

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE



MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE  
UNIVERSITE MOULOU MAMMERI DE TIZI OUZOU  
FACULTE DE GENI ELECTRIQUE ET INFORMATIQUE  
DEPARTEMENT INFORMATIQUE

# Mémoire de fin d'études

En vue de l'obtention d'un diplôme de master en  
informatique.

**Thème :**

**Transformée à base de  
codification d'ordre**

Proposé et dirigé par :  
Mr DIB Ahmed

Réalisé par :  
MOKEDDES Anis  
BOUDLAL Mohamed Ali

Promotion 2013 / 2014



# Remerciements

Nous tenons tout d'abord à remercier très chaleureusement notre promoteur Mr DIB pour sa disponibilité, son soutien et la qualité de son encadrement.

Nos remerciements vont aussi vers :

- Tous les enseignants du département d'Informatique de l'université de Mouloud MAMMARI de Tizi-Ouzou, et particulièrement à ceux qui nous ont encadrés tout au long du cursus universitaire.
- Tous ceux qui, de près ou de loin, nous ont aidés à élaborer ce travail.



# Dédicasse

Je dédie ce modeste travail :

- A la mémoire de ma grand-mère, et de mon premier enseignant Mr IMSOUDANE Omar ;
- A mes très chers parents adorés qui m'ont toujours soutenu et encouragé à aller de l'avant ;
- A mon frère RAFIK, à mes soeurs ASMA et WISSAM, à mon oncle : HAMID, mes complices de toujours que j'adore ;
- A mon cher grand père adoré « ABA » qui j'espère sera présent pour partager toutes nos joies ;
- A Toute la famille MOKEDDES et MORSLI.
- A mon binome BOUDLAL Mohamed Ali ainsi qu'à mes meilleurs amis : Younes, Khirredine, Bilal, Mohammed, Massi, Belaid, Abd El Ghafour, Mehdi, Amine, Faycal, Massilva, Ferrou, Yacine, Loubna, Mounir, Amirouche, Madjid, Ninou... ;
- A tous ceux qui ont contribué de près ou de loin à ma réussite

**MOKEDDES Anis**

Je dedie ce modeste travail :

- A mes très chers parents auxquels je souhaite une longue vie pleine de bonheur ;
- A tous mes amis (es) sans exception
- A tous les étudiants du département Informatique.
- A tous mes proches et toute personne que j'aime.

**BOUDLAL Mohamed Ali**

# Table des matières

<b>Remerciements</b>	<b>i</b>
<b>Dédicasse</b>	<b>iii</b>
<b>Table des matières</b>	<b>v</b>
<b>Table des figures</b>	<b>ix</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Introduction Générale</b>	<b>1</b>
<b>Partie théorique</b>	<b>3</b>
<b>1 La compression de données</b>	<b>5</b>
1.1 Introduction . . . . .	5
1.2 La théorie de l'information . . . . .	5
1.3 Quelques éléments de la théorie de l'information . . . . .	6
1.3.1 Sources d'informations . . . . .	6
1.3.1.1 Définition . . . . .	6
1.3.1.2 Sources discrètes à mémoire . . . . .	6
1.3.1.3 Sources discrètes sans mémoire . . . . .	6
1.3.2 La quantité de l'information . . . . .	7
1.3.3 L'entropie . . . . .	8
1.3.3.1 Entropie maximale . . . . .	8
1.3.3.2 Utilité de l'entropie . . . . .	8
1.3.4 La redondance . . . . .	8
1.4 La modélisation . . . . .	9

## TABLE DES MATIÈRES

---

1.5	Codage et code . . . . .	9
1.5.1	Définitions . . . . .	9
1.5.2	Codage de la Source . . . . .	9
1.5.3	Propriétés d'un code . . . . .	10
1.5.4	Les types de codes (selon la taille) . . . . .	10
1.5.5	Code déchiffrable . . . . .	11
1.5.6	Propriété de préfixe . . . . .	11
1.5.7	Classification de codage . . . . .	12
1.6	La compression et la décompression de données . . . . .	12
1.6.1	La compression . . . . .	12
1.6.1.1	Définition . . . . .	12
1.6.1.2	Intérêt . . . . .	12
1.6.2	La décompression . . . . .	13
1.6.3	Le schéma général de compression/décompression . . . . .	13
1.6.3.1	Le module de compression (compresseur, encodeur)	14
1.6.3.2	Le module de décompression . . . . .	14
1.6.4	Les types de compression de données . . . . .	14
1.6.4.1	La compression physique . . . . .	14
1.6.4.2	La compression logique . . . . .	14
1.6.4.3	Compression symétrique/asymétrique . . . . .	14
1.6.4.4	La compression statistique / numérique . . . . .	15
1.6.4.5	La compression adaptative, semi adaptative et non adaptative . . . . .	15
1.6.4.6	La compression sans perte et la compression avec perte . . . . .	16
1.7	Topologie de données à compresser . . . . .	18
1.7.1	Le texte . . . . .	18
1.7.2	Le son . . . . .	18
1.7.3	L'image . . . . .	18
1.7.3.1	La définition . . . . .	18
1.7.3.2	La numérisation . . . . .	18
1.7.4	La vidéo . . . . .	19
1.8	Les formats de fichiers de données . . . . .	19
1.8.1	Les formats graphiques . . . . .	19
1.8.2	Format vidéo . . . . .	21
1.8.3	Format audio . . . . .	22
1.9	Evaluation des performances d'une compression . . . . .	22
1.9.1	Le taux de compression . . . . .	23
1.9.1.1	Le degré de compression . . . . .	23
1.9.2	Le gain de compression . . . . .	23

1.9.3	Le Taux d'information . . . . .	23
1.9.4	Mesure de fidélité . . . . .	23
1.9.5	La vitesse de compression . . . . .	24
1.9.6	Mesure de perte d'information . . . . .	24
1.10	Conclusion . . . . .	24
<b>2</b>	<b>Etat de l'art de la compression de donnée par la transformée</b>	<b>25</b>
2.1	Introduction . . . . .	25
2.2	Définition des ondelettes . . . . .	25
2.3	Transformée de Fourier . . . . .	26
2.3.1	Transformée de Fourier à fenêtre glissante . . . . .	26
2.4	Transformée en ondelettes . . . . .	27
2.4.1	Transformation continue en ondelettes . . . . .	28
2.4.2	Transformation discrète en ondelettes . . . . .	29
2.4.3	Familles d'ondelettes . . . . .	29
2.4.3.1	Ondelettes orthogonales . . . . .	29
2.4.3.2	Ondelettes biorthogonales . . . . .	30
2.4.3.3	Ondelettes usuelles . . . . .	30
2.4.3.4	Ondelettes sur un intervalle . . . . .	30
2.4.3.5	Ondelettes multidimensionnelles . . . . .	30
2.5	Bancs de filtres . . . . .	31
2.5.1	Notion des bancs de filtres . . . . .	31
2.5.2	Réalisation d'un banc de filtres à base d'ondelettes . . . . .	31
2.6	Compression d'image par transformée en ondelettes . . . . .	32
2.6.1	Compression en deux dimensions . . . . .	32
2.6.2	Compression en trois dimensions . . . . .	33
2.7	Conclusion . . . . .	34
	<b>Partie pratique</b>	<b>37</b>
<b>3</b>	<b>Codage EZW</b>	<b>39</b>
3.1	Introduction . . . . .	39
3.2	Présentation du codage EZW . . . . .	39
3.3	Algorithme . . . . .	42
3.3.1	Traitement principal . . . . .	42
3.3.2	Traitement secondaire . . . . .	44
3.3.3	Protocole de codage . . . . .	44
3.3.4	Décodage . . . . .	44
3.4	Conclusion . . . . .	47

<b>4</b>	<b>Implimentation , simulation et discution</b>	<b>49</b>
4.1	Introduction . . . . .	49
4.2	Simulation . . . . .	50
4.2.1	Structure du programme . . . . .	50
4.2.2	Décomposition tridimensionnelle en ondelettes . . . . .	52
4.2.3	Codeur EZW3D . . . . .	53
4.3	Résultats . . . . .	55
4.3.1	Les séquences de test . . . . .	55
4.3.2	Estimation des performances . . . . .	55
4.4	Les tableaux récapitulatifs des résultats . . . . .	58
4.5	Discussions . . . . .	60
4.6	Conclusion . . . . .	61
	<b>Conclusion et perspectives</b>	<b>63</b>
	<b>Bibliographie</b>	<b>65</b>

# Table des figures

1.6.1	Schéma général de compression/décompression . . . . .	13
1.6.2	Compression de type symétrique . . . . .	15
2.6.1	Un étage de décomposition multi-résolution bidimensionnelle . . .	33
3.2.1	Représentation de l'organisation en arbre des coefficients d'ondelettes	41
3.3.1	Méthodes de parcours des coefficients . . . . .	43
3.3.2	Exemple de Shapiro . . . . .	45
4.2.1	Algorithme général de codage et décodage . . . . .	51
4.2.2	Décomposition en ondelettes 3D dyadique à deux échelles spatio-temporelles . . . . .	53
4.2.3	Illustration de chaque parent ayant huit fils . . . . .	54
4.3.1	Image 01 de la séquence Brain (Droite) et Image 01 de la séquence Head (Gauche) . . . . .	55
4.3.2	Graphe : Taux de compression / PSNR . . . . .	56
4.3.3	Coefficient des filtres de décompositions et de reconstructions associés obtenus par l'ondelette de Haar . . . . .	57



# Liste des tableaux

4.1	Résultats obtenus par le filtre Haar . . . . .	58
4.2	Résultats obtenus par le filtre db4 . . . . .	58
4.3	Résultats obtenus par le filtre Bior3.3 . . . . .	58
4.4	Résultats obtenus par le filtre Haar . . . . .	59
4.5	Résultats obtenus par le filtre db4 . . . . .	59
4.6	obtenus par le filtre Bior3.3 . . . . .	59



# Introduction Générale

La compression d'image est un domaine de recherche déjà riche d'une longue histoire. Trouvant ses racines théoriques dans la théorie de l'information initiée par C. Shannon en 1948, elle a depuis fait appel à de nombreux outils mathématiques de plus en plus sophistiqués. La problématique est cependant d'actualité et découle des applications de stockages des données ou de transmission à travers les canaux à bande passante limitée. Malgré les développements technologiques considérables ces dernières années, l'être humain, insatiable, déclare indispensable ce qu'hier il considérait superflu, et nous sommes sans cesse confrontés à de nouveaux besoins exigeant l'accès à toujours plus d'information : les intérêts de la compression sont d'une grande importance et indispensables pour de nombreuses applications.

La compression d'image met en jeu différentes techniques. On peut grossièrement les diviser en quelques grandes familles. Certaines ont pour but de structurer la redondance présente dans l'image que l'on souhaite compresser. C'est le cas par exemple de la transformation, qui cherche à concentrer l'information dans un petit nombre d'éléments en changeant de domaine, et de la prédiction, qui calcule un estimateur de l'information courante à partir d'une information passée. Nous allons dans une chaîne de compression d'images fixes accorder notre réflexion à l'étape de transformation des coefficients en vue d'un changement d'espace en exploitant les outils de transformée en ondelettes et en curvelets puis en évaluant comparativement chacune leur performance en terme de taux de compression, qualité d'image et rapidité de compression.

Ce rapport est composé de quatre chapitres. Les trois premiers présentent le contexte théorique des travaux et recherches déjà réalisés dans le domaine sont parfaitement dépendants, décrivent et présentent les limites et avantages respectivement des transformées Fourier, Ondelettes et Curvelets dans un cadre général et puis particulièrement dans le domaine de la compression d'images fixes. Le dernier chapitre présentera à travers une plate forme dynamique les résultats des compressions obtenues par ces deux transformées et une étude comparative basée sur les critères de performances en termes de taux de compression, qualité de l'image compressée et la rapidité de compression.



# Partie théorique

« Dix mille difficultés ne font pas un doute. »"

---

*(Isaac Newton)*



# Chapitre 1

## La compression de données

### 1.1 Introduction

La compression de données est un vaste sujet qui a fait l'objet de nombreux ouvrages et articles. Elle donne lieu aujourd'hui à de nombreuses recherches en raison des enjeux économiques. Elle est utilisée majoritairement dans les applications informatiques.

La compression de données est considérée comme solution qui peut être employée pour alléger une portion de problème de stockage pour faire en sorte que les données occupent le minimum d'espace pour une vitesse de restriction acceptable. Par ailleurs, la compression peut participer à l'opération de sauvegarde de données et offrir un niveau de cryptage contre des intrusions illicites puisqu'un texte initialement représenté par un code conventionnel comme le standard ASCII est converti en un code différent.

### 1.2 La théorie de l'information

C'est la théorie mathématique traitant des bases théoriques de la transmission et du traitement de l'information. La théorie de l'information est une approche statistique dont les premiers rudiments ont été proposés par HARLEY en 1928, mais dont les fondements n'ont vraiment été établis qu'en 1948 par le mathématicien CLAUDE, SHANNON dans son livre intitulé « théorie mathématique de la communication ».

Cette théorie s'intéresse à la mesure de la quantité d'information, à la représentation de cette information, dite aussi codage, ainsi qu'aux systèmes de communication qui la transmettent et la traitent. Ce codage peut ainsi se référer à la conversion du son et d'images en signaux électromagnétiques.

Outre les télécommunications l'électronique et l'informatique, la théorie de l'information s'applique à divers domaines comme la cybernétique, la linguistique et la psychologie.

## 1.3 Quelques éléments de la théorie de l'information

### 1.3.1 Sources d'informations

Nous considérons ici des sources discrètes d'information. C'est-à-dire des sources qui produisent un nombre fini d'éléments s'appelaient symboles de sources et noté  $S_i$ .

#### Exemple

Source d'information clavier d'ordinateur.

#### 1.3.1.1 Définition

L'ensemble des symboles de la source constitue un alphabet.

#### Exemple

L'alphabet d'un clavier de téléphone portable est l'ensemble  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, \#\}$ .

#### 1.3.1.2 Sources discrètes à mémoire

L'hypothèse de symboles indépendants n'est pas toujours vérifiée. Prenons exemple à un texte en français : la probabilité d'avoir la lettre "u" est très grande si la lettre précédente est "q". Cette probabilité d'avoir conditionnelle est noté  $P(\frac{u}{q})$ . Les pixels d'une image sont aussi fortement corrélés car il est rare que la couleur ou la luminosité d'un pixel change complètement par rapport aux pixels voisins.

