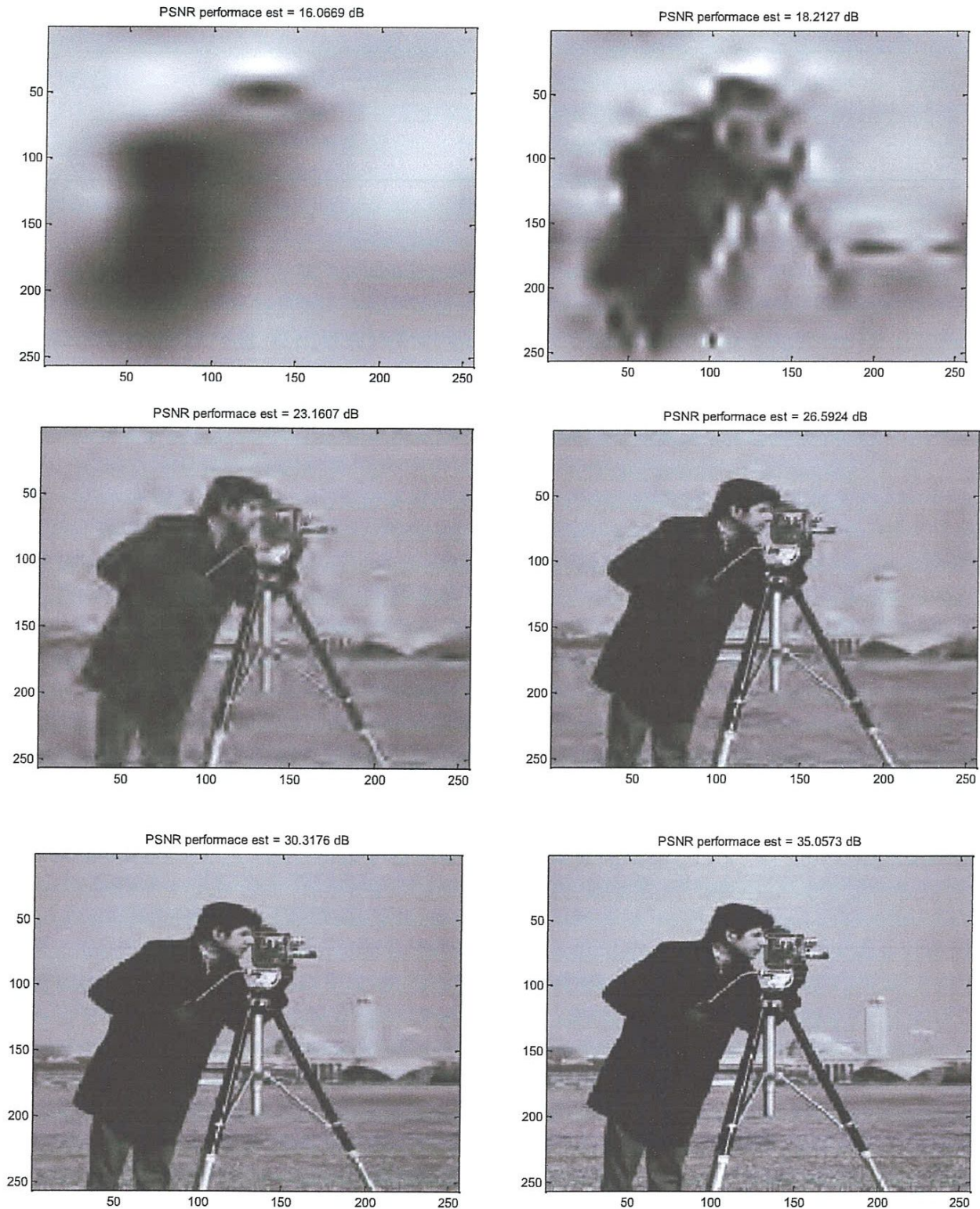


Figure (4.8) : Compression par EZW avec un bitrate = 1.43bpp  
( $ewz\_encoding\_threshold=10$ )

