

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de L'ingéniorat
Département : Electronique
Domaine : Sciences et Techniques
Filière : Electronique
Spécialité : Instrumentation

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

**COMPRESSION DES IMAGES MEDICALES PAR
TRANSFORMEE EN ONDELETTE COUPLEE AVEC CODEUR
SPIHT**

Présenté par : *BEKRAR Amir*

Encadrant : *KADDECH Nadia*

MCA

UBM - ANNABA

Jury de Soutenance :

BOUSBIA-Salah M	Professeur	UBM ANNABA	Président
KADDECH Nadia	Grade	UBM ANNABA	Encadrant
BOUGHAZI	Professeur	UBM ANNABA	Examineur

Année Universitaire : 2020/2021

ملخص

ضغط الصور يلعب دورا كبيرا تخزين و نقل البيانات بما في ذلك الصور الطبية. بحيث تم تطوير العديد من الطرق للقيام بذلك بعضها بفقدا و أخرى لا. من بينها طريقة تحويل المويجات, و التي اكثرهم استخداما وفعالية. الهدف من هذا العمل تطوير لوغارتمية لضغط الصور الطبية عن طريق تحويل المويجات باستعمال شفرة spiht تعني الاشجار الهرمية لتقسيم المجموعات , قمنا بتنفيذ عدة أنواع من المويجات و درسنا نتائجها باستخدام اعدادات لتقييم جودة الصور (psnr & mssim)

ABSTRACT

Picture compression play big role in data transfer and storing, including medical picture . However, many ways were developed to make this operation some it with losses and others without . Moreover, the most effective and useful one called wavelet transform. The goal from this work is to develop an algorithm to compress medical pictures using (SPIHT) method. Therefore, we did too many test studying the results with deferent settings to review the quality of the pictures (PsNR & MSSIM)

RESUME

La compression d'image suscite un intérêt considérable dans le domaine de stockage et de transmission de données, notamment les images médicales. Divers méthodes de compression d'images ont été élaborées dont certaines avec pertes et d'autre sans pertes. Parmi les plus utilisées et les plus efficaces on trouve la méthode de compression d'images par transformée en ondelettes. L'objectif de ce travail est de développer un algorithme de compression d'image médicales en couplant cette transformée en ondelette avec le codeur SPIHT (Set Partitioning In Hierarchical Trees) Nous avons implémenté plusieurs types d'ondelette et étudié les résultats par des paramètres d'évaluation de la qualité d'image (PSNR et MSSIM)

REMERCIEMENTS

AVANT TOUT NOUS REMERCIONS "ALLAH"

EN PRÉAMBULE DE CE MÉMOIRE, JE TIENS À EXPRIMER MES SINCÈRES
REMERCIEMENTS AUX PERSONNES QUI M'ONT AIDÉ ET QUI ONT CONTRIBUÉ À LA
PRÉPARATION DE CE MÉMOIRE.

JE TIENS À REMERCIER SINCÈREMENT MADAME. « KADDECHE NADIA », EN TANT
QU'ANIMATRICE DE MÉMOIRE, A TOUJOURS ÉTÉ TRÈS ATTENTIVE ET DISPONIBLE
DANS LA RÉALISATION DE CE SOUVENIR, AVEC L'INSPIRATION, L'AIDE ET LE TEMPS
DONT ELLE DISPOSAIT. ELLE VOULAIT SE CONSACRER ET SANS CES NOTES JE NE
VERRAIS PAS LE JOUR.

JE REMERCIE ÉGALEMENT LE PRÉSIDENT ET LES MEMBRES DU JURY QUI ME
GUIDERONT DANS L'ÉVALUATION DE MON TRAVAIL

ENFIN, JE TIENS À REMERCIER TOUT LE MONDE
DES COLLÈGUES QUI M'ONT TOUJOURS SOUTENU ET ENCOURAGÉ LORS DE LA
RÉALISATION DE CETTE THÈSE.

MERCI À TOUS.

Table des matières

Introduction Générale

Chapitre I : Généralités sur les techniques de compression

I. INTRODUCTION	1
II. BUT DE LA COMPRESSION D'IMAGE	1
▪ II.1) Dé-corrélation.....	2
▪ II.2) Quantification.....	2
▪ II.3) Codage.....	2
III. COMPRESSION SANS PERTE D'INFORMATIONS.....	2
▪ III.1) Codage entropique	2
• III.1.1) Codage de Huffman	2
• III.1.2. Codage Arithmétique.....	3
▪ III.2) Codage par plages de zéros (RLE)	3
▪ III.3) Codage Lempel-Zif-Welsh (LZW).....	3
IV. COMPRESSION AVEC PERTE D'INFORMATIONS	3
▪ IV.1) Quantification.....	4
• IV.1.1). Quantification scalaire.....	4
• IV.1.2). Quantification vectorielle.....	5
▪ IV.2) Compression par transformation.....	6
▪ IV.3) Codage par prédiction.....	7
▪ IV.4) Codage Des Sous-Bandes.....	8
• IV.4.1). Algorithme EZW.....	8
• IV.4.2) Algorithme SPIHT.....	8
• IV.4.3) Algorithmes EQ.....	8
V. EVALUATION DE LA QUALITE DE COMPRESSION.....	9
• V.1 Taux de compression.....	9
• V.2 Taux d'information (Entropies)	9
• V.3 Mesures de distorsion (Mesures de fidélité)	10
VI. CONCLUSION.....	11

Chapitre II : transformée on ondelette

I. INTRODUCTION	13
II. TRANSFORMEE EN ONDELETTES.....	13
III. ANALYSE MULTI RESOLUTION.....	13
▪ III.1.Principe	13
▪ III.2. Algorithmes de Mallat.....	14
▪ III.3.Algorithmes et disposition des images dans une décomposition en 2D.....	14
VI. Ondelettes biorthogonales.....	14
V. CONCLUSION.....	15

Chapitre III : Algorithme SPIHT

I.CODEUR EZW	16
▪ I.1) Principe.....	16
▪ I.2) Algorithme de codage EZW.....	16
II.CODEUR SPIHT.....	17
▪ II.1) Principe.....	18
▪ II.2) Algorithme de codage SPIHT.....	19
III.EXEMPLE.....	20
IV. CONCLUSION.....	22

Chapitre IV : Résultats et discussions

I. INTRODUCTION	23
II. ENVIRONNEMENT DE TRAVAIL	23
▪ II.1) Base d'image utilisée.....	23
▪ II.2) Choix des ondelettes.....	23
III. Résultats et discussions.....	24
III.1 Influence du niveau de décomposition.....	25
a. Application 1	26
b. Application 2	27
c. Application 3.....	29
III. 2. Influence du type d'ondelette sur la performance de compression.....	32
a. Application 1	32
b. Application 2	35
c. Application 3	38
III .3Influence de la taille	41
a. Test 1	42
b. Test 2.....	43
c. Test 3	45
IV. CONCLUSION.....	47

Conclusion Générale

Bibliographie

Liste des Figures

Fig	Titre	Page
Chapitre I		
Figure (I-1)	Schéma synoptique de compression	1
Figure (I-2)	Schéma synoptique de codeur avec pertes idéal	4
Figure (I-3)	Schéma synoptique d'une fonction en escalier	4
Chapitre II		
Figure (II.1)	Banc de filtres numériques en AMR	12
Figure (II.2)	Algorithme de décomposition en ondelettes par AMR d'un signal en 2D	14
Figure (II.3)	Disposition des images dans une décomposition en ondelettes par analyse multi résolution (Cas où la décomposition s'effectue jusqu'au 3ème niveau)	14
Figure (II.4)	Algorithme de reconstruction a l'aide des ondelettes par AMR d'un signal en 2D.	15
Chapitre III		
Figure (III.1)	Structure d'arbre des coefficients d'ondelette	16
Figure (III.2)	Dépendance de coefficients et arbres d'orientation spatiale dans SPIHT utilisant la décomposition dyadique avec deux résolutions	17
Figure (III.3)	Organigramme de l'algorithme de SPIHT	18
Figure (III.4)	Exemple de SHAPIRO.	19
Chapitre IV		
Figure (IV.1)	Image scanner du nez .	24
Figure (IV.2)	Image Poitrine	24
Figure (IV.3)	Image IRM du cerveau	24
Figure (IV.4)	Image IRM Rachis cervical	24
Figure (IV.5)	Résultats de l'image nez 512*512 reconstruite par SPIHT	26
Figure (IV.6)	Résultats de l'image poitrine 512*512 reconstruite par SPIHT	28
Figure (IV.7)	Variation du PSNR en fonction de CR	30
Figure (IV.8)	Résultats de l'image « rachis cervical »512*512 reconstruite par SPIHT	31
Figure (IV.9)	Variation du PSNR en fonction de CR	33
Figure(IV.10)	Résultats de l'image « rachis cervical »reconstruite par SPIHT	34
Figure(IV.11)	Variation du PSNR en fonction de CR	36
Figure(IV.12)	Résultats de l'image « IRM du cerveau» reconstruite pour différentes ondelettes	37

Figure(IV.13)	Variation du PSNR en fonction de CR	39
Figure(IV.14)	Résultats de l'image « Poitrine » reconstruite pour différentes ondelettes	40
Figure(IV.15)	Résultats de l'image « Rachis cervical » reconstruite pour différentes taille	42
Figure(IV.16)	Résultats de l'image « : IRM du cerveau » reconstruite pour différentes taille	44
Figure(IV.17)	Résultats de l'image « scanner du nez » reconstruite pour différentes taille	46

Liste des Tableaux

Tab	Titre	Page
Tableau III.1	Tableau montrant étape par étape le déroulement du premier passage de l'algorithme SPIHT 2D.	21
Tableau IV.1	Variation du PSNR, MSE et du temps en fonction du niveau L	25
Tableau IV.2	Variation du PSNR, MSE et du temps en fonction du niveau L	27
Tableau IV.3	Variation du PSNR, MSE et du temps en fonction de CR du niveau L	29
Tableau IV.4	Variation du PSNR, MSE et du temps en fonction de CR de l'image «Rachis cervical».	32
Tableau IV.5	Variation du PSNR, MSE et du temps en fonction de CR de l'image « IRM du cerveau».	35
Tableau IV.6	Variation du PSNR, MSE et du temps en fonction de CR de l'image « Poitrine »	38
Tableau IV.7	Variation du PSNR du temps en fonction de CR de l'image «Rachis cervical».	41
Tableau IV.8	Variation du PSNR du temps en fonction de CR de l'image « IRM du cerveau».	43
Tableau IV.9	Variation du PSNR du temps en fonction de CR de l'image « scanner du nez».	45

Introduction Générale

L'imagerie médicale joue un rôle très important dans l'observation in vivo chez l'humain. Elle permet en effet d'observer la structure et le comportement d'organes grâce à un ensemble important de techniques d'acquisition.

La représentation numérique de ces images médicales crée ainsi des problèmes de stockage et de transmission. La solution à ces problèmes nécessite de faire appel à des algorithmes de compression. Cette opération devient plus que nécessaire vu le volume important d'information mis en œuvre lors de l'utilisation des images numériques. De plus, la compression présente un intérêt évident pour la transmission rapide des images, elle consiste à éliminer les informations superflues et permet la représentation la plus compacte et la plus succincte des données.

On distingue deux types de compression : la compression sans perte (ou réversible), qui consiste à préserver l'intégralité des données en conduisant à une reconstruction parfaite.

Il y a aussi la compression avec perte (ou irréversible), c'est la plus répandue, elle consiste à la représentation approximative et compacte de l'information originale en générant des dégradations.

La compression avec pertes maîtrisées peut être la réponse la plus appropriée, à condition bien entendu que les pertes n'affectent pas la qualité des images pour l'usage régulier des praticiens. Hier encore inenvisageable, l'idée d'une compression avec pertes semble aujourd'hui de mieux en mieux acceptée par les médecins, comme en témoigne par exemple, l'American College of Radiology (ACR) qui estime que les techniques de compression avec pertes peuvent être utilisées à des taux raisonnables, sous la direction d'un praticien qualifié, sans aucune réduction significative de la qualité de l'image pour le diagnostic clinique.

Depuis quelques années, les Transformées en Ondelettes Discrètes (DWT) ont gagné un intérêt considérable pour le traitement des signaux. Elles permettent une représentation du signal par un nombre limité de coefficients tout en localisant les discontinuités avec précision.

Plusieurs algorithmes de compression utilisant les ondelettes ont été proposés. L'une des applications les plus connues en compression d'image est la norme JPEG 2000 qui donne généralement des résultats plus satisfaisants que ceux obtenus par la norme JPEG.

D'autres algorithmes de compression par ondelettes dites codages imbriqués ont été proposés. Ces codeurs sont basés sur la notion d'arbre de zéros (zerotree) proposée par Shapiro en 1993.

Le premier codeur en sous-bandes par zerotree à avoir été introduit est le codeur EZW (Embedded Zerotree Wavelet). Ce codeur offre une représentation progressive de l'image tout en apportant d'excellentes performances de débit-distorsion par rapport à des codeurs bien plus complexes et non progressifs de l'époque.

Dans ce mémoire nous nous intéressons à la compression d'images médicales en niveaux de gris en appliquant l'algorithme SPIHT basé sur la transformée en ondelettes et ce dans le but d'atteindre des taux de compression élevés avec une qualité satisfaisante de l'image reconstruite.

Notre mémoire est organisée selon quatre chapitres :

Le premier chapitre est dédié à un état de l'art sur la compression d'images, notamment la compression sans pertes. Un rappel sur les codages imbriqués est aussi proposé.

Le deuxième chapitre est consacré aux ondelettes continues, discrètes et l'analyse multi résolution, ce chapitre est au cœur de la compression avec perte. Ce chapitre débute en premier temps par une définition sur les ondelettes continues, discrètes, ainsi que des notions sur les filtres qui sont la base des ondelettes discrètes. Nous appliquons cette transformée couplée au codeur SPIHT à la compression de quelques images médicales en niveau de gris .

Dans le troisième chapitre nous penchons sur une présentation détaillée de l'algorithme SPIHT avec des exemples d'utilisations.

Le quatrième chapitre présente les résultats obtenus sur différentes images tests. L'estimation et le jugement de la qualité d'image compressée sont donnés par les paramètres d'évaluation le PSNR et le MSE. Le manuscrit s'achève par une conclusion générale et des perspectives envisagées concernant notre travail.

Chapitre I : Généralités sur les techniques de compression

I. INTRODUCTION :

Notre quotidien s'est vu peu à peu envahit par une évolution technologique au sens propre du terme, notre vie s'est vu numérisée en images ou en vidéos même le vieux domaine de la médecine n'y a pas échappé.

Cependant, toutes ces données numériques présentent un inconvénient majeur lié essentiellement au volume d'informations numériques qu'il est nécessaire de traiter, transmettre et/ou de stocker.

La compression est très vite devenue la solution indispensable dans ce cas de figure grâce à l'avancement des technologies de l'information, elle est utilisée pour réduire le nombre de bits nécessaire pour la représentation des données volumineuses.

Notons que les techniques de compression d'images médicales peuvent être classées en deux catégories :

Compression sans perte et celle avec perte. Le premier type de compression, utilise uniquement le principe de la réduction de l'information et n'engendre pas de perte, le deuxième type est une représentation approximative de l'information sans perte et avec perte des images médicales, nous présentons les différents codeurs existants dans la littérature, leurs avantages et leurs inconvénients.

II. BUT DE LA COMPRESSION D'IMAGE :

Avec le développement de l'outil informatique, on effectue des échanges de volumes importants d'informations, or la gestion d'une telle masse pose des problèmes de stockage et de transfert. Pour cela, des études ont été menées afin de mettre en évidence des algorithmes de compression et de décompression de données.

Leurs but est de changer le format des informations de telle sorte qu'elles occupent moins de volume. Une fois compressées, les données ne sont plus accessibles en tant que données cohérentes; pour les récupérer il suffit de les décompresser.

Le principe de la compression d'une image se fait en général selon le schéma synoptique suivant, **Figure (I-1) :**

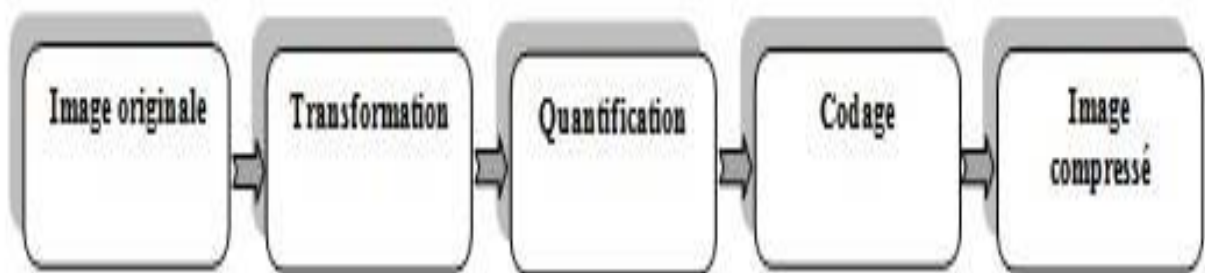


Figure (I-1) : Schéma synoptique de compression

II .1) Dé-corrélation:

La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduisent une corrélation très forte sur l'image.

On essaie donc de tirer profit de cette corrélation, pour réduire le volume d'information en effectuant une opération de dé-corrélation des pixels.

La dé-corrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible [1].

II.2) Quantification:

La quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. Elle représente une étape clé de la compression.

Elle approxime chaque valeur d'un signal par un multiple entier d'une quantité Q, appelée Quantum élémentaire ou pas de quantification. Elle peut être scalaire ou vectorielle.

II.3) Codage:

Une fois les coefficients quantifiés, ils sont codés. Un codeur doit satisfaire à priori les conditions suivantes :

- Unicité: deux messages différents ne doivent pas être codés de la même façon.
- Déchiffrabilité : deux mots de codes successifs doivent être distingués sans ambiguïté.

III. COMPRESSION SANS PERTE D'INFORMATIONS

Les différentes méthodes de compression d'images peuvent être classées en deux grandes catégories d'après leurs propriétés [2] avec perte d'informations et sans pertes d'informations.

Les méthodes de compression sans pertes d'informations permettent de retrouver exactement toute l'information contenue dans l'image numérique originale. Il existe plusieurs méthodes de compression sans pertes [3], [4], [5].

III.1) Codage entropique :

Un codeur entropique est une méthode de codage de source sans pertes qui permet de compresser une séquence de symboles en se basant sur leur probabilité d'apparition. Chaque symbole se voit assigner une nouvelle représentation (dépendant de sa probabilité d'apparition) de manière à ce que le codage de la source s'approche au plus de son entropie.

Les codeurs entropiques qui vont être présentés permettent de compresser l'information en générant un code sans ambiguïté, ils autorisent donc de faire une restitution sans perte.

III.1.1) Codage de Huffman :

Les codes de Huffman, mis au point en 1952[6], [7], sont parmi les premiers à avoir émergés et sont donc également parmi les plus répandus.

Le principe est d'attribuer un code court à un symbole fréquent et un code plus long à un symbole plus rare. En se plaçant dans le cas d'un alphabet de sortie binaire, l'attribution du nombre de bits pour chaque symbole se fait par le biais d'un arbre binaire, construit en fonction de la fréquence d'apparition des symboles.

A chaque nœud de l'arbre est associé la fréquence d'apparition de l'ensemble des symboles présents dans son sous arbre. L'arbre est construit de manière ascendante à partir de ses feuilles. Chaque feuille correspond à un symbole, et à sa fréquence d'apparition. Itérativement, jusqu'à l'obtention de la racine, un nœud père est généré pour le couple des deux nœuds ayant la fréquence d'apparition la plus faible.

Le code de chacun des symboles est alors le chemin partant de la racine jusqu'à la feuille lui correspondant (bit 0 pour aller au fils gauche, et bit 1 pour aller au fils droit par exemple). Ce code est donc défini de telle sorte qu'aucun des symboles de l'alphabet ne soit le préfixe d'un autre : chaque symbole étant associé à une feuille de l'arbre, le chemin qui permet d'y accéder ne peut en aucun cas être le sous-chemin d'un autre. Ainsi, le codage est non ambigu.

Le code Huffman peut être statique, transmis ou adaptatif. Lorsqu'il est statique, le code est fixé pour le codeur et le décodeur et n'a donc pas besoin d'être transmis.

Cependant, il n'est pas toujours adapté aux données.

Lorsqu'il est transmis, l'arbre doit être construit par le codeur à partir des fréquences d'apparition des symboles : un parcours supplémentaire des données est donc nécessaire (sur la totalité, ou un échantillon représentatif). Il a également un coût pour la compression puisqu'il doit être transmis.

Enfin lorsqu'il est adaptatif, en partant d'un arbre particulier les fréquences des symboles sont mises à jour au fur et à mesure de leur apparition. L'arbre doit donc être mis à jour régulièrement, ce qui demande un temps de calcul important. L'arbre de départ peut également être statique ou transmis.

Le principe est le suivant:

Les probabilités d'occurrence de chaque message sont placées dans une liste dans un ordre décroissant. Nous dirons que la liste est composée d'enfants.

