

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE HADJ LAKHDAR BATNA



FACULTE DE TECHNOLOGIE
DEPARTEMENT D'ELECTRONIQUE

THESE DE DOCTORAT EN SCIENCES EN ELECTRONIQUE

Option : Contrôle industriel

Présentée Par

BOUKHENNOUFA Nabil

Thème

ELABORATION D'ALGORITHMES POUR LA COMPRESSION DE DONNEES

Soutenue le : 01/06/2013

Devant le jury composé de :

BENYOUCEF Moussa	Professeur	Univ. de Batna	Président
BENMAHAMMED Khier	Professeur	Univ. de Sétif	Rapporteur
BOUTTOUT Farid	Professeur	Univ. de Bordj Bou Arréridj	Examineur
CHAFAA Kheireddine	M.C. A	Univ. de Batna	Examineur
MEZACHE Amar	M. C. A	Univ. de M'sila	Examineur

Dédicace

A ma très chère mère

A toute ma famille

A tous mes amis et collègues

Remerciements

En premier lieu je tiens à remercier Le Bon Dieu Tout-Puissant de la bonne santé, de la volonté et de la patience qu'il m'a accordé durant toutes mes années d'études.

Mes vifs remerciements vont tout d'abord à mon Directeur de thèse, le Professeur K. BENMAHAMMED de l'Université de Sétif, qui m'a fait l'honneur d'être le rapporteur de cette thèse, ainsi que pour ses conseils, son soutien et sa contribution dans ce travail de recherche.

Que Pr. M. BENYOUCHEF trouve ici l'expression de mes sincères remerciements pour avoir accepté de présider le jury de soutenance.

Que Pr. F. BOUETTOU, Dr. K. CHAFAA et Dr. A. MEZACHE, soient chaleureusement remerciés d'avoir accepté de faire partie du jury de soutenance en qualité d'examineurs pour juger ce travail de recherche.

Mes profonds remerciements vont à mon ami Dr. R. BENZID qui m'a aidé et encouragé le long de réalisation de ce travail scientifique.

Mes remerciements vont aussi à mes amis et à toute personne pour l'aide, le soutien moral ou les encouragements.

ملخص

لضغط المعطيات في يومنا هذا دور كبير في تخزين ونقل المعلومات. لذا اقترحنا في هذه الأطروحة بعض الحلول لتخفيض الحجم الفيزيائي لإشارات تخطيط القلب الكهربائية والصوتية. عملنا يركز على الجوانب اللوغارتمية باستخدام تحويل الموجات المنفصلة. أول تقنية تستند على الكشف عن قمم الإشارات الكهربائية. في الطريقة الثانية، تم استخدام ترميز هرمي تدريجي "SPIHT" إلى جانب التشفير "VKTP" لضغط إشارات تخطيط القلب الكهربائية. أيضا، تم استصدار في هذا العمل طريقة جديدة لضغط إشارات القلب الصوتية سميت "EEZW". مما ساعد في الحصول على امتيازات جيدة الأداء مقارنة بطرق أخرى أكثر تعقيدا.

كلمات مفتاحية : إشارات تخطيط القلب الكهربائية، إشارات تخطيط القلب الصوتية، تحويل الموجات المنفصلة، الترميز الهرمي، نسبة الضغط، النسبة المئوية للتشويه.

Résumé

La compression des données présente aujourd'hui un grand intérêt pour l'archivage et la transmission des informations. On a proposé dans cette thèse quelques solutions pour réduire la taille physique des signaux électrocardiogrammes (ECG) et phonocardiogrammes (PCG). Notre travail porte sur les aspects algorithmiques de compression utilisant les transformées d'ondelettes discrètes. La première technique développée est basée sur la détection des pics R des signaux ECG. Dans la deuxième méthode, un codage hiérarchique progressif appelé SPIHT (Set Partitioning In Hierarchical Trees) combiné avec le codeur VKTP (Vector K Tree Partitioning) a été employé pour la compression des signaux ECG. Aussi, une nouvelle approche proposée nommée EEZW (Enhanced Embedded Zerotree Wavelet) a été élaborée pour la compression des signaux PCG. Elle a permis d'apporter d'excellentes performances débit-distorsion par rapport à autres codeurs plus complexes.

Mots clés : ECG, PCG, Transformées d'ondelettes discrètes, Codage hiérarchique, Rapport de compression, Taux de distorsion.

Abstract

Data compression has become a great interest in the storage and transmission of information. We have proposed in this thesis some solutions to reduce the physical size of ECG (Electrocardiogram) and PCG (Phonocardiogram) signals. Our work focuses on algorithmic aspects of compression using discrete wavelet transforms. The first developed technique is based on the detection of the peaks R of ECG signals. In the second method, a hierarchical progressive coding named SPIHT (Set Partitioning In Hierarchical Trees) combined with VKTP (Vector K Tree Partitioning) encoder was employed for compression of ECG signals. Also, the proposed approach named EEZW (Enhanced Embedded Zerotree Wavelet) was developed for compression of PCG signals. It provides an excellent rate-distortion performance compared with many existing complex encoders.

Keywords: ECG, PCG, Discrete Wavelet Transforms, Hierarchical coding, Compression Ratio, Percent Root Square Difference.

SOMMAIRE

<i>Introduction générale.....</i>	<i>1</i>
1. Introduction.....	1
2. Etat de l'art et problématique.....	2
3. Structure de la thèse.....	3
<i>Chapitre 1: Techniques de compression de données.....</i>	<i>5</i>
1.1. Introduction.....	5
1.2. Classification des algorithmes de compression.....	5
1.2.1. Compression symétrique et asymétrique.....	5
1.2.2. Compression physique et logique.....	6
1.2.3. Compression sans et avec perte.....	6
1.3. Concepts généraux.....	6
1.3.1. Entropie.....	6
1.3.2. Redondance.....	7
1.3.3. Codage.....	7
1.4. Techniques de codage entropique.....	7
1.4.1. Codage de Shannon-Fano.....	7
1.4.2. Codage de Huffman.....	8
1.4.3. Codage arithmétique.....	9
1.5. Algorithmes de compression sans perte.....	11
1.5.1. Technique RLE.....	11
1.5.2. Technique LZW.....	12
1.6. Techniques de compression à base des transformées.....	14
1.6.1. Transformée en cosinus discrète	14
1.6.2. Transformée en ondelettes discrète	15
1.7. Quantification.....	15
1.7.1. Quantification scalaire	15

1.7.1.1. Quantification scalaire uniforme.....	16
1.7.1.2. Quantification scalaire non uniforme.....	16
1.7.2. Quantification vectorielle	17
1.8. Conclusion.....	18

Chapitre 2: Transformées en ondelettes..... 19

2.1. Introduction.....	19
2.2. Historique des ondelettes.....	19
2.3. Définition des ondelettes.....	20
2.4. De la transformée de Fourier à la transformée en ondelette.....	21
2.5. Transformée en ondelettes continue.....	22
2.5.1. Transformée en ondelettes continue 1D.....	23
2.5.2. Transformée en ondelettes continue 2D.....	23
2.5.3. Transformée en ondelettes continue 3D.....	24
2.6. Transformée en ondelettes discrète.....	24
2.6.1. Transformée en ondelettes discrète 1D.....	24
2.6.2. Transformée en ondelettes discrète 2D.....	26
2.6.3. Transformée en ondelettes discrète 3D.....	27
2.7. Comparaison entre CWT et DWT.....	27
2.8. Familles de la transformée en ondelettes.....	27
2.9. Avantages de la transformée en ondelettes.....	29
2.10. Applications des ondelettes.....	30
2.11. Conclusion.....	32

Chapitre 3: Techniques de codage hiérarchique progressif..... 33

3.1. Introduction.....	33
3.2. Relation parent-enfants	34
3.3. Codage EZW.....	35
3.3.1. Caractères du codeur.....	36
3.3.2. Compression des coefficients non significatifs.....	37
3.3.3. Quantification par approximations successives.....	37

3.3.4. Etapes de l'algorithme EZW.....	37
3.3.5. Exemple de déroulement du codeur EZW.....	40
3.4. Codage SPIHT.....	42
3.4.1. Etapes de l'algorithme SPIHT.....	44
3.4.2. Exemple de déroulement de SPIHT.....	45
3.5. Conclusion.....	47

Chapitre 4: Généralités sur les signaux ECG et PCG..... 48

4.1 Introduction.....	48
4.2. Cœur humain.....	48
4.3. Activité électrique du cœur.....	49
4.3.1. Principe de l'électrocardiogramme.....	50
4.3.1.1. Ondes de l'ECG	51
4.3.1.2. Intervalles et segments de l'ECG.....	51
4.3.2. Artefacts de l'enregistrement ECG	52
4.4. Phonocardiogramme.....	54
4.4.1. Sons cardiaques.....	55
4.4.2. Souffles cardiaques.....	56
4.5. Conclusion	57

Chapitre 5: Techniques élaborées pour la compression des signaux ECG et PCG... 58

5.1. Introduction.....	58
5.2. Compression des signaux ECG.....	58
5.2.1. Compression basée sur la détection RR et la DWT 3D (RR-DWT3).....	58
5.2.1.1. Description de la technique.....	58
5.2.1.2. Résultats expérimentaux	62
5.2.2. Compression basée sur la technique SPIHT (SPIHT-VKTP).....	66
5.2.2.1. Description de la technique.....	66
5.2.2.2. Résultats expérimentaux.....	69
5.3. Compression des signaux PCG.....	72
5.3.1. Description de la technique.....	72

5.3.1.1. Passage des dominants.....	74
5.3.1.2. Passage des subordonnés.....	75
5.3.1.3. RLE et codeur arithmétique.....	77
5.3.2. Résultats expérimentaux.....	78
5.4. Comparaison entre les techniques développées.....	83
5.5. Conclusion.....	83
<i>Conclusion générale et perspectives.....</i>	85
<i>Bibliographie.....</i>	87

LISTE DES FIGURES

Figure 1.1 : Exemple de Huffman.....	9
Figure 1.2 Schéma de codage par transformée.....	14
Figure 1.3 : Quantificateur scalaire non uniforme.....	17
Figure 1.4 : Schéma synoptique d'un quantificateur vectoriel.....	17
Figure 2.1 : Ondelette mère.....	20
Figure 2.2 : Découpage en temps fréquence.....	22
Figure 2.3 : Transformée en ondelettes continue 1D.....	25
Figure 2.4 : Principe de l'analyse multirésolution.....	25
Figure 2.5 : Filtres pour la DWT2.....	26
Figure 2.6 : Application de la DWT2 sur trois niveaux de décompositions.....	26
Figure 2.7 : Transformée en ondelettes discrète 3D.....	27
Figure 2.8 : Quelques types de familles d'ondelettes.....	29
Figure 3.1 : Répartition de l'information après une décomposition 3 niveaux.....	34
Figure 3.2 : Relations entre les coefficients d'ondelette dans différentes sous-bandes....	35
Figure 3.3 : Balayage en Z de l'image.....	35
Figure 3.4 : Organigramme de la technique EZW.....	38
Figure 3.5 : Affectation des symboles pour les coefficients.....	39
Figure 3.6 : Exemple de coefficients d'une image 8x8 à 3 niveaux.....	41
Figure 4.1 : Anatomie du cœur humain.....	48
Figure 4.2 : Formes de potentiels d'action dans les différentes parties du cœur et séquence temporelle de la propagation de l'activité électrique cardiaque ; analogie avec les ondes de l'ECG.....	49
Figure 4.3 : Enregistrement ECG normal : différents ondes, intervalles et segments.....	50
Figure 4.4 : Artefacts du tracé ECG.....	53
Figure 4.5 : Evénements du cycle cardiaque.....	55
Figure 5.1 : Schéma synoptique de la technique proposée.....	58
Figure 5.2 : Algorithme de détection des intervalles RR.....	59
Figure 5.3 : Détection des pics R.....	60

Figure 5.4 : Exemple de principe de l’algorithme RLE optimisé.....	62
Figure 5.5 : Enregistrements 100 et 117 au format 3D.....	64
Figure 5.6 : Compression/Décompression du signal ECG 100.dat.....	65
Figure 5.7 : DWT du signal ECG jusqu’à 3 niveaux de décompositions.....	66
Figure 5.8 : Relation Parent-Enfant des coefficients des sous-bandes.....	67
Figure 5.9 : Exemple de partitionnement dune chaine de bits utilisant l’algorithme VKTP.....	68
Figure 5.10 : Illustration de l’algorithme VKTP, avec un arbre limitée à trois niveaux...	68
Figure 5.11 : Compression/Décompression du signal ECG 117 pour CR=6.....	70
Figure 5.12 : Compression/Décompression du signal ECG 117 pour CR=7.....	71
Figure 5.13 : Schéma bloc du codeur proposé.....	73
Figure 5.14 : Exemple de décomposition jusqu'au niveau 3 pour 16 cellules.....	74
Figure 5.15 : Passage des subordonnées pour toutes les itérations du vecteur de test.....	76
Figure 5.16 : Codeur de symboles.....	77
Figure 5.17 : Compression du signal PCG “Normal” pour deux seuils.....	80
Figure 5.18 : Comparaison des différentes méthodes de compression appliquées au signal PCG “Normal”.....	81
Figure 5.19 : Comparaison des différentes méthodes de compression appliquées au signal PCG “4hs”.....	81

LISTE DES TABLEAUX

Tableau 1.1 : Exemple de Codage de Shannon-Fano.....	8
Tableau 1.2 : Association des probabilités et de l'intervalle pour chaque symbole.....	11
Tableau 1.3 : Exemple de codage Arithmétique.....	11
Tableau 3.1 : Seuillage des coefficients par rapport au seuil $T=32$	41
Tableau 3.2 : Seuillage des coefficients par rapport au seuil $T=32$	46
Tableau 5.1 : Résultats de simulation pour la détection des pics R.....	63
Tableau 5.2 : Résultats de comparaisons entre trois codeurs.....	64
Tableau 5.3 : Résultats de compression du signal 117 pour quatre techniques.....	66
Tableau 5.4 : Symboles et codes utilisés pour coder chaque niveau de l'arbre de partitionnement de la séquence binaire de la figure 5.9.....	69
Tableau 5.5 : Résultats obtenus pour l'ensemble des signaux de tests.....	72
Tableau 5.6 : Résultats de compression du signal ECG 117.....	72
Tableau 5.7 : Fréquence d'apparition, probabilité et entropie des symboles utilisés par les algorithmes EZW, MEZW et EEZW appliqués au signal "Normal" pour un seuil $Th = 500$	79
Tableau 5.8 : Résultats expérimentaux pour quelques signaux PCG.....	82
Tableau 5.9 : Résultats obtenus pour les signaux de tests 103, 117 et 119.....	83

LISTE DES ACRONYMES

CR	Compression Ratio
CWT	Continuous Wavelet Transform
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
ECG	ElectroCardioGramme
EEZW	Enhanced EZW
EZW	Embedded Zerotree Wavelets
FT	Fourier Transform
JPEG	Joint Photographic Expert Group
LBS	Length of Bit String
LZW	Lempel Ziv Welch
MP3	Motion Picture experts group audio layer 3
MSE	Mean Square Error
PCG	PhonoCardioGramme
PRD	Percentage of Root mean square Difference
RLE	Run Length Encoding
STFT	Short Time Fourier Transform
SPIHT	Set Partitioning In Hierarchical Tree
VKTP	Vector K-Tree Partitioning

INTRODUCTION GENERALE

Contenu

1. Introduction.....	1
2. Etat de l'art et problématique.....	2
3. Structure de la thèse.....	3

INTRODUCTION GENERALE

1. Introduction

Le monde de l'information et de la communication a apporté une nouvelle dimension dans le domaine médical. Il est actuellement possible d'enregistrer un signal ECG (Electrocardiogramme) ou un signal PCG (Phonocardiogramme) avec un portable ultraléger et même de transformer un micro-ordinateur personnel en électrocardiographe ou phonocardiographe avec, dans les deux situations, possibilité d'une aide au diagnostic par interprétation automatisée.

Le stockage et la transmission des signaux biomédicaux sont devenus actuellement des sujets de grande importance, notamment en raison du développement fulgurant qu'a connu la télémédecine ces dernières années. Dans la plupart des systèmes d'acquisition d'ECG, le signal est enregistré à des fréquences allant de 250 Hz jusqu'à 500 Hz pour l'ECG classique et à une fréquence qui peut aller jusqu'à 1 KHz pour l'ECG à haute résolution. Les signaux PCG sont échantillonnés à des fréquences plus élevées que les signaux ECG, de l'ordre de 11 KHz à 40 KHz. Dans le cas de l'ECG haute résolution, un enregistrement par jour à une résolution de 11 bits/échantillon nécessite en moyenne presque 1 Giga bits d'espace mémoire. Ces chiffres dépassent largement les capacités des systèmes conventionnels de stockage et de transmission. La compression de l'ECG ou du PCG est donc désirable sinon nécessaire pour l'archivage ou la transmission en vue d'une analyse ultérieure.

Un système de compression est constitué d'un codeur permettant d'extraire l'information, pour la décrire par une taille réduite et d'un décodeur permettant de reconstituer l'information parfaitement ou approximativement à partir de l'information codée. Les différents algorithmes de compression sont choisis en fonction de : leur taux de compression (rapport de la taille du fichier compressé sur la taille du fichier initial), la qualité de compression (pourcentage de pertes), la vitesse de compression et de décompression.

En effet, la compression demeure un outil très utile pour l'enregistrement et la transmission mobile des signaux ECG et PCG permettant de fournir une foule de renseignements précieux concernant l'état structurel et fonctionnel de l'appareil cardio-circulatoire [Che08], [Kta09].

2. Etat de l'art et problématique

La compression des signaux médicaux a fait particulièrement l'objet de nombreux travaux de recherche depuis plusieurs décennies. Les premiers algorithmes qui parurent dans les années 60 visaient à réduire le coût de la transmission des signaux médicaux à travers les lignes téléphoniques. Les cardiologues étaient peu à cette époque et il fallait donner la possibilité aux généralistes d'enregistrer les signaux des patients, les transmettre aux spécialistes et recevoir en retour les prescriptions des spécialistes. La télémédecine s'installe et gagne de plus en plus de terrain avec l'arrivée des nouvelles techniques de communication : internet et téléphonie mobile. Les années 90 ont vu l'arrivée du système holter équipé de mémoires de masse de très fortes capacités permettant l'enregistrement ambulatoire de L'ECG sur 24 heures. Rollins *et al* ont montré la possibilité de concevoir un enregistreur portable qui permet d'enregistrer les signaux ECG pendant quelques semaines [Rol00]. R.S.H. Istepanian et A. A. Petrosian ont aussi montré la compression du signal ECG en télécardiographie portable à travers le système GSM [Ist00]. Ce qui rend la compression des signaux ECG incontournable. Dans cette optique Kumar *et al* ont mené des études intenses en reconsidérant dans un premier temps, les méthodes temporelles de compression de l'ECG (AZTEC [Cox68], SAPA [Ish83]) et par la suite des modifications substantielles afin d'obtenir de nouveaux algorithmes adaptés aux spécificités de la télémédecine [Kum06], [Tch07a].

L'objectif d'analyser les signaux ECG et PCG est de représenter et remplacer les raisonnements et les expériences des spécialistes par des systèmes experts. Ce qui a poussé les chercheurs à développer des centaines de techniques à savoir : [Ala07], [Deb05], [Deb07], [Deb08], [Kha09], [Kun82], Aussi dans la compression plusieurs travaux scientifiques récents ont été publiés, dont on peut citer [Ben03], [Ben08], [Che08], [Che98], [Hil97], [Mar08], [Tch07b], ...

Les techniques de compression des signaux ECG et PCG sont habituellement classées en trois grandes catégories à savoir, les méthodes directes, les méthodes utilisant des transformées et enfin les méthodes fondées sur le principe d'extraction des paramètres

caractéristiques. Dans les méthodes directes, comme leur nom l'indique, les échantillons du signal sont directement codés sans aucune transformation particulière. La deuxième catégorie exige une transformation du signal dans un domaine autre que temporel. Cela peut être, une transformation du type transformée en ondelettes discrète. Enfin, la troisième catégorie utilise le principe de la modélisation paramétrique des signaux ECG et PCG. Ces paramètres sont évidemment réutilisés dans la phase de reconstruction [Kta09].

Nous proposons dans cette thèse une solution pour augmenter les performances de la compression des signaux ECG et PCG. Notre travail porte sur les aspects algorithmiques de compression appliqués aux signaux médicaux (ECG et PCG). Les résultats seront comparés et validés avec ceux existants dans la littérature.

Nous allons utiliser les transformées en ondelettes qui s'avèrent très bien adaptées à localiser l'énergie du signal dans un nombre réduit de coefficients. Dans la littérature, plusieurs algorithmes de compression des signaux utilisant les ondelettes ont été proposées, on peut citer [Benz07], [Che98], [Lee99], [Mia02], [Poo05], [Tai05], [Toh07] pour les signaux ECG et [Ele07], [Mar04], [Wic92] pour les signaux PCG.

Dans cette thèse trois techniques ont été proposées et élaborées. Les deux premières méthodes ont été développées et appliquées pour la compression des signaux ECG. La première est basée sur le principe de détection des pics R et les transformées en ondelettes discrètes 3D. La deuxième technique est basée sur SPIHT (Set Partitioning In Hierarchical Tree) [Lu00] combinée avec l'approche VKTP (Vector K-Tree partitioning) [Mar08]. Concernant les signaux PCG, une technique de compression des signaux à encapsulation d'ondelettes EZW (Embedded Zerotree Wavelet) [Sha93] améliorée (EEZW : Enhanced EZW) a été développée et appliquée. Les codeurs EZW et SPIHT permettent une représentation progressive des signaux tout en apportant d'excellentes performances débit-distorsion par rapport à des codeurs bien plus complexes.

3. Structure de la thèse

Afin d'aborder tous ces aspects, la thèse est divisée en cinq chapitres. Elle est organisée comme suit :

Dans le premier chapitre, on va aborder le domaine de la compression en introduisant des généralités et des définitions. Puis, on va décrire brièvement les différentes techniques de compression.

Le deuxième chapitre présente les transformées en ondelettes en citant quelques types d'ondelettes mères à savoir (Daubechies, Haar, Bior,...). Aussi, l'analyse multirésolution sera décrite et appliquée aux signaux unidimensionnels, bidimensionnels et tridimensionnels.

Le troisième chapitre donne une description des codeurs hiérarchiques progressifs EZW et SPIHT appliqués initialement aux images.

Des généralités sur le système cardiovasculaire et sur les signaux ECG et PCG seront présentés dans le chapitre quatre. Ces signaux procurent des informations essentielles pour le diagnostic, le traitement de diverses anomalies cardiaques et la compression utilisant l'extraction des paramètres.

Les techniques proposées et développées pour la compression des signaux ECG et PCG seront détaillées dans le chapitre cinq. Aussi, dans ce dernier, les résultats expérimentaux obtenus seront comparés qualitativement et quantitativement avec quelques méthodes abordées dans la littérature.

Finalement on termine cette thèse par une conclusion générale résumant les travaux effectués et les objectifs atteints tout en proposant les perspectives de ce domaine de recherche.