#### 1.3.1.3 Sources discrètes sans mémoire

Pour caractériser les sources, de nombreux termes sont empruntés de la description du langage courant. Une source dispose d'un "alphabet" constitué d'éléments ou symboles ou caractères  $\{x_1, x_2, x_3, \dots, x_k\}$   $k$  est la longueur de l'alphabet. Ces symboles sont associés pour constituer un message. Emettre un message revient à émettre une succession de symboles appartenant à une source. Chaque symbole  $x_i$  de l'alphabet a une probabilité d'utilisation  $p_i$ . Pour simplifier le problème, une catégorie de sources est plus simple à modéliser : celle des sources pour lesquelles la probabilité d'émission d'un caractère est indépendante de ce qui a été émis avant ou sera émis après. C'est ce qui définit une source sans mémoire.

### 1.3.2 La quantité de l'information

L'information produite par une source doit être quantifiée. Pour cela on a d'abord définie la quantité d'information contenue dans un symbole. Partant de l'idée que si la source produit toujours le même symbole. Ce symbole est prévisible et ne contient pas d'information. A l'opposé, plus se symbole est rare, plus il contient de l'information. La notion de l'information donc liée à la probabilité d'apparition  $P(S_i)$  des symboles  $S_i$  pour une source dont les symboles sont statistiquement indépendants (source discrète sans mémoire) la quantité d'information d'un symbole est définie par :

$$l(S_i) = \log_2 \left( \frac{1}{P(S_i)} \right)$$

Cette définition peut être généralisée à la quantité d'information d'un message. C'est-à-dire une suite de symboles, en remplaçant  $P(S_i)$  par  $P(M)$  la probabilité du message, si un message est composé de ' $m$ ' symboles statistiquement indépendant, sa quantité d'information équivaudra à la sommes des quantités d'information de chaque symboles car :

$$P(M) = P(S_1) * P(S_2) * \dots * P(S_m)$$

Où :  $P(M)$  est la probabilité du message en question, soit produit. Donc : La quantité d'information d'un message  $M$  de  $m$  symboles est :

$$l_m = \sum_1^n l_i$$

La quantité d'information d'un message désigne également le nombre de symboles binaire nécessaires pour représenter ce message. A partir des remarques suivantes :

- La quantité d'information d'un symbole est d'autant plus grande que celui ci est peu probable.
- La quantité d'information de deux symboles successifs est la somme de leur quantité d'information.

La quantité d'information notée  $I$  est une fonction qui doit ainsi avoir les propriétés suivantes :

1.  $I()$  est une fonction continue de la probabilité  $P_i$
2.  $I(P_k)$  croit si  $P_k$  décroît donc  $I(P_k)$  est une fonction décroissante de  $P_k$
3.  $I(P_k \text{ et } P_j) = I(P_k) + I(P_j)$ .
4. Un symbole qui possède une quantité d'information nulle :  $I(P_k = 1) = 0$

Une fonction mathématique remplit les conditions : 1, 3 et 4 :  $\log(P_k)$  Pour obtenir la propriété 2 il suffit de prendre  $\log(P_k) = -\log(\frac{1}{P_k})$ .

La quantité d'information d'un symbole XK de probabilité PK est ainsi définie par Shannon comme suit :

$$I(x_k) = -\log(P_k) = \log(\frac{1}{P_k})$$

### 1.3.3 L'entropie

En générale, l'entropie mesure le degré de désordre dans un system, en théorie de l'information, il indique le poids d'information mathématique que porte un message, et donc sa compressibilité théorique. L'entropie (H) d'un message est l'information moyenne contenue par chaque symbole, elle est donnée par la relation suivante :

$$H(x) = \sum_{i=1}^n P(i) \log_2\left(\frac{1}{P(i)}\right) = -\sum_{i=1}^n P(i) \log_2 P(i)$$

#### 1.3.3.1 Entropie maximale

L'entropie est maximale  $H_{max}(x)$  si la probabilité d'apparition des symboles est égale à  $\frac{1}{n}$ .

#### 1.3.3.2 Utilité de l'entropie

D'un strict point de vue quantitatif, on ne peut que constater que la plupart des langues comprennent beaucoup moins de mots que les possibilités offertes par l'alphabet, ainsi les codages traditionnels, qui représentent ou cherchent à représenter toutes les combinaisons théoriques, se révèlent extrêmement simplistes et peu performants. Le codage ASCII sur 7 bits, par exemple, fournit plus de 4 000 milliards de possibilités pour coder les mots de 8 lettres, alors que les dictionnaires de la langue française comptent de moins de 3 000... En tenant compte des caractéristiques d'entropie des données à traiter, il est donc possible de définir des codages plus performants. Les travaux de Shannon et ses collaborateurs ont conduit à développer des codages fondés sur la fréquence d'apparition d'une information.

### 1.3.4 La redondance

La source  $X$  est dite sans redondance

si tous les événements de  $X$  apparaissent avec la même probabilité ( $P(x_1) = P(x_2) = \dots = P(x_n) = \frac{1}{n}$ ).

La redondance ( $R$ ) d'une source caractérise la différence qu'il existe entre la quantité d'information que transportée cette source et la quantité d'information que cette source transporterait si tous ses symboles étaient équiprobables et indépendants. On a :

$$R = 1 - \left( \frac{H(x)}{H_{max}(x)} \right)$$

$R$  est compris entre 0 (les symboles de la source sont indépendants et équiprobables) et 1 (l'entropie de la source est nulle).

### Exemple

Dans la langue française la lettre «e» se trouve plus de fois que la lettre « f » dans les messages de texte donc «e» à plus de redondance que « f ».

## 1.4 La modélisation

C'est l'opération qui consiste à extraire à partir du flux de données en entrées, les informations qui représenteront ce flux dans le codeur, et à base des quelles ce dernier effectuera le codage.

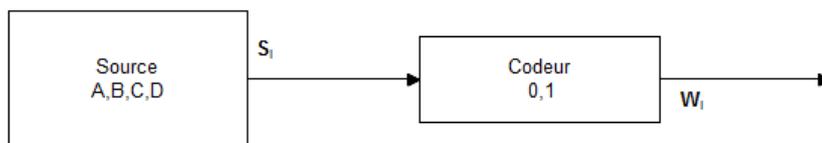
## 1.5 Codage et code

### 1.5.1 Définitions

C'est une fonction qui fait correspondre à tout symbole ou groupes de symboles du flot de données à compresser, par des symboles plus compactes.

### 1.5.2 Codage de la Source

Le codage de source consiste à représenter les symboles de source  $S_i$  par des mots  $W_i$  appelés Mots de code. Le code est la table de conversion entre les symboles  $S_i$  et les mots de code  $W_i$  qui sont souvent des mots binaires. La conversion est effectuée par un codeur. La longueur d'un mot de code est le nombre de symboles binaires qu'il contient.



### 1.5.3 Propriétés d'un code

Pour avoir un codage correct et bon il faut qu'il vérifie les propriétés suivantes :

- Tous les mots du code peuvent être distingués.
  - Le décodage ne donne lieu à aucune ambiguïté.
  - Il doit être préfixé : Aucun mot du code n'est un sous mot initial d'un autre.
- Cette propriété permet d'assurer un décodage unique.

### 1.5.4 Les types de codes (selon la taille)

Les différents types de code peuvent se présenter sous les formes suivantes :

#### 1.5.4.1 Code à longueur fixe(FLC)

(fixed length codes) est un code qui associé à tous les messages de la source un nombre constant de bits(les événements de la source sont codés sur le même nombre de bits).

#### 1.5.4.2 Code à longueur variable(VLC)

(variable length codes) est un code qui associé les messages de la source à un nombre de bits).

Les codes à longueur variable sont souvent choisis pour la compression des données, parce qu'avant toute opération d'enregistrement des données assigne des codes plus courts pour les symboles plus fréquents, et des codes plus longs pour des symboles plus rares.

#### Exemple

Considérons le code à longueur variable suivant (0, 100, 101, 110, 111) avec des longueurs de code (1, 3, 3, 3, 3) pour l'alphabet (A, B, C, D, E) et la chaîne source BAAAAAAC avec des fréquences pour chaque symbole (7, 1, 1, 0, 0). la moyenne en nombre de bit est :

$$l = \frac{1 * 7 + 1 * 3 + 3 * 1}{9} = 1.4 \text{ bits/symbole}$$

C'est presque un enregistrement avec la moitié de bits comparé avec 3 bits/symbole si on utilise un code de longueur fixe à 3 bits. Le plus court code est attribué au plus fréquent symbole dans la chaîne source.

Comme ça le code est le meilleur de point de vue compression.

### Exemple d'application des VLCs

Le code morse assigne les petits codes pour les lettres les plus fréquentes et les codes plus long pour les moins fréquentes.

Le domaine de fabrication de microprocesseurs. Le principe est de chercher une instruction en mémoire et de l'exécuter, ces instructions sont généralement codés avec des tailles fixes, cela facilite leur manipulation par le microprocesseur. Mais les processeurs modernes optent pour une taille d'instructions variables. Il est alors possible de réduire la taille des programmes en utilisant des petits codes pour les instructions les plus fréquentes dans des programmes (exemple : l'instruction de changement : load). cela réduit aussi la taille physique du microprocesseur et consommation d'énergie.

Le standard ISBN (International Standard Book Number), est un nombre unique qui est attribué aux livres. Une partie de ce code désigne le code de pays de publication, la longueur de ce code varie entre 1 et 5 digits, et il assigne des codes pour les pays qui publient plus.

Le code d'appel téléphonique international. La recommandation standard internationale attribue un code de longueur variable pour les pays selon la quantité d'appareil téléphonique que ce pays possède.

### 1.5.5 Code déchiffrable

Un code  $C$  sur un vocabulaire  $V$  est dit uniquement déchiffrable (on dit parfois non ambigu) si seulement si, pour tout  $x = x_1 \dots x_n \in V_+$ . Il existe au plus une séquence  $c = c_1 \dots c_m \in C_+$  tel que  $c_1 \dots c_m = x_1 \dots x_n$ .

#### Propriété

Un code  $C$  sur un vocabulaire  $V$  est uniquement déchiffrable si et seulement si pour toutes séquence  $c = c_1 \dots c_n$  et  $d = d_1 \dots d_m$  de  $C_+$ .

$$C = d \rightarrow (n = m \text{ et } 1 \leq i \leq n, c_i = d_i)$$

#### Théorème 1 (Kraft)

il existe un code uniquement déchiffrable sur un vocabulaire  $V$  dont les mots  $\{c_1 \dots c_n\}$  sont de longueur  $\{l_1 \dots l_n\}$  si et seulement si :

$$\sum_{i=0}^n \frac{1}{|V|^{l_i}} \leq 1$$

### 1.5.6 Propriété de préfixe

On dit qu'un code sur un vocabulaire  $V$  à la propriété du préfixe (on dit parfois qu'il est instantané, ou irréductible) si et seulement si pour tout couple de mots de code distincts  $(c_1, c_2)$ ,  $c_2$  n'est pas un préfixe de  $c_1$ .

Grace à la propriété de préfixe, on peut déchiffrer les mots d'un tel code dès la fin de la réception du mot( $x_n$ ), ce qui n'est pas toujours le cas pour les codes uniquement déchiffrables.

### Exemple

Si  $V = 0, 01, 11$  et si on reçoit le message  $m = 001111111111111111 \dots$  il faut attendre le symbole qui suit 0 pour pouvoir savoir déchiffrer (0 ou 01?).

### Remarque

- Tout code possède la propriété de préfixe est uniquement déchiffrable
- Tout code dont tous les mots sont de la même longueur possède la propriété de préfixe

## 1.5.7 Classification de codage

On peut donner trois types de codage :

### 1.5.7.1 Codage de canal

Il est utilisé pour le cryptage des données à transmettre pour des raisons de sécurité d'information.

### 1.5.7.2 Codage correcteur d'erreurs

on les utilise pour la correction des erreurs de transmission.

### 1.5.7.3 Codage de la source

Il est utilisé pour la compression de données.

## 1.6 La compression et la décompression de données

### 1.6.1 La compression

#### 1.6.1.1 Définition

La compression de données (en anglais data compression ou data compaction) traite de la manière dont on peut réduire l'espace nécessaire à la représentation d'une certaine quantité d'information.

Elle a donc sa place aussi bien lors de la transmission que lors du stockage des données. Elle est très utile pour plusieurs applications informatiques.

#### 1.6.1.2 Intérêt

De nos jours, la puissance des processeurs augmente plus vite que les capacités de stockage, et énormément plus vite que la bande passante des réseaux, car cela

demande d'énormes changements dans les infrastructures de télécommunication.

Ainsi, pour pallier ce manque, il est courant de réduire la taille des données en exploitant la puissance des processeurs plutôt qu'en augmentant les capacités de stockage et de transmission des données.

### 1.6.2 La décompression

On utilise le verbe décompacter (to unpack), c'est la restauration des éléments compressés ou compactes à leurs format d'origine. Donc c'est le processus inverse de la compression.

### 1.6.3 Le schéma général de compression/décompression

Il montre de manière globale l'opération de compression décompression de données.