Les deux probabilités les plus faibles sont identifiées en fin de liste.

La somme des deux probabilités est placée à sa place dans la liste triée. Elle constitue un nœud parent. Les deux enfants sont retirés de la liste.

Le chemin « enfant de plus faible probabilité, parent » est codé par un 1, l'autre par un 0.

La procédure reprend à l'étape 2 jusqu'à ce qu'il ne reste plus qu'une probabilité dans la liste.

III.1.2. Codage Arithmétique:

Contrairement à l'algorithme de Huffman qui associe aux symboles des motifs binaires dont la taille dépend de leur distribution. Le codeur arithmétique traite le vecteur de données dans son ensemble, en lui associant un unique nombre décimal rationnel varie entre 0 et 1 [9].

Ce nombre code est construit par subdivisions récursives d'intervalles. Un intervalle est subdivisé pour chaque nouveau symbole qui appartient à la séquence. On obtient, à la fin, un sous-intervalle de l'intervalle $[0, 1[$. Tel que tout nombre réel appartenant à cet intervalle représente la séquence à coder. Ce nombre compris entre 0 et 1, possède d'autant moins de chiffres après la virgule que la redondance du vecteur de données. Ces chiffres décimaux dépendent non seulement des symboles du vecteur dans l'ordre où ils apparaissent, mais aussi de leur distribution statistique.

III.2) Codage par plages de zéros (RLE) :

Toutefois, il est courant qu'en sortie du quantificateur, la source ne soit pas totalement dé-corrélée, c'est le phénomène de mémoire. Par exemple dans le cas des coefficients d'ondelettes, les coefficients de faible amplitude ont tendances à s'agglutiner. Les codeurs entropiques évoqués ci-dessus ne peuvent exploiter cette mémoire de la source quantifiée. Il existe néanmoins des codeurs capables d'exploiter les statistiques d'ordre supérieures : les codeurs entropiques adaptatifs et les codeurs par plages de zéros. Les codeurs par plages [10] (ou run-length en anglais) sont particulièrement adaptés pour coder les coefficients d'ondelettes.

L'idée de base consiste à coder la longueur d'une série d'échantillons nuls plutôt que de coder chaque échantillon indépendamment. Le code run-length doit être associé ensuite à un codeur entropique classique.

Le principe de compression RLE (Run Length Encoding, parfois notée RLC pour Run Length Coding) est assez simple à mettre en œuvre. Il repose sur le fait que dans une image, il existe de nombreuses répétitions d'un même pixel, ou d'une même séquence de pixels, tous juxtaposés.

Ainsi, au lieu de coder chaque pixel d'une image, le RLE regroupe les valeurs voisines identiques et ne transmet la valeur qu'une seule fois, précédée par le nombre de répétitions.

La compression RLE est régie par des règles particulières permettant de compresser lorsque cela est nécessaire et de laisser la chaîne telle quelle lorsque la compression induit un gaspillage lors que dans la redondance des caractères est faible.

III.3) Codage Lempel-Zif-Welsh (LZW):

Le codage LZW [8] est une technique de compression réversible qui peut être appliquée à tout type de fichier de données, que ce soit : texte, image, fichier informatique...elle a été adoptée pour la mise en œuvre du format de compression d'images 'GIF'.

Le principe général de cette méthode consiste à créer un dictionnaire contenant toutes les répétitions.

Il doit être construit de la même manière, à la compression et à la décompression, et contenir les mêmes informations.

Tous les ensembles de lettres qui sont lus sont placés dans le dictionnaire, et sont numérotés. A chaque fois qu'un ensemble est lu, on regarde s'il en existe déjà un qui est identique. Si c'est le cas, on émet son numéro vers le fichier compressé. Sinon, on le rajoute à la fin du dictionnaire, et on écrit chacune des lettres dans le fichier compressé.

Quand on écrit un numéro au lieu d'écrire des lettres, il y a un gain de place, mais pour cela, il faut déjà avoir beaucoup de chaînes dans le dictionnaire. L'apprentissage est donc nécessaire pour que la méthode soit efficace.

Les caractéristiques essentielles de LZW sont:

*Il n'existe pas de table d'en-tête : le dictionnaire est construit au fur et à mesure de la lecture du fichier tant à la compression qu'à la décompression.

*L'algorithme ne fonctionne pas sur un nombre fixe de motifs mais apprend les motifs du fichier durant la lecture du fichier à comprimer.

*La compression se fait en une seule lecture.

IV. COMPRESSION AVEC PERTE D'INFORMATIONS :

Contrairement aux méthodes de compressions sans pertes d'informations, ce type de compression comporte une perte de données pendant le processus. Le résultat qu'on peut en obtenir est une version dégradée de l' image originale. Le but de ce type de compression est d'éliminer le plus d'informations possible sans atténuer la qualité de l'image perçue par le système visuel humain. On peut distinguer deux grands types de compression pouvant générer des pertes: le codage prédictif et le codage par transformée, ce dernier est de loin le plus utilisé dans la compression d'image.

Un codeur avec pertes idéal [3] est composé d'un module de transformée, un module de quantification et un codeur entropique.

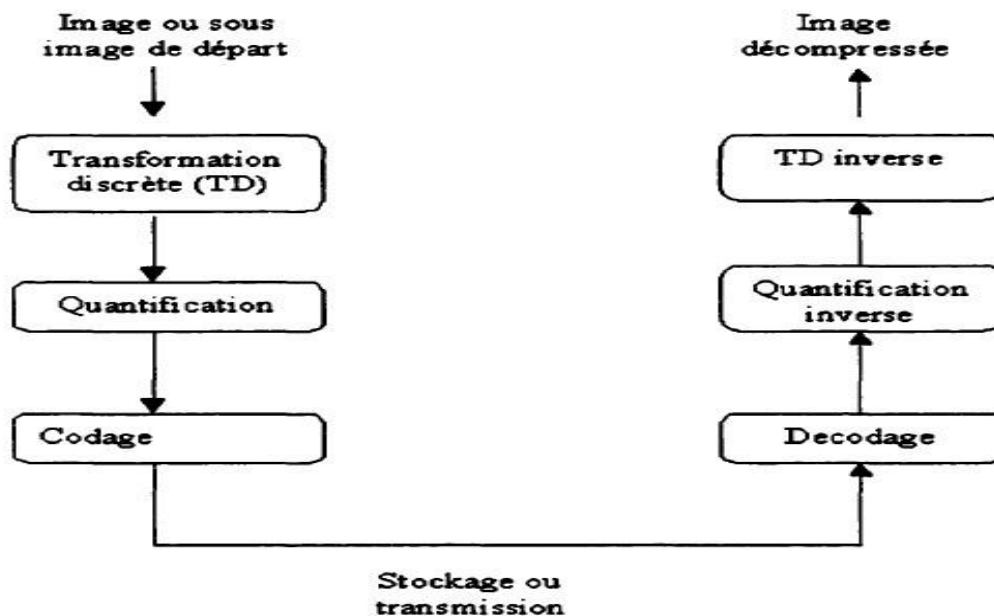


Figure (I-2) : Schéma synoptique du codeur

IV.1) Quantification

On distingue en général deux types de quantification: la quantification scalaire et la quantification vectorielle [4].

IV.1.1). Quantification scalaire :

Soit 'u' une variable aléatoire de densité de probabilité p(x). Une quantification est alors une fonction 'Q' définie sur 'R' et à valeurs dans 'Z' qui à 'u' associe 'ü'.

En général, l'image de 'Q' est un ensemble fini { ; j=1 ;... ; L} où 'L' est le nombre de niveaux de quantification (on dit aussi de reconstruction). On choisit souvent pour 'Q', une fonction en escalier (Figure (I-3)).

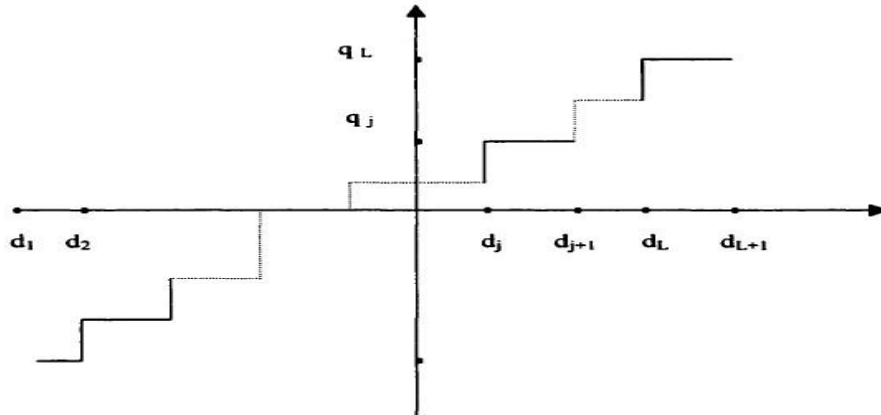


Figure (I-3) : Schéma synoptique d'une fonction en escalier

La règle de quantification est alors la suivante : on définit $d_j \{ j=1 ; \dots ; L+1 \}$, un ensemble croissant de niveaux de transitions ou de décisions et l'application 'Q', est alors définie par $Q([d_j ; d_{j+1}) = q_j, j = 1, \dots, L$, ceci revient à subdiviser l'ensemble des valeurs de u en L intervalles $(d_j ; d_{j+1})$ et à associer à chaque 'u' appartenant à l'intervalle $(d_j ; d_{j+1})$, la valeur q_j .

L'objet de la quantification scalaire revient à déterminer les niveaux de transition d_j et de quantification q_j optimaux, connaissant la densité de probabilité et se fixant un critère d'optimisation.

De toute évidence, l'application ainsi définie introduit par conséquent une distorsion entre les valeurs de 'u' et $Q(u)$ qu'il faut rendre minimale.

Si l'on prend pour critère d'optimisation, la minimisation de l'erreur quadratique moyenne, pour un nombre donné de niveaux de transition, on obtient un quantificateur optimal au sens des moindres carrés. C'est le quantificateur de LloydMax [11].

En désignant par $d_j (j = 1 : L+1)$, (resp. $q_j (j = 1 : L)$) les niveaux de transition (resp. de quantification) et par $p(x)$ la fonction de densité de probabilité, l'erreur quadratique se définit par :

$$Erreur = \sum_{j=1}^L \int_{d_j}^{d_{j+1}} (x - q_j)^2 p(x) dx \quad (1.1)$$

La minimisation de cette fonction revient à trouver les quantités et solutions du système d'équations suivant:

$$d_j = \frac{1}{2}(q_j + q_{j-1}) \quad (1.2)$$

$$q_j = \frac{\int_{d_j}^{d_{j+1}} x p(x) dx}{\int_{d_j}^{d_{j+1}} p(x) dx} \quad (1.3)$$

Ces équations sont non linéaires et par conséquent nécessitent la mise en œuvre d'algorithmes spécifiques à des systèmes non linéaires. Elles montrent que les niveaux de décision sont les centres des intervalles délimités par les niveaux de reconstruction et que les niveaux de quantification sont les moyennes normalisées dans chaque intervalle de décision.

On modélise le plus souvent la distribution de densité des niveaux de gris par une Gaussienne ou une Laplacienne. Ces deux densités de probabilité sont des cas particuliers de la Gaussienne généralisée dont la formule est donnée par :

$$p(x) = \alpha * e^{-|\beta(x)|^r} \quad (1.4)$$

$$\text{Avec : } \alpha = \frac{\beta r}{2r(\frac{1}{r})} \quad \text{Et} \quad \beta = \frac{1}{\delta} \left[\frac{r^3}{r^{\frac{1}{r}}} \right]^{\frac{1}{2}}$$

Où est la variance de l'image et $r(r)$, la fonction Gamma. Notons que pour $r=1$, on a la densité de probabilité Laplacienne et celle de la gaussienne pour $r = 2$.

Si l'on suppose que la densité de probabilité est uniforme, les niveaux de décision et de quantification du quantificateur optimal sont alors définis par :

$$d_j = d_i + (j - 1)p \quad (1.5)$$

$$q_j = d_j + \frac{p}{2} \quad (1.6)$$

$$p = (d_{L+1} - d_i)/L \quad (1.7)$$

Ce quantificateur est souvent appelé quantificateur linéaire. Au lieu de la fonction densité de probabilité, on peut utiliser l'histogramme de l'image à quantifier.

En général, les lois ne sont pas connues ou sont complexes à évaluer, ce qui complique le calcul.

Pour certaines lois classiques (loi normale, loi de Laplace...), ces valeurs sont données par des tables, ce qui facilite la mise en œuvre de la quantification quand on utilise ces modèles.

IV.1.2). Quantification vectorielle:

Le principe général du quantificateur vectoriel consiste à quantifier une suite d'échantillons (vecteur) d'une image au lieu de quantifier chaque valeur comme c'est le cas pour la quantification scalaire.

Pour réaliser ceci, on commence par créer un catalogue de vecteurs \hat{U} ; $i = 1..N_c$ de référence appelé Codebook ou dictionnaire à partir d'un ensemble d'images tests par un algorithme d'apprentissage.

L'image à quantifier est décomposée en un ensemble de vecteurs U_i , de même taille que ceux du Codebook. Ensuite, utilisant une métrique 'd'; chaque vecteur U_i de l'image à quantifier est comparé à tous les vecteurs \hat{U} de la séquence d'apprentissage générée précédemment, il s'agit de choisir dans le Codebook le vecteur \hat{U}_k tel que $d(U_k; \hat{U}_k) \geq d(U_i; \hat{U}_i)$; $i = 1..N_c$.

La distorsion mesurée par d doit être minimale et permettre d'obtenir une image de bonne qualité visuelle. La métrique communément utilisée pour ce type de quantification est la métrique euclidienne qui correspond à

$$d^2(U, \hat{U}) = \sum_{i=1}^N (U(i) - \hat{U}(i))^2 \quad (1.8)$$

On peut envisager, pour mieux tenir compte de la valeur de chaque pixel, une pondération de cette métrique :

$$d_{\omega}^2(U, \hat{U}) = \sum_{i=1}^N \omega_i (U(i) - \hat{U}(i))^2 \quad (1.9)$$

Où les ω_i sont des poids pouvant être identiques. Après avoir sélectionné parmi les vecteurs du Codebook le vecteur le plus proche (au sens de la métrique choisie) du vecteur à quantifier, ce dernier est représenté par son indice qui peut alors être transmis ou stocké en vue de la reconstruction de l'image originale. En fait si l'on a une image de taille $N \times N = 512 \times 512$, on peut la décomposer en 512 vecteurs de taille 512 chacun.

Son indice est alors codé sur $\log 2^N$ bits c.à.d 9 bits. On se rend ainsi compte que si au lieu de manipuler (stocker ou transmettre) l'image toute entière on ne travaille que sur les indices, on réalise un gain important au niveau de la quantité d'information à traiter.

A la réception d'un processus de transmission, l'image originale est reconstruite en utilisant simplement une table de correspondance.

Il est à noter que soit le Codebook est transmis en même temps que les index ou bien un autre dictionnaire est construit de manière identique c.à.d. en utilisant la même métrique et les mêmes images dans la séquence d'apprentissage que celles utilisées pour la quantification.

Les difficultés principales pour l'utilisation de ce type de quantification sont :

- La génération du dictionnaire ou Codebook, cette phase permettant la création du catalogue de référence est très importante et dépend de la structure et de la nature de l'image.
- La structuration du dictionnaire pour une recherche efficace et rapide des index de référence.

S'il est vrai que la mise en œuvre de cette quantification est complexe, elle permet d'obtenir des taux de compression importants. Par exemple, si l'on considère comme indiqué ci-dessus un dictionnaire formé de vecteurs de taille 512 chacun, avec les index codés sur 9 bits, ceci donne un codage de 9/512 bit/pixel (bpp), soit un taux de compression de $512 * 8/9$ pour une image ayant 2^8 niveaux de gris.

Remarquons que ce taux ne dépend pas de la taille de l'image, mais de la longueur des mots (ici vecteurs) du dictionnaire.

En général, si l'on veut coder un bloc de taille $n*m$ pixels avec un dictionnaire ayant des mots sur d bits, on a un rapport de $d/n*m$ bit/pixel d'où un taux de compression de $n*m*b/d$ pour une image codée sur 2 bits.

Les expériences quantitatives réalisées [5] montrent que la quantification vectorielle est meilleure que la quantification scalaire parce qu'elle donne de meilleurs taux de compression.

IV.2) Compression par transformation:

Les méthodes de compression par transformation n'agissent pas directement sur l'image numérique dans sa représentation canonique, mais sur le domaine de sa transformée.

Cette transformation peut être linéaire ou non.

Il est bien connu qu'une transformation peut permettre de mettre en évidence certaines propriétés de l'image que la représentation originale ou canonique ne laisse pas apparaître.

En partant d'un ensemble de valeurs numériques corrélées d'une image, le but est d'obtenir un autre ensemble de valeurs le moins corrélées possible dans l'espace transformé.

En général, les schémas de codage par transformation subdivisent l'image de taille $N \times N$ en sous-images de taille plus petites avant de faire subir à chacune de ces sous images une transformation.

On privilégie les transformations qui sont unitaires et qui conservent l'énergie. La transformation consiste en la décomposition de l'image selon une base adéquate de fonctions telles que:

- Les coefficients de la transformation soient indépendants

- Qu'un nombre minimum de ces coefficients contienne une proportion importante de l'énergie de l'image.

Ainsi, on pourra mettre à zéro certains d'entre eux sans nuire de manière significative ni à la quantité d'énergie, ni à l'aspect visuel de l'image reconstruite.

On peut distinguer plusieurs transformées dont [11]:

- ✓ Transformation de Karhunen-Loeve (TKL).
- ✓ Transformation de Fourier discrète (TFD).
- ✓ Transformation de Hadamard (TH).
- ✓ Transformation de Haar (THA).
- ✓ Transformation en cosinus discrète (DCT).
- ✓ Transformation par ondelette discrète (DWT).

La transformation de Fourier et celles qui en découlent, telles la transformation en sinus, la transformation en cosinus, sont très utilisées en analyse et en filtrage du signal [12].

Ces transformations peuvent être mises en œuvre à l'aide d'algorithmes rapides comme la FFT et ses variantes.

La variable de l'espace transformé étant la fréquence une telle décomposition permet de mieux observer la répartition fréquentielle de l'image.

Étant donné que ce sont les premières harmoniques qui contiennent la quasi-totalité de l'énergie, il est donc possible de mettre à zéro une proportion importante des coefficients et de coder l'image à moindre coût.

Malgré la rapidité de la transformation de Fourier, elle décompose l'image en une partie réelle et une partie imaginaire pouvant se convertir en module et argument ce qui n'est pas facile à manipuler ou à interpréter.

Les traitements de ces données peuvent s'avérer lourds, d'où la préférence accordée à la transformation en cosinus qui bénéficie de toutes les caractéristiques de la FFT.

La transformée en cosinus a été choisie comme standard par JPEG (Joint Photographie Experts Group) pour le codage d'images fixes.

Actuellement, la transformée qui est sujette à multitudes recherches et qui a été choisie comme standard du JPEG2000 est la transformée en ondelette, cette transformée a révolutionné plusieurs domaines de l'imagerie notamment la compression d'images, où elle permet un meilleur taux de compression que les autres et ceci avec une meilleure qualité d'images. [13].

IV.3) Codage par prédiction:

Les méthodes prédictives proposent de transformer de manière réversible les chaînes de symboles en des chaînes plus simples pour l'algorithme de compression.

Il s'agit d'aider celui-ci en introduisant un modèle permettant de prédire chaque symbole apparaissant dans la chaîne en fonction des symboles précédents et de coder à l'aide de l'algorithme de codage universel l'erreur de prédiction au lieu du symbole lui-même. Ceci suppose que les symboles possèdent une interprétation, ce qui est le cas pour les images.

Il n'est donc pas étonnant qu'en 1995, lorsque l'algorithme PNG, conçu comme une alternative non encombrée par des brevets de GIF, a été proposé, cette prédiction y'a été ajoutée.

Les modèles proposés utilisent des prédictions linéaires de l'intensité lumineuse en fonction de celle des voisins déjà connus.

Les erreurs de prédictions sont alors codées par une autre variante de l'algorithme

LZ77, c'est l'algorithme "deflate".

L'amélioration de performances est notable puisque l'on peut atteindre des taux de l'ordre de 3 avec l'algorithme PNG [14].

IV.4) Codage des Sous-Bandes:

IV.4.1). Algorithme EZW:

Le premier algorithme inter-bandes pour les images 2D se nomme EZW (*Embedded coding with Zéro tree of Wavelet coefficients*) [13], [15] proposé en 1993 par Jerrom Shapiro.

L'ensemble des méthodes qui ont suivi s'appuient sur des techniques communes. Elles exploitent complètement la notion de multi-résolution associée aux ondelettes.

Leur schéma de codage utilise un modèle simple pour caractériser les dépendances inter-bandes parmi les coefficients d'ondelettes localisés dans les sous bandes ayant la même orientation.

IV.4.2) Algorithme SPIHT:

L'algorithme SPIHT (Set Partitioning In Hierarchical Trees) [18] reprend les principes évoqués dans EZW tout en proposant de partitionner récursivement les arbres de coefficients.