Chapitre 1

TECHNIQUES DE COMPRESSION DE DONNEES

Contenu

I.1. Introduction.....	5
1.2. Classification des algorithmes de compression.....	5
1.2.1. Compression symétrique et asymétrique.....	5
1.2.2. Compression physique et logique.....	6
1.2.3. Compression sans et avec perte.....	6
1.3. Concepts généraux.....	6
1.3.1. Entropie.....	6
1.3.2. Redondance.....	7
1.3.3. Codage.....	7
1.4. Techniques de codage entropique.....	7
1.4.1. Codage de Shannon-Fano.....	7
1.4.2. Codage de Huffman.....	8
1.4.3. Codage arithmétique.....	9
1.5. Algorithmes de compression sans perte.....	11
1.5.1. Technique RLE.....	11
1.5.2. Technique LZW.....	12
1.6. Techniques de compression à base des transformées.....	14
1.6.1. Transformée en cosinus discrète	14
1.6.2. Transformée en ondelettes discrète	15
1.7. Quantification.....	15
1.7.1. Quantification scalaire	15
1.7.1.1. Quantification scalaire uniforme.....	16
1.1.1.2. Quantification scalaire non uniforme.....	16
1.7.2. Quantification vectorielle	17
1.8. Conclusion.....	18

CHAPITRE 1

TECHNIQUES DE COMPRESSION DE DONNEES

1.1. Introduction

La compression est la technique utilisée pour réduire la taille physique d'un bloc d'informations. Les données compressées occupent moins d'espace que les données originales. En compressant des données, on peut placer plus d'informations dans le même espace de stockage, ou utiliser moins de temps pour le transfert à travers un réseau de transmission téléinformatique. Généralement les informations requièrent une place importante en terme de volume de stockage, la compression est devenue part intégrante des données. Presque tous les formats de fichiers utilisent l'une ou l'autre méthode de compression. Les différents algorithmes de compression sont basés sur trois critères : le taux de compression, la qualité de compression et la vitesse de compression et de décompression [Bea08], [Mor94].

Dans ce chapitre, nous allons exposer quelques notions de base sur la compression de données. On va donner une description générale sur les concepts de la compression de données tout en ouvrant un champ de réflexion en indiquant quelles sont les méthodes les plus utilisées aujourd'hui dans la compression de données.

1.2. Classification des algorithmes de compression

1.2.1. Compression symétrique et asymétrique

Dans le cas de la compression symétrique, la même méthode est utilisée pour compresser et décompresser l'information, il faut donc la même quantité de travail pour chacune de ces opérations. C'est ce type de compression qui est généralement utilisé dans les transmissions de données.

La compression asymétrique demande plus de travail pour l'une des deux opérations, on cherche souvent des algorithmes pour lesquels la compression est plus lente que la décompression. Des algorithmes plus rapides en compression qu'en décompression seront nécessaires lorsque l'on archive des données auxquelles on n'accède pas souvent (pour des raisons de sécurité par exemple) [Bena07].

1.2.2. Compression physique et logique

La compression physique agit directement sur les données, il s'agit ainsi de regarder les données redondantes d'un train de bits à un autre. La compression logique par contre est effectuée par un raisonnement logique en substituant une information par une information équivalente [Bena07].

1.2.3. Compression sans et avec perte

On distingue en général deux types de compression: la compression sans pertes (non destructible, réversible, ou conservative) qui permet de récupérer l'intégralité des données originales, et la compression avec pertes qui au contraire, perturbe les données de manière irréversible. L'intérêt de la compression avec pertes vient du fait que, pour certains types de données, il peut être possible d'introduire des dégradations indiscernables ou suffisamment faibles qui ont une influence minime sur la qualité du résultat. Cela permet d'augmenter substantiellement le taux de compression et/ou les performances des algorithmes de compression et de décompression [Mar95].

1.3. Concepts généraux

1.3.1. Entropie

En informatique, l'entropie désigne la quantité d'information que peut représenter ou contenir une source d'information. Ce concept a été introduit en 1948 par Claude Shannon [Sha48]. Une source d'information est généralement un fichier informatique quelconque. Plus une source d'information est redondante, moins elle contient d'informations au sens de Shannon. Une source d'information dont tous les symboles sont équiprobables a une entropie maximale.

L'entropie est aussi la plus petite quantité moyenne de bits nécessaire pour communiquer la vraie valeur d'une variable aléatoire. En d'autre terme, l'entropie est la plus petite quantité moyenne de bits capable de représenter l'information tirée de la variable aléatoire. Il s'agit d'une limite mathématique fondamentale pour la compression de données sans perte [Bea08].

L'entropie d'une variable aléatoire discrète X , qui peut prendre les valeurs $\{x_1, x_2, \dots, x_n\}$, est calculé : selon l'expression (1.1) où $p(x_i)$ désigne la probabilité d'occurrence de x_i .

$$H(X) = -\sum_{i=1}^n p(x_i) \log_2(p(x_i)) \quad (1.1)$$

1.3.2. Redondance

La redondance dans la théorie de l'information est la différence entre le nombre de bits utilisés pour stocker ou transmettre des données et le nombre de bits minimal nécessaire pour représenter l'information que ces données renferment. La compression de données sans perte a pour objectif de réduire le plus possible la redondance [Bea08].

1.3.3. Codage

Le codage consiste à donner un mot de code à chaque symbole d'un alphabet donné. Une fonction de codage $C : \Sigma \rightarrow \{0, 1\}$ permet de traduire un symbole de l'alphabet Σ en une séquence de bits, ou mot de code. De plus, grâce au théorème de Shannon sur le codage source, nous savons que la longueur optimale d'un mot de code m est $-\log(p)$, p étant la fréquence d'apparition du symbole représenté par m [Bea08].

1.4. Techniques de codage entropique

Un codeur entropique permet de compresser une séquence de symboles en se basant sur leurs probabilités d'apparition. Chaque symbole se voit assigner une nouvelle représentation (dépendant de sa probabilité d'apparition) de manière à ce que le codage s'approche au plus de son entropie. Les codeurs entropiques, qui vont être présentés par la suite, permettent de compresser l'information en générant un code sans ambiguïté. Ils autorisent donc de faire une restitution sans perte [Taq11].

1.4.1. Codage de Shannon-Fano

Le codage de Shannon-Fano a été proposé en 1948 par Claude Shannon [Sha48]. L'idée réside dans la répartition des symboles en deux groupes de valeurs à peu près équivalentes [Rab91].

• Algorithme

1. *Construire une table des fréquences d'apparition des symboles triés par ordre décroissant.*
2. *Diviser cette table en deux parties, celles-ci doivent avoir une somme de fréquences égale à celle de l'autre.*
3. *Affecter le chiffre binaire "0" à la moitié supérieure, la moitié inférieure prends la valeur "1".*

4. Répéter les étapes 2 et 3 aux deux parties, jusqu'à ce que chaque symbole ne représente plus qu'une partie de la table [Nel95].

• **Mise en œuvre**

Imaginons par exemple que le message Σ est constitué de 181 symboles, les fréquences d'apparition des lettres est donnée comme suit:

A	I	M	N	O	S
92/181	30/181	11/181	17/181	28/181	3/181

L'affectation des codes est réalisée selon le tableau 1.1.

Symbole	Probabilité					Etat final
A	92/181	0				0
I	30/181	1	0	0		100
O	28/181	1	0	1		101
N	17/181	1	1	0		110
M	11/181	1	1	1	0	1110
S	3/181	1	1	1	1	1111

Tableau 1.1 : Exemple de Codage de Shannon-Fano

1.4.2. Codage de Huffman

Ce type de codage a été introduit en 1952 par David A. Huffman [Huf52]. Il a été utilisé intensivement en compression de données. En effet, Le codage de Huffman produit un code préfixe optimal. La construction d'un code de Huffman est assez simple. Bien qu'il existe des techniques plus efficaces pour construire un code de Huffman, la technique présentée est simple et est souvent utilisée dans plusieurs applications. L'algorithme pour construire un code de Huffman pour un alphabet Σ est le suivant [Bea08]:

1. Les messages constituent les feuilles d'un arbre portant chacune un poids égal à la probabilité P d'occurrence du message correspondant.
2. Joindre les deux nœuds de moindre poids en un nœud parent auquel on attache un poids égal à la somme de ces deux poids.
3. Répéter la deuxième étape jusqu'à l'obtention d'une seule racine à l'arbre (de poids égal 1).

4. Affecter les codes "0" et "1" aux nœuds descendants directs de la racine.
5. Continuer à descendre en affectant des codes à tous les nœuds, chaque paire de descendants reçoit les codes "0" ou "1".

• **Mise en œuvre**

Imaginons toujours l'exemple de l'alphabet Σ du paragraphe 1.4.1. On peut construire un arbre binaire selon le codage de Huffman (Figure 1.1). Où A, I, M, N, O et S seront codés respectivement par 1, 011, 010, 001, 0001 et 0000.

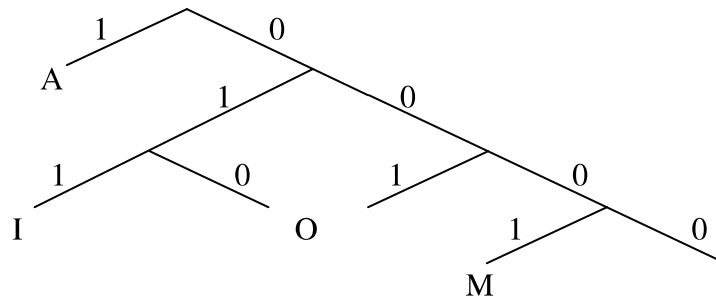


Figure 1.1 : Exemple de Huffman

Le coût espéré d'un symbole avec le codage de Shannon-Fano et le coût espéré d'un symbole avec le codage de Huffman seront calculées à partir des équations (1.2) et (1.3) respectivement:

$$P(A) \times 1 + P(I) \times 3 + P(O) \times 4 + P(N) \times 3 + P(M) \times 4 + P(S) \times 4 = 2.21 \text{ bits} \quad (1.2)$$

$$P(A) \times 1 + P(I) \times 3 + P(O) \times 3 + P(N) \times 3 + P(M) \times 4 + P(S) \times 4 = 2.06 \text{ bits} \quad (1.3)$$

Où $P(x)$ représente la probabilité du symbole x . Le coût espéré d'un symbole avec le codage de Huffman est 2.06 bits et avec le codage de Shannon-Fano 2.21 bits. Le codage de Shannon-Fano n'est donc pas optimal, par contre; le codage de Huffman est un code préfixe optimal. En combinant les symboles en courtes séquences, il y a des moyens d'atténuer l'effet néfaste de l'intégralité des bits dans les mots de code.

1.4.3. Codage arithmétique

Le codage arithmétique traite l'ensemble d'une séquence de symboles comme une seule entité. Ce qui lui permet de réussir à sortir de la contrainte imposée par les approches VLC (Variable Length Coding), qui attribuent un mot de code court aux symboles probables et des mots plus longs aux symboles moins fréquents. Ainsi, le codage

arithmétique permet de représenter une séquence de symboles par un intervalle de nombres réels compris entre 0 et 1. Il est à signaler que toute valeur appartenant à ce dernier intervalle représentera d'une manière unique la séquence à coder [Mas07].

A mesure que la séquence s'allonge, l'intervalle requis pour le représenter diminue et le nombre de bits qui servent à préciser cet intervalle s'accroît. Les symboles successifs d'une séquence réduisent cet intervalle en concordance avec la probabilité d'apparition du symbole. Finalement, les données comprimées consistent en la partie fractionnaire du plus court nombre binaire qui se trouve dans l'intervalle final.

L'algorithme du codage arithmétique est le suivant:

1. Soit X une séquence de m symboles $X = x_1 x_2 \dots x_m$ qui prend ses valeurs à partir d'une source $S = \{s_1, s_2, \dots, s_n\}$.
2. Calculer la probabilité associée à chaque symbole de la source S , notons $p_i = \text{probabilité}(S=s_i)$.
3. Associer à chaque symbole s_k un sous intervalle $[Ls_k, Hs_k]$ proportionnel à sa probabilité, avec $Hs_k - Ls_k = p_k$.
4. Initialiser la limite inférieure L de l'intervalle de travail à la valeur 0 et la limite supérieure H à la valeur 1.
5. Tant qu'il reste un symbole dans la séquence à coder :
$$\text{Largeur} = H - L$$
$$L = L + \text{largeur} \times Ls_k$$
$$H = L + \text{largeur} \times Hs_k$$
6. A la fin, n'importe quelle valeur de l'intervalle $[L, H]$ représente d'une manière unique la séquence d'entrée.

Généralement, on choisit comme représentant de la séquence la limite inférieure (L) de l'intervalle.

Il est à noter qu'à la fin du processus de codage, le codeur arithmétique génère un fichier composé par un en-tête suivi de la représentation binaire du code choisi. De plus, l'en-tête peut contenir soit la table des probabilités de tous les symboles.

• Mise en œuvre

Soit, par exemple, à coder la séquence $X = \text{"BILL GATES"}$. La probabilité d'occurrence de chaque symbole, ainsi que l'intervalle associé à chaque symbole sont données dans le tableau 1.2. En appliquant l'algorithme de codage arithmétique sur la séquence X , on obtiendra l'évolution du tableau 1.3.

Symbole	Probabilité	Borne inférieure	Borne supérieure
Espace	1/10	0	0.0999
A	1/10	0.1	0.1999
B	1/10	0.2	0.2999
E	1/10	0.3	0.3999
G	1/10	0.4	0.4999
I	1/10	0.5	0.5999
L	2/10	0.6	0.7999
S	1/10	0.8	0.8999
T	1/10	0.9	0.9999

Tableau 1.2 : Association de la probabilité et de l'intervalle pour chaque symbole

Symbole	L	H
B	0.2	0.3
I	0.25	0.26
L	0.256	0.258
L	0.2572	0.2576
Espace	0.25720	0.25724
G	0.257216	0.257220
A	0.2572164	0.2572168
T	0.25721676	0.2572168
E	0.257216772	0.257216776
S	0.2572167752	0.2572167756

Tableau 1.3 : Exemple de codage Arithmétique

Donc 0.2572167752 est la représentation arithmétique du message "BILL GATES".

1.5. Algorithmes de compression sans pertes

Dans cette section, nous allons présenter deux types d'algorithmes utilisant la compression sans perte : la technique RLE (Run Length Encoding) et l'algorithme LZW (Lempel Ziv Welch).

1.5.1. Technique RLE

La technique RLE (codage par plages) est probablement l'algorithme de compression de données le plus simple. Il ne fait que remplacer plusieurs apparitions d'un même symbole par un exemplaire du symbole et le nombre de fois qu'il apparaît consécutivement. Par exemple, si nous avons une série de A dans un document : AAAAA, RLE remplace la série de A par un message du type A5. Evidemment, il faut un moyen

pour différencier le message A5 de la séquence A5 dans les données originales. Une séquence d'échappement, c'est à dire un nouveau symbole que l'on introduit dans notre alphabet, peut être utilisée à cette fin. Il suffit alors de transmettre la séquence d'échappement avant de transmettre le nombre d'apparitions d'un symbole, le cas échéant [Bea08].

La méthode RLE est très efficace avec les fichiers dont les mêmes données se répètent souvent, comme une image en noir et blanc.

1.5.2. Technique LZW

L'algorithme universel LZW crée par Lempel, Ziv et Welch, il a été publié par Welch en 1984 [Wel84]. C'est une amélioration de l'algorithme LZ78 publié par Lempel et Ziv en 1978 [Lem78]. Cette technique permet de substituer un texte à un autre via des règles de traduction définies dans un dictionnaire dynamique, c'est-à-dire un dictionnaire mis à jour lors de l'encodage même. Cette compression est efficace pour de gros fichiers. Il n'en est pas de même pour un petit fichier, comme on le voit ci-dessous dans l'exemple : le message est trop court pour pouvoir bénéficier des avantages de la compression LZW [Tao04].

Algorithme

1. *On initialise le dictionnaire en plaçant les codes des caractères ASCII étendu.*
2. *On regarde le caractère à transmettre :*
 - *S'il existe déjà dans la table, on regarde le caractère suivant.*
 - *Si le groupe des deux existe également, on regarde le suivant, etc.*
3. *Lorsqu'un nouveau groupe est découvert, on le définit en l'insérant dans le dictionnaire.*
4. *Dans un premier temps, on transmet les codes des morceaux qui le composent. La prochaine fois qu'on le rencontrera, on transmettra son code propre. Les mots ajoutés au dictionnaire seront déterminés par l'intermédiaire d'une fenêtre évoluant au fil de l'analyse du texte à compresser.*
5. *On procède de la sorte jusqu'à la fin de la transmission.*
6. *Lorsque le dictionnaire est plein, soit on procède à son extension, soit on se borne à utiliser les codes déjà existants. La base du dictionnaire repose sur les 256 caractères du code ASCII étendu.*

• **Mise en œuvre**

Imaginons le message "ma maison" à compresser.

1. Le premier caractère à analyser est "m". Il est dans le dictionnaire.
2. Le caractère suivant est concaténé avec "m" et forme la chaîne "ma". Elle n'est pas dans le dictionnaire, il faut donc l'ajouter. On lui affecte donc l'entrée 256 dans le dictionnaire (on commence à 0). On envoie la valeur correspondante à "m"(109).
3. Un nouveau mot vient d'être ajouté, on reprend alors au dernier caractère pris en compte ("a"). Le caractère suivant est concaténé et forme la chaîne "a ". Elle n'est pas dans le dictionnaire, il faut donc l'ajouter. On lui affecte donc l'entrée 257 dans le dictionnaire. On envoie la valeur correspondante à "a"(97).
4. Un nouveau mot vient d'être ajouté, on reprend alors au dernier caractère pris en compte (" "). Le caractère suivant est concaténé et forme la chaîne " m". Elle n'est pas dans le dictionnaire, il faut donc l'ajouter. On lui affecte donc l'entrée 258 dans le dictionnaire. On envoie la valeur correspondante à " "(32).
5. Un nouveau mot vient d'être ajouté, on reprend alors au dernier caractère pris en compte ("m"). Le caractère suivant est concaténé et forme la chaîne "ma". Cette chaîne est déjà dans le dictionnaire. On concatène le caractère suivant et cela forme la chaîne "mai". Elle n'est pas dans le dictionnaire, il faut donc l'ajouter. On lui affecte donc l'entrée 259 dans le dictionnaire. On envoie donc la valeur correspondante à "ma"(256).
6. Un nouveau mot vient d'être ajouté, on reprend alors au dernier caractère pris en compte ("i"). Le caractère suivant est concaténé et forme la chaîne "is". Elle n'est pas dans le dictionnaire, il faut donc l'ajouter. On lui affecte donc l'entrée 260 dans le dictionnaire. On envoie la valeur correspondante à "i"(105).
7. Un nouveau mot vient d'être ajouté, on reprend alors au dernier caractère pris en compte ("s"). Le caractère suivant est concaténé et forme la chaîne "so". Elle n'est pas dans le dictionnaire, il faut donc l'ajouter. On lui affecte donc l'entrée 261 dans le dictionnaire. On envoie la valeur correspondante à "s"(115).
8. Un nouveau mot vient d'être ajouté, on reprend alors au dernier caractère pris en compte ("o"). Le caractère suivant est concaténé et forme la chaîne "on". Elle n'est pas dans le dictionnaire : il faut donc l'ajouter. On lui affecte donc l'entrée 262 dans le dictionnaire. On envoie la valeur correspondante à "o"(111).
9. Un nouveau mot vient d'être ajouté, on reprend alors au dernier caractère pris en compte ("n"). Le caractère suivant est concaténé et forme la chaîne "n(eof)". Elle n'est pas dans le dictionnaire, il faut donc l'ajouter. Etant donné le caractère spécifique de fin de

fichier, il n'est pas ajouté au dictionnaire. On envoie la valeur correspondante à "n"(110).

A la fin, le flux compressé sera (109) (97) (32) (256) (105) (115) (111) (110).

1.6. Techniques de compression à base des transformées

La compression est dite avec perte lorsqu'il y a permission d'altérer les données lors de la compression. Cette perte d'information permet, entre autres, de compresser davantage les données. La compression de données avec perte est plus appropriée aux données comme l'image, car il est rarement nécessaire de conserver toute l'information disponible à la source [Bea08].

Plusieurs techniques de compression avec perte sont basées sur le codage des coefficients dans le plan transformé. On peut citer JPEG (Joint Photographic Expert Group) basée sur la DCT (Discrete Cosine Transform) et JPEG2000 basée sur la DWT (Discrete Wavelet Transform).

L'étape de transformation T consiste à représenter l'information dans un autre domaine afin de faciliter la compression (figure 1.2). Pour atteindre ce but, on choisit des transformées qui ont la propriété de concentrer l'information sur un petit nombre de coefficients. Cette étape est inversible, T^{-1} sera donc appliquée comme dernière étape de la chaîne de décodage. Parmi les transformations les plus fréquemment utilisées; la transformée de Fourier discrète (DFT) et ses dérivées (transformée en cosinus discrète (DCT)) [Coi95], transformée de Karhunen-Loeve (KLT) et les décompositions en sous-bandes qui incluent les transformées en ondelettes ou en paquets d'ondelettes [Coi93], [Mat02].

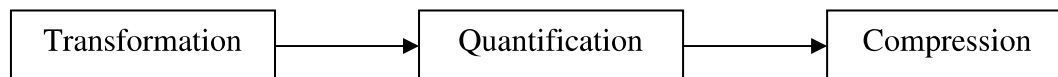


Figure 1.2 : Schéma de codage par transformée

1.6.1. Transformée en cosinus discrète

La DCT est une transformation analogue à la transformée de Fourier discrète. Elle sert à concentrer l'information et la rendre ainsi exploitable pour la compression des images, c'est le cas de la norme JPEG provenant du groupe d'experts internationaux.

Sachant que la DCT a été proposée en 1974 par le professeur Rao de l'université de Texas en 1974 [Sal04].

La DCT présente plusieurs avantages : elle est réelle et peut être calculée à l'aide d'un algorithme rapide (d'où son implémentation facile). D'autre part, les coefficients sont bien décorrélés dans le domaine transformé. Elle présente une excellente concentration d'énergie pour les données hautement corrélées.

La DCT peut être vue comme une version discrète des séries de Cosinus de Fourier. Par ses propriétés, elle est plus proche de la transformée KLT calculée en utilisant une transformée de Fourier rapide (FFT: Fast Fourier Transform). A la différence de la FFT, la DCT produit des coefficients à valeurs réelles et fournit une meilleure approximation du signal avec moins de coefficients.

1.6.2. Transformée en ondelettes discrète

Souvent appelée analyse multirésolution en ondelettes. Cette méthode consiste à faire introduire le signal à analyser dans deux filtres passe-bas (L) et passe-haut (H). A ce niveau, deux vecteurs seront obtenus : $cA1$ et $cD1$. Les éléments du vecteur $cA1$ sont appelés coefficients d'approximation au niveau 1, ils correspondent aux plus basses fréquences du signal, tandis que les éléments du vecteur $cD1$ sont appelés coefficients de détail au niveau 1, ils correspondent aux plus hautes fréquences du signal. La procédure peut être répétée avec les éléments du vecteur $cA1$ et successivement avec chaque nouveau vecteur cAk obtenu, k représente le numéro du niveau de décomposition.

1.7. Quantification

1.7.1. Quantification scalaire

La quantification scalaire est l'approximation de chaque valeur du signal aléatoire $x(t)$ par une valeur qui appartient à un ensemble fini de codes $\{y_1, y_2, \dots, y_L\}$. Pour toute amplitude x comprise dans l'intervalle $[x_{i-1}, x_i]$, on fait correspondre une valeur quantifiée y_L située dans cet intervalle [Baa04].

Considérons un signal à temps continu $x(t)$ et à bande limitée $[-B, +B]$. La fréquence de Nyquist $f_e = 2B$ définit la fréquence d'échantillonnage la plus basse n'entraînant aucune perte d'informations [Del91]. Dans la pratique, il n'existe pas de signaux strictement à bande limitée puisqu'ils sont de durée finie mais on peut s'en approcher autant que l'on veut en réalisant un filtrage analogique. Cette procédure d'échantillonnage fournit la suite des échantillons $x(n)$.

Il faut maintenant réaliser une discrétisation numérique du signal. On parle alors de quantification, dont la résolution représente le nombre de bits disponibles pour représenter un échantillon.

1.7.1.1. Quantification scalaire uniforme

Considérons un signal à temps discret $x(n)$ prenant ses valeurs dans l'intervalle $[-A, +A]$ avec une loi uniforme. Pour définir un quantificateur, on doit suivre les démarches suivantes :

1. Partitionner l'intervalle $[-A, +A]$ en $L=2^b$ intervalles distincts $\{P^1 \dots P^L\}$ de même longueur $\Delta=2A/2^b$,
2. Numéroté chaque intervalle,
3. Définir un représentant par intervalle, par exemple le milieu de l'intervalle.