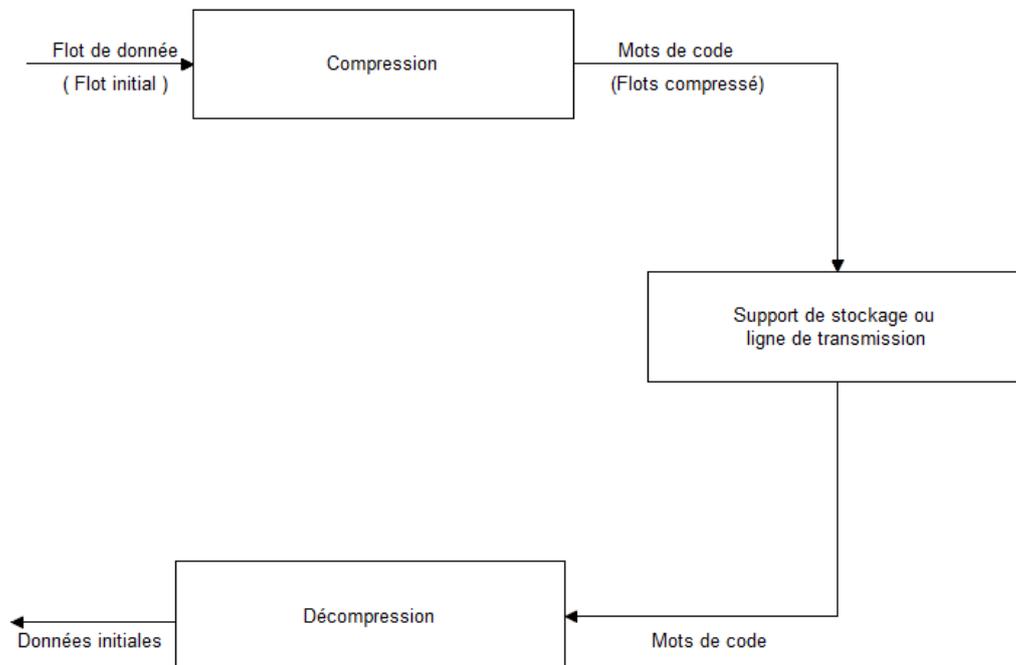


FIGURE 1.6.1 : Schéma général de compression/décompression

### 1.6.3.1 Le module de compression (compresseur, encodeur)

Il reconnaît les messages délivrés par la source, code ces messages pour avoir une suite de codes en sortie.

### 1.6.3.2 Le module de décompression

Il restitue les données initiales à partir des codes délivrés par une ligne de transmission ou par un support de stockage.

## 1.6.4 Les types de compression de données

Dans le Domain de la compression, il existe plusieurs façons de comparer les types de compression. Pour cette raison, nous allons voir comment classifier les types de compression de données.

### 1.6.4.1 La compression physique

Elle agit directement sur les données, il s'agit de regarder les données redondantes d'un train de bits à un autre.

### 1.6.4.2 La compression logique

Elle est effectuée par un raisonnement logique en substituant une information par une information équivalente (recorder les données dans une représentation différente plus compacte contenant la même information).

#### Exemple

Remplacer un symbole alphabétique, numérique ou binaire en un autre. Changer « United State of America » en « USA » est un bon exemple de substitution logique car « USA » est dérivé directement de l'information contenue dans la chaîne « United State of America » et garde la même signification. La substitution logique ne fonctionne qu'au niveau du caractère ou plus haut et est basée exclusivement sur l'information contenue à l'intérieur des données.

### 1.6.4.3 Compression symétrique/asymétrique

Dans le cas de la compression symétrique, la même méthode est utilisée pour compresser et décompresser l'information, il faut donc la même quantité de travail pour chacune de ces opérations. C'est ce type de compression qui est généralement utilisée dans les transmissions de données. La compression asymétrique demande plus de travail pour l'une des deux opérations, la plupart des algorithmes requiert plus de temps de traitement pour la compression que pour la décompression. Des

algorithmes plus rapides en compression qu'en décompression peuvent être nécessaires lorsque l'on archive des données auxquelles on accède peu souvent (pour des raisons de sécurité par exemple).

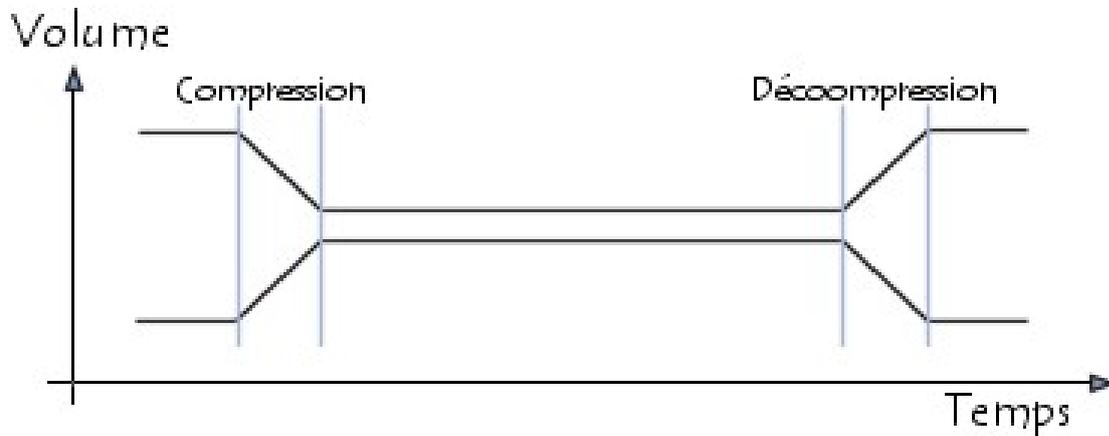


FIGURE 1.6.2 : Compression de type symétrique

#### 1.6.4.4 La compression statistique / numérique

On peut encore distinguer les algorithmes qui travaillent au niveau statistique et ceux qui opèrent au niveau numérique.

Pour les premiers types de compression la valeur des motifs ne compte pas. Ce sont les probabilités qui comptent, et le résultat est inchangé par substitution des motifs tandis que les seconds, les valeurs des motifs influent sur la compression (par exemple JPEG), et les substitutions sont interdites.

#### 1.6.4.5 La compression adaptative, semi adaptative et non adaptative

##### La compression non adaptative

Certaines méthodes statiques utilisant le codage Huffman sont conçues pour compresser seulement des types spécifiques de données. On utilise des codages non-adaptatifs qui contiennent un dictionnaire statique de chaînes de caractères prédéfinies qui sont connues comme apparaissant à de grandes fréquences dans les données à encoder. Par exemple, un codeur non-adaptatif conçu spécifiquement pour compresser la langue française contiendra un dictionnaire avec des chaînes de caractères telles que "et", "mais", "le", car ces chaînes apparaissent très fréquemment dans les textes en français.

### **La compression adaptative**

Un algorithme adaptatif comme LZW ou AFE, à l'inverse n'intégrera pas de données relatives à la fréquence d'apparitions des données à compresser. Des compresseurs adaptatifs comme LZW ou Huffman dynamique déterminent la dépendance des données en construisant leur dictionnaire à la volée. Ils n'ont pas de listes prédéfinies de chaînes de caractères par exemples mais les construisent dynamiquement à l'encodage.

La compression adaptatif est capable de s'adapter à n'importe quelles données d'entrées et de retourner une sortie avec le taux de compression le meilleur possible. C'est une des principales différences avec les compressions non-adaptatives qui sont capable d'avoir des codages efficaces uniquement avec un type de données d'entrées très restreint pour lequel ils ont été conçus.

### **La compression semi adaptative**

Un mélange de ces deux méthodes. Un encodage semi-adaptatif fait un premier passage sur les données pour construire le dictionnaire et un second passage pour effectuer l'encodage. En utilisant cette méthode, un dictionnaire optimal est construit avant qu'un quelconque encodage soit effectué.

Enfin le critère de classification le plus pertinent est basé sur la perte des données.

## **1.6.4.6 La compression sans perte et la compression avec perte**

### **1.6.4.6.1 La compression sans perte (non destructive)**

Une compression est dite « sans perte » lorsqu'on peut obtenir les données initiales compressées sans qu'il ait eu aucune altération. Les programmes informatiques, codes sources et rapports sont toujours encodés de cette façon.

On utilise la compression sans perte pour compacter les sources de données ou toute perte d'information est une destruction de contenu de la source.

### **Quelque domaine qui demande ce type de compression**

- Les informations provenant des sondes spéciales dont les données qui les représentent sont sensible au contexte, tout changement (suppression de donnée) engendre une modification radicale dans les valeurs scientifique.
- L'imagerie médicale, ou pour faire une analyse médicale exacte on a besoin d'une image originale qui donne les informations réelles, parce que le moindre détail peut changer le résultat de l'analyse.

### **Remarque**

La compression sans perte utilise des algorithmes qui compressent les données sans les dégrader. Autrement dit, après la décompression des données, celles-ci sont identiques à celles utilisées pour la compression.

Ces algorithmes de compactage sont utilisés pour compresser tous types de données : des données textuelles, des images, du son, des programmes, etc.

Les trois principales catégories d'algorithme sans perte sont :

– Les Algorithmes à base de dictionnaire :

Découpent les données en « mot » qui sont mis dans un dictionnaire, si le mot est déjà dans le dictionnaire on le remplace par son indexe si non on l'ajoute dans le dictionnaire, on espère que tous les mots finiront dans le dictionnaire pour aboutir à une compression maximale.

– Les Algorithmes à base de transformation :

Appliqueront plutôt une transformation sur les données qui mettra en évidence les répétitions, de façon à pouvoir les mieux exploiter pour la compression. Ces algorithmes sont souvent suivis d'une autre tape de compression comme méthode à base de dictionnaire ou statistique.

– Les Algorithmes à base modélisation statistique :

Calculent des probabilités pour aider à la compression, typiquement, on tentera d'estimer le plus précisément possible la fonction de distribution de la prochaine donnée, de façon à pouvoir générer un code efficace pour cette prochaine donnée, en théorie ils peuvent mener à une compression optimale.

#### 1.6.4.6.2 La compression avec pertes (destructive)

Il est question de compression « avec pertes » lorsque certaines informations sont volontairement supprimées, par exemple les détails d'une image, ou les fréquences inaudibles pour un fichier sonore. Certains éléments sont quasiment imperceptibles par l'humain (par exemple les très basses ou hautes fréquences audio), il est alors intéressant de profiter de cette « faiblesse » pour réduire la taille du fichier. la qualité finale du média dépendra donc de la quantité de données altérée, le but étant de trouver un bon compromis entre « qualité » et « taux de compression ».

#### Exemple

Un algorithme avec perte pourra par exemple d'éliminer le bruit d'un signal et n'encoder que le signal. Nous avons une perte parce que le bruit contient aussi de l'information. Mais puisque c'est une information qui peut détruire le signal alors sa suppression apporte un avantage.

#### Remarque

Les algorithmes de compression destructrice sont utilisés pour compresser des données graphiques, audio et vidéo. Ils ne pourraient être appliqués sur des données textuelles ou sur celles d'un programme au risque de les rendre illisibles ou inexécutables.

## 1.7 Topologie de données à compresser

### 1.7.1 Le texte

De manière générale, toute information structurée en ASCII ou composée de caractères alphanumérique. Dans un traitement de texte et en PAO (publication assistée par ordinateur), partie principale d'un document, par opposition aux titres, tableaux, figures, notes de bas de page et autres éléments.

### 1.7.2 Le son

Un son peut être défini comme une vibration générée mécaniquement, transmise généralement par l'air sous forme d'ondes qui aboutissent au tympan de l'oreille, avant d'être interprété par le cerveau c'est donc un signal que l'on peut représenter sous forme d'une courbe mathématique indiquant l'intensité en fonction du temps qui s'appelle un signal analogique ce signal doit encore être numérisé pour pouvoir être exploité par l'ordinateur pour ce la, il est échantillonné, c'est-à-dire découpé dans le temps, par une carte son.

### 1.7.3 L'image

C'est une représentation plane d'un objet tridimensionnel, perçue par l'œil humain, ou plus généralement un capteur dont le fonctionnement est semblable (exemple d'objet : une scène, un portrait, une échographie, une observation astronomique...). Elle peut être décrite sous la forme d'une fonction  $I(x, y)$  de brillance analogique continue définie dans un domaine borné, tel que  $x$  et  $y$  sont les coordonnées spatiales d'un point de l'image et  $i$  une fonction d'intensité lumineuse ou de couleur.

#### 1.7.3.1 La définition

Pour une image numérisée, on utilise une autre notion qui est la définition, laquelle peut s'exprimer par plusieurs valeurs, soit le nombre de point par ligne soit le nombre de lignes, soit le produit des deux.

#### 1.7.3.2 La numérisation

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représenté par une matrice bidirectionnelle de valeur numérique  $f(x, y)$  Ou :  $x, y$  : coordonnées cartésiennes d'un point d'image.

### 1.7.3.2.1 Modes de numérisation (destructive)

#### Mode de niveau gris

Dans ce mode la valeur du rouge ( $R$ ), du vert ( $V$ ) et du bleu ( $B$ ) est la même, donc codé sur  $n = 8bits$ , nous pouvons donc définir 256 niveaux de gris, allant du niveau 0 (noir) au niveau 255 (blanc).

#### Mode couleur RGB

RGB (Red-Green-Blue) ce triplet permet de définir une couleur : chaque couleur peut prendre une valeur de 0 à 255, indiquant son intensité. Par exemple, un triplet (255, 0, 0) représente la couleur rouge foncé. De même, (0, 0, 0) représente le noir et (255, 255, 255) le blanc. C'est ce que on appelle la synthèse additive (rouge+vert+bleu) à la quelle notre œil est sensible, (256) couleurs, soit pratiquement ce que l'œil humaine peut voir. Cette échelle additive s'imagine facilement en ajoutant des couleurs à la couleur noire. Donc plus on ajout de couleurs ( $R$ ,  $V$ ,  $B$ ), plus on se rapproche du blanc. Elle est utilisée pour des écrans.

Elle s'oppose à l'échelle soustractive, où l'on soustrait des couleurs à la couleur blanc (CMYK : Cyan, Magenta, Yellow, black ; soit CMJN en français). Cette quadrichromie est le principe utilisé par les imprimantes couleur, la photographie. On trouve également des autres modèles, comme le YUV (luminance et chrominance), utilise pour les téléviseurs (et le format JPEG), et HSV (ou HSB) : Hue, saturation, Brightness.

## 1.7.4 La vidéo

Du latin vidéo « je vois » technique de transformation d'images animées en signaux électriques (signaux vidéo), destinée à permettre leur diffusion ou leur enregistrement. Les premiers pas en terme de compression ont été fait à partir du JPEG et on donné naissance au M-PEGE (Moving JPEG), qui consiste à compresser chaque image au format JPEG. Ceci a amené à coder par JPEG la première image, et à coder les différences entre chaque pair d'images successives par JPEG également.

## 1.8 Les formats de fichiers de données

### 1.8.1 Les formats graphiques

#### BMP (BitMap)

Le format BMP est le format par défaut du logiciel Windows. C'est un format matriciel.

Les fichiers BMP sont constitués d'une image de type Bitmap. Ce format a été créé par Microsoft.