Ces deux algorithmes seront détaillés dans le chapitre III.

IV.4.3) Algorithmes EQ:

L'algorithme d'Estimation Quantization (EQ) [14] consiste à modéliser les coefficients de chaque sous-bande par un mélange de gaussiennes généralisées, dont la variance est conditionnée par un voisinage causal et le paramètre de forme est fixe. L'expression de la gaussienne généralisée est donnée par la formule suivante :

$$P(X = x) = \frac{\alpha\beta}{2\lambda(\frac{1}{\beta})} e^{-|\alpha x|^\beta} \quad (1.11)$$

Où β est le paramètre de forme, et α est donné en fonction de la variance δ^2 par

$$\alpha^2 = \frac{\lambda(\frac{3}{\beta})}{\delta^2(\frac{1}{\beta})}$$

Cette loi se réduit à l'expression d'une Laplacienne lorsque $\beta=1$ et d'une gaussienne lorsque $\beta= 2$. Elle représente de manière assez fidèle la distribution stationnaire des coefficients pour un paramètre β autour de 0,6 – 0,7. Dans l'algorithme EQ, ce paramètre est estimé pour chaque sous-bande à l'encodeur et transmis au décodeur. La variance est quant à elle estimée localement par maximum de vraisemblance, en fonction de la valeur des coefficients précédemment quantifiés dans un voisinage causal du coefficient considéré.

Lorsque tous les coefficients voisins sont quantifiés à zéro, la variance du coefficient est imprévisible et une valeur par défaut est utilisée. Cette variance par défaut est également estimée à l'encodage et transmise au décodeur.

Une fois les paramètres de la gaussienne généralisée estimés pour le coefficient courant, celui-ci est quantifié pour le débit cible donné. Un quantificateur uniforme par zone morte est sélectionné parmi un ensemble de quantificateurs prédéfinis indexés par la pente $-\tau$ correspondante sur la courbe débit-distorsion. Les probabilités des symboles quantifiés sont également stockées dans cette table pour l'étape de codage entropique.

Ce codeur a l'avantage d'être simple et de complexité très réduite, offrant des performances très satisfaisantes. Il a par contre l'inconvénient de ne pas permettre un décodage progressif efficace de l'image, car chaque coefficient est codé totalement avant de passer au suivant.

V. EVALUATION DE LA QUALITE DE COMPRESSION

V.1 Taux de compression :

Le taux de compression est une grandeur qui calcule le rapport entre le nombre de bits des données originales avec celui présent dans les données compressés. Le taux de compression est noté par T_c .

$$T_c = \frac{\text{Le nombre de bits de l'image compressée}}{\text{Le nombre de bits de l'image original}} \quad (1.12)$$

Il existe une autre quantité permettant de définir le taux de compression maximal sans perte d'information c'est l'entropie.

V.2 Taux d'information (Entropies):

L'entropie est une grandeur qui caractérise la quantité d'information que contient une image.

Par exemple, une image dont tous les pixels ont la même valeur contient très peu d'informations car elle est extrêmement redondante, donc son entropie est faible. En revanche une image dont tous les pixels ont une valeur aléatoire contient beaucoup d'informations, son entropie est forte. L'entropie (en bits) est calculée par la formule suivante :

$$H = - \sum_{i=1}^{i=N} P_i \log_2 P_i \quad (1.13)$$

Où N est le nombre de bits par pixels.

L'entropie H d'une image originale fournit le débit minimal qu'il est possible d'atteindre par compression, pixel par pixel sans dégrader l'image, est par la même, un taux de compression sans perte maximale.

V.3 Mesures de distorsion (Mesures de fidélité):

Deux techniques sont utilisées pour évaluer la distorsion : subjective et objective.

- Les méthodes subjectives, nécessitent des tests psycho visuels de l'œil humain.

Les tests sont réalisés à plusieurs échelles avec des groupes de personnes.

- Les méthodes objectives, s'agissant de définir des quantités permettant d'évaluer numériquement la qualité de l'image reconstruite.

La distorsion (D) est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale. La mesure de distorsion utilisée généralement en compression d'image, est l'erreur quadratique moyenne MSE (Mean Square Error). Cette grandeur est définie par la moyenne des écarts au carré entre le pixel I (i, j) de l'image originale et le pixel \hat{I} (i, j) de l'image reconstruite.

$$MSE = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I(i, j) - \hat{I}(i, j)]^2 \quad (1.14)$$

Le rapport signal sur bruit (Signal to Noise Ratio) est défini par

$$SNR = 10 \log_{10} \frac{\sum_{i=0}^{I-1} \sum_{j=0}^{J-1} x[(i, j)]^2}{MSE} \quad (1.15)$$

Le rapport signal crête sur bruit pour une image dont le maximum est $(2^R - 1)$ dénoté PSNR (Peak Signal to Noise Ratio) entre la source initiale et celle distordue, où R est le nombre de bits de l'image originale :

En compression d'images le PSNR d'une image de taille $(8 \times (512)^2)$ bits ($(512)^2$ indique une image de taille 512 par 512 pixels, chaque pixel est codé sur 8 bits) est défini plus souvent par :

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (1.16)$$

Le PSNR d'une image couleur RGB est défini par l'équation :

$$PSNR = 10 \log_{10} \frac{255^2 \times 3}{MSE(R) + MSE(G) + MSE(B)} \quad (1.17)$$

Les mesures de distorsion sont très utiles pour déterminer la performance d'une méthode par rapport à d'autres. Ces mesures de distorsion sont objectives et simples à calculer. Certaines méthodes de compression recherchent le meilleur compromis entre la performance et la distorsion.

Parmi les métriques les plus populaires et les plus utilisées, on trouve SSIM (Structural Similarity Index) qui est basée sur la similarité structurelle entre deux images comme le cas de l'œil humain qui est sensible aux changements dans la structure. Cette métrique est basée sur trois composantes : la luminance, le contraste et la structure de l'image.

Soit l'image originale et compressée respectivement

L'équation de la métrique SSIM est la suivante :

$$SSIM(x, y) = [l(x, y)]^\alpha [c(x, y)]^\beta [s(x, y)]^\gamma \quad (1.18)$$

Avec α , β , et γ sont des paramètres qui définissent l'importance des trois composantes $l(x, y)$ qui est la comparaison de la luminance, $c(x, y)$ est la comparaison du contraste, et $s(x, y)$ est la comparaison structurelle :

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad (1.19)$$

Avec : C_1, C_2, C_3 sont des constantes, μ_x, μ_y sont les moyennes et $\sigma_x, \sigma_y, \sigma_{xy}$ sont les variances

Enfin, la mesure de la qualité peut fournir une carte spatiale de la section locale qualité d'image, qui fournit plus d'information sur la qualité de l'image de dégradation, ce qui est utile dans les applications d'imagerie médicale. Pour l'application, nous avons besoin d'une seule mesure globale de l'ensemble qualité d'image qui est donnée par la formule suivante:

$$MSSIM(X, Y) = \frac{1}{M_1} \sum_{j=1}^M SSIM(x_j, y_j) \quad (1.20)$$

Où X et Y sont respectivement les images de référence et dégradée, x_j et y_j est le contenu des images à la fenêtre i-ème local.

M est le nombre total de fenêtres locales dans l'image. Les valeurs MSSIM présentent une plus grande cohérence avec la qualité visuelle.

VII. CONCLUSION

Dans ce chapitre nous avons fait un tour d'horizon sur les différentes techniques de compression d'images médicales ainsi que les paramètres d'évaluation de leurs qualités .

On en a déduit qu'elles peuvent être classées en deux grandes catégories d'après leurs propriétés, la compression sans pertes d'informations qui permet de retrouver exactement toute l'information contenue dans l'image originale ainsi que la compression avec pertes d'informations qui comporte une perte de données pendant le processus.

Chapitre II : transformée on ondelette

I. INTRODUCTION

L'objectif de ce chapitre est de présenter les ondelettes et la transformée d'ondelette que nous allons utiliser dans la suite de ce travail.

Nous les situerons dans le panorama le plus large de l'analyse du signal 2D par les ondelettes, puis nous rappellerons leurs propriétés fondamentales ainsi leurs applications.

II. TRANSFORMEE EN ONDELETTES

L'intérêt que présente de nombreuses transformées pour comprimer l'information est de projeter le signal sur une base de fonctions orthogonales, c'est-à-dire de distribuer l'énergie de ce signal sur des composantes décorrélées entre elles. La transformée en ondelettes permet d'obtenir une représentation locale de l'information contenue dans le signal analyse, en respectant le principe de l'incertitude, alors que la transformée de Fourier fournit une information globale sur le signal. Similairement à la transformée de Fourier, la transformée en ondelettes permet de décomposer un signal sur une base de fonctions élémentaires : les ondelettes. A partir d'une fonction de base, appelée ondelette-mère ou ondelette analysante, on construit cette base de fonctions élémentaires par des opérations de dilatation et de translation. Dans le cas unidimensionnel, cette opération s'obtient à l'aide de la relation suivante:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \quad (a,b \in \mathbb{R}, a \neq 0) \quad (2.1)$$

a est appelé paramètre de dilatation ou de compression liée à la fréquence ou à la résolution tandis que b est le paramètre de translation lié au temps ou à l'espace.

III. ANALYSE MULTIRÉSOLUTION

III.1. Principe

L'idée de base en analyse multi résolution est de pouvoir séparer les composantes hautes fréquences du signal qu'on appelle détails des composantes basses fréquences qui constituent une approximation du signal à une résolution donnée. On pourra alors exécuter les traitements dont on veut sur ces nouvelles composantes du signal comme les filtrages, les compressions ou bien les détections de structures. Par ailleurs, l'intérêt réside dans le fait qu'il

est toujours possible de reconstruire le signal de départ à partir de ces composantes à l'aide des procédures inverses

Ce principe peut être schématisé d'après la Figure II.1 suivante :

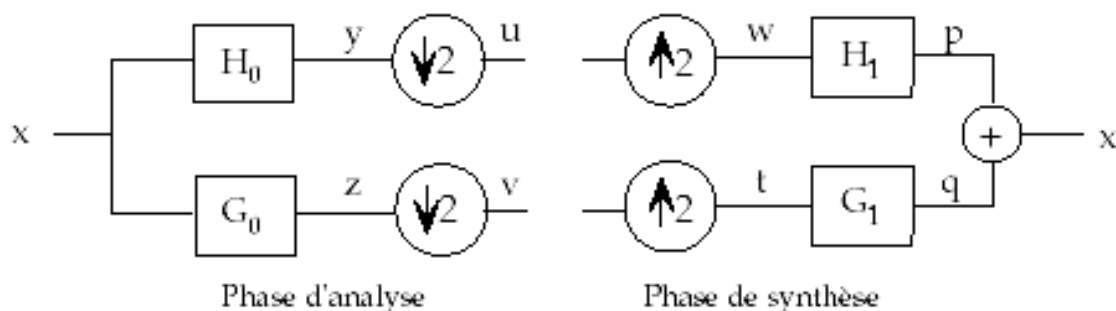


Figure II.1. Banc de filtres numériques en AMR

Pour la phase d'analyse, (H_0 , G_0) est la paire de filtres utilisés, H_0 étant un filtre passe-bas tandis que G_0 est un filtre passe-haut. L'opérateur $\downarrow 2$ désigne l'opération de sous échantillonnage de données (downsampling

). Cette opération est nécessaire pour éviter que le volume de données issues de la phase d'analyse soit double. Pour la phase de reconstruction ou de synthèse, il suffit d'inverser les procédures précédentes. Une phase de dilatation des données avec insertion de zéros (upsampling), obtenu à l'aide de l'opérateur $\uparrow 2$, précède les opérations de filtrage. H_1 et G_1 étant ici les filtres conjugués respectifs de H_0 et de G_0 . Les filtres (H_0, G_0) ainsi constitués désignent ce qu'on appelle filtres miroirs en quadrature; H_1, G_1 désignent les filtres conjugués respectifs de H_0 et de G_0 .

Ainsi, à partir de l'utilisation d'un banc de filtres à reconstruction exacte défini par la donnée de la paire de filtres (H_0, G_0) pour l'analyse et la paire de filtres (H_1, G_1) pour la synthèse, on pourra retrouver les données initiales.

Notons que l'on peut poursuivre cette décomposition du signal jusqu'à un certain niveau d'échelle dépendant de la taille des données au départ. Si cette taille a pour valeur 2^n , n étant un entier, alors on peut aller jusqu'au niveau n . On obtient ce qu'on appelle une décomposition pyramidale des données. Le sommet de la pyramide représente ainsi la valeur moyenne des données initiales.

Normalement, l'analyse multi résolution commence par la détermination d'une fonction de base $\varphi(x)$ appelée fonction d'échelle plutôt qu'avec l'ondelette. Une fois l'analyse multi résolution définie, à l'aide de la fonction d'échelle, il est possible de définir l'ondelette associée.

III.2. Algorithmes de Mallat

On pourra approximer une fonction à l'aide d'un opérateur à une résolution donnée. Soit A_{2^j} l'opérateur qui approxime une fonction f à la résolution 2^j . Il existe alors une fonction $\varphi(x) \in L^2(\mathbb{R})$ appelée fonction d'échelle telle que:

$$\varphi_{j,k}(x) = 2^{-\frac{j}{2}} \varphi(2^{-j}x - k) \quad (j, k \in \mathbb{Z})$$

constitue une base orthonormée de V_j pour une valeur de j fixée.

L'approximation d'une fonction $f(x)$ à la résolution 2^j s'écrit alors à l'aide de la relation suivante :

$$A_{2^j} f = a_{j,k} = \langle f, \varphi_{j,k} \rangle = \int_{-\infty}^{+\infty} f(x) \overline{\varphi_{j,k}(x)} dx \quad (2.3)$$

Ainsi, en terme d'images, la différence d'informations comprise entre les résolutions 2^{j+1} et 2^j , appelée détails, peut être extraite en décomposant l'image suivant une base ortho normale d'ondelettes de $L^2(\mathbb{R})$. Soit :

$$C_{j,k}(f) = \langle f, \psi_{j,k} \rangle = \int_{-\infty}^{+\infty} f(x) \overline{\psi_{j,k}(x)} dx \quad (2.4)$$

Autrement dit, les coefficients d'ondelettes $C_{j,k}$ contiennent l'information perdue lorsque l'on passe de l'approximation de résolution 2^{j-1} à l'approximation de résolution inférieure 2^j (Mallat, 1989) tandis que les coefficients $a_{j,k}$ représentent l'approximation de l'image à la résolution 2^j . En pratique, le calcul des coefficients $C_{j,k}$ et $a_{j,k}$ se réalise par des

opérations de filtrages. Il existe une étroite relation entre banc de filtres à reconstruction exacte et la décomposition en ondelettes par analyse multi résolution. Ainsi, à la fonction d'échelle on associe un filtre passe-bas h qui permet de conserver les éléments caractéristiques du signal tout en éliminant le bruit. Un filtre passe-haut g est associé à l'ondelette et permet aux coefficients d'ondelettes d'être considérés comme des détails extraits d'une résolution à une autre.

III.3. Algorithmes et disposition des images dans une décomposition en 2D

Une image est un signal 2D. Ainsi, Mallat (1989) a fait une extension de l'étude 1D vue précédemment en 2D. Dans ce cas, il a introduit une fonction d'échelle 2D

$$\varphi(x, y) = \varphi(x) \varphi(y), \quad (2.5)$$

avec $\varphi(x)$ et $\varphi(y)$ sont des fonctions d'échelle à une dimension. Il a défini alors les trois ondelettes suivantes :

$$\begin{aligned}\psi^H(x,y) &= \varphi(x) \psi(y) \\ \psi^V(x,y) &= \psi(x) \varphi(y) \\ \psi^D(x,y) &= \psi(x) \psi(y)\end{aligned}\quad (2.6)$$

où les ondelettes YH, YV, YD privilégient respectivement les orientations horizontale, verticale et diagonale. L'algorithme de décomposition et la disposition des images dans une décomposition en ondelettes par analyse multi résolution sont donnés respectivement sur les Figure II.2 et Figure II.3.

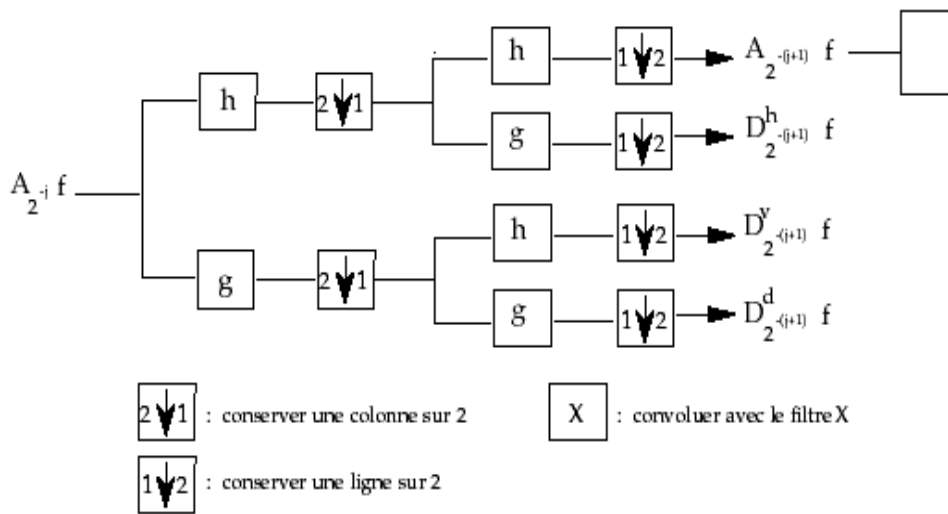


Figure II.2 : Algorithme de décomposition en ondelettes par AMR d'un signal en 2D

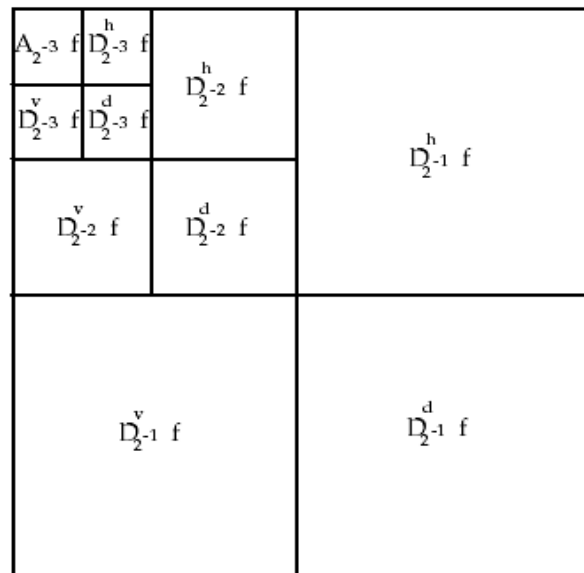


Figure II.3 : Disposition des images dans une décomposition en ondelettes par analyse multi résolution (Cas où la décomposition s'effectue jusqu'au 3ème niveau)

Dans le cas de 2D, pour un $J > 0$, la représentation en ondelettes orthogonales de l'image de départ est donnée par l'ensemble de $3J+1$ images:

$$\left(A_{2^{-j}} f, (D_{2^j}^h f)_{-J \leq j \leq -1}, (D_{2^j}^v f)_{-J \leq j \leq -1}, (D_{2^j}^d f)_{-J \leq j \leq -1} \right)$$

$A_{2^{-j}} f$ est l'approximation de l'image à la résolution 2^{-j} et les $D_{2^j} f$ images donnent les images de détails pour différentes orientations et résolutions. Si l'image originale à N pixels, chaque image $A_{2^j} f$, $D_{2^j}^h f$, $D_{2^j}^v f$, $D_{2^j}^d f$ a $2^j N$ pixels ($j < 0$). Ainsi on retrouve le nombre total de pixels de l'image de départ. Le volume des données n'a pas augmenté.

L'algorithme de reconstruction par ondelettes à l'aide de l'analyse multi résolution est montré sur la Figure II.4 ci-après :

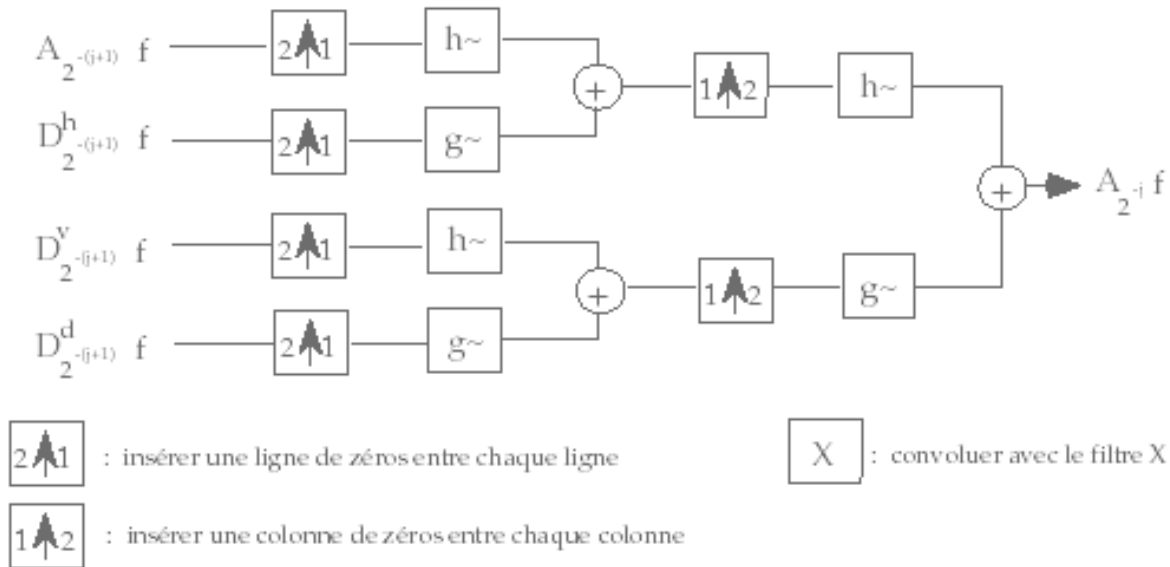


Figure II.4 : Algorithme de reconstruction a l'aide des ondelettes par AMR d'un signal en 2D.