La procédure d'encodage consiste à décider à quel intervalle appartient $x(n)$ puis à lui associer le numéro $i(n) \in \{1 \dots L=2^b\}$ correspondant. C'est le numéro de l'intervalle choisi, le symbole canal, qui sera transmis ou stocké [Mor94].

Une quantification uniforme est caractérisée par :

- Nombre de niveaux quantifiés $L+1=2^b$ (b bits)

- Pas de quantification : $\Delta=x_i-x_{i-1}$ (1.4)

- Erreur de quantification : $e=y-x$ (1.5)

- Erreur de granulation : $|e| \leq \Delta/2$ (1.6)

1.7.1.2. Quantification scalaire non uniforme

Les hypothèses faites précédemment sont, bien entendu mal adaptés à un signal de parole, de musique ou d'image. Les échantillons $x(n)$ ne peuvent pas être interprétés comme la réalisation d'un processus aléatoire stationnaire prenant ses valeurs de façon uniformément réparties dans l'intervalle $[-A, +A]$. Le quantificateur uniforme n'est plus le quantificateur optimal. Le fait que ce dernier doit avoir la forme de la figure 1.3. La partition $\{P^1 \dots P^L\}$ de l'axe réel n'est plus composée d'éléments de longueur constante. La longueur est d'autant plus petite que la densité de probabilité correspondante est importante. Les représentants $\hat{x}^i$ ne sont plus au milieu des intervalles [Mor94].

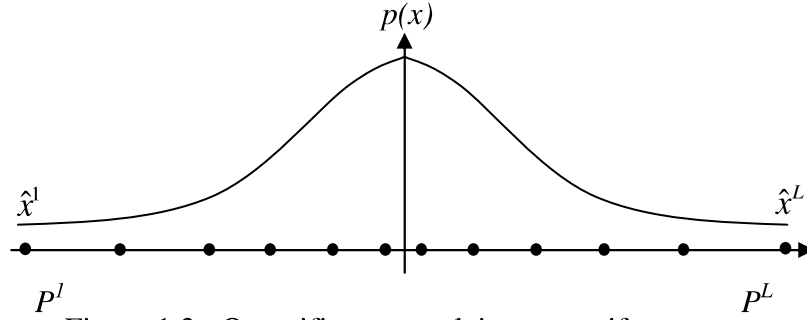


Figure 1.3 : Quantificateur scalaire non uniforme

1.7.2. Quantification vectorielle

La quantification vectorielle permet de passer d'un espace vectoriel multidimensionnel à un sous-espace discret de plus petite dimension. Diminuer le nombre de dimensions permet ainsi de réduire la taille des données.

La quantification vectorielle est une généralisation immédiate de la quantification scalaire. On appelle quantificateur vectoriel de dimension N et de taille L une application de R^N dans un ensemble fini C contenant L vecteurs de dimension N .

$$Q : R^N \rightarrow C \text{ Avec } C = \{y^1 \dots y^L\} \quad (1.7)$$

L'espace R^N est partitionné en L régions ou cellules définies par :

$$\theta^i = \{x : Q(x) = y^i\} \quad (1.8)$$

On appelle C un dictionnaire, assimilable à une matrice si nécessaire, et y^i un représentant, un vecteur de sortie ou un vecteur de reproduction. On dit également que C représente l'alphabet de reproduction et y^i les symboles de reproduction conformément au vocabulaire habituel en théorie de l'information (figure 1.4).

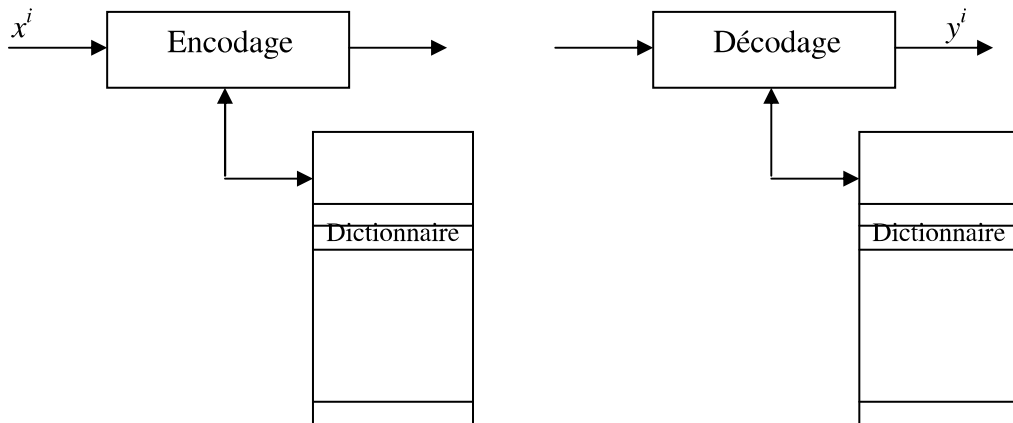


Figure 1.4 : Schéma synoptique d'un quantificateur vectoriel

Généralement Il s'agit de déterminer le dictionnaire C , c'est à dire les vecteurs de reproduction $\{y^1 \dots y^L\}$. Cette notion conduit à deux conditions nécessaires d'optimalité qui s'expriment de façon identique au cas scalaire :

- Etant donné un dictionnaire $C = \{y^1 \dots y^L\}$.
- Etant donné une partition, les meilleurs représentants sont obtenus par la condition du centroïde [Mor09].

$$y^i = E\{x/x \in \theta\} \quad (1.9)$$

- Algorithme des k-moyennes (k-means)

L'algorithme des k-moyennes est un simple algorithme de quantification vectorielle qui utilise un critère d'erreur quadratique comme mesure de distorsion. Il est initialisé avec un partitionnement aléatoire, réassigne continuellement les données aux partitions en se basant sur la distance entre les données et le centre de la partition (également appelé prototype), et ce jusqu'à ce qu'un critère de convergence soit atteint. Il peut être résumé comme suit :

- *Choix de k prototypes arbitraires.*
- *Associer chaque donnée au prototype le plus proche (distance euclidienne).*
- *Recalculer les prototypes comme barycentres des données associées. Autrement dit, si les points $x_1, \dots, x_{N_j}$ sont associés à la j^{ieme} partition, alors le nouveau prototype est le suivant:*

$$c_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i \quad (1.10)$$

- *Répéter les deux étapes précédentes jusqu'à la convergence.*

1.8. Conclusion

On a présenté dans ce chapitre les techniques de compression de données : une introduction aux concepts de base sur la compression et les techniques de compression sans et avec perte.

Toutes les méthodes présentées dans ce chapitre produisent un flux de données permettant de reconstruire d'une part une copie exacte des données compressées et d'autre part une copie dégradée des données compressées.

Aussi pour améliorer le taux de compression, il s'avère absolument nécessaire d'utiliser les techniques à base des transformées. Le type de transformation qui nous intéresse dans notre travail est la transformée en ondelettes. Elle sera détaillée dans le chapitre suivant.

Chapitre 2

TRANSFORMEES EN ONDELETTES

Contenu

2.1. Introduction.....	19
2.2. Historique des ondelettes.....	19
2.3. Définition des ondelettes.....	20
2.4. De la transformée de Fourier à la transformée en ondelette.....	21
2.5. Transformée en ondelettes continue.....	22
2.5.1. Transformée en ondelettes continue 1D.....	23
2.5.2. Transformée en ondelettes continue 2D.....	23
2.5.3. Transformée en ondelettes continue 3D.....	24
2.6. Transformée en ondelettes discrète.....	24
2.6.1. Transformée en ondelettes discrète 1D.....	24
2.6.2. Transformée en ondelettes discrète 2D.....	26
2.6.3. Transformée en ondelettes discrète 3D.....	27
2.7. Comparaison entre CWT et DWT.....	27
2.8. Familles de la transformée en ondelettes.....	27
2.9. Avantages de la transformée en ondelettes.....	29
2.10. Applications des ondelettes.....	30
2.11. Conclusion.....	32

CHAPITRE 2

TRANSFORMEES EN ONDELETTES

2.1. Introduction

Les ondelettes est un outil de traitement du signal permettant l'analyse, à plusieurs échelles de temps, des propriétés locales de signaux complexes. Elles donnent lieu à de nombreuses applications dans des domaines variés, comme par exemple la géophysique, l'astrophysique, les télécommunications, l'imagerie et le codage vidéo. Elles sont à la base de nouvelles techniques d'analyse et de synthèse du signal et trouvent une multitude d'applications pour des problèmes généraux tels que la compression et le débruitage.

Dans ce chapitre, on va exposer les notions de base de la transformée en ondelettes à savoir la transformée en ondelettes continue et la transformée en ondelettes discrète. Pour chaque approche on définit les transformées en ondelettes pour les trois dimensions 1D, 2D et 3D. Par la suite, on décrit brièvement quelques types de familles utilisées dans les transformées en ondelettes. Les applications de ces dernières seront résumées à la fin de ce chapitre.

2.2. Historique des ondelettes

Avant le début des années 1980, les techniques préfigurant la théorie des ondelettes avaient déjà été introduites au sein de certaines communautés scientifiques. En dépit d'une perspective scientifique commune, chacun avait imaginé et développé le concept des ondelettes pour ses propres besoins, selon une technique particulière.

Ce n'est qu'en 1989 que Stéphane Mallat [Mal89] relie les communautés entre elles, puisqu'il découvre les relations étroites entre les filtres miroirs en quadrature, les algorithmes pyramidaux utilisés en traitement numérique de l'image et les bases orthonormées d'ondelettes. Ce rapprochement des communautés a relancé des travaux de recherche très actifs sur la théorie des ondelettes accompagnés de toutes sortes d'applications.

Les premières utilisations de l'ondelette comme outil d'analyse ou de codage sont ensuite arrivées très rapidement, ce qui montre l'intérêt de cette technique pour l'industrie.

Le premier usage industriel qui en a été fait, fut l'archivage des empreintes digitales de la population américaine par le FBI. Un taux de compression de 90% sur les coefficients en ondelettes a été atteint autorisant un archivage qui restitue des images visuellement quasi sans pertes. Cette importante application d'archivage des années 80 a ouvert la voie à l'élaboration d'un projet de standard en 1996 pour la compression d'images. Un circuit intégré de compression vidéo par les ondelettes a également vu le jour en 1997.

2.3. Définition des ondelettes

Une ondelette est une petite onde de durée limitée admissible et possédant une énergie concentrée vers les hautes fréquences et à valeur moyenne nulle vérifie la condition générale :

$$\int_{-\infty}^{+\infty} \psi(t) d^n t = 0 \quad (2.1)$$

Avec n la dimension de l'espace, les dimensions les plus courantes étant la dimension 1 pour les signaux sonores et de télécommunication, la dimension 2 pour l'imagerie et la dimension 3 pour l'imagerie volumique et les vidéos. Si l'ondelette mère ψ est une fonction admissible [Lou97], les ondelettes enfants $\psi_{\tau,s}$ seront obtenues utilisant la formule suivante [Som10], [Ste05]:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \quad (2.2)$$

- τ : Facteur de translation.
- s : Facteur d'échelle (dilatation).
- $\frac{1}{\sqrt{s}}$: Facteur de normalisation de l'énergie afin que le signal transformé ait la même énergie à toutes les échelles.

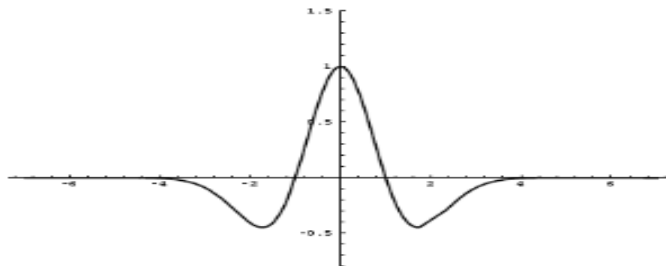


Figure 2.1 : Ondelette mère

2.4. De la transformée de Fourier à la transformée en ondelette

La plupart des signaux ne sont pas stationnaires, l'essentiel de l'information qu'ils contiennent réside dans la non stationnarité. L'analyse de Fourier propose une approche globale du signal. Toute notion temporelle dans l'espace de Fourier (espace fréquentiel) disparaît. Il faut trouver une transformation qui nous renseigne sur le contenu fréquentiel tout en préservant la localisation afin d'obtenir une représentation temps / fréquence.

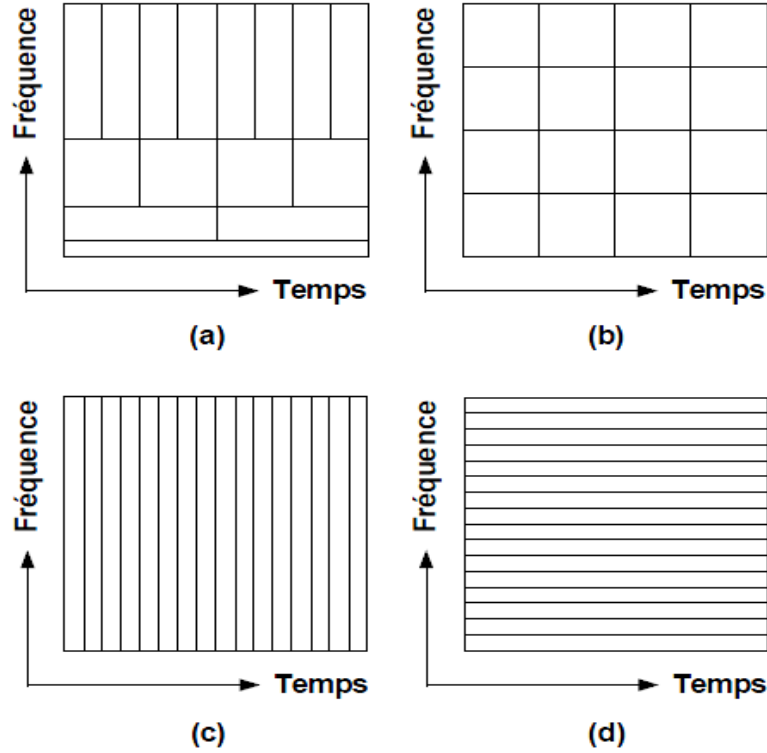
Plusieurs solutions ont été proposées, à savoir la transformée de Fourier à fenêtre glissante et la transformée de Gabor. Mais ces deux méthodes donnent une même résolution temporelle pour les hautes et les basses fréquences, donc l'analyse n'est pas idéale. C'est dans ce contexte que la transformée en ondelettes intervient, elle propose une solution au compromis entre la résolution temporelle et la résolution fréquentielle [Gra95].

On considère la transformée de Fourier à court terme (Short Time Fourier Transform : STFT) exprimée par :

$$STFT(\tau, f) = \int_{-\infty}^{+\infty} f(t)g(t-\tau)e^{-j2\pi ft} dt \quad (2.3)$$

La différence entre la STFT et la transformée de Fourier usuelle (FT) est l'ajout d'une fenêtre glissante $g(t)$ qui se déplace dans le temps d'un intervalle τ . La CWT et la STFT permettent donc de produire une analyse du contenu fréquentiel d'un signal de façon quasi instantanée, c'est à dire pour une portion du signal allant de t à $t+\tau$. Elles décomposent le signal exprimé en fonction du temps en une multitude d'atomes ou coefficients dans le domaine temps-fréquence. La principale différence entre les deux outils est le découpage atomique du plan temps-fréquence, pour la STFT, la largeur temporelle de l'atome dépend de la fenêtre $g(t)$ qui est constante, peu importe la fréquence. Tandis que le découpage par CWT produit des fenêtres plus larges dans le temps lorsque la fréquence est basse et inversement. Ainsi, les transitoires rapides d'un signal sont prises en compte par des atomes de hautes fréquences qui ont une résolution temporelle très fine, alors que les mouvements plus lents, de basses fréquences, peuvent bénéficier d'une représentation fréquentielle plus précise.

La figure 2.2 montre également les découpages atomiques obtenus par l'échantillonnage du signal dans le temps, la FT, la STFT et la transformée en ondelette [Sim05]. Il est à noter que la transformée en ondelettes donne une meilleure résolution temporelle en hautes fréquences et une meilleure résolution fréquentielle en basses fréquences.



(a) Transformée en ondelettes (b) STFT (c) Échantillonnage temporel (d) FT.

Figure 2.2 : Découpage en temps fréquence

Les transformées en ondelettes se divisent en deux catégories :

- La transformée en ondelettes continue ou Continuous Wavelet Transform (CWT).
- La transformée en ondelettes discrète ou Discrete Wavelet Transform (DWT).

2.5. Transformée en ondelettes continue

La transformée en ondelettes continue utilise des translations et des dilatations de la fonction ondelette mère durant tout l'intervalle de temps de manière continue.

La définition de la CWT est donnée par la formule ci-dessous :

$$CWT(\tau, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t) \psi\left(\frac{t-\tau}{s}\right) dt \quad (2.4)$$

La transformée en ondelettes inverse, c'est-à-dire la reconstruction du signal d'origine, est donnée par l'équation 2.5 [Ste05].

$$f(t) = \frac{1}{C_{\psi}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} CWT(\tau, s) \psi\left(\frac{t-\tau}{s}\right) \frac{d\tau ds}{s^2} \quad (2.5)$$

L'existence de la transformation inverse dépend de la convergence du coefficient C_ψ :

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\psi(f)|}{f} df < \infty \quad (2.6)$$

2.5.1. Transformée en ondelettes continue 1D

La transformée en ondelettes continue d'un signal $f(t)$ à une dimension fournit une décomposition du signal à différentes échelles grâce à l'utilisation de l'ondelette mère ψ dilatée d'un facteur s et translatée d'un facteur τ [Dau92].

On obtient de cette manière une analyse dont la résolution fréquentielle et temporelle est variable. C'est grâce à cette propriété que la théorie des ondelettes a connu un tel succès. Faire varier τ permet d'analyser localement le signal, et ce à différentes échelles grâce à s .

2.5.2. Transformée en ondelettes continue 2D

Les ondelettes continues 2D sont engendrées par des translations et dilatations de l'ondelette mère $\psi_{a,b}(x, y)$ qu'on peut définir comme :

$$\psi_{a,b}(x, y) = \frac{1}{\sqrt{a_x a_y}} \psi\left(\frac{x - b_x}{a_x}, \frac{y - b_y}{a_y}\right) \quad (2.7)$$

On définit la transformée en ondelettes continue 2D d'une fonction $f(x, y)$ par :

$$CWT(a, b) = \frac{1}{\sqrt{a_x a_y}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \psi\left(\frac{x - b_x}{a_x}, \frac{y - b_y}{a_y}\right) dx dy \quad (2.8)$$

Où a est le facteur d'échelle constitué de a_x et a_y qui sont respectivement les paramètres d'échelles responsables de la contraction ou de la dilatation selon les directions x et y . b est le facteur de translation composé de b_x et b_y qui sont les paramètres de la translation selon les directions x et y [Sim05].

Le coefficient $CWT(a, b)$ est plus élevé en valeur absolue si la corrélation entre $\psi_{a,b}(x, y)$ et $f(x, y)$ est importante.

En pratique, la transformée continue 2D est obtenue en appliquant deux transformées en ondelettes 1D successives, l'une horizontale, l'autre verticale. Ainsi, la transformée en ondelettes continue 2D peut se résumer à l'utilisation de deux transformées 1D.

2.5.3. Transformée en ondelettes continue 3D

On définit la transformée en ondelettes continue 3D d'une $f(x, y, z)$ par :

$$CWT(a, b) = \frac{1}{\sqrt{a_x a_y a_z}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y, z) \psi\left(\frac{x-b_x}{a_x}, \frac{y-b_y}{a_y}, \frac{z-b_z}{a_z}\right) dx dy dz \quad (2.9)$$

Où a_x , a_y et a_z sont respectivement les paramètres d'échelles responsables de contraction ou de dilatation selon les directions x , y et z . b_x , b_y et b_z sont les paramètres de la translation selon les directions x , y et z .

2.6. Transformée en ondelettes discrète

La transformée en ondelettes continue est en théorie infiniment redondante puisque l'ondelette est translatée de manière continue. Cependant il existe des méthodes pour diminuer cette redondance l'une de ces méthodes consiste en l'emploi de la transformée en ondelettes discrète.

2.6.1. Transformée en ondelettes discrète 1D

La DWT 1D est la version discrète de la transformée en ondelettes continue 1D. Elle est réalisée à l'aide de l'algorithme de Mallat [Mal89] en utilisant l'analyse multirésolution. Cet algorithme est basé sur la définition d'une paire de filtres appelés filtres à miroirs quadratiques LPF (filtre basses fréquences) et HPF (filtre hautes fréquences) pour donner deux vecteurs respectivement L (Coefficients ondelette d'approximation) et H (Coefficients ondelette de détails). Tous les deux sont de taille approximativement égale à la moitié du vecteur d'origine. Ceci est dû au fait de la décimation par 2. La reconstruction parfaite est théoriquement possible. Les deux vecteurs L et H, en les faisant passer par un sur échantillonnage puis un filtrage respectivement par les filtres LPF (filtre passe bas de reconstruction), HPF (filtre passe haut de reconstruction) donnent en somme le vecteur d'origine $x(n)$ (Figure 2.3) [Maa08], [Ste05].

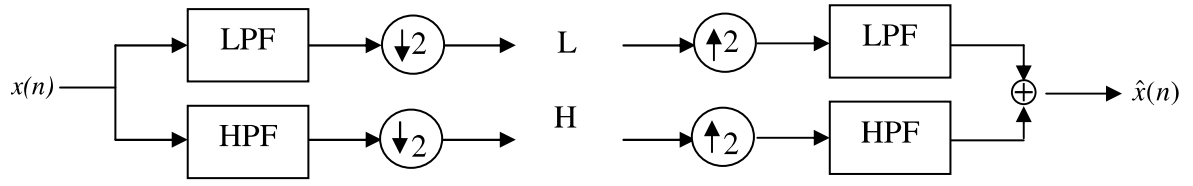


Figure 2.3 : Transformée en ondelettes discrète 1D

La transformée en ondelettes discrète 1D translate et dilate l'ondelette selon des valeurs discrètes. Ces coefficients τ et s seront discrétisés de la manière suivante :

$$s = s_0^j \text{ et } \tau = k\tau_0 s_0^j \text{ avec } s_0 > 1 \text{ et } \tau_0 > 0.$$

Les ondelettes sont alors définies de la manière suivante :

$$\psi(\tau_0, s_0) = \frac{1}{\sqrt{s_0^j}} \psi\left(\frac{nT - k\tau_0 s_0^j}{s_0^j}\right) \quad (2.10)$$

- s_0^j : Facteur d'échelle.
- τ_0 : Facteur de translation.
- k et j : entiers.
- T période d'échantillonnage.

La DWT du signal $x(n)$ est donnée par la formule suivante :

$$DWT(\tau_0, s_0) = \sum_n x(n) \psi(\tau_0, s_0) \quad (2.11)$$

L'analyse multirésolution est un outil de traitement du signal qui permet de décomposer un signal à plusieurs échelles (résolutions) et de le reconstruire à partir des éléments de cette décomposition. Une analyse multirésolution est un partitionnement de l'espace des fonctions d'énergie finie par une famille de sous-espaces vectoriels :

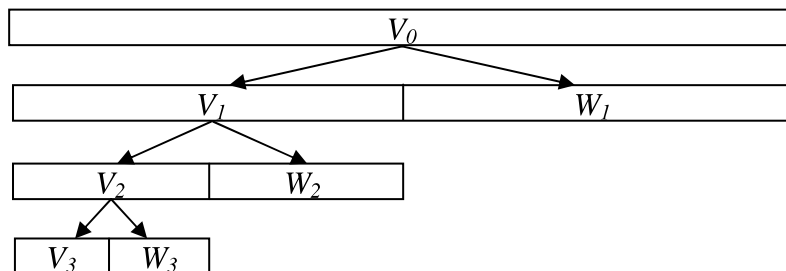


Figure 2.4 : Principe de l'analyse multirésolution

L'idée est de projeter un signal $x(n)$ appartenant à un espace V_j sur un sous espace V_{j+1} et un sous espace W_{j+1} dans le but de réduire la résolution de moitié. V_{j+1} est le sous espace d'approximation et W_{j+1} est le sous espace de détails (Figure 2.4).

