République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département d'Electronique
Option : *Signal & Communications*

Mémoire de Magister

Présenté par

BOUHADDOUZA Bouzid
Ingénieur d'Etat en Electronique
Université Ferhat Abbas de Sétif

Thème

Algorithmes de Compression d'Image Médicale

Devant le jury d'examen composé de :

Président :	*Mr. ZERGUERRAS Ahmed*	*Professeur*	*(ENP)*
Rapporteur :	*Mr. BERKANI Daoud*	*Professeur*	*(ENP)*
Examinatrices :	*M^{me}.HAMAMI Latifa*	*Professeur*	*(ENP)*
	M^{elle} .GUERTI M'hania	*Professeur*	*(ENP)*

30/10/2011.

Ecole Nationale Polytechnique **10**, ***Avenue Hassen Badi, El Harrach – Alger***
Email : ***enp@ist.cerist.dz***

ملخص

في هذا العمل، قمنا بتنفيذ طريقتي ضغط تم تطبيقها على الملفات الطبية DICOM. لكل طريقة اخترنا اثنين من الخوارزميات . الطريقة الأولى دون خسارة تشمل كل من الخوارزميتين *RLE* و LZW . و الثانية بخسارة تعمل على أساس استخدام المحولة بالتمويجات والنتائج على حد سواء تم اختزالها وفقا لترميز التقنيات SPIHT وكذلكEZW . و لكل طريقة قدمنا مقارنة. عملية الاختزال تمت بواسطة MATLAB.

مفتاح الكلمات : التصوير الطبي،DICOM، ضغط، المحولة بالتمويجات، LZW،RLE الاختزال التدريجي ، SPIHT-EZW

Résumé

Les méthodes de compression d'images numériques peuvent être sans perte ou au contraire. Dans ce travail, nous avons réalisé une implémentation de deux méthodes de compression des fichiers médicaux de norme DICOM. Pour chaque méthode, nous avons effectué une étude comparative. La première méthode est sans perte consiste en les algorithmes RLE, LZW .La seconde présente des pertes, basée sur l'utilisation des ondelettes dont les résultats ont été codés suivant les deux techniques SPIHT et EZW. L'implémentation a été réalisée avec MATLAB.

Mots Clés : Compression de donnés, Imagerie médicale-DICOM, Algorithmes, Transformée en Ondelettes, Compression, RLE, LZW, Codage Progressif, SPIHT, EZW.

Abstract

Compression methods of digital images can be lossless or on the contrary. in this work, we carried out an implementation of two methods of compression medicals files of standard DICOM. For each method we carried out a comparative study. The first method is lossless consists of RLE, LZW algorithms. The second has lossy , based on the use of wavelets and the results were coded according to both technical EZW and SPIHT.

Implementation was carried out using MATLAB.

Key words : Data Compression, Medical Imaging, DICOM, Algorithms, Wavelet Transform, Compression, RLE,LZW, Progressive Encoding, SPIHT,EZW.

Remerciements

Je tiens à remercier mon Dieu de m'avoir donné la patience et le courage pour terminer ce travail.

Je tiens à remercier tout particulièrement Monsieur BERKANI Daoud, Professeur, ENP, pour avoir proposé un sujet d'actualité aussi intéressant ainsi que pour ses conseils tout au long de ce travail.

Je tiens à remercier Monsieur ZERGUERRAS Ahmed, Professeur, ENP, Pour m'avoir l'honneur de présider le jury.

Je tiens à remercier Madame HAMAMI Latifa, Professeur, ENP, Pour l'honneur qu'elle me fait en participant au jury.

Je tiens à remercier Mademoiselle GUERTI M'hania, Professeur, ENP, Pour l'honneur qu'elle me fait en participant au jury.

Je tiens à remercier *ici toutes les personnes : enseignants, étudiants avec qui j'ai eu le plaisir de travailler de* manière *directe ou indirecte durant ces mois. Qu'ils y trouvent une expression de ma reconnaissance.*

Dédicaces

A mes très chers parents, qui sont toujours à mes côtés que Dieu

Vous accorde une longue et heureuse vie.

A mes frères et mes sœurs.

A toute ma famille.

A tous mes amis, qu'ils m'excusent de ne pas pouvoir les citer au risque d'oublier
Quelqu'un.
A tous ceux qui ont contribué un jour à notre éducation et formation.
A tous je dédie ce travail.

Liste des Abréviations

Nous donnons dans cette liste les symboles utilisés et leurs significations :

ψ	Ondelette Mère
AAD	American Association of Dermatology.
ACC	American College of Cardiology.
ACR	American College of Radiology.
ADA	American Dental Association.
AE	Application Entity.
ASGE	American Society of Gastro- Enterology.
Bpp	Bit per pixel.
CAP	College of Anatomo-Pathologists.
CR	Computed Radiography.
CT	Computed tomography.
CCITT	Comité Consultatif International Téléphonique et Télégraphique.
COCIR	European Coordination Committee of the Radiological and Electromedical Industry.
DICOM	Digital Imaging and Communications on Medicine.
DRG	Deutsche RöntgenGeselschaft.
ESC	European Society of Cardiology.
ECG	Electrocardiographe.
H(S)	Entropie de la source S.
ISO	International Organization for Standardization.
I(s)	Information propre.
$\mathbf{I_0(i,j)}$	Valeur du pixel d'image originale.
$\mathbf{I_r(i,j)}$	Valeur du pixel d'image reconstruite.
IRM	Imagerie par résonance magnétique.
IOD	Information Object Definition.
JPEG	Joint Photographic Experts Group
JIRA	Japanese Industry Radiology Apparatus.
LSP	List of Significant Pixels.
LIP	List of Insignificant Pixels.
$\mathbf{log_2}$	Logarithme de base 2..
LZW	Lempel-Ziv-Welch.
MSE	Erreur quadratique moyenne.
MSB	Most significant bit.
MR	Magnetic resonance
NEMA	National Electrical Manufacturers Association.

Ng	Niveau de gris.
MxN	Résolution d'image.
OSI	Open Systems Interconnection.
O	Offsprings.
PSNR	Peak Signal to Noise Ratio.
P(s)	Probabilité de réalisation du symbole s.
$\mathbf{p\_R(i,j)}$	Niveau de gris du pixel Rouge de coordonnées ligne **i** et colonne **j**.
$\mathbf{p\_V(i,j)}$	Niveau de gris du pixel vert de coordonnées ligne **i** et colonne **j.**
$\mathbf{p\_B(i,j)}$	Niveau de gris du pixel Bleu de coordonnées ligne **i** et colonne **j.**
$\mathbf{p(i,j)}$	Niveau de gris du pixel de coordonnées ligne **i** et colonne **j** de l'image.
$\mathbf{p(s)}$	Probabilité d'occurrence de symbole **s**.
PACS	Picture Archivage and Communication System.
RVB	Rouge- Vert- Bleu.
RLE	Run length coding.
$\mathbf{R_C}$	Rapport de copmpression.
SFR	Société Française de Radiologie.
SIRM	Italianadi Radiologia Medica.
SOP	Service/Object Pair.
SCU	Service Class User ou SCU.
SCP	Service Class Provider.
SPIHT	Set Partitioning in Hierarchical Trees.
$\mathbf{S_t}$	Fonction de seuil.
s	Symbole ou lettre d'alphabet de la source S.
TCP/IP	Transmission Control Protocol/Internet Protocol.
$\mathbf{T_C}$	Taux de compression.
UID	Unique Identifier.
VLC	Variable length coding.
WG	Working Group.
2D-DWT	2Dimential discrete wavelets transform.

Liste des Figures

Liste des Tableaux

Sommaire

Chapitre II : La Norme DICOM

Chapitre III : Méthodes Appliquées

Chapitre IV : Résultats et Interprétations

Introduction Générale

Introduction Générale

Le traitement d'images numériques est une discipline nouvelle, qui s'est développée rapidement grâce à l'émergence des nouvelles technologies de l'information. Il s'appuie notamment sur les mathématiques liées à l'information, le traitement du signal, les systèmes électroniques, et sur l'avancée des capacités de calcul des microprocesseurs, notamment ceux qui sont développés exclusivement pour le traitement du signal et qui offrent de grandes capacité et vitesse de calculs (*DSP*…). **[9].**

Les microprocesseurs depuis les années 70 ont connu des changements importants des capacités de calcul, qui ont offert pour les institutions de télémédecines des moyens très avancés de traitement d'images médicales. L'utilisation croissante des ordinateurs dans les applications cliniques a poussé l'American College of Radiology (ACR) et National Electrical Manufacturers Association (*NEMA*) à identifier le besoin de créer le standard *DICOM* (Digital Image Communication On Medicine) pour transférer des images et à associer l'information entre les dispositifs construits par divers fournisseurs **[35]**. Auparavant, ces dispositifs produisaient différents formats d'images numériques**.**

Quotidiennement dans les institutions hospitalières, les images médicales sont stockées et transmises sous forme numérique. La taille et le nombre de ces images ne cessent de croitre de plus en plus. Cependant, la limite de la bande passante impose aux chercheurs la nécessité de développer des algorithmes efficaces de compression d'image pour réduire l'espace mémoire de stockage et augmenter la vitesse de la transmission.

En effet, pour remédier à ces problèmes, l'objet des recherches et des études depuis de nombreuses années, est la nécessité d'élaborer des techniques de compression d'images de plus en plus performantes et adaptées aux caractéristiques des images et aux contraintes des applications actuelles.

Actuellement, la compression du signal est une discipline en plein développement. Et ce afin de trouver ou proposer des solutions, qui économisent l'espace mémoire occupé et le coût de transmission en utilisant des algorithmes bien adaptés en termes de performances et de qualité d'image reconstruite. Néanmoins, certains de ces derniers peuvent présenter des pertes d'informations et d'autres pas.

L'objectif de ce travail est d'examiner les deux types de méthodes de compression (avec et sans perte). Dans la méthode sans perte nous avons pris deux algorithmes : RLE (Run Length Encoding) et *LZW* (Lempel-Ziv-Welch), qui permettent de restituer d'une manière exacte l'image originale.

Pour la méthode avec perte, on a appliqué la technique d'encodage progressif (*SPIHT, EZW*) (Set Partitioning in Hierarchical Trees, Embedded Zerotree Wavelet), qui tolère une certaine perte d'information. Tous les algorithmes cités ont été appliqués sur des fichiers médicaux de norme *DICOM* (Digital Image Communication On Medicine).

- Pour mener à bien notre travail, nous avons adopté le plan suivant :

- Dans le premier chapitre, nous présentons des généralités sur la compression du signal et ses différentes applications sur des images.

- Nous exposerons dans le deuxième chapitre la norme *DICOM*, qui est la plus utilisée dans l'imagerie médicale, et en particulier dans le domaine de la télémédecine ainsi que les différentes spécificités de ces fichiers.

- Dans le troisième chapitre, nous abordons les algorithmes de compression d'image appliqués dans notre application. On commence le chapitre par une présentation générale d'utilité des ondelettes dans la compression d'image et après on cite les détails de chaque algorithme de compression, suivi par un exemple d'application.

- Enfin, le quatrième chapitre sera consacré à exposer les résultats obtenus lors des différents tests de compression effectués sur des fichiers médicaux de norme *DICOM*. Les performances de chaque méthode sont exposées et suivies par des interprétations.

Le travail proposé dans ce mémoire est terminé par une conclusion générale et des suggestions pour de futurs travaux.

Chapitre I

Introduction à la Compression du Signal

Chapitre I : Introduction à la Compression du Signal

I-1 Introduction

On définit par le traitement d'images numériques, l'ensemble des méthodes permettant de modifier une image numérique dans le but de l'améliorer (quantitativement ou qualitativement avec le codage à compression) ou d'en extraire des informations.

Quotidiennement, nous rencontrons des images de toutes sortes dans notre environnement : des photographies, des personnes, des peintures, des dessins par ordinateur, des images de radiologie médicale, des images prises par des satellites etc.…Certaines de ces images (satellite, médicale,…) ne peuvent pas être observées directement, d'autres images présentent des caractéristiques à extraire automatiquement, à stocker et à envoyer. Les traitements envisageables sur les images sont très variés, car les images sont diverses par leurs natures et leurs caractéristiques. Elles sont toutes différentes et il n'est pas bien évidemment de créer un traitement spécifique pour chacune d'entre elles. Il a donc fallu de créer une classification non pas en fonction des caractéristiques des images, mais en fonction de l'objectif du traitement.

Il existe quatre types de domaines d'application du traitement d'image numérique se distinguent :

- La restauration et l'amélioration d'images.
- L'analyse d'images.
- Le codage avec compression d'information.
- La synthèse d'images**. [1].**

I-2 Images Numériques

Une image réelle est obtenue à partir d'un signal continu bidimensionnel comme par exemple un appareil photo ou une caméra... Sur un ordinateur, on ne peut pas représenter les signaux continus, on travaille donc sur des valeurs discrètes. L'image numérique est l'image dont la surface est devisée en éléments de tailles fixes appelés cellules ou pixels , ayant chacune comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle , ou calculé à partir d'une description interne de la scène à représenter .

I-3 Caractéristiques d'une Image Numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants :

- **Pixel**

Contraction de l'expression anglaise 'Picture Elements' : éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le débit est la plus petite unité d'information que peut traiter un ordinateur, le

pixel est le plus petit élément que peuvent manipuler les matériels et les logiciels d'affichages ou d'impression**. [2].**

Une image numérique est aussi définie comme un signal fini bidimensionnel échantillonné à valeurs quantifiées dans un certain espace de couleurs. Elle est constituée de points (pixels) :

- **Signal Fini**
  Une image possède des dimensions finies, exemple : 640x480, 800x600 points…
- **Signal Echantillonné**
  Les pixels d'une image sont régulièrement espacés sur une grille carrée. **[3].**
- **Valeurs Quantifiées**
  Les valeurs des pixels appartiennent à un intervalle borné connu.
- **Espace de Couleur**
  Il existe de nombreuses façons de percevoir les couleurs d'une image, l'espace de représentation le plus connu est l'espace RVB (Rouge-Vert-Bleu).

L'image numérique peut être représentée comme une matrice d'échantillons, où chaque échantillon s'appelle un pixel. Le nombre de bits qui constitue le pixel détermine combien de niveaux de l'intensité peuvent être représentés, et la résolution d'image est exprimée comme nombre de bits/pixel. **[4].**

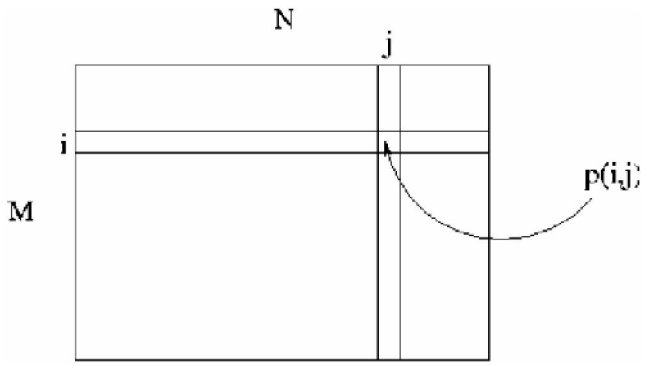


Figure I- 1. Représentation d'une Image Numérique.

Où $p(i,j)$ est le niveau de gris du pixel de coordonnées ligne i et colonne j dans l'image.
$p(i,j)$: Varie dans l'intervalle [0, Ng]. Les valeurs des niveaux de gris sont des entiers.

Selon la représentation des pixels, les images numériques peuvent être classifiées dans:

I-3.1 Image Binaire

Une image binaire est une image MxN où chaque point peut prendre uniquement la valeur 0 ou 1. Les pixels sont noirs (0) ou blancs (1). Le niveau de gris est codé sur un bit (*Binary digit*). Dans ce cas, on revient au cas donné. Avec $Ng = 2$ et la relation sur les niveaux de gris devient: $p(i,j) = 0$ ou $p(i,j) = 1$.

I-3.2 Image en Niveaux de Gris

Une image de niveaux de gris accepte un dégradé de gris entre le noir et le blanc. En général, le niveau de gris est codé sur un octet (8 bits) soit 256 nuances de dégradé. L'expression de la valeur du niveau de gris avec $Ng = 256$.

I-3.3 Image Couleur

Une image couleur est la composition de trois (ou plus) images en niveaux de gris sur trois (ou plus) composantes. On définit donc trois plans de niveaux de gris, un Rouge, un Vert et un Bleu. La couleur finale est obtenue par synthèse additive de ces trois (ou plus) composantes.

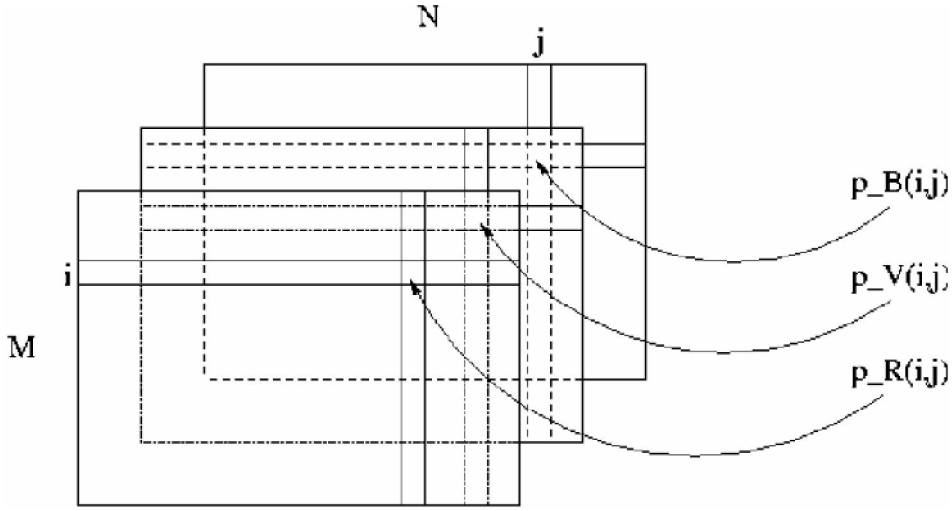


Figure I- 2. La Représentation d'une Image Couleur.

On a les relations sur les niveaux de gris: $p\_R(i,j)$, $p\_V(i,j)$, $p\_B(i,j)$. On voit bien sur la figure qu'une image couleur est en fait l'association de trois plans de niveau de gris, chacun d'eux étant une couleur de base.

- Images binaires, représentées par 1 bit/pixel.
- Infographie, représentée par 4 bits/pixel.
- Images de niveaux de gris, représentées par 8 bits/pixel.
- Et images de couleur, représentées avec 16, 24 bits/pixel ou plus. **[3].**

La résolution d'image se rapporte à ses possibilités pour fournir le détail fin. Une résolution plus élevée exige des systèmes plus complexes pour leur représentation en temps réel. Dans les systèmes informatiques, la résolution est caractérisée avec le nombre des pixels. **[4].**

I-4 Concepts de la théorie de l'information

La théorie de l'information a été énoncée par *Claude E. SHANNON* en 1948, est considérée comme la base théorique de la recherche à la compression des données. **[5].**

On définit une source d'informations comme un modèle mathématique pour une entité physique, qui produit une succession des symboles d'une manière aléatoire. Les symboles produits peuvent être des nombres réels tels que les mesures de tension d'un capteur ou nombres binaires dans le cas des données d'ordinateur, ou bien champ bidimensionnels d'intensité comme dans le cas des séquences des images , continues ou discontinues etc.., l'espace contenant tous les symboles possibles s'appelle l'alphabet de la source. **[6].**

SHANNON aussi a défini la quantité d'information propre ramenée par chaque symbole « self information » comme :

$$\boldsymbol{i(s) = log_2 \frac{1}{p(s)}} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (\boldsymbol{I.1})$$

SHANNON a formulé la définition de l'entropie d'une source d'information discrète sans mémoire comme l'information moyenne statistique par symbole .

$$\boldsymbol{H(S) = \sum_{s \in S} p(s)\, log_2 \frac{1}{p(s)}} \qquad \boldsymbol{\frac{bits}{symbole}} \cdots\cdots\cdots (\boldsymbol{I.2})$$

Où $p(s)$ est la probabilité d'occurrence du symbole s**.**

Dans une source d'information, on comprend intuitivement que tous les symboles n'apportent pas la même information. Un symbole inattendu (faible probabilité) apportera plus d'information qu'un symbole attendu (probabilité forte).

La fonction de l'entropie $H(S)$ est définie par trois conditions :

- Si la source ne délivre qu'un seul symbole (donc une probabilité égale à 1), l'information associée à ce message est nulle.
- En considérant (s_i) l'union de deux événements indépendants (s_j et s_k), s_i=s_j U s_k avec $p(s_i)$= $p(s_j)$ $p(s_k)$, l'information i (s_i) est égale à la somme des informations associées à s_j et s_k.
- H est continue, monotone, et positive. **[7].**

I-5 Types d'Algorithmes de Compression du Signal

Il existe deux grandes familles d'algorithmes de compression du signal :

Ceux qui peuvent reconstituer l'information exacte (Algorithmes sans perte) et ceux qui tolèrent une perte d'information (algorithmes avec perte).

- La compression sans perte, il y a beaucoup de situations qui exigent la reconstruction de signal soit la même que l'original. Dans la compression des textes, Il est très nécessaire que la reconstruction de texte original soit identique, car les petites modifications peuvent avoir conséquence des significations différentes.

- La Compression avec perte, le signal décompressé ne peut pas être récupéré ou reconstruit exactement. Dans beaucoup d'applications, ce manque de reconstruction exacte n'est pas un problème pendant le stockage ou la transmission, du signal parole, de l'image, d'internet multimédia.

I-6 Codeurs d'Images Fixes

La compression d'une image fixe, qu'elle présente ou non des pertes passe essentiellement par l'exploitation des redondances existant dans la source d'information traitée. C'est pourquoi, l'une des premières étapes est la représentation des données sous la forme la plus décolérée possible, et par la suite, une opération de quantification permet de réduire l'entropie de ces données, déjà décolérées, de façon à réaliser un codage efficace.

Dans un premier temps, nous rappellerons donc succinctement les principes de base de la compression d'image fixe, puis expliciterons les techniques d'optimisation d'une telle compression**. [8].**

I-6.1 Codeurs d'Image Sans Perte

L'image décomprimée est numériquement identique à l'image originale pixel par pixel. Ces codeurs sont essentiellement utilisés en imagerie médicale, aérienne, satellitaire ou pour des images obtenues dans les expériences physiques coûteuses. **[9].**

Les types des algorithmes de compression sans perte peuvent être classifiés comme suit:

- Prédictif avec modèle statistique, dans lequel les différences entre les pixels et leur voisinage sont calculées et modélisées avant codage.
- Transformée par ondelettes avant modélisation et codage.
- A codage optimisé **[10].**
- A redondance consécutive compressé (tel que *RLE*).

Dans ce qui suit, nous allons présenter les algorithmes les plus utilisés dans la compression des signaux sans perte :

I-6.1.1 Les Codeurs Entropiques

Ces techniques consistent à coder les symboles qui ont une probabilité plus forte par des mots binaires plus courts que les symboles qui ont une probabilité d'apparition faible. Les algorithmes le plus connus sont le code de *Shannon*, le code de *Huffman*. Ce codage est appelé encore code à longueur variable *(VLC)*. Ils sont destinés à la compression de code binaire sans mémoire.

I-6.1.1.1 Le Codeur de Huffman

L'algorithme a été développé en 1952 par *Huffman*. C'est l'algorithme optimal pour construire un code entropique. Il consiste à construire progressivement un arbre binaire en partant systématiquement des nœuds terminaux de l'arbre précédent. Son principe est comme suit :

Le codage de *Huffman* comprend les étapes suivantes : **[11].**

1. Disposer tous les symboles de source de telle manière que leurs probabilités d'occurrence soient dans un ordre décroissant.
2. Combiner les deux symboles les moins probables de source en formant une probabilité égale à la somme des probabilités des deux symboles moins probables.

- Assigner un 0 au symbole moins probable, un 1 à l'autre symbole qui suit.

3. Répétition jusqu'à ce que l'alphabet auxiliaire nouvellement créé de source contienne seulement un symbole de source.
4. On obtient le code de *Huffman* depuis la probabilité la plus forte à la plus faible.

Le codage de *Huffman* sert à coder une source sans mémoire discrète finie, il a trouvé une large application dans le codage d'image et de vidéo. Il est utilisé aussi dans le codage différentiel et le codage par transformée.

Exemple **[12].**

Soit une source $\boldsymbol{s}$ de l'alphabet de quatre symboles avec les probabilités suivantes **:**

$p[s_1] = 1 / 2$.

, $p[s_2] = 1 / 4$.

$p[s_3] = 1 / 8$.

, $p[s_4] = 1 / 8$.

L'entropie de la source s est 1.75.

L'arbre de *Huffman* est illustré comme suit :

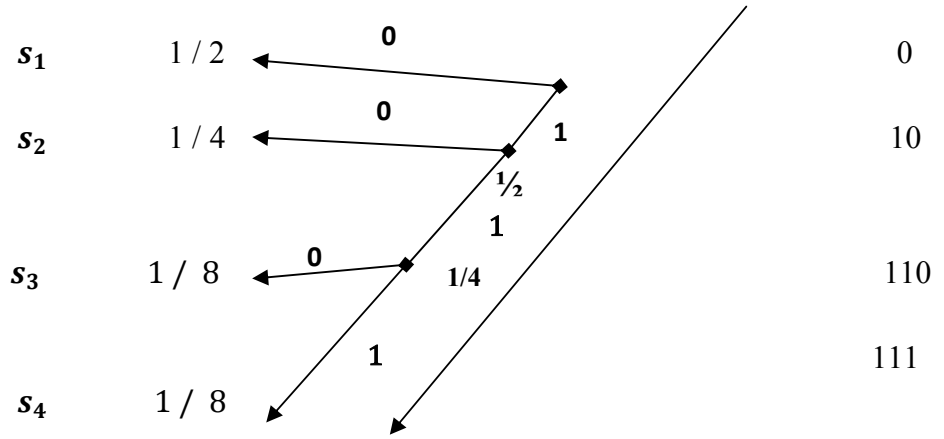


Figure I- 3. La Représentation de l'Arbre de Huffman .