VI. Ondelettes biorthogonales

La nécessité de la construction des bases bi orthogonales d'ondelette s'est accrue pour deux raisons :

- Obtenir une base de représentation dans laquelle on s'est affranchi de la contrainte d'orthogonalité des premières bases d'ondelettes.
- Utilité au codage par bancs de filtres où dans la conception initiale, les filtres d'analyse sont différents des filtres de synthèse.

Il s'agit dans tous les cas d'introduire une flexibilité dans la construction des bases d'ondelettes. Cette construction commence par la construction des fonctions qui définissent une analyse multi résolution à partir des réponses impulsionnelles des filtres. Notons qu'une telle base bi orthogonale avait déjà été construite par Tchamitchian, par une approche différente [37], [42].

a. Intérêt des ondelettes biorthogonales :

Les Ondelettes bi orthogonales sont à support compact et peuvent être symétriques. De plus, les conditions sur les filtres sont plus souples que pour les ondelettes orthogonales.

b. Inconvénient des ondelettes biorthogonales :

Cette transformée nécessite deux ondelettes. De plus, elles peuvent introduire des régularités très différentes.

V. CONCLUSION

Dans ce chapitre, nous avons vu une introduction à la théorie des Ondelettes qui sont les plus utilisées en traitement d'images en particulier pour la compression.

Nous avons parlé de l'analyse multi résolutions et ses propriétés, la transformée en ondelette bi orthogonale ainsi que leurs avantages et leurs inconvénients.

Chapitre III : Algorithme SPIHT

Chapitre III : Codeur SPIHT

Avant d'entamer le codeur spiht, il est intéressant d'expliquer le codeur EZW qui est à l'origine de ce codeur.

I.Codeur EZW

I.1.Principe

Le premier algorithme inter-bandes pour les images 2D se nomme EZW (*Embedded coding with Zero tree of Wavelet coefficients*) [13], [15] proposé en 1993 par Jerrom Shapiro. L'ensemble des méthodes qui ont suivi s'appuient sur des techniques communes. Elles exploitent complètement la notion de multi-résolution associée aux ondelettes. Leur schéma de codage utilise un modèle simple pour caractériser les dépendances inter-bandes parmi les coefficients d'ondelettes localisés dans les sous bandes ayant la même orientation.

La **Figure (III.1)** illustre ces dépendances à travers toutes les échelles. Le modèle est basé sur l'hypothèse des arbres de zéros, qui suppose que si un coefficient d'ondelettes w est non significatif pour un seuil donné T , c'est-à-dire $|w| < T$, alors tous les coefficients de la même orientation dans la même localisation spatiale à des résolutions plus fines sont supposés non significatifs pour ce même seuil T . Cette hypothèse des arbres de zéros est illustrée en 2D sur la **Figure (III.2)**. On peut voir qu'un coefficient "parent" à une résolution donnée va engendrer 4 coefficients "enfants" à la résolution supérieure [15], [16], [17].

Notons que le noeud racine (correspondant aux coefficients de la dernière sous bande LL) de l'arbre a seulement 3 enfants, alors que tous les autres nœuds à l'exception extrémités en possèdent 4.

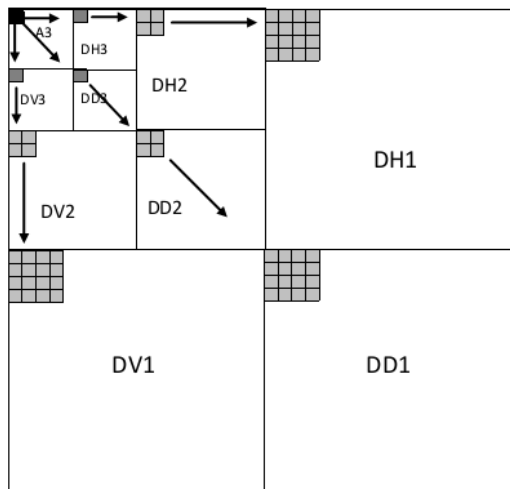


Figure III-1 : Structure d'arbre des coefficients d'ondelette

DH : Détails Horizontaux

DD : Détails Diagonaux

DV : Détails Verticaux

I.2.Algorithme de codage EZW

L'algorithme de codage est le suivant :

On part d'un seuil entier qui est T_n une puissance de 2 de type :

$$T_n = 2^n \text{ avec } n = \log_2(|\omega_{max}|) \text{ ou } \omega_{max}$$

Est le coefficient le plus grand en valeur absolue de toutes les sous-bandes. Les coefficients d'ondelettes sont scannés dans un ordre hiérarchique de la résolution la plus faible vers la plus grande, et chaque coefficient w est testé pour savoir si sa valeur absolue est supérieure ou égale au seuil T_n , c'est-à-dire s'il est significatif pour le seuil donné. Si un coefficient est trouvé significatif, il est codé suivant son signe par le symbole POS pour un coefficient positif et NEG pour un coefficient négatif. Il est alors placé dans la liste des coefficients significatifs. Si un coefficient est testé non significatif $|W| < T_n$ on examine tous ses descendants pour tester leur signification. Dans le cas où aucun descendant n'est pas significatif, on code un arbre de zéros (AZ). Si un descendant significatif apparaît on code un zéro isolé (ZI). Les coefficients qui descendent d'une racine d'arbre de zéros n'ont pas besoin d'être codés. Cette partie se nomme la passe dominante pour Shapiro ou passe de signification dans de nombreux articles.

Ensuite, la seconde passe (passe de raffinement ou subordonnée pour Shapiro) est réalisée sur les coefficients dans la liste significative. Pour chaque coefficient de cette liste, le bit situé dans le plan de bit inférieur (plan de bit) est codé. L'encodeur divise le seuil T_n par 2 ($T_n \rightarrow T_n / 2$) et exécute une nouvelle passe dominante et de raffinement.

2^{n-1} Cette procédure se poursuit jusqu'à ce que l'on atteigne le débit voulu. Si un coefficient est trouvé significatif à une passe postérieure, il sera encore dans la liste significative à la passe courante et n'aura pas besoin d'être identifié comme significatif une autre fois. Le nombre de coefficients dans cette liste croît de façon monotone au fur et à mesure que les seuils T_n décroissent. Si l'on va jusqu'au dernier plan de bits (LSB), on obtient un codage sans perte car il n'y a plus d'étape de quantification.

Le décodeur utilise un algorithme similaire. Il initialise tous les coefficients à zéro et scanne à travers les mêmes directions que l'encodeur. Le décodeur reçoit un symbole du flux binaire pour chaque coefficient. Si ce symbole est POS ou NEG, l'amplitude du coefficient est au-dessus du seuil et on détermine le signe. Dans les deux cas, le coefficient est placé dans la liste significative. Si un symbole AZ est reçu, aucun des descendants du coefficient courant n'est visité pendant la passe dominante. On réalise ensuite la passe de raffinement. Pour chaque coefficient dans la liste de signification, un bit est sorti du flux binaire. Si c'est un 1, il est utilisé pour remplacer le bit 0 à la localisation $\log_2(T_n) - 1$ dans la représentation binaire des coefficients [16].

II. Codeur SPIHT

II.1. Principe

SPIHT (*Set Partitioning In Hierarchical Trees*) est un algorithme récemment proposé par Amir Said et William A. Pearlman [42], basé sur une transformation en ondelettes discrètes (DWT).

Normalement, la plus grande partie de l'énergie d'une image est concentrée dans les basses fréquences. Par conséquent, plus le niveau de résolution de l'image est grand, plus les variances des images de détails doivent être faibles. De plus, il semble qu'il existe une similarité entre les pixels de même orientation spatiale et de résolutions différentes. En particulier, les amplitudes des pixels de même orientation spatiale semblent être assez bien ordonnées si l'on évolue dans la pyramide. On est alors naturellement conduit à adopter une organisation arborescente des coefficients (figure 3.6), où chaque noeud de l'arbre correspond à un pixel et ses descendants directs aux pixels de même orientation spatiale à la résolution suivante.

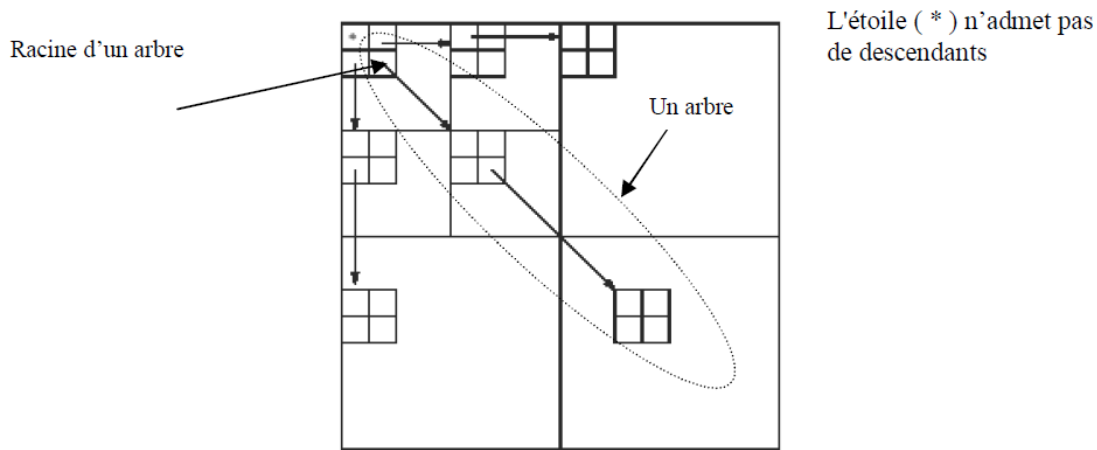


Figure III.2 : Dépendance de coefficients et arbres d'orientation spatiale dans SPIHT utilisant la décomposition dyadique avec deux résolutions

L'algorithme SPIHT utilise trois listes ordonnées qui contiennent les ensembles de test de la signifiante avec la même méthode pour le codeur et décodeur :

- la liste des ensembles insignifiants (**LEN**)
- la liste des pixels insignifiants (**LCN**)
- la liste des pixels significatifs **S** (**LCS**).

Il y a des listes de coordonnées (i, j) comme dans LCN et LCS qui représentent des coefficients individuels, mais la liste LIS représente des ensembles des coefficients, chaque ensemble est soit un ensemble $D(i, j)$ marqué d'une entrée de type A (tous les descendants d'un nœud) ou un ensemble $L(i, j)$ marqué d'une entrée de type B(tous les descendants d'un nœud excepté ses quatre enfants [42].

L'algorithme SPIHT (Set Partitioning In Hierarchical Trees) [18] reprend les Principes évoqués dans EZW tout en proposant de partitionner récursivement les arbres de coefficients. Ainsi, là où EZW codait un coefficient non significatif isolé ('ZI'), SPIHT effectue un partitionnement récursif de l'arbre de manière à déterminer la position des coefficients significatifs dans la descendance du coefficient considéré.

Les coefficients significatifs sont codés de manière similaire à EZW : leur signe est envoyé dès qu'ils sont identifiés comme étant signifiants et ils sont ajoutés à la liste des coefficients à raffiner.

Cet algorithme fonctionne également par plans de bits. Il offre des performances remarquables, atteignant celles d'EZW sans codage entropique. En ajoutant un codage entropique de l'information de signifiante, un gain supplémentaire entre 0.3 et 0.6 dB peut être obtenu [19].

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

La procédure de codage se déroule de la manière suivante :

La liste des coefficients signifiants est initialement vide, tandis que la liste de coefficients insignifiants contient les racines de chaque arbre (coefficients de la bande basse) et la liste d'ensembles insignifiants contient l'ensemble des descendants de chaque arbre.

Cette partition initiale est segmentée récursivement au moyen de deux règles. Si un ensemble de descendants d'un nœud est significatif, il est séparé en quatre coefficients fils directs de ce nœud, et l'ensemble des autres descendants.

Les fils directs sont ajoutés à la LCN ou la LCS en fonction de leur signifiante. Si au moins un élément de l'ensemble des autres descendants est significatif, cet ensemble est séparé en quatre ensembles insignifiants ajoutés à la LEN. Le fait de traiter les coefficients par groupes de quatre

permet d'effectuer un codage entropique efficace par la suite. Comme dans EZW, la passe de raffinement consiste à coder progressivement les bits de poids faibles des coefficients significatifs [13], [19].

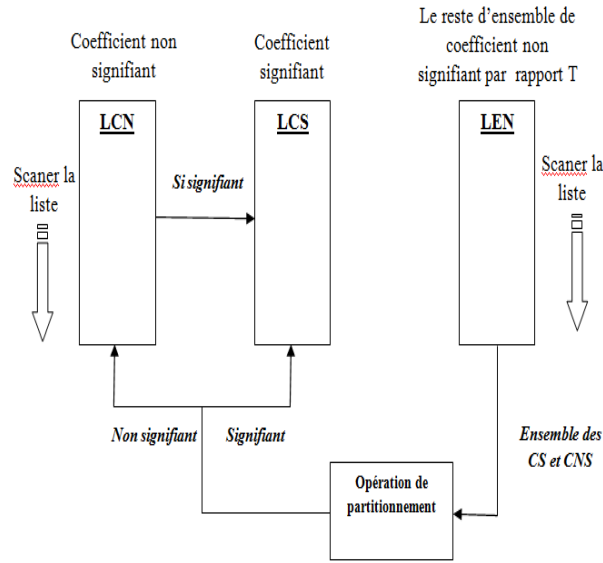


Figure III.3: Organigramme de l'algorithme de SPIHT

II.2. Algorithme de codage SPIHT

1. Initialisation :

$$n = \lceil \log_2(|\omega_{max}|) \rceil \leftrightarrow T_n = 2^n, LCS = \emptyset, LCN = (i, j) \in LL.$$

Sortie LEN contient les mêmes coefficients que LCN excepté ceux qui n'ont pas de descendants.

2. Passe de signification (étape de tri):

1.1. Pour chaque $(i, j) \in LCN$ faire :

1.1.1. Sortie $\sigma T_n(i, j)$

1.1.2. Si $\sigma T_n(i, j) = 1$ alors mettre (i, j) dans LCS et coder le signe de $w(i, j)$

1.2. Pour chaque $(i, j) \in LNE$ faire :

1.2.1. Si l'entrée est de type A

1.2.1.1. Sortie $\sigma T_n(D(i, j))$

1.2.1.2. Si $\sigma T_n(D(i, j)) = 1$ alors

Pour chaque $(l, m) \in O(i, j)$ faire : Sortie $\sigma T_n(l, m)$

✓ Si $\sigma T_n(l, m) = 1$ alors mettre (l, m) dans LCS et coder le signe de $W(l, m)$

✓ Si $\sigma T_n(l, m) = 0$ alors mettre (l, m) à la fin de LCN

Si $(i, j) \neq \emptyset$ alors mettre (l, m) à la fin de LEN comme une entrée de type B

1.2.2. si l'entrée est de type B

1.2.2.1. sortie $\sigma T_n(L(i, j))$

1.2.2.2. si $(IL(l, m)) = 1$ alors

- mettre (l, m) à la fin de LEN comme une entrée de type A
- supprimer (i, j) de LEN

3. Passe de raffinement :

Pour chaque coefficient $(i, j) \in LCS$ à l'exception de ceux inclus dans la même passe de signification (c.à.d pour le même n). Sortie le nième bit le plus significatif de $|W(i, j)|$ Ensuite, modifiez le seuil T en décrémentant n de 1 et renouveler l'étape de signification. Le codeur doit transmettre progressivement, au

décodeur, au moins le seuil de départ, le code des signes des coefficients significatifs ainsi que le bit le plus significatif de ces derniers.

Pour obtenir, l'algorithme de décodage, il suffit simplement de remplacer le mot Sortir par Entrée dans l'algorithme précédent.

II.3. Exemple

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

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

Figure III.4 : Exemple de SHAPIRO.

Seuils successifs:

T5= 32, T4= 16, T3= 8, T2= 4, T1= 2, T0= 1 (LSP, LIP, LIS sont respectivement LCS, LCN, LEN en français)

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

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

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

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

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

Pour obtenir l'algorithme de décodage, il suffit de remplacer le mot sortie par entrée dans l'algorithme précédent.

III. Conclusion

Nous avons expliqué dans ce chapitre les codeurs à arbre de zéros (EZW et SPIHT) adaptés au domaine des ondelettes .

L'application du codeur SPIHT (adapté au domaine d'ondelette) aux différents types d'images tests ainsi que les résultats obtenus feront l'objet du chapitre suivant.

Chapitre IV : Résultats et discussions

Résultats et discussions

I. Introduction

Le but de notre travail est d'effectuer la compression de données d'images avec la technique de la transformée en ondelettes (DWT) tout en gardant la qualité essentielle de l'image originale, c'est à dire, avoir un bon compromis entre PSNR et le taux de compression. Tout d'abord nous allons définir l'environnement de notre travail, puis nous décrivons la base d'image, les ondelettes utilisées pour les tests, finalement on discutera les résultats obtenus

II. Environnement de travail :

Le logiciel Matlab consiste en un langage interprété qui s'exécute dans une fenêtre dite d'exécution. L'intérêt de Matlab tient, d'une part, à sa simplicité d'utilisation : pas de compilation, déclaration implicite des variables utilisées et, d'autre part, à sa richesse fonctionnelle : arithmétique matricielle et nombreuses fonctions de haut niveau dans divers domaines (analyse numérique, statistique, commande optimale, représentation graphique, ...). Il est à noter que toutes les commandes sont en anglais et l'aide en ligne également l'intérêt primordial de notre choix revient à la pertinence et la rapidité de traitement des données matriciel.

II.1 Base d'image utilisée

Le choix des images de test est un problème fondamental lors de l'évaluation des systèmes de Compression, pour notre cas on a choisi des image médicales en niveaux de gris,elles sont acquises de la base de données Médicale **GE-SYSTEM**, de dimension 512*512 et codées sur 8 bits .

II.2. Choix des ondelettes

Les ondelettes choisies pour le test sont classées par famille. Chaque famille a un ensemble d'ondelettes dont le numéro qui suit le nom, indique l'ordre des filtres utilisés pour leurs Conception nous avons choisi les ondelettes les plus répondues en compression d'image :

- Ondelettes biorthogonales: Bior4.4, et Bior6.8.
- Ondelettes de Daubechies: Db6.
- Ondelette de Haar.