2.6.2. Transformée en ondelettes discrète 2D

La DWT 2D peut être rapprochée de la notion d'analyse en sous bandes dans laquelle le spectre du signal à analyser est décomposé récursivement en une sous bande de détails (hautes fréquences) et une sous bande d'approximation (basses fréquences) (figure 2.5) [Som10].

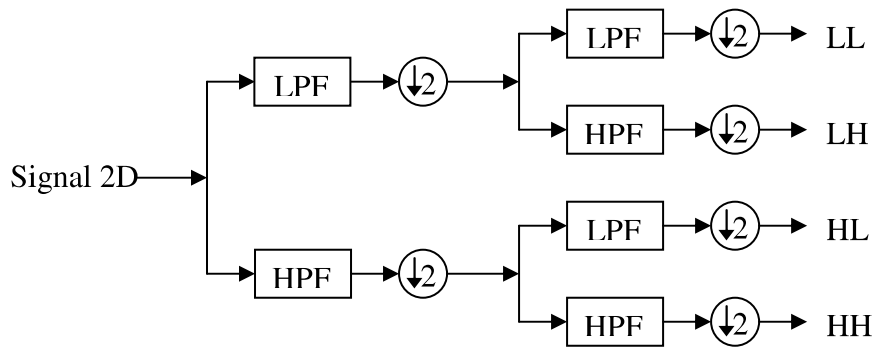


Figure 2.5 : Filtres pour la DWT2

Pour un signal à deux dimensions, ce ne sont pas deux, mais quatre sous bandes qui sont extraites à chaque niveau de décomposition: une sous bande d'approximation (LL) contenant l'information basse fréquence selon les deux directions spatiales, une sous bande de détails horizontaux (LH), une sous bande de détails verticaux (HL) et une sous bande de détails diagonaux (HH) contenant les composantes hautes fréquences suivant les deux directions (Figure 2.6) [Sav04].

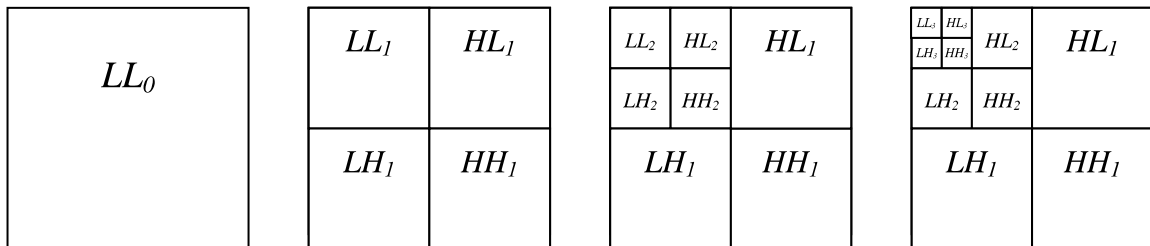


Figure 2.6 : Application de la DWT2 pour trois niveaux de décompositions

2.6.3. Transformée en ondelettes discrète 3D

Le principe de base de la transformée en ondelettes discrète 3D (Figure 2.7) est de séparer le signal entrée en deux composantes, l'une représentant l'allure générale du signal (approximation), l'autre représentant ses détails. L'allure générale d'une fonction est représentée par ses basses fréquences, les détails par ses hautes fréquences. Cette séparation s'effectue à l'aide d'une paire de filtres : un filtre passe bas (LPF) pour obtenir l'approximation, et un filtre passe haut (HPF) pour estimer ses détails. En répétant cette opération sur les trois directions x , y et z .

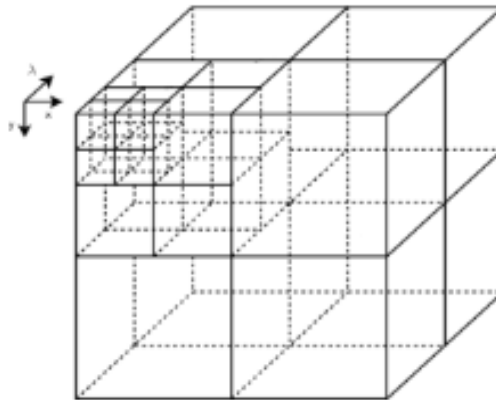


Figure 2.7 : Transformée en ondelettes discrète 3D

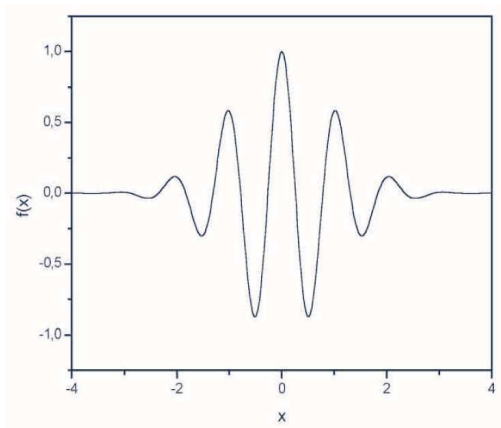
La transformée en ondelettes discrète 3D décompose un volume de façon pyramidale et réversible en huit sous bandes; une sous bande constituée des coefficients basses fréquences et les sept autres bandes contiennent les coefficients hautes fréquences.

2.7. Comparaison entre CWT et DWT

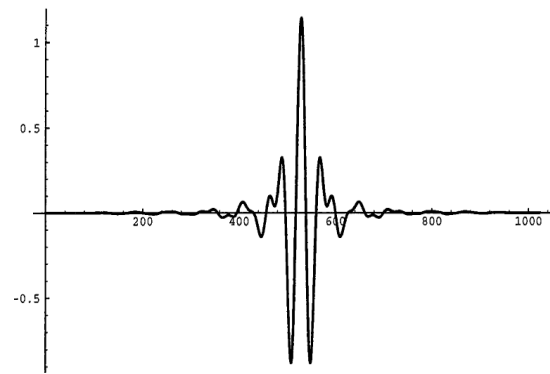
Les sections précédentes ont montré que les transformées discrète et continue produisent des analyses du signal légèrement différentes. La transformée en ondelettes discrète permet une compression plus efficace des données, puisqu'elle minimise la redondance.

2.8. Familles de la transformée en ondelettes

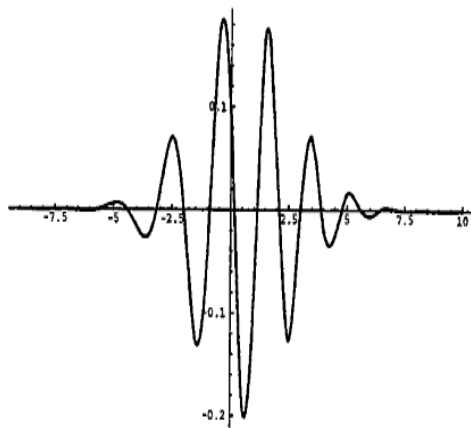
Les différentes familles d'ondelettes sont utilisées selon leurs propriétés en fonction du problème à résoudre. La figure 2.8 présente quelques types de familles d'ondelettes :



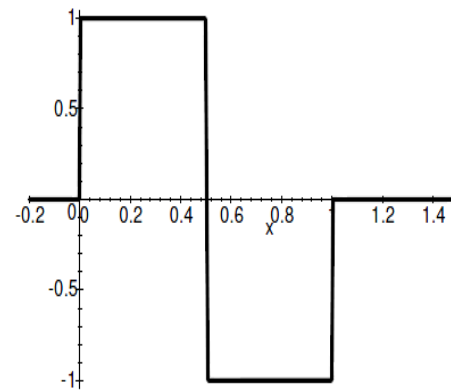
Ondelette de Morlet



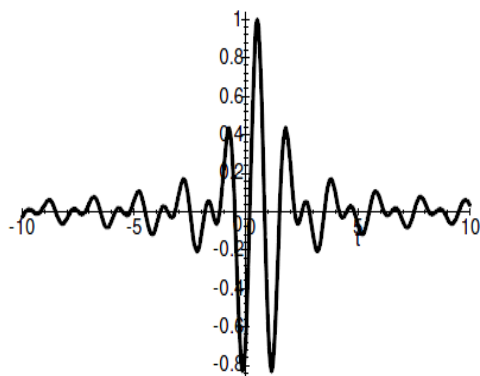
Ondelette de Meyer



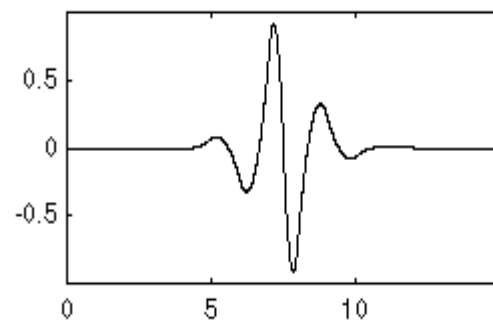
Ondelette de Daubechies



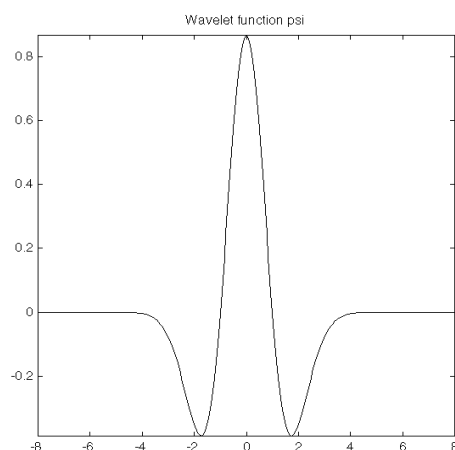
Ondelette de Haar



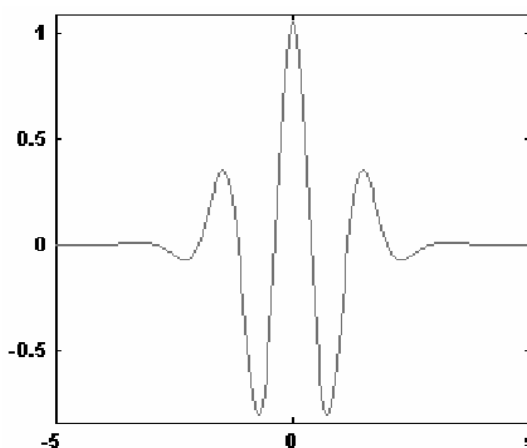
Ondelette de Littlewood Paley



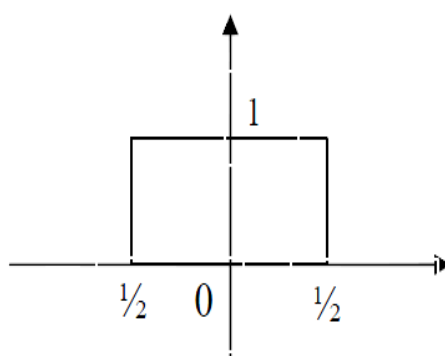
Ondelette Biorthogonal (bior 3.7)



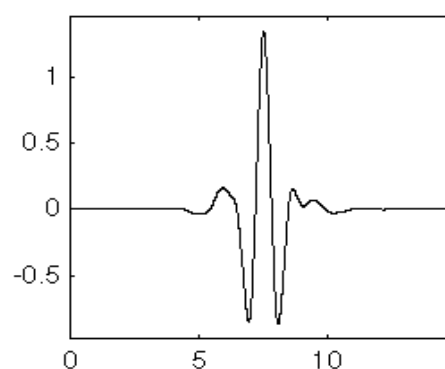
Ondelette de Mexican Hat



Ondelette Gaussienne



Ondelette Splines



Ondelette de Symlets

Figure 2.8 : Quelques types de familles d'ondelettes

2.9. Avantages de la transformée en ondelettes

La transformée en ondelette utilise des fonctions bien localisées dans le plan temps-fréquence. Cette spécificité lui donne beaucoup d'avantages:

- Pour des signaux physiques présentant des variations très rapides, des sauts, des marches ou des discontinuités, l'analyse en ondelettes est mieux adaptée car l'ondelette permet la détection et l'analyse de ces variations. Cette particularité rend l'analyse en ondelettes complémentaire à l'analyse de Fourier. En effet, avec l'analyse de Fourier, les discontinuités d'un signal ne sont pas facilement analysables, car les coefficients des fréquences correspondantes sont étalés dans toute la transformée.

- La transformée en ondelettes peut représenter complètement et efficacement un signal quelconque avec peu de coefficients [Sav04].
- La localisation en temps/fréquence est précieuse pour un bon nombre d'applications.

2.10. Applications des ondelettes

Les ondelettes sont employées dans un grand nombre de domaines d'applications. On peut citer, entre autres, la géophysique, l'astrophysique, les signaux sonores en médecine, l'imagerie dans tous ses aspects et l'imagerie médicale en particulier, l'imagerie satellitaire, le codage des signaux vidéo, la modélisation du trafic dans les réseaux de communication comme Internet, l'analyse de la turbulence atmosphérique ou celle des souffleries et la liste est encore longue. La capacité des ondelettes à faire ressortir les variations d'un signal peut être exploitée de diverses manières.

- En géophysique, les chercheurs de l'Institut de Physique du Globe, à Paris, utilisent les ondelettes pour étudier le minuscule effet sur la vitesse de rotation de la terre [Bou04].
- En imagerie médicale, on exploite le fait que la présence d'erreurs dans une transformée en ondelettes n'altère pas toute la transformée. Les mathématiciens D. M. Healy et D. Coll et le radiologue J. Weaver [Hea92], ont montré qu'avec les ondelettes, ces erreurs sont très réduites.
- En astronomie, les ondelettes servent à étudier la distribution de la matière à grande échelle dans l'univers. Selon l'astrophysicien français Albert Bijaoui [Bij95], une première tâche consiste à faire l'inventaire des galaxies de l'univers. Puisque les ondelettes identifient les structures aux différentes échelles, elles sont capables de distinguer une étoile d'une galaxie, ce qui n'est pas toujours évident.
- En biométrie, on utilise les ondelettes dans un algorithme de reconnaissance du visage, en associant certains points caractéristiques du visage (milieu des yeux, du nez, de la bouche, etc.) avec de nombreuses ondelettes paramétrées afin d'extraire différents coefficients qui pourront ensuite être utilisés pour vérifier la ressemblance entre deux visages [Toh05].
- En sismologie, on tente de détecter les séismes en identifiant de façon automatique les différentes composantes usuellement appelées P et S d'un signal de séisme, court et non stationnaire, en se servant des ondelettes comme outil de reconnaissance de forme [Gua06], [Som10].

- Dans le domaine des sciences biomédicales, les ondelettes entrent dans un processus visant à découvrir un dysfonctionnement, à comparer du normal de l'anormal et à améliorer les informations recueillies pour assurer un travail plus clair : signaux localisés comme des pics en EEG (ElectroEncéphaloGramme) et de signaux diffus comme le bruit de fond cardiaque ou l'écoulement sanguin. Le signal cardiaque ECG se décompose en séquences dont la plus importante est appelée le complexe QRS. Le repérage de forme par ondelettes permet la discrimination de patterns normaux et anormaux [Add00], [Gau12].
- Concernant les nanotechnologies, et plus particulièrement dans le domaine de l'analyse d'hétéro structures en couches minces par réflectométrie et diffraction des rayons X pour l'optoélectronique, les ondelettes permettaient d'effectuer une analyse non destructive de ces matériaux, donnant non seulement l'épaisseur des couches nanométriques mais aussi, dans certains cas, l'ordre d'empilement des couches [Efi11], [Xia08].
- Passons maintenant au débruitage. Les ondelettes fournissent des moyens simples et efficaces pour estimer les signaux en présence de bruit. La procédure est simple, on transforme le signal en ondelettes, on élimine à toutes les résolutions les coefficients supérieurs à une valeur seuil et on reconstruit le signal avec les coefficients qui restent grâce à une transformée inverse en ondelettes. L'énergie du bruit blanc est donc dispersée sur toute la transformée et donne des coefficients petits qu'on élimine. Précisons enfin que le débruitage par ondelettes est très performant pour des signaux constitués à la fois de parties lisses et de zones présentant des irrégularités et des variations brusques [Som10].
- En estimation, les vertus de localisation des ondelettes font un instrument de choix pour certains problèmes d'estimation. L'estimation d'une fonction ou d'une densité sera remplacée par l'estimation de ses coefficients d'ondelettes à une certaine échelle [Wu96].
- Terminons avec la compression. On le décompose d'abord sur une base orthogonale d'ondelettes au moyen de la transformée discrète. Ensuite, on sélectionne une partie des coefficients par seuillage, en conservant intact les coefficients d'approximation de niveau convenablement choisi. Enfin, à partir des coefficients seuillés, on reconstruit le signal original en leur appliquant la transformée discrète inverse. Le nouveau standard de compression d'image est le JPEG 2000, il est basé sur la transformée en ondelettes. Cette fois ci, l'image est analysée de manière globale et la qualité reste très bonne même à un taux de compression élevé. A titre d'exemple, si l'on considère une image originale de 512 x 512 pixels et d'une taille de 256 Ko, il est très difficile de voir la différence avec la même image compressée par ondelettes [Tau04].

2.11. Conclusion

Le chapitre deux a présenté brièvement les notions théoriques de la transformée en ondelettes. On a défini les deux types de transformées en ondelettes, la transformée en ondelettes continue et la transformée en ondelette discrète dans les divers dimensions 1D, 2D et 3D. On a présenté aussi quelques familles utilisées dans les transformées en ondelettes. Aussi on a énuméré quelques domaines d'applications des transformées en ondelettes.

Le prochain chapitre donne un aperçu sur le codage hiérarchique progressif qui exploite les propriétés des transformées en ondelettes discrètes.

Chapitre 3

TECHNIQUES DE CODAGE HIERARCHIQUE PROGRESSIF

Contenu

3.1. Introduction.....	33
3.2. Relation parent-enfants	34
3.3. Codage EZW.....	35
3.3.1. Caractères du codeur.....	36
3.3.2. Compression des coefficients non significatifs.....	37
3.3.3. Quantification par approximations successives.....	37
3.3.4. Etapes de l'algorithme EZW.....	37
3.3.5. Exemple de déroulement du codeur EZW.....	40
3.4. Codage SPIHT.....	42
3.4.1. Etapes de l'algorithme SPIHT.....	44
3.4.2. Exemple de déroulement de SPIHT.....	45
3.5. Conclusion.....	47

CHAPITRE 3

TECHNIQUES DE CODAGE HIERARCHIQUE PROGRESSIF

3.1. Introduction

L'opération des transformées en ondelettes a permis une évolution des opérations de codage selon les propriétés particulières des coefficients. En effet leur organisation sous forme de pyramide multirésolution est particulièrement adaptée à un codage progressif.

Malgré le travail de décorrélation opéré par les transformations, il existe tout de même des dépendances entre les coefficients des différentes sous bandes. De plus au sein même des sous-bandes, les coefficients voisins ne sont pas complètement indépendants. C'est pourquoi, de nombreux travaux ont porté sur la définition de schémas de compression tirant partie de ces dépendances.

Bien que n'importe quel algorithme de compression puisse être utilisé pour compresser un signal (décomposé en sous-bandes) après sa transformation en ondelettes, certains algorithmes tirent mieux profit de la décomposition en sous-bandes d'un signal que d'autres. L'algorithme EZW (Embedded Zerotree Wavelets) est la base de nombreux autres algorithmes, il a été utilisé pour réaliser la compression par ondelette. Les approches basées sur la similarité des coefficients d'ondelette suivant les différentes échelles sont toutes inspirées de la méthode EZW.

Plusieurs algorithmes basées sur les transformées en ondelettes discrètes existent. On peut citer EZW [Sha93], SPIHT (Set Partitioning In Hierarchical Trees) [Pea98], SFQ (Space-Frequency Quantization) [Xio98], EBCOT (Embedded Block Coding with Optimized Truncation) [Tau02], et autres codages d'images utilisant des paquets d'ondelettes avec la quantification vectorielle [Sah00], [Ant92]. Dans ce chapitre, on va décrire seulement les deux algorithmes de progressivité utilisés dans notre travail : EZW et SPIHT. Ces algorithmes consistent à exploiter les dépendances statistiques existantes entre

les coefficients de mêmes localisations spatiales et orientations, mais appartenant à des sous-bandes de résolutions différentes.

Les techniques EZW et SPIHT ont été appliquées initialement aux images. C'est pour cela on va décrire ces algorithmes pour ce type d'application.

3.2. Relation parent-enfants

Suite à la transformation d'une image par ondelettes discrètes, il n'y a pas directement de compression de la taille de l'image. En effet, une image de 640*480 pixels sur 8 bits donne, après traitement par la DWT, 640*480 coefficients sur 8 bits, répartis entre les coefficients de détails (CD) et les coefficients d'approximations (CA). Comme dans tout encodage par transformation, le calcul de la transformée du signal à compresser est l'une des étapes de traitement en vue de la compression. La transformation vise principalement à assurer la décorrélation des pixels voisins pour permettre une quantification scalaire plus performante. La figure 3.1 montre une décomposition utilisant l'algorithme de Mallat sur 3 niveaux.

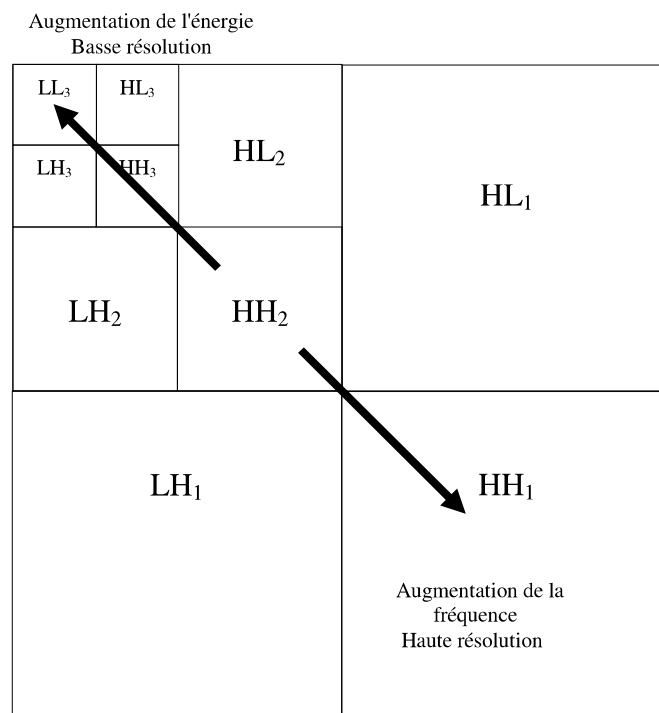


Figure 3.1 : Répartition de l'information après une décomposition 3 niveaux

Dans ce cas, on trouve une série de coefficients d'approximations LL_3 , trois séries de HL : HL_1 , HL_2 , HL_3 , trois séries de LH : LH_1 , LH_2 , LH_3 et trois séries de HH : HH_1 , HH_2 , HH_3 . Dans ces séries de coefficients de même nature, les coefficients descendent les uns

des autres, comme indiqué dans la figure 3.2. Prenons par exemple les coefficients HH, un coefficient HH dans le plus niveau de décomposition HH_3 possède quatre descendants (enfants) dans HH_2 . Chaque coefficient de HH_2 possède quatre enfants dans HH_1 . L'algorithme EZW cherche à exploiter ces relations de descendance entre les coefficients.