I-6.1.1.2 Codage Arithmétique

Dans le codage de *Huffman*, chaque symbole est substitué par un code de nombre entier de bits. Cependant, le codage arithmétique, une suite des symboles est représentée par un intervalle de valeurs réelles entre 0,0 et 1,0. En conséquence, le résultat du codage peut atteindre à la limite l'entropie de *SHANNON.*

I-6.1.2 Codage par plage

Il est destiné aux modèles aléatoires avec mémoire .Dans le cas des images, il s'agit par exemple d'images qui contiennent des zones uniformes : on a alors une dépendance statistique entre un pixel et son précédent. Les algorithmes les plus connus sont *Run-Length-Coding (RLC)*, et l'algorithme *Lempel –Ziv- Welch (LZW)* utilisé pour créer les fichiers .zip,.gz ou.gif. **[13].**

I-6.2 Codeurs d'Image avec Perte

Actuellement, la compression dans un service de radiologie est toujours effectuée sans perte, elle est réalisée par des standards comme *JPEG* (*Joint Photographic Experts Group*) sans perte dont la syntaxe est prévue dans le standard de format d'images médicales *DICOM1*. Ce type de compression garantissant l'intégrité des données demeure le choix des praticiens pour des raisons évidentes de diagnostic. Cependant il offre de faibles performances en termes de débit binaire, les taux de compression (TC) potentiels varient environ de 2 à 8 suivant le contenu informatif de l'image et la méthode appliquée. L'origine de la préférence des médecins pour la compression sans perte par rapport à la compression avec pertes est, comme on l'a dit, d'éviter les erreurs médicales liées à une mauvaise reconstruction de l'image.

En effet, le principal problème de la compression avec perte pour les images médicales est dû au fait que des détails importants pourraient disparaître (d'autres pourraient éventuellement apparaître). Ces détails sont généralement des structures difficiles à discerner car elles entraînent de faibles changements de contraste. Ainsi par exemple, des images peuvent révéler des lésions à travers des détails potentiellement sensibles à la compression avec perte puisqu'ils sont petits et ont des bords faiblement définis (comme par exemple : le contour d'un pneumothorax, un faible nodule d'une image pulmonaire, etc.). Notons enfin que les erreurs liées à l'utilisation éventuelle de la compression avec pertes pourraient entraîner pour le médecin des problèmes réglementaires et juridiques importants. Il faudrait en effet prouver qu'une erreur de diagnostic ne vient pas d'un problème de la qualité de l'image reconstruite. **[14].**

Pour autant, la compression avec perte est plus que jamais à l'ordre du jour en imagerie médicale, et ce pour les raisons suivantes. Tout d'abord, les études bien que peu nombreuses ont montré que les images médicales possédaient des tolérances à la compression avec perte. On définit cette tolérance comme le maximum de taux de compression pour lequel l'image compressée est jugée acceptable, tant pour l'interprétation humaine que pour celle assistée par ordinateur. Ainsi par exemple, les radiographies de poitrine digitalisées sont très tolérantes à la compression **[1],** (au moins 40 : 1 pour une compression avec la méthode *SPIHT 2D* par exemple).

La quantification est une partie importante dans un schéma de compression avec perte. Le codeur doit tenir compte du type de quantification (scalaire ou vectorielle).

I-6.2.1 La Quantification Scalaire

Une manière simple de mettre en application la compression avec perte, c'est d'associer chaque symbole de séquence à coder par une valeur de son dictionnaire. Les lettres du dictionnaire constituent l'alphabet de quantificateur scalaire, qui est de nombre fini et leur dimension est d'ordre **1**.

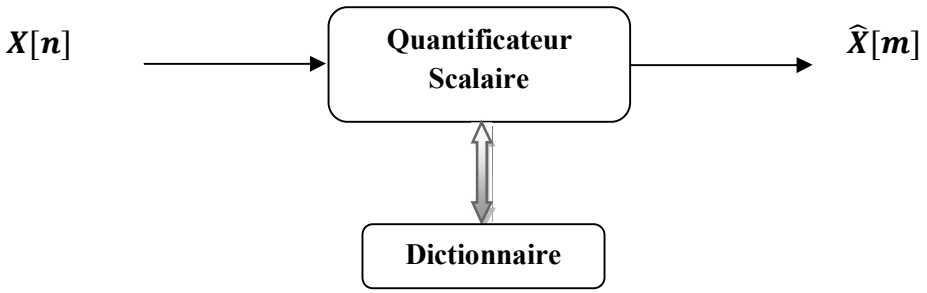


Figure I- 4. Schéma du Quantificateur Scalaire.

Par définition le dictionnaire de la quantification doit se limiter à un nombre fini de variables discrètes. La quantification s'emploie suivant deux approches :

- Si les données à compresser sont sous forme de grands nombres, la quantification est utilisée pour les convertir en petits nombres. Les petits nombres prennent moins d'espace que les grands, ainsi la quantification produit de la compression. D'autre part, les petites valeurs contiennent généralement moins d'information que les grandes valeurs, ainsi la quantification a comme conséquence la compression avec perte.

- Si les données à comprimer sont analogiques (i.e. une tension variant avec le temps) la quantification les numérise d'une façon adéquate. En vue d'un meilleur taux de compression avec la perte d'information admissible.

I-6.2.2 La Quantification Vectorielle

Elle définit un ensemble de blocs d'échantillons (vecteurs de quantification) pour représenter les blocs d'échantillons extraits du signal. Ces vecteurs de quantification constituent le dictionnaire. Et chaque vecteur du signal remplacé par l'indice du vecteur de code le plus ressemblant dans le dictionnaire, Plus le dictionnaire est petit, plus l'indice sera court (forte compression) mais plus le signal reconstruit va être dégradé (perte de qualité).

L'avantage de la quantification vectorielle sur la quantification scalaire c'est que l'on peut exploiter la corrélation entre échantillons. Elle est très puissante car son dictionnaire est construit de façon à minimiser l'erreur de codage (*Algorithme Lloyd Généralisé)*. Elle est souvent associée à une transformation préalable (ex: transformée en ondelettes). **[15].**

I-6.2.3 Codage par Transformée en ondelettes

Plusieurs travaux ont été réalisés avec différentes transformées notamment au niveau spectral : utilisation d'une transformée de *Karhunen-Loeve* (aussi appelée Analyse en Composantes Principales), une *DCT*, une analyse en composantes indépendantes *(ACI)*, ondelettes.

Les principaux développements ont lieu autour de la transformée en ondelettes. En effet, la transformée en ondelettes a montré de bonnes capacités de décorrélation dans un but de compression sur les images réelles, des implémentations avec une complexité limitée.

Il existe principalement deux tendances pour tirer parti de la décorrélation après transformation en ondelettes. La première méthode qui est appliqué dans le standard *JPEG 2000* est d'utiliser un codeur arithmétique pour coder les coefficients d'ondelettes. La seconde méthode tire parti du lien existant entre les coefficients de la transformée situés dans différentes sous-bandes (même si la corrélation est presque nulle).Ces dernières méthodes sont appelées codage par arbres de zéros. Les deux méthodes de codage par arbres de zéros les plus populaires sont *EZW* et *SPIHT*.

Le terme ondelette désigne une fonction qui oscille pendant un temps donné (si la variable est le temps) ou sur un intervalle de longueur finie (si la variable est de type spatial). Au-delà, la fonction décroît très vite vers zéro**. [16].**

Pour les images *2D*, la transformée en ondelettes classique est isotropique i.e. pour une sous-bande donnée, le niveau de décomposition dans la direction horizontale est le même que dans la direction verticale. Cette alternance entre décomposition des lignes et des colonnes conduit à des sous bandes carrées. Un exemple de la décomposition multi résolution classique est illustré sur la figure (I.5). Les coefficients gris représentent des valeurs proches de zéro, tandis que les noirs et blancs correspondent respectivement à des valeurs fortement négatives ou positives.

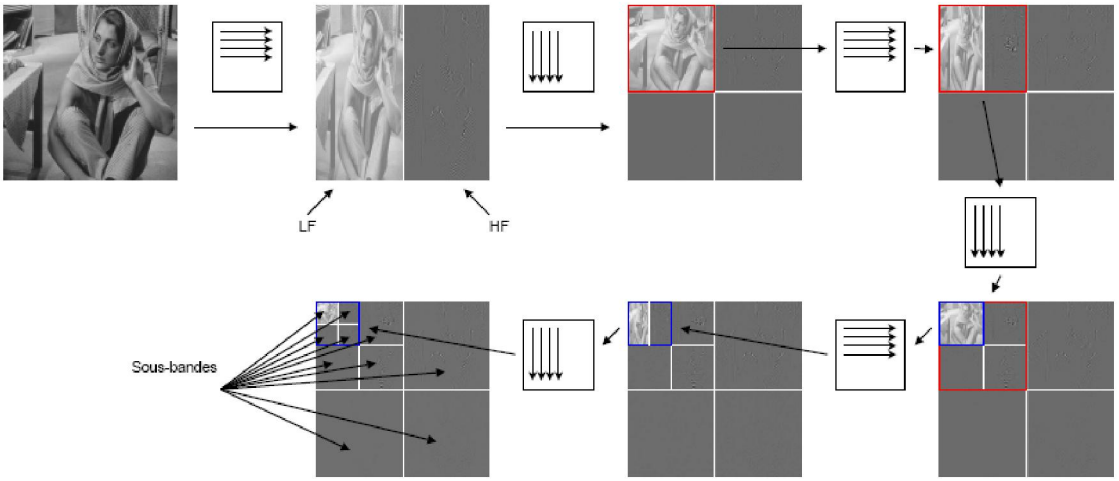


Figure I- 5. La décomposition (2D-DWT) d'une Image.

La Figure I.5 montre la décomposition multi résolution classique : on décompose successivement suivant les lignes et les colonnes. LF est la partie basse fréquence (LF = low frequency) et HF la partie haute fréquence (HF = high frequency).

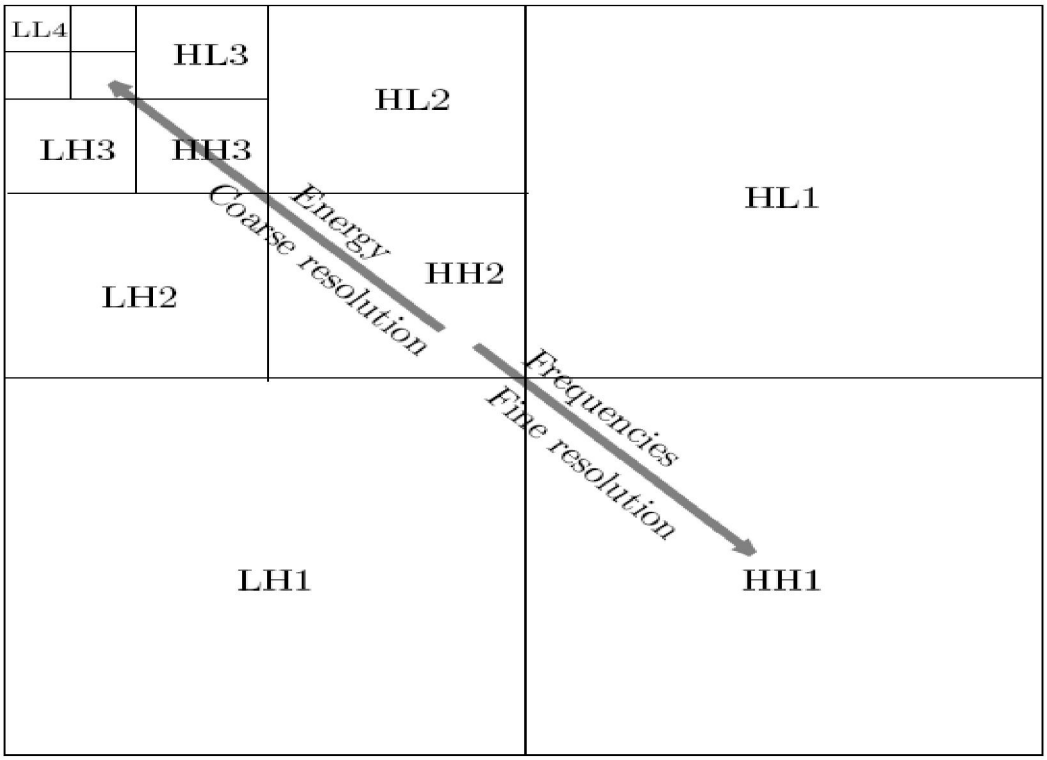


Figure I- 6. Les Sous - Bandes de Décomposition à 4 Niveaux.

I-7 Performances d'Algorithme de Compression

Un algorithme de compression peut être évalué diversement selon sa complexité relative, la mémoire nécessaire à son application, le temps d'exécution sur une machine, le rapport de compression, la qualité de reconstitution du signal d'origine. **[16].**

I-7.1 Rapport de Compression

Il représente le rapport entre le nombre de bits de la forme canonique au nombre de bits après codage : **[18]**

$$\mathbf{R_C = Rapport_C} = \frac{\textbf{Nombrs de Bits Avant Compression}}{\textbf{Nombre de Bits Aprés Compression}} \quad \cdots\cdots (\mathbf{I.3})$$

Le taux de compression est un pourcentage de l'espace obtenu après la compression par rapport à l'espace total requis par les données avant la compression :

$$\mathbf{T_C = Taux_C} = \mathbf{(1-\frac{1}{R_C})\times 100} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (\boldsymbol{I.4})$$

I-7.2 Qualité d'Image Décompressée

Certaines méthodes de compression permettent de trouver exactement les valeurs numériques initiales de l'image originale. Dans de tels cas, le problème de la qualité de l'image

codée ne se pose pas puisque celui-ci n'est pas affecté par l'opération. D'autre Méthodes, en revanche ne permettent pas de reconstituer l'image originale exactement. Dans ces cas, il est nécessaire de comparer la qualité de l'image après la reconstitution à celle de l'originale pour juger la méthode.

Cette qualité peut se mesurer de deux manières :

- Par la qualité subjective de l'image reconstruite lors de l'opération de décodage. Malheureusement il n'existe pas aujourd'hui de mesure universelle de qualité, en effet, d'une part cette qualité elle doit prendre en compte les propriétés encore imprécises du récepteur visuelle humain.

- Par critère objectif de qualité comme par exemple MSE (Mean Square Error):

$$MSE = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} ||I_0(i,j) - I_r(i,j)||^2 \quad \cdots\cdots\cdots\cdots\cdots\cdots (I.5)$$

Où :
$m \times n$: La taille d'image en pixel ou les dimensions de l'image
$I_O(i,j)$: Les valeurs du pixel de l'Image originale
$I_r(i,j)$: Les valeurs du pixel de l'image reconstruite
La valeur $m \times n$ est définie comme la résolution d'image (nombre total des pixels).

I-7.3 Rapport *PSNR*

Dans un système de compression d'images, l'image dégradée par compression I_r est toujours comparée à l'originale I_O, pour déterminer son rapport de ressemblance. On trouve le critère quantitatif le plus utilisé qui est basé sur la mesure de l'erreur quadratique moyenne *(MSE)* calculée entre les pixels originaux et reconstruits. **[19].**

Le $PSNR$ (Acronyme Peak Signal to Noise Ratio) est calculé en échelle logarithmique, leur valeur dépend de MSE. Il est utilisé pour comparer la qualité d'image reconstruite par rapport à l'image originale. **[20].**

$$PSNR_{dB} = 10.\log_{10}\left(\frac{d^2}{MSE}\right) \quad [\text{dB}] \quad \cdots\cdots\cdots\cdots\cdots\cdots (I.6)$$

Où d est la dynamique du signal. Dans le cas standard d'une image où les composantes d'un pixel sont codées sur 8 bits, $d = 255$.
Les valeurs typiques de $PSNR$ pour des images de bonne qualité varient entre 30 et 45 *dB.*

Le $PSNR$ peut avoir des valeurs dans la gamme *[0 dB, +∞ dB],* avec $PSNR$ égal à $+\infty$ *dB* quand les deux images comparées sont identiques (compression sans perte). **[21].**

I-7.4 Similarité Structurelle [SSIM]

Pour les applications dans lesquelles des images doivent finalement être vus par les êtres humains, la seule méthode correcte de mesurer la qualité visuelle d'image est l'évaluation subjective. Dans la pratique, cependant, l'évaluation subjective est habituellement trop incommode **[22].** Pour une évaluation objective, les méthodes objectives pour évaluer la qualité perceptuelle de l'image ont traditionnellement essayé de mesurer la visibilité des erreurs entre une image dégradée et une image de référence en utilisant une variété de propriétés connues du système visuel humain *(SVH)* **[19].** *SSIM (**S**tructural **S**imularity)* est une mesure de similarité entre deux images numériques, plutôt qu'une différence pixel à pixel comme le fait par exemple le $PSNR$.

La métrique *SSIM* entre deux fenêtres de taille $N \times N$ de deux images x et y est :

$$\boldsymbol{SSIM(x,y)=\frac{(2\mu_x\mu_y+c_1)(2cov_{xy}+c_2)}{(\mu_x{}^2+\mu_y{}^2+c_1)(\mu_x{}^2+\sigma_x^2+\sigma_y^2+c_2)}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{(I.7)}$$

Alors la similarité *SSIM* est déterminée à travers les *M* fenêtres et dans lequel l'index *MSSIM* est appliqué.

$$\boldsymbol{MSSIM\ (I_O,I_r)=\frac{1}{M}\sum_{j=1}^{M} SSIM(x_j,y_j)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{(I.8)}$$

Où :

μ_x , μ_y: les valeurs moyennes de x et y .

σ_x^2 : La variance de .

σ_y^2 : La variance de y.

cov_{xy} : La covariance de x,y .

$c_1 = (K_1L^2)$, $c_2 = (K_2L^2)$, deux variables destinées à stabiliser la division quand le dénominateur est très faible.

L : La dynamique des valeurs des pixels, soit 255 pour des images codées sur 8 bits ;

L'indice de similarité est exploité pour un choix approprié des constantes K_1 et K_2. Pour généraliser l'évaluation à toute l'image.

$K_1 = 0.01$; $K_2 = 0.03$. Par défaut.

Pour l'évaluation de qualité d'une image, la formule précédente est appliquée sur la luminance uniquement. Typiquement, les grandeurs sont calculées sur des fenêtres de taille $\boldsymbol{8 \times 8}$. La fenêtre courante peut se déplacer pixel par pixel sur l'ensemble de l'image.
Les auteurs proposent de ne considérer qu'un ensemble de ces fenêtres, par exemple en réduisant leur nombre d'un facteur 2 dans les deux dimensions. Ceci permet de diminuer la complexité du calcul.

Le *SSIM* et *MSSIM* satisfait les conditions suivantes :

- La symétrie : $SSIM(x, y) = SSIM(y, x)$
- $SSIM < 1$.
- Unificité du maximum : $SSIM(x, y) = 1$ si et seulement $x = y$.

L'indice de *MSSIM* peut avoir des valeurs dans l'intervalle [0, 1], avec *MSSIM* égal à 1 quand les deux images comparées sont identiques. Les valeurs dans l'intervalle [- 1, 0] sont possibles quand le contraste de l'image d'essai est renversé.

I-8 La Chaine Principale de la Compression/Décompression d'Image

Les principales parties de la chaine de la compression /décompression d'image sont présentées dans la figure suivante :

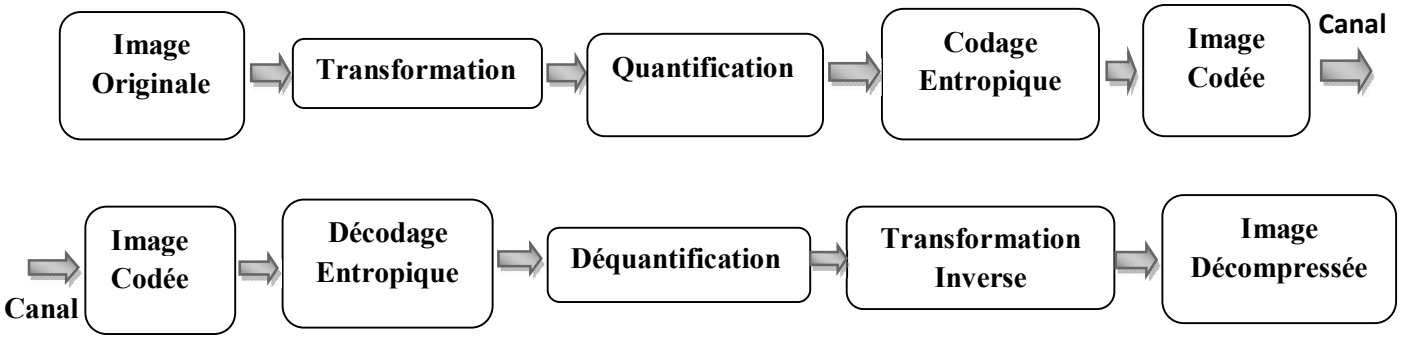


Figure I- 7. Chaine de Compression/Décompression d'Image.

I-8.1 La Transformée d'ondelettes en Imagerie

La transformée *DCT* (discret Cosine Transform) a montré de bonnes performances pour la compression des images. Elle est notamment utilisée dans le standard *JPEG*. Les dernières avancées montrent que la transformée en ondelettes est plus intéressant car elle supprime notamment les effets de blocs. Elle est utilisée dans le nouveau standard *JPEG2000* ainsi que par *EZW*, et *SPIHT*. **[23].**

Elle existe également des transformées adaptées aux données. Par exemple la *KLT* (Karhhunen-Loeve Transform) consiste à projeter l'image sur des vecteurs propres.

Ces transformées sont utilisées afin de représenter les pixels de l'image par des coefficients réorganiser de manière à pouvoir éliminer les redondances statistiques et la corrélation entre les pixels voisins, et compacte l'information (c.-à-d. réduit l'entropie de la source).

L'idée est simple, comme tout filtre appliqué à une image, il suffit de multiplier les coefficients de l'ondelette (après l'avoir discrétisée) par la valeur des pixels de l'image à traiter. On notera que l'ondelette peut alors être représentée sous la forme d'un tableau (de Dimension **1**)

de valeurs décimales. En réalité il y a *2* tableaux qui représentent l'ondelette car une ondelette se compose de *2* filtres : un Passe-Bas et un Passe-Haut. **[24].**

Voici ci-dessous les coefficients de l'ondelette de Haar discrétisé ainsi que ceux de l'ondelette de Daubechies :

Ondelette de Haar : ***Ondelette de Daubechies :***

Passe-Bas *:* ***Passe-Bas :***

[0.71; 0.71] *[0.027 ; -0.017 ; -0.078 ; 0.267 ;* ***0.603*** *; 0.267 ; -0.078 ; -0.017 ; 0.027]*

Passe-Haut : ***Passe-Haut:***

[-0.71 ; 0.71] *[0 ; 0 ; 0.046 ; -0.029 ; -0.296 ; 0.558 ; -0.296 ; -0.029 ; 0.046]*

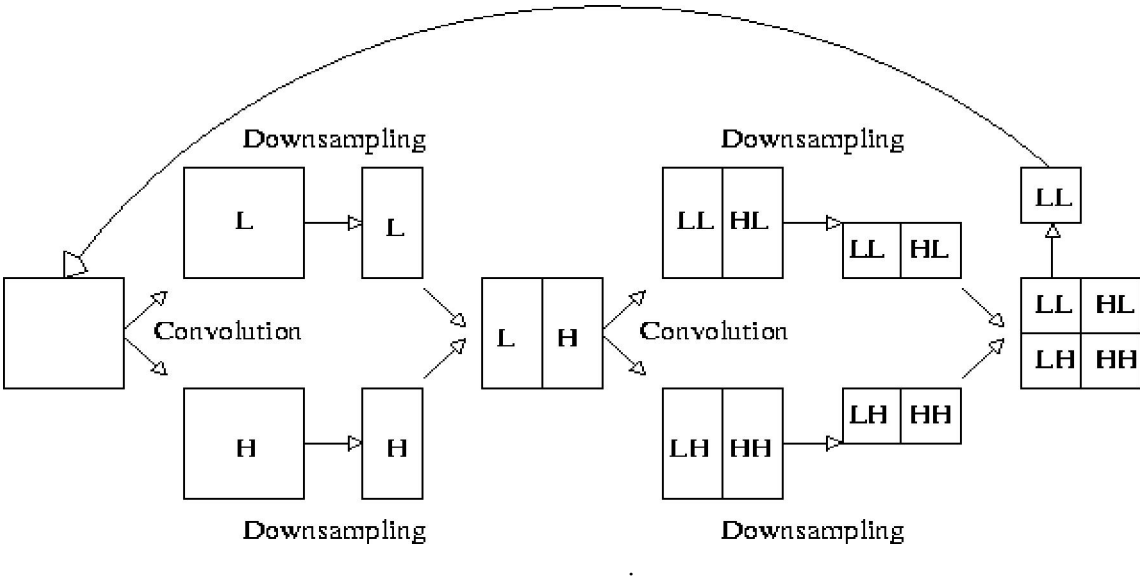


Figure I- 8. Schéma général de traitement d'une image par Ondelettes.

- La lettre *H* correspond au résultat d'un filtre passe-haut, la lettre *L* à celui d'un passe-bas.
- Le downsampling correspond à un sous-échantillonnage de l'image dans le sens verticale ou horizontale suivant le cas d'utilisation.
- La convolution correspond à l'application d'une ondelette comme décrit ci-dessus.

Ainsi, on remarque que les étapes qui se succèdent comprennent certaines ressemblances et ne varient que d'une application horizontale à une application verticale.

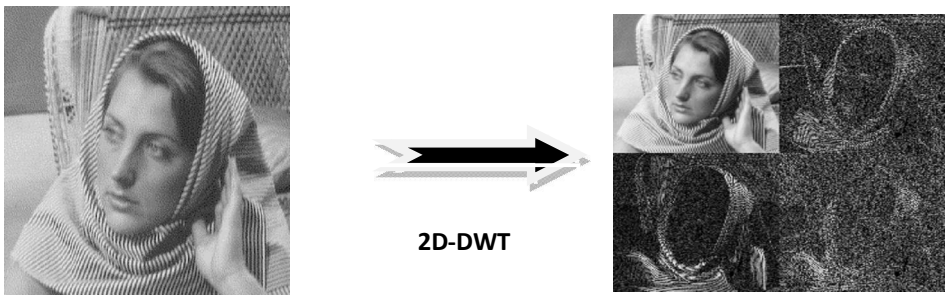


Figure I- 9. La Décomposition en Ondelette.