III. Résultats et discussions

Pour tester notre système de compression, on a appliqué notre algorithme sur les quatre images suivantes issues de la base de données GE-SYSTEM :

1. Image scanner du nez (Figure IV. 1)

2. Image Poitrine (Figure IV. 2)

3. Image IRM du cerveau (Figure IV. 3)

4. Image IRM Rachis cervical (Figure IV. 4)

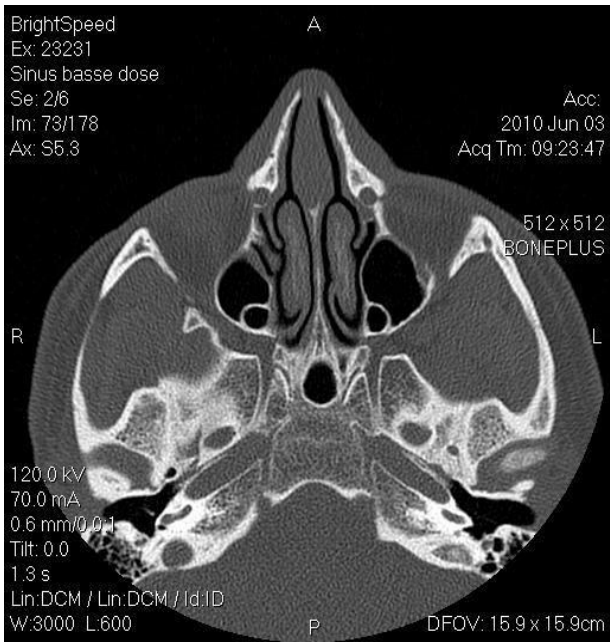
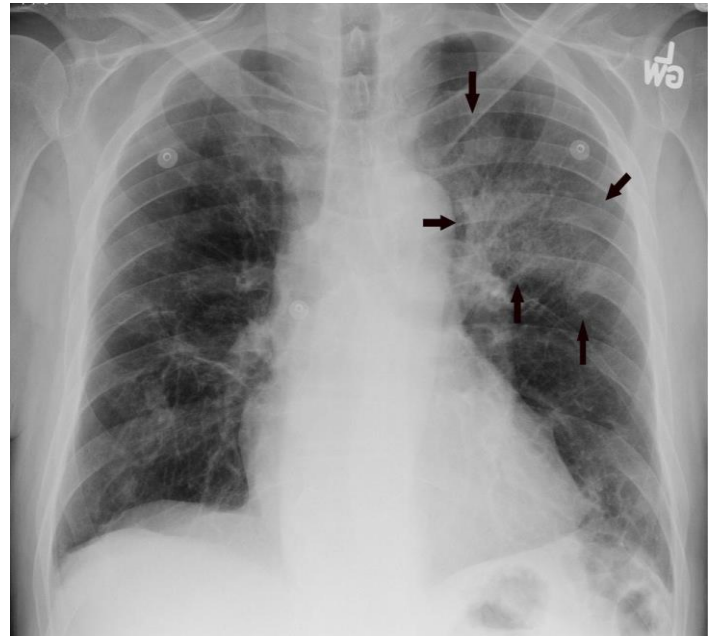
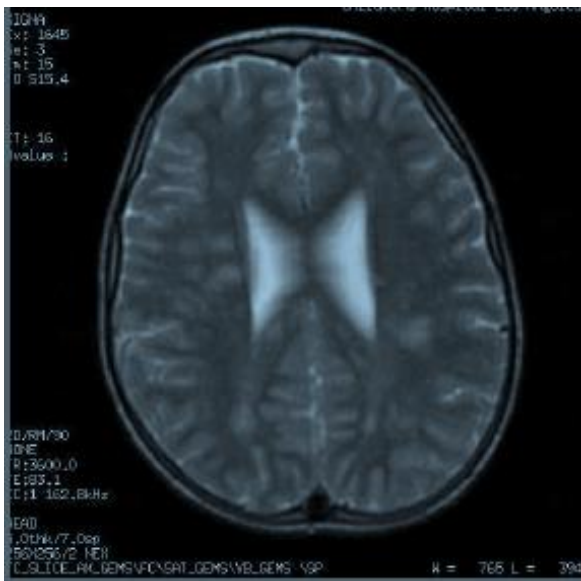


Figure IV. 1 :Scanner du nez



FigureIV .2 .Poitrine



FigureIV.3 :IRM du cerveau



(Figure IV.4) : Rachis cervical

III.1 Influence du niveau de décomposition

Pour tester l'influence du niveau de décomposition sur la performance de compression, on a fait varier le niveau de décomposition L de 3 à 7.

a. Application 1 :

Image test : nez

L'ondelette choisi : « haar »

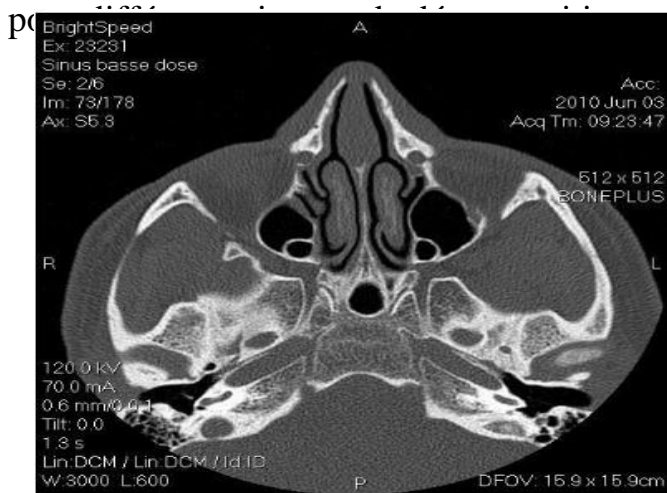
CR=1bpp

Niveau	L=3	L=4	L=5	L=6	L=7
PSNR	28.80	31.86	32.63	32.77	32.81
MSE	85.75	42.86	35.51	34.35	34.06
T(s)	21.43	32.10	36.02	35.93	36.11

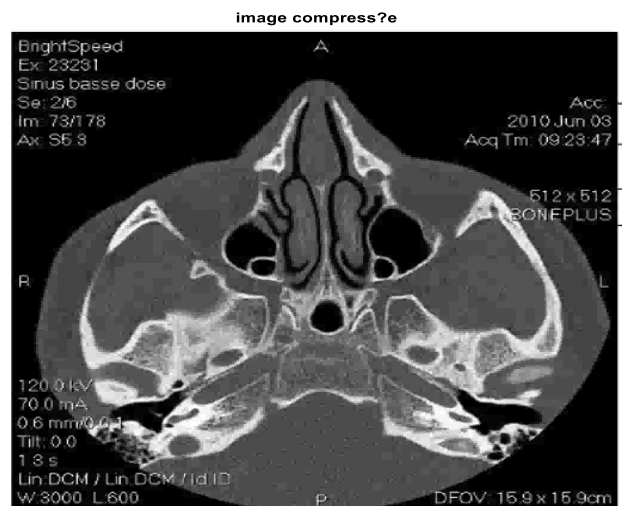
Tableau IV.1 : Variation du PSNR, MSE et du temps en fonction du niveau L

D'après les résultats obtenus, nous remarquons que la valeur du PSNR(32.77db) se stabilise autour du niveau L=6 avec un temps de compression qui diminue légèrement (T=35.93s).

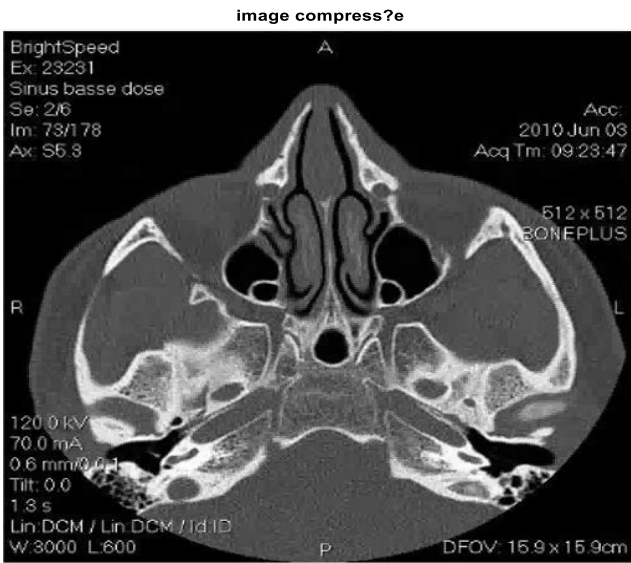
Nous représentons sur les figures suivantes l'image originale ainsi que les images reconstruites



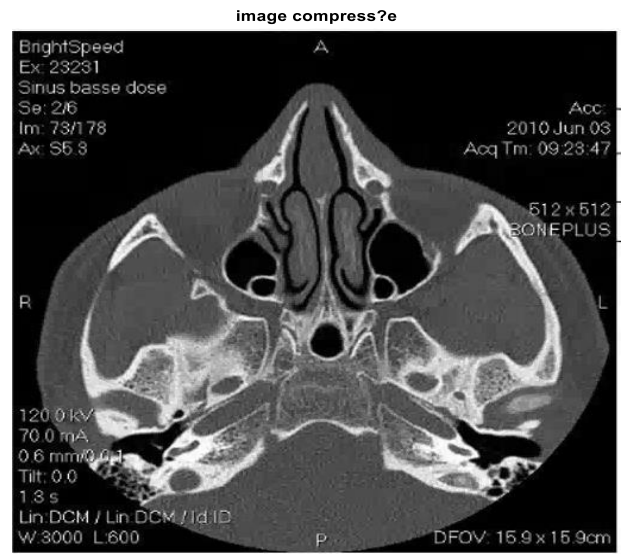
(a) Image originale



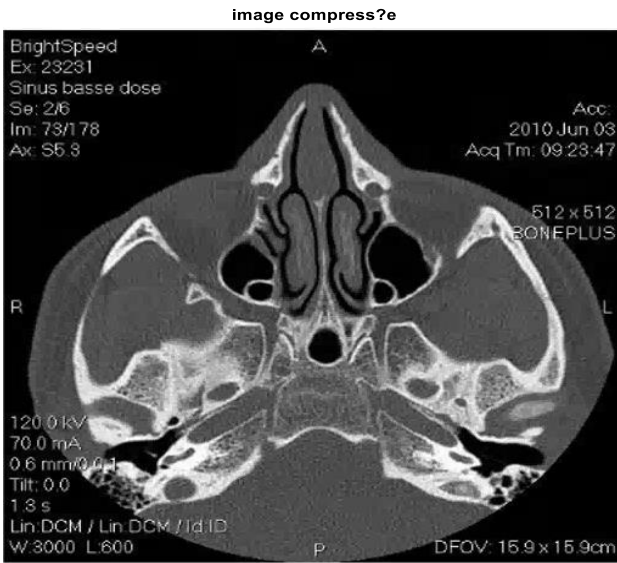
(b) L=3 PSNR=28.80



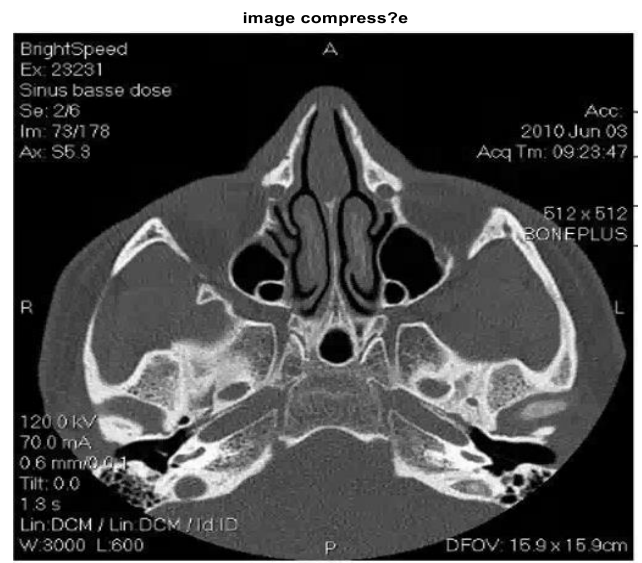
(c) L=4 PSNR=31.86



(b) L=5 PSNR=32.63



(e) L=6 PSNR=32.77



(f) L=7 PSNR=32.81

Figure. IV.5 : Résultats de l'image nez 512*512 reconstruite par SPIHT

b. Application 2 :

Image test : poitrine

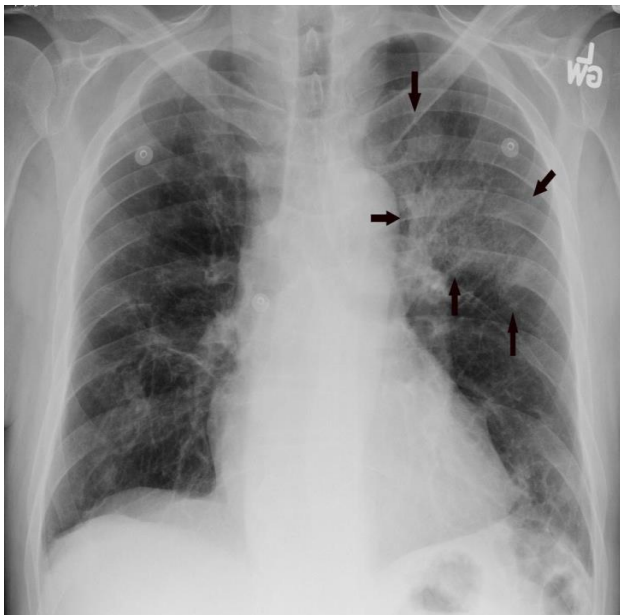
L'ondelette choisi : « db6 »

CR=1bpp

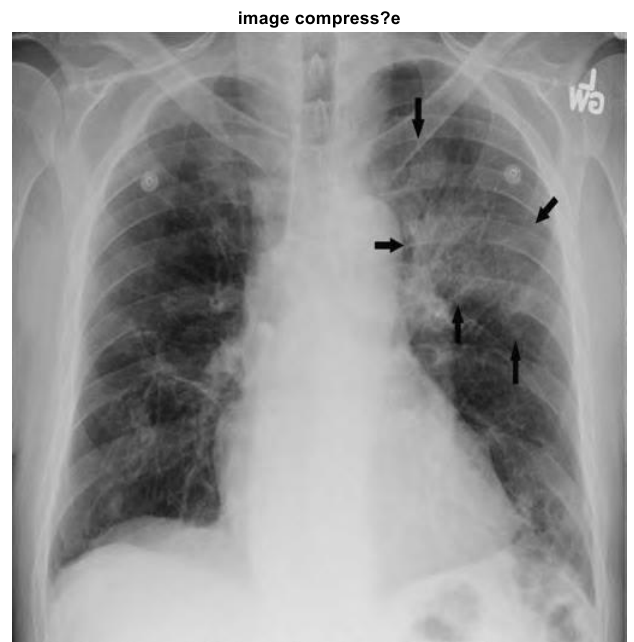
Niveau	L=3	L=4	L=5	L=6	L=7
PSNR	40.88	45.98	46.87	47.03	47.06
MSE	5.31	1.64	1.34	1.29	1.28
T(s)	11.07	24.08	35.39	35.35	35.41

Tableau (IV.2) : Variation du PSNR, MSE et du temps en fonction du niveau L

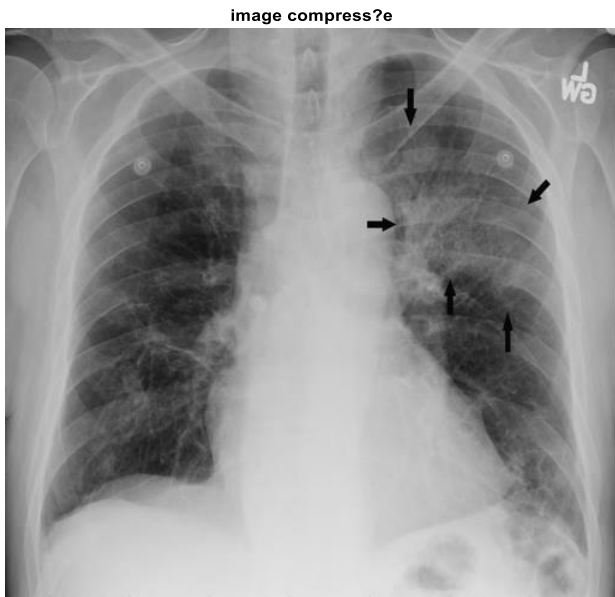
Nous représentons sur les figures suivantes (**Figure IV.6**), l'image originale ainsi que les images reconstruites pour différents niveaux de décompositions



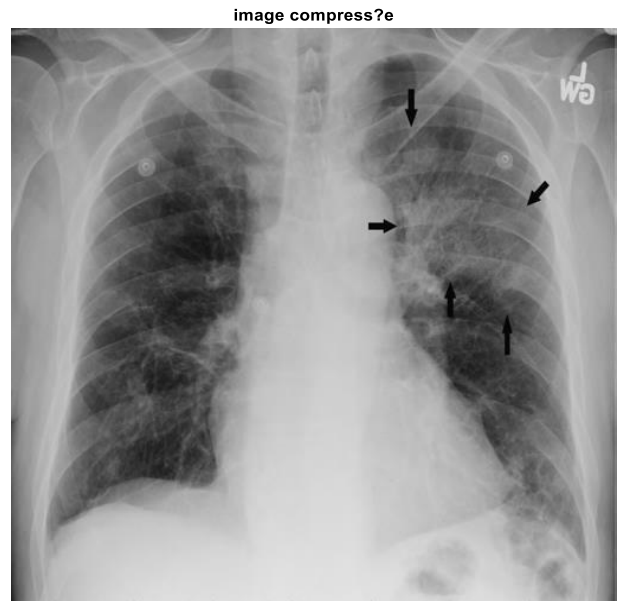
(a) Image originale



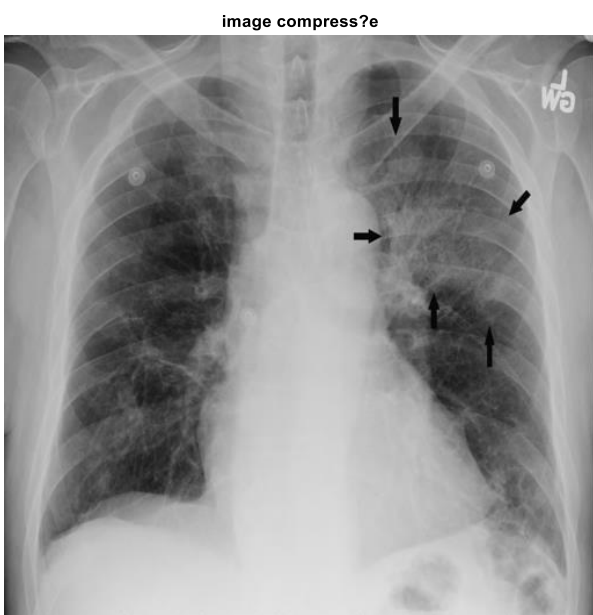
(b) L=3 PSNR=40.88



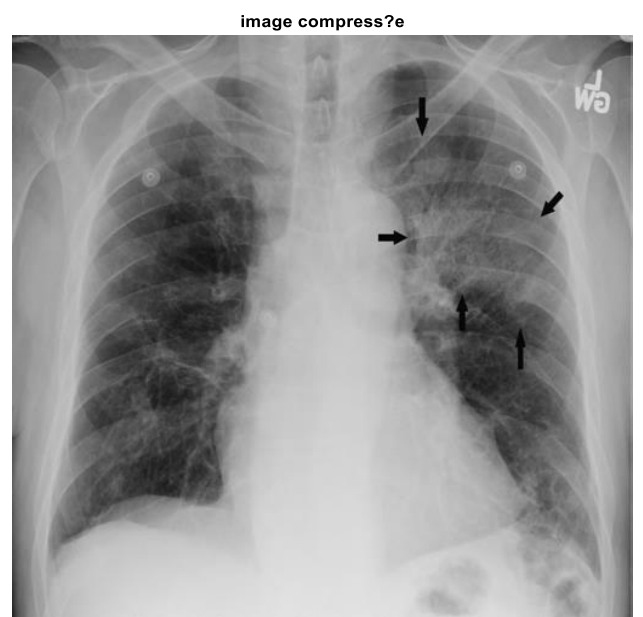
(c) L=4 PSNR=45.98



(d) L=5 PSNR=46.87



(e) L=5 PSNR=36.53



(f) L=7 PSNR=36.53

Figure. IV.6 : Résultats de l'image poitrine 512*512 reconstruite par SPIHT

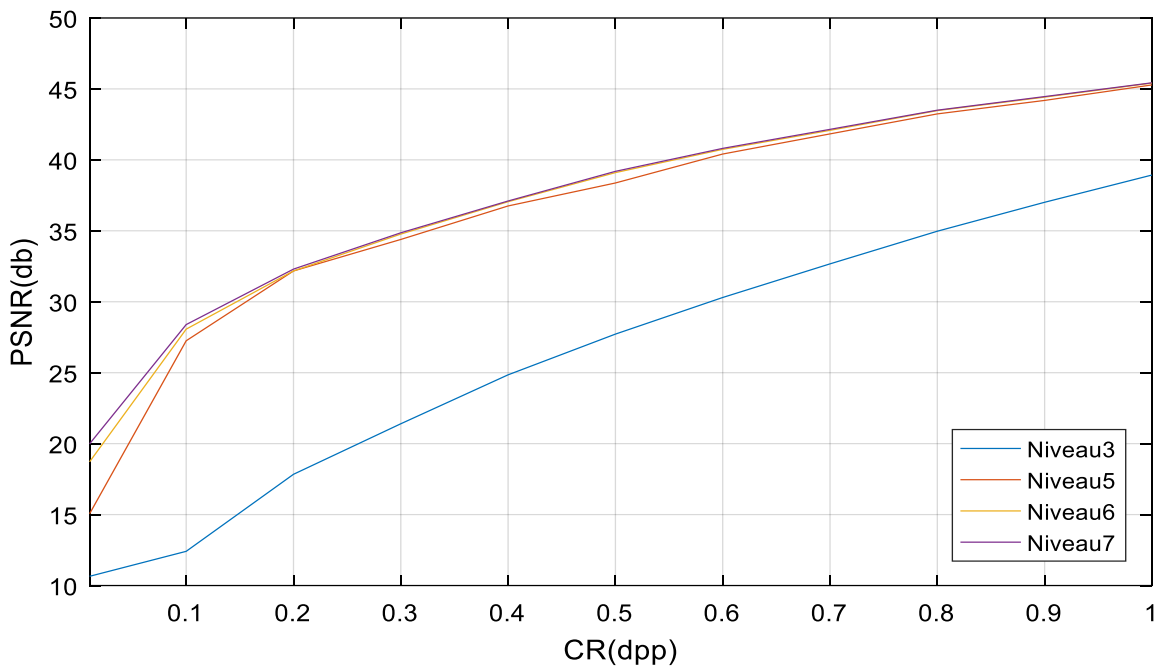
D'après les résultats obtenus, nous optons pour un niveau de décomposition L=6 qui donne un PSNR=47.03db et un temps de compression qui diminue légèrement (T=35.35s). L'image reconstruite pour ce niveau de décomposition est de bonne qualité.

c. Application 3:

Dans cette application, nous avons choisi une nouvelle image test « rachis cervical » et nous avons fait varier le CR de 0.01 à 1bpp ainsi que le niveau de décomposition L. (le Tableau IV.3) ainsi que la (Figure IV. 7) résume les résultats obtenus. L'ondelette utilisée est bior6.8,

	Niveau 3			Niveau 5			Niveau 6			Niveau 7		
CR	PSNR	MSE	T(s)	PSNR	MSE	T(s)	PSNR	MSE	T(s)	PSNR	MSE	T(s)
0.01	10.65	5604.8	1.17	15.06	2026.9	1.19	18.72	873.9	1.30	19.99	652.1	1.49
0.1	12.41	3736.6	1.56	27.25	122.6	1.85	28.07	101.3	2.22	28.39	94.24	2.25
0.2	17.84	1068.9	1.99	32.17	45.88	2.74	32.17	39.44	3.41	32.30	38.31	3.51
0.3	21.40	471.3	2.43	34.39	23.64	4.50	34.77	21.69	5.03	34.86	21.25	5.10
0.4	24.85	212.9	3.40	36.75	13.76	6.75	37.05	12.81	7.02	37.10	12.69	7.36
0.5	27.72	109.94	4.08	38.37	8.71	8.69	39.10	8.00	9.08	39.19	7.83	9.15
0.6	30.30	60.65	6.51	40.41	5.91	11.10	40.74	5.49	12.40	40.81	5.39	12.58
0.7	32.67	35.15	7.06	41.83	4.27	14.32	42.08	4.03	13.90	42.15	3.96	14.05
0.8	34.97	20.68	8.05	43.24	3.09	16.86	43.47	2.92	18.10	43.50	2.91	18.40
0.9	37.01	12.93	11.15	44.19	2.48	23.60	44.42	2.35	23.70	44.46	2.33	24.93
1	38.93	8.32	12.44	45.28	1.93	29.32	45.41	1.87	29.30	45.43	1.86	29.36

Tableau IV.3 : Variation du PSNR, MSE et du temps en fonction de CR.