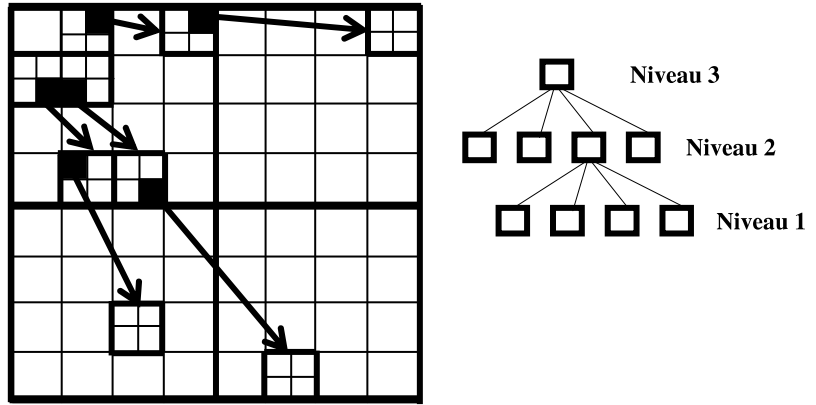


Figure 3.2 : Relations entre les coefficients d'ondelette dans différentes sous-bandes

Ainsi, lors du balayage de l'image, tous les coefficients d'un niveau sont balayés avant de passer au niveau suivant; ceci donne lieu au balayage en Z de la figure 3.3.

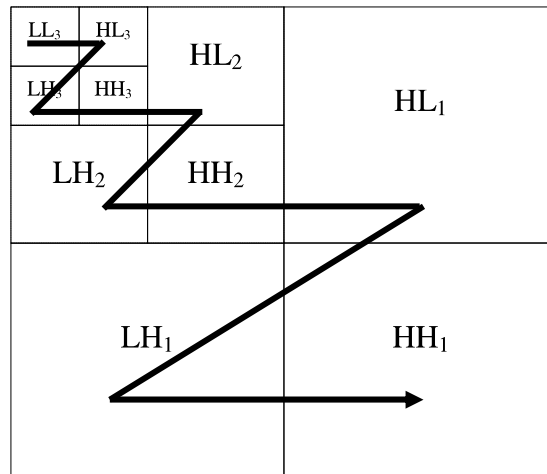


Figure 3.3 : Balayage en Z de l'image

3.3. Codage EZW

L'abréviation EZW signifie encapsulation d'ondelettes par arbres de zéros. C'est elle qui est généralement utilisée sous des noms divers dans la plupart des implémentations de moteurs de compression par ondelettes. En effet la technique EZW possède une propriété

extrêmement intéressante : elle code l'information par ordre exact dans le signal, les coefficients sont codés du plus significatif au moins significatif. L'intérêt est évident : pour obtenir une compression sans perte, on code tous les coefficients et pour obtenir une compression avec pertes, il suffit d'interrompre le décodage à l'instant souhaité en fonction des pertes admises.

Comme les autres méthodes de quantification, l'approche EZW n'opère que sur les sous-bandes de la transformée, sans modifier le signal basse résolution. Elle consiste à effectuer plusieurs passages successifs dans le signal transformé en utilisant un seuillage de plus en plus bas. A chaque passage, un coefficient lu n'est codé que s'il est significatif (par rapport à la valeur de seuil), auquel cas il est remplacé par sa différence avec le seuil. L'avantage est que les seuils sont prédéterminés. Par exemple si les sous bandes sont codées sur 256 valeurs -typiquement des entiers de 0 à 255- on effectuera des passages successifs avec 128, 64, 32,... comme valeurs de seuils.

Le codage d'une image utilisant la procédure EZW, ainsi que quelques optimisations ont comme conséquence un compresseur d'images remarquablement efficace. Le flux de données comprimé peut avoir n'importe quel débit binaire désiré [Sha93].

3.3.1. Caractères du codeur

L'algorithme EZW contient les caractères suivants :

- ✓ Une transformation d'ondelette discrète qui produit une représentation multirésolution.
- ✓ Le codage EZW fournit une représentation compacte. Ainsi une table de signification (vecteur des éléments significatifs), indique les positions des coefficients significatifs à travers les échelles.
- ✓ Un protocole de priorité par lequel l'ordre d'importance est déterminé. On ordonne par précision, magnitude, échelle et localisation spatiale les coefficients d'ondelettes. Remarquons en particulier que les plus grands coefficients sont jugés plus importants que les plus petits coefficients sans considérer leur échelle.
- ✓ Le codage d'arithmétique fournit une méthode rapide et efficace pour les chaînes de codage entropiques des symboles et ne nécessite pas de formation ou des tables préétablies [Bea08].

Cet algorithme repose sur deux idées majeures : la compression de la carte des coefficients significatifs basée sur le principe des "zerotree" et le principe de quantification par approximations successives.

3.3.2. Compression des coefficients non significatifs

Etant donné que la structure de "zerotree" est composée d'un ancêtre et de descendants. L'ancêtre (père) est le coefficient négligeable, c'est la structure appartenant à l'échelle la plus grossière. Les descendants sont tous les coefficients négligeables de même direction et de même localisation que l'ancêtre appartenant aux échelles plus fines dans cet arbre.

Chaque père à l'échelle ' i ' possède quatre fils à l'échelle ' $i-1$ '. Si un ancêtre et tous ses descendants sont négligeables alors on attribue à l'ancêtre le code "zerotree". Au décodage, lorsqu'on rencontre ce symbole, on peut conclure que le coefficient et tous ses descendants sont négligeables. Lorsqu'un ancêtre est négligeable, cela n'entraîne pas que tous ses descendants soient négligeables. En pratique, sur des signaux réels, la probabilité que ce phénomène survienne est très importante. La structure des "zerotree" permet de coder la carte des coefficients par un très faible nombre de bits.

3.3.3. Quantification par approximations successives

L'idée est d'envoyer en premier lieu, l'information la plus significative possible. Pour cela, nous émettons en premier les données qui réduisent le plus le carré de l'erreur moyenne (acronyme anglais MSE) entre le signal d'origine et le signal reconstruit. Les coefficients d'ondelettes à transmettre pour réduire au maximum le MSE sont ceux de norme la plus grande. La quantification par approximations successives minimise de manière optimale le MSE pour chaque bit émis. Si, par exemple, nous devons envoyer uniquement un bit du signal représenté par le code binaire P égal à 01001101. Nous choisirons de transmettre le deuxième bit en partant de la gauche car c'est le bit porteur du plus d'informations.

3.3.4. Etapes de l'algorithme EZW

Expliquons maintenant comment J. Shapiro a développé un algorithme respectant les concepts énoncés ci-dessus [Sha93]. Le principe consiste à lancer successivement l'algorithme pour des valeurs de seuils décroissantes $T_0, T_1, \dots$. Nous prenons comme valeur de seuil la plus élevée :

$$T_0 = 2^{\lfloor \log_2 c_{\max} \rfloor} \quad (3.1)$$

Où $c_{\max}$ représente le plus grand coefficient d'ondelette. De plus le seuil T_i à l'itération i est égal à $\frac{T_{i-1}}{2}$. A la première étape le seuil est très grand et il y a très peu de coefficients d'ondelette significatifs. La représentation à l'aide des "zerotree" de la carte des coefficients significatifs est très intéressante car très compacte dans ce cas [Bea08].

La figure 3.4 décrit les étapes de l'algorithme EZW :

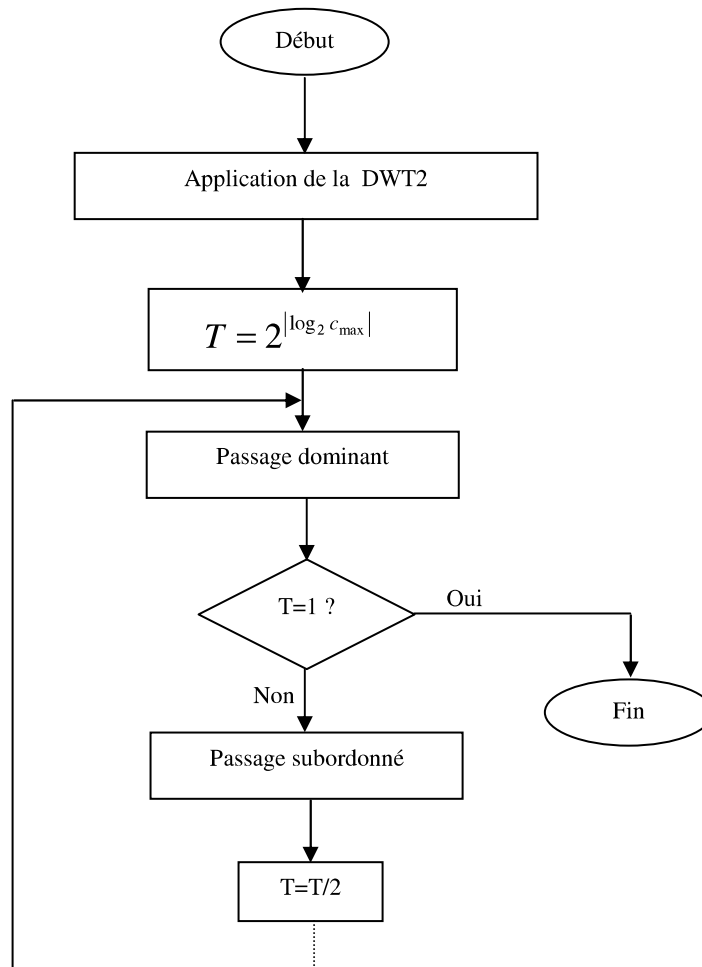


Figure 3.4 : Organigramme de la technique EZW

- **Initialisation**

On applique en premier lieu la transformée en ondelettes à l'image. Le seuil T_0 est calculé de telle sorte qu'il soit égal à la plus grande puissance de deux inférieure à la valeur maximale des coefficients d'ondelettes, il est calculé utilisant l'équation (3.1).

- **Passage dominant**

Dans cette étape, on parcourt les coefficients d'ondelettes selon le chemin représenté sur la figure 3.3. Que les "zerotree" soient les plus efficaces possibles. Pour chaque coefficient on affecte un des quatre symboles suivants en les comparant au seuil courant T_i (figure 3.5):

PS (positif et significatif) : si la valeur absolue du coefficient est au-dessus du seuil T_i et de signe positif.

NS (négatif et significatif) : si la valeur absolue du coefficient est au-dessus du seuil T_i et de signe négatif.

IZ (zéro isolé) : si la valeur absolue du coefficient est inférieure au seuil T_i possédant un ou plusieurs descendants qui ne sont pas négligeables devant T_i .

ZTR (zerotree) : tous les coefficients descendants d'un coefficient qui ont le symbole ZTR ne seront pas codés puisque tous les descendants sont négligeables.

On ne transmettra pour cette étape seulement les quatre symboles cités ci-dessus.

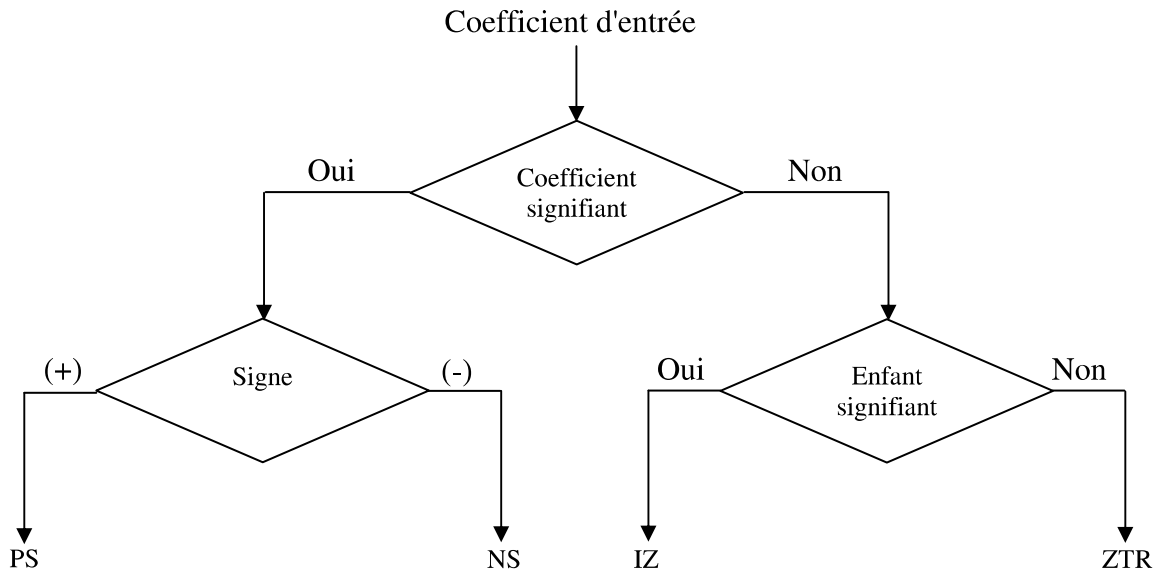


Figure 3.5 : Affectation des symboles pour les coefficients

Tous les coefficients significatifs détectés dans cette étape à l'itération i seront insérés en valeurs absolues dans la liste des subordonnées.

- **Passage subordonné (codage par '1' et '0')**

Pour toutes les valeurs significatives déterminées dans l'étape précédente, on doit émettre le bit correspondant pour augmenter la précision des valeurs significatives transmises.

Cette étape (passage subordonné) est basée principalement sur les valeurs significatives qui ont été codées par 'PS' et 'NS'. Ainsi, cette étape comprend les points suivants :

- Calculer les intervalles d'incertitudes $[a_j, b_j[$, tel que $j=0,1,\dots$ avec $a_j < 2T_0$

$$a_j = b_{j-1} \text{ avec } a_0 = T_i \quad (3.2)$$

$$b_j = a_j + T_i \quad (3.3)$$

- les valeurs qui appartiennent à un intervalle $[a_j, m[$ seront codées par "0". Tel que $m = \frac{3}{2} a_j$. m représente la limite de décision.
- les valeurs qui appartiennent à un intervalle $[m, b_j[$ seront codées par "1".

On recommence l'algorithme à l'itération 2 sur le résidu du signal en incrémentant i de un et en divisant le seuil T_i par deux. Nous itérons le processus jusqu'à ce que le critère de qualité du signal soit atteint ou que le nombre de bits transférables soit dépassé.

3.3.5. Exemple de déroulement du codeur EZW

Pour comprendre les subtilités de l'algorithme EZW, il est utile d'appliquer à la main cet algorithme sur des données simples pour faciliter sa compréhension. On applique ici le premier passage de cet algorithme utilisant l'exemple original de l'article [Sha93]. Il est utilisé également comme application de la technique SPIHT. Rappelons que le codage est réalisé sur des plans de bits successifs [Chr06].

Dans cette section, un exemple simple sera employé pour mettre en évidence l'ordre d'opérations employées dans l'algorithme EZW.

On suppose qu'après transformée en ondelettes d'un signal à 2D 8x8 en trois niveaux de décomposition, on aura les coefficients de la figure 3.6.

Rappelons qu'un coefficient est déclaré significatif (symbole PS ou NS suivant son signe) si sa valeur absolue est supérieure au seuil T_i . Sinon, soit il s'agit d'un zéro isolé (IZ) si au moins un de ses descendants est significatif, soit d'un arbre de zéros (ZTR) si aucun de ses descendants n'est significatif. Le classement des coefficients de la figure 3.6 est donné dans le tableau 3.1.

63	-34	49	10	07	13	-12	07
-31	23	14	-13	03	04	06	-01
15	14	03	-12	05	-07	03	09
-09	-04	-14	08	04	-02	03	02
-05	09	-01	47	04	06	-02	02
03	00	-03	02	03	-02	00	04
02	-03	06	-04	03	06	03	06
05	11	05	06	00	03	-04	04

Figure 3.6 : Exemple de coefficients d'une image 8x8 à 3 niveaux

Sous-bande	Valeur du coefficient	Symbole
LL ₃	63 (0,0)	PS
HL ₃	-34 (0,1)	NS
LH ₃	-31 (1,0)	IZ
HH ₃	23 (1,1)	ZTR
HL ₂	49 (0,2)	PS
HL ₂	10 (0,3)	ZTR
HL ₂	14 (1,2)	ZTR
HL ₂	-13 (1,3)	ZTR
LH ₂	15 (2,0)	ZTR
LH ₂	14 (2,1)	IZ
LH ₂	-9 (3,0)	ZTR
LH ₂	-4 (3,1)	ZTR
HL ₁	7 (0,4)	IZ
HL ₁	13 (0,5)	IZ
HL ₁	3 (1,4)	IZ
HL ₁	4 (1,5)	IZ
HL ₁	-1 (4,2)	IZ
HL ₁	47 (4,3)	PS
HL ₁	-3 (5,2)	IZ
HL ₁	-2 (5,3)	IZ

Tableau 3.1 : Seuillage des coefficients par rapport au seuil T=32

1. Le premier coefficient à la valeur (63), il est supérieur au seuil courant (32), le symbole PS est émis.
2. (-31) est non significatif, mais il possède un descendant qui est significatif (47) c'est un zéro isolé, IZ est émis.

3. (23) est non significatif et tous ses descendants sont non significatifs, il s'agit donc d'un arbre de zéros, ZTR est émis.
4. (7) est non significatif et comme les sous-bandes HL_1 n'a pas de descendant, les symboles IZ et ZTR peuvent être regroupés en un seul symbole.
5. (47) est significatif, le symbole PS est émis. Comme ce symbole sera traité pendant les prochains passages dominants, il doit être considéré comme étant nul pour les passages suivants (le symbole 14 de la LH_2 sera alors une racine d'un arbre de zéros).

Les ensembles $D_1...D_6$ ci-dessous représentent le résultat des passages dominants pour toutes les itérations. Le symbole PS a été remplacé par "P", NS par "N", IZ par "Z" et ZTR par "T". Les ensembles $S_1...S_5$ ci-dessous représentent le résultat des passages subordonnés pour toutes les itérations.

D_1 : PNZTPTTTTZTTTTTTPTT

S_1 : 1010

D_2 : ZTNPTTTTTTTT

S_2 : 100110

D_3 : ZZZZPPNPPNTTNNPTPTTNTTTTTTTTTPTTTPTTTTTTTTTTTTTTTTTTTTTTTT

S_3 : 10011101111011011000

D_4 : ZZZZZZZTZNZZZZPTTPTPTPNPTNTTTTTPTPNPPPPTTTTTPTPTTTPNP

S_4 : 11011111011001000001110110100010010101100

D_5 : ZZZZZTZZZZZTPZZZTPTTTTNTPTPTTPTTNPNTTTTTPNNTPTTPTPTT

S_5 : 10111100110100010111110101101100100000000110110110011000111

D_6 : TTTTTTTTTTTTTTTNNTTT

3.4. Codage SPIHT

L'approche SPIHT est une généralisation de l'EZW. On rassemble les coefficients nécessaires au codage par ondelettes dans un arbre. Cet arbre est divisée en sous-arbres, qui sont organisés de manière à garder les coefficients importants (au sens de ceux qui sont à la base de reconstruction du signal) ensemble.

Les coefficients d'ondelettes sont d'abord divisés en arbre commençant à la bande de résolution la plus basse. Pour les bandes de résolutions plus élevées, les coefficients codés sont en quelque sorte les coefficients complémentaires (en terme de qualité) à ceux déjà encodés précédemment. Si on définit un niveau de seuil, on peut dire qu'un coefficient est

important s'il possède une amplitude plus grande que ce seuil. La valeur initiale du seuil est celle donnée par l'équation (3.1).

On construit alors une liste de bits significatifs et une liste de bits insignifiants. A chaque passage on code les bits significatifs. Les bits insignifiants deviennent les bits de départ de la seconde passe pour lesquels un nouveau calcul de seuil est appliqué et ainsi de suite par récurrence jusqu'à certain nombre maximale de passes fixant un certain rapport de compression ou un certain taux de distorsion [Chr06].

L'algorithme SPIHT utilise trois structures de données :

- ✓ La liste des points significatifs (LSP, List of Significant Pixels), qu'on peut noter PixelSignif.
- ✓ La liste des points non significatifs (LIP, List of Insignificant Pixels), nommée PixelNonSignif.
- ✓ La liste des ensembles non significatifs (LIS, List of Insignificant Sets). Elle-même est subdivisée en deux parties de type A et de type B, on note DescNonSignif pour le type A (tous les descendants ne sont pas significatifs "arbre de degré 1") et PetitDescNonSignif pour le type B (tous les descendants, à l'exception d'au moins un des enfants, ne sont pas significatifs "arbre de degré 2").

Chaque liste contient un sous ensemble de coefficients de l'image de façon formelle, la présence d'un coefficient donné dans un ensemble est équivalent à la satisfaction d'un prédicat de la carte de signifiante, tel que tous les enfants du coefficient sont non significatifs. Les ensembles et les prédicats de SPIHT sont choisies de telle sorte à améliorer la compression mais dans l'idée ils correspondent à une version plus élaborée de la répartition suivante : la carte de signifiante $S(T)$ est subdivisée en ensemble d'arbre de zéros, et pour tous les coefficients en dehors de tels arbres : en ensembles de points significatifs et de points non significatifs. A une itération donnée, l'ensemble des points contenus dans ces listes définit donc la signifiante $S(T)$ ainsi :

Les points dans PixelSignif et dans PixelNonSignif sont des points individuels, dont l'algorithme sait de façon certaine s'ils sont dans la carte de signifiante ou non.

Les points dans DescNonSignif sont des points dont tous les descendants sont non significatifs, c'est à dire précisément des arbres de zéros dont les sommets sont exclus.

Les pixels dans PetitDescNonSignif sont des points dont tous les enfants directs peuvent être ou ne pas être significatifs, mais dont les petits enfants et leurs descendants sont tous non significatifs.

On peut voir que ces listes correspondent à différentes structures dérivées des autres arbres de zéros : arbres de zéros tronqués au-dessus des descendants directs ou au-dessus des descendants de deuxième génération.

L'algorithme est compliqué par le fait qu'il s'agit en réalité d'une amélioration et optimisation de l'EZW, qui utilise uniquement des arbres de zéros avec une performance moindre. Ici il est clairement visible que les sous-arbres de zéros, DescNonSignif et PetitDescNonSignif, sont indépendants de la valeur de signifiante du coefficient qui lui est la racine.

On considère un coefficient de coordonnées x et y dans l'image transformée I . la transformation d'ondelettes est décomposée en k niveaux par exemple, de sorte que l'image transformée sera constituée des bandes $LL_k, HL_k, LH_k, HH_k, \dots, HL_2, LH_2, HH_2, HL_1, LH_1$ et HH_1 . Pour $k=3$, on aura la décomposition de la figure 3.1.

3.4.1. Etapes de l'algorithme SPIHT

Les ensembles suivants sont utilisés pour présenter l'algorithme SPIHT [Pea98]:

- ✓ $O(i, j)$: ensemble des enfants du noeud (i, j) (un seul niveau de descendance).
- ✓ $D(i, j)$: ensemble de tous les descendants du noeud (i, j) .
- ✓ $L(i, j) = D(i, j) - O(i, j)$: ensemble des descendants à l'exception des enfants.
- ✓ $S(i, j)=0$ si tous les descendants du noeud (i, j) sont en dessous du seuil T_{ik} (arbre de zéros) et $S(i, j)=1$ dans le cas contraire.
- ✓ $H(i, j)$: ensemble de coordonnées de toutes les orientations spatiales de l'arborescence (noeuds du niveau hiérarchique élevé).

L'algorithme SPIHT, s'écrit comme suit avec les notations déjà citées :

1. Initialisation

Calculer $n=\log_2(\max_{(i,j)}\{c_{i,j}\})$, mettre LSP à vide

Ajouter les $H(i, j)$ dans LIP, ou LIS comme entrée type D.

2. Premier passage

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

2.1.1. Sortir $S_n(i, j)$

2.1.2. Si $S_n(i, j)=1$, mettre (i, j) dans LSP et sortir signe $c_{i, j}$

2.2. Pour chaque entrée de LIS faire

2.2.1. Si l'entrée est de type D alors :

- Sortir $S_n(D(i,j))$
- Si $S_n(D(i,j))=1$ alors :
 - Pour chaque $(k,l) \in O(i,j)$ faire :
 - Sortir $S_n(k,l)$
 - Si $S_n(k,l)=1$, ajouter (k,l) dans LSP et sortir signe $c_{k,l}$
 - Si $S_n(k,l)=0$, ajouter (k,l) à la fin de LIP
 - Si $L(i,j) \neq 0$, Déplacer (i,j) à la fin de LIS avec le type L, aller à 2.2.2.
 - Si $L(i,j)=0$, Supprimer (i,j) de LIS

2.2.2. Si l'entrée est de type L alors :

- Sortir $S_n(L(i,j))$
- Si $S_n(L(i,j))=1$ alors :
 - Ajouter les $(k,l) \in O(i,j)$ à la fin de LIS en type D
 - Supprimer (i,j) de LIS

3. Deuxième passage

Pour chaque (i,j) de LSP, sauf ceux qui viennent d'être inclus dans la dernier passage, sortir le $n^{\text{ème}}$ bit significatif des coefficients d'ondelettes $c_{i,j}$

4. Décrémenter n de 1 et retourner en 2.

3.4.2. Exemple de déroulement de SPIHT

Quelques précisions sont données pour certaines étapes utilisant l'exemple de la figure 3.6 [Chr06]. On résume les résultats pour le seuil 32 dans le tableau 3.2.