La décomposition en sous-bande a été présentée la première fois par *CROCHIERE* et autres en 1976, et depuis elle s'avère être une technique simple et puissante pour la compression de la parole et de l'image. Le principe de base est la division du spectre signal en bandes de fréquences codées et transmises chacune séparément. Comme en particulier, les images normales tendent à avoir un spectre non-uniforme de fréquence, avec la majeure partie de l'énergie concentrée dans la bande de plus basse fréquence. La perception humaine du bruit tend abaisser au-delà des hautes fréquences en deca des basses fréquences, et ceci permet au concepteur d'ajuster la déformation de compression selon des critères subjectifs, puisque les images sont traitées dans leur intégralité, et pas dans les blocs artificiels, il n'y a aucune déformation de structure de bloc dans l'image codée, comme cela se produit dans les encodeurs tels que *DCT*. **[25].**

I-8.2 La Quantification

La quantification est une partie importante dans un schéma de compression **[26]**. Les coefficients à la sortie de la transformation (ou des filtres d'analyse dans le cas d'un banc de filtres) prennent en général des valeurs réelles qui doivent être quantifiées pour réaliser la compression. La quantification dans sa forme la plus simple s'applique à chaque coefficient séparément, on parle alors de quantification scalaire. Une autre méthode plus complexe consiste à quantifier plusieurs coefficients à la fois, c'est la quantification vectorielle. L'idée est trouver une bijection qui applique l'ensemble, qui fournit théoriquement la performance optimale pour tout codeur**.**

I-8.3 Le Codage Entropique

La dernière étape dans un schéma de compression est le codage entropique. C'est une opération parfaitement réversible, c'est-à-dire qu'elle n'introduit aucune distorsion contrairement à la quantification. Il existe deux grandes classes de codeurs entropiques.

La première est constituée des codeurs qui n'exploitent pas la redondance entre symboles consécutifs et qui donnent au mieux, comme longueur moyenne du flot de bits, l'entropie d'ordre *1* des données à coder.

La deuxième est constituée des codeurs qui exploitent cette redondance pour donner une longueur moyenne qui tend au mieux vers le débit entropique de la source. Après la quantification, les coefficients prennent des valeurs dans un ensemble fini de symboles $\{s_0, s_1, \ldots, s_{L-1}\}$.L'idée est de trouver une bijection qui applique l'ensemble $\{s_i : 0 \leq i \leq L\}$ dans un nouvel ensemble $\{z_i : 0 \leq i \leq L\}$ de manière à minimiser le nombre moyen de bits par coefficient. **[27].**

I-8.4 Image Codée

Le résultat de codage entropique présente le format final de l'image codée, ce fichier peut être stocké dans notre environnement de travail ou bien envoyer par canal avec une suite des opérations de codage canal pour rendre le fichier compatible avec la chaine de transmission.

I-9 La Chaine de Décompression

Dans tous les algorithmes de compression, la chaine de décompression est exactement l'inverse de la chaine de compression. Dans ce qui suit on va présenter la fonctionnalité de chaque bloc de décompression avec un certain détail.

I-9.1 Le Décodage Entropique

Cette étape sert à appliquer la méthode de décodage entropique qui a été choisie, tels que le décodeur de *Huffman* ou le décodeur arithmétique, qui sont utilisés dans le but d'augmenter les valeurs de $PSNR$, $MSSIM$ et minimise la distorsion exprimée par MSE.

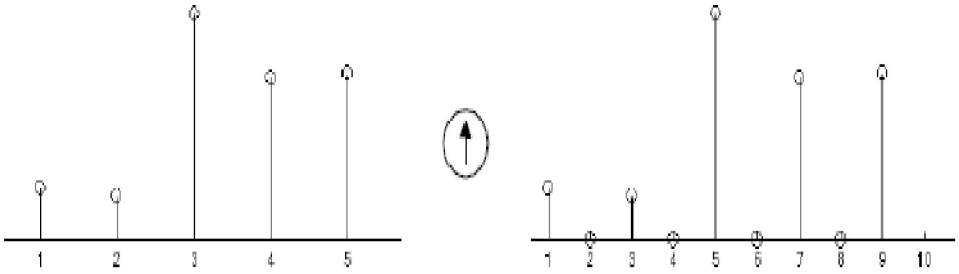
I-9.2 La Déquantification

Cette partie est la base des techniques de compression avec perte, ces techniques généralement basées sur dictionnaire tels que les quantificateurs scalaires ou vectorielles, les dictionnaires sont générés pendant l'étape de quantification, donc la déquantification substitue à chaque mot produit dont l'étape de décodage entropique, une valeur du dictionnaire de quantificateur utilisé.

I-9.3 Transformation Inverse

Le passage de l'espace transformé à celui d'image se fait par les moyens usuels tels que la transformée de Fourier inverse*(IFT)*, la transformée d'ondelettes inverse *(IDWT)*…. **[28].**

Le processus de reconstruction d'ondelettes se compose de sur- échantillonnage et filtres de reconstitution. Le sur-échantillonnage est le processus de rallonger un composant de signal en insérant des zéros entre les échantillons :



Séquence d'un Signal Séquence du Signal Sur-échantillonnée

Figure I- 10. Principe de Sur-échantillonnage d'un Signal.

La partie de filtrage du processus de reconstruction prend également une certaine partie de discussion, parce que c'est le choix des filtres qui est crucial en réalisant la reconstruction parfaite du signal original.

I-10 Conclusion

Nous avons exposé durant ce chapitre un ensemble de notions de base sur la compression du signal, parmi lesquelles nous avons constaté qu'il y a deux familles d'algorithmes :La première famille est sans perte, elle permet de restituer l'image exacte après la décompression. La deuxième famille présente des pertes, i.e. Elle reconstitue l'image décompressée avec une certaine distorsion par rapport à l'image originale. Ces deux types de familles d'algorithmes de compression sont appliqués sur des fichiers médicaux de norme *DICOM*.

Dans le chapitre suivant, nous présentons les détails du fichier *DICOM*.

Chapitre II

La Norme DICOM

Chapitre II : La Norme DICOM

II-1 Introduction

L'imagerie médicale est le procédé par lequel un médecin peut examiner l'intérieur d'un corps d'un patient sans l'opérer. Elle peut être utilisée à des fins cliniques, à la recherche de diagnostic ou pour le traitement d'un grand nombre de pathologies mais également pour la recherche dans le but d'étudier la physiologie des êtres vivants.

Les méthodes de l'imagerie médicale sont nombreuses et utilisent plusieurs procédés physiques tels que :

- Les rayons X.
- Les ultrasons.
- Les émissions de rayonnement par les particules radio actives.
- Le magnétisme de rayon des atomes. **[29].**

Le traitement d'image médicale a connu une forte progression au cours des années 1970, induit par l'émergence des nouvelles technologies de l'information. La nécessité d'un format spécifique en imagerie médicale indépendant des équipements électroniques fournis par différents constructeurs (marque), imposa l'adoption d'une normalisation pour éviter la disparité des matériels, la norme *DICOM* est l'une de celles-ci. **[30].**

II-2 Présentation de la norme DICOM

La norme *DICOM* « Digital Image Communication On Medicine » est la plus utilisée en radiologie à travers le monde **[31],** car elle facilite la connectivité, la distribution et le visionnement d'image médicale entre les différents dispositifs (matériels) utilisés dans les périphériques médicaux de traitement d'image. **[32].**

Actuellement la norme *DICOM* est structurée comme un fichier des **18** parties suivantes :

- **Introduction et Vision Générale (Introduction and Overview)**
  Elle rassemble l'historique des normes d'images médicales et l'ensemble des raisons qui ont mené à leur création, et les objectifs à atteindre.

- **Conformité (Conformance)**
  Elle définit les principes généraux qui doivent être satisfaites par n'importe quelle déclaration d'implémentation du standard. C'est suivant cette partie que les déclarations de conformité sont rédigées.

- **Définitions des Objets d'Informations (Information Object Definitions IODs)**
  Elle définit un certain nombre de classes d'objet qui donne une définition abstraite des entités réelles applicables à la communication des images médicales numériques et de l'information relative (par exemple, formes d'onde, rapports structurés, dose de thérapie radiologique, etc...).

- **Définition des Classes de Service (Service Class Specifications)**
  Une classe de service est associée à un ou plusieurs objets d''information plus une ou plusieurs commandes d'exécution sur ces objets. Les exemples des classes de service incluent ce qui suit :

— La classe de service de stockage (Storage Service Class).
— La classe de service question / Recherche (Query/Retrieve Service Class).
— La classe de service de gestion de liste du travail (basic Worklist Management Service Class).
— La classe de service de gestion d'impression (Print Management Service Class).

- **Structure de Données et Codage (Data Structure and Semantics)**
  Elle définit comment les applications de *DICOM* construisent et codent l'information de modem résultant de l'utilisation des objets de l'information et entretient des classes de la norme *DICOM*. L'appui est indiqué sur un certain nombre de techniques standards de compression d'image (par exemple, *JPEG* lossless et lossy).

- **Dictionnaire de Données (Data Dictionary)**
  Elle définit tous les codes des balises et les d'informations qui définissent la norme *DICOM*.

- **Echange de Message (Message Exchange)**
  Elle définit la structure de service et le protocole de communication utilisé par une application dans un environnement de traitement de l'image médicale pour échanger des messages. Le *DICOM* Message Service Element *(DIMSE)* échangé entre les services de communication point – à-point ou à travers un réseau. Un *DIMSE* définit un élément de communication incluant un service et un protocole. Il est utilisé par les entités *DICOM* application Entity (*AE*) pour transformer les informations médicales. Le protocole *DIMSE* définit les règles nécessaires pour la construction de messages conformes**. [33].**

- **Support Réseau de Communication d'Echange de Message (Network Communication Support for Message Exchange)**
  Dans un environnement géré en réseau, elle spécifie les services de communication et les protocoles des couches supérieures nécessaires à la communication entre les applications *DICOM*. Cette définition spécifie l'utilisation des protocoles des couches supérieures de *DICOM* en même temps que des protocoles de transport *TCP/IP* (Transmission Control Protocol/ Internet Protocol). **[34].**
- **Support de Communication Point-à-Point (Point-to-Point Communication Support for Message Exchange)**
  Ainsi la norme *DICOM* a précédemment spécifiée aussi les services et les protocoles utilisés pour les communications point-à-point afin de les rendent compatibles avec *ACR-Nema 2,0*.

- **Stockage de Médias et Format de Fichier (Media Storage and File Format)**
  La norme *DICOM* spécifie le format général pour le stockage d'information médicale sur des médias. Le but de la présente partie est de fournir un cadre permettant l'échange de divers types d'images médicales et d'informations relatives sur une large gamme de supports de stockage physiques.

- **Profile des Applications aux Supports de Stockage** (**Media Storage Application Profiles**)
  Elle définit les sous-ensembles d'application spécifiques de norme *DICOM* à laquelle une exécution nécessite la conformité. Ces sous-ensembles d'application spécifiques ont des paramètres tels que le rapport de l'échange des images médicales et de l'information relative sur les supports de stockage.

- **Fonctions de Stockage et Formats de Médias pour l'Echange de Données (Storage Functions and Media Formats for Data Interchange)**
  Elle spécifie la facilité de l'échange d'information entre les applications dans les environnements médicaux par :
  — Une structure qui décrit la relation entre le modèle de stockage et les supports physiques et des formats spécifiques.
  — Les caractéristiques des supports physiques des formats associés.

- **Gestion Support de Communication point-à-point (Management Point-to-point Communication Support)**
  Elle a précédemment défini les services et les protocoles utilisés pour la gestion de la communication point-à-point et des services de gestion d'impression.

- **Fonction Standard d'Affichage à Niveaux de Gris (Grayscale Standard Display Function)**
  Elle définit une fonction normalisée d'affichage d'image en niveaux de gris. Cette fonction fournit des méthodes pour calibrer un système de visualisation particulier afin de présenter des images uniformément sur différents supports d'affichage (par exemple des moniteurs).

- **Profils de Sécurité et de Gestion du Système (Security and System Management Profiles)**
  Elles se réfèrent aux protocoles standards tels que les clés publiques, les cartes smartes, le cryptage des données etc...

- **Ressources Terminologiques (Content Mapping Resource)**
  Elle définit l'ensemble des abréviations utilisées dans le document.

- **L'Information Explicative (Explanatory Information)**
  Elle Permet de définir des annexes instructives et normatives contenant l'information explicative du document.

- **Accès Web (Web Access to DICOM Persistent Objects WADO)**
  Elle définit un service qui permet de faire une demande d'accès à un objet *DICOM* par web : par exemple *HTTP URL/URI*.

II- 2.1 Buts

La norme *DICOM* facilite l'interopérabilité entre différents dispositifs revendiquant une conformité en :

- Définissant la sémantique des commandes de communications et des données associées. De sorte à fournir un certain nombre de règles à suivre par les constructeurs .En fait, pour que deux ou plusieurs dispositifs puissent communiquer en liaison point-à-point ou à travers un réseau, il doit y avoir des normes sur la manière dont ils sont censés réagir à la différente commande et aux données qui y sont associées, et non seulement aux informations qui sont simplement transférées de l'un à l'autre.

- Définissant la sémantique des services d'archivage, des formats de dossier et des annuaires de l'information nécessaires pour la communication non-connecté (Off-line).

- Définissant de manière explicite les conditions de conformité pour son implémentation. En particulier, un rapport de conformité doit obligatoirement être émis avec chaque nouveau dispositif respectant la norme : document qui doit contenir assez d'informations permettent de définir les fonctions d'interopérabilité avec d'autres dispositifs.

- Facilitant les opérations dans un environnement connecté en réseau.

- Etant structuré de manière à faciliter l'introduction de nouveaux services, ainsi il lui est facile de s'adapter aux futures applications d'images médicales.

- En se basant sur des normes internationales déjà existantes(le modèle *OSI* par exemple).

II-2.2 Historique

Le développement de la médecine et les méthodes d'exploration radiologiques et la qualité des diagnostiques de l'imagerie médicale d'une part, l'utilisation croissante des ordinateurs dans des applications cliniques dans les années 70 d'autre part, ont poussé l'American College of Radiology *(ACR)* et la National Electrical Manufactures Association (NEMA) à identifier la nécessité d'élaborer un standard qui permet de transférer des images et les informations associées entre les différents dispositifs construits par divers fournisseurs qui produisent, jusque-là, des images numériques de différent format.

Pour résoudre ce problème de l'interopérabilité entre ces dispositifs, La *(ACR)* et *(NEMA)* ont formé un comité mixte en 1983 pour développer une norme qui satisfait les besoins suivants : **[34].**

- Promouvoir la communication d'image numérique et leurs informations indépendamment du fabricant de matériels.

- Faciliter le développement et l'essor des systèmes d'archivage et de communication d'image (Picture Archivage and Communication System : *PACS*) qui peuvent également dialoguer avec d'autres systèmes de traitement de l'information.

- Permettre la création des bases de données qui peuvent être consultées par un grand nombre des médecins indépendamment de l'espace géographique.

Le standard *ACR-NEMA* avait publié en 1985, la version 1,0 de numéro 300-1985 de la publication de norme *d'ACR-NEMA*, La norme a été suivie de deux révisions: Numéro 1, daté octobre 1986 et numéro 2, daté janvier 1988.

Le standard ACR-NEMA encore fois en 1988 a édité la version 2,0, 300-1988 de la revue de leur publication de la norme *d'ACR-NEMA*, Inclut La version 1,0 et les éditions de révisions additionnelles. Elle contenait des nouveaux outils qui permettent de:

- Faciliter le support des dispositifs d'affichage.
- Présenter un nouveau schéma hiérarchique pour la représentation des informations relatives de l'image.
- Ajouter des données supplémentaires à l'image afin d'accroître sa spécificité.

Ces publications de la *ACR-NEMA* ont donnés une configuration matérielle et logicielle minimale qui facilitent l'exploitation correcte par les constructeurs.

II-2.3 La Version Actuelle

Cette norme prend un certaines améliorations par rapport aux versions précédentes de la norme *ACR-NEMA*. Parmi lesquelles : **[33]**

- Elle est applicable à un environnement géré en réseau. La norme d'*ACR-NEMA* était applicable seulement dans l'environnement point-à -point l'opération dans un environnement géré en réseau l'unité d'interface de réseau (Network Interface Unit *NIU*) a été exigée.
  La norme *DICOM* quant à elle, elle est capable de gérer les opérations dans un environnement connecté au réseau en utilisant le protocole de gestion du réseau standard *TCP/IP*.

- Elle est applicable en utilisant différents supports amovibles (*CDs,DVDs*…). Par contre Le standard *ACR-NEMA* ne définit pas de format du fichier et le choix de support physique ou de système de fichiers logique. *DICOM* supporte quant à elle l'utilisation des standards de l'industrie, physiques tels les *CD-R,* ou bien les systèmes de fichiers logiques, tel le *PC* file system *FAT 32*.

- Elle spécifie comment les dispositifs revendiquant la conformité à la norme réagissent aux commandes et aux données échangées. La norme *d'ACR-NEMA* a été limitée au transfert des données, mais *DICOM* spécifie, par le concept des classes de service (service classes), la sémantique des commandes et des données associées.

- Elle spécifie des niveaux de conformité. La norme d'*ACR-NEMA* avait spécifié un niveau minimum de conformité. *DICOM* décrit explicitement comment un constructeur doit structurer une déclaration de conformité (Conformance Statement) pour choisir des options spécifiques.

- Elle est structurée en plusieurs parties sur un document. Ceci facilite l'évolution de la norme dans un environnement en pleine évolution en simplifiant l'ajout de nouvelles fonctionnalités. Des directives d'*ISO* définissant la manière dont doit être structuré un document en plusieurs parties ont été suivies lors de la construction de la norme *DICOM.*

.

- Elle introduit des informations explicites pour des images et des graphiques mais également pour des formes d'onde, des rapports, l'impression, etc.

- Elle spécifie une technique établie pour l'identification unique de chaque information. Ceci facilite la définition sans ambigüité des relations entre les différentes informations.

Cette dernière version de la Norme qui a été maintenue, avec quelques modifications au fil des années pour s'adapter aux nouvelles fonctionnalités et capacités des dispositifs.

II-2.4 Le Comité *DICOM* (*DICOM Committee*)

Il s'agit d'un comité international qui associe les industriels de l'imagerie médicale et les sociétés professionnelles médicales connues également sous l'appellation de sociétés Savantes.

Le Comité *DICOM* dispose de trois secrétariats correspondant aux principales zones de répartition des industriels de l'imagerie médicale : Amériques, Japon et Europe.
Actuellement, on recense quelque vingt-deux industriels membres du comité. Les associations d'industriels en sont également membres :

- Le *NEMA* aux États-Unis, elle assure à la fois le secrétariat américain et le secrétariat général du Comité *DICOM*.

- Le *JIRA* (Japanese Industry Radiology Apparatus) au Japon, il assure le secrétariat de *DICOM* Japan,

- Le *COCIR* (European Coordination Committee of the Radiological and Electromedical Industry) en Europe qui assure le secrétariat d'Euro *DICOM*.
DICOM est un standard applicable à toute l'imagerie médicale et des sociétés Savantes de multiples disciplines en sont membres :

    - **Aux États-Unis:** *ACR* (American College of Radiology), *ACC* (American College of Cardiology), *AAO* (American Association of Ophtalmology), *AAD* (American Association of Dermatology), *ADA* (American Dental Association), *CAP* (College of Anatomo-Pathologists), *ASGE* (American Society of Gastro- Enterology).

- **En Europe:** *ESC* (European Society of Cardiology), *SFR* (Société Française de Radiologie), *DRG* (Deutsche RöntgenGeselschaft), *SIRM* (Società Italianadi Radiologia Medica).

Des experts de nombreux pays (*USA*, Japon, Allemagne, Pays-Bas, Belgique, France...) participent aux réunions du comité *DICOM* et de ses 21 groupes de travail (Working Group = *WG*).

II-2.5 Domaine

Le domaine de *DICOM* est l'imagerie médicale. Aujourd'hui, lorsqu'on parle d'image médicale on entend l'image et son environnement regroupés dans ce que l'on appelle l'acte d'imagerie médicale. Le cycle de vie de cet acte débute avec la rédaction de la demande d'examen par le clinicien et s'achève avec d'une part l'expédition des résultats au clinicien et d'autre part leur archivage. Durant tout ce cycle de vie, il devrait y avoir interaction entre le système d'information de santé et le matériel d'acquisition et de traitement des images. Ces considérations ont amené à définir ainsi le domaine de *DICOM*.

Si le domaine de *DICOM* est l'imagerie médicale numérique, pour réaliser son service, *DICOM* a besoin d'un certain nombre de technologies déjà normalisées par ailleurs (les aspects réseau et communication, compression, …). Lorsque des standards existent, *DICOM* se contente de notifier que pour telle ou telle problématique, il faut utiliser tel ou tel standard et ne fait que spécifier la manière dont les standards correspondants doit être utilisée dans le cadre d'applications *DICOM*. Ainsi à titre d'exemple, DICOM s'appuie sur le modèle *OSI* de l'*ISO* en ce qui concerne l'aspect communication, *JPEG* pour la compression etc.

II-2.6 Organisation des Données dans un Fichier DICOM

L'organisation de l'information contenue dans les fichiers *DICOM* a été inspirée par la sauvegarde des données sur des bandes magnétiques. L'information est organisée sous une forme séquentielle.

Chaque information élémentaire est constituée de 3 champs de données. Le premier champ est codé sur 8 octets, il s'agit d'une "balise" ou "tag'', répertoriée dans le dictionnaire *DICOM*, qui indique le type d'information qui va suivre. Le deuxième champ de 8 octets indique la longueur de l'information contenue dans les 3èmes champs, jusqu'à la balise suivante.

II-2.6 .1 Caractéristiques Principales de DICOM

La production quotidienne massive d'images médicales ne peut être archivée dans un format commun de type *JPEG* au risque de perdre des données associées à l'image telles que nom du patient, type d'examen, hôpital, etc… Le format *DICOM* permet de rendre unique chaque image produite et de lui associer des informations spécifiques. Cela a pour conséquence de produire des images autonomes dans la mesure où il est toujours possible d'identifier formellement leurs origines en cas de perte. Le format est de taille variable. Il contient des informations obligatoires et d'autres optionnelles.

Chaque image *DICOM* contient obligatoirement plusieurs types de numéros d'identification unique *UID* (Unique Identifier) générés automatiquement par les appareils. Il ne peut exister deux *UID* identiques pour désigner des informations différentes, et ceci quel que soit la machine et sa localisation. Cette unicité est nécessaire non seulement pour des raisons médico/médico-légal, mais aussi pour permettre la formation et la gestion de bases de données.

II-2 .6. 2 Principes du SOP

La norme *DICOM* est un langage orienté objet. Chaque objet *DICOM*, le plus souvent une image, contient à la fois des informations (nom du patient, pixels de l'image, etc..) et des fonctions (imprimer, sauvegarder, etc. ...) que doivent subir ces informations. Le traitement *DICOM* d'une information consiste donc à apparier un objet *DICOM* « Information Object » à une fonction spécifique « Service Class ». Cette combinaison est appelée « Service/Object Pair » ou « *SOP* ».

II-2 .6. 3 Format du Fichier DICOM

Le format de fichier *DICOM* fournit un moyen d'encapsuler l'ensemble des données représentées par une instance *SOP* relative à une définition de l'objet d'une classe *SOP*. Comme nous le montre la figure suivante. Les fichiers contenant une instance se succèdent et le tout constitue un ensemble de fichiers *DICOM*. Dans un fichier *DICOM*, les données sont organisées sous une forme séquentielle en commençant par un entête et suivi par des données brutes de l'image.

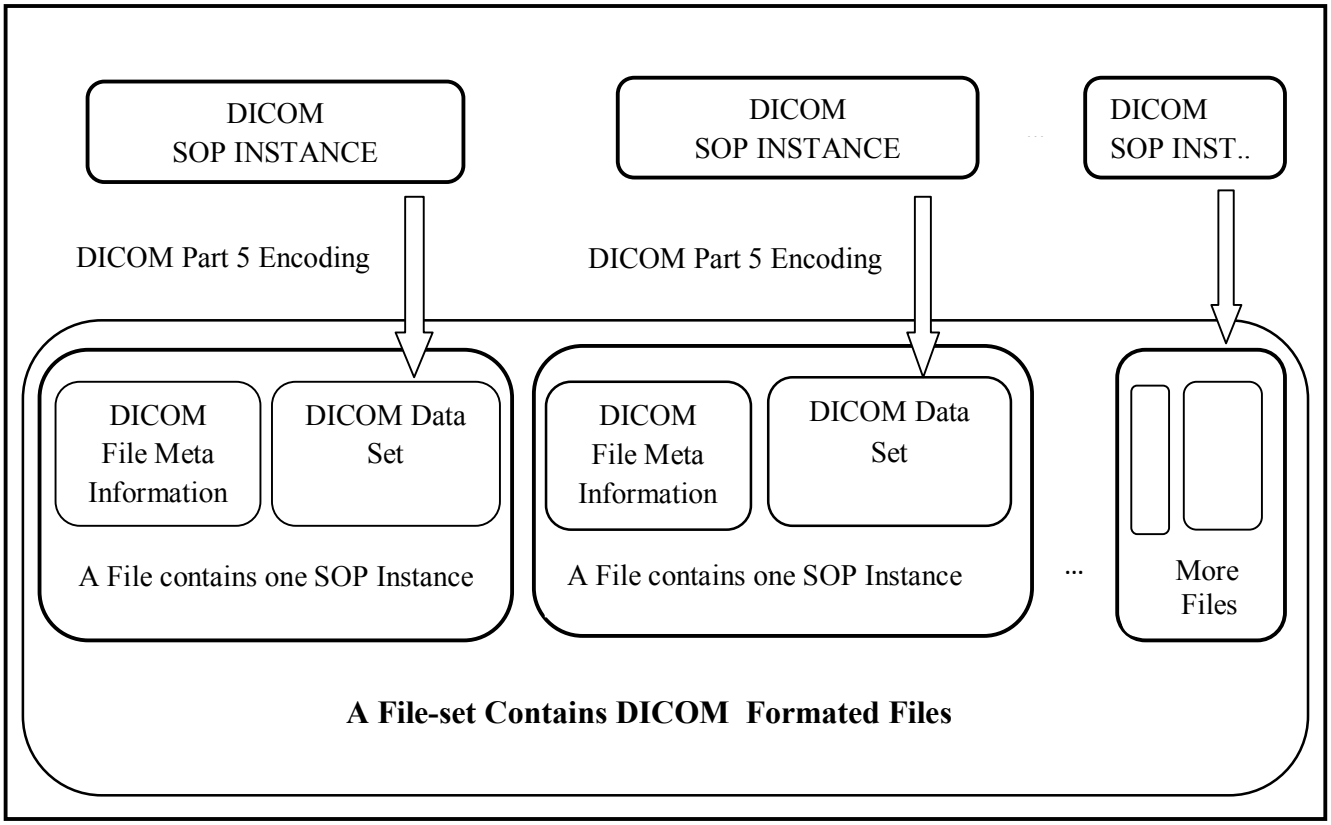


Figure II- 1. Ensemble de Fichiers DICOM.