(Figure IV. 7) : Variation du PSNR en fonction de CR

D'après les résultats obtenus, nous remarquons que la valeur du PSNR (45.41db) se stabilise autour du niveau L=6 avec un temps de compression $T=29.30s$, $MSE=1.87$ et $CR=1bpp$

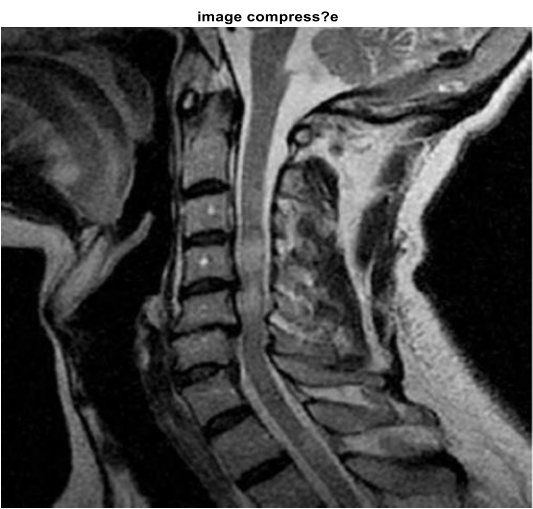
Nous représentons sur les figures suivantes **(Figure IV.8)**, l'image originale ainsi que les images reconstruites pour différents niveaux de décompositions



(a) Image originale



(b) L=3 PSNR=38.93



(c) L=5 PSNR=45.28



(d) L=6 PSNR=45.41



(e) L=7 PSNR=45.43

FigureIV.8: Résultats de l'image « rachis cervical »512*512 reconstruite par SPIHT

III. 2. Influence du type d'ondelette sur la performance de compression

A part le niveau de décomposition, il est également nécessaire d'évaluer l'influence du type d'ondelette sur l'algorithme SPIHT, et de déterminer le bon type d'ondelette, présentant un excellent rapport qualité PSNR(db) / taux de compression (bpp)

a) Application 1 :

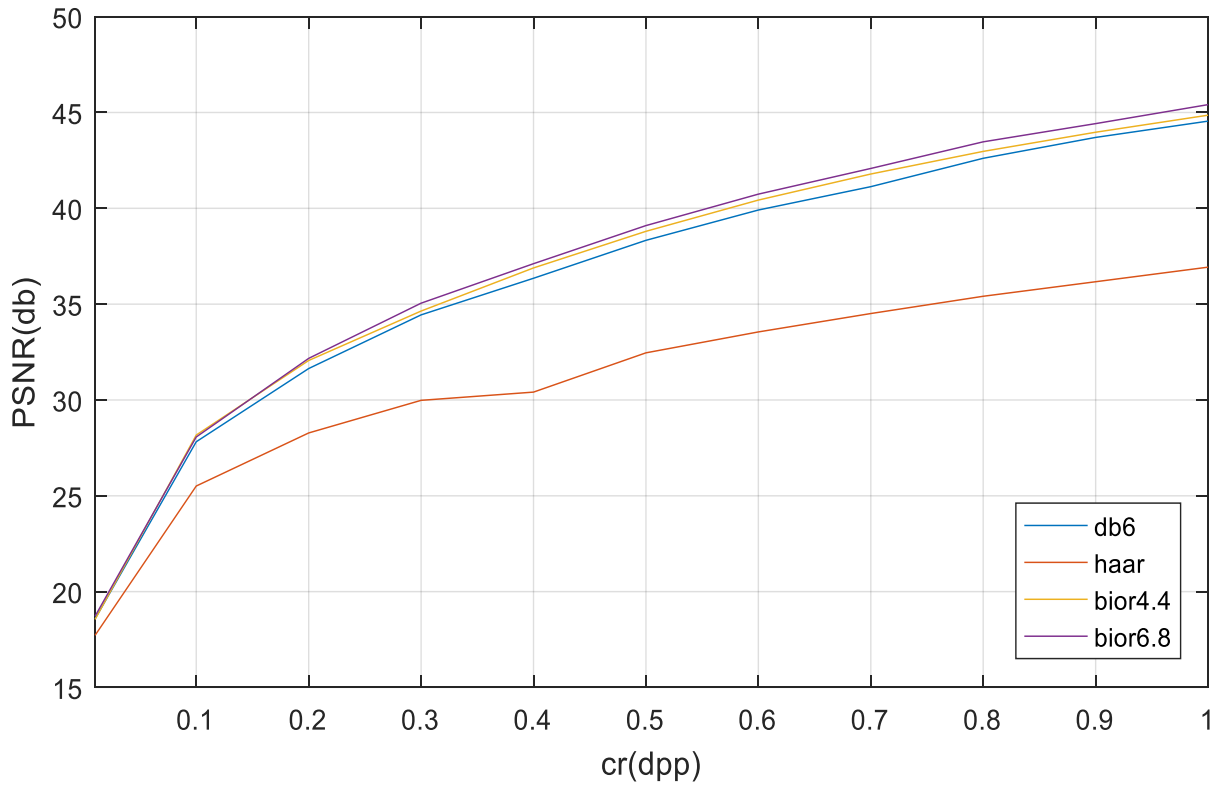
Image test : Rachis cervical

Niveau :6

CR : 0.01 à 1bpp

	db6			Haar			bior4.4			bior6.8		
CR	PSNR	MSE	T(s)	PSNR	MSE	T(s)	PSNR	MSE	T(s)	PSNR	MSE	T(s)
0.01	18.58	902.40	1.20	17.72	1098.56	0.98	18.55	908.14	1.40	18.72	873.91	1.25
0.1	27.82	107.36	1.98	25.51	182.64	1.88	28.17	99.05	2.05	28.07	101.35	1.85
0.2	31.64	44.58	3.30	28.28	96.69	3.16	32.06	40.44	3.31	32.17	39.44	3.26
0.3	34.44	23.40	5.31	29.98	65.29	5.28	34.64	22.35	4.93	35.05	12.81	4.20
0.4	36.35	15.08	7.39	30.41	46.94	7.45	36.89	13.32	6.87	37.11	12.64	6.85
0.5	38.33	9.94	10.02	32.46	36.94	11.59	38.80	8.58	9.23	39.10	8.00	8.48
0.6	39.91	6.63	13.51	33.55	28.73	14.65	40.43	5.89	12.90	40.74	5.49	12.24
0.7	41.13	5.02	16.11	34.51	23.01	17.81	41.79	4.30	14.27	42.08	4.03	13.32
0.8	42.61	3.56	18.45	35.41	18.72	24.31	42.97	3.28	20.15	43.47	2.92	17.75
0.9	43.70	2.78	24.31	36.17	15.69	31.14	43.97	2.61	24.28	44.42	2.35	24.42
1	44.55	2.28	30.11	36.93	13.20	40.20	44.86	2.12	31.71	45.41	1.87	30.34

Tableau IV.4 : Variation du PSNR, MSE et du temps en fonction de CR



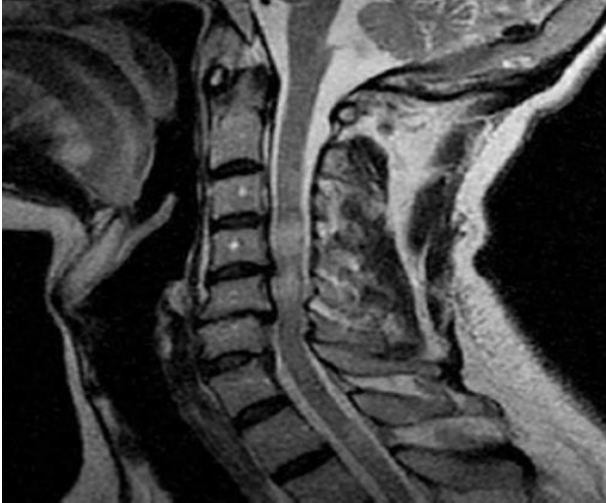
(Figure IV. 9) : Variation du PSNR en fonction de CR

Pour un niveau de décomposition $L=6$, $CR=1\text{bpp}$, les figures suivantes montrent l'image originale et les images reconstruites obtenues pour différentes ondelettes



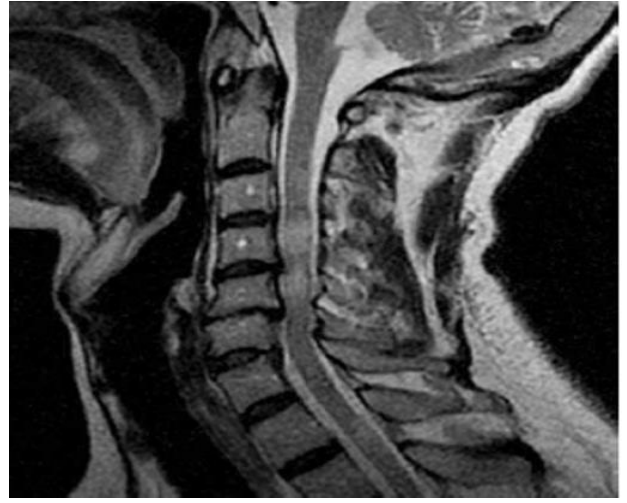
(a) Image originale

image compress?e



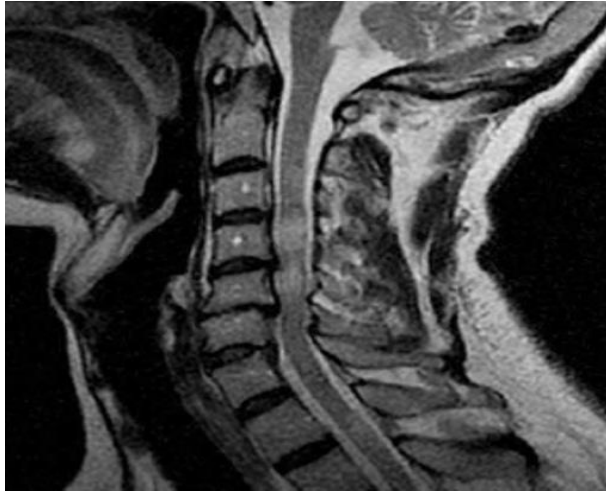
(b) "db6", PSNR=44.45db,cr=1bpp

image originale



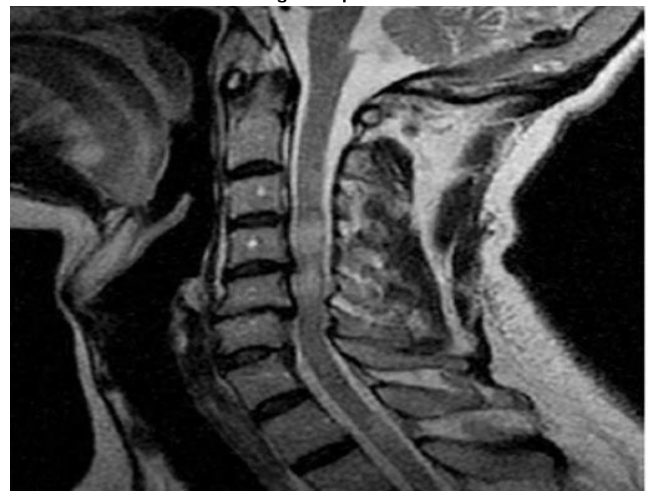
(d) "haar", PSNR=36.93db,cr=1bpp

image compress?e



(e) "bior4.4", PSNR=44.86db,cr=1bpp

image compress?e



(f) "bior6.8", PSNR=45.41db,cr=1bpp

FigureIV.10: Résultats de l'image « rachis cervical »512*512 reconstruite par SPIHT

b) Application 2 :

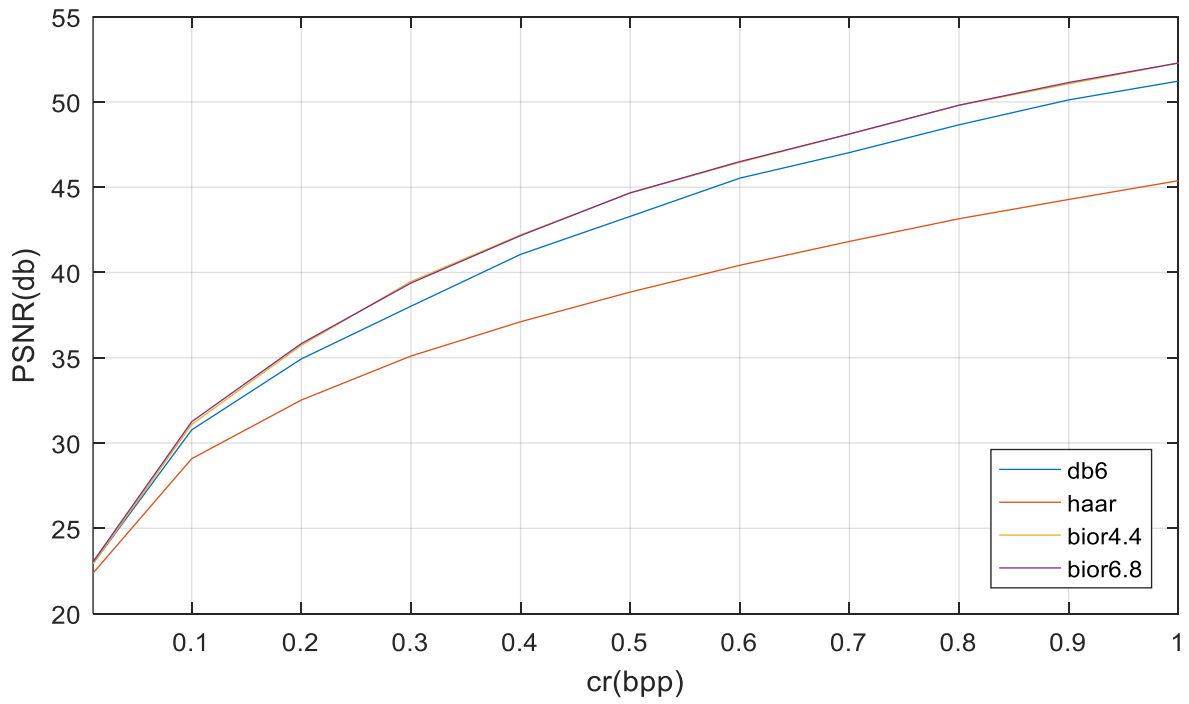
Image test : IRM du cerveau

Niveau :6

CR : 0.01 à 1bpp

	db6			Haar			bior4.4			bior6.8		
CR	PSNR	MSE	T(s)	PSNR	MSE	T(s)	PSNR	MSE	T(s)	PSNR	MSE	T(s)
0.01	22.94	330.47	1.24	22.37	377.04	1.26	22.92	332.24	0.95	23.04	323.26	0.85
0.1	30.77	54.45	1.49	29.08	80.29	1.84	31.09	50.54	1.85	31.25	48.81	1.76
0.2	34.93	20.92	2.51	32.52	36.39	2.89	35.73	17.36	2.75	35.83	16.98	2.69
0.3	38.02	10.25	4.88	35.10	20.08	4.93	39.47	7.34	4.74	39.38	7.49	4.37
0.4	41.06	5.10	6.18	37.11	12.64	6.50	42.20	3.92	6.15	42.16	3.95	6.18
0.5	43.29	3.05	7.79	38.85	8.48	9.01	44.66	2.22	7.32	44.67	2.22	7.32
0.6	45.53	1.82	10.21	40.42	5.90	11.57	46.46	1.47	10.03	46.50	1.46	10.01
0.7	47.03	1.29	13.14	41.82	4.27	14.60	48.11	1.01	12.22	48.12	1.00	12.20
0.8	48.66	0.89	14.46	43.15	3.15	18.43	49.80	0.68	14.96	49.81	0.68	14.69
0.9	50.12	0.63	18.24	44.28	2.43	23.78	51.06	0.51	19.89	51.14	0.50	19.67
1	51.22	0.49	24.65	45.38	1.89	29.85	52.28	0.39	23.57	52.28	0.38	23.43

Tableau IV.5 : Variation du PSNR, MSE et du temps en fonction de CR.