1. Initialisation des listes: la liste 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.
2. (0,0) est significatif, il est mis dans la liste LSP, le signe de cette valeur est également émis, il est noté +.
3. On teste donc les quatre enfants de (0,1), un des descendants de (0,1) est significatif (49).
4. Comme un des enfants de (0,1) était significatif, il est classé de type B et mis à la fin de la liste LIS.
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. (0,1) est de type B, on vérifie alors L(0,1) qui n'est pas significatif.
7. (1,0) est de type B, on vérifie donc L(1,0) qui est significatif (valeur 47), on partage donc l'ensemble et on ajoute (2,0), (2,1), (3,0), (3,1) à la fin de la LIS comme éléments de type A et on enlève (1,0) de la LIS.
8. Comme (2,1) n'a pas de petits enfants, au lieu de le passer en élément de type B comme on aurait dû le faire, on le retire de la liste.
9. Ces listes constituent les listes d'entrées pour le passage suivant où le seuil aura une valeur double "16".

Coefficients ou ensembles testés	Sorties	Action	Listes	Etape
			LIS={ (0,1)A,(1,0)A,(1,1)A } LIP={ (0,0),(0,1),(1,0),(1,1) } LSP=∅	(1)
(0,0)	1+	(0,0) dans LSP	LIP={ (0,1),(1,0),(1,1) } LSP={ (0,0) }	(2)
(0,1)	1-	(0,1) dans LSP	LIP={ (1,0),(1,1) } LSP={ (0,0),(0,1) }	
(1,0)	0	Rien		
(1,1)	0	Rien		
D(0,1)	1	Tester les descendants	LIS={ (0,1)A,(1,0)A,(1,1)A }	(3)
(0,2)	1+	(0,2) dans LSP	LSP={ (0,0),(0,1),(0,2) }	
(0,3)	0	(0,3) dans LIP	LIP={ (1,0),(1,1),(0,3) }	
(1,2)	0	(1,2) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2) }	
(1,3)	0	(1,3) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2),(1,3) }	
		Chgt type	LIS={ (1,0)A,(1,1)A,(0,1)B }	(4)
D(1,0)	1	Tester les descendants	LIS={ (1,0)A,(1,1)A,(0,1)B }	(5)
(2,0)	0	(2,0) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2),(1,3),(2,0) }	
(2,1)	0	(2,1) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1) }	
(3,0)	0	(3,0) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0) }	
(3,1)	0	(3,1) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1) }	
		Chgt type	LIS={ (1,1)A,(0,1)B,(1,0)B }	
D(1,1)	0	Rien	LIS={ (1,1)A,(0,1)B,(1,0)B }	(6)
L(0,1)	0	Rien	LIS={ (1,1)A,(0,1)B,(1,0)B }	(7)
L(1,0)	1	Ajouter des ensembles	LIS={ (1,1)A,(0,1)B,(2,0)A,(2,1)A,(3,0)A,(3,1)A }	(8)
D(2,0)	0	Rien	LIS={ (1,1)A,(0,1)B,(2,0)A,(2,1)A,(3,0)A,(3,1)A }	(9)
D(2,1)	1	Tester les descendants	LIS={ (1,1)A,(0,1)B,(2,0)A,(2,1)A,(3,0)A,(3,1)A }	(10)
(4,2)	0	(4,2) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2) }	
(4,3)	1+	(4,3) dans LSP	LSP={ (0,0),(0,1),(0,2),(4,3) }	
(5,2)	0	(5,2) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2) }	
(5,3)	0	(5,3) dans LIP	LIP={ (1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2),(5,3) }	
		Enlever (2,1)	LIS={ (1,1)A,(0,1)B,(2,0)A,(3,0)A,(3,1)A }	(11)
D(3,0)	0	Rien	LIS={ (1,1)A,(0,1)B,(2,0)A,(3,0)A,(3,1)A }	(12)
D(3,1)	0	Rien	LIS={ (1,1)A,(0,1)B,(2,0)A,(3,0)A,(3,1)A }	
			LIS={ (1,1)A,(0,1)B,(2,0)A,(3,0)A,(3,1)A } LIP={ (1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2),(5,3) } LSP={ (0,0),(0,1),(0,2),(4,3) }	(13)

Tableau 3.2 : Seuillage des coefficients par rapport au seuil T=32

3.5. Conclusion

Dans ce chapitre, on a montré l'exploitation des dépendances parents-enfants existantes entre les coefficients d'ondelettes dans les différentes sous bandes. Les algorithmes EZW et SPIHT sont des codeurs hiérarchiques progressifs basées sur les structures d'arbres de zéros. Pratiquement, ces techniques permettent d'atteindre des débits élevés. Ces deux méthodes peuvent être adaptées aux signaux unidimensionnels ECG et PCG. Ces derniers font l'objet du chapitre suivant.

Chapitre 4

GENERALITES SUR LES SIGNAUX ECG ET PCG

Contenu

4.1 Introduction.....	48
4.2. Cœur humain.....	48
4.3. Activité électrique du cœur.....	49
4.3.1. Principe de l'électrocardiogramme.....	50
4.3.1.1. Ondes de l'ECG	51
4.3.1.2. Intervalles et segments de l'ECG.....	51
4.3.2. Artefacts de l'enregistrement ECG	52
4.4. Phonocardiogramme.....	54
4.4.1. Sons cardiaques.....	55
4.4.2. Souffles cardiaques.....	56
4.5. Conclusion	57

CHAPITRE 4

GENERALITES SUR LES SIGNAUX ECG ET PCG

4.1. Introduction

Nos applications sont basées sur la compression des signaux ECG et PCG. C'est pour cela on va présenter dans ce chapitre des généralités sur l'électrocardiogramme et le phonocardiogramme avec une brève description du système cardiovasculaire.

4.2. Cœur humain

Le cœur est l'élément principal du système cardiovasculaire. Dans le corps humain, le cœur se situe dans la poitrine entre les poumons, son poids est entre 200 et 450 grammes. Chaque jour il bat en moyenne 100000 fois pompant environ 7600 litres de sang au corps. C'est un muscle appelé myocarde, d'un volume de poignet qui sert à propulser le sang vers tout le corps humain à travers les vaisseaux sanguins. Ces derniers sont décomposés en deux grands systèmes, le système artériel et le système veineux (figure 4.1). Le premier système assure l'acheminement du sang oxygéné vers les organes et le deuxième système sert à ramener le sang pauvre en oxygène au cœur [Kha09].

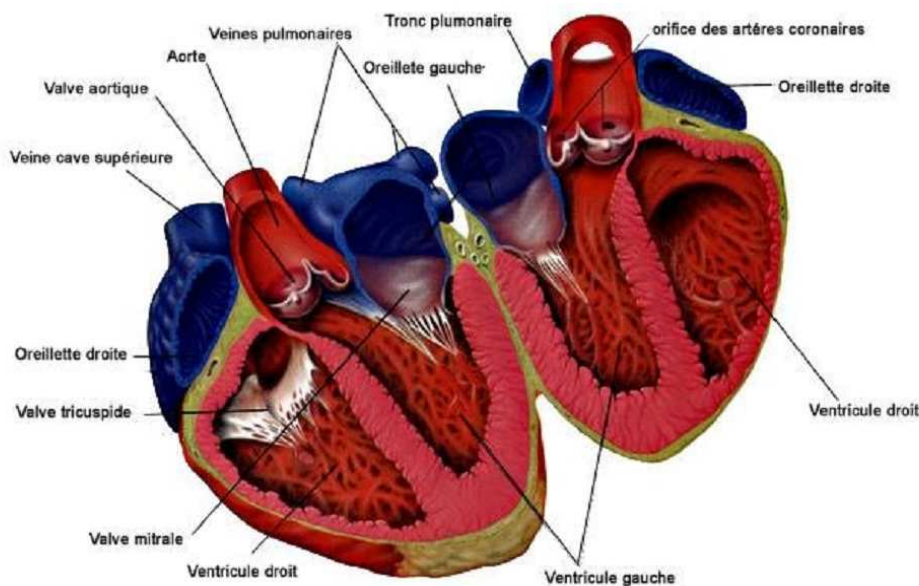


Figure 4.1: Anatomie du cœur humain

Le sang se propulse de la circulation systématique vers l'oreillette droite. En premier lieu, il passe vers le ventricule droit puis vers les poumons à travers les veines et les artères pulmonaires. Le sang riche d'oxygène revient des poumons vers l'oreillette gauche, puis vers le ventricule gauche qui le propulse vers les organes "circulation systématique" à travers la valve aortique [Kha06].

4.3. Activité électrique du cœur

Les scientifiques ont remarqués, depuis la première découverte, l'importance d'exploiter l'activité électrique du cœur dans le domaine du diagnostic pathologique au niveau du myocarde et qu'il peut être la source d'une foule de renseignements précieux de l'état structural et fonctionnel de l'appareil cardio-circulatoire, d'où la nécessité d'un outil permettant l'exploration et l'enregistrement de cette activité électrique, afin de l'exploiter aisément [Kha09].

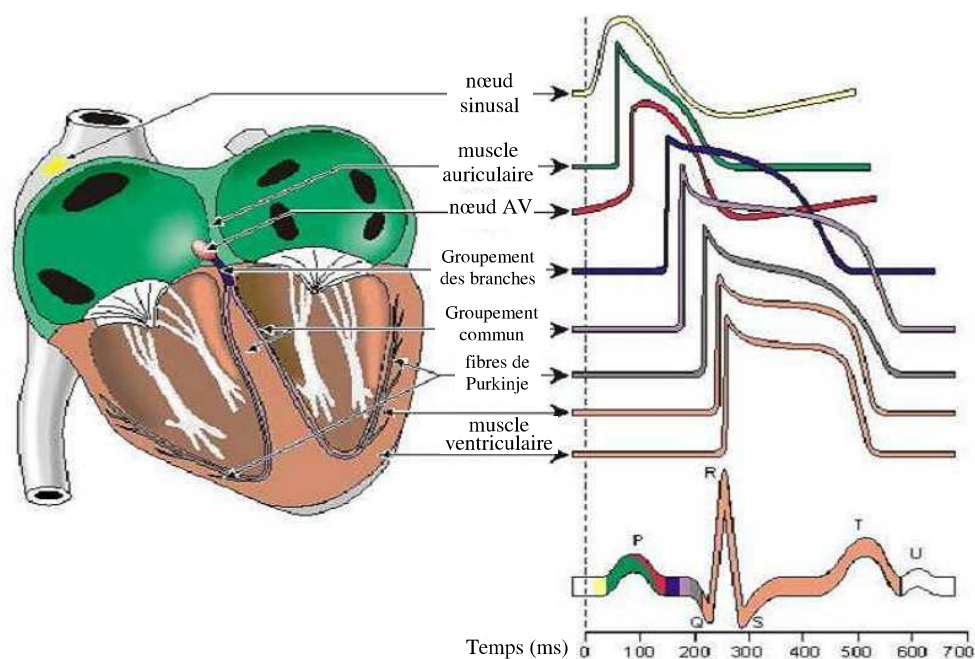


Figure 4.2 : Formes de potentiels d'action dans les différentes parties du cœur et séquence temporelle de la propagation de l'activité électrique cardiaque ; analogie avec les ondes de l'ECG

Parmi les systèmes de dérivations ECG, on rencontre :

- Les dérivations orthogonales corrigées ou bien le système de dérivations de Frank : sept électrodes placées sur la poitrine, le dos, le cou et la jambe gauche sont utilisées pour visualiser le cœur à partir du côté gauche X, le côté du dessous Y et de face Z.

- Le système de projection du potentiel sur la surface du corps : c'est une extension des systèmes conventionnels par l'augmentation du nombre d'électrodes sur la surface du corps (figure 4.2) [Ala07].

4.3.1. Principe de l'électrocardiogramme

L'électrocardiogramme est une projection graphique de l'activité électrique du cœur, c'est une image électrique de l'activité cardiaque. L'activité électrique est captée par des électrodes placées à la surface des téguments. Chaque électrode capte les ondes d'activation selon le plan du cœur qu'elle explore. Le tracé est effectué sur un papier millimétré et quadrillé.

L'électrocardiogramme est ainsi l'enregistrement sur papier des courants d'actions cardiaques transmis à la surface du corps. On utilise deux électrodes de contact reliées par un fil à un galvanomètre mesurant l'intensité des courants électriques pour constituer une dérivation. Par convention, le tracé utilise en abscisse l'échelle de temps qui correspond à la vitesse de déroulement du papier et en ordonnée le voltage. Grâce au quadrillage, on appréciera l'amplitude des ondes enregistrées d'une part en durée, d'autre part en intensité. Ce quadrillage est par convention d'un millimètre sur un millimètre avec un trait renforcé tous les 5 mm :

1mm (1 petit carreau) horizontal correspond à 1 millivolt.

1mm (1 petit carreau) vertical correspond pour un déroulement de 25 mm/sec du papier à 0.04 seconde (soit 0.2 seconde par trait renforcé).

Un enregistrement ECG normal représente un cycle cardiaque normal. Il est caractérisé par un spectre de fréquences distribuées au-dessous de 40-50 Hz et par des ondes et des intervalles de temps qui se répètent d'une manière quasi périodique (figure 4.3) [Kun82].

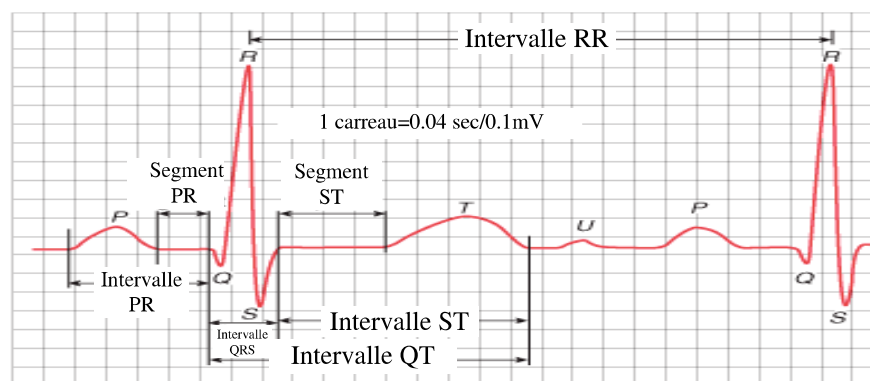


Figure 4.3: Enregistrement ECG normal : différents ondes, intervalles et segments

4.3.1.1. Ondes de l'ECG

- Onde P

La dépolarisation des oreillettes est aperçue comme l'onde P de l'ECG. Elle a une durée de 0.1 à 0.12 secondes et une amplitude de 0.1 à 0.3 mV. La fréquence de l'onde P est entre 10 et 15 Hz [Ham04].

- Complexe QRS

La première dépolarisation ventriculaire correspond au complexe QRS, il est constitué par trois ondes : l'onde R qui est l'onde positive, l'onde Q qui est la première onde négative précédant l'onde R et enfin l'onde S est la première onde négative après R. Le spectre de fréquences du complexe QRS se trouve entre 10 et 50 Hz [Ben09].

- Onde T

L'onde T exprime la ré-polarisation des ventricules, elle est toujours positive.

- Onde U

Après l'onde T, un enregistrement ECG peut parfois présenter une onde de très basse amplitude dite l'onde U, elle est similaire dans la forme à l'onde P et elle est souvent associée aux processus de ré-polarisation ventriculaire tardive [Cab08].

4.3.1.2. Intervalles et segments de l'ECG

Parmi les intervalles et les segments qui caractérisent le tracé ECG, on rencontre :

- Intervalle PR

Cet intervalle mesure la distance entre le début de l'onde P et le début du complexe QRS, il représente la dépolarisation des oreillettes et du noeud AV. Sa durée normale est entre 0.12 et 0.2 secondes [Kun82].

- Segment PR

C'est la période temporelle comprise entre la fin de l'onde P et le début du complexe QRS, il représente le temps de transmission du front de dépolarisation par le noeud AV [Kha06].

- Segment ST

Le segment ST est compris entre la fin du complexe QRS et le début de la phase ascendante de l'onde T. Ce segment correspond au temps pendant lequel l'ensemble des cellules myocardiques sont dépolarisées et donc, dans le cas normal, il doit être isoélectrique. Dans le cas contraire, le niveau d'amplitude et la pente de ce segment sont des indicateurs de l'état ischémique du myocarde.

- Intervalle QT

L'intervalle QT est le temps entre le début du complexe QRS et la fin de l'onde T. Il représente une indication de la longueur des phases de dépolarisation et ré-polarisation ventriculaires. Sa durée varie avec la fréquence cardiaque entre 0.3 et 0.38 secondes [Ham04].

- Intervalle PP

L'intervalle PP mesure la durée du cycle des oreillettes, c'est un indicateur de la fréquence des oreillettes.

- Intervalle RR

Cet intervalle représente la distance entre deux ondes R successives, c'est l'indicateur de la fréquence des ventricules [Kha06].

4.3.2. Artefacts de l'enregistrement ECG

Tout enregistrement ECG peut contenir des événements indésirables qui peuvent brouiller le tracé et introduire des erreurs dans le diagnostic final. Ces bruits sont reconnaissables à l'œil expérimental qui les identifie avant tout diagnostic. Ces perturbations ont fait l'objet d'études et restent, pour certaines, encore difficiles à traiter de manière automatique. Les effets indésirables peuvent avoir plusieurs sources : techniques, physiques, pathologiques, ou pharmacologiques.

Donc les artefacts sont des déflexions du tracé ECG qui sont dues à des influences autres que l'activité cardiaque du cœur.

*** Artefacts dus aux mouvements d'électrodes**

Les mouvements d'électrodes produisent des perturbations de la distribution des ions qui cause des changements dans le potentiel de la cellule, le résultat de ces mouvements

pour le tracé ECG se traduit par un décalage de la ligne de base qui apparaît beaucoup plus dans le segment PR qui se décale de la ligne isoélectrique d'amplitude zéro (figure 4.4.a). Cet éloignement de la ligne de base représente une perturbation de basse fréquence de l'ordre de 0.1 Hz [Ben09].

*** Mouvements de la ligne de base**

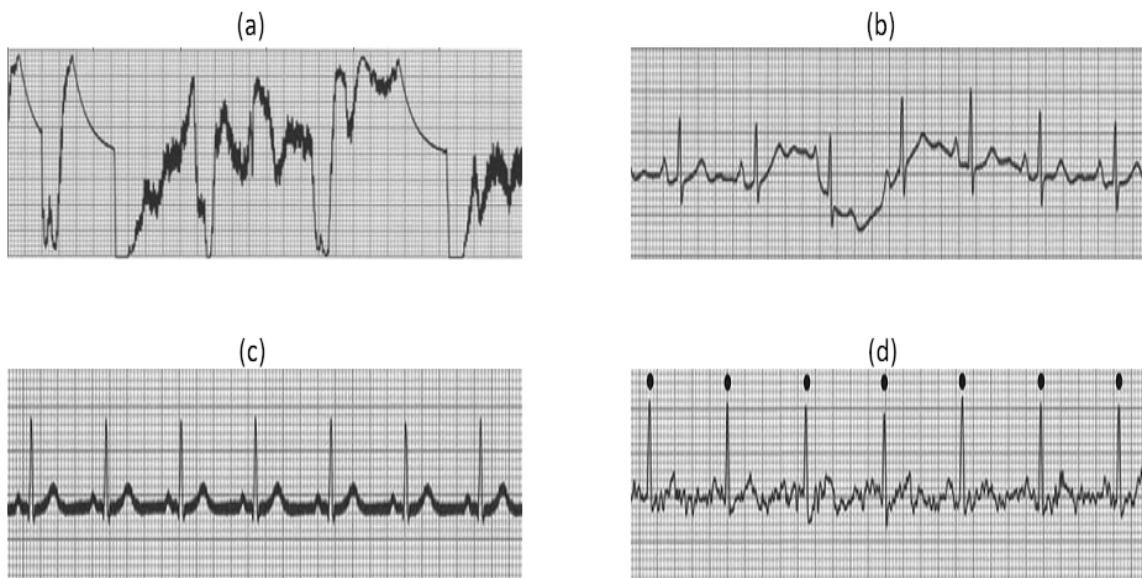
En plus des mouvements des électrodes, la ligne de base se décale de la ligne isoélectrique (figure 4.4.b) à cause des mouvements du câble et la fréquence d'un tel bruit est de l'ordre de 0.1 à 10 Hz.

*** Artefacts dus au secteur**

Ces bruits sont dus à la source de tension qui alimente l'appareil à travers le secteur, ce sont des bruits de haute tension de 50 ou 60 Hz (figure 4.4.c).

*** Artefacts musculaires**

Ce type de bruit est dû aux contractions musculaires. On remarque que l'intervalle RR du tracé ECG pour ces artefacts est très régulier (figure 4.4.d) [Ben09].



(a) artefacts dus aux mouvements d'électrodes, (b) mouvement de la ligne de base,
(c) artefacts dus au secteur et (d) artefacts musculaire

Figure 4.4: Artefacts du tracé ECG

4.4. Phonocardiogramme

L'activité cardiaque fait l'objet, depuis longtemps, de différentes études, notamment le premier test effectué par les cardiologues envers leurs patients et l'auscultation des battements cardiaques par l'intermédiaire d'un stéthoscope ordinaire. Le terme phonocardiographie se réfère à la technique de traçage des bruits du cœur et l'enregistrement des vibrations acoustiques cardiaque au moyen de micro-capteur. Le traitement des bruits cardiaques en termes d'enregistrements s'avère très important pour le diagnostic des différentes pathologies cardiaques. Le signal enregistré est connu sous le nom de phonocardiogramme (PCG) [Bou12]. Le signal PCG confirme, et surtout, affine les données de l'auscultation et apporte des renseignements complémentaires sur l'activité cardiaque en les situant par rapport aux bruits normaux du cœur [Deb05]. Les bruits cardiaques sont par définition des signaux non-stationnaires et sont situés dans la gamme des basses fréquences, approximativement entre 10Hz et 300Hz. L'analyse des bruits cardiaques par auscultation, basée seulement sur l'ouïe humaine, reste insuffisante pour un diagnostic fiable des cardiopathies et pour qu'un clinicien puisse obtenir toutes les informations qualitatives et quantitatives de l'activité cardiaque. Ces informations comme la localisation temporelle des bruits du cœur, le nombre de leurs composantes internes, leur contenu fréquentielle, l'importance des souffles diastoliques et systoliques peuvent être étudiée directement sur le signal PCG par l'utilisation de méthodes utilisant de techniques numériques du traitement de signal [Deb01], [Deb07], [Deb08], [Kha06], [Kha09], [Kun82].

Généralement, le PCG est composé de deux types de sons acoustiques: les sons du cœur (figure 4.5) et les souffles cardiaques. Dans un cycle cardiaque, jusqu'à quatre sons du cœur peuvent être trouvés: S1, S2, S3 et S4. S1 et S2 sont normalement observées chez tous les sujets. Tandis que S3 et S4 sont normales chez les sujets jeunes, mais pathologique chez les personnes âgées. Les sons du cœur sont des signaux transitoires de courte durée alors que les souffles cardiaques sont relativement plus longues et comme des signaux aléatoires. Ils sont généralement causés par un dysfonctionnement des valves cardiaques, sauf pour les murmures innocents qui peuvent se produire pendant la systole chez des sujets jeunes avec un cœur fort et normal. Les souffles cardiaques sont généralement séparés en deux types selon la chronologie du cycle cardiaque: Les souffles systoliques survenant entre S1 et S2 et les souffles diastoliques survenant entre S2 et S1 [Gue12], [Jon05], [Kha06].