- **L'entête**

Chaque entête commence par un préambule de 128 octets généralement mis à zéro suivi de 4 octets pour y inscrire les caractères 'D'.'I'.'C'.'M'. A la suite du préambule, toutes sortes d'informations se succèdent. Elles sont organisées en plusieurs groupes d'informations. Chaque élément de données est constitué de 3 champs de données si VR est implicite, et de 4 champs si VR est explicite. L'ensemble de ces données représente un dictionnaire de données *DICOM*. **[33].**

II-2.7 Objets DICOM

DICOM décrit de nombreux objets : des objets-images correspondant aux différents types d'images (e. g. objets scanner, *IRM*, ultrasons …), mais aussi des objets correspondant à l'environnement de l'image (e.g. objets signaux physiologiques pour les acquisitions synchronisées à l'*ECG* ou avec d'autres signaux. Le compte rendu structuré débouchant sur la gestion d'actes d'imagerie médicale).

Les objets sont décrits dans le standard sous l'appellation d'*IOD* (Image Object Definition). Un *IOD* comporte un certain nombre d'attributs permettant d'une part d'identifier le patient et d'autre part le décrivant.

II-2.8 Services DICOM

DICOM décrit également des services qui s'appliquent aux objets décrits dans le standard. Ces services servent à tester la connectivité, créer des objets, les stocker, les déplacer, les effacer Depuis début 2000 il existe des services de sécurité.

II-2.9 Associations service-objet

Enfin *DICOM* décrit les associations possibles entre objets et services. Ce sont les *SOP* Classes (Service Object Pair). Lorsqu'un matériel revendique sa compatibilité *DICOM*, sa description est basée sur les *SOP* Classes qu'il supporte. Une *SOP* Class peut être vue comme une phrase, *DICOM* décrit des phrases génériques. Lorsqu'on se trouve sur un équipement et que l'on demande l'exécution d'une opération sur un ou des objets DICOM, il y a génération d'une *SOP* instance qui peut être vue comme une phrase spécifique. **[35]**

DICOM est basé sur le modèle de communication *OSI* proposé par l'*ISO*. Lorsqu'un utilisateur demande un échange de données, il y a d'abord l'établissement d'une association entre les deux systèmes devant participer à cet échange. A l'occasion de l'établissement de cette association il y a sélection de la syntaxe d'échange et échange de la liste des *SOP* classes supportées par chacun des deux systèmes. Il y a ensuite exécution de la *SOP* instance demandée par l'utilisateur. Enfin il y a clôture de l'association

II-2.9.1. Les Classes de Service Actuellement Disponibles dans La Norme DICOM

Le traitement *DICOM* d'une information consiste à apparier un objet *DICOM* (par exemple une image) à une fonction spécifique ou *Service* à appliquer à cet objet (imprimer, sauvegarder, etc..).

Ainsi la combinaison d'un " Information Object" (par ex une image) avec un "Service" (par exemple l'impression de cette image) est appelée *:* ***Service/Object Pair*** *ou* ***SOP***.

Information Object + Service Class = Service /Object Pair ou SOP

Ou encore, par exemple

Une Image +Son impression= Un service DICOM

Cette parité Information/Service est l'élément principal de la conformité à la norme. Une machine en conformité à la norme doit gérer au minimum une classe de parité Service/Objet et ne pas seulement émettre ou recueillir des fichiers d'images à la norme *DICOM*.

Pour se conformer à une Classe de Parité Objet/Service (*SOP* Class) une machine (ou toute entité applicative *DICOM*, Application Entity *AE*) doit pouvoir gérer un type particulier d'image et réaliser un type spécifique de traitement (ou service) correspondant à la définition de cette Classe de Parité. DICOM définit toutes les classes de parités possibles**. [36].**

Par ailleurs la Classe de Parité Objet/Service doit spécifier si le service *DICOM* est employé en tant qu'utilisateur (Service Class User ou *SCU*) ou en tant que fournisseur (Service Class Provider *SCP*). Exemple : Un scanner utilise le service que lui fournit le reprographe, le scanner est alors doté d'une Classe de Service Utilisateur *SCU* pour reprographe. Le reprographe de son côté est doté d'une Classe de Service Fournisseur, SCP pour le scanner.

Classes de Service	Type de Service
Vérification (Verification Service Class)	Utilisé pour les tests, permet de savoir si les machines "s'entendent" mutuellement, cette classe n'est pas associée à un objet *DICOM*, elle renvoie l'information sous la forme d'un écho. (*C-ECHO*)
Storage (Storage Service Class) Media Storage Service Class	Permet le transfert et la sauvegarde des images entre deux entités *DICOM*. (*CR, CT, MR* Storage Service Class). Il existe une variante : Media Storage Service Class qui spécifieles échanges entre 2 machines par l'intermédiaire d'un média (CD rom , disquettes etc...)
Query/Retrieve	Implémente des commandes types : *FIND, MOVE, GET* . *FIND* permet de demander une liste d'image, *MOVE et GET* permettent d'initier un transfert,qui sera en réalisé effectué via la classe"Storage Service Class"
Study Contents Notification	Utilisée pour notifier l'arrivée d'une nouvelle image ou série d'images, peut être utilisée pour initier un transfert ou vérifier si le transfert d'une série d'image est complet.
Print Management	Permet la connexion avec un reprographe, spécifie le type d'image, (Couleurs, niveaux de gris etc...)
Patient Management	Permet d'interfacer la machine au réseau hospitalier *PACS* ou *HIS/RIS* (*Hospital Information service* / *Radiological information service*) Gestion des données des patients, démographie, admission et sortie des patients
Study Management	Création, gestion de rendez-vous, suivi des examens.
Result Management	Permet la gestion des résultats des examens.

Tableau II- 1. Table des Classes de Service actuellement disponibles dans la norme DICOM.

II-2.9.2 Réalisation Pratique de la Connexion de Deux Machines

Lorsqu'une entité applicative (un scanner) possède un Objet *DICOM* (une image à reproduire), si elle est en conformité *DICOM* pour utiliser la classe de service d'impression (Service Class User of Printer), alors elle s'adresse à une Entité Applicative (le reprographe) qui fournit une classe de service d'impression correspondante (Service Class Provider for *CT*).Il en résulte alors une phase de négociation pendant laquelle les machines se mettent d'accord sur le type de données à échanger et la façon dont elles vont les échanger.(par exemple, pour un mot de deux octets, elles se mettent d'accord pour transmettre l'octet le plus significatif en premier).

Cette phase permet d'établir le '*Presentation Context*', la façon dont est échangée l'information (la *syntaxe de transfert*) est appelée Value Représentation (*VR*) , si l'octet le plus significatif est transmis en premier , il s'agit d'une VR *Little Endian* dans le cas inverse on parle de VR *big endian.*

Les différentes syntaxes de transfert proposées par les Entités *DICOM* sont répertoriées dans les Documents de Conformité.

II-3 Conclusion

En résumé, un fichier *DICOM* comprend les données correspondantes aux pixels de l'image, habituellement, ces données sont regroupées à la fin du fichier *DICOM*, elles sont précédées par d'autres fichiers techniques.

La norme *DICOM* est orientée objet, cela signifie que chaque objet *DICOM* (le plus souvent une image) contient à la fois les informations (le nom du patient, les pixels de l'image...) et les méthodes (ou fonctions) que doit subir cette information.

Dans la version actuelle de *DICOM*, les équipements communiquent soit sous la forme d'un réseau *TCP/IP*, ou *OSI*, ou encore par média (*CD* rom, disquettes, bandes magnétiques etc..)**.[36].**

DICOM permet de visualiser la couleur et le noir et blanc. La couleur peut être transmise sous la forme d'une palette de couleur (Look up table) ou d'un codage direct. Le noir et blanc peut être codé sur une profondeur de 8 à 32 octets. *DICOM* supporte tous les types connus de compression.

Dans le prochain chapitre, on va décrire les méthodes de compression qui seront appliquées par la suite sur cette norme d'image.

Chapitre III

Méthodes Appliquées

Chapitre III : Méthodes Appliquées

III -1 Introduction

La compression d'une image numérique a pour but de réduire le nombre de bits qui la représente, pour ce faire, elle fait appel à différentes méthodes de compression. Selon les qu'elles peuvent être avec perte ou sans perte. Ce chapitre présente les différents algorithmes étudiés.

III- 2 Méthode Sans Perte

Les deux algorithmes *RLE* et *LZW* sont Appliqués dans ce projet permettent de reconstituer les valeurs exactes des pixels de l'image originale. La partie suivante inclut le principe de fonctionnement de chacun.

III-2.1 Algorithme RLE (Run Length Encoding)

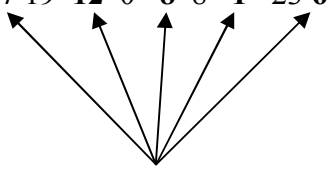
Les pixels voisins dans une image de norme *DICOM* sont fortement corrélés entre eux. Souvent, on observe que les pixels consécutifs dans une région d'une image sont identiques ou la variation parmi les pixels voisins est très petite. Le codage par plage *(RLE)* est une forme très simple de la compression de données, son principe est de constituer un code pour chaque élément de données suivi par un nombre de répétions successives. C'est le plus utile sur les données qui contiennent des zones identiques : par exemple, images binaires ... Il n'est pas utile avec les fichiers qui n'ont pas beaucoup de répétions car le fichier pourrait doubler de volume.

- **Exemple**

Soit le vecteur $\boldsymbol{v}$ suivant :
$\boldsymbol{v}$ = 5 5 5 5 5 5 5 19 19 19 19 19 19 19 19 19 19 19 19 0 0 0 0 0 0 0 0 8 23 23 23 23 23 23.

Le codage (RLE) de vecteur $\boldsymbol{v}$ nous donne le code suivant :

RLE code ($\boldsymbol{v}$) = 5 **7** 19 **12** 0 **8** 8 **1** 23 **6**



Nombre de répétions

L'opération de décodage *RLE* est une simple réécriture de fichier code en remplaçant le nombre de répétition par la valeur qui la précède. **[7].**

III-2.2 Algorithme LZW (Lempel-Ziv-Welch)

Est un algorithme de compression sans perte. Il s'agit d'une amélioration de l'algorithme *LZ78* inventé par Abraham Lempel et Jacob *Ziv en 1978.LZW* fut créé en 1984 par Terry Welch, d'où son nom.

LZW dans la compression d'image est comme d'autres méthodes de compression qui code une suite des symboles, il peut coder n'importe quelles données exprimées comme une suite des octets. En effet, il est largement répandu pour coder des données d'image, particulièrement dans le GIF standard, où chaque valeur de pixel est assignée jusqu' à un octet. **[37].**

L'algorithme a été conçu de manière à être rapide à implémenter, mais n'est la plupart du temps pas optimal car il effectue une analyse limitée des données à compresser. Pour comprendre bien le fonctionnement de l'algorithme on va présenter un exemple :

L'exemple de *LZW* sert à coder une phrase dans un texte, L'algorithme de compression construit une table de *traduction* de *chaînes de caractères* à partir du texte à compresser. Cette table relie des codes de taille fixée (généralement 12-bits) aux chaines de caractères. La table est initialisée avec tous les caractères (256 entrées dans le cas de caractères codés sur 8 bits). A mesure que le compresseur examine le texte, il ajoute chaque chaîne de 2 caractères dans la table en tant que *concaténation* de code et de caractères, avec le code correspondant au premier caractère de la chaîne. En même temps qu'il enregistre ces chaines, le premier caractère est envoyé en sortie.

Chaque fois qu'une chaîne déjà rencontrée est lue, la chaîne la plus longue déjà rencontrée est déterminée, et le code correspondant à cette chaîne avec le caractère concaténé (le caractère suivant du flux entrant) est enregistré dans la table. Le code pour la partie la plus longue de la chaîne de caractère rencontré est envoyée en sortie et le dernier caractère est utilisé comme base pour la chaîne suivante.

L'algorithme de décompression a seulement besoin du texte compressé en entrée. En effet il reconstruit une table chaine de *caractères / code identique* à mesure qu'il régénère le texte original. Quand l'algorithme de décompression lit le code pour *caractère/chaîne/caractère*, il ne peut pas le traiter car il n'a pas encore stocké ce code dans la table. Ce cas particulier peut être géré car le programme de décompression sait que le caractère supplémentaire est le *précédent caractère rencontré.*

- **Algorithme de Compression LZW**

w = ***Nul***;

Tant que *(lecture d'un caractère* ***c****)* ***Faire***
Si *(****wc*** *existe dans le dictionnaire)* ***Alors***
w = wc;
Sinon
Ajouter ***wc*** *au dictionnaire;*
Écrire *le code de* ***w****;*
w = c*;*
Fin Si
Fin tant que
Écrire *le code de* ***w****;*

- **Algorithme de Décompression LZW**

Lecture d'un caractère ***c****;*
Écrire c*;*
w = c*;*
Tant que *(lecture d'un caractère* ***c****) faire*
Si *(****c*** *> 255 && l'index* ***c*** *existe dans le dictionnaire)* ***faire***
Entrée *= l'entrée du dictionnaire de* ***c****;*
Sinon si *(****c*** *> 255 && l'index* ***c*** *n'existe pas dans le dictionnaire)* ***faire***
Entrée *=* ***w*** *+* ***w*** *[0];*
Sinon
Entrée *=* ***c****;*
Fin si*;*
Écrire ***Entrée****;*
Ajouter ***w****+****Entrée [0]*** *au dictionnaire;*
w = Entrée;

Fin tant que*;*

- ***Exemple de la* Compression /Décompression LZW**

La table suivante montre le résultat de l'exécution de l'algorithme de compression *LZW* sur la chaîne suivante « *COMPRESSIONDECOMPRESSION »*

Caractère (C)	w	WC [W&C]	Sortie (code)	Dictionnaire
C	<NIL>	C		
O	C	C**O**	C	CO=<256>
M	**O**	O**M**	O	OM=<257>
P	**M**	M**P**	M	MP=<258>
R	**P**	P**R**	P	PR=<259>
E	**R**	R**E**	R	RE=<260>
S	**E**	E**S**	E	ES=<261>
S	**S**	S**S**	S	SS=<262>
I	**S**	S**I**	S	SI=<263>
O	**I**	I**O**	I	IO=<264>
N	**O**	O**N**	O	ON=<265>
D	**N**	N**D**	N	ND=<266>
E	**D**	D**E**	D	DE=<267>
C	**E**	E**C**	E	EC=<268>
O	**C**	**CO**		
M	**CO**	CO**M**	<256>	COM=<269>
P	**M**	**MP**		
R	**MP**	MP**R**	<258>	MPR=<270>
E	**R**	**RE**		
S	**RE**	RE**S**	<260>	RES=<271>
S	**S**	**SS**		
I	**SS**	SS**I**	<262>	SSI=<272>
O	**I**	**IO**		
N	**IO**	IO**N**	<264>	ION=<273>
	N	N	N	N=<274>

Tableau III- 1. Exemple de Codage LZW.

Après la compression LZW, nous obtenons la séquence de code:
LZW (code)= COMPRESSIONDE<256><258><260><262><264>N

➢ **Décompression LZW**

Le résultat de décompression de la séquence compressée est présenté dans le tableau suivant :

k(le Code)	*w*	*Entry*	*W+Entry [0]*	*Sortie*	*Dictionnaire*
C	*C*			*C*	
O	*O*	*O*	***CO***	*O*	CO=<256>
M	*M*	*M*	***OM***	*M*	OM=<257>
P	*P*	*P*	***MP***	*P*	MP=<258>
R	*R*	*R*	***PR***	*R*	PR=<259>
E	*E*	*E*	***RE***	*E*	RE=<260>
S	*S*	*S*	***ES***	*S*	ES=<261>
S	*S*	*S*	***SS***	*S*	SS=<262>
I	*I*	*I*	***SI***	*I*	SI=<263>
O	*O*	*O*	***IO***	*O*	IO=<264>
N	*N*	*N*	***ON***	*N*	ON=<265>
D	*D*	*D*	***ND***	*D*	ND=<266>
E	*E*	*E*	***DE***	*E*	DE=<267>
<256>	*CO*	*CO*	***ECO***	*CO*	EC=<268>
<258>	*MP*	*MP*	***COMP***	*MP*	COM=<269>
<260>	*RE*	*RE*	***MPRE***	*RE*	MPR=<270>
<262>	*SS*	*SS*	***RESS***	*SS*	RES=<271>
<264>	*IO*	*IO*	***SSIO***	*IO*	SSI=<272>
N	*N*	*N*	***ION***	*N*	*ION=<273>*
			N		N=<274>

Tableau III- 2. Exemple de Décodage LZW.

La chaine décompressée par la méthode *LZW* est la concaténation des caractères de la sortie de tableau de décompression au-dessus, après l'examinassions de l'exemple *LZW*, on reconstitue la chaine *« COMPRESSIONDECOMPRESSION »*

III-3 Méthode avec Perte

La méthode de compression *SPIHT*, basée sur l'utilisation des ondelettes et la notion de codage progressif. On va présenter dans ce qui suit l'atout donnée par les ondelettes au codage d'image par cette méthode.

III -3.1 Transformée en Ondelettes d'une Image

La transformée en ondelettes discrète *(2D-DWT)* est une opération qui décompose un signal (une série d'´échantillons numériques) en deux parties par projection sur un filtre passe-bas *L* et un filtre passe-haut *H*. La partie résultante du filtrage passe-bas représente une approximation

du signal d'origine à la nouvelle résolution (ou échelle), et celle résultante du filtrage passe-haut représentant les détails perdus entre les deux résolutions. **[39].**

Puisqu'une image est typiquement un signal en deux dimensions, une *(2D-DWT)* est réalisée, en appliquant d'abord les filtres *L* et *H* sur les échantillons ligne par ligne, puis en réappliquant les mêmes filtres sur les échantillons résultants, mais colonne par colonne cette fois-ci. Au final, l'image est divisée en *4* parties, les sous-images *LL, LH, HL* et *HH* comme présenté sur la *figure (a)*. La sous-image *LL* fournit une version à l'échelle *1/2* de l'image d'origine, *LH, HL* et *HH* représentant les détails perdus respectivement dans les directions horizontale, verticale et diagonale. La *(2D-DWT)* peut être réitérée sur *LL* pour obtenir plusieurs niveaux de résolution. La *figure (b)* présente une image décomposée en deux niveaux de résolution.

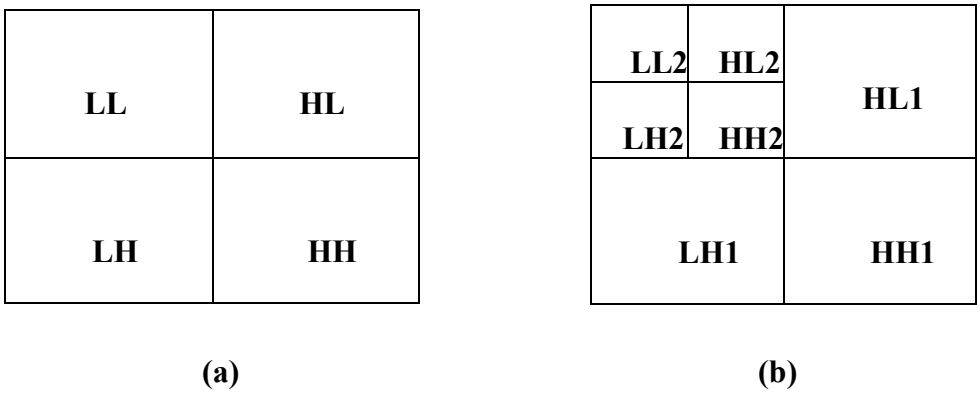


Figure III- 1. La (2D-DWT) Appliquée une fois (a) ou deux (b).

La *(2D- DWT)* donc est appliquée sur une image pour présenter les pixels par d'autres coefficients (coefficients d'ondelettes) qui se sont regrouper sous forme des sous bandes, ces coefficients sont coudés par la suite par la méthode SPIHT qu'on a utlisée dans ce projet. Les principales caractéristiques de la méthode sont données dans le paragraphe suivant.

III-3.2 Méthode SPIHT (Set Partitioning in Hierarchical Trees)

L'algorithme *SPIHT* est l'amélioration de la version de l'algorithme *EZW (Embedded Zerotree Wavelet*). Il a été présenté par *SAID* et *PEARLMAN*. **[25**]

On désigne les pixels de l'image originale I_O par $p_{i,j}$, et quel que soit les filtres d'ondelettes choisis, l'image transformée **C** est donnée par l'équation suivante :

$$\mathbf{C}= (\boldsymbol{2D-DWT})\ (\boldsymbol{I_O}) \cdots\cdots\cdots\cdots\cdots\cdots\cdots (\boldsymbol{III.1})$$

La *(2D-DWT)* fait présenter les pixels de l'image I_O par des coefficients d'ondelettes $\boldsymbol{c_{i,j}}$, Ces derniers constituent l'image transformée c **.**

SPIHT est une méthode de transmission progressive des coefficients d'ondelettes $c_{i,j}$. Dans le décodage, la technique commence par l'initialisation des coefficients $\hat{c}$ de l''image reconstruite à zéro. Ces coefficients sont améliorés progressivement pour produire d'un nouveau $\hat{c}$ amélioré. Ces derniers coefficients sont utilisés pour produire une meilleure image reconstruite I_r:

$$\boldsymbol{I_r = (2D - IDWT)(\hat{c})} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \boldsymbol{(III.2)}$$

Le but principal du codage progressif est de transmettre en premier l'information d'image la plus importante. Cette information transmise ainsi réduit la déformation.

Pour évaluer la qualité d'image reconstruite par rapport à l'image originale, l'algorithme SPIHT utilise la mesure d'Erreur Quadratique Moyenne *(MSE)* défini par :

$$\textbf{MSE}= \frac{1}{m\times n} \sum_{i=1}^{m} \sum_{j=1}^{n} ||\boldsymbol{I_O(i,j) - I_r(i,j)}||^2 \cdots\cdots\cdots \boldsymbol{(III.3)}$$

Ou :

$$\boldsymbol{MSE = \frac{1}{m\times n} \sum_{i=1}^{m} \sum_{j=1}^{n} ||c_{i,j} - \hat{c}_{i,j}||^2} \quad \cdots\cdots\cdots\cdots\cdots\cdots \boldsymbol{(III.4)}$$

L'équation $(\boldsymbol{III.4})$ nous montre que la valeur de *MSE* diminue près de $|c_{i,j}|^2 / m.n$ quand le décodeur reçoit le coefficient de transformation ondelette $c_{i,j}$ **,** (supposent que la valeur exacte du coefficient est reçue par le décodeur, i.e. il n'y a aucune perte de précision due aux limitations imposées par les opérations arithmétiques d'ordinateur). Il est maintenant clair que les plus grands coefficients $c_{i,j}$ (plus grand en valeur absolue, indépendamment de leurs signes) contiennent l'information qui réduit plus la déformation MSE. **[40]**

À la reconstruction d'image, la transmission des coefficients d'ondelettes par ordre bit-plane a le même comportement que la déformation obtenue par les transmettre par ordre décroissante d'amplitude, mais la première méthode de transmission présente deux propriétés additionnelles qui sont : **[**41**]**

- Pour chaque itération de triage *(Sorting Pass)* des coefficients par amplitude est suivie par une opération de transmission de bits d'information sur ces coefficients.

- L'erreur de reconstruction est réduite au minimum puisque le bit significatif est le premier transmis.

Dans le spectre d'ondelette de multi-résolution, il existe un rapport de similitude individuelle et spatiale entre les coefficients des niveaux et entre les différentes sous-bandes de fréquence. Ce rapport spatial est défini dans une pyramide hiérarchique construite avec la quatre-sous-bande récursive se dédoublant, comme représenté sur la figure suivante: Un coefficient $\boldsymbol{c_{i,j}}$ dans la représentation en ondelette a quatre descendants directs (enfants) (offsprings) aux endroits : *O (i, j) = {(2i ,2j); (2i, 2j +1) ; (2i+1, 2j) ; (2i+1, 2j +1)},* et chacun d'eux maintient

périodiquement une similitude spatiale à ses quatre enfants correspondant. Que le genre de structure de pyramide est généralement connu en tant qu'arbre d'orientation spatiale.

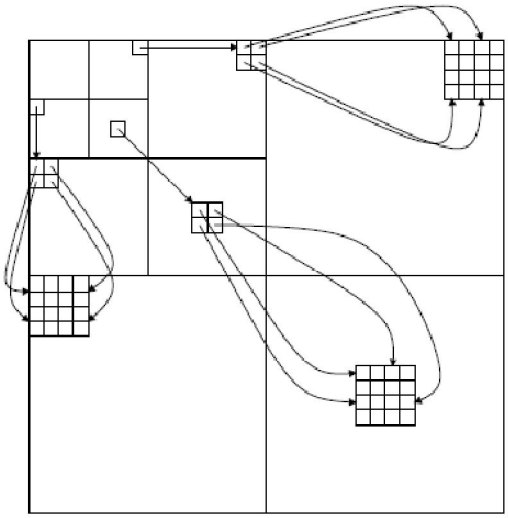


Figure III- 2. Exemples des dépendances de parent-enfant dans l'orientation des arbres spatiaux créés dans la 2D -DWT.

- L'algorithme SPIHT est basé sur l'utilisation de la similitude spatiale dans l'espace d'ondelette pour comprimer efficacement les coefficients en bit plane, *la figure III.2* montre la similitude entre les sous-bandes et aux niveaux adjacents.

- *SPIHT* est une méthode de compression rapide et efficace d'image, elle sert à tester les coefficients d'ondelettes dans un ordre décroissant de bit, et quantifie seulement des coefficients significatifs. L'efficacité obtenue par cet algorithme est élevée, elle est due à un test d'ensemble des coefficients qui appartiennent à un arbre d'ondelette. Le test d'ensemble a un avantage en raison de la corrélation d'inter-bande existant entre les coefficients appartenant à un arbre.