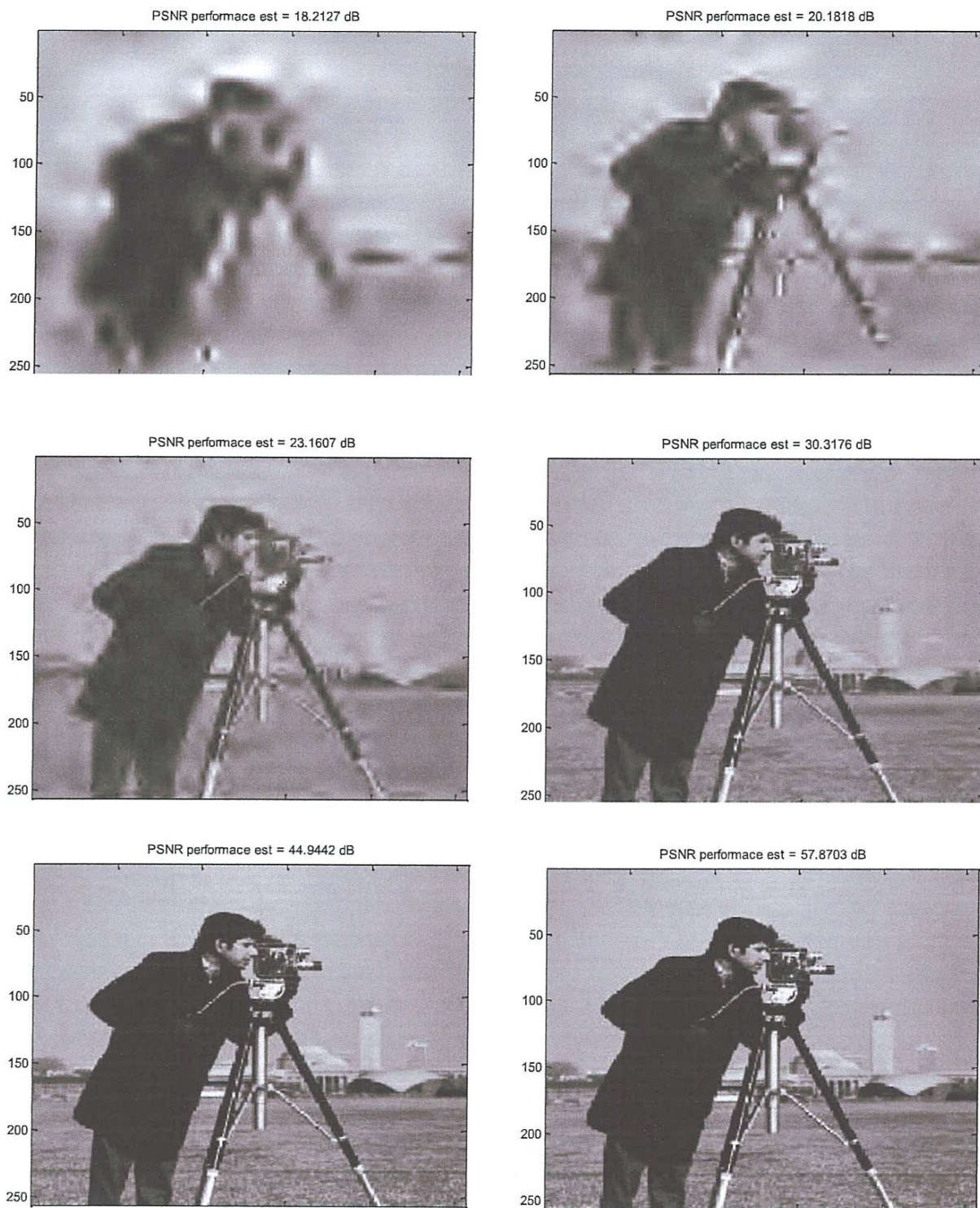


Figure (4.9) : Compression par EZW avec un bitrate =7.62bpp (ezw\_encoding\_threshold=1)

Pour la compression par EZW, on remarque la nature de progressivité de cette méthode, elle donne des résultats acceptable avec un bitrate= 1.43 bits par pixels (bpp) avec un PSNR=35.06dB, et un meilleur résultat avec bitrate=7.62 bpp et un PSNR=57.87dB.

#### 4. Conclusion:

L'algorithme de EZW tire fortement profil de la structure de décomposition en ondelette afin de réaliser une compression efficace de l'image .La compression provient principalement de deux processus .Le premier consiste en l'encodage itératif des coefficients dont les valeurs sont fixées par rapport au seuil lors l'itération considérée. Le second processus de compression réside dans l'utilisation d'arbre de zéro. Lors qu'une racine d'arbre est identifiée, tous les coefficients descendants ne sont plus codés.

Cet algorithme est un exemple d'algorithme cherchant à tirer profit de la décomposition en ondelettes en vue de compresser l'image .On peut notamment citer aussi le SPIHT (Set Partitionning In Hierarchical Trees), cet algorithme tire profit des caractéristiques multi résolution de la composition en ondelettes en vue de permettre de reconstituer l'image progressivement.

Ainsi sur l'internet, il est possible de former rapidement une image complète initialisation floue puis progressivement de plus en plus nette .Les détails sont ajoutés progressivement à l'image au cours de la décomposition. La norme JPEG2000 utilise aussi la transformée en ondelettes pour traiter les images. Dans cette dernière, les ondelettes sont justes une maille d'un dispositif complexe qui vise à obtenir une compression efficace et une grande flexibilité lors de la décompression multiple .L'image peut notamment être décompressée par morceau, ce qui permet à l'utilisateur de zoomer sur les zones qui l'interessent.



# *Conclusion Générale*

## Conclusion

Les images (2D et 3D) constituent une partie importante des données qui transitent sur les réseaux de communication. Le volume de ces données peut être très important. Il est donc nécessaire, pour satisfaire à des contraintes de temps de réaction, de transmettre ces images d'une manière progressive en comprimant au maximum les données.