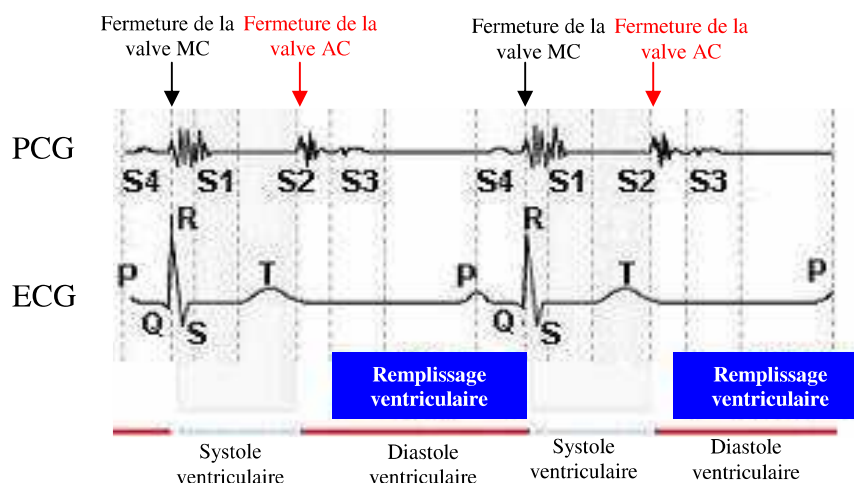


Figure 4.5 : Événements du cycle cardiaque

4.4.1. Sons cardiaques

* Premier son cardiaque (S1)

Ce son est généré au début de la phase systolique, c'est le résultat de la fermeture de la valve mitrale (MC) suivi de près par la fermeture de la valve tricuspide (TC). Il est composé de deux grandes composantes (M1) pour la valve mitrale et (T1) pour la valve tricuspide. Comme le ventricule gauche se contracte d'abord, la composante M1 se produira plus tôt. Ce délai entre M1-T1 a une importance clinique majeure et est considéré comme physiologique si moins de 30 ms. Au-dessus de cette valeur, il peut indiquer une rupture pathologique de S1. S1 a une durée typique de 100-200 ms et son spectre de fréquence se situe principalement entre 10-200 Hz [Czi99], [Gue12], [Kun82], [Sen00].

Les deux valves mitrale et tricuspide se ferment lorsque la pression ventriculaire est supérieure à celle de l'oreillette. La valve aortique s'ouvre (AO), suivie de près par l'ouverture de la valve pulmonaire (PO). L'ouverture des deux valves semi-lunaires n'est normalement pas entendue. Une valve aortique malade ou pulmonaire peut produire des sons d'éjection [Czi99], [Kun82], [Sen00].

Le premier son cardiaque est généralement perçu près de l'apex cardiaque sur la surface mitrale, dans le quatrième espace intercostal gauche. S1 se produit juste après le complexe QRS de l'ECG [Sen00].

* Deuxième son cardiaque (S2)

Le son S2 indique la fin du régime systolique et le début du régime diastolique. S2 est le résultat de la fermeture de la valve aortique (AC) suivie de près par la fermeture de la

valve pulmonaire (PC). Tout comme S1, S2 dispose également de deux composantes principales, A2 et P2 qui se rapporte à la fermeture de la valve aortique et pulmonaire. La durée entre A2 et P2 est supérieure à 30 ms elle peut correspond à une maladie (split). Ces deux valves se ferment lorsque la pression systolique d'éjection dans l'aorte et l'artère pulmonaire diminue. Lorsque les pressions ventriculaires et auriculaires sont les mêmes, les valves qui les séparent en silence seront ouvertes (mitrale et tricuspide). Si ces vannes sont anormaux ou sténoses, leurs ouvertures peuvent ne pas rester silencieuses [Czi99], [Kun82], [Sen00]. S2 a une durée minimale de 80 ms. Les composants de fréquence de S2 sont généralement plus élevés que ceux de S1 et son spectre se situe principalement entre 50-250 ms [Gue12], [Sen00].

Les deux composantes de S2 sont les mieux entendus sur la zone pulmonaire, lors du deuxième ou troisième espace intercostal. S2 se produit juste après l'onde T de l'ECG du patient [Sen00].

*** Troisième son cardiaque (S3)**

Habituellement, S3 est mieux entendu avec le Bell du stéthoscope sur la zone mitrale. Il est entendu mieux lors de l'expiration lorsque le flux sanguin dans le ventricule gauche est augmenté. S3 est entendu au début de la diastole et suit S2 par 0.14 à 0.20 secondes. S3 se produit pendant l'intervalle TP juste après le son S2. Normalement, l'intervalle entre S2 et S3 est constant [Gue12], [Sen00].

*** Quatrième son cardiaque (S4)**

Le son S4 est généré après la contraction auriculaire et avant le premier son du cœur. Il est causé par les vibrations ventriculaires après la contraction des oreillettes. Il est également de faible amplitude et de basse fréquence. Son apparition nous renseigne sur la présence d'une pathologie [[Czi99], Sen00]. S4 est généralement mieux entendu avec la cloche du stéthoscope pendant l'expiration près de l'apex du cœur sur la région mitrale. On peut le voir facilement dans la forme d'onde PCG: S4 se produit au cours de l'intervalle PR et se précède S1 [Gue12], [Sen00].

4.4.2. Souffles cardiaques

Les souffles cardiaques sont généralement associées à des états pathologiques tels que la sténose valvulaire cardiaque ou d'insuffisance, par exemple, un «souffle innocent» est un souffle causé par des conditions physiologiques et peut être entendu dans les enfants

et les sportifs. Par conséquent les souffles cardiaques ont divisés en quatre types: sténose aortique, sténose mitrale, insuffisance aortique et insuffisance mitrale. Les souffles cardiaques sont des vibrations prolongées dues à un écoulement turbulent du sang. Les souffles cardiaques sont des sons de hautes fréquences résultants de la circulation sanguine turbulente. Le flux turbulent est causé par des anomalies telles que la constriction ou les fuites des valves cardiaques, ils sont décrits selon la position dans le cycle cardiaque (systolique ou diastolique) [Bea97], [Gue12].

4.5. Conclusion

Dans ce chapitre on a exposé l'anatomie du cœur humain et son fonctionnement dans le système cardiovasculaire. Puis on a vu que l'activité des cellules cardiaques est la cause des potentiels électriques qui à leurs tours sont responsables de la naissance des ondes constituant le tracé électrocardiogramme. On a vu aussi que ce tracé est, parfois, affecté par différents types d'artefacts et peut comporter des arythmies caractérisées par des déformations des ondes ou des intervalles entre ces dernières.

Aussi, le PCG procure des informations importantes concernant le cœur en particulier la détection des troubles du rythme cardiaque.

Chapitre 5

TECHNIQUES ELABOREES POUR LA COMPRESSION DES SIGNAUX ECG ET PCG

Contenu

5.1. Introduction.....	58
5.2. Compression des signaux ECG.....	58
5.2.1. Compression basée sur la détection RR et la DWT 3D (RR-DWT3).....	58
5.2.1.1. Description de la technique.....	58
5.2.1.2. Résultats expérimentaux	62
5.2.2. Compression basée sur la technique SPIHT (SPIHT-VKTP).....	66
5.2.2.1. Description de la technique.....	66
5.2.2.2. Résultats expérimentaux.....	69
5.3. Compression des signaux PCG.....	72
5.3.1. Description de la technique.....	72
5.3.1.1. Passage des dominants.....	74
5.3.1.2. Passage des subordonnés.....	75
5.3.1.3. RLE et codeur arithmétique.....	77
5.3.2. Résultats expérimentaux.....	78
5.4. Comparaison entre les techniques développées.....	83
5.5. Conclusion.....	83

CHAPITRE 5

TECHNIQUES ELABOREES POUR LA COMPRESSION DES SIGNAUX ECG ET PCG

5.1. Introduction

Dans ce chapitre, on va décrire les méthodes proposées et développées pour la compression des signaux ECG et PCG. La première partie consiste à effectuer la compression des signaux ECG utilisant deux approches : la première technique est basée sur la détection des intervalles RR des signaux ECG en vue d'appliquer les transformées en ondelettes 3D. La deuxième procédure est une combinaison de deux approches SPIHT et VKTP permettant d'augmenter les performances de la compression. La deuxième partie concerne la compression des signaux PCG utilisant notre nouvelle approche nommée EEZW. Les résultats de simulation ont été obtenus utilisant Matlab 2010.

5.2. Compression des signaux ECG

5.2.1. Compression basée sur la détection RR et la DWT 3D (RR-DWT3)

5.2.1.1. Description de la technique

Il est bien connu que les signaux ECG possèdent un caractère pseudopériodique [Van11], avec un motif du battement cardiaque qui se répète plus ou moins régulièrement et une forme qui varie quelque peu. Alors on peut exploiter cette propriété de corrélation pour minimiser les redondances entre échantillons voisins dans la même période, dans des périodes voisines et dans des ensembles de périodes adjacentes. Donc, on peut dire que notre approche a été développée afin d'exploiter la corrélation existante entre les voxels adjacents. Notre algorithme est formé de six étapes (Figure 5.1).

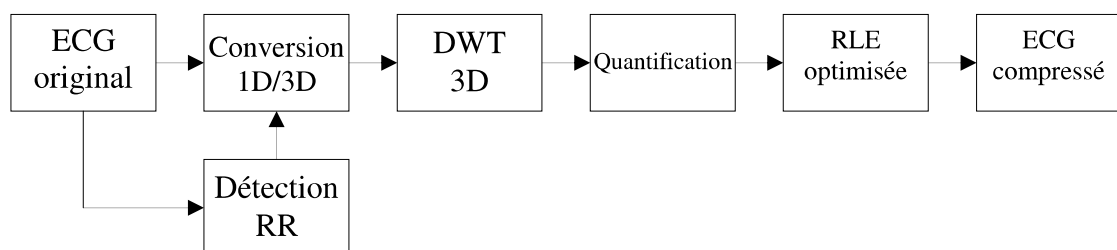


Figure 5.1 : Schéma synoptique de la technique proposée

La détection des intervalles RR sert à localiser les positions des pics R selon l'algorithme [Pan85]. Cette partie comprend cinq étapes (Figure 5.2) : Le signal ECG est filtré par un filtre passe bande (filtre passe bas suivi d'un filtre passe haut) accordés sur les fréquences 5 à 15 Hz choisis pour réduire les bruits additifs (bruits des muscles et les interférences de la ligne électrique (60Hz)). Le gain de cette dernière issu du filtre passe bas est -340db. Le filtre passe haut sert à atténuer l'onde P et l'onde T. Les deux fonctions de transfert de ce filtre sont données comme suit [Pan85], [Sin10]:

$$H_1(Z) = \frac{1}{35} \frac{(1 - Z^{-6})^2}{(1 - Z^{-1})^2} \quad (5.1)$$

$$H_2(Z) = \frac{1}{35} \frac{-1 + 35Z^{-16} - 35Z^{-17} + Z^{-32}}{1 - Z^{-1}} \quad (5.2)$$

Le filtre dérivateur permet de connaître la localisation des ondes R. La dérivée du signal à ce niveau présente des valeurs maximales élevées. La fonction de transfert de ce filtre est donnée par l'expression (5.3), avec T période d'échantillonnage du signal ECG.

$$H_3(Z) = \frac{1}{8T} (-Z^{-2} - 2Z^{-1} + 2Z^1 + Z^2) \quad (5.3)$$

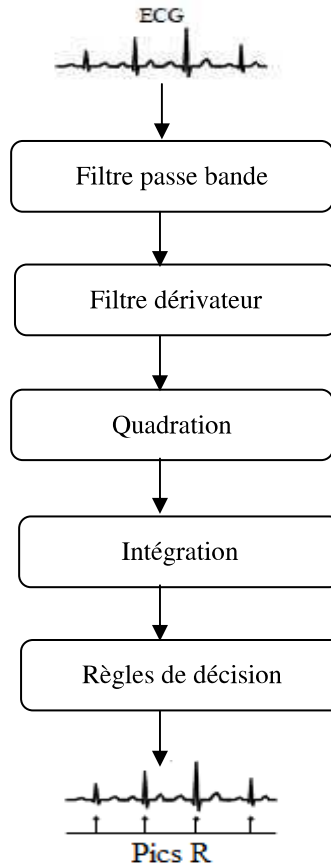


Figure 5.2 : Algorithme de détection des intervalles RR

Afin d'accentuer le contraste mis en évidence par la dérivation, on élève au carré le signal obtenu, ce qui permet en outre, d'éliminer le signe du signal et augmenter la possibilité de la détection des pics R.

L'intégration du signal permet d'obtenir un maximum unique pour chaque onde R. le maximum est décalé en temps par rapport à la position du pic R. La fonction de transfert de ce filtre est donnée par:

$$H_4(Z) = \frac{1}{N} (Z^{N-1} + Z^{N-2} + \dots + 1) \quad (5.4)$$

Où N est le nombre d'échantillons dans la largeur de la fenêtre qui est la même que la largeur du complexe QRS. N=54 échantillons (correspond à 150ms) pour une fréquence d'échantillonnage de 360 échantillons par seconde.

La dernière étape de la chaîne de détection des ondes R est basée sur des seuils qui sont continuellement ajustés par un ensemble de règles de décision, afin de suivre les changements du signal ECG. Les formes des signaux après chaque étape sont illustrées dans la figure 5.3.



(a) Signal ECG (b) Filtrage passe bande (c) Dérivation (d) Quadrature
(e) Intégration (f) Détection du maximum

Figure 5.3 : Détection des pics R

La conversion de format 1D/3D consiste à remplir un volume d'échantillons. Chaque ligne d'un plan du volume supporte les échantillons dans un intervalle RR et l'ensemble des plans forment le volume. Le nombre de colonnes dans chaque plans représente le nombre d'échantillons pour l'intervalle RR le plus long que peut contenir le signal ECG. Par conséquent, de nouveaux échantillons doivent être insérés pour compléter les lignes incomplètes. Ces échantillons seront éliminés lors de la phase de reconstruction.

Les méthodes de compression des signaux ECG basées sur la transformée en ondelettes discrète, admet généralement, une transformation biorthogonale qui est appliquée au signal original. C'est pour cela, nous avons appliqué la DWT 3D utilisant l'ondelette mère «bior 4.4» pour un seul niveau de décomposition. On peut obtenir avec ses choix de meilleurs compromis entre le taux de compression et l'erreur quadratique moyenne.

La quantification scalaire permet d'améliorer le taux de compression. C'est-à-dire, approximer chaque coefficient réel par un autre type entier le plus proche possible. Alors si le pas de quantification est grand, la ressemblance des valeurs quantifiées augmente, mais la perte d'information est grande. Dans cette étape, on perd de l'information, nous devons la perdre d'une manière confirmée et astucieuse puisque la qualité du signal restituée dépend du pas de quantification. Ce qui permet d'augmenter la redondance des éléments.

L'étape de codage va permettre de générer la séquence de bits pour laquelle l'algorithme de codage sera capable d'améliorer le taux de compression d'une manière confirmée. Nous avons développé à ce niveau une nouvelle variante RLE que nous avons appelé RLE optimisée. On résume les étapes de cet algorithme en ce qui suit:

Etape 1 : Calcul du nombre «n» en bits pour toutes les apparitions.

Etape 2 : Calcul de la taille «f» en bits des fréquences de répétition

Etape 3 : Pour toutes les apparitions de la séquence :

- *Si l'apparition est unique (pas de répétition), alors insérer un '0' puis la valeur d'indice i en binaire sur «n» bits.*
- *S'il y a une séquence d'apparitions de mêmes valeurs, alors insérer "1" puis le nombre de répétitions en binaire sur «f» bits, puis la représentation de cette valeur en binaire sur «n» bits.*

Etape 4: Retour à l'étape 3

Le principe de fonctionnement de cet algorithme est illustré aussi dans l'exemple de la figure 5.4.

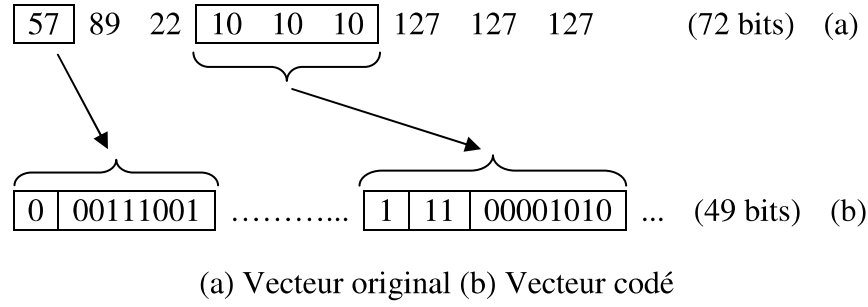


Figure 5.4 : Exemple de principe de l'algorithme RLE optimisé

5.2.1.2. Résultats expérimentaux

Plusieurs bases de données de signaux ECG existent à savoir AHA, ESC, NST, CU, ST,... etc. La majorité des recherches scientifiques développées et appliquées aux signaux ECG utilisent la base de données Arrhythmia MIT-BIH [DB1]. C'est pour cela nous avons opté à utiliser une telle base dont l'accès est libre. Nos enregistrements de tests sont : 100, 103, 109, 113, 117, 119, 121, et 124. Ils sont échantillonnés à une fréquence de 360 Hz et chaque échantillon est codé sur 11 bits.

Deux critères sont généralement retenus pour l'évaluation des algorithmes de compression: le rapport de compression (CR : Compression Ratio) et la racine carrée de l'erreur en pourcentage PRD (Percentage of Root mean square Difference) ou taux de distorsion. Le rapport de compression est donné par l'expression (5.5) [Bou09a].

$$CR = \frac{\text{Nombre de bits du signal original}}{\text{Nombre de bits du signal compressé}} \quad (5.5)$$

La distorsion est l'erreur introduite par l'opération de compression, due au fait que le signal reconstruit n'est pas exactement identique au signal original. Le PRD des signaux ECG à valeur moyenne 1024 est définie comme suit :

$$PRD = \sqrt{\frac{\sum_{i=0}^N (x_{org} - x_{rec})^2}{\sum_{i=0}^N (x_{org} - 1024)^2}} * 100 \quad (5.6)$$

Où x_{org} et x_{rec} représentent respectivement le signal original et le signal reconstruit. N représente le nombre d'échantillons.

Pour juger la qualité de la détection des pics R, le tableau 5.1 donne quelques résultats de détection pour quelques enregistrements ECG.

<i>ECG</i>	<i>Nombre total de battements (NT)</i>	<i>Nombre de battements détectés (ND)</i>	<i>Différence (D) D=NT-ND</i>
100	2273	2273	0
103	2084	2083	1
109	2532	2536	-4
112	2539	2553	6
117	1535	1535	0
119	1987	1960	27
121	1863	1863	0
122	2476	2475	1

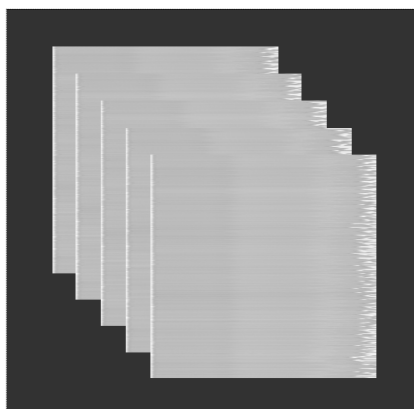
Tableau 5.1 : Résultats de simulation pour la détection des pics R

Si on note :

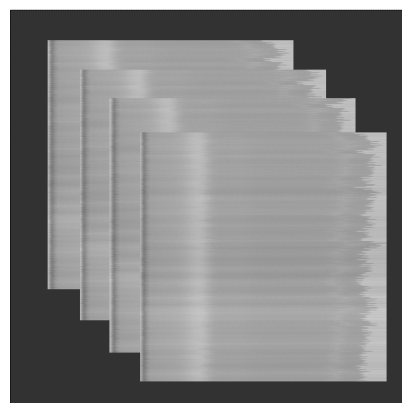
- NT : nombre total de battements analysés par l'algorithme.
- ND : nombre de battements détectés par l'algorithme.
- FP (Faux Positifs): c'est le nombre d'ondes R qui ont été détectées par l'algorithme alors qu'elles ne font pas partie de cette catégorie. Dans ce cas, il s'agit d'une autre onde caractéristique.
- FN (Faux Négatifs): c'est le nombre d'ondes étiquetées R que l'algorithme n'a pas pu détecter.
- Différence (D): $D=NT-ND$. Quatre cas se présentent:
 - ✓ $D=0$ ($NT=ND$) pour $FP=FN=0$
 - ✓ $D=0$ ($NT=ND$) pour $FP \neq 0$ avec $FP=FN$.
 - ✓ $D < 0 \Rightarrow FP > FN$
 - ✓ $D > 0 \Rightarrow FP < FN$

Pour $D=0$ avec $FP=FN$, le volume d'échantillons en voxels est optimale (occupe moins d'espace) par rapport aux trois autres situations. Donc, l'efficacité de compression est plus élevée dans ce cas.

Les allures des enregistrements 100 et 117 obtenus au format 3D après conversion 1D/3D sont visualisées dans la figure 5.5. Les pics R sont situés sur l'extrémité gauche (première colonne) pour chaque plan, ce sont généralement les colonnes les plus brillantes. Le volume obtenu occupe les dimensions $n \times n \times m$, tel que n est la longueur (nombre d'échantillons) du battement le plus long du signal ECG et m représente le nombre de plans constitués par le signal.



Enregistrement 100



Enregistrement 117

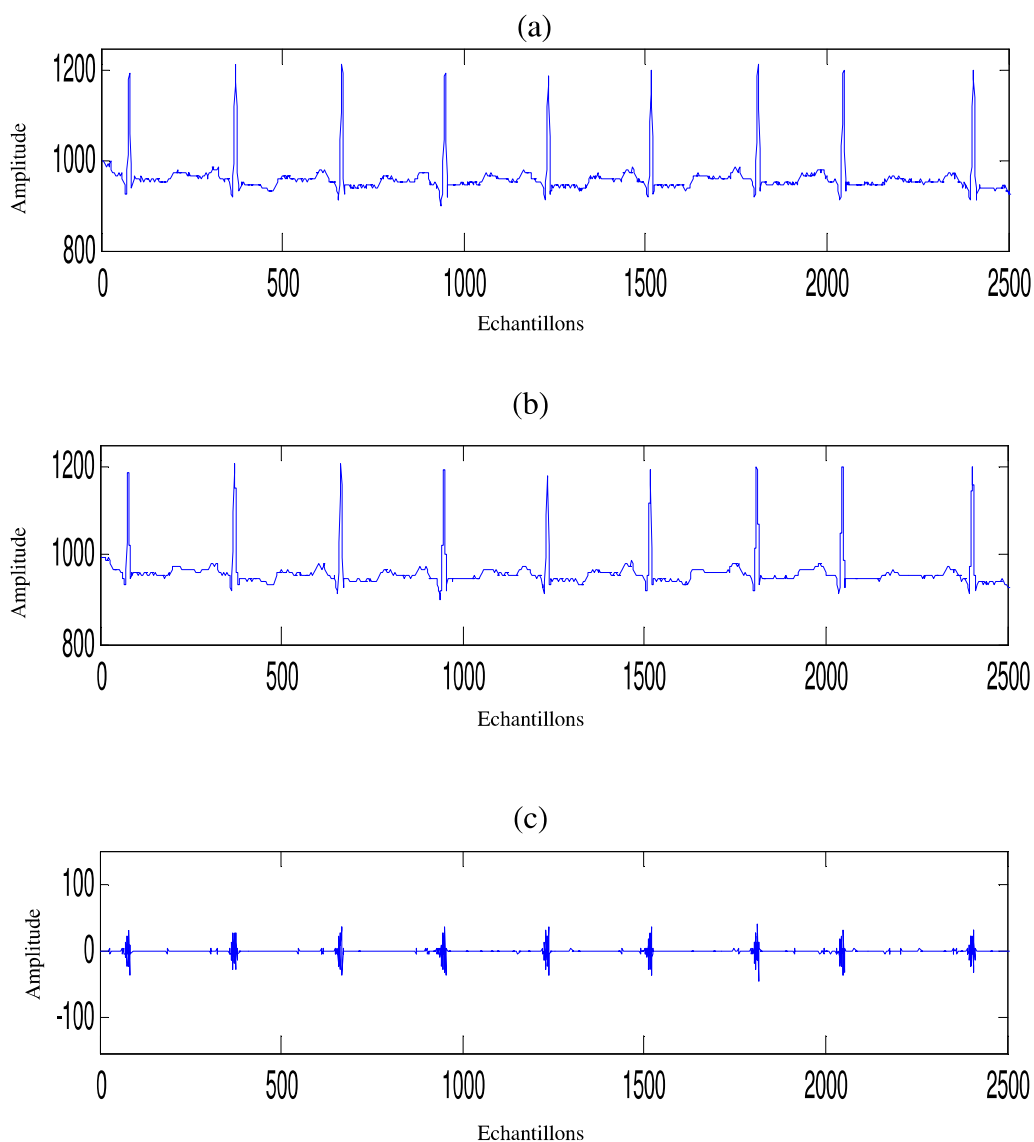
Figure 5.5 : Enregistrements 100 et 117 au format 3D

Le tableau 5.2 montre les résultats expérimentaux de comparaison entre la nouvelle approche développée RLE optimisée, RLE standard et le codage arithmétique. On remarque que le codeur RLE optimisé a apporté des résultats améliorés dans notre application par rapport aux deux autres pour un CR moyen 7.57 contre 4.60 et 6.77 utilisant la technique RLE standard et le codage arithmétique respectivement.