- L'algorithme *SPIHT* regroupe les coefficients et les arbres d'ondelette en ensembles suivant des tests de signification. **[41]**

- L'algorithme SPIHT utilise trois listes de coefficients : la liste des coefficients significatifs (List of Significant Pixels ou *LSP*), la liste des coefficients non significatifs (List of Insignificant Pixels ou LIP) et la liste des ensembles non significatifs (List of Insignificant Sets ou LIS). $O(x,y)$est l'ensemble des enfants de (x,y) (un seul niveau de descendance), $D(x,y)$ est l'ensemble de tous les descendants et $L(i, j) = D(i, j) - O(i', j')$ est l'ensemble des descendants à l'exception des enfants .

- La fonction $S_t\ (i,j)$ est égale à **0** si tous les descendants de (x,y)sont en dessous du seuil Tt (arbre de zéros) et **1** dans le cas contraire.

Considérons deux types d'arbres de zéros : le type A où tous les descendants ne sont pas significatifs (arbre de degré 1) et de type *B* où tous les descendants, à l'exception d'au moins un des enfants, ne sont pas significatifs (arbre de degré 2). Dans les deux cas , la valeur du coefficient de la racine est significative.

L'algorithme *SPIHT* complet est décrit comme suit :

Initialisation :

– **t**=$|log_2(max_{(i,j)}\{|c_{i,j}|\}|$

– Pour chaque plan de bits **t** :

– $\boldsymbol{LSP} = \emptyset$

– $\boldsymbol{LIP}$: tous les coefficients sans parent (coefficients de la LL)

– $\boldsymbol{LIS}$: tous les coefficients de la $\boldsymbol{LIP}$ qui ont des descendants (marqués comme type A, par défaut)

Sorting Pass :

$$S_t(\mathrm{T})=\begin{cases} 1, & max_{(i,j)}|c_{i,j}| \geq 2^t \\ 0 & Ailleurs \end{cases}$$

Pour chaque coefficient (i,j) de la $\boldsymbol{LIP}$

– Ecrire $\boldsymbol{S_t}(i,j)$.

– **Si** $\boldsymbol{S_t}(i,j)$ = 1, déplacer (i,j) dans la LSP et écrire le signe de **c** (i,j).

Pour chaque coefficient (i,j) de la $\boldsymbol{LIS}$

– **Si** le coefficient est de **type A**

– Ecrire $\boldsymbol{S_t}$**(D**(i,j))

– **Si** $\boldsymbol{S_t}$(D(i,j)) = 1 alors

– Pour tout (i', j') **appartient** O (i,j) : Ecrire $\boldsymbol{S_t}$ (i', j') ; **si** $\boldsymbol{S_t}(i', j')$ = 1, Ajouter (i', j') à la **LSP** et Ecrire le signe de **c** (i', j') **sinon**, ajouter (i', j') à la fin de la **LIP**.

– $\boldsymbol{Si}$ **L**$(i,j) \neq \emptyset$, déplacer (i, j) à la fin de la **LIS** comme une entrée de **type B**

–$\boldsymbol{Sinon}$, retirer (i, j) de la $\boldsymbol{LIS}$

– **Si** l'entrée est de **type B**

– **Ecrire** $\boldsymbol{S_t}$(T) (L(i, j))

– Si $\boldsymbol{S_t}$ (L(i, j)) = 1

– Ajouter tous les (i', j') **appartient** à O(i, j) à la fin de la **LIS** comme entrée de **type A**

– Retirer (i, j) de la **LIS**.

Refinement pass :

Pour tous les coefficients (i,j) de la **LSP** à l'exception de ceux ajoutés au cours de la dernière sorting pass : Ecrire le $t^{ième}$ bit le plus significatif de c (i,j).

Décrémenter $\boldsymbol{t}$ et retourner à la **Sorting pass.**

Le décodeur est obtenu en remplaçant **Ecrire** par **lire** dans l'algorithme précédent

- **Exemple de Compression SPIHT**

La technique est appliquée pour coder les coefficients du tableau suivant :
La première étape sert à déterminer le coefficient le plus grand.

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

- **First Pass :** Dans ce cas **t** = 4.
2^4=16 seuil.
Initialisation des listes
LIP : Contient {(0,0) → 26, (0.1) → 6, (1,0) → -7, (1,1) → 7}
LIS : Contient {(0,1) D, (1,0) D, (1,1) D}
LSP : ∅ (ensemble vide)

On examine le contenu de la liste ***LIP***

Le coefficient à l'endroit (**0, 0**) est 26 plus grand que le seuil **16**. Donc, il est significatif; alors, nous transmettons un **1** suivi d'un **0** pour indiquer que le coefficient est positif, on déplace **26** à ***LSP***. Les trois prochains coefficients **(6,-7, 7)** sont tous insignifiants à cette valeur du seuil; donc, nous transmettons un **0** pour chaque coefficient et on les laisse dans la ***LIP***.

La prochaine étape est de tester le contenu de l'ensemble ***LIS***
On examine les descendants du coefficient à l'endroit **(0, 1)**, qui sont **(13, 10,6, et 4)**, on remarque qu'aucun d'eux n'est significatif à cette valeur du seuil donc on transmet un **0**.
On examine les ensembles des descendants c_{10} et c_{11}, on voit qu'aucun de ces derniers n'est significatif à cette valeur du seuil. Donc, on transmet un **0** pour chaque ensemble.

SPIHT dans le *First Pass* ne fait rien dans le l'étape de ***Refinement Pass.***

Donc à la fin de ce passage on transmet au décodeur un total de **8** bits (**10000000**),

Donc l'état des trois listes devient :

LIP*:* {(0, 1) → 6, (1, 0) → -7, (1, 1) → 7}
LIS*:* {(0, 1) D), (1, 0) D), (1, 1) D)}
LSP*:* {(0, 0) → 26}

- **Second Pass :**

Pour la ***Second Pass*** on décrémente ***t*** de **1**, il devient **3**, cette valeur correspond à une valeur du seuil est égale à **8.**
On recommence notre passage en examinant le contenu de la *LIP*. Il y a trois éléments dans la *LIP,* qui sont **(6,-7,7)**.

Aucun de ces coefficients ne dépasse pas le seuil, donc on transmet trois **0**.
La prochaine étape est d'examiner le contenu de l'ensemble *LIS*.
Le premier élément de *LIS* est l'ensemble contenant les descendants du coefficient à l'endroit (0, 1). Dans cet ensemble, **13** et **10** sont significatifs à cette valeur du seuil; en d'autres termes, l'ensemble ***D (0, 1)*** est significatif. On signale ceci en envoyant un ***1*** et examinons les progénitures de c_{01} ; La première progéniture (enfant) a une valeur de **13**, qui est significative et positive, donc on envoie un **1** suivi d'un **0**. Le même est vrai pour la deuxième progéniture, qui a une valeur de **10**. Ainsi nous envoyons encore **1** suivi d'un **0**.
Nous déplaçons les coordonnées de ces deux coefficients au ***LSP***. Les deux prochaines progénitures (enfants) sont les deux insignifiantes à ce niveau; donc, nous déplaçons ces derniers à la ***LIP*** et transmettons un **0** pour chacun. Comme **£** (0, 1) = 0 ; nous enlevons (0, 1) D de ***LIS***. On examine les autres éléments de ***LIS***, on voit que tous les deux insignifiant à ce niveau; donc, nous envoyons un **0** pour chacun.

Dans le passage ***Refinement Pass*** on examine le contenu de ***LSP*** du passage précédent. Il y a seulement un élément dans le premier passage ***Sorting Pass,*** et il a une valeur de 26. Le troisième MSB de 26 est **1**; donc, on transmet un **1** pour accomplir ce passage.

Le code généré par cette deuxième itération composé de **13** bits : (**0001101000001).**

L'état des listes à la fin de la deuxième passe est comme suit:

LIP: {(0, 1) → 6, (1, 0) → - 7, (1, 1) → 7, (1, 2) → 6, (1, 3) → 4}
LIS: *{(1,0) D,* (1, 1) D}
LSP: {(0, 0) → 26, (0,2) → 13, (0, 3) → 10}

- ***Third Pass***

La troisième *passe* se poursuit par **t = 2**. Le seuil est maintenant est **4** plus, il y a un nombre important des coefficients qui sont considérés significatifs, on suit la même étape de test et code généré est composé de **26 bits**. Vous pouvez facilement vérifier pour assurer vous-même que le train de bit généré transmis pour la troisième ***Third Pass*** est (**10111010 101101100110000010).**

L'état des listes à la fin é de la troisième passe est comme suit:

LIP: {(3, 0)→ 2, (3,1) →-2, (2,3) → -3, (3,2) → -2, (3,3) → 0}
LIS: ∅.
LSP: {(0, 0) → 26, (0,2) → 13, (0,3) → 10, (0,1) → 6, (1,0) → -7, (1,1) → 7, (1,2)→ 6, (1, 3) → 4, (2,0) → 4, (2, 1) → -4, (2,2) → 4}

- **Décodage SPIHT**

Pour décoder le train de bits. Le décodeur SPIHT procède d'utiliser les mêmes listes que l'encodeur

Initialisation des Listes

***LIP**:* {(0, 0), (0, 1), (1,0), (1, 1)}
***LIS**:* {(0, 1) **D**, (1, 0) **D**, (1, 1) **D**}
***LSP**:* ∅

Nous supposons que la valeur initiale de t est **4**, elle est transmise au décodeur. Ceci nous permet de placer la valeur seuil à **16**.
Quand le décodeur voit les résultats de la *First Pass* **(10000000)**, il peut constater que le premier élément de la ***LIP*** est significatif et positif et aucun autre coefficient est significatif à ce niveau. On peut reconstruire les coefficients à ce seuil comme :

Et, après le même procédé que l'encodeur, les listes peuvent être mises à jour comme suit :

24	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

LIP: {(0, 1), (1,0), (1, l)}
LIS: {(0, 1) **D**, (1, 0) **D**, (1, 1) **D**}
LSP: {(0, 0)}

- **Second Pass** on décrémente **t** par **1** et on examine le train de bits (bitstream) transmis: **(0001101000001).**

Puisque les **3** premiers bits sont à **0** et il y a seulement trois entrées dans la ***LIP,*** toutes les entrées dans la LIP sont encore insignifiantes. Les **9** prochains bits nous donnent des informations sur les ensembles dans **LIS**. Le quatrième bit du train de bit (bitstream) reçu est **1**. Ceci signifie que l'ensemble avec la racine à la coordonnée **(0.1)** est significatif. Puisque cet ensemble est de type **D**, le prochain bit se relie à ses progénitures (enfants). Les bits **101000** indiquent que les deux premières progénitures (enfants) sont significatives à ce niveau et positifs et les deux derniers sont insignifiants. Par conséquent, on déplace les deux premières progénitures (enfants) à **LSP** et les deux derniers à la **LIP**, donc on reconstruit ces deux coefficients significatifs par $\mathbf{1{,}5 \times 2^3 = 12}$. On enlève également (0, 1) **D** du **LIS**. Les deux prochains bits sont les deux **0**, indiquant que les deux ensembles restants sont encore insignifiants. Le bit final correspond au ***Refinement Pass*** est 1.

Ainsi nous mettons à jour la reconstruction du coefficient (0, 0) qui devient 24 + 8/2 = 28.

La reconstruction est à cette étape est :

28	0	12	12
0	0	0	0
0	0	0	0
0	0	0	0

Et les listes deviennent comme suit:

LIP*:* {(0, 1), (1, 0), (1, 1), (1, 2), (1, 3)}
LIS*:* {(1, 0)D, (1, 1)D}
LSP*:* {(0, 0), (0,2), (0,3)}

- **Third Pass**

On décrémente **t**, qui est maintenant **2**, donnant une valeur- seuil de ***4***. On décode de la manière que le passage précédent génère le code pendant la ***Third Pass*** **(10111010101101100110000010),**

Nous mettons à jour de notre reconstruction qui devient :
Et les listes deviennent :

26	6	14	10
6	6	6	6
6	-6	6	0
0	0	0	0

LIP: {3, 0), (3, 1)}
LIS: *{ }*
LSP: **{(**0, 0), (0,2), (0,3), (0, 1), (1,0), (1,1), (1,2), (2,0), (2, 1), (3, 2)}

À cette étape nous n'avons aucun ensemble dans ***LIS*** et nous mettons à jour simplement les valeurs des coefficients. **[17]**

III-4 Conclusion

Dans ce chapitre, nous avons exposé l'état de l'art des méthodes de compression qui nous allons appliquées par la suite. La première méthode est sans perte, comporte deux algorithmes : *RLE, LZW*. La deuxième est avec perte, basée sur la notion d'encodage progressif décrit par l'algorithme *SPIHT*. Ces algorithmes de compression sont appliqués sur des fichiers *DICOM*. Le chapitre prochain présente les résultats obtenus pour chacun.

Chapitre IV

Résultats et Interprétations

Chapitre IV : Résultats et Interprétations

IV -1 Introduction

Dans ce chapitre, on présente les résultats de l'application des méthodes utilisés pour la compression des fichiers *DICOM.*

La première méthode de compression est sans perte, sert à coder les pixels des images suivant les algorithmes de compression *RLE et LZW*. Ces derniers réversibles restituent exacte de l'image originale. Pour évaluer l'efficacité de la méthode, on a utilisé : le rapport de compression *(RC),* le temps de compression *(TC)* et de décompression comme paramètres de mesure des performances de la méthode sans perte.

La deuxième méthode est avec perte, vise à coder des fichiers *DICOM* , leur principe est basé sur l'outil des ondelettes, les résultats de la décomposition sont codés progressivement suivant les techniques décrites par l'algorithme *SPIHT*. Et durant l'opération de compression, on va étudier l'effet des fonctions d'ondelette sur la qualité des images reconstruite et donner les principales caractéristiques des filtres d'ondelettes utilisés. Nous avons utilisé le rapport maximal de signal-bruit *(PSNR)* et sa variation par rapport au débit binaire Bit par pixel (Bpp) pour l'évaluation objective des performances de la méthode avec perte. On utilise l'indice *MSSIM* en fonction de (Bpp) pour estimer la similitude structurelle entre l'image originale et l'image décompressée.

Les programmes sont exécutés avec un PC portable des caractéristiques : Processeur : *Intel(R) Pentium(R) Dual (CPU) TC 2390 @186 GHz, Mémoire vive 2Go.*

Une conclusion de ce chapitre inclut l'ensemble des évaluations des techniques de compression utilisées qui sont appliquées sur des fichiers médicaux de Norme *DICOM.*

IV-2 Méthode Sans Perte

Elle consiste en 2 algorithmes :

- ***RLE*** *(Run Lenght Encoding).*
- ***LZW*** *(Lempel-Ziv-Welch)*

IV-2.1 Compression / Décompression RLE

Les deux schémas suivants présentent les différents blocs des chaines de compression et de décompression du fichier *DICOM* en utilisant l'algorithme *RLE*.

IV-2.1.1 Compression RLE

La compression RLE des fichiers DICOM est effectuée selon le schéma blocs suivant :

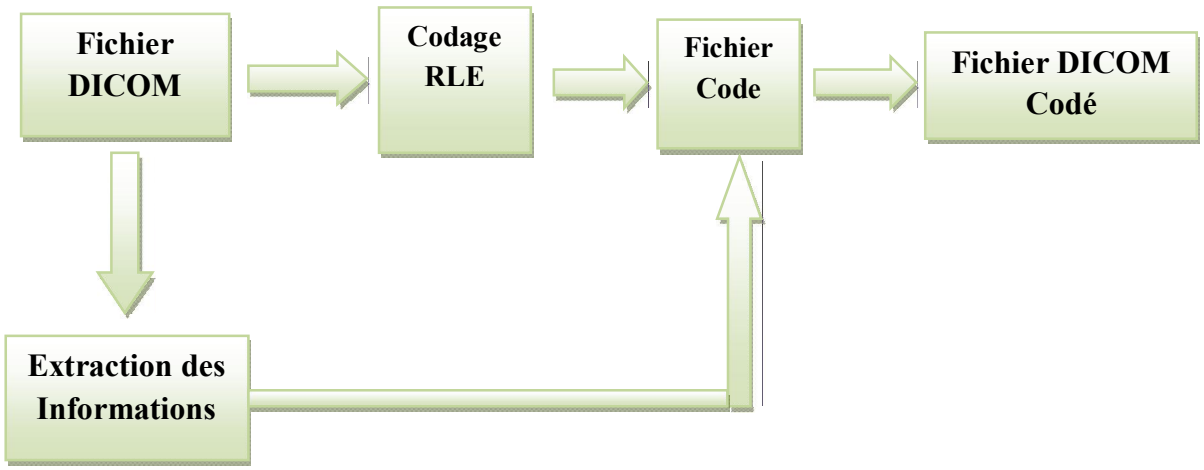


Figure IV- 1. Schéma Général de la Compression RLE.

Les étapes de cette chaine de compression sont expliquées comme suit :

Etape 1 : Séparer l'image et les informations

La séparation du fichier *DICOM* de l'image et des informations relatives est faite à l'aide des commandes *MATLAB* « dicomread », qui a comme sortie l'image contenue dans le fichier *DICOM* d'entrée, sous forme d'une matrice de dimension égale à la résolution de l'image et « dicominfo» qui a comme sortie toutes les informations relatives au fichier *DICOM* d'entrée sous forme d'une structure.

Etape 2 : Codage RLE

La commande *MATLAB* « dicomread » ressortit l'image contenue dans le fichier *DICOM* sous forme d'une matrice. On transforme cette matrice en un vecteur, en juxtaposant ces vecteurs l'un après l'autre et ce vecteur par la suite est codé par l'algorithme *RLE*. Il est nécessaire de

sauvegarder la dimension de l'image avant le codage pour la reconstruire après l'étape de décodage.

Etape 3 : Construire le Fichier Code

Le résultat du codage est un vecteur qui contient le code généré par l'algorithme *RLE* et sauvegarde la valeur de chaque pixel suivie directement par une valeur entière indiquant le nombre de répétitions successives de chaque pixel de l'image.

Etape 4 : Construire le Fichier Code DICOM

On construit le fichier *DICOM* codé en utilisant la commande *MATLAB* « dicomwrite » qui sert à sauvegarder les mêmes informations du fichier *DICOM* original.

IV-2.1.2 Décompression *RLE*

L'opération de décompression *RLE* est réalisée selon le schéma bloc suivant :

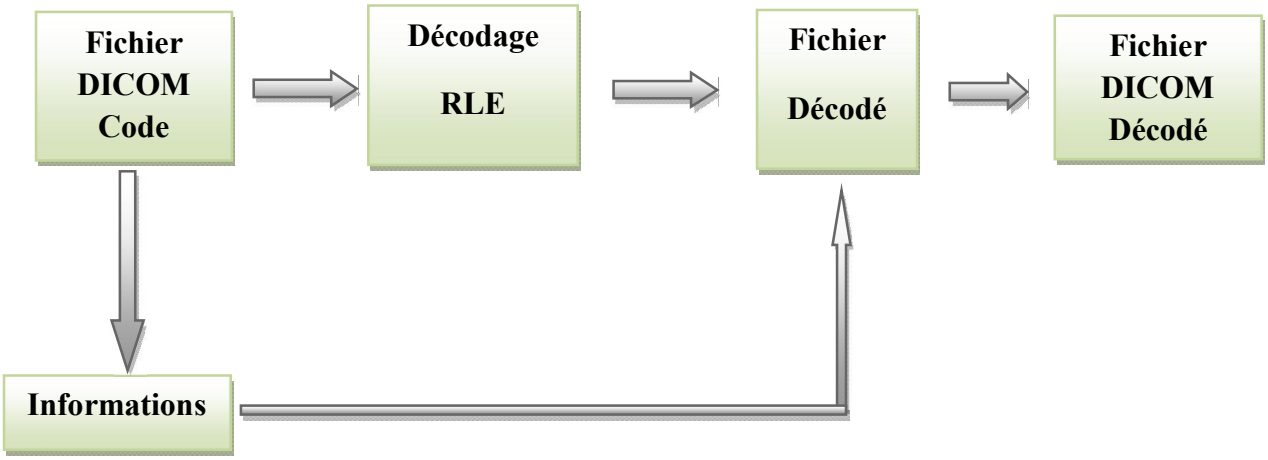


Figure IV- 2. Schéma Général de la Décompression RLE.

Les étapes de cette chaine sont expliquées comme suit :

Etape 1 : Extraction des informations

Avant de procéder à l'étape de décodage *RLE,* il est nécessaire de séparer les informations relatives du fichier *DICOM* original et le code généré par le codage *RLE.*

Etape 2 : Décodage RLE

Une fois la séparation faite entre les informations relatives et l'image *DICOM*. Le décodage RLE est une simple réécriture du vecteur en remplaçant les redondances par les valeurs des pixels.

Etape 3 : Reconstruction du Fichier DICOM

Cette étape sert à transformer le vecteur généré par l'étape *2* en une matrice dont la dimension est égale à la dimension de l'image originale. La commande *MATLAB* « dicomwrite » permet de sauvegarder les mêmes informations du fichier *DICOM* original.

Dans ce qui suit, on va présenter le deuxième algorithme de compression sans perte.

IV-2.2 Compression /Décompression LZW

Les deux schémas suivants présentent les différents blocs des chaines de compression et de décompression du fichier *DICOM* en utilisant l'algorithme *LZW*.

IV-2.2.1 Compression LZW

Le schéma en bloc suivant présente la chaine de compression *LZW*

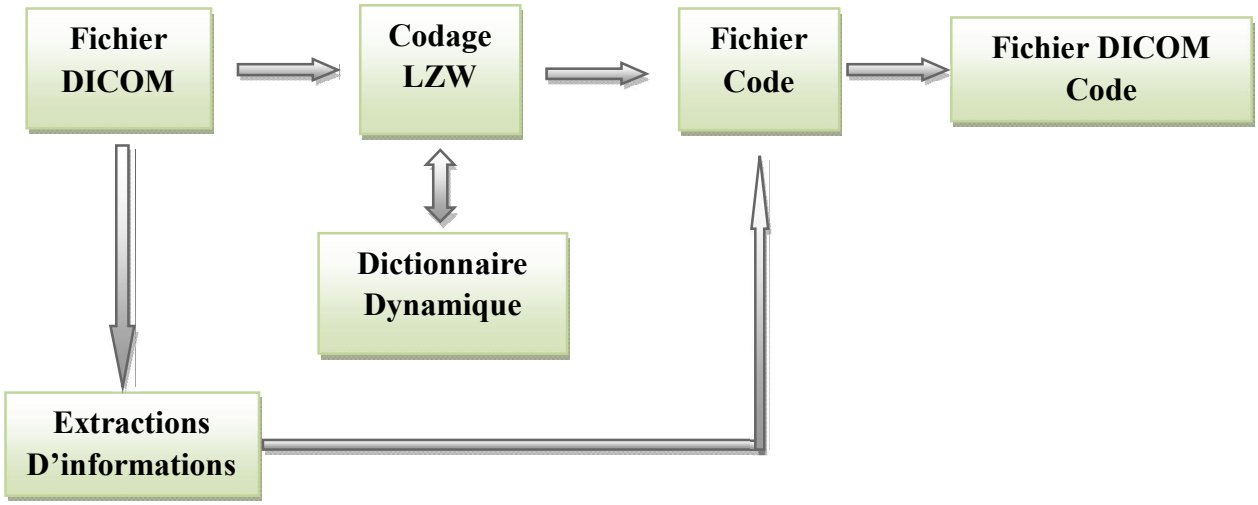


Figure IV- 3. Schéma Général de la Compression LZW.

Etape 1 : Séparation de l'image et les informations

Cette étape est identique à l'étape 1 de la chaine de compression *RLE*.

Etape 2 : Codage LZW

La commande *MATLAB* « dicomread » présente l'image sous forme d'une matrice, ensuite on la transforme en un vecteur. Le codage *LZW* sert à enregistrer les combinaisons des symboles dans un dictionnaire construit au cours du codage. Une séquence qui réapparait est codée par son indice dans le dictionnaire. La Compression *LZW* comme a été présentée au chapitre précédent sert à générer un code et une table de translation (dictionnaire). Le dictionnaire est construit d'une manière dynamique.

Etape 3 : Construction du Fichier Code

A l'aide de la commande *MATLAB* « dicomwrite », on sauvegarde le fichier codé avec les informations relatives du fichier *DICOM* originale.

IV-2.2.2 Décompression LZW

Le schéma en bloc suivant présente la chaine de décompression *LZW*.

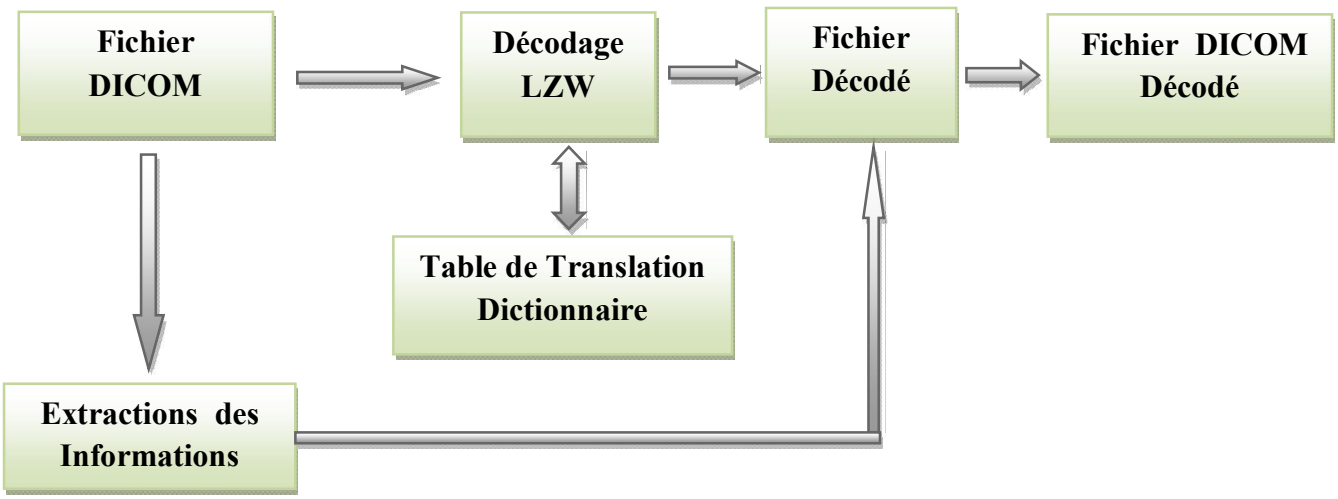


Figure IV- 4. Schéma Général de la Décompression LZW.

Etape 1 : Extraction des Informations

Cette étape est identique à l'étape de la chaine de décompression *RLE* expliquée dans la partie précédente.