Ce format existe sous les environnements Microsoft Windows et OS/2, sous différentes versions.

La taille maximale d'une image est de  $64k * 64k$  (en point).

La couleur est codée sur 1 bit, 8 bit. L'image peut donc avoir 2, 16, 256 ou 16 millions de couleurs. Les données peuvent être enregistrées telles quelles ou compressées suivant l'algorithme RLE.

### **GIF (Graphical Interchange Format)**

Le format GIF est un format-propritaire qui a ouvert la voie à l'image sur le Word Wide Web.

Conçu à l'origine par la compagnie H&R block, la renommée du format GIF est due au réseau COMPUSERVE. C'est un format de compression qui n'accepte que les images en couleurs indexées, codé sur 8 bits, soit en 256 couleurs.

Les images RGB(RVB) ou CYMK(CMJN) en milliers de couleurs doivent d'abord être converties en 256 couleurs avant d'être exportées en format GIF. Le format GIF est un format qui à l'origine n'existait que dans le monde des IBM/compatibles.

Toute fois, depuis quelques années, plusieurs logiciels Macintosh sont capables d'importer, de modifier et d'exporter des images en format GIF.

Le format GIF ne peut toutefois pas être utilisé par les logiciels de mise en page.

C'est un format qui présentement perd beaucoup de son marché suite à une bataille juridique concernant les droits d'utilisation sur internet.

### **JPEG (Joint Photographie Experts Group)**

Le format JPEG est un format de compression des images graphiques.

Ce format accepte les images RGB (RVB) et CYMK (CMJN). Il est sans doute le monde de compression le plus efficace qui soit, avec un bon compromis entre gain d'espace disque.

Temps de compression et décompression et qualité des images.

Ainsi une image brute de 2M0. N'occupera après conversion en JPEG que 400Ko, selon la qualité d'image voulue.

L'inconvénient des images JPEG, c'est qu'elles ne peuvent être importées directement dans un logiciel de mise en page.

Le format JPEG est un des formats les plus utilisés dans le monde du Word Wide Web. il peut être lu directement par les principaux outils de navigation pour le WWW.

### **JPEG 2000**

JPEG de l'avenir, ou le format qui réunit les avantages du GIF et du JPEG. Reposant sur la compression d'images par ondelettes, le JPEG est très attendu,

il propose une compression avec ou sans perte, et tente d'unifier la méthode de décompression pour qu'un programme puisse ouvrir n'importe quel fichier JPEG (on en dénombre aujourd'hui plus de quarante variantes!).

Ce standard pourra faire appel à un procédé pour comprimer les zones réservées aux textes, et à un autre pour les zones photographiques.

## **PNG**

Le format PNG (Portable Network Graphics) est un format matriciel destiné à remplacer progressivement le format GIF sur Internet.

Ce dernier offre tous les avantages du format GIF (entrelacé, dégradation minime de l'image compressée, etc.) mais avec un meilleur taux de compression.

Il accepte les images en couleurs en 24 bits (millions de couleurs), contrairement au format GIF limité à 256 couleurs.

Bien qu'étant très peu répandus, certains logiciels de retouches de photographies permettent la conversion en format PNG. Citons à titre d'exemple adobe Photoshop (Macintosh et Windows) et GraphicConverter (Macintosh), Macromedia Fireworks.

## **1.8.2 Format vidéo**

### **AVI (Audio Vidéo Interleaved)**

Format de fichier utilisé par Windows pour le stockage des images vidéo et du son, dans lequel les données audio, accélérant ainsi la vitesse de restitution. Dans ce format, on dit que l'image et le son sont entrelacés.

### **MPEG 1**

Destinée aux vidéos de faible résolution (325\*288\*25 images/s) et une qualité de son CD. Le débit est maximum. 1.15 Mbits/s. Elle convient pour les applications multimédias.

### **MPEG 2**

Pour la vidéo numérique de qualité TV (800\*600\*25 images/s). Le débit est compris entre 4 et 10 Mbits/s. la qualité supérieure à la précédente.

### **MPEG 4**

Prévu : pour les applications avec un très faible débit (4800 à 64000 bits/s) et destiné aux communications mobiles comme la visioconférence. La résolution est aussi, très faible (170\*144\*10 images/s).

### **MOV (QuickTime)**

Il est aussi basé sur l'utilisation de 5 codecs : le codec graphisme et photo/JPEG, le codec animation ainsi que les codecs Apple vidéo compacte. Lorsqu'on souhaite

visualiser des fichiers d'extension AVI ou MOV sur un ordinateur, il faut que le codec correspondant soit installé.

### 1.8.3 Format audio

#### WAV

Le format WAV (Waveform audio File Format) est l'équivalent audio du format BMP. Egalement développé par IBM, il reprend le principe de la compression minimale. Pour simplifier, on peut dire que le WAV encode directement le son numérisé, sans aucune forme de compression, ce qui tend à produire des fichiers de taille conséquente. Pour cette raison, les fichiers WAV sont uniquement destinés aux sons très courts (tels que les jingles de Windows par exemple). Pour les fichiers musicaux plus long, on lui préfère indéniablement le format MP3.

#### MP3

Le MP3 « MPEG » Audio layer 3 » est un format de compression de données audio par filtrage des données audio. Ce format permet de compresser à un taux de 1/12 les formats audio habituels (WAV ou CD audio). Il permet de faire tenir l'équivalent en fichiers de douze albums de musique sur un seul CD-ROM. De plus, le format MP3 n'altère que faiblement le son pour l'oreille humaine.

#### OGG

Le format OGG (Ogg Vorbis) est une alternative au format MP3 qui est censée produire des fichiers de bien meilleure qualité pour une taille toutefois légèrement supérieur. Réclamé par les audiophiles qui jugent la compression MP3 trop drastique, il est développé en tant que logiciel libre, à l'inverse des formats MP3, AAC et WMA.

#### AAC

Le format ACC, développé par le Moving Picture expert Group, a pour but de remplacer le MP3, prévu originellement pour accompagner les vidéos MPEG-1.

Celui-ci gère le son des fichiers MPEG-4 et doit offrir une qualité bien meilleure que le MP3 et gérer plus de canaux.

## 1.9 Evaluation des performances d'une compression

On ce moment il existe beaucoup de méthodes de compression de données, adaptées a des structures de données différentes. Pour mesurer leurs efficacités et leurs performances, il faut s'intéresser sur des critères qui sont :

- Le taux de compression
- Le temps de compression/décompression

- Le gain de compression
- Le taux d'information
- Mesure de fidélité la subjectivité des experts du domaine
- Types de données (image, vidéo, texte)
- Mesure de perte d'information

Le domaine de compression des données (médicales, géologie, météorologie)

## 1.9.1 Le taux de compression

### 1.9.1.1 Le degré de compression

Le degré de compression des données se mesure par la comparaison de la taille des données à compresser à la taille des données initiales. On la nomme aussi : Quotient de compression donnée par la formule suivante :

$$\text{Quotient de compression} = \frac{\text{Taille des données initiales}}{\text{Taille des données initiales}}$$

#### Remarque

Plus le Quotient n'est élevé, plus la méthode de compression employer est efficace donc le taux de compression est :

$$\text{Taux de compression} = \frac{1}{\text{Quotient de compression}} * 100\%$$

## 1.9.2 Le gain de compression

Le gain de compression représenté en %, c'est l'espace dont on a bénéficié après une opération de compression telle que plus le gain est important, plus la compression est efficace, il est donnée par la formule suivante :

$$\text{Gain de compression} = 100 * \text{taux de compression}$$

## 1.9.3 Le Taux d'information

Qui peut servir à noter la différence entre la qualité et la quantité d'information.

## 1.9.4 Mesure de fidélité

C'est la distorsion introduite lors des deux processus de compression / décompression par rapport à un fichier source.

### **1.9.5 La vitesse de compression**

C'est le temps mis par l'opération compression / décompression sur le flux de donnée. Il y a plusieurs facteurs qui permettent de juger la rapidité d'un compresseur.

### **1.9.6 Mesure de perte d'information**

C'est un critère d'évaluation des performances d'un compresseur/décompresseur de donnée qui ne peut restituer, les données originales aux quelles il a fait subir un processus de compression.

## **1.10 Conclusion**

Dans ce chapitre nous avons abordé les différentes notions et intérêt de la compression de données. Nous avons vu qu'il existait plusieurs manières de coder, de représenter, de compresser les différents médias .ensuite nous y avons décrit les divers types de la compression de données à savoir la compression sans perte et la compression avec perte. Enfin nous avons exposé quelques mesures de l'évaluation de la performance d'une compression le taux, la vitesse, et le gain de compression ainsi la mesure de perte d'information.

Dans le chapitre qui suit nous allons traiter les méthodes de la compression de données.

## Chapitre 2

# Etat de l'art de la compression de donnée par la transformée

### 2.1 Introduction

Les ondelettes ont généré dans les dernières années un grand intérêt dans le domaine théorique et également dans le domaine pratique. La théorie des ondelettes est issue de nombreux travaux en traitement du signal et en compression d'images.

En traitement du signal, la théorie des bancs de filtres a donné le schéma de décomposition reconstruction de Stephane Mallat [2.1], en compression d'images, l'algorithme de décomposition pyramidal d'une image a servi de base pour l'analyse multi résolution.

Dans ce chapitre nous allons voir les limites de la transformée de Fourier.

Ainsi que la théorie des ondelettes ; nous évoqueront différents types d'ondelettes et la généralisation sur les images en deux et trois dimensions.

### 2.2 Définition des ondelettes

Les ondelettes sont des fonctions élémentaires très particulières ; ce sont des vibrations très courtes [2.2]. Mathématiquement, le but essentiel de l'analyse par ondelettes est de décomposer les espaces fonctionnels usuels sur des bases ayant d'excellentes propriétés tel que l'orthogonalité et la régularité.

La transformée en ondelettes est une méthode de représentation temps fréquence d'un signal qui consiste à le décomposer en une somme de fonctions élémentaires qui dérivent toutes d'une même fonction appelée mère, par translation, contraction

et dilatation.

## 2.3 Transformée de Fourier

Les séries de Fourier sont utilisées pour l'analyse des signaux périodiques. Pour les signaux non périodiques on a recours à une intégrale de Fourier.

Cette méthode consiste à représenter le signal étudié par une superposition d'ondes sinusoïdales de toutes les fréquences possibles.

Les amplitudes associées à chaque fréquence représentent les importances respectives des diverses ondes sinusoïdales dans le signal global.

Ces amplitudes forment alors une fonction de la fréquence  $f$  appelée "spectre continu des fréquences du signal" : c'est la transformée de Fourier du signal  $s(t)$ , notée  $S(f)$  :

$$Sf = s(t)dt$$

(Le nombre complexe  $S(f)$  s'identifie ; pour une fréquence  $f$  donnée, à un point du plan).

L'analyse du signal consiste à dégager des informations contenues dans celui-ci. Ceci en fonction d'une seule variable (temps) ou de deux variables (temps et fréquence). Dans ce dernier cas, L'analyse de Fourier classique est inadéquate, car la représentation d'un signal  $f$  par l'intermédiaire de sa transformée de Fourier  $S(f)$  ne fournit qu'une information globale sur le signal.

L'évolution dans le temps des composantes fréquentielles du signal n'est pas directement accessible par cette représentation. C'est pour atteindre cette information que la représentation temps fréquence a été créé.

### 2.3.1 Transformée de Fourier à fenêtre glissante

représentation temps fréquence met en jeu deux opérations réciproques : analyse et la synthèse.

Pour effectuer l'analyse du signal, on le décompose en une somme de fonctions élémentaires  $\theta_a, b$  (fonctions sinusoïdales pour J'analyse de Fourier) où 'a' est lié à la fréquence et 'b' est lié au temps. Pour décomposer un signal quelconque on affecte à chaque fonction élémentaire des coefficients  $C_{a,b}$  ou :

$$C_{a,b} = S(t)\theta_a * b(t) dt$$

Ces coefficients donnent une information directe sur les propriétés temporelles et fréquentielles du signal. Les fonctions  $\phi_{a,b}$  doivent être bien localisées dans le temps, de sorte que les coefficients  $C_{a,b}$  dépendent seulement des valeurs que prend le signal dans l'intervalle de temps sur lequel la fonction  $\phi_{a,b}$  n'est pas négligeable.

La synthèse donne les règles permettant de reconstruire un signal à partir des éléments  $C_{a,b}$  fournis par l'analyse.

En 1940, D.Gabor [2.4] découvre la première forme de représentation temps fréquence. Il obtient une analyse temporelle en découpant arbitrairement le signal en des plages de longueur limitée. Chaque plage, centrée autour du paramètre 'b' de localisation en temps, est alors étudiée séparément des autres par l'analyse traditionnelle de Fourier, ce qui revient à décomposer le signal sur des fonctions élémentaires  $\phi_{a,b}$  qui dérivent toutes d'une même "fonction fenêtre"  $\phi(t)$  par translation et modulation en temps. C'est la transformation de Fourier à fenêtre glissante.

L'inconvénient majeur de ce procédé est que la longueur de la plage est fixée une fois pour toutes et que l'on ne peut pas analyser simultanément des phénomènes dont les échelles de temps sont différentes. Ce problème est résolu par l'analyse multi échelle par ondelettes où il y a des familles d'ondelettes qui correspondent à des décompositions différentes. Elles ont des propriétés différentes et permettent ainsi des analyses différentes. L'analyse par ondelettes est une méthode mathématique pour représenter le signal.

## 2.4 Transformée en ondelettes

J. Morlet [2.5] a construit une famille d'ondelettes engendrée par une seule ondelette  $\phi(t)$  dite analysante.

Il a construit les ondelettes  $\phi_{a,b}$  à partir de l'ondelette analysante  $\phi$ , non pas par translation et modulation, comme pour la transformée de Fourier à fenêtre glissante, mais par translation en temps (paramètre 'b') et contraction ou dilatation en temps selon que le paramètre 'a' (fréquence) est plus petit ou plus grand que un. Il suffit donc de "jouer à l'accordéon" avec l'ondelette analysante  $\phi$  pour obtenir la famille des ondelettes  $\phi_{a,b}$ . La transformée en ondelettes réalise une analyse à toutes les échelles. Elle est une fonction  $S(a,b)$  qui associe aux paramètres 'a' et 'b' la valeur du coefficient  $C_{a,b}$  de l'ondelette  $\phi_{a,b}$  dans la décomposition du signal. La quantité 'b' est le paramètre de localisation temporelle, tandis que  $(1/a)$  est le paramètre de fréquence.  $C_{a,b}$  est une intégrale qui mesure la somme des aires algébriques décrites par la courbe (produit  $s(t)\phi_{a,b}$ ).