(Figure IV. 11) : Variation du PSNR en fonction de CR

Pour un niveau de décomposition $L=6$, $CR=1\text{bpp}$, les figures suivantes montrent l'image originale et les images reconstruites obtenues pour différentes ondelettes



(a) Image originale



(b) “bior4.4” ,PSNR=44.86db,cr=1bpp



(c) “bior6.8” ,PSNR=45.41 db,cr=1bpp



(d) “bior4.4” ,PSNR=44.86db,cr=1bpp



(e) “bior6.8” ,PSNR=45.41 db,cr=1bpp

FigureIV.12: Résultats de l'image « **IRM du cerveau** » reconstruite pour différentes ondelettes

Application 3 :

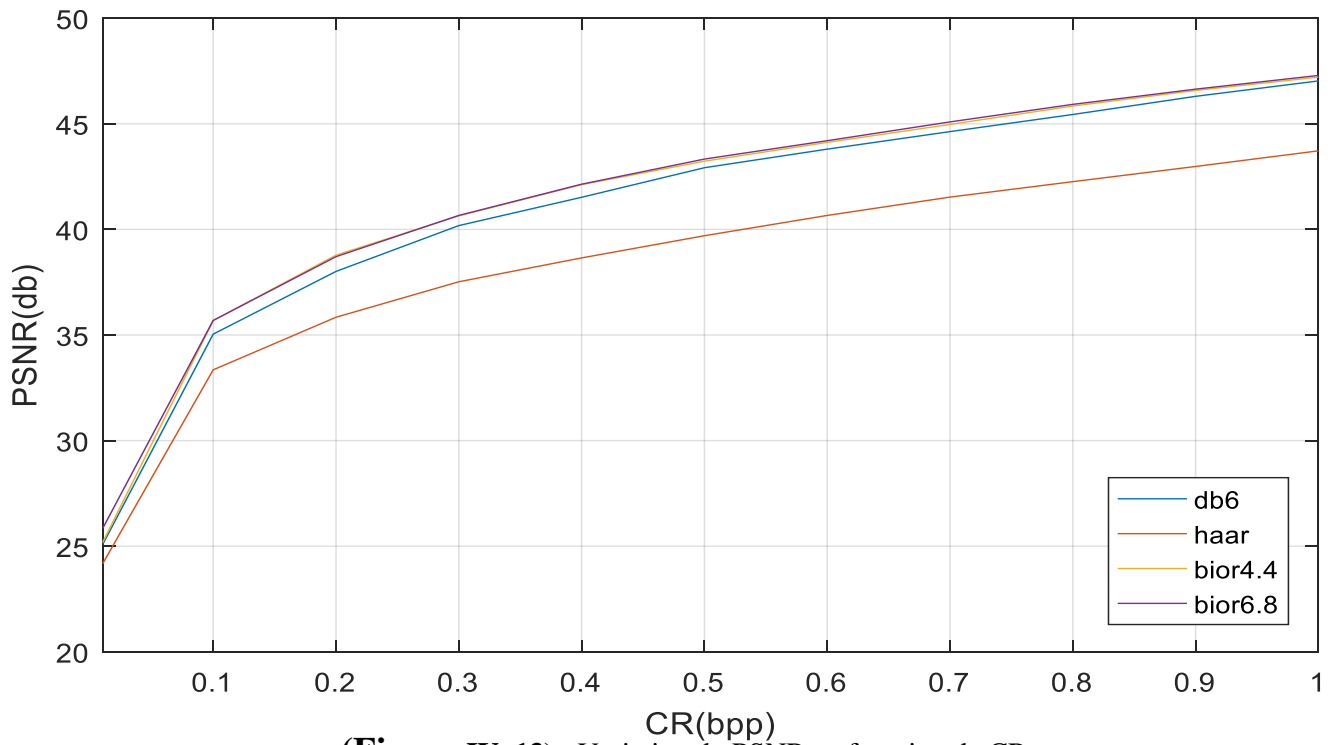
Image test : Poitrine

Niveau :6

CR : 0.01 à 1bpp

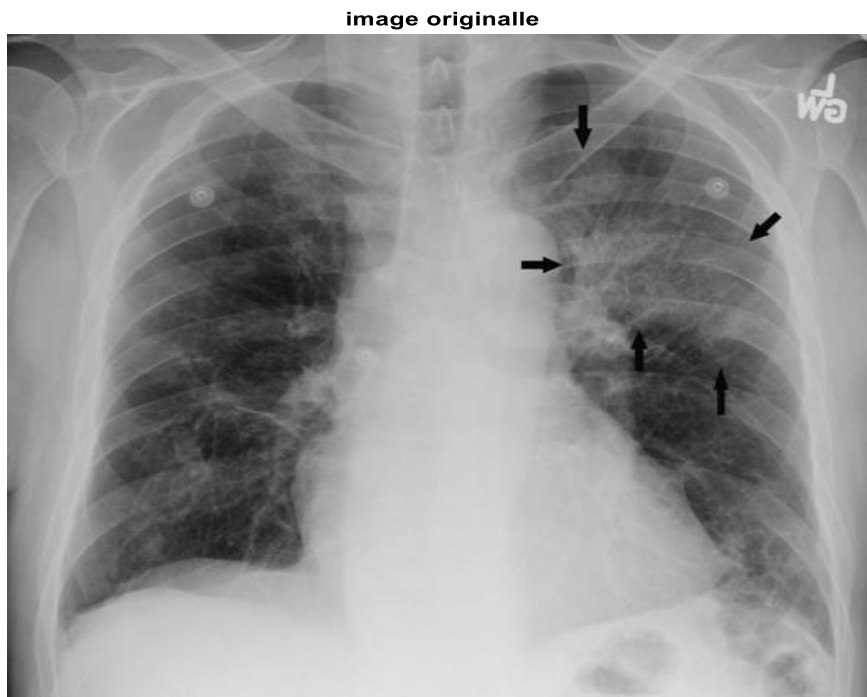
CR	db6			Haar			bior4.4			bior6.8		
	PSNR	MSE	T(s)	PSNR	MSE	T(s)	PSNR	MSE	T(s)	PSNR	MSE	T(s)
0.01	25.08	201.74	0.85	24.17	249.01	0.87	25.18	197.45	0.82	25.84	187.17	0.79
0.1	35.04	20.37	1.41	33.35	30.04	1.53	35.68	17.57	1.48	35.69	17.54	1.47
0.2	38.01	10.28	2.98	35.84	16.93	3.10	38.78	8.61	2.94	38.71	8.75	2.93
0.3	40.18	6.25	5.65	37.52	11.50	5.28	40.64	5.61	5.01	40.66	5.59	4.94
0.4	41.52	4.58	7.35	38.65	8.87	8.20	42.12	3.99	6.97	42.14	3.98	6.87
0.5	42.92	3.32	9.92	39.70	6.96	12.22	43.23	3.09	9.90	43.33	3.02	9.50
0.6	43.80	2.71	14.25	40.66	5.58	14.47	44.12	2.52	14.30	44.20	2.47	14.22
0.7	44.63	2.24	19.87	41.53	4.57	20.44	44.97	2.07	19.97	45.09	2.02	19.95
0.8	45.44	1.86	23.46	42.26	3.87	26.75	45.84	1.70	22.96	45.92	1.66	22.29
0.9	46.30	1.52	28.44	42.98	3.28	34.80	46.58	1.43	30.83	46.64	1.41	29.75
1	47.03	1.29	35.54	43.72	2.76	43.25	47.21	1.24	37.67	47.29	1.21	36.73

Tableau IV.6 : Variation du PSNR, MSE et du temps en fonction de CR

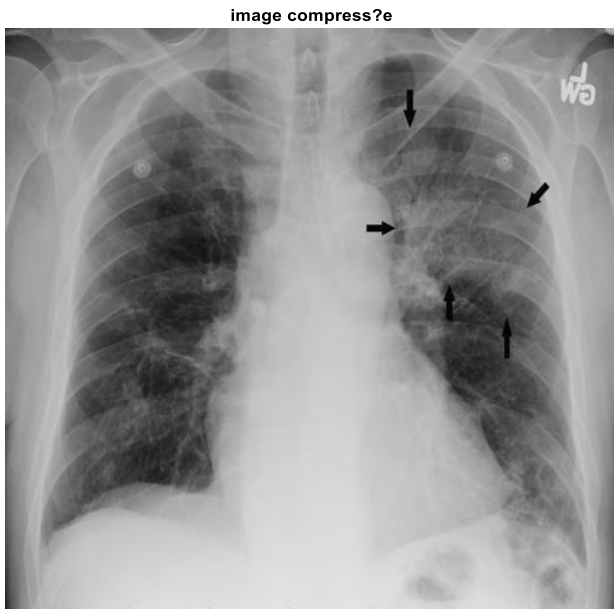


(Figure IV. 13) : Variation du PSNR en fonction de CR

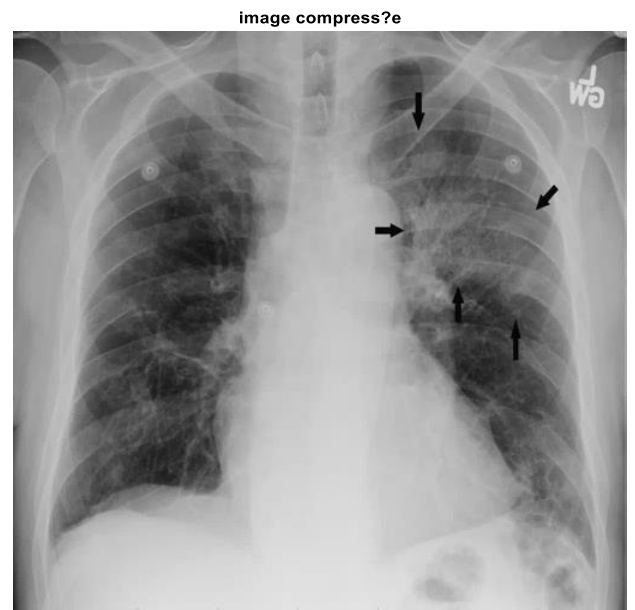
Pour un niveau de décomposition $L=6$, $CR=1\text{bpp}$, les figures suivantes montrent l'image originale et les images reconstruites obtenues pour différentes ondelettes.



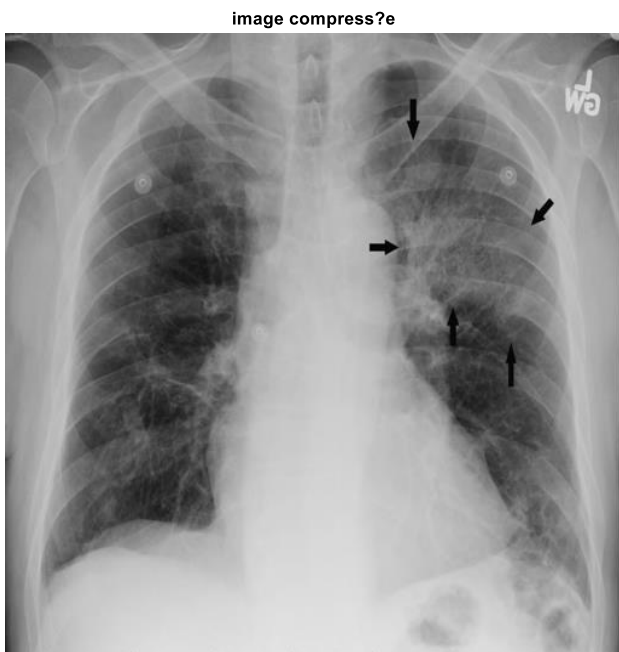
(a) Image originale



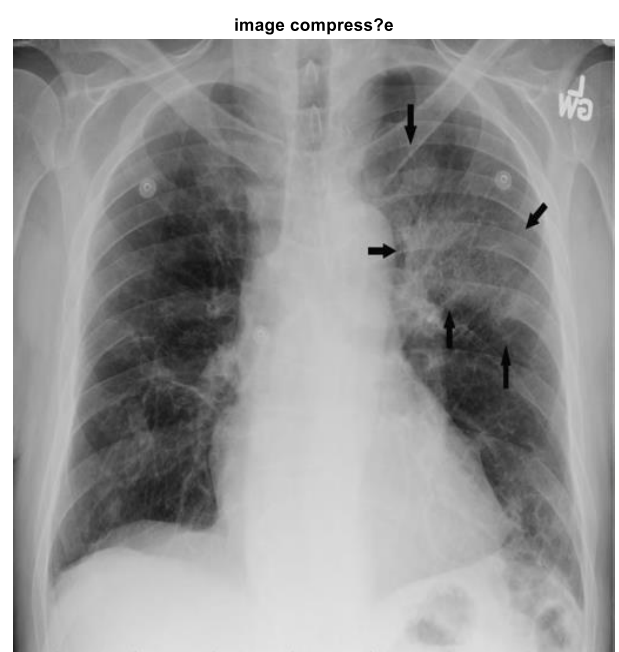
(b) “db6”, PSNR=32.77db, CR= 1bpp



(c) “haar”, PSNR=32.81db, CR=1bpp



(d)“bior4.4”, PSNR=32.77db, CR= 1bpp



(e) “bior6.8”, PSNR=32.81db, CR=1bpp

FigureIV.14: Résultats de l'image « **Poitrine** » reconstruite pour différentes ondelettes

III .3Influence de la taille :

Pour montrer l'influence de la taille de l'image sur les résultats, nous avons réduit la taille de quelques images tests de 512*512 à 256*256 puis à 128*128.

A. Test1

Image: Rachis cervical

L= 6

Ondelette: bior6.8

Le tableau IV.7.donne les résultats obtenus pour les trois tailles utilisées de l'image «Rachis cervical».

	512*512		256*256		128*128	
CR	PSNR	T(s)	PSNR	T(s)	PSNR	T(s)
0.5	39.10	9.51	30.51	1.36	25.92	0.54
1	45.41	30.70	34.83	2.89	30.14	0.62
1.5	49.50	105.61	38.15	4.50	31.56	0.78
1.6	49.58	113.27	38.77	6.36	34.05	0.91
2	52.97	199.19	41.23	7.21	36.22	1.05

Tableau IV.7 : Variation du PSNR du temps en fonction de CR

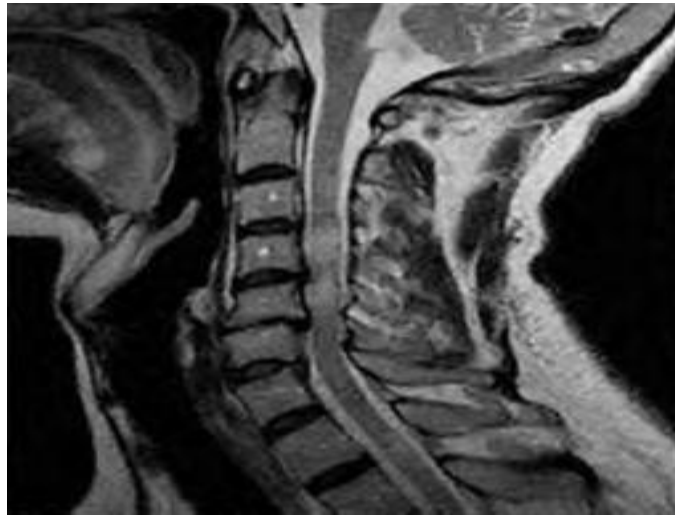
En fixant un CR=1, la figure suivante montre les images reconstruites pour différentes tailles.

image compress?e



(a).la taille :512*512,PSNR=45.41db

image compress?e



(b)la taille :256*256,PSNR=34.83db

image compress?e



(c) la taille :128*128,PSNR=30.14db

FigureIV.15: Résultats de l'image « Rachis cervical » reconstruite pour différentes taille

B. Test : 2

Image : IRM du cerveau

L= 6

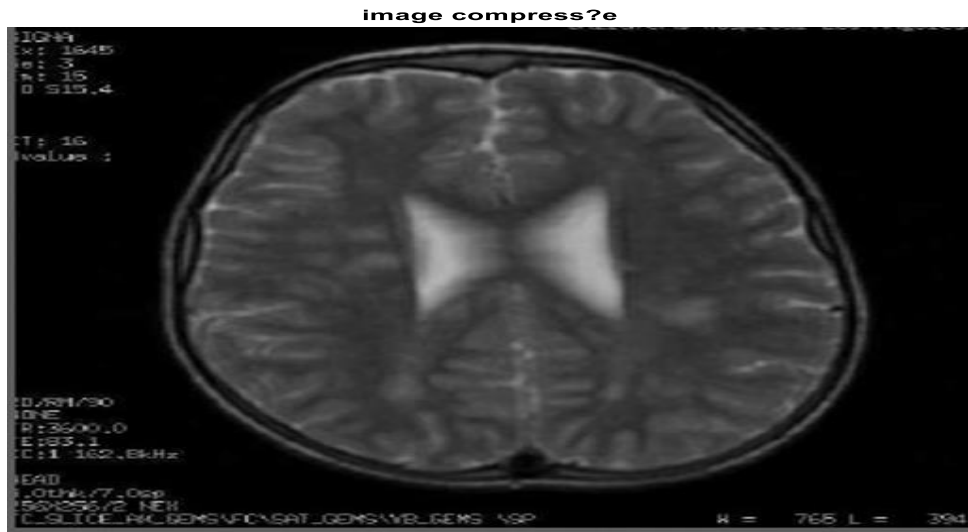
Ondelette : bior6.8

Le tableau IV.8.donne les résultats obtenus pour les trois tailles utilisées de l'image « IRM du cerveau».

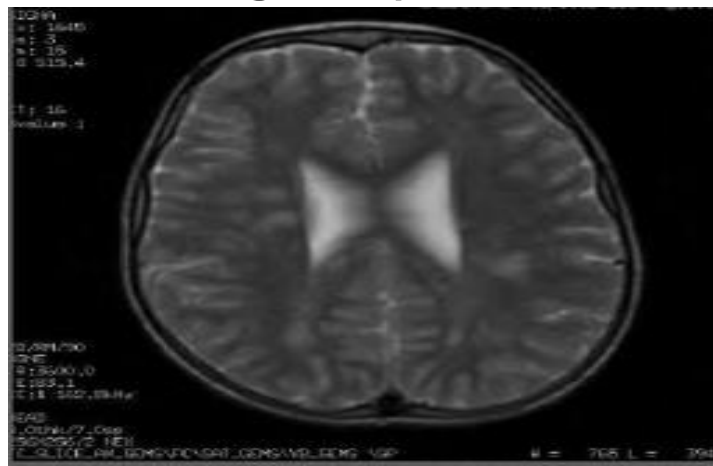
	512*512		256*256		128*128	
CR	PSNR	T(s)	PSNR	T(s)	PSNR	T(s)
0.5	44.67	6.81	35.15	1.15	31.34	0.75
1	52.28	22.40	42.20	2.28	36.56	0.82
1.5	57.03	45.81	47.03	3.76	40.72	0.91
1.6	57.94	46.73	47.81	4.50	41.54	1.02
2	60.74	139.64	50.64	4.19	44.15	1.07

Tableau IV.8 : Variation du PSNR du temps en fonction de CR

En fixant un CR=1, la figure suivante montre les images reconstruites pour différentes tailles

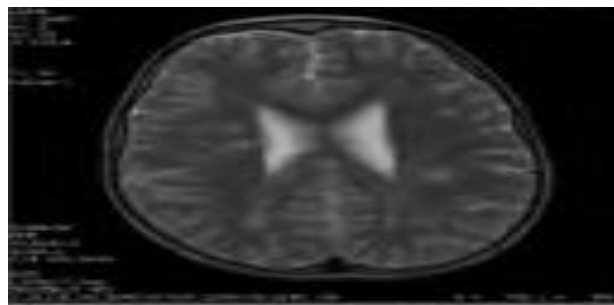


(a).la taille :512*512,PSNR=52.28db
image compress?e



(b).la taille :256*256,PSNR=52.20db

image compress?e



(c).la taille :128*128,PSNR=36.56

FigureIV.16: Résultats de l'image « : IRM du cerveau » reconstruite pour différentes tailles

C. Test :3

Image : scanner du nez

L= 6

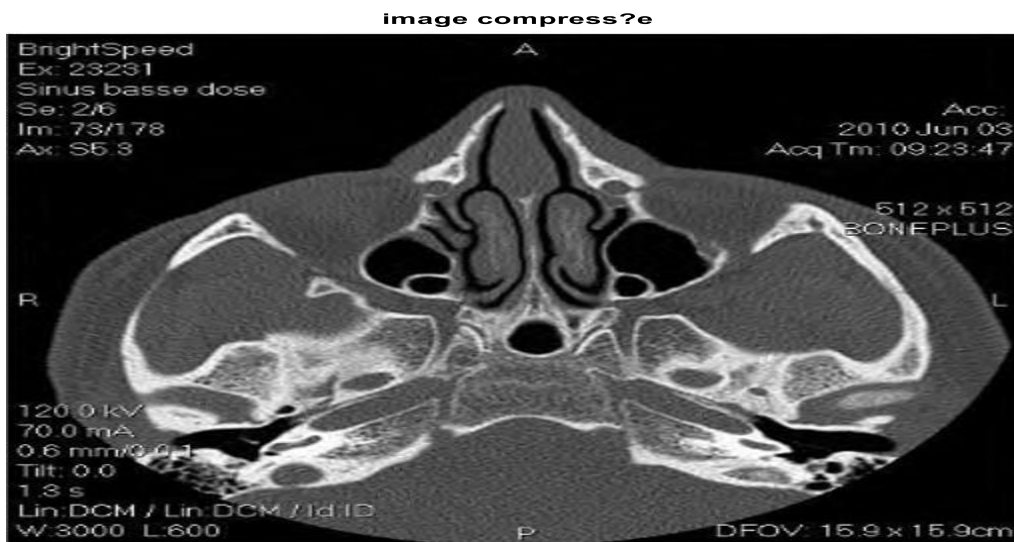
Ondelette : bior6.8

Le tableau IV.9.donne les résultats obtenus pour les trois tailles utilisées de l'image « scanner du nez».