Etape 2 : Décodage LZW

Une fois la séparation a été faite entre les informations relatives de l'image *DICOM* originale et le vecteur code *LZW,* on applique le procédé de décodage *LZW* qui va reconstruire le

dictionnaire dans le sens inverse d'une manière dynamique, et il remplace les indices des séquences envoyés par les valeurs exactes des pixels originaux.

Etape 3 : Reconstruction du Fichier DICOM

Cette étape sert à transformer le vecteur généré par l'étape 2 en une matrice de dimension égale à la dimension de l'image originale. La commande *MATLAB* « dicomwrite » permet de sauvegarder les mêmes informations du fichier *DICOM* original.

IV-2.3 Résultats de la Compression Sans Perte

Dans cette partie, on présente les performances des algorithmes de compression *RLE* et *LZW*. Les performances des algorithmes sont mesurées objectivement par : *le taux de compression*, *les temps de compression* et *de décompression. Les tests* ont été effectués sur un ensemble de fichiers *DICOM* de différentes résolutions.Les deux tableaux suivants contiennent l'ensemble des résultats obtenus dans l'application des deux algorithmes :

IV-2.3.1 Résultats de l'Algorithme RLE

FICHIER DICOM	***Taille Originale [Octets]***	***Taille de code [Octets]***	***TEMPS de Compression En [Seconde]***	***TEMPS de Décompression En [seconde]***	***Rapport Compression***	***Taux de Compression***
US-PAL-8-10x-echo	2580000	796264	129.21	3754.39	3.24	69.14 %
CT-MONO-16-ankle	525436	231024	3.38	14.15	2.26	55.94 %
CT-MONO2-8-abdo	262144	187038	3.99	7.21	1.40	28.65 %
MR-MONO2-8-16x-heart	1048576	995956	218.84	452.62	1.05	5.02 %

Tableau IV- 1. Performances de la Compression RLE.

IV-2.3.2 Résultats de l'Algorithme LZW

FICHIER DICOM	*Taille Originale [Octets]*	*Taille de Code en [Octets]*	*Temps de Compression [Seconde]*	*Temps de Décompression [Seconde]*	*Rapport de Compression*	*Taux de Compression*
US-PAL-8-10x-echo	2580000	721384	76.02	21.82	3.5765	72.04 %
CT-MONO2-8-abdo	262144	151532	8.57	3.43	1.7300	42.20 %
OT-MONO2-8-a7	262144	214944	9.08	4.39	1.2196	18.01 %
OT-MONO2-8-colon	262144	259166	8.97	5.02	1.0115	01.14 %

Tableau IV - 2. Performances de la Compression LZW.

D'après les tests effectués d'algorithme *RLE* sur un ensemble des fichiers *DICOM,* nous avons constaté que cette méthode donne de bons résultats de compression pour des fichiers qui présentent des zones identiques, par contre, donne de moins bons résultats dans le cas contraire ,car les temps de compression et de décompression calculés deviennent plus long lorsque la taille du fichier grandit.

Le codeur *LZW* a donné de bons résultats, car il code les séquences d'image sous des tailles plus réduites. On remarque que le temps de leur exécution est plus court.

IV-2.3.3 Résultats de Comparaison RLE/LZW

Le tableau suivant nous montre les résultats de comparaison entre les performances des algorithmes LZW et RLE testés sur deux fichiers DICOM :

FICHIER DICOM	*Algorithmes*	*Taille Originale [Octets]*	*Taille de Code [Octets]*	*Temps de Compression [Seconde]*	*Temps de Décompression [Seconde]*	*Rapport de Compression*	*Taux de Compression*
US-PAL-8-10x-echo.	**LZW**	2580000	721384	76.02	21.81	**3.5765**	**72.04 %**
	RLE	2580000	796264	129.21	3754.39	**3.2401**	**69.14 %**
CT-MONO2-8-abdo.	**LZW**	262144	151532	8.57	3.43	**1.7300**	**42.20 %**
	RLE	262144	187038	4.00	7.22	**1.4016**	**28.65 %**

Tableau IV - 3. Comparaison des performances entre LZW et RLE.

On remarque que le temps d'exécution de *LZW* est plus court que celui de *RLE*. En conclusion nous avons constaté que les rapports de compression donnés par la méthode sans perte en utilisant les algorithmes *LZW* et RLE sont inferieurs à 4, d'où la limite de cette méthode. Pour remédier à ces insuffisances, nous allons exposer la deuxième méthode qui est avec perte.

IV-3 La Méthode avec Perte

Cette méthode sert à coder les pixels des fichiers médicaux de norme *DICOM* suivant le procédé du codage progressif décrit par les algorithmes *SPIHT, EZW*.

IV-3.1 Compression du fichier DICOM par SPIHT

Le diagramme en bloc qui suit schématise la chaine de compression du fichier DICOM.

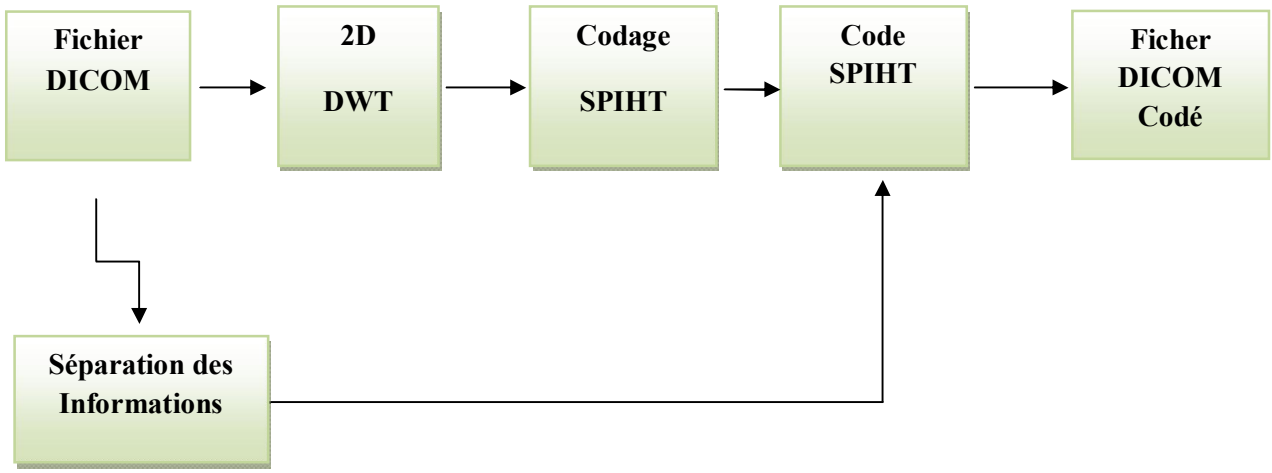


Figure IV- 5. Schéma Général de la Compression SPIHT.

Les étapes de cette chaine sont expliquées comme suit :

Etape 1 : <u>Séparation entre l'Image et les Informations</u>

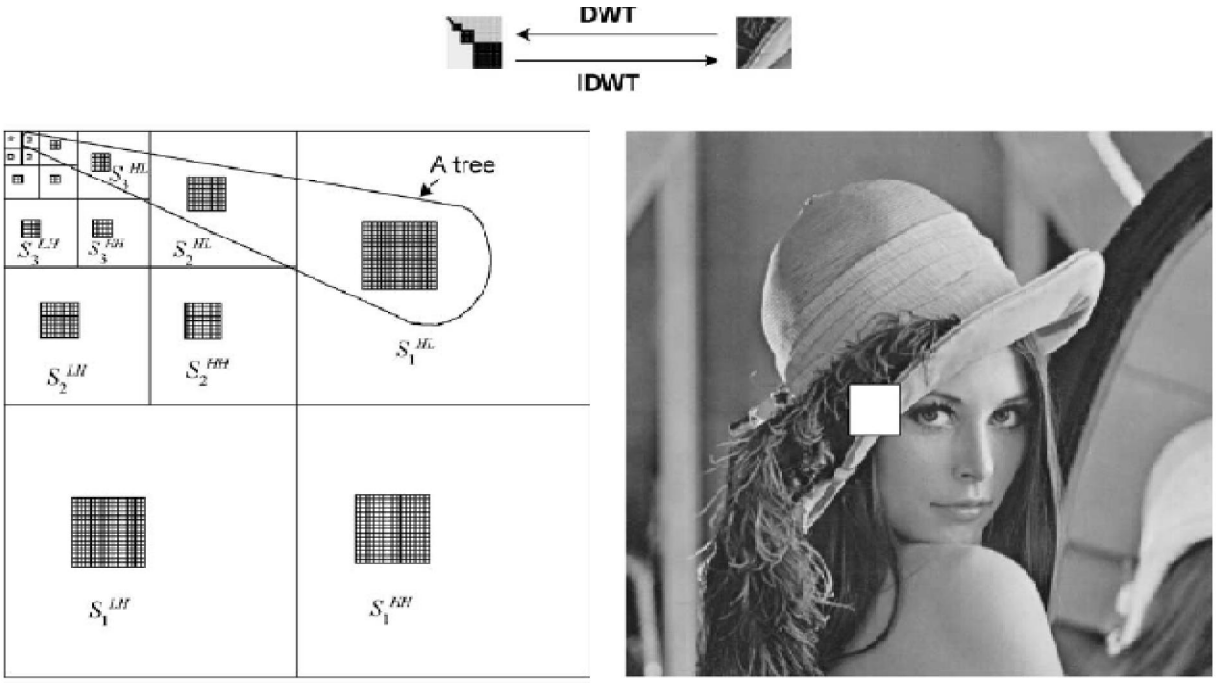
On extrait du fichier *DICOM* l'image et les informations, ceci est fait à l'aide des commandes *MATLAB* « dicomread » qui a comme sortie l'image contenue dans le fichier *DICOM* d'entrée sous forme d'une matrice de dimension égale à la résolution de l'image, et « dicominfo» qui a comme sortie toutes les informations relatives au fichier *DICOM* d'entrée sous forme d'une structure.

Etape 2 : <u>La Transformée (2D-DWT)</u>

On applique la transformée en ondelette (2D-DWT) sur l'image DICOM qui a été extraite à l'étape 1. Avant de procéder au codage nous allons présenter l'intérêt de l'utilisation de la transformée *(2D-DWT)* sur la compression des images par la méthode *SPIHT*. Parmi ces avantages, on peut citer **[43] :**

- L'opération de décomposition de l'image en ondelette exploite deux filtres linéaires : l'un passe -bas et l'autre passe - haut. Cette opération peut être réitérée plusieurs fois sur l'image créant des régions (ou des sous-bandes).

  Elle est faite en utilisant des fonctions implémentées sous *MATLAB* telles que *« dwt2 »* et *« wfiltrer »* qui permet de faire le choix du type des filtres d'ondelette de décomposition et de reconstruction utilisés et le choix de niveau de décomposition désiré.



Domaine Ondelette Domaine Image

Figure IV- 6. Structures Hiérarchiques (2D-DWT).

La figure précédente montre la structure hiérarchique (pyramidale) dans le domaine de la décomposition en ondelette. Un arbre se compose de coefficients d'ondelettes (nœuds) et de tous ses descendants. Les arbres sont formés des coefficients de la même orientation (indiquée par de petites cases du même modèle). L'union de trois arbres de différentes orientations s'appelle un arbre total. Un arbre carré correspond à un bloc carré dans le domaine d'image (bloc blanc dans figure à droite). Les blocs correspondants dans les domaines d'ondelettes et d'image sont montrés sur la *figure IV.6.* L'orientation de S_1 représente une sous-bande de « l » niveau de décomposition. L'orientation $S_l^{orietation}$ représente un sous-bande de niveau décomposition l et une des trois orientations *(LH, HL, HH)*. **[42].**

En général, quel que soit le type de filtre d'ondelette choisi, la répartition de l'énergie de l'image est comme suit :

- La grande partie de l'énergie dans l'image est concentrée dans les régions de basses fréquences, et les régions de hautes fréquences contiennent les informations des détails (contours, texture …).[**44].**

- La *(2D-DWT)* permet aussi de présenter les pixels par d'autres coefficients appartenant à l'espace transformé. i.e. convertir les pixels de l'image en coefficients d'ondelettes.

  Ces coefficients sont regroupés suivant le nombre de décomposition en ondelette sous forme des sous bandes, des bandes approximatives et autres présentant les détails de l'image.

- La *(2D -DWT)* présente aussi la possibilité de rendre l'énergie dans les bandes à haute fréquence concentrée dans un nombre relativement petit de coefficients.

La étape de compression *SPIHT* est appliquée par la suit pour coder ces coefficients ondelettes suivant leur technique.

Etape 3: Le Codage SPIHT (Set-Partitioning in Hierarchical Trees)

Le principe de base de la méthode *SPIHT* est de : classer les coefficients d'ondelettes par amplitude et par la suite les transmettre des bits du plus significatifs aux moindres. Pour classer les coefficients, la technique *SPIHT* utilise la base de corrélations inter bandes existantes entres les coefficients d'ondelettes. Dans l'étape de classement des coefficients la technique de codage regroupe les coefficients selon des seuils successifs (plan de bits s'appelle *Sorting Pass* **,** Quand une valeur- seuil est indiquée, elle divise les coefficients ou les ensembles de coefficients dans des coefficients significatifs ou insignifiants. Les coefficients significatifs sont ajoutés à la liste des pixels significatifs*(LSP).* Les coefficients non significatifs sont ajoutés à la liste des pixels non significatifs *(LIP)* ou à la liste des ensembles non-significatifs *(LIS).*

Dans l'étape $Refinement\ pass$: Pour tous les coefficients (i,j) de la *LSP* à l'exception de ceux ajoutés au cours de la dernière Sorting Pass : Ecrire le $\boldsymbol{t^{ième}}$ bit le plus significatif de ***c (i,j).*** [Pour plus de détails voir le chapitre III].

Le résultat de cette opération est un train binaire qui représente le code généré par le codage *SPIHT.*

Etape 4 : Construction du fichier Codé

La sortie du codeur *SPIHT* est un fichier binaire qui comporte les résultats du test de signification par rapport à un seuil donné et des bits d'amélioration des pixels significatifs. Pour assurer un décodage de notre image, il nous faut ajouter à ce fichier des informations qui indiquent la taille, le niveau de décomposition en ondelette, le nombre de bits par pixel, le type des filtre ondelettes utilisé et le type de la méthode de compression utilisée. La commande *MATLAB* « dicomwrite » permet de sauvegarder l'image codée avec les mêmes informations relatives du fichier *DICOM* original.

IV-3.2 Décodage D'Image DICOM par l'algorithme SPIHT

Le schéma en bloc qui suit montre le déroulement de la chaine de décompression d'image *DICOM.*

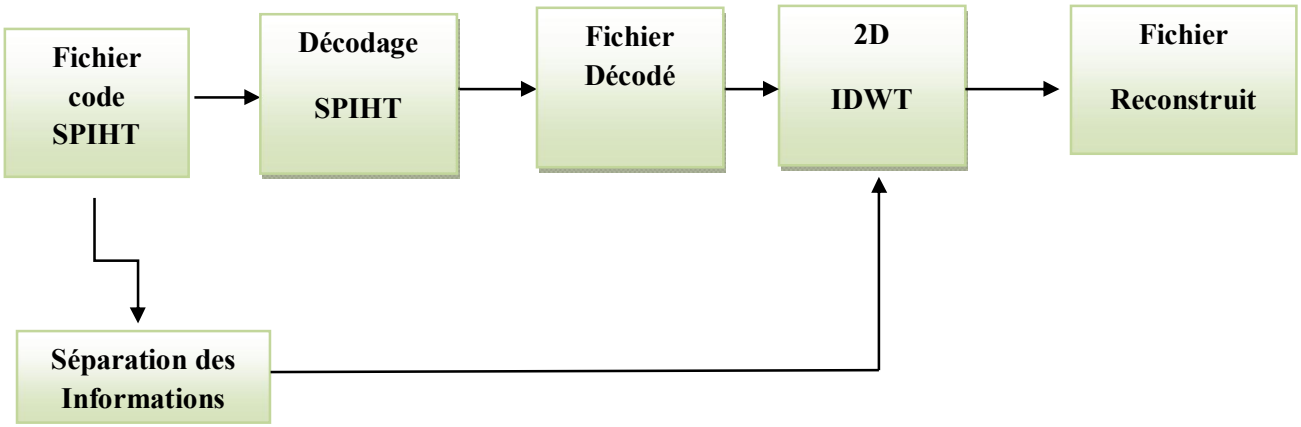


Figure IV- 7. Schéma Général de la Décompression SPIHT.

Les étapes de cette chaine de décompression sont expliquées comme suit :

Le code final généré à la sortie de la chaine de compression *SPIHT* précédente est un fichier qui comporte deux parties. La première s'appelle l'entête, contient les informations relatives de fichier *DICOM*. La deuxième partie contient le code généré pendant le codage *SPIHT*.

- **Séparation des informations**

Cette étape sert à séparer l'image codée par *SPIHT* des informations relatives du fichier *DICOM*. On applique par la suite à l'image codée, la technique de décompression *SPIHT*. Par contre les informations relatives sont les mêmes conservées pendant l'étape 1 de la chaine de compression.

- **Décodage SPIHT**

Le décodage est une opération de reconstruction des coefficients d'ondelettes. Il initialise tous les coefficients à zéros et après pour chaque plan de bit, il y aura une mise à jour (amélioration) de ses coefficients. Les trois listes *LSP, LIP, LIS* sont initialisées comme l'étape de codage *SPIHT*. [L'exemple donné au chapitre précédant montre bien son fonctionnement].

- **Reconstruction du fichier Décodé**

Le fichier décodé contient le résultat obtenu du décodage *SPIHT* et les informations relatives de l'algorithme de compression. Ce fichier, présente l'image reconstruite dans l'espace transformé en ondelettes.

- **Transformée en Ondelette Inverse (2D-IDWT)**

Cette opération permet de faire le passage du domaine transformée en ondelettes à l'espace réel de l'image *DICOM* en appliquant les deux filtres d'ondelette inverses de reconstruction. Le résultat de cette opération est l'image décompressée finale de l'algorithme *SPIHT*.

IV-3.3 Résultats de l'Algorithme SPIHT

Nous avons appliqué l'opération de Compression/Décompression *SPHIT* sur des fichiers de norme *DICOM*. Pour ce faire, nous avons réalisé plusieurs expériences :

Expérience 1

Le tableau suivant contient les résultats d'application *SPIHT* sur *l'image 'OT-MONO2-8-colon' :* image *DICOM* de dynamique 8 bit par pixel et de taille 512×512.

Rapport de Compression [CR]	***Bit Rate [Bpp]***	***PSNR [dB]***	***MSSIM***
80	0.1	28.31	0.7800
40	0.2	31.70	0.8553
26.66	0.3	34.00	0.8922
20	0.4	35.59	0.9118
16	0 .5	36.92	0.9263
13.33	0.6	37.72	0.9342
11.42	0.7	38.44	0.9408
10	0.8	39.00	0.9493
8.88	0.9	39.76	0.9547
8	1	40.34	0.9589
7.27	1.1	40.84	0.9623
6.66	1.2	41.37	0.9664
6.15	1.3	41.96	0.9701
5.71	1.4	42.58	0.9742
5.33	1.5	43.28	0.9793
5	1.6	43.94	0.9823
4.70	1.7	44.65	0.9841
4.44	1.8	45.21	0.9857
4.21	1.9	45.82	0.9873
4	2	46.50	0.9891

Tableau IV - 4. Table des performances de l'Algorithme SPIHT sur l'image « OT-MONO2-8-colon.dcm ».

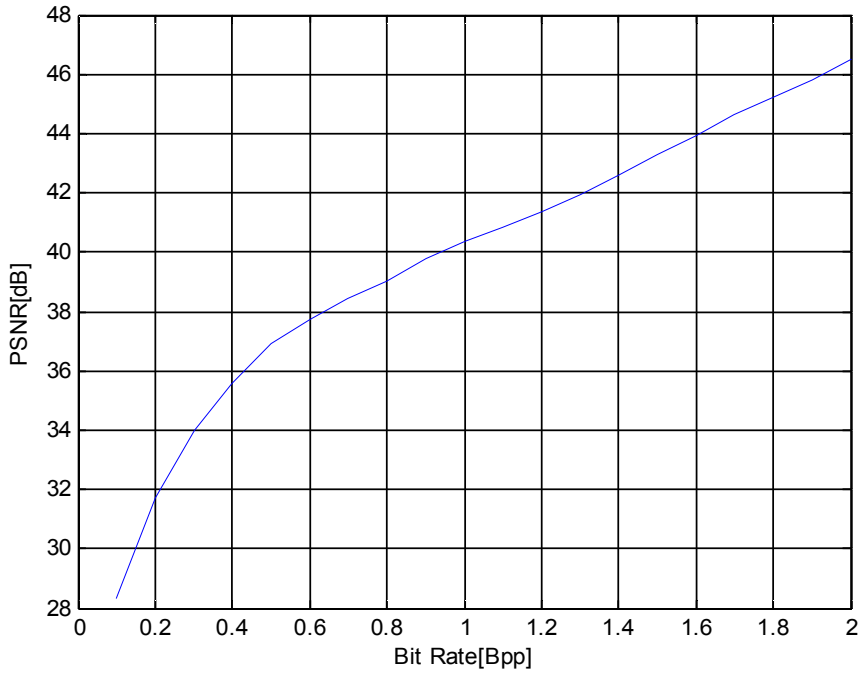


Figure IV- 8. Rapport Signal-à-Bruit pic en fonction du débit pour l'image OT-MONO2-8-colon (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

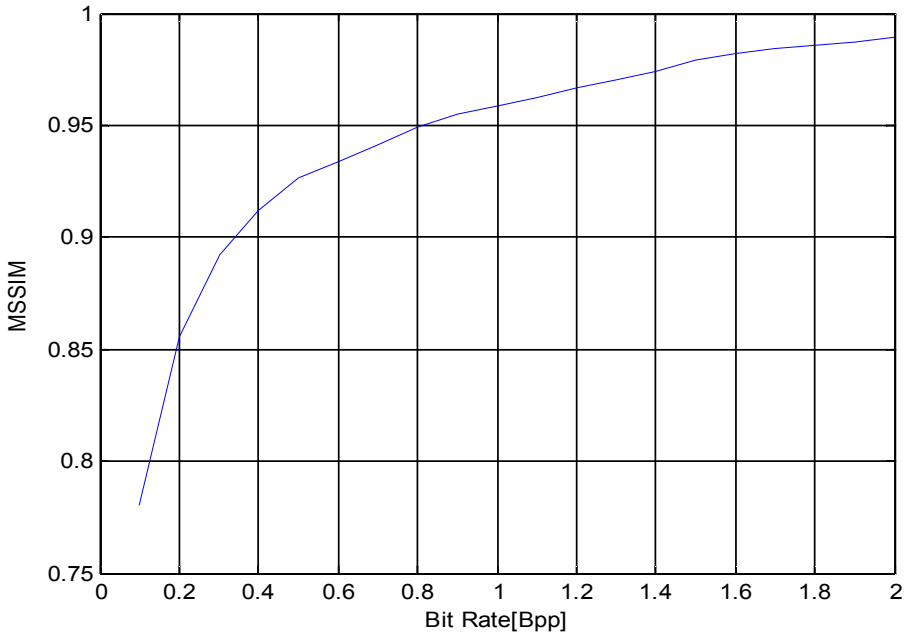


Figure IV- 9. l'indice de similarité structurelle moyenne en fonction du débit pour l'image OT-MONO2-8-colon (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

Voici l'image reconstruite successivement suite à trois rapports de compression déférents en utilisant la technique de compression SPIHT :

Image Originale

Image Décompressée

Bit Rate=0.2,PSNR=31.70dB

Image Originale

Image Décompressée

Bit Rate=0.4,PSNR=35.59 dB

Image Originale

Image Décompressée

Bit Rate=0.6,PSNR=37.72 dB

Image Originale

Image Décompressée

Bit Rate=0.8,PSNR=39.00 dB

Expérience 2

Le tableau suivant contient les résultats de l'application *SPIHT* sur l'image *« OT-MONO2-8-a7 »*. L'image *DICOM* de dynamique *8 bit par pixel* et de taille 512×512.

Rapport de compression [CR]	***Bit Rate [Bpp]***	***PSNR [dB]***	***MSSIM***
80	0.1	30.60	0.7961
40	0.2	33.46	0.8453
26.66	0.3	35.08	0.8698
20	0.4	36.44	0.8917
16	0.5	37.23	0.9029
13.33	0.6	37.99	0.9116
11.42	0.7	38.53	0.9233
10	0.8	39.24	0.9328
8.88	0.9	39.80	0.9376
8	1	40.29	0.9438
7.27	1.1	40.82	0.9481
6.66	1.2	41.37	0.9544
6.15	1.3	41.90	0.9584
5.71	1.4	42.40	0.9637
5.33	1.5	42.90	0.9694
5	1.6	43.47	0.9730
4.70	1.7	44.09	0.9753
4.44	1.8	44.58	0.9773
4.21	1.9	45.12	0.9794
4	2	45.67	0.9817

Tableau IV - 5. Table des performances de l'algorithme SPIHT pour l'image « OT-MONO2-8-a7.dcm ».