### 2.4.1 Transformation continue en ondelettes

La transformation continue en ondelette, ou transformation intégrale en ondelette a la possibilité de faire un "zoom", c'est à dire que la dimension de la cellule de Gibbs (voir la figure 2.2) peut s'adapter à la position du centre dans le plan (t, w), et devient plus étroite si l'on se déplace vers les hautes fréquences, et plus large vers les basses fréquences [2.6].

Une expansion d'ondelette utilise des transformations et des dilatations d'une fonction fixe, l'ondelette  $\phi$ , Dans le cas de la transformation continue en ondelettes, les paramètres de translation et de dilatation varient continuellement .

Alors, la transformation continue en ondelettes définie par :

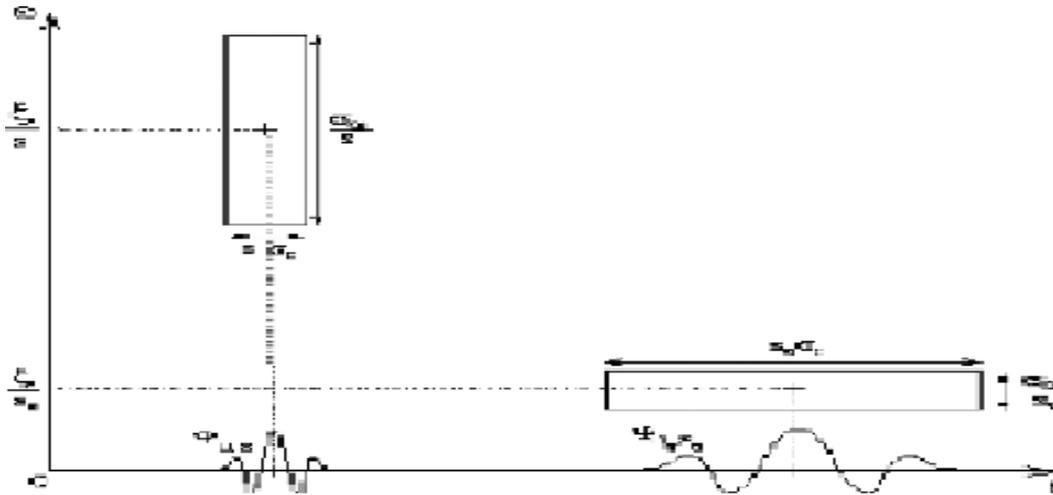
$$W(a, b) = \langle f\theta(a, b) \rangle \quad 2.5$$

(a) Ondelette de morlet $\phi_{a, b}$ de fréquence $1/a$ centrée en $b$ .
(b) signal $S(t)$ .
(c) produit $S(t) \phi_{a, b}$ .
(d) mesure du coefficient $C_{a, b}$ représenté par L'intégrale « aire » du signal produit.

Pour pouvoir définir correctement la transformation en ondelettes il faut que  $\phi$  possède les propriétés suivantes : 1.  $\phi$  est aussi une fonction fenêtre, dont le centre est  $w_0 > 0$ , pour localiser les fréquences en utilisant l'effet de "zoom".

1.  $\phi$  est aussi une fonction fenêtre, dont le centre est  $w_0 > 0$ , pour localiser les fréquences en utilisant l'effet de "zoom".
2. pour permettre la reconstruction de  $f$  à partir des valeurs ( ) de la transformée par ondelette  $W(a, b)$ .

La transformation continue en ondelettes nous permet d'utiliser des ondelettes plus générales. Et elle est utilisée dans la détection de la singularité et dans l'interprétation (caractérisation).



### 2.4.2 Transformation discrète en ondelettes

Cette approche [2.7] peut être utilisée pour traiter des images. Nous supposons que les données  $\{f_i(x)\}$  sont les produits scalaires en un pixel 'i' de la fonction  $f(x)$  et d'une fonction d'échelle donnée  $\phi(x)$  :  $f_i(x) = \langle f(x), \phi(x - x_i) \rangle$ . Où  $\phi(x)$  doit satisfaire l'équation de dilatation .

C'est l'ensemble discret associé a la transformation en ondelettes correspondant à  $\phi(x)$  L'ondelette associée  $\psi(x)$  est :  $\psi(x) = \phi(x/2) - \phi(x/4)$

Filtrage avec un facteur décroissant de distance deux entre les échantillons La distance entre deux échantillons croit par un facteur de deux de l'échelle  $(n-1)$  à la suivante,  $(k)$

### 2.4.3 Familles d'ondelettes

#### 2.4.3.1 Ondelettes orthogonales

Soit  $\{V_n; n \in \mathbb{Z}\}$  une analyse multirésolution, engendrée par une fonction  $\phi$ , et soit  $\psi$  une ondelette associée, qui engendre les sous espaces complémentaires  $W_n$ ,  $n \in \mathbb{Z}$ . L'ondelette  $\psi$  est orthogonale si et seulement si : pour  $k \neq l$ , et pour tout  $j \in \mathbb{Z}$

Dans ce cas  $W_n$ ,  $n \in \mathbb{Z}$  sont des espaces complémentaires orthogonaux. Cependant, en général nous n'avons pas  $\psi_k$ ,  $j$  orthogonal à  $\psi_l$ ,  $j$ .

### 2.4.3.2 Ondelettes biorthogonales

Il existe une fonction d'échelle duale  $\tilde{\phi}$  et une ondelette duale  $\tilde{\psi}$ , qui génèrent une analyse multirésolution duale avec les sous espaces  $V_j$ , et  $W_j$

### 2.4.3.3 Ondelettes usuelles

Un exemple très simple mais très utile pour illustrer les meilleures propriétés des ondelettes, est l'ondelette de Haar, où l'on peut illustrer facilement les propriétés de la fonction d'échelle et de l'ondelette.

Cette ondelette a aussi des utilisations pratiques. Fonction d'échelle, elle est définie par :  $\phi(x) = 1$  si  $0 \leq x < 1$  et  $\phi(x) = 0$  ailleurs. Le sous espace  $V_0$  est étendu par la fonction d'échelle  $\phi(x - k)$ , qui est formée de translations entières de la même fonction.

Les sous espace  $V_1$  est étendu par  $\phi(2x - k)$  qui est formée de translations de  $k/2$  de la fonction d'échelle sur un intervalle de  $1/2$ . En général,  $V_j$  est étendu par des translations de  $j$

Les relations de reconstruction sont données par les relations de décomposition sont l'inverse des celles de reconstruction et il sont données par :  $\sum_k \langle f, \psi_{j,k} \rangle \phi_{j,k}(x) = f(x)$ . Il existe plusieurs autres ondelettes usuelles telles que : l'ondelette de Daubechies.

### 2.4.3.4 Ondelettes sur un intervalle

Les ondelettes définies sur un intervalle le sont sur un ensemble compact, tel qu'un intervalle unidimensionnel ou bidimensionnel. Pour être spécifique, nous pouvons considérer le cas d'une fonction  $f(x)$  définie sur un intervalle  $[a, b]$  et telle que  $f(x)$  est nulle à l'extérieure de  $[a, b]$ . Les ondelettes comportent des discontinuités aux points d'extrémités  $a$  et  $b$ , et sont efficaces pour détecter les singularités.

Pour construire des ondelettes bornées sur un intervalle, considérons un ensemble fini de fonctions linéaires et indépendantes  $\phi_0, \dots, \phi_{m-1}$  définies sur cet intervalle et supposons  $V_0$  l'espace vectoriel étendu par ces fonctions, en observant que pour une fonction d'échelle  $\phi$  on considère un ensemble fini de fonctions d'échelle.

- Comme exemple des ondelettes sur un intervalle, citons parmi d'autres :
- L'ondelette linéaire et quadratique de Legendre, l'ondelette de premier ordre et de second ordre de Flatlet et l'ondelette borne de Meyer

### 2.4.3.5 Ondelettes multidimensionnelles

Une méthode simple pour obtenir des ondelettes multidimensionnelles est d'utiliser le produit tensoriel.

Les fonctions  $\phi_{i,l} \psi_{j,k}$  impliquent deux échelles,  $-i/2$  et  $-j/2$ , et chacune est supportée sur un rectangle. Pour cela cette décomposition est appelée décomposition rectangulaire d'ondelette.

## 2.5 Bancs de filtres

### 2.5.1 Notion des bancs de filtres

La théorie des ondelettes trouve ses fondements dans la théorie des bancs de filtres couramment utilisée en traitement du signal et en télécommunication. L'idée est de séparer le signal original en plusieurs bandes de fréquence (basse fréquence et haute fréquence) pour mieux le traiter et le transmettre. Au récepteur, on reconstruit le signal en rassemblant ces diverses bandes. Le problème est de savoir comment on peut avoir un signal reconstruit  $X$  identique au signal original  $X_0$ . Banc de filtres (banc d'analyse/synthèse à un étage).

On rappelle qu'une décimation par  $M$  implique (notation avec la transformée en  $z$ )

Dans le cas de filtres orthogonaux, l'énergie des coefficients transmis est la même que celle du signal original. Tandis que dans le cas où les filtres sont biorthogonaux, ce n'est pas le cas. Par contre, dans les deux cas, on reconstruit un signal identique à l'original mais avec une phase pouvant être différente. Il est bien connu [2.10; 2.11] que le seul banc de filtres RIF réels ayant une phase linéaire est celui avec des filtres de Haar.

### 2.5.2 Réalisation d'un banc de filtres à base d'ondelettes

#### Design d'un banc de filtres à deux étages

Un banc de filtres est un ensemble de filtres reliés entre eux par des décimateurs et des interpolateurs. Pour un banc de filtres à deux canaux, les filtres analysants sont normalement un passe-bas et un passe-haut.

Un banc de filtres à reconstruction parfaite décompose un signal par filtrages et sous échantillonnages. Il le reconstruit par insertions de zéros, filtrages et sommation.

Un banc de filtres (discrets) sous échantillonnés à deux canaux convolue un signal  $X_0(n)$  avec un filtre passe-bas  $h_a(n) = h(-n)$  et un filtre passe-haut  $g_a(n) = g(-n)$  et sous-échantillonné par deux les sorties.

On dit qu'on a un banc de filtres à reconstruction parfaite quand  $X_0(n) = X(n)$ . Lorsqu'en plus  $h_a = h_s$  et  $g_a = g_s$ , on parle de filtres miroirs conjugués.

## 2.6 Compression d'image par transformée en ondelettes

### 2.6.1 Compression en deux dimensions

La théorie des ondelettes peut être généralisée, en plusieurs dimensions. Nous étudierons les ondelettes bidimensionnelles et ces applications sur l'image.

Chaque sous-espace correspond à un produit tensoriel de deux espaces identiques suivant la formule :  $V_m(x, y) = V_m(x) \otimes V_m(y)$  2.28 La fonction d'échelle bidimensionnelle est alors un produit tensoriel de deux fonctions d'échelle monodimensionnelles :  $\hat{O}(x, y) = \hat{O}(x) \hat{O}(y)$  . 2.29 Où  $\hat{O}(x)$  est la fonction d'échelle monodimensionnelle.

L'approximation d'un signal bidimensionnel  $\hat{E}(x, y)$  à la résolution  $2^{-m}$ .

Comme dans le cas monodimensionnel, le détail est obtenu en projetant le signal  $f(x, y)$  sur un espace complémentaire  $W_m$ .

Une base de cet espace complémentaire peut être obtenue par translation et dilation d'une fonction d'ondelette. Soit  $\phi(x)$  l'ondelette associée à  $\hat{O}(x)$  on peut alors définir les trois ondelettes bidimensionnelles.

La différence entre deux approximations successives caractérisées par les trois coefficients d'ondelette représentant les détails .

Le calcul d'une image  $S_m(n_x, n_y)$  à une résolution inférieure et les calculs des coefficients d'ondelettes  $\{ (n_x, n_y), d \in \{ H, V, D \} \}$  se font par convolution en utilisant des filtres séparables 2D. Le filtrage 1D défini pour les signaux monodimensionnels est appliqué indépendamment sur les lignes et les colonnes, nous présentons d'une façon générale dans les figures 2.6-2.7 le principe de décomposition et de reconstruction dans le cas

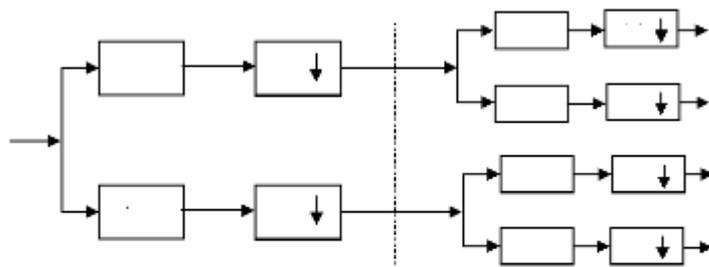


FIGURE 2.6.1 : Un étage de décomposition multi-résolution bidimensionnelle

Donc, à partir d'image initiale à la 3eme résolution, on obtient quatre sous image (voir la figure 2.7 (a) .Après, on fait la décomposition sur trois niveaux du résolution et la figure (b) représente un exemple de décomposition d'image sur trois niveaux de résolution  $m=3, m=2, m=1$ .

## 2.6.2 Compression en trois dimensions

d'un produit tensoriel d'une analyse multirésolution [2.8 ; 2.1] à une dimension  $V_0 = V \otimes V_0$ , ou  $V_j, j \in \mathbb{Z}$  est une multirésolution de  $(\cdot) L^2 \mathbb{R}$ .

La multirésolution est similaire à celle d'une seule dimension, elle est comme suit : Est une base orthonormale de  $V_0$ ; la base  $V_j$  est obtenue comme [2.1].

Le complément orthogonal dans  $V_{j-1}$  pour  $V_j$  est  $W_j : V \otimes V \otimes V \otimes W \otimes V \otimes W$ .