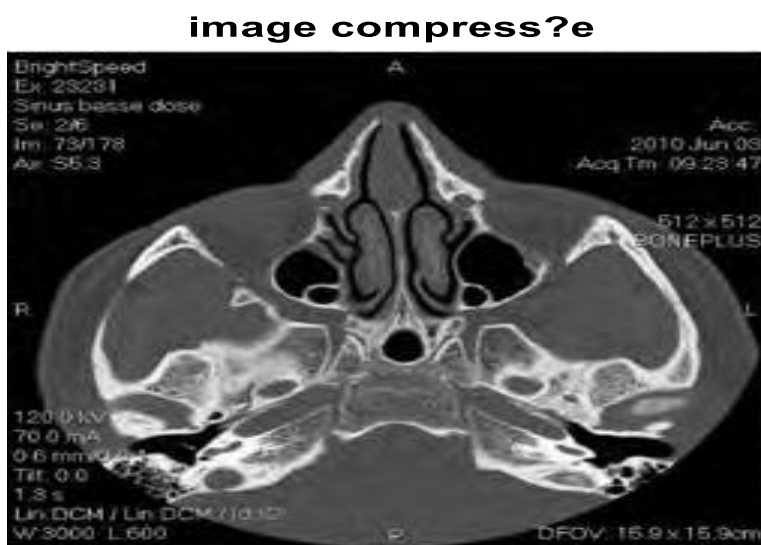
	512*512		256*256		128*128	
CR	PSNR	T(s)	PSNR	T(s)	PSNR	T(s)
0.5	30.72	9.12	35.15	1.26	23.95	0.55
1	36.12	29.57	32.41	2.95	28.15	0.99
1.5	40.39	102.59	34.35	4.45	31.79	1.20
1.6	41.22	115.22	34.94	5.16	32.52	1.22
2	44.52	202.26	37.48	7.19	35.38	1.42

Tableau IV.9: Variation du PSNR du temps en fonction de CR

En fixant un CR=1, la figure suivante montre les images reconstruites pour différentes tailles

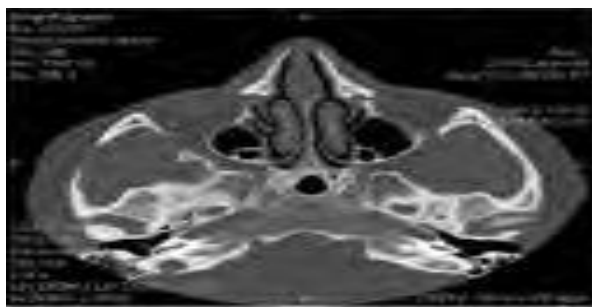


(a).la taille :512*512,PSNR=36.12db



(b).la taille :256*256,PSNR=32.41db

image compress?e



(c).la taille :128*128,PSNR=28.15db

FigureIV.17: Résultats de l'image « scanner du nez » reconstruite pour différentes taille

Interprétation :

Nous pouvons constater d'après les résultats obtenus dans les tableaux (IV.7, IV.8, IV.9) et les figures (IV.16, IV.16, IV.16) qu'effectivement plus la taille de l'image augmente, plus le temps d'exécution de l'algorithme augmente. Nous constatons aussi que quand la taille augmente, nous obtenons de meilleurs résultats en qualité d'image

IV. Conclusion

D'après les résultats obtenus, nous constatons que :

- Le PSNR dépend du filtre choisi ainsi que les niveaux de décomposition et les différents taux de compression (bpp).
- Le PSNR est proportionnel au niveau de décomposition, plus le niveau de décomposition augmente, plus le PSNR augmente et plus est meilleure la qualité de l'image restituée.
- Le temps d'exécution de notre application est acceptable vu la rapidité de l'algorithme.

Conclusion générale

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

Dans notre mémoire nous nous sommes intéressés à une technique de compression basée sur la DWT et opérant sous forme progressive à savoir la technique SPIHT. Nous avons donc mis au point cet algorithme et nous l'avons testé et évalué sur plusieurs images tests. Nous avons introduit deux paramètres d'évaluation (PSNR, MSE) afin de juger la qualité de l'image reconstruite. Les différentes simulations nous ont montré que l'algorithme proposé donne des résultats satisfaisants, encourageants et très concordants avec les travaux d'autres chercheurs dans le même domaine essentiellement pour certains types d'images.

PERSPECTIVES:

Le travail présenté dans ce mémoire peut être étendu dans diverses directions. Plusieurs améliorations peuvent être introduites telles qu'une application aux images 3D et de la vidéo, l'ajout d'un codeur MSPIHT (SPIHT Modifié) au lieu de SPIHT sera plus efficace. Cet algorithme est tributaire de la transformée en ondelette et les limites de cette dernière l'handicapent un peu. Nous proposons ainsi de tester d'autres types d'ondelettes tel que la transformée en paquet d'ondelette, les ondelettes lifting et beaucoup d'autres type d'ondelettes plus performantes comme la CDF9.7.

De même, il serait intéressant d'utiliser cet algorithme basé sur la DWT couplée avec le codeur arithmétique dans la compression d'images médicales 2D et 3D.

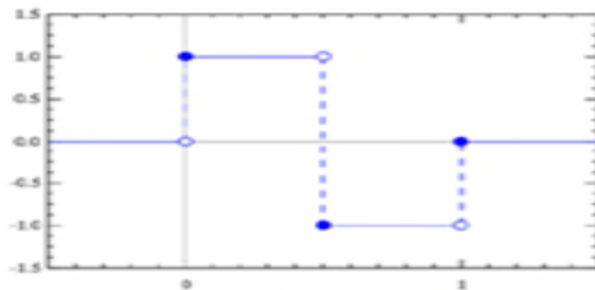
ANNEXE A

Quelques exemples d'ondelettes :

1. Ondelette de Haar:

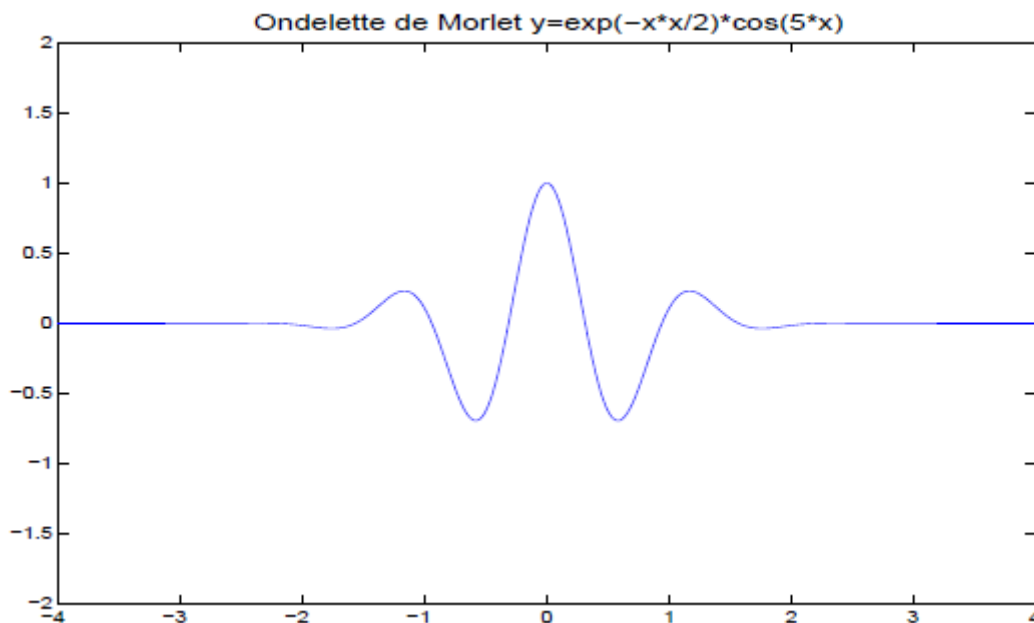
Ondelette assez classique, elle se caractérise par sa fonction d'échelle

$$\psi(t) = \begin{cases} 1 & \text{pour } 0 \leq t < \frac{1}{2}, \\ -1 & \text{pour } \frac{1}{2} \leq t < 1, \\ 0 & \text{sinon} \end{cases}$$



Représentation de l'ondelette de Haar dans le domaine fréquentiel

2. Ondelette de Morlet



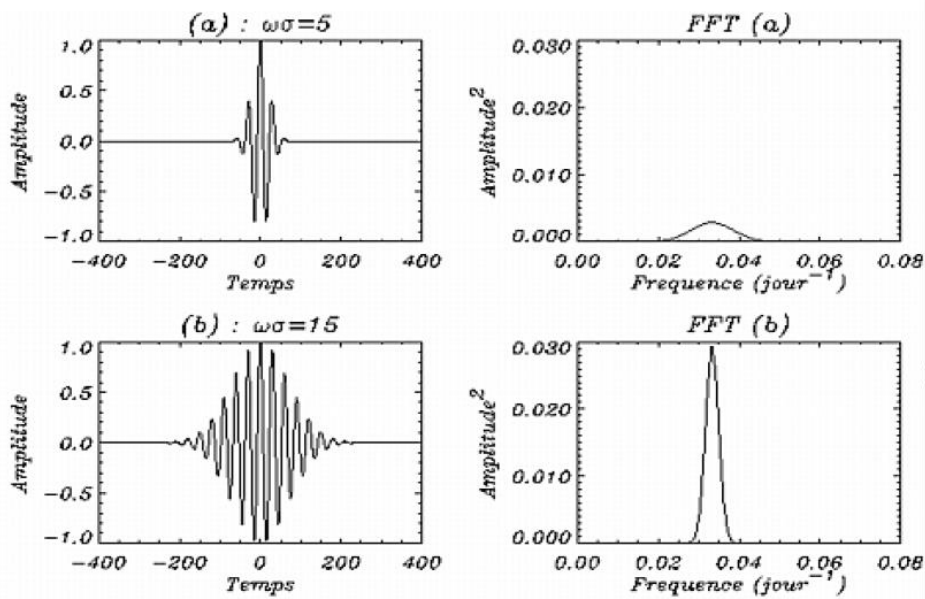


Figure (II.10) : Ondelette Morlet

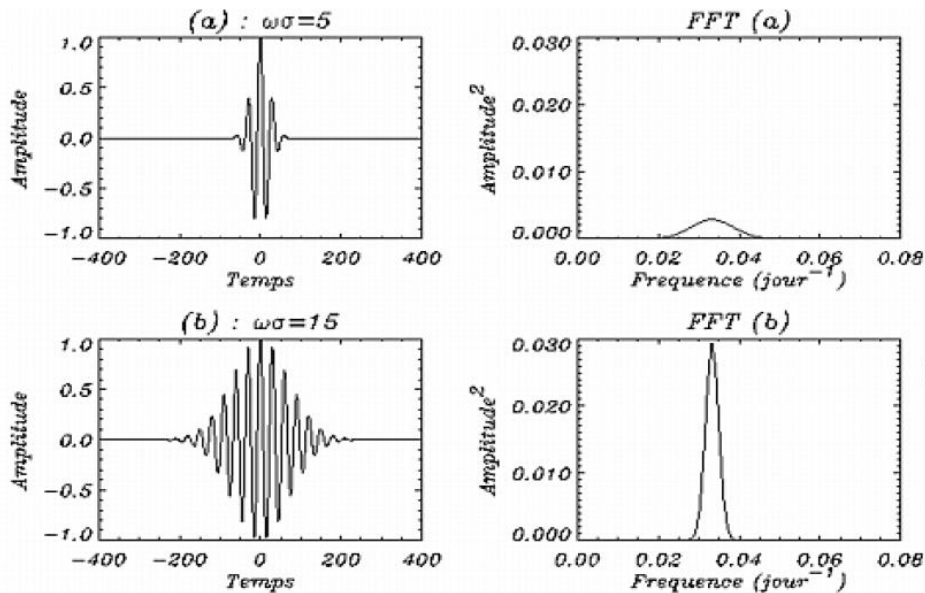
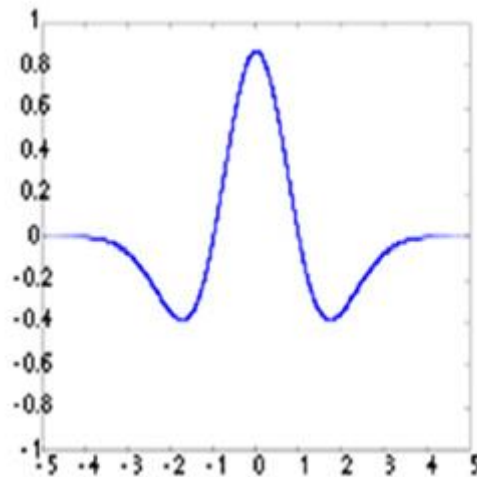


Figure (II.10) : Ondelette Morlet

3. Ondelette chapeau Mexicain:

Elle est définie par :

$$\psi(x) = \frac{2}{\sqrt{3}} \pi^{-1/4} (1 - x^2) e^{-x^2/2}$$

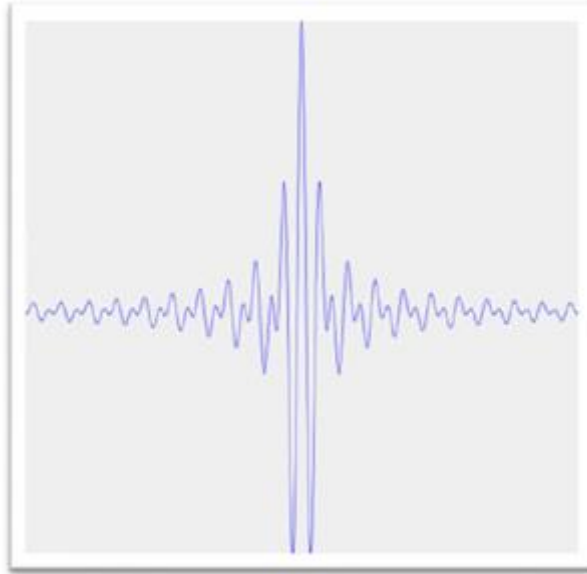


Représentation de l'ondelette chapeau mexicain

4. Ondelette de Shannon

$$\Psi^{(\text{Sha})}(w) = \Pi\left(\frac{w - 3\pi/2}{\pi}\right) + \Pi\left(\frac{w + 3\pi/2}{\pi}\right).$$

$$\Pi(x) := \begin{cases} 1, & \text{if } |x| \leq 1/2, \\ 0 & \text{if otherwise.} \end{cases}$$



Représentation de l'ondelette de Shannon

Bibliographie

- [1]. N. Morceau, "Techniques de compression des signaux", Edition Masson, 1995.
- [2]. M. Kunt, "Traitement numérique des signaux", Edition Dunod, 1981.
- [3]. Gonzalez, Rafael C., and Richard E. Woods (1997), Digital Image Process inf: Addison-Wesley.
- [4]. Guillois, Jean Paul, (1996), Techniques de compression des images: Paris, Hermès.
- [5]. Moreau, Nicolas, (1991), Techniques de compression des signaux : Masson, Paris et CENT ENST
- [6]. A. Mostefaoui, F. Prêteux, V. Lecuire et J. M. Moureaux, "Gestion des données multimédias", traité IC2 Information - Commande - Communication, hermès sciences publications, 2004.
- [7]. D.A. Huffman, "A method for the construction of minimum-redundancy codes", Proc. of the IRE, vol.40, pp. 1098-1101, September 1952.
- [8]. Sayood, K. (1996) Introduction to data compression: Morgan Kaufman Publishers
- [9]. D. Taubman, W.M. Marcellin "JPEG2000 image compression fundamentals, standards and practice", Kluwer Academic Publishers, 2002.
- [10]. R. Hunter et A.H. Robinson, "Runlength", Proceedings of the IEEE, vol.68, pp. 854-867, 1980.
- [11]. Netravali, A., Haskel, B., (1988) Digital pictures Representation and compression. New York : plenum
- [12]. Weeks, Arthur R. (1996), Fundamentals of electronic image processing. New York: IEEE presse.
- [13]. A. Boukaache "Compression d'images hautes définitions en utilisant une transformation hybride" Thèse de doctorat 2012 Guelma Algérie

- [14]. S. G. Mallat, "A theory for multi resolution signal decomposition: The wavelet representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11 (no 7): pp. 674-693, Juillet 1989
- [15]. J. Shapiro, "Embedded image coding using zero trees of wavelet coefficients," IEEE Transactions Signal Processing, vol. 41, pp. 3445-3462, Dec. 1993.
- [16]. 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.
- [17]. V.A. Allen & J. Bellian, "Sub-Band Coding of the Discrete Cosine Transform in ECG Compression," Proc. 15th Ann. Inter. Conf. IEEE Eng. Med. Biol. Soc., pp.790-791, Oct. 1993.
- [18]. A.Said, & W. A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," IEEE Trans. On Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 243-250, 1996.
- [19]. V. Chappelier "Codage progressif d'images par ondelettes orientées" Thèse de Doctorat. Université de Rennes 1, 2005.
- [20]. David A. Clunie, "Lossless Compression of Grayscale Medical Images Effectiveness of Traditional and State of the Art Approaches", SPIE Medical Imaging, San Diego, Feb 2000.
- [21]. Denecker K, Van Overloop J, Lemahieu I, "An experimental comparison of several lossless image coders for medical images", Proc. 1997 IEEE Data Compression Conference.
- [22]. Kivijärvi J, et al, "A comparison of lossless compression methods for medical images", Computerized Medical Imaging and Graphics, 22, pp 323-339, 1998.
- [23]. W.Philips, S. Van Assche, D De Rycke et K Denecker, "State of-the-art for lossless compression of 3D medical images sets", Computerized Medical Imaging and Graphics, 2001.
- [24]. X.Wu and J.-H. Chen, "Context modeling and entropy coding of wavelet coefficients for image compression", in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal, pp. 3097-3100, New York, 1997.
- [25]. X.Wu , "Lossless compression of continuous-tone images via context selection, quantization and modeling", IEEE Transactions on Image Processing, vol. 6, pp. 656-664, 1997.
- [26]. M. Weinberger, G. Seroussi, G.Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS", Technical Report HPL-98-193, HP Computer Systems Laboratory, novembre1998. <http://www.hpl.hp.com/techreports/98>.
- [27]. M.W. Marcellin, M.J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000", in Proc. DCC 2000, Snowbird, UT, pp. 523-541, mars 2000.
- [28]. Bradley J.Erickson, "Irreversible Compression of Medical Images", Society for computer applications in radiology, 2000.
- [29]. P. Raf y, Y. Gaudeau, D. P. Miller, et J-M. Moureaux, "Computer Aided Detection (CAD) of Solid Lung Nodules in Lossy Compressed MDCT Chest Exams", ECR, Vienne, mars 2006.
- [30]. P.C.Tseng,C.Gray,R.M.Olshen,R.A.Moses,L.E.Davidson,H.C.Bergin,C .J. Riskin, "Tree-structured vector quantization of CT chest scans : image quality and diagnostic accuracy",

- [31]. Goldberg MA, Gazelle GS, Boland GW, et al, "Focal hepatic lesions: effect of three dimensional wavelet compression on detection at CT". Radiology 1994, vol. 190, pp. 517-524.
- [32]. Zalis ME, Hahn PF, Arellano RS, et al, "CT colonography with teleradiology effect of lossy wavelet compression on polyp detection - initial observations", Radiology, vol. 220, pp. 387-392, 2001.
- [33]. A.Abu –Hajar and R.Sankar .Integer-to-integer shapae adaptive wavelet transform for region of interest image coding. In Digital Signal Processing Workshop, 2002 and the 2nd Signal processing Education Workshop. Proceeding of 2002 IEEE 10th, pages 94-97, oct, 2002.
- [34]. A.Abu.Hajar and R.Sankar .Region of interest coding using Partial SPIHT .In Acoustics, Speech, and Signal Processing,2004 .Proceedings.(ICASSP'04).IEEE International Conference on,volume3,pages657-60 vol.3,may2004
- [35]. Wen-Jyi Hwang, Ching-Fung Chine,and Kuo-Jung Li. Scalable medical data compression and transmission using wavelet transform for telemedecine application. Information Technology in biomedicine, IEEE transactions on,7(1):54-63,march2003
- [36]. B.Prabhakar and M.Ramasubba Reddy. HVS scheme for dicom image compression: Design and comparative performance evaluation. European Journal of Radiology, 63Issue1):128135, July2007.
- [37]. S.Yea,S.Cho,and W.A.Pearlman.Integrated lossy,near-lossless, and lossless compression of medical volumetric data. In Proc.SPIE5685, editor, Image and Video Communications and Processing 2005, Sumposium on Electronic Imaging,volume 2,pages 151-159,Jan,2005.
- [38].S.Yea, S.Cho, and W.A.Pearlman. A wavelet –based two –stage near-lossless coder. Image Processing, IEEE Transactions on 15(11):3488-3500, Nov 2006
- [39]. Jonathan Taquet, "Techniques avancées pour la compression d'images médicales", Thèse de doctorat, Université de Rennes1, 2011
- [40]. A. CZIHÓ, "Quantification vectorielle et compression d'image. Application à l'imagerie médicale", Thèse de doctorat, Université de Rennes1, Mai 1999.
- [41]. P. Le Callet et D. Barba, "Modèle de perception couleur : application a l'évaluation de la qualité", Traitement du signal, vol. 21, pp. 461-477, 2004
- [42]. W.S. GEISLER, M.S. BANKS, "Visual performance", in Handbook of Optics (M.Bass, ed.), McGraw-Hill, 1995.
- [43]. P. Le Callet, C. Viard-Gaudin et D. Barba, «A convolutional neural network approach for objectivevideoqualityassessment", IEEE Transactionson Neural Networks, September, 2006.
- [43]. A. Przelaskowski, "Vector quality measure of lossy compressed medical images Vector quality measure of lossy compressed medical images", Computers in Biology and Medecine, 2003
- [44]. Ouafi, Abdelkrim. Compression d'images fixes biomédicales par les transformées en ondelettes associées aux algorithmes de quantification vectorielle et de codage entropique. Mémoire de Magister. s.l. : Université Mohamed Khider de Biskra, 2001.
- [45].S. G. Mallat. A Wavelet Tour of Signal Processing. Academic Press, 1997.

[46].BELADGHAM.M, « Construction d'une technique d'aide au diagnostic en Imagerie médicale. Application à la Compression d'images», thèse de doctorat en électronique, université de Tlemcen. Décembre 2012 .

[47].R.W. BUCCIGROSSI, E.P. SIMONCELLI, "Image compression via joint statistical characterization in the wavelet domain", IEEE Trans. Image processing, Vol. 8, pp.1688– 1701, December 1999.