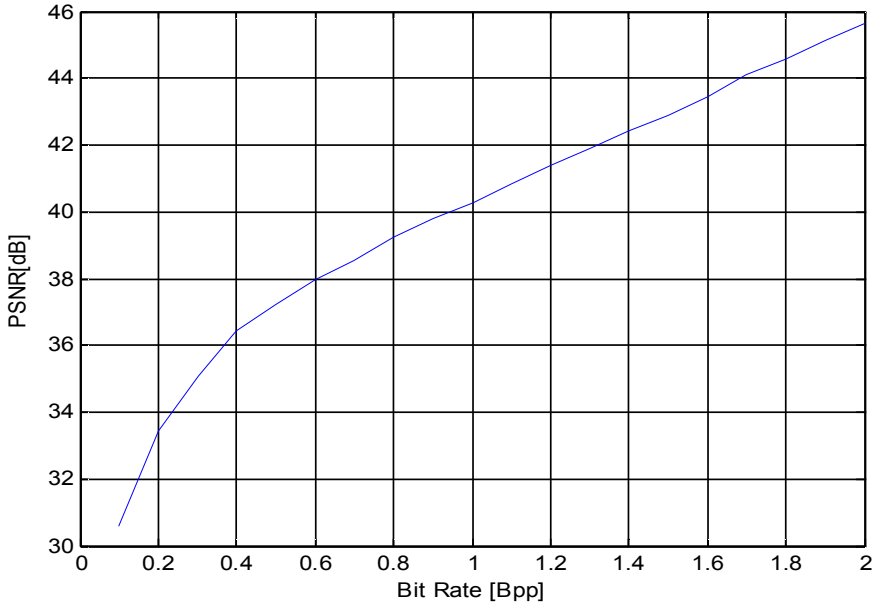


Figure IV- 10. Rapport Signal-à-Bruit pic en fonction du débit pour l'image OT-MONO2-8-a7 (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

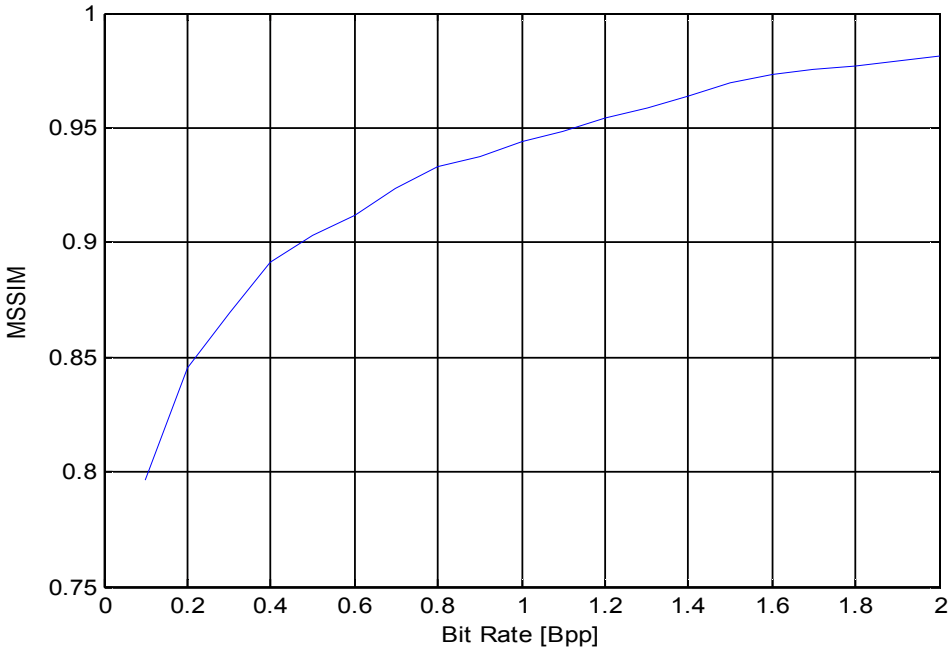


Figure IV- 11. L'indice de similarité structurelle moyenne en fonction du débit pour l'image OT- MONO2-8-a7 (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

Voici l'image reconstruite successivement suite à trois rapports de compression déférents en utilisant la technique de compression *SPIHT :*

Image Originale

Image Décompressée

Bit Rate=0.2,PSNR=33.46 dB

Image Originale

Image Décompressée

Bit Rate=0.4, PSNR=36.44dB

Image Originale

Image Originale

Bit Rate=0.6,PSNR= 37.99 dB

Image Originale

Image Décompressé

Bit Rite=0.8,PSNR =39.24 dB

Expérience 3

Le tableau suivant contient les résultats d'application *SPIHT* sur l'image **« CT-MONO2-8-abdo »**. l'image est de dynamique 8 bit par pixel et de taille 512×512

Rapport Compression[CR]	***Bit Rate [Bpp]***	***PSNR [dB]***	***MSSIM***
80	0.1	21.41	0.6511
40	0.2	25.16	0.7270
26.66	0.3	28.40	0.8000
20	0.4	30.60	0.8406
16	0.5	32.75	0.8823
13.33	0.6	34.20	0.9082
11.42	0.7	35.67	0.9261
10	0.8	37.14	0.9456
8.88	0.9	38.22	0.9556
8	1	39.34	0.9631
7.27	1.1	40.45	0.9707
6.66	1.2	41.62	0.9783
6.15	1.3	42.62	0.9826
5.71	1.4	43.51	0.9854
5.33	1.5	44.60	0.9884
5	1.6	45.59	0.9905
4.70	1.7	46.76	0.9928
4.44	1.8	47.60	0.9944
4.21	1.9	48.61	0.9956
4	2	49.57	0.9964

Tableau IV - 6. Table des performances de l'algorithme SPIHT pour l'image « CT-MONO2-8-abdo.dcm ».

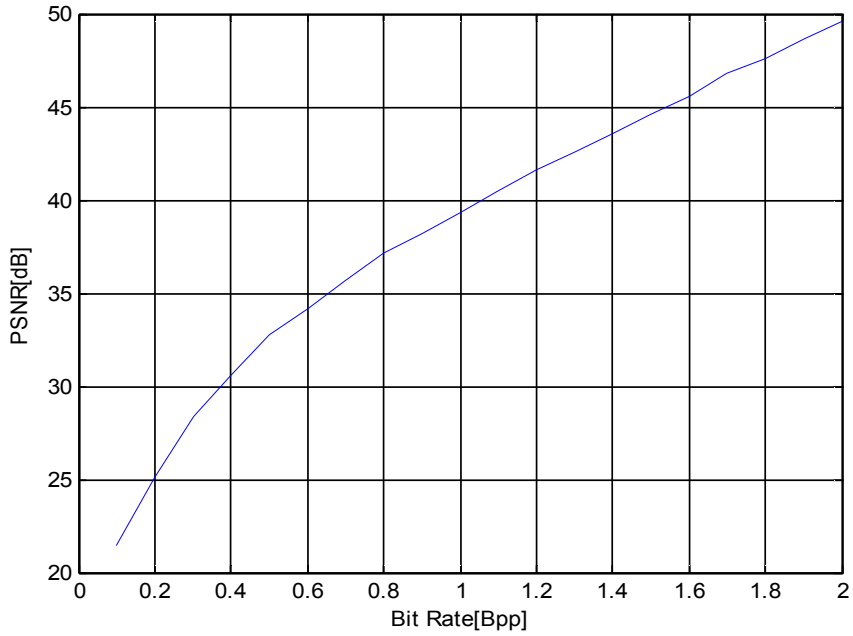


Figure IV- 12. Rapport Signal-à-Bruit pic en fonction du débit pour l'image « CT-MONO2-8-abdo » (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

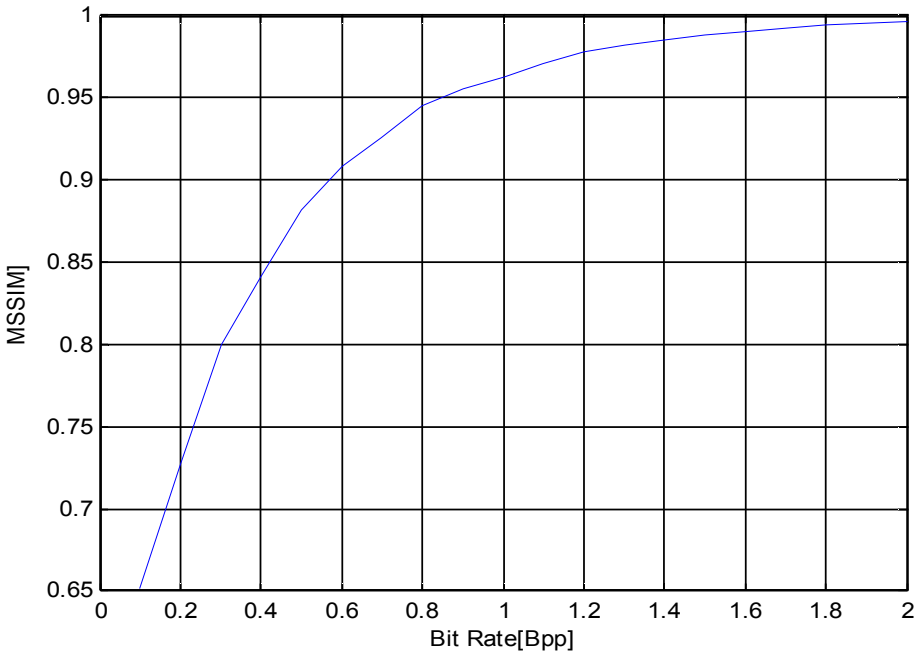


Figure IV- 13. l'indice de similarité structurelle moyenne en fonction du débit pour l'image « CT-MONO2-8-abdo» (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

Voici l'image reconstruite successivement suite à trois rapports de compression déférents en utilisant la technique de compression SPIHT :

Image Originale

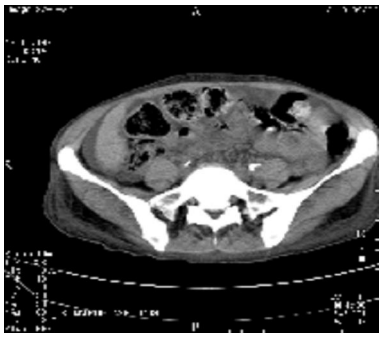


Image Décompressée

Bit Rate =0.2 , PSNR= 25.16 dB

Image Originale

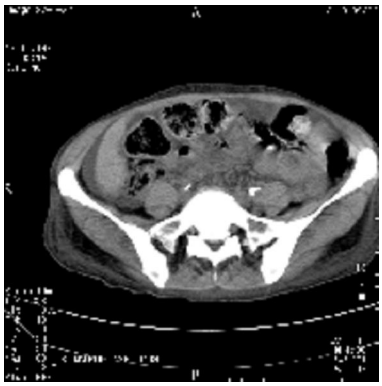
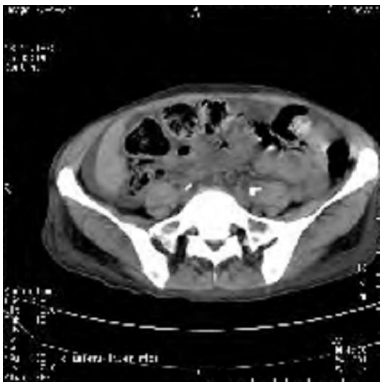


Image Décompressée



Bit Rate = 0.4 , PSNR = 30.60 dB

Image Originale

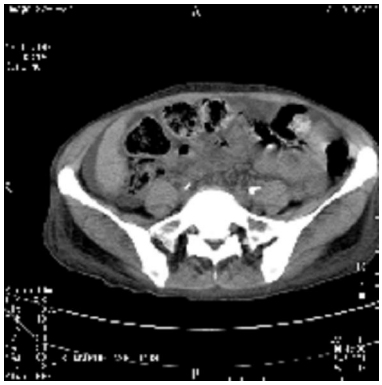
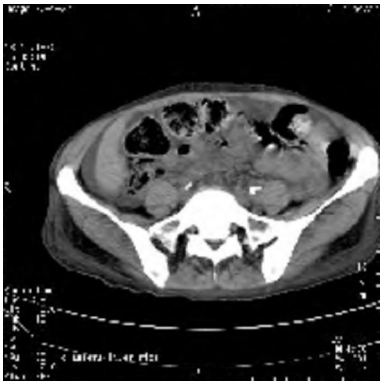


Image Décompressée



Bit Rate = 0.6 , PSNR = 34.20 dB

Image Originale

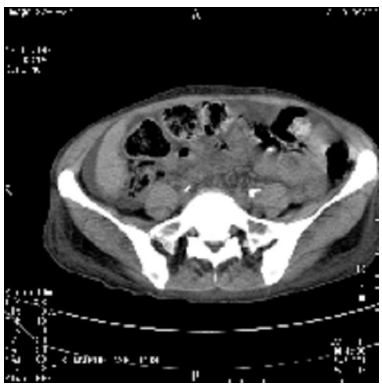
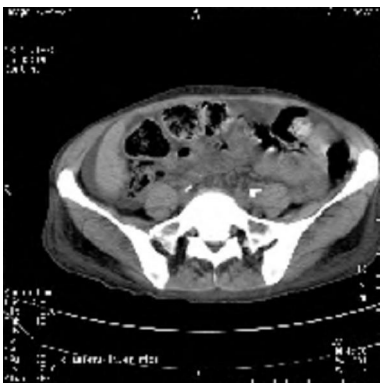


Image décompressée



Bit Rate =0.8 , PSNR = 37.14 dB

D'après les tests de compression par *SPIHT* qu'on a fait sur des fichiers *DICOM*, nous avons constaté que cet algorithme donne de bons résultats que ce soit : en termes des valeurs de PSNR [dB] qui sont dans l'intervalle [30,45] qui sont satisfaisantes pour une méthode de compression avec perte, soit en termes l'indice *MSSIM* qui est aussi très grand , C'est-à-dire la similarité structurelle entre l'image décompressée et l'image originale est très grande.

Comme étant La méthode *SPIHT* est basée principalement sur l'utilisation des ondelettes, donc dans cette partie on va étudier la variation de *PSNR* en fonction des ondelettes choisies.

Ondelette	**0.2 [Bpp]**	**0.4**	**0.6**	**0.8**	**1**
Haar	28.87	32.53	34.90	36.81	38.29
Db7	30.71	34.78	37.22	38.55	39.96
Db9	30.73	34.61	37.08	38.44	39.85
Db15	30.13	34.15	36.73	38.11	39.50
Bior2.4	31.23	34.90	37.32	38.85	39.95
Bior4.4	**31.70**	**35.59**	**37.72**	**39.00**	**40.34**
Bior6.8	**31.61**	**35.60**	**37.73**	**39.03**	**40.37**
RBior6.8	30.91	34.84	38.41	38.41	39.91
Coif5	31.13	35.07	37.45	38.69	40.14

Tableau IV - 7. Rapport Signal-à-Bruit pic en fonction du débit et divers filtres d'ondelette pour l'image OT-MONO2-8-colon (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

Suivant les résultats de compression SPIHT à l'aide de divers filtres d'ondelettes, on remarque que les deux ondelettes *bior4.4, bior6.8* donnent des résultats meilleurs par rapport aux autres ondelettes. Sachant que la meilleure ondelette est celle qui présente les pixels de l'image par des coefficients appartenant à l'espace de transformée et elle fait la bonne reconstruction d'image à partir de ces coefficients.

Dans ce qui suit, on va étudier l'effet du niveau de décomposition en ondelette sur la qualité de l'image décompressée par l'algorithme SPIHT. L'expérience a été faite pour un rapport de compression CR=40, i.e. Bit Rate=0.2 [Bpp]. Les résultats sont donnés dans tableau suivant :

Niveau	1	2	3	4	5	6	7	8	9
PSNR [dB]	5.72	5.73	14.28	27.27	30.91	31.53	31.67	31.70	31.70

Tableau IV - 8. La variation de PSNR en fonction du niveau de décomposition en ondelette.

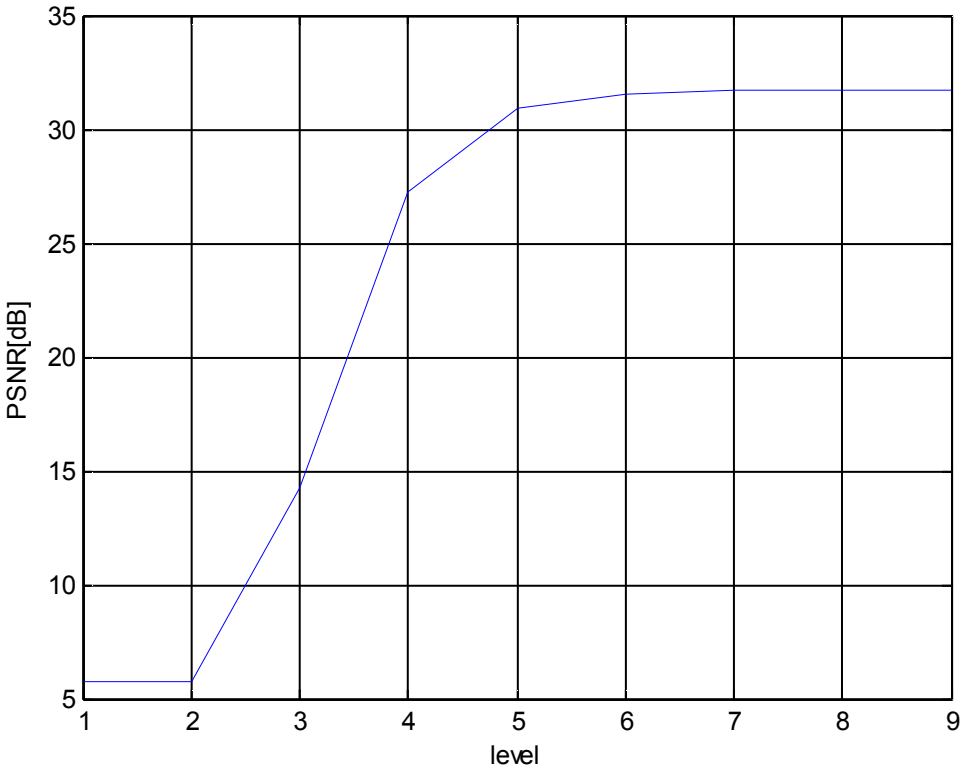


Figure IV- 14. Rapport Signal-à-Bruit pic en fonction de niveaux de décomposition pour la transformée en ondelettes pour l'image OT-MONO2-8-colon (0.2 bpp−512 × 512 pixels).

La figure nous montre que le niveau de décomposition en ondelette a un effet très important dans la détermination de la qualité d'image par la méthode SPIHT. Et ce, car cette technique utilise les relations de similitudes existantes entre les différentes sous bandes en formant des arbres des zéros ainsi que à localiser les coefficients d'odelettes significatifs. Après plusieurs expériences de l'application de cette méthode de compression sur des images de norme DICOM, on a constaté aussi que le niveau de décomposition en ondelettes est supérieur ou égal à 4. Dans cette condition (niveau de décomposition à partir de 4) , on évite l'effet du niveau de décomposition sur la qualité d'image reconstruite.

IV- 3. 4 Résultats de Comparaison SPIHT/EZW

Dans la partie suivante, on va évaluer les deux algorithmes de compression *SPIHT et EZW* par une simple comparaison des valeurs de *PSNR* et *MSSIM,* testés sur le fichier *« CT-MONO2-8-abdo.dcm ».*

Les deux tableaux suivants contiennent les valeurs de *PSNR* [dB] et de *MSSIM* obtenues par les deux algorithmes de compression *SPIHT et EZW :*

BitRate [BPP]	0.09	0.20	0.41	0.76	1.45
SPIHT	27.59	31.70	35.75	38.72	42.92
EZW	25.48	28.25	30.09	30.90	31.18

Tableau IV - 9. PSNR en fonction du débit des Algorithmes (SPIHT-EZW) pour l'Image « OT-MONO2-8-colon.dcm ».

BitRate[BPP]	0.09	0.20	0.41	0.76	1.45
SPIHT	0.7698	0.8553	0.9137	0.9457	0.9763
EZW	0.7052	0.7966	0.8669	0.9037	0.9213

Tableau IV - 10. MSSIM en fonction du débit des Algorithmes (SPIHT-EZW) pour l'Image « OT-MONO2-8-colon.dcm ».

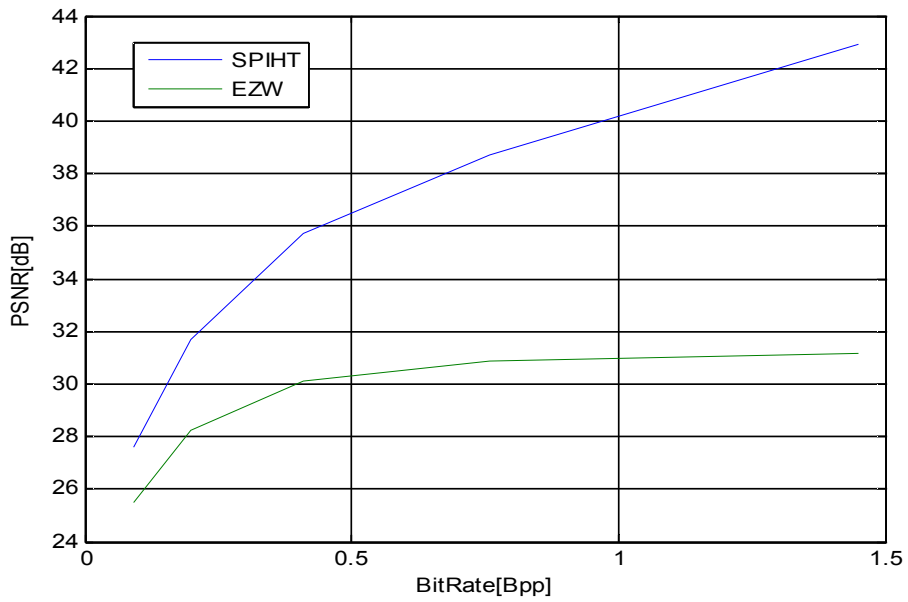


Figure IV- 15. Rapport Signal-à-Bruit pic en fonction du débit des algorithmes (SPIHT-EZW) pour l'image « OT-MONO2-8- colon.dcm» (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

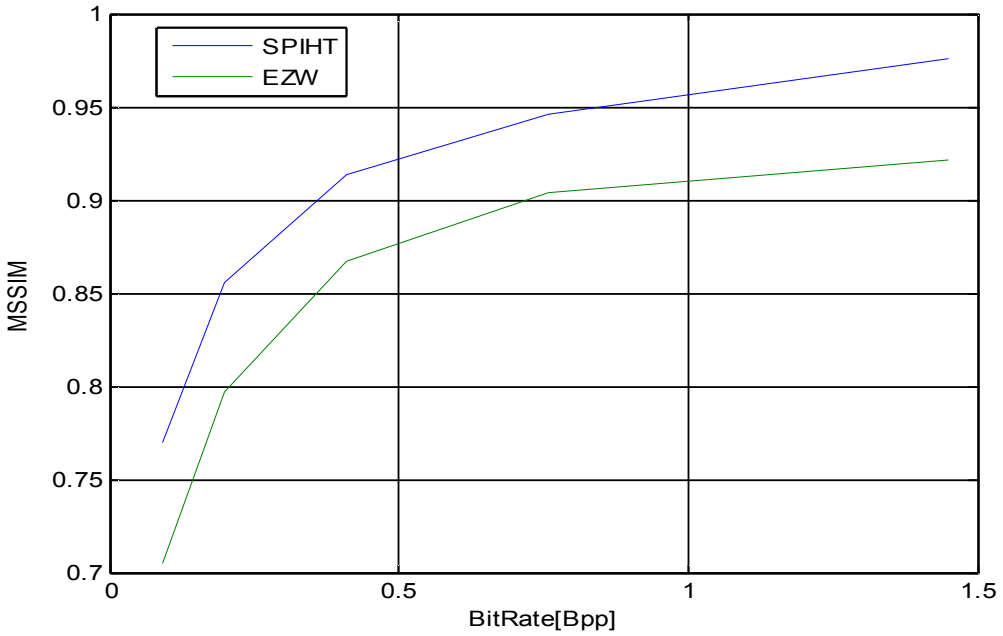


Figure IV- 16. L'indice de similarité structurelle moyenne des algorithmes (SPIHT-EZW) en fonction du débit pour l'image OT-MONO2-8-a7 (8 bpp−512 × 512 pixels). 9 niveaux de décomposition pour la transformée en Ondelettes.

Remarque

La chaine de compression et de décompression *EZW* utilise les mêmes paramètres dans la méthode *SPIHT*. Tels que le type d'ondelette choisi et le niveau de décomposition.

Les figures montrent que les valeurs de *PSNR* et *MSSIM* en utilisant la méthode de compression *SPIHT* sont meilleurs que celles obtenues par la méthode *EZW*. La qualité d'image reconstituée pour un même taux de compression en utilisant la méthode SPIHT est meilleure que celle obtenue par l'algorithme *EZW*.

Image Originale

SPIHT Image Decompressée

PSNR =31.70 dB, BitRate= 0.20 BPP

Image Originale

EZW Image Decompressée

PSNR= 28.25 dB , BitRate = 0.2 BPP

IV -4 Conclusion

Dans ce chapitre, nous avons exposé les différents résultats obtenus par les méthodes de compression que nous avons appliqués sur des fichiers médicaux de norme *DICOM*. Nous avons constaté que la méthode sans perte présenté par les algorithmes *RLE et LZW* donne des résultats modestes en termes de taux de compression calculé.

Cependant, la méthode *SPIHT* montre des résultats satisfaisants en termes de performances de qualités *PSNR* et *MSSIM* d'image reconstruite, Qui sont bonnes que celles obtenues par l'algorithme *EZW* pour des mêmes taux de compression.

Conclusion Générale

Conclusion Générale

Nous avons implémenté deux méthodes de compression, qui sont appliquées sur des fichiers médicaux de norme *DICOM* : La première est sans perte, permet de restituer d'une manière exacte l'image originale , la deuxième présente certaines pertes d'informations mais, néanmoins efficace. L'implémentation est faite avec *MATLAB*.

Généralement, La méthode de compression d'image sans perte est la plus utilisée dans le domaine médical, où la précision de traitement et de diagnostic sont exigés. Cela nous a amené à implémenter deux algorithmes de compression : *RLE et LZW*.

Le codage *RLE* (Run Lenght Encoding) est simple, son principe est d'utiliser les redondances existantes entre les pixels dans l'image *DICOM*. Cet algorithme donne de très bons taux de compression pour des fichiers qui présentent des zones homogènes, par contre, il donne de moins bons résultats dans le cas contraire.

L'algorithme *LZW* est basé sur l'utilisation des redondances entre des séquences d'image pour générer d'autres séquences-codes plus réduites en utilisant une table de translation créée d'une manière dynamique i.e. au cours du codage. Cet algorithme est plus utilisé dans le standard de compression Tiff et la compression du texte. Dans notre projet, nous avons compressé par cette méthode des fichiers *DICOM,* ce qui nous a donné des résultats satisfaisants en termes d'efficacité et de rapidité par rapport au codage *RLE*.

Les rapports de compression obtenus par la méthode sans perte sont limités. Pour remédier à ces insuffisances, nous avons utilisé une méthode avec perte qui prend en considération les tolérances du maximum de taux de compression (Tc), pour lequel l'image décompressée est acceptable. Et ce, aussi pour l'interprétation humaine que pour celle assistée par ordinateur telles que les valeurs de *PSNR, MSSIM*.

La méthode *SPIHT* (Set Partitioning in Hierarchical Trees) satisfait les conditions exigées par la compression des fichiers médicaux. Elle utilise les ondelettes comme un étage de décomposition et applique la technique d'encodage progressif décrite par des structures en arbre. Ses résultats sont meilleurs par rapport à d'autres tel que *EZW*.