Et les,  $W_j$  dépend de trois parties, qui sont des bases de  $\emptyset$ , ces des combinaisons a une dimension de la fonction d'échelle  $\tilde{\phi}$  et la fonction d'ondelette  $\phi : k \in L^2$  l'ensemble  $\{ \phi_n ; n \in \mathbb{Z} \}$ ,  $\{ \tilde{\phi}_n ; n \in \mathbb{Z} \}$ ,  $\{ \phi, \tilde{\phi} \}$  est une base orthogonale de  $L^2(\mathbb{R}^2)$ .

Dans cette construction, l'échantillonnage se fait séparément : verticalement et horizontalement, mais les bases d'ondelettes sont non separable. La transformation en ondelette rapide en deux dimension est obtenue en utilisons des opérations de filtrage dans les directions horizontal et vertical de l'image. L'image originale est filtrée en quadrants et ensuite le quadrant d'approximation est filtré lui aussi. Si la taille de l'image originale en  $N \times N$  alors chaque quadrant est de taille  $N / 2 \times N$ .

1. La transformation a la propriété de reconstruction parfaite. Une approche similaire à celle de la transformation en deux dimensions est prévue. Le cas de trois dimensions est appliquée par exemple pour des images médicales, l'analyse multirésolution donne la configuration suivante :
2. Le filtrage des l'images est fait en utilisant une fonction d'échelle et sept ondelettes, qui sont Où toutes les dimensions sont dilatées de la même manière et l'échantillonnage se fait séparément le long de chaque dimension de l'image 3D. Si l'image originale est de taille  $N * N * N$  alors celle filtrée sera de taille  $N / 2 * N / 2 * N / 2$  comme l'illustre.

## 2.7 Conclusion

La transformée en ondelettes a été étudiée en ses débuts par des physiciens théoriciens dans le domaine de mécanique quantique. Plus récemment elle a été utilisée en théorie constructive des champs. La transformée en ondelettes en mode continu coïncide avec la notion d'analyse multirésolution développée par les chercheurs en vision. La transformation en ondelettes constitue un puissant outil d'analyse.

Une de ses premières applications a été en séismologie. Cette transformation a été ensuite appliquée à l'analyse des sons, des images, et de toutes sortes de signaux.

Du fait que les ondelettes servent à analyser des phénomènes qui se produisent simultanément à des échelles différentes, elles constituent un outil naturel pour étudier les objets de type fractal, où les objets qui demeurent semblables à eux-mêmes lorsqu'on les considère à des échelles différentes.

L'analyse et la synthèse par ondelettes permettent d'analyser efficacement des signaux où se combinent des phénomènes d'échelles très différentes.

L'idée de l'analyse par ondelette est de décomposer un signal sur une base de fonctions d'un sous espace ayant des propriétés bien déterminée. En particulier, on peut chercher à analyser un signal en tentant de localiser dans le temps les irrégularités du signal c'est-à-dire les variations brusques dans le signal qui correspondent aux hautes fréquences.

Schématiquement, la transformation en ondelettes revient à effectuer une série de filtrage passe-bande dans l'espace réciproque (ou plan de Fourier de l'objet), et pour chaque échelle, à reconstituer l'image de l'objet après filtrage.

La mise en oeuvre de la transformation en ondelettes sur ordinateur ne présente pas de difficultés techniques importantes. Toutefois sa rapidité est fortement liée à la résolution utilisée.



# Partie pratique

"A force de croire en ses rêves,  
l'homme en fait une réalité."

---

*(Georges Rémi)*



# Chapitre 3

## Codage EZW

### 3.1 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 : c'est le EZW.

La méthode de codage progressif connue sous le nom de Embedded Zerotree Wavelet coding (EZW), proposée par Shapiro, est une méthode simple et très efficace de compression d'image par ondelettes. Elle a démontré sa puissance dans les deux formes de compression (avec et sans perte d'informations) depuis son élaboration en 1993. Plusieurs variantes de ce type de codage ont été proposées par différents chercheurs dans le domaine, ce qui fait sa force. On peut citer par exemple le SPIHT (Set Partitioning in Hierarchical Tree) réalisé par A. SAID et W. PEARLMAN qui est la variante la plus populaire de l'EZW. L'intérêt de consacrer ce chapitre à cette méthode est de la mettre en évidence vu qu'elle sera à la base des algorithmes proposés dans notre travail.

### 3.2 Présentation du codage EZW

En général, dans une représentation d'image par coefficients d'ondelettes, l'image obtenue est organisée de façon à représenter les principaux traits de l'image dans les bandes de basse fréquence, puis les détails dans les bandes de haute fréquence. Le principe de l'EZW s'appuie sur cette représentation, pour coder les coefficients

d'une manière progressive. Ainsi, on commence par les basses fréquences L, ensuite on code les détails (hautes fréquences), l'avantage de cet algorithme, est que l'on a en tout temps un niveau de compression et que l'on peut arrêter en tout moment le codage.

Le codage EZW est basé sur deux principales observations :

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

Ces deux observations sont exploitées en codant les coefficients d'ondelettes par ordre décroissant, dans plusieurs passages. Pour chaque passage on choisit un seuil par rapport auquel tous les coefficients d'ondelettes sont comparés. Si un coefficient d'ondelette est supérieur au seuil, il est codé et retiré de l'image, sinon, il est laissé pour le prochain passage. Quand tous les coefficients d'ondelettes ont été examinés, le seuil est abaissé et l'image est rebalayée pour ajouter plus de détails à l'image déjà codée. Ce processus est répété jusqu'à ce que tous les coefficients d'ondelettes soient encodés complètement ou qu'un autre critère soit satisfait (PSNR choisit est atteint) selon le mode de compression utilisé.

L'algorithme emploie la dépendance entre les coefficients d'ondelettes à travers différentes échelles pour coder efficacement les grandes parties de l'image qui sont au dessous du seuil actuel, c'est ici qu'intervient le zero tree.

Dans la représentation de « l'image ondelettes », chaque coefficient peut être considéré en tant qu'ayant quatre descendants dans la prochaine plus haute sous-bande. Ainsi que pour les quatre descendants, chacun à également quatre fils dans la prochaine plus haute sous-bande et nous voyons un arbre à quatre descendants émerger ; chaque racine a quatre branches, ce processus se poursuit jusqu'aux fréquences les plus hautes.

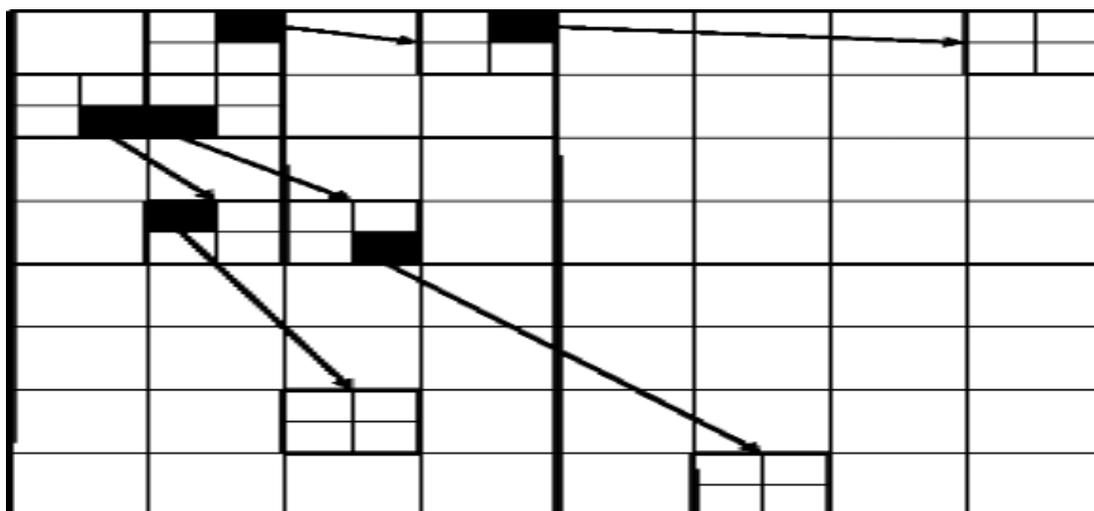


FIGURE 3.2.1 : Représentation de l'organisation en arbre des coefficients d'ondelettes

Nous pouvons maintenant donner une définition du zerotree. Un zerotree est un quadruple-arbre dont tous les noeuds enfants sont égaux ou plus petits que les noeuds parents. L'arbre est codé avec un symbole unique et reconstruit par le décodeur comme quadruple-arbre rempli de zéros. Nous devons insister sur le fait que la racine doit être plus petite que le seuil par rapport auquel les coefficients d'ondelettes sont comparés, sinon, ce coefficient ne serait pas considéré comme base de zero tree.

Le codeur d'EZW exploite le fait qu'il y a une probabilité très élevée que tous les coefficients dans un arbre quadruple soient plus petits qu'un certain seuil si la racine de cet arbre est plus petite que ce seuil. Ceci entraîne alors un seul code zerotree pour tout l'arbre.

Ceci dit, en balayant toute la représentation des coefficients d'ondelettes, des basses aux hautes fréquences, on aura automatiquement beaucoup de zerotree ce qui constituera un gain considérable au niveau de la compression.

Une bonne approche est d'utiliser un seuil et seulement un signal au décodeur si les valeurs sont plus grandes ou plus petites que le seuil. Si nous transmettons également le seuil au décodeur, il peut reconstruire une bonne partie de l'image. Pour arriver à une reconstruction parfaite, on doit cependant répéter le processus après abaissement du seuil, jusqu'à ce que le seuil devienne inférieur au plus petit coefficient que nous avons voulu transmettre. Nous pouvons rendre ce processus beaucoup plus efficace par la soustraction du seuil du coefficient d'ondelette qui lui était supérieur. Le choix des seuils peut être optimisé en considérant un lien

entre eux, si l'ordre prédéterminé est un ordre des puissances de deux, le codage est appelé codage « bitplane », puisque les seuils correspondent dans ce cas-ci aux bits dans la représentation binaire des coefficients. Le codage d'EZW tel que décrit dans [17] utilise ce type de seuils.

L'information additionnelle minimale exigée par le décodeur, outre le code des coefficients significatifs, est le nombre de niveaux de transformées d'ondelettes utilisés et le seuil initial. Si on retranche le coefficient après son codage, il est nécessaire d'envoyer aussi la valeur moyenne de l'image, ça permet une meilleure reconstruction et l'obtention d'un meilleur PSNR.

### 3.3 Algorithme

L'algorithme peut être divisé en deux parties : une tâche principale (aussi passage dominant) qui constitue le fond de tâche du codeur, et une partie secondaire ou subalterne qui est plus pour le raffinement de la reconstruction de l'image pendant le décodage.

#### 3.3.1 Traitement principal

Comme initialisation de l'algorithme, on choisit le seuil de départ, pour le « bitplane coding », le seuil de départ est défini par :

$$t * \max i(x; y)$$

Où  $\text{im}(x, y)$  désigne les coefficients d'ondelettes,  $x$  et  $y$  étant leur position dans l'espace.

Le symbole « Max » correspond à la valeur maximale de tous les coefficients de la représentation en ondelettes de l'image. Ce seuil, comme le montre la définition, est un multiple de deux, en fait c'est la puissance de deux la plus proche (inférieur ou égale) du maximum des coefficients de l'image. Cette puissance fera partie par la suite du paquet d'informations transmis pour initialiser le décodage, cette partie sera développée plus en détail dans la description du protocole du codec EZW.

La matrice de coefficients est parcourue par l'une des méthodes suivantes :

« Raster scan » ou « Morton scan ». Ces méthodes de parcours ont été choisies de manière à préserver l'ordre d'importance des coefficients traités, ainsi, pour les deux types, on commence par parcourir les coefficients de basse fréquence, et on avance graduellement vers les détails (hautes fréquences), la différence entre elles est la façon avec laquelle est parcourue une même sous bande. La figure 3.2 illustre l'ordre de parcours des coefficients pour les deux méthodes sur une matrice issue de transformée en ondelette à trois niveaux de décomposition.

Chacun des coefficients parcourus est comparé (en valeur absolue) au seuil  $t_0$ , si le coefficient est supérieur au seuil il est codé 'Positif' ou 'Négatif', sinon il est soit 'Zéro isolé' ou bien 'Zerotree', on se ramène ainsi de X (selon niveaux de gris, résolution de l'image...) symboles à coder à un dictionnaire de quatre symboles :

#### Positif (P)

indique que la valeur absolue du coefficient traité est supérieure au seuil et que son signe est positif.

#### Négatif (N)

indique que la valeur absolue du coefficient traité est supérieure au seuil et que son signe est négatif.



FIGURE 3.3.1 : Méthodes de parcours des coefficients

#### Zéro isolé (Z)

indique que la valeur absolue du coefficient traité est inférieure au seuil et qu'il existe parmi ses descendants (selon l'arborescence présentée dans la Figure 3.1) ceux qui sont significants (c'est à dire supérieurs au seuil).

#### Zerotree (ZTR)

indique que la valeur absolue du coefficient traité est insignifiante par rapport au seuil considéré ainsi que tous les coefficients qui lui succèdent dans l'arbre de descendance.

Après le parcours de tous les coefficients, le seuil est divisé par deux, et l'opération est refaite selon le nouveau seuil, cette méthode est appelée 'quantification par approximations successives' et peut être refaite tant que le seuil  $t_n$ , est supérieur

ou égal à 1 , sachant que :  $t_n = t_{n-1} / 2$

### 3.3.2 Traitement secondaire

Le traitement secondaire (passage subalterne) sert à faire du raffinement sur les valeurs des coefficients, ainsi, après chaque passage dominant, chaque coefficient codé ' Positif ou Négatif ' subit une autre comparaison sur un autre seuil  $t$  qui est proportionnel au seuil du passage dominant  $n$ .

Ceci peut être ramené à :

Où  $n$  est la puissance du seuil du passage dominant. Ce passage secondaire permet au décodeur dans le cas d'une compression avec pertes d'informations, d'avoir plus de précision sur la valeur du coefficient codé, certes, la reconstruction ne sera pas parfaite, mais, elle sera de loin meilleure que si on ne code qu'avec le passage dominant.

Dans le cas où l'on veut effectuer une compression sans pertes d'informations, le passage subalterne devient inutile.