Dans ce mémoire, nous nous intéressons à la compression hiérarchique d'images fixes. Lors de la consultation d'une banque d'images numérisées, la principale application en est la transmission par étapes <<progressive>> d'une image donnée. Une version grossière (fort taux de compression) parvenant rapidement à l'utilisateur, puis étant <<raffinée>> au cours des étapes successives, jusqu'à une qualité de reconstruction désirée.

Parmi les techniques de transmission progressive utilisées couramment sur les sites WEB, citons le standard JPEG progressif. Cependant, depuis quelques années, dans le domaine du traitement d'images, d'autres schémas de compression hiérarchique ont été développés, s'appuyant dans la plupart des cas sur des représentations multi résolutions du signal-image. Le schéma de compression hiérarchique retenu pour la démonstration s'inspire des schémas de codages développés par Shapiro [24] pour la transmission progressive sans perte <<le codeur EZW>>.

L'idée de l'algorithme EZW est de trouver le meilleur ordre de transmission des coefficients ondelettes, qui est l'ordre décroissant de leurs valeurs absolue. Shapiro proposa de transmettre les coefficients sous forme d'une suite de bits obtenus par enchâssement progressif des bits des coefficients les plus significatifs en commençant par les bits les plus importants.

Ceci permet à l'algorithme EZW de faire la transmission progressive d'images puisque le décodeur peut s'arrêter n'importe où dans la suite de bits transmise, et produire la meilleure image reconstruite possible avec cette suite de bits tronqués.

Après avoir calculé une DWT (Discret Wavelet Transform) de l'image, l'algorithme code les coefficients transformés à l'aide d'une suite décroissante de seuils. L'algorithme effectue récursivement deux passes successives, ne traitant à chaque fois que les coefficients significatifs par rapport au seuil courant. Avant la transmission, l'algorithme compresse les suites de symboles produites par un codeur arithmétique adaptatif. L'algorithme de décodage suit les opérations inverses, c.-à-d. décodage arithmétique, passe dominante puis passe subordonnée et enfin la IDWT (Inverse Discret Wavelet Transform) pour reconstruire l'image.

La robustesse du codeur EZW, face aux différentes attaques que peut subir est remarquablement sauf s'il y a une grande quantité d'information perdue dans l'image transmise qui devient alors inexploitable.

# *Bibliographie*

# Références bibliographiques

## Ouvrage



- [1] KADDOUR Chakib et AISSA BRAHIM Salim, «Compression des images fixes par les approximations fractales basée sur la triangulation de Delaunay et la quantification vectorielle. », Mémoire de fin d'études Université des sciences et de la technologie houari Boumediene Algérie, 1999.
- [2] Mr. BOUMAZA Mouloud, « Contribution à la restauration et fermeture de contours par une approche neuronale », Mémoire de magister, ESI 2001.
- [3] YOUSSEF BENTALEB, « Analyse par ondelettes des signaux sismiques : Applications aux ondes de surface », Thèse doctorat, UNIVERSITÉ MOHAMMED V – AGDAL FACULTÉ DES SCIENCES Rabat-Maroc 2007.
- [4] M. Saaidia, « compression des signaux mono et bidimensionnels, application a la parole aux images », mémoire de magister, Université de Annaba, Institut d'électronique 1994.
- [5] J.P.Gourret, « Modélisation d'image fixes et animées », Masson 1994.
- [6] D. Samai, « Contribution a l'étude et au développent d'algorithme en compression d'image numérique fixes », mémoire de magister, Université de Annaba, Institut d'électronique 1998.
- [7] Harate Hossine, « Etude et analyse de l'approche EZW dans la compression des images », mémoire de magister, Université de Guelma, Département d'électronique 2007.
- [8] LUONG Hông Viêt, « Étude de la méthode de la transformation en ondelette et l'application à la compression des images », Rapport final d'un stage 2005.
- [9] Christophe Damerval, « Ondelettes pour la détection de caractéristiques en traitement d'images », Thèse doctorat UNIVERSITE JOSEPH FOURIER (Grenoble I) 2008.

[10] MESSAOUDENE Ouahiba, BISKRI Badreddine, « réseau de neurones adapté a base d'ondelettes pour la reconnaissance de micro-calcification sur des images mammographiques », Mémoire d'ingénieur ESI 2000.

[11] DAHMANE Omar, RECHID Samir, « compression d'image fixe par transformation ondelettes et quantification vectoriel », Mémoire d'ingénieur ESI 1997.

[12] SANCHEZ Lucie, MEURISSE Christophe, « Traitement des images par les Ondelettes », Mini-projet ENSICAN France 2006.

## Sites Internet

[13] Wikipedia « l'encyclopédie libre» [fr.wikipedia.org](http://fr.wikipedia.org).

[14] <http://www.iict.ch/Tcom/Cours/PDF/CompressionGraphiques.pdf>.

[15] [http://fr.wikipedia.org/wiki/Compression\\_de\\_donn%C3%A9es#Types\\_de\\_compression](http://fr.wikipedia.org/wiki/Compression_de_donn%C3%A9es#Types_de_compression).