Ces approches de la compression d'image en complémentarité avec d'autres conduiront, dans le futur, à des performances de compression plus élaborées tout en facilitant la transmission des images qui s'avère souvent délicate du fait des bandes passantes existantes. Une méthode de compression hybride à la fois sans et avec perte sur une formulation type Metropolis peut être porteuse de performance meilleure.

Bibliographies

Bibliographies

[1] O. LE CADET « Méthodes d'ondelettes pour la Segmentation d'images, Applications à l'imagerie médicale et au tatouage d'images » Thèse de Doctorat, Institut National Polytechnique de Grenoble ,2004.

[2] M. KADIK Rachid «Mise en Œuvre de La norme DIFFSERV sous NS2 Pour la Transmission de Video» Mémoire de Magister en électronique, École Nationale Polytechnique,Alger , 2009.

[**3]** http://www.isir.upmc.fr/UserFiles/File/clady_homepage/EPU/matlab_ti.pdf.

[4] R . WESTWAR, B. FURTHT « Real-Time Video Compression,Techniques and Algorithms »Florida Atlantic Univresity,Kluwer Academic Publishers,1997.

[5] T. ACHARYA , P. TSAI « JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures», Edition John Wiley & Sons, Inc.2005.

[6] R. M. GRAY«Entropy and Information Theory», Information Systems Laboratory Electrical Engineering Department Stanford University; Springer-Verlag 1990.

[7] GUYE.BLELLOCH« Introduction to Data Compression».Computer science Departement carnegie Mellon University sity, October 16, 2001.

[8] D. LEGUEN « Etude de schémas de régulation à double contraintes, debit-qualitélocale, pour la compression embarquée d'images satellitales » Thèse de Doctorat de l'université de RENNES1, 4 janvier 2001.

[9] M. BARLAUD, C. LABIT « Compression et Codage des Images et des Vidéos », LAVOISIER, Edition 2002.

[10] David A.Cluinie «Quintiles Intelegent Imaging,Lossless Compression of Grayscale Medical Image-Effectivesness of Traditional and State of the Approaches.

[11] Q .YUN SHI , S.Huifang « Image and video Compression for Multimedia Engineering Fundamentals,Algorithms,and standard». by CRC Press LLC, 2000.

[12] http://cnx.org/content/m0092/2.19/.

[13] P.BAS « Compression d'Images Fixes et de Sequences Vidéo»,Cours ENSERG/INPG,Laboratoire des Images et des signaux de Grenoble.

[14] Y.Gaudeau « Contributions en compression d'images médicales 3D et d'images naturelles 2D », Thèse de Doctorat de l'Université Henri Poincaré, Nancy 1, 2006.

[15] M. Rabbani, P. W. Jones«Digital Image Compression Techniques », SPIE OpticalEngineering Press, 1991.

[16] http://www.neurones.espci.fr/Theses_PS/OUSSAR_Y/05_CHAPITRE_3.pdf.

[17] K. SAYOOD « Introduction to Data Compression» Elsevier INC San Francisco; 2006.

[18] R. Souadnia , Kh. Benmahamed «Compression des Images Numeriques Fixes par Fractales »,laboratoire de traitement des signaux ,université de Ferhat Abbas, Sétif .factales 2000 , Constantine , Algerie, Conference Mediterrranneene 11-12 -2000.

[19] F. DERRAZ, M.BELADGHAM, M. KHELIF « Mesure Objective de la Qualité d'Image Médicale Dérivée de l'Index de Similarité Structurelle » ; Laboratoire de GénieBiomédicale–TLEMCEN-ALGERIE.http://www-lil.univ littoral.fr/~lewandowski/majecstic/articles/art_14_1_4_derraz.pdf.

[20] E.LAIN , G.RICHARDSON « VIDEO CODEC DESIGN,Developping Image and Video Compression systems»;the Robert Gorden University ,Aberdeen,Uk,JOHN WILY& SONS , LTD ,2002.

[21] F.DE SIMONE, D.TICCA, F. DUFAUX, MI. ANSORGE, T. EBRAHIMI «A comparative study of color image compression standards using perceptually driven quality metrics, Multimedia Signal Processing Group Institute of Electrical Engineering», Ecole Polytechnique Fédérale de Lausanne (EPFL). 2009.

[22] Z. Wang «Image Quality Assessment: From Error Visibility to Structural Similarity»,IEEE Image Transaction on Image compressing, VOL, 13, NO.4, APRIL 2004.

[23] E. CHRISTOPHE « Compression des images Hyperspectrales et son impact sur la qualité des Données». Thèse de Doctorat de l'école Nationale Supérieure de l'Aéronautique et de l'Espace ;Toulouse, 20 octobre 2006.

[**24]** http://www-ljk.imag.fr/membres/Valerie.Perrier/SiteWeb/node16.html .

[25] K.R.Rao , P.C.YIP, «The Transform and Data compression Handbook»;Boca Raton,CRC Press LLC,2001.

[26] I. AKAM BITA « Sur une approche de l'analyse en Composantes Indépendantes à la Compression des Images multi Composantes » ; Thèse de Doctorat ; Université Joseph Fourier de Grenoble, le 12 Février 2007.

[27] H. BEKKOUCHE « Synthèse de bancs de filtres adaptés, application à la Compression des Images». Thèse de Doctorat, Université Paris-Sud XI, 18 juin 2007.

[28] M. MISITI , Y.MISITI ,G.OPPENHEIM , J-M. POGGI« Wavelets Toolboxs for use With Matlab ». by The MathWorks, Inc , 1996 - 1997.

[29] S.BOUCHAMA«le tatuage des images appliqué à l'imagerie medicale»,Mémoire de Magister en Electronique,Ecole Nationale Polytechnique, Alger,2007.

[30] Digital imaging and Communications in Medicine A basic review Indrajit IK - Indian J Radiol Imaging.htm.

[31] http://www.codeproject.com/KB/graphics/dicomImageViewer.aspx.

[32] B. DAHO Omar, El. Salim «Quantification Vectorielle des Images DICOM», Mémoire d'Ingéniorat ; Ecole Nationale Polytechnique (Alger), promotion 2009.

[33] ftp://medical.nema.org/medical/dicom/2009/09_01pu.pdf.

[34] www.creatis.insa-lyon.fr/~dsarrut/.../rapportSeda2004.pdf.

[35] Les standards en Imagerie Médicale : (Série documents d'information) Version 1.0 – DICOM : Digital Image Communication in Médicine.

[36] http://eviewbox.sourceforge.net/JFR98/intro.html.

[37] S. G. HOGGAR «MATHEMATICS OF DIGITAL IMAGES, Creation, Compression, Restoration, Recognition», © Cambridge University Press 2006.

[38] A. SAID , W. PEARLMAN « An Image Multiresolution Representation for Lossless and Lossy compression», IEEE transactions on Image Processing. Cambridge MA- Nov. 1993.

[39] C. D.-FAUNDEZ « Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'´energie », Thèse de Doctorat, l'Université Henri Poincaré, Nancy 1, le 23 juin 2009.

[40] D. SALOMON « Data Compression The Complete Reference » 3rd Edition, springer 2004.

[41] G. Sadashivappa1, K.V.S. Ananda Babu «WAVELET FILTERS FOR IMAGE COMPRESSION», AN ANALYTICAL STUDY, ICGST-GVIP Journal, Volume 9, ISSN: 1687-398X.Issue 5, September 2009.

[42] K.TAEKON, E.ROBERT, V. DYCK, D. J. Miller «Hybrid fractal zero tree wavelet image coding, Signal Processing». Image Communication 17 (2002) 347–360 ,http://www.sciencedirect.com/science/article/pii/S0923596502000036.

[43] A. Said and W.A. Pearlman. « A New Fast and Effecient Image Codec Based on Set Partitioning in Hierarchical Trees». IEEE Transactions on Circuits and Systems for Video Technology, pages 243-250, 1996.

[44] G. Sadashivappa, K. V. S. Anandababu « Evaluation of Wavelet Filters for Image Compression » . World Academy of Science, Engineering and Technology 51 2009.

[45] N.T.HoàngLan, Étude de la Méthode de la Transmission en ondelette et l'application à la Compression des Images , 15 Juillet 2005.

[46] J.M.Shapiro « Embedded Image coding using Zero trees of wavelet coefficients», IEEE Transactions of signal Processing .Vol 41 NO 12 December 1993.

[47] http://perso.telecomparistech.fr/ ~cagnazzo/doc/Compression%20images%20 par %20ondelettes

Annexes

Annexes

Annexe A

Présentation des ondelettes

Introduction

Les ondelettes ont prouvé leur efficacité en théorie du signal depuis des dizaines d'années. Elles peuvent être utilisées d'une façon avantageuse dans la compression des données, telle que s'est exploité par SPIHT et EZW.

La transformée en ondelette

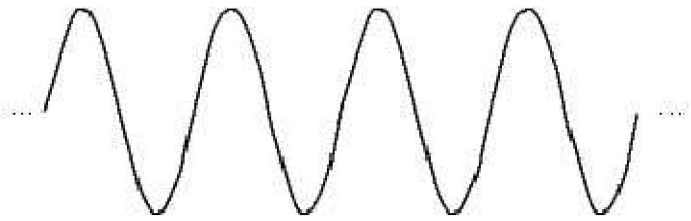
La transformée en ondelette (TO) est apparue en géophysique au début des années 1980 pour l'analyse des signaux sismiques, et a été formalisée plus tard par Grossmann et Morlet (1984) et Goupillaud. C'est au sein de ces dernières communautés que des développements théoriques et appliqués majeurs ont eu lieu ces quinze dernières années. Des avancées significatives ont notamment été faite par Meyer, Mallat, Daubechies , Chui, Wornell et Holschneider. Ces avancées aussi ont alors influencé d'autres domaines de recherche, dont en particulier, des applications pour la compréhension des processus géophysique (Foufoula-Georgiou et Kumar, 1994). Ces interactions entre développement et application favorisent encore aujourd'hui l'évolution rapide de l'outil « ondelettes ».

Définition d'une Ondelette

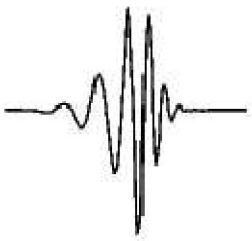
Une ondelette est une forme d'onde à durée limitée généralement courte, autour d'une valeur moyenne zéro.

En remarquant les images des ondelettes et les ondes sinusoïdales, vous pouvez voir intuitivement que les signaux avec des changements pointus pourraient mieux être analysés avec une ondelette irrégulière qu'avec une sinusoïde douce.

D'un point de vue historique, l'analyse d'ondelette est une nouvelle méthode, après la théorie de base de Joseph Fourier au dix-neuvième siècle. Il a créé les bases avec ses théories d'analyse de fréquence, qui se sont avérées énormément importantes et influentes.



Onde de sinus



Ondelette(10db)

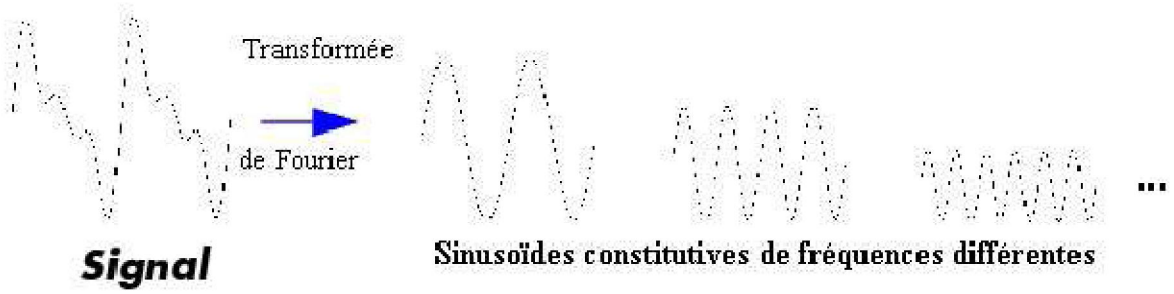
L'analyse en ondelette plus riche que la transformée de fourrier, compte tenu de sa robustesse au bruit, intéresse de plus en plus les chercheurs.

La Transformation en Ondelette Continue

Le processus d'analyse de Fourier est représenté par :

$$\boldsymbol{F(\omega)= \int_{-\infty}^{+\infty} f(t)\, e^{-i\omega t} dt} \qquad \text{.....(1)}$$

Les résultats de la transformation sont des coefficients $F(\omega)$. Quand on multiplie ces coefficients par une sinusoïde de fréquence ω on peut obtenir les composants du signal original.



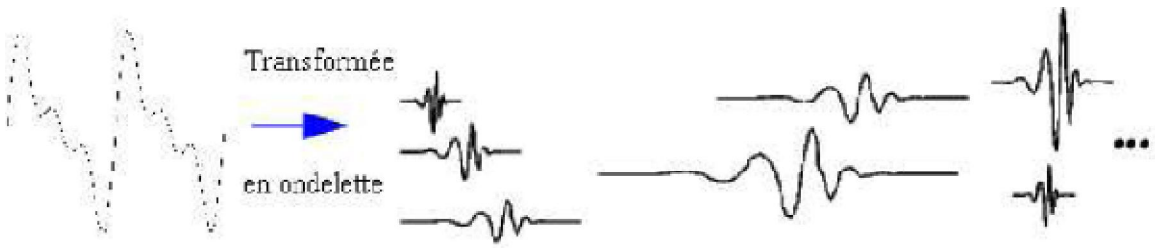
De même la transformation en ondelette continue (TOC) est définie comme la somme sur tout le temps du signal multiplié par des échelles :

$$\boldsymbol{C(échelle; position) = \int_{-\infty}^{+\infty} f(t)\psi(échelle, position, t)dt} \quad \cdots\cdots\cdots\cdots\cdots \quad (2)$$

Avec :

$$\boldsymbol{\psi_{s,\tau(x)} = \frac{1}{\sqrt{s}}\psi\left(\frac{x-\tau}{s}\right)} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (3)$$

t *Est le coefficient de translation de temps.*
s *Est coefficient d'échelle.*
C *coefficients d'ondelette.*
$\boldsymbol{\psi}$ *Ondelette Mère*

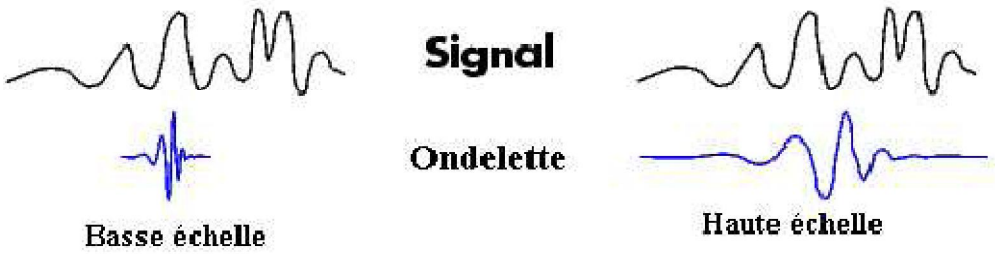


Signal Ondelettes Constitutives d'Echelles et Positions Différentes
Transformée en Ondelette

Donc, prendre des échelles, cela signifie l'étirage ou la compression de l'ondelette. Le décalage de temps signifie le déplacement d'ondelette. Les résultats de TOC sont des coefficients d'ondelette $\boldsymbol{C}$ qui est la fonction de l'échelle et la position multiplié chaque coefficient par ondelette d'échelle.

La Relation entre échelle et Fréquence

Il y a une correspondance entre les échelles d'ondelette et la fréquence comme indiqué par analyse d'ondelette. Comme la figure suivante :

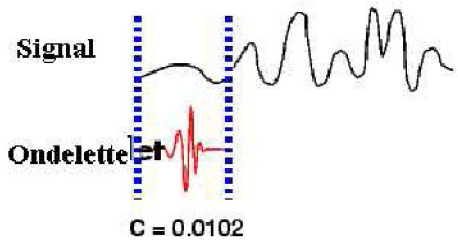


Échelle et Fréquence

- Basse échelle a → ondelette compressée → changer détails rapidement → haute fréquence ω.
- Haute échelle a → ondelette tirée → changer détails lentement→ basse Fréquence ω

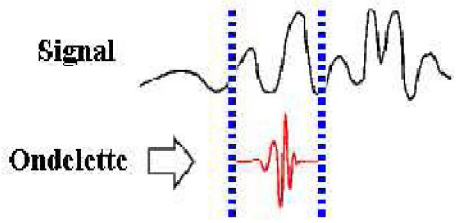
Il y a **5** étapes pour transformer en ondelette continue.

1. Prendre une ondelette et la comparer à une section au début du signal original.
2. Déterminer le nombre C, qui quantifie la liaison de l'ondelette avec le signal original. Une grande valeur de C implique une grande similarité. Plus précisément, si l'énergie du signal et l'énergie de l'ondelette sont égales à 1, C peut être interprété comme coefficient de corrélation. A noter que les résultats dépendent de la forme d'ondelette choisie.



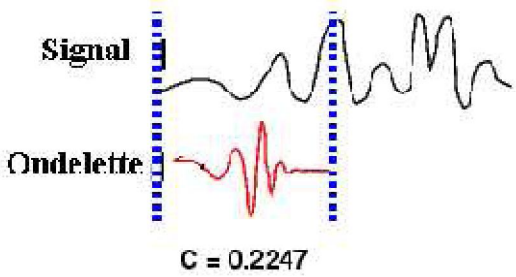
Étape 2

3. Se déplacer l'ondelette à droite et répéter les étapes 1 et 2 jusqu'à sur tous les signaux.



Étape 3

4. Étirer l'ondelette et répéter l'étape 1 jusqu'à 3



Étape 4

5. Répéter les étapes 1 à 4 pour toutes les échelles.

Quand on a fini toutes les étapes ci dessus, on fera produire les coefficients à différentes échelles par différentes sections du signal.

La Transformation en Ondelette Discrète

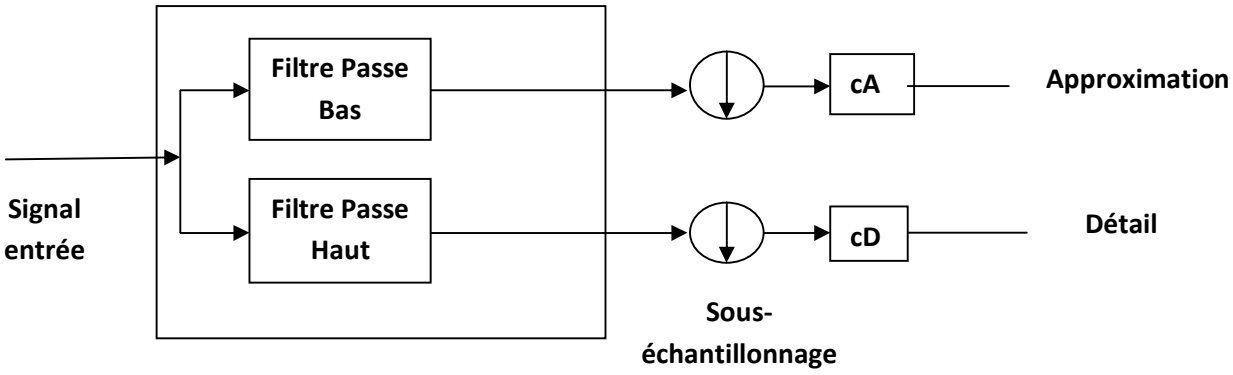
La transformée en ondelette continue ne peut pas être réalisée sur les signaux discrets. En plus, le calcul des coefficients à chaque échelle possible dans la TOC nécessitent beaucoup de temps et il peut générer trop de données. Donc, on peut choisir un sous ensemble d'échelles et de positions. Une autre technique est la transformée en ondelette discrète.

La transformée en ondelette discrète, ou TOD (en anglais : Discrete Wavelet Transform, ou DWT) est une technique utilisée dans la compression de données numériques avec perte. La compression est réalisée par approximation successives de l'information initiale ; du plus grands au plus fin.

Approximation et détail

Pour des signaux (image, parole …), le contenu basse fréquence est le plus significatif pour leur Identité. Par exemple la voix humaine, sans les composantes à haute fréquence, reste compréhensible même bruitée. Cependant, ce n'est pas le cas avec les composantes basses fréquences.

Dans la transformation en ondelette, il existe deux voies : L'approximation sont haute échelle i.e. des composants de basse fréquence du signal et les détails sont basses échelles i.e. des composants de hautes fréquences. Le processus de filtrage est comme suit : Le signal original S, traverse deux filtres selon la figure suivante**.[45].**



Annexe B

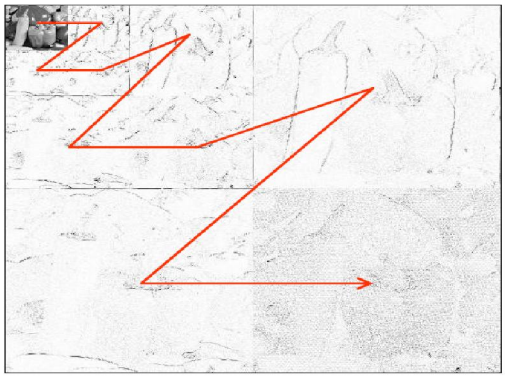
Algorithme EZW

Dans cette partie, on détaille le fonctionnement du codeur EZW (codeur en arbre des zéros), qui a été présenté dans un article de J. Shapiro**. [46]** se situant en référence **Shap [93].**

Caractéristiques principales :

- Scalabilité en qualité, c'est-à-dire représentation progressive
- Codage *lossy-to-lossless*
- faible complexité

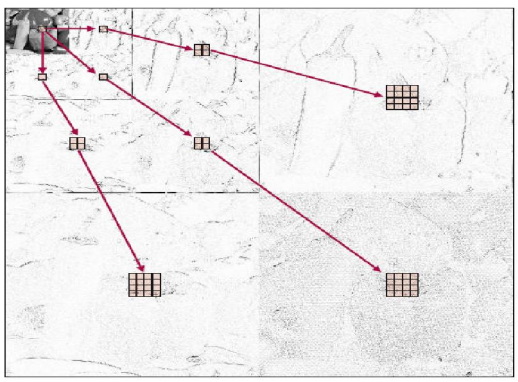
Représentation progressive: ordre des sous-bandes

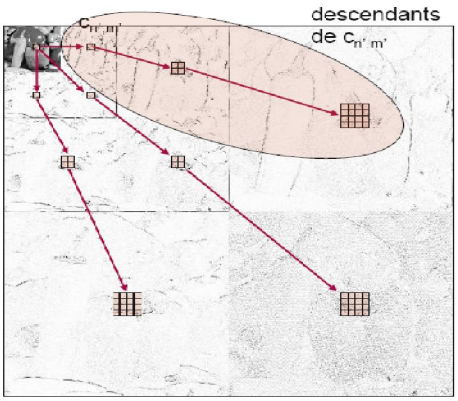


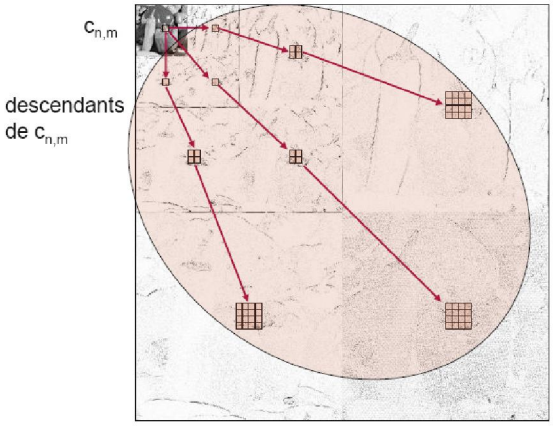
Un seul ordre de balayage des sous bandes n'est pas suffisant pour garantir que les coefficients plus grands soient envoyés en premiers

- Il faut localiser les coefficients significatifs. **[47]**
- **Idée**: Exploiter la corrélation inter-bande pour prévoir les coefficients *non-significatifs.*
- Si la prédiction est correcte, on économise la description de beaucoup de coefficients.

Arbre de coefficients d'ondelette







Idée de EZW

- Quand un coefficient est petit (i.e est inferieur de seuil), il est probable que tous ses descendants soient petits aussi.

⇒ Alors on utilise **un seul symbole** de codage pour représenter un coefficient $\boldsymbol{c}$ tel que :

– $\boldsymbol{c}$ est non significatif
– **Tous ses descendants sont non significatifs**

- En ce cas *c* est une *racine d'arbre de zéros* (zero-tree root).
- Avec un seul symbole (**ZT**) on code ($1+4+4^2+\ldots 4^{N-n}$) coefficients.
- Information de localisation implicite dans l'information de significativité.

Schéma de l'Algorithme EZW

$k = 0$;
– **n = floor(log2($|c|$max))**
– $\boldsymbol{T_k = 2^n}$
– **While (débit < débit disponible)**
- Dominant pass
- Raffinement pass

– • $T_{k+1} \leftarrow T_k/2$
• k^{++}
– **end**

Dominant Pass

– **Balayage des coefficients**
– **Si** $|c| > T_n$ ***Coefficient Significatif***
▪ **Si** $c > 0$ on encode SP (*significant positive*)
▪ **Si** $c < 0$ on encode SN (*significant negative*)
– **Sinon,** on compare *tous* les descendants de c avec le seuil

- **Si** tous les descendants sont ***Non-Significatifs***, ***c*** est codé comme ***Zero-Tree root***(ZTR) et ses descendants ne seront plus considérés à partir de ce pass.
- **Sinon** le coefficient est codé comme ***Isolated Zero*****(IZ)**

Raffinement Pass

- On code un ultérieur bit pour tous les coefficients significatifs
- C'est équivalent à dire qu'on réduit de moitié la cellule de quantification des coefficients significatifs

Itération et terminaison

Le pas dominant *k* permet de coder le k-me plan de bit de la matrice des coefficients d'ondelette

- Au fait, dire que un coefficient est significatif est équivalent à dire que $2^n < |c| < 2^{n+1}$
- On réduit le seuil de moitié : on passe au prochain plan de bit
- L'algorithme termine quand
- on a épuisé le nombre de bit disponible ou
- quand on a codé *tous* les plans de bit (transformée en entière, codage sans perte)

- **Codage par plan de bits :**
- Au pas k on code le bit plane $log_2 T_k$
- **Description progressive :**
- Chaque nouveau plan de bit permet de raffiner la quantification des coefficients
- **Codage lossless-to-lossy**
- On utilise une transformée en entiers (tous coefficients entiers)
- Quand on code tous les plans de bit, on peut reconstruire parfaitement tous les coefficients. **[47]**