### 3.3.3 Protocole de codage

Pour que le codage soit parfait, un protocole entre codeur et décodeur est établi, ainsi, le décodeur doit connaître le dictionnaire de codage (dans ce cas les 4 symboles utilisés au codage), et le type de parcours des coefficients effectués (Raster ou Morton scan). Pour sa part, le codeur doit transmettre au moins le seuil de départ, de préférence la puissance associée à ce seuil (une puissance de deux puisque le codage est bitplane), et le nombre de niveaux de décomposition par ondelettes. On peut aussi trouver dans certains cas une condition d'arrêt si l'on effectue une compression sans pertes d'informations.

### 3.3.4 Décodage

En premier lieu, le décodeur crée une matrice de dimension égale à l'image traitée à partir du nombre de niveaux de décomposition, ensuite, comme le codeur, il calcule le premier seuil dont la puissance lui a été transmise. Le parcours de la matrice 'image' commence alors et selon les symboles lus par le décodeur un traitement est effectué :

- Si le symbole est 'Positif' (P), la valeur du seuil est additionné au contenu de la case en cours.
- Si le symbole est 'Négatif' (N), Le seuil est retranché du coefficient parcouru.
- Si le symbole est 'Zerotree' (ZTR), tout l'arbre associé à ce coefficient sera ignoré par rapport au seuil courant.

- Si le symbole est 'Zero isolé' (Z) : cela veut dire qu'il existe au moins un coefficient appartenant à l'arborescence du coefficient étudié qui est significatif par rapport au seuil courant d'où, aucun coefficient ne sera ignoré dans cette arborescence.

A la fin du parcours, le seuil est divisé par deux et l'algorithme reprend. Si le seuil atteint la valeur 1, la reconstruction ne sera parfaite sans aucune perte, mais au cas où l'on désire arrêter avant le décodage idéal, on peut avoir recours au traitement secondaire qui permettra plus de précision au niveau de la compression avec pertes d'informations.

Le principe général de la méthode est illustré par l'organigramme suivant : Coefficients de l'image .

### Exemple

La méthode est illustrée par l'exemple suivant, le codage a été appliqué sur la matrice de coefficients à trois niveaux de décomposition suivante :

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

FIGURE 3.3.2 : Exemple de Shapiro

Type de parcours choisi Raster scan.

## Résultats obtenus

### Premier passage dominant

Le tableau suivant montre les coefficients parcourus pour  $t=32$ , et les résultats obtenus avec l'algorithme EZW; nous employons les symboles DL et SL pour les liste dominantes et subordonner, respectivement, Le signe F dans la liste dominante indique que le coefficient est significatif pour le seuil courant.

Tree Output	Root Symbol	DL : dominantlist
	$DL = (0, 0)SL = \emptyset$	(0,0)
POS	$DL = (0, 0)F, (0, 1), (1, 0), (1, 1)SL = 63$	(0,1)
NEG	$DL = (0, 0)F(1, 3), (0, 1)FSL = 63, 34$	(1,0)
	$DL = (0, 0)F(1, 0), F, (2, 0), (2, 1)$	(1,1)
	$DL = (0, 0)F(0, 1)F, (2, 0), SL = 63, 34, 49)$	(0,2)
ZTR		(0,3)
ZTR	$DL = (0, 0)F(1, 1), (0, 3)FSL = 63, 34, 49, 47\}$	(5,2)
Z		(5,2)

les coefficients parcourus pour  $t=32$

### Commentaires

- Le coefficient '63' est supérieur au seuil, et comme il est positif, il est codé 'P', il est aussi supérieur au seuil secondaire donc, son second code est 1, ce coefficient va changer de valeur pour le prochain passage dominant, on aura dans cette position 31.
- On a ici un coefficient négatif '-34' dont la valeur absolue est supérieur au seuil actuel, il est donc codé 'N', cependant, dans le passage secondaire, il est inférieur au seuil secondaire, donc il est codé '0' pour le passage subalterne.
- Le coefficient 31 est inférieur au seuil (en valeur absolue), et comme il possède un descendant dont la valeur est supérieur au seuil (47 dans la sous-bande LH1), il est codé '2' (Zero isolé). Le traitement secondaire n'est pas effectué dans ce cas vu que le coefficient n'est codé ni positif ni négatif.
- On remarque que le coefficient '23' ainsi que tout ses descendants sont insignifiants par rapport au seuil considéré, d'où, le coefficient actuel sera code 'Zerotree' (ZTR), et tous ses descendants ne seront pas traités pendant ce passage.
- Les principales remarques c'est que les sous bandes HH1 et HH2 ne figure pas dans la liste, et ceci parce qu'ils sont descendant d'un arbre de zéros (zerotree), c'est pourquoi on a sur 64 coefficients dans la matrice, juste 20 d'entre eux sont codés.

## 3.4 Conclusion

Malgré toutes les recherches qui ont suivi la mise en oeuvre du codage zero tree, il demeure une des méthodes de codage les plus utilisés et la plus souvent citée dans les revues spécialisées.

L'avantage de cette méthode est de classer les coefficients par ordre d'importance et de permettre un codage progressif qui permet par la suite d'avoir une bonne représentation de l'image selon un taux de compression désiré. Cependant, le codage EZW en lui même n'effectue pas de la compression, il doit être associé au codage arithmétique, qui jusqu'à ce jour demeure le codage entropique le plus efficace associé au EZW.



# Chapitre 4

## Implimentation , simulation et discussion

### 4.1 Introduction

La tendance actuelle est à l'utilisation croissante d'images digitalisées. La plupart des techniques modernes d'imagerie et de vidéo produisent des données 3D, (télé HD, IRM, scanner, échographie).

Certaines images sont intrinsèquement volumiques alors que d'autres correspondent au contraire à une succession d'images 2D (encore appelée pile d'images), à laquelle on ajoute une dimension supplémentaire, à savoir l'écart entre deux coupes successives .Ce qui est notre cas dans ce mémoire.

De fait, la majorité des vidéos produites de nos jours peuvent être vues comme des images à au moins trois dimensions. La quantité importante de ces images volumiques produites par les différents canaux de transmissions se chiffre à plusieurs Téra octets.

Par exemple dans le cas de l'imagerie médicale, dans un service classique de radiologie les données d'une seule année nécessitent un volume de stockage conséquent.

L'augmentation croissante et continue des capacités de stockage apporte une réponse partielle à ce problème mais demeure la plupart du temps insuffisante. La compression semble donc incontournable pour résoudre ce problème d'archivage. De plus, elle présente un intérêt évident pour la transmission de ces images qui peut s'avérer délicate du fait des bandes passantes existantes.

Actuellement, le meilleur moyen pour répondre aux exigences est d'effectuer une compression sans perte. Ce type de compression avec une reconstruction exacte de l'image de départ, garantissant l'intégrité des données demeure le préféré.

Pour ce type de compression, il faut toujours trouver un compromis entre le

taux de compression et la fidélité des données, c'est le défi majeur qu'il faut relever.

Une manière simple d'effectuer la compression est d'appliquer un algorithme de compression 2D pour chaque coupe indépendamment. Ainsi, l'idée de base des algorithmes de compression des images 3D est de s'appuyer sur la corrélation des coefficients dans les trois dimensions pour améliorer les performances de codage.

La majorité des approches utilise une transformée 3D décorrélante avec des algorithmes de quantification/codage qui ont prouvé leur efficacité dans le cas des images 2D.

## 4.2 Simulation

### 4.2.1 Structure du programme

Dans notre application pour la compression d'image en mouvement ; nous utilisons des transformations en ondelettes tridimensionnelles pour décomposer la séquence d'images en sous-bandes selon une architecture pyramidale. Cette opération nous permet d'avoir alors des sous bandes non corrélées entre elles , ce qui permet de réaliser au mieux les étapes suivantes de codage .

On a utilisé le codage EZW 3D qui est une extension du codage EZW2D déjà mentionné dans le chapitre précédent.

La structure générale de la chaîne de compression (analyse et synthèse) sur laquelle repose ce travail est représentée en la figure.

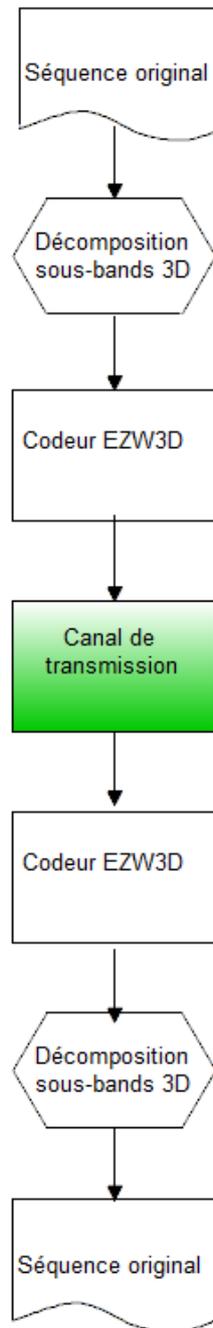


FIGURE 4.2.1 : Algorithme général de codage et décodage

Cette structure est divisée en 2 étapes :

1. Décomposition en ondelettes tridimensionnelles.
2. Codage EZW.

### 4.2.2 Décomposition tridimensionnelle en ondelettes

Comme la transformée bidimensionnelle, la transformée 3D peut être obtenue par une décomposition séparable à base de la transformée 1D appliquée dans les trois directions (horizontale, verticale et temporelle). Cette dernière est à l'origine de la conception des systèmes de codage vidéo qui ne nécessitent pas une compensation de mouvement, et qui exploitent la redondance temporelle de la même manière que la redondance spatiale tout en considérant que le mouvement est assez lent en fonction du temps.

De ce fait, une partie importante d'énergie du signal est concentrée principalement dans la sous-bande de plus basses fréquences spatio-temporelles dans le domaine transformé.

Il existe deux différents types de décomposition 3D en ondelettes la décomposition dyadique utilisée dans le cadre de ce mémoire 1 et la décomposition en paquets d'ondelettes, [4.2,4.3] .Dans le cas dyadique, une décomposition temporelle est suivie par une décomposition spatiale et le processus est itéré pour la sous-bande spatio-temporelle des plus basses fréquences jusqu'à ce qu'on obtienne un certain niveau de décomposition De cette façon, le nombre de niveaux de décompositions dans les directions spatiale ou temporelle est le même, et le nombre des sous-bandes dans ce cas est  $7N+1$  ( où  $N$  est le nombre de niveaux de décompositions spatiotemporelles).

La figure précédente montre la structure de décomposition 3D dyadique en ondelettes à deux échelles spatio-temporelles où ' Ht ' et ' Bt ' représentent les sous-bandes temporelles hautes fréquences et basses fréquences respectivement, et ' Hh ', ' Bh ', ' Hv ' et ' Bv ' représentent les sous-bandes spatiales hautes fréquences en horizontale, basses fréquences en horizontale, hautes fréquences en verticale et basses fréquences en verticale respectivement.

Dans la transformation en paquet d'ondelettes, le nombre de décompositions spatiales et temporelles peut être différent. Dans ce cas, la transformée en ondelettes 1D est appliquée successivement suivant la direction temporelle pour avoir le nombre désiré d'échelles de décomposition Ensuite, toutes les images de la séquence sont séparément décomposées dans les directions horizontale et verticale Le nombre des sous-bandes qu'on peut avoir est  $(N_t + 1) (3 N_s + 1)$  où  $N_t$  et  $N_s$ , sont les niveaux de décomposition temporelle et spatiale,

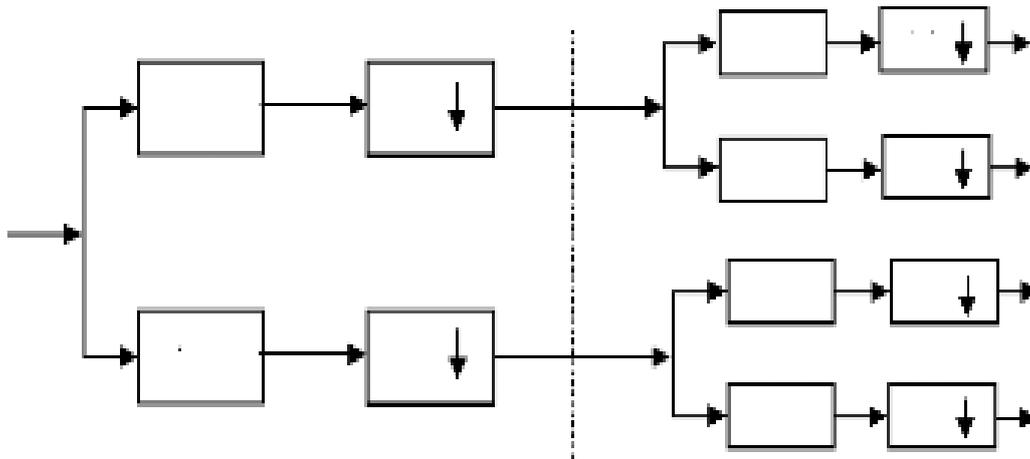


FIGURE 4.2.2 : Décomposition en ondelettes 3D dyadique à deux échelles spatio-temporelles

### 4.2.3 Codeur EZW3D

L'algorithme EZW3D est une extension de l'algorithme EZW2D, que nous avons vu au chapitre précédent, présentant ainsi des caractéristiques similaires : arrangement par amplitude des coefficients, transmission des bits les plus significatifs dans la passe de raffinement et exploitation des autosimilarités à travers les régions spatio-temporelles de la structure arborescente du signal décomposé.

De cette façon, le flux de bits reste parfaitement emboîté, et la qualité progressive de la vidéo est garantie. La phase de codage peut être arrêtée à tout moment pour un taux de bits cibles.

Admettant qu'une certaine distorsion à la reconstruction, on continue le traitement jusqu'à ce que toute l'information soit transmise en cas de reconstruction sans perte ce qui est parfois désirée dans certaines applications telle que la télévision haute définition HDTV.

Dans l'algorithme EZW3D, la passe de triage des coefficients est effectuée de la même manière que dans les algorithmes EZW2D, la seule différence qui existe est la structure de l'arbre défini dans le domaine transformé à travers les sous-bandes. Une fois les coefficients triés (au sens de signification), la passe de raffinement reste inchangée.

Dans la structure 3D des sous-bandes, un nouvel arbre d'orientation spatio-temporelle avec sa propre relation parent-enfants a été introduit [4.1] [4.2] [4.3] [4.4]. Il est défini de telle sorte que chaque noeud ait huit enfants (Figure 4.3 et

4.4). Pour la liste dominante contient tous les pixels de la sous-bande de plus basses fréquences (le niveau le plus haut de la pyramide) qui sont les racines de l'arbre .Et à l'exception des sous-bandes de plus bas niveaux de la pyramide, les enfants d'un pixel de coordonnées  $(i, j, k)$  dans une décomposition dyadique est l'ensemble  $O(i, j, k)$  comme on peut le voir sur cette illustration :

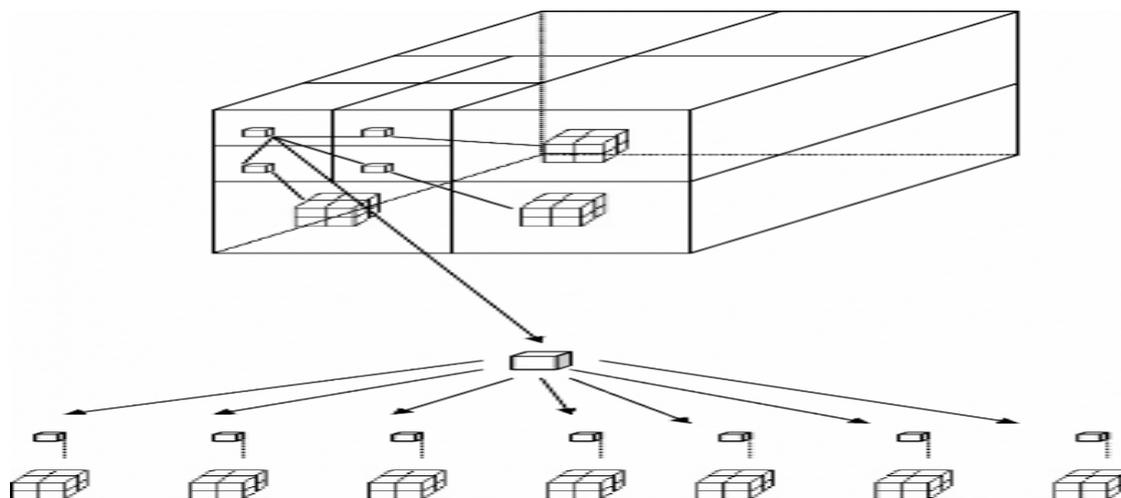


FIGURE 4.2.3 : Illustration de chaque parent ayant huit fils

## 4.3 Résultats

### 4.3.1 Les séquences de test

Des séquences vidéo à niveaux de gris représentent des séquences de test connues et utilisées dans beaucoup de travaux de recherche (images IRM) ont été utilisées pour évaluer les performances du codeur EZW.

Le tableau ci-dessous illustre les caractéristiques de chacune de ces séquences :

Séquence	Dimension	Nombre d'images	Taille Brute (Octets)	Description
Brain	256*256	16	90.208	Séquence médicale (IRM)
Head	256*256	16	90.991	Séquence médicale (IRM)

caractéristique des séquences de test



FIGURE 4.3.1 : Image 01 de la séquence Brain (Droite) et Image 01 de la séquence Head (Gauche)

### 4.3.2 Estimation des performances

#### Taux de compression (RC)

Le taux de compression, souvent utilisé, est l'inverse du quotient de compression, il est habituellement exprimé en pourcentage .il est mesuré par le rapport entre le volume des données initiales et celui des données codées .Plus le taux de

compression est élevée ; plus l'espace nécessaire pour le stockage diminue, ainsi que le temps nécessaire pour la transmission .Il est calculé par la formule suivante : Taille de l'image après le codage entropie

$$RC = 1 - ([Volume\ final]/[Volume\ initial])$$

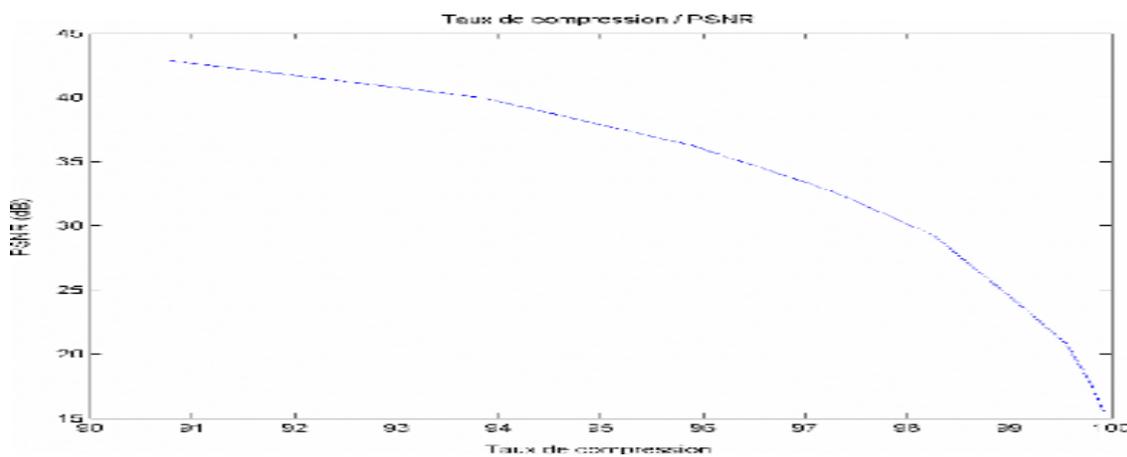


FIGURE 4.3.2 : Graphe : Taux de compression / PSNR

### Choix des filtres

Pour effectuer une étude comparative entre les deux séquences, nous avons choisi trois types d'ondelettes (Haar, Daubechies4, Biorthogonale3.3) sachant que le filtrage temporel s'effectue toujours avec l'ondelette de Haar.

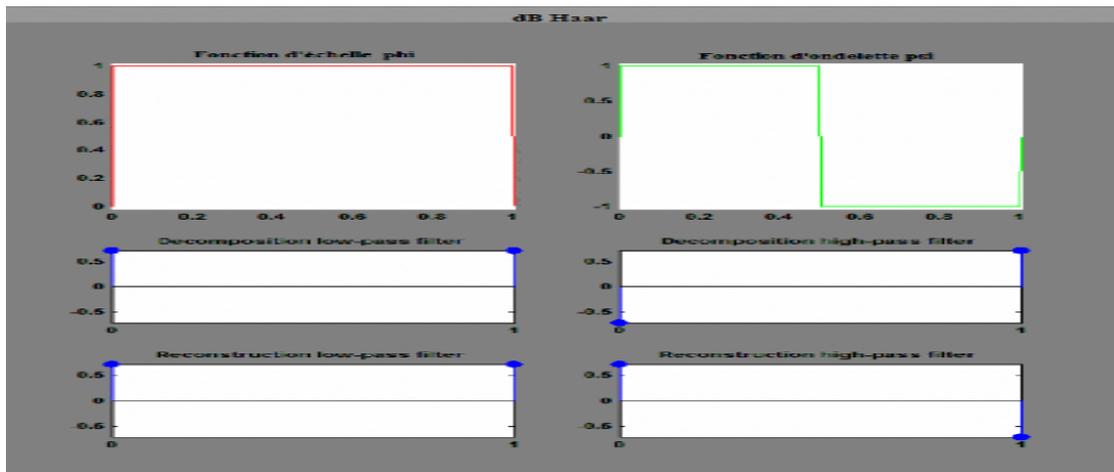


FIGURE 4.3.3 : Coefficient des filtres de décompositions et de reconstructions associés obtenus par l'ondelette de Haar

## 4.4 Les tableaux récapitulatifs des résultats

– Séquence head

RC %	PSNR (dB)
22.8284	93.85
27.0078	88.84
31.3420	97.11

TABLE 4.1 : Résultats obtenus par le filtre Haar

RC %	PSNR (dB)
23.6177	94.41
28.2399	90.19
32.3919	97.16

TABLE 4.2 : Résultats obtenus par le filtre db4

RC %	PSNR (dB)
24.9943	92.70
29.5087	88.09
32.0410	96.09

TABLE 4.3 : Résultats obtenus par le filtre Bior3.3

– Séquence Brain

RC %	PSNR (dB)
22.333	93.61
26.7253	88.49
30.7962	96.87

TABLE 4.4 : Résultats obtenus par le filtre Haar

RC %	PSNR (dB)
23.1504	94.14
27.7794	90.08
31.7530	96.93

TABLE 4.5 : Résultats obtenus par le filtre db4

RC %	PSNR (dB)
24.7447	92.75
29.6394	88.37
32.0093	96.01

TABLE 4.6 : obtenus par le filtre Bior3.3

## 4.5 Discussions

D'après ces résultats, nous remarquons que la méthode de codage EZW 3D conserve sa propriété de transmission progressive des données, et nous donne un flux de bits décroissant par ordre de signifiante, c'est à dire une transmission des bits de poids forts en premier lieu.

Cela nous permet d'avoir des taux de compression élevés et une qualité d'images reconstituée acceptable et exploitable.

On constate une augmentation du taux de compression, quand le PSNR se dégrade ; à cause de la diminution des coefficients significatifs à coder.

Le PSNR (rapport signal sur bruit) obtenu dans les trois filtres se situe entre 22 et 33db et le taux de compression entre 88% et 98%, car pour des images médicales un PSNR de 30 dB est largement suffisant pour leur exploitation.

Un compromis entre un PSNR de 30dB et Un taux de compression supérieur à 88% est toujours préférable puisque il garantit la qualité et la compression en même temps.

Le choix du filtre n'influe pas sur la qualité des images reconstituées, il montre une marge de 1dB seulement.

Le temps de calcul reste le seul désavantage , une séquence 3D ( 256x256x16) nous donne 1048576 pixels à traité , car avec un PC d'une configuration moyenne le temps d'attente est un peu long.

## 4.6 Conclusion

En général; on peut dire que le codeur EZW utilisé dans ce travail permet d'attendre des très bons résultats de point de vue qualité des séquences d'images; et un bon taux de compression. L'EZW en effet permet de réduire énormément la quantité des données contenues dans les séquences d'images tout en conservant leurs propriétés originales.

La compression n'est pas affectée par le choix des filtres utilisés (Haar, db4, Bior3.3).

Un compromis est toujours trouvé entre le taux de compression et la qualité des images.



## Conclusion et perspectives

Nous avons étudié une méthode de compression basée sur la codification d'ordre TOD 3D (la transformée en ondelettes tridimensionnelles) et le codage hiérarchique EZW 3D (Embedded Zerotree wavelet 3D).

Ce travail se base sur l'utilisation de ce codeur sur différent transformée en ondelettes (Haar, daubechies, Biorthogonale) et leur influence sur des séquences d'images biomédicale (IRM) sur le plan qualitatif et quantitatif.

Cette étude nous a permis d'obtenir des résultats satisfaisants de point de vue PSNR (rapport signal sur bruit) et taux de compression (qui peut atteindre 90%).

Notre travail a été appliqué et analysé sur des images 3D « deux séquences d'images biomédicales ». Le codeur EZW à la propriété de la transmission progressive des données. Il permet aussi le choix entre une compression conservatrice ou non selon le choix du seuil de signifiante.

La qualité des images reconstituées peut être améliorée en utilisant progressivement plus de coefficients d'ondelettes. Ceci est possible car le codage EZW permet de classer ces coefficients par ordre décroissant d'importance. Notons aussi que la séquence de transmission des coefficients d'ondelettes dépend des résultats désirés. Pour avoir une image reconstituée qui ressemble à l'image originale, il faut transmettre au début des coefficients d'ondelettes importants qui contribuent à faire apparaître sur l'image reconstituée les informations de hautes fréquences comme les bords et par la suite, des coefficients d'ondelettes de moindre importance qui à leur tour contribuent à faire apparaître sur l'image reconstituée les informations de basses fréquences comme des zones de niveaux de gris constants.

En perspectives nous proposons :

- D'utiliser un autre langage de programmation tel que le langage C. Parce que la programmation du schème de compression utilise énormément de structures.
- L'Application d'un codage arithmétique sur le flux à transmettre.
- Remplacer le code d'ordre par un codage ascii pour diminuer la taille de ce dernier



# Bibliographie

- [1] <http://www.wikipedia.org>, Wikipédia, l'encyclopédie libre.
- [2] Transmission de l'information, cours de l'EPU de Tours.
- [3] <http://danjean.developpez.com/video/signal-video-numerique>
- [4] Antonini M., Badaud M., Mathieu P. and Daubechies I., (1992). Image Coding Using Wavelet Transform. IEEE Transactions on image Processing, i(2), pp. 205-220.
- [5] Grossmann, A., Morlet, J., (1987). Math. & Phys., Lectures on recent results. L.Streit, World Scientific.
- [6] Battle G., (1987). A Block Spin construction of Ondelettes. Comm. Math. Phys., 110, pp. 601-615.
- [7] Y.chen and W. A.pearlman « Three-dimension Subband Coding of video using the zerotree methode in visual communications and image precessing » Proc SPIE2727 pages 1302-1309 mars 1996.
- [8] Belfiore J.C. and Vallet R., (1992). Apport des Modulation fractales pour les Canaux à Évanouissements.
- [9] Chui C.K., (1992). An introduction to Wavelets. Academic Press.
- [10] Chui C.K., (1992). Wavelets : A Tutorial in the Theory and Applications. Academic Press.
- [11] Said, A., Pearlman, W. (1996) New, fast and efficient image codec based on set partitioning hierarchical trees, IEEE Trans. Circ.& Sys. Video Tech, vol .6 no.3, pp. 243-249.
- [12] Bond D.M. and Vavasis S.A., (1994). Fast Wavelet Transforms Matrices Arising from Boundary Element Methods.
- [13] [http://www.ulb.ac.be/cours/acohen/travaux\\_2006\\_infodoc/CompressionNumerique/Introduction](http://www.ulb.ac.be/cours/acohen/travaux_2006_infodoc/CompressionNumerique/Introduction) Université libre de Bruxelles.
- [14] Un schéma de compression avec pertes efficace pour les images médicales volumiques.

- [15] Beong-Jo Kim and William A.pearlman, ( An Embedded wavelet video Coder Using Three-Dimensional Setprtitioning in hiérarchical trees ( SPIHT) )