<i>ECG</i>	<i>CR</i> <i>(arithmétique)</i>	<i>CR</i> <i>(RLE standard)</i>	<i>CR</i> <i>(RLE optimisé)</i>
100	7.70	5.23	8.39
103	7.55	4.13	7.67
109	6.18.	3.82	6.30
113	7.36	4.40	8.95
117	6.46	4.84	7.26
119	6.13	3.68	6.36
121	6.43	5.99	8.20
124	6.40	4.74	7.44
CR moyen	6.77	4.60	7.57

Tableau 5.2 : Résultats de comparaisons entre trois codeurs

La figure 5.6 montre les résultats de restitution (qualité visuelle) du signal ECG 100. On remarque que ce dernier est presque identique à l'enregistrement original, l'erreur de reconstruction est très faible. Ce qui permet de préserver la qualité du signal après reconstitution pour une éventuelle analyse.



(a) Signal ECG 100 original (b) Signal ECG 100 reconstruit (c) Signal erreur

Figure 5.6 : Compression/Décompression du signal ECG 100

Les résultats de notre algorithme ont été comparés quantitativement avec d'autres méthodes utilisant le format 1D (Tableau 5.3). L'algorithme de Hilton [Hil97] est basé sur la compression des signaux ECG par paquets d'ondelettes. Le deuxième algorithme est celui de Djohan [Djo95] qui est basé sur la compression des signaux ECG par la transformée en ondelette symétrique discrète. Enfin l'algorithme de Tchiotsop [Tch07b] qui est fondé sur la compression des signaux ECG par la transformée des polynômes de Jacobi.

En utilisant l'enregistrement 117, les résultats obtenus montrent que notre algorithme a montré son efficacité en offrant moins de déformations dans le signal reconstitué (PRD=2.08%) contre 2.60%, 3.90% et 13.91% pour les trois autres méthodes. Aussi, avec un meilleur taux de compression (CR=11.74) contre 8.00 pour Hilton et Djohan et 11.57 pour Tchiotsop.

Algorithme	CR	PRD (%)
Approche proposée	11.74	2.08
Hilton [Hil97]	8.00	2.60
Djohan [Djo95]	8.00	3.90
Tchiotsop [Tch07b]	11.57	13.91

Tableau 5.3 : Résultats de compression du signal 117 pour quatre techniques

5.2.2. Compression basée sur la technique SPIHT (SPIHT-VKTP)

5.2.2.1. Description de la technique

Comme il a été signalé, l'une des approches les plus performants utilisant la DWT est la technique SPIHT [Lu00]. Cette méthode permet de contrôler le débit de transmission. Dans cette section une nouvelle approche sera détaillée utilisant la technique SPIHT suivi du codeur VKTP (développé par Martinez-Alajarin) [Mar08]. Cette combinaison permet d'améliorer les performances de compression.

La transformée en ondelettes permet de décorréler un signal et de concentrer ses coefficients dans un intervalle réduit de coefficients. La figure 5.7 nous montre la mise en cascade de filtres, qui permettent de décomposer un signal à 1D à trois niveaux [Bou09b].

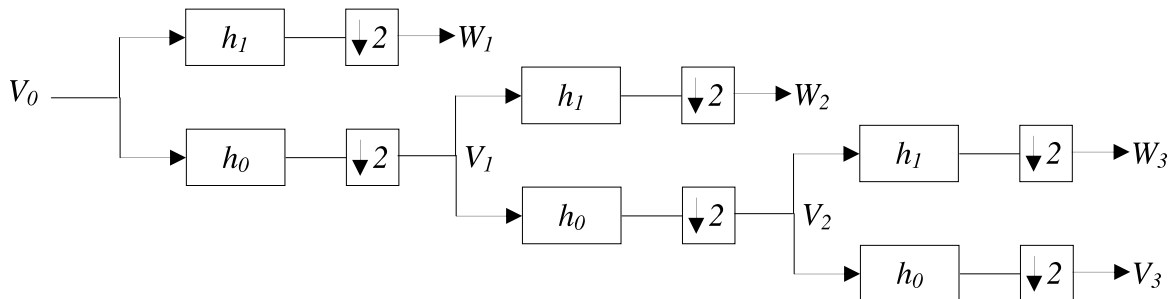


Figure 5.7 : DWT du signal ECG jusqu'à 3 niveaux de décompositions

Le signal au niveau j passe à travers un filtre passe bas h_0 et un filtre passe haut h_1 , ensuite un sous échantillonnage par deux est appliqué ; alors ces mêmes opérations sont

répétées pour les niveaux $j+1$ et ainsi de suite. Les valeurs obtenues représentent les coefficients d'ondelettes à différentes échelles.

La relation entre les coefficients entre les différents niveaux est présentée dans la figure 5.8.

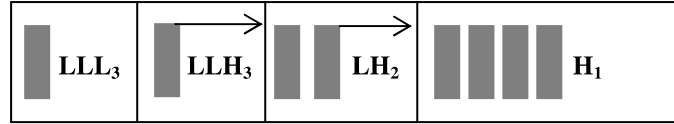


Figure 5.8 : Relation Parent-Enfant des coefficients des sous-bandes

Dans les sous-bandes des coefficients d'ondelettes à l'exception de la sous-bande de haute fréquence, chaque coefficient à un niveau donné est appelé *parent*. Ce dernier peut être lié à un ensemble de coefficients au niveau juste inférieur pour une position semblable, où le même endroit spatial s'appelle *enfant*. Les dépendances Parent-Enfants sont montrées dans la figure 5.8. Ainsi un coefficient dans la bande LLH_3 est le parent de deux coefficients dans la bande LH_2 . Les lettres L et H désignent bas (Low) et haut (High) respectivement. On note que le nombre de coefficients dans LLH_3 est la moitié du nombre de coefficients dans LH_2 par exemple [Ahm00].

Les coefficients dans les différentes sous-bandes sont collectés dans un vecteur W représenté par les éléments suivants :

$$W=[V_3 \ W_3 \ W_2 \ W_1] \quad (5.7)$$

Un codeur SPIHT est spécialement désigné pour être combiné avec la transformée en ondelettes, et qui a été utilisé dans la compression des images. Mais il peut être aussi employé dans les signaux ECG.

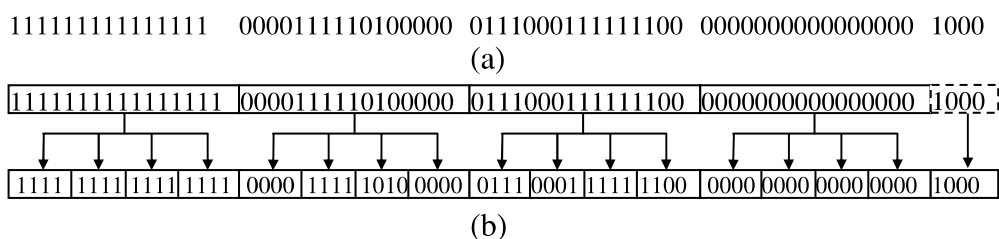
L'algorithme SPIHT permet d'obtenir un flot de bits contrôlable, c'est à dire le processus peut s'arrêter à n'importe quel débit spécifiée ou à n'importe quel qualité de reconstruction désirée. Généralement la technique SPIHT est suivie par un codeur arithmétique. Dans cette section dans le but d'augmenter le rapport de compression et réduire l'erreur de reconstruction, une technique VKTP est utilisée [Mar08].

L'algorithme VKTP consiste essentiellement à diviser itérativement le vecteur résultant (taille $1 \times LBS$, où LBS (Length of Bit String) est la longueur du flot binaire) dans un arbre constitué de K-branches (avec $K > 1$). Toutes les branches ont la même taille = LBS/K (avant la décomposition, le vecteur doit être complet, il faut obtenir une longueur multiple de K).

Pour le premier niveau de décomposition, les bits du vecteur résultant du codeur SPIHT seront groupés dans K sous vecteurs. Un tel codage utilise le symbole "1", si tous les bits d'un sous ensemble sont des "1"; "0", si tous les éléments sont des "0"; et "X", si l'ensemble des éléments contient des "0" et des "1". L'ensemble "X" sera décomposée itérativement en K sous vecteurs jusqu'à l'obtention des valeurs "1" et "0" au dernier niveau de décomposition.

Finalement, les symboles "1", "0" et "X" pour les différents niveaux seront remplacés par les codes "1", "00", et "01", selon les règles suivantes: (i) pour coder l'arbre de répartitions, il faut suivre l'ordre commençant du niveau haut allant au niveau bas; (ii) le symbole de plus grande fréquence d'apparition pour tous les niveaux à l'exception du dernier, est remplacé par "1"; (iii) les deux autres symboles seront remplacés par les codes "00" et "01", choisis indifféremment [Mar08].

La Figure 5.9 donne une illustration graphique de l'algorithme VKTP, commençant par le flot binaire à l'entrée en arrivant à la formation de sous-groupes au deuxième niveau de décomposition.



(a) Séquence de bits du vecteur original
(b) Vecteur partitionné en K-arbres avec K = 4

Figure 5.9 : Exemple de partitionnement d'une chaîne de bits utilisant l'algorithme VKTP

La figure 5.10 montre les symboles utilisés pour coder la structure de la Figure 5.9. Le tableau 5.4 résume l'affectation de symboles aux branches pour chaque niveau de l'arbre de la figure 5.9, la fréquence d'apparition de chaque symbole et les codes utilisés [Bou09a].

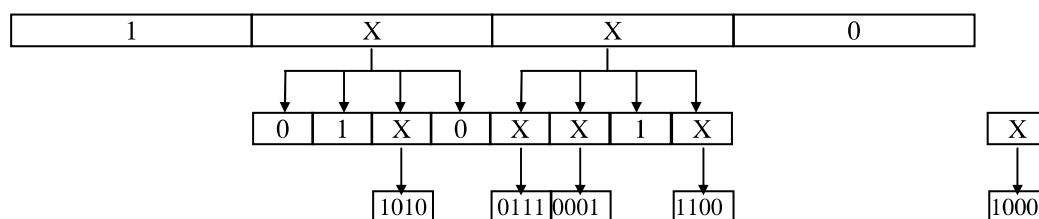


Figure 5.10 : Illustration de l'algorithme VKTP avec un arbre limitée à trois niveaux

Cette technique de compression des signaux ECG est constituée des étapes suivantes :

Etape 1: Décomposition du signal ECG utilisant la DWT.

Etape 2: Quantification linéaire des coefficients d'ondelettes.

Etape 3: Compression des coefficients quantifiés utilisant l'algorithme SPIHT 1D.

Etape 4: Troncature de flot de bits émis en sortie pour atteindre le débit désiré avec LBS multiple de K.

Etape 5: Compression du vecteur résultant utilisant la technique VKTP.

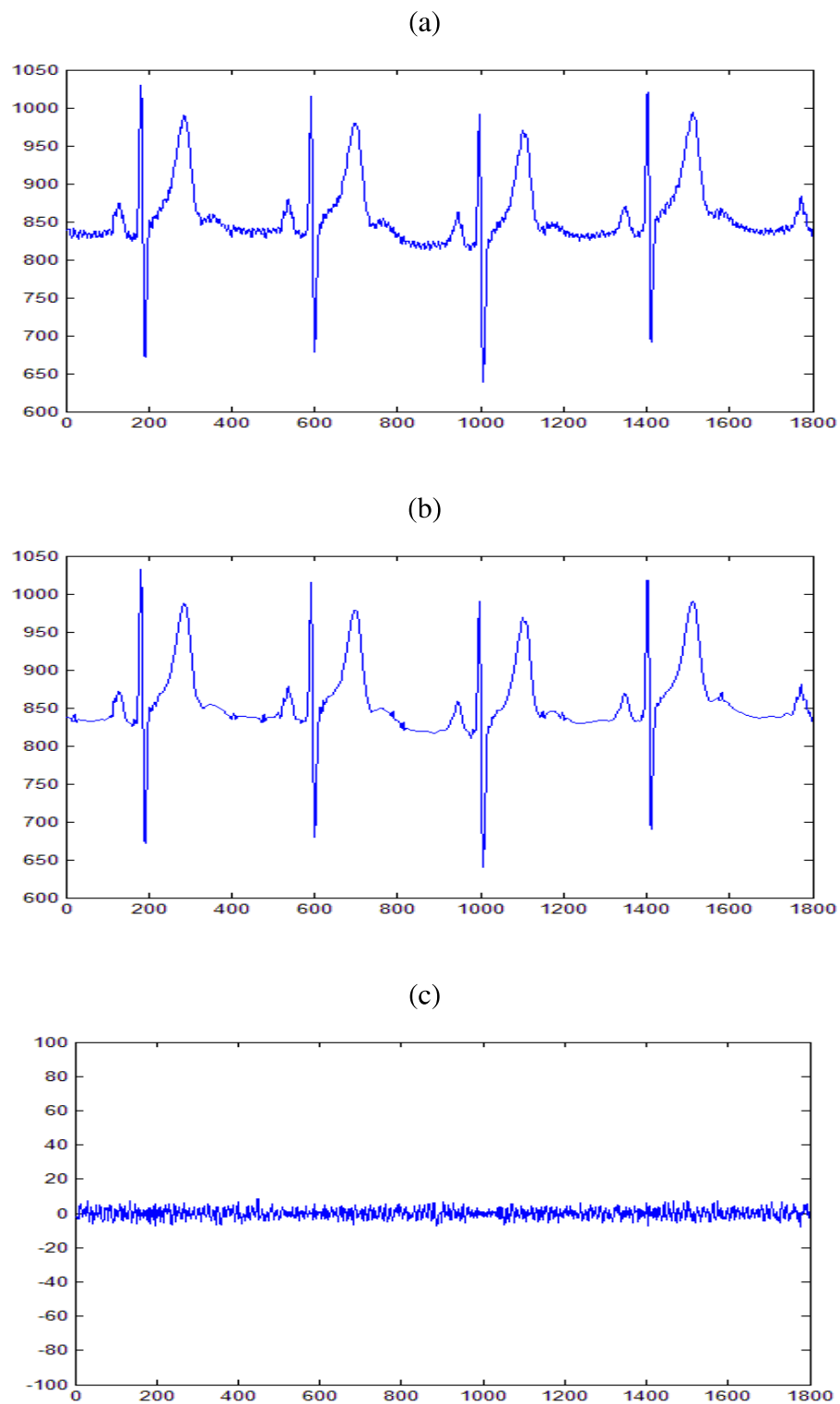
Niveau	Symboles	Codes
1	1-X-X-0	01-1-1-00
2	0-1-X-0-X-X-1-X-X	00-01-1-00-1-1-01-1-1
3	1010-0111-0001-1100-1000	1010-0111-0001-1100-1000
Nombre d'apparences	'X':7	'1':3 '0':3
Equivalence	'X' $\Rightarrow$ "1"	'1' $\Rightarrow$ "01" '0' $\Rightarrow$ "00"

Tableau 5.4 : Symboles et codes utilisés pour coder chaque niveau de l'arbre de partitionnement de la séquence binaire de la figure 5.9

5.2.2.2. Résultats expérimentaux

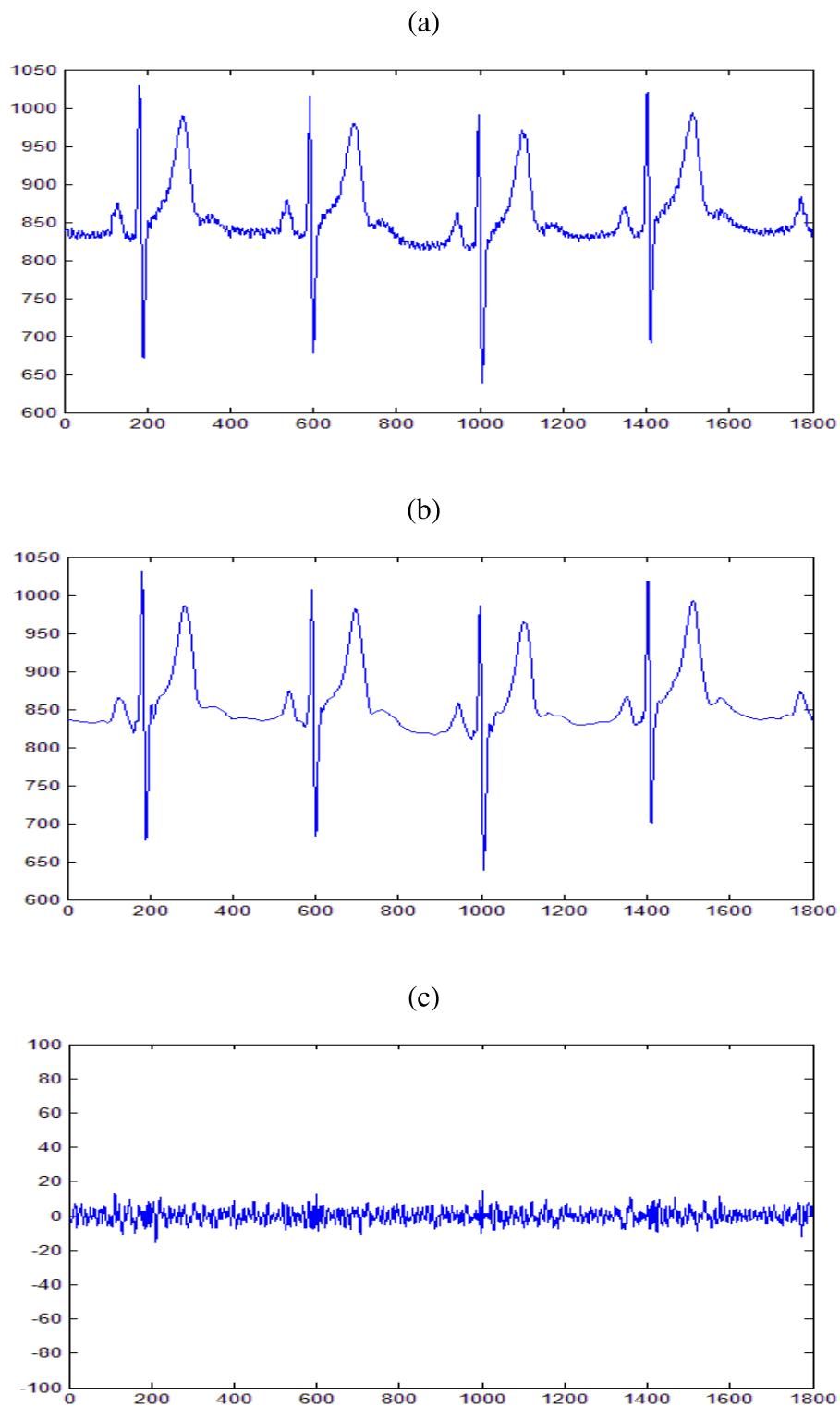
L'ensemble des signaux de tests est constitué des enregistrements 100, 101, 102, 103, 107, 109, 111, 115 et 117 pour 10 minutes de longueur. Les signaux ECG sont décomposés en blocs de 1024 échantillons. La DWT est appliquée pour chaque bloc pour obtenir les coefficients d'ondelettes pour cinq niveaux de décompositions utilisant l'ondelette mère «bior 4.4». Les critères utilisés pour évaluer notre approche sont ceux des expressions (5.5) et (5.6).

Pour CR=6:1 et CR=7:1, les signaux ECG 117 reconstruits sont visualisés dans les figures 5.11.b et 5.12.b respectivement. Les Figures 5.11.c et 5.12.c représentent les signaux erreurs obtenus. On remarque, en plus des hautes performances de compression obtenus, que les signaux ont été filtrés des bruits additifs.



(a) Signal ECG 117 original (b) Signal ECG 117 reconstruit (c) Signal erreur

Figure 5.11 : Compression/Décompression du signal ECG 117 pour CR=6



(a) Signal ECG 117 original (b) Signal ECG 117 reconstruit (c) Signal erreur

Figure 5.12 : Compression/Décompression du signal ECG 117 pour CR=7

Le tableau 5.5 montre les résultats de comparaisons entre notre approche et celui de [Lu00] utilisant l'ensemble des signaux de test déjà cités. Notre algorithme dépasse les performances de la technique SPIHT pour tous les signaux en termes de CR pour les mêmes valeurs de PRD.

PRD	1.67	1.92	2.43	3.62	4.67	5.66
CR de la technique proposée	9.12	11.42	14.31	16.29	19.34	21.14
CR de SPIHT [Lu00]	8.69	10.59	13.24	15.10	17.66	19.32

Tableau 5.5 : Résultats obtenus pour l'ensemble des signaux de tests

Le tableau 5.6 montre aussi les résultats de comparaisons avec autres techniques existantes [Bou09a]. Hilton [Hil97] a utilisé la technique EZW basée sur la transformée en ondelettes par paquets. Il a obtenu un PRD=2.6% avec un CR=8:1 pour le signal 117. La valeur du PRD dans notre approche est 1.18% pour le même enregistrement avec un CR=8.11:1. On peut dire que les performances de notre approche sont meilleures que les performances des codeurs dans [Lu00] et [Hil97].

Algorithme	CR	PRD (%)
Méthode proposée	8.11	1.18
SPIHT [Lu00]	8.00	1.18
Hilton [Hil97]	8.00	2.60
Chen [Che08]	8.31	1.07
Benzid [Ben03]	16.24	2.55
Blanco-Velasco [Bla04]	8.24	1.17

Tableau 5.6 : Résultats de compression du signal ECG 117

5.3. Compression des signaux PCG

5.3.1. Description de la technique

On va maintenant évaluer l'impact du système de compression utilisant notre approche proposée appelée EEZW. Les différentes étapes de conception ainsi que les tests

et les résultats obtenus seront exposés par la suite. Rappelons que l'application visée est la compression des signaux PCG.

Le flot de bits est généré utilisant le codeur du schéma bloc de la figure 5.13. Notre technique est divisée en six parties. Dans le premier bloc, les coefficients d'ondelettes sont générés utilisant la DWT. Ensuite un quantificateur linéaire scalaire uniforme est appliqué. Deux listes sont obtenues aux sorties des blocs des passages des dominants et passage des subordonnés. Durant le passage des dominants, les coefficients quantifiés sont comparés avec le seuil T_i pour déterminer leurs signifiante, et s'ils sont signifiants, leurs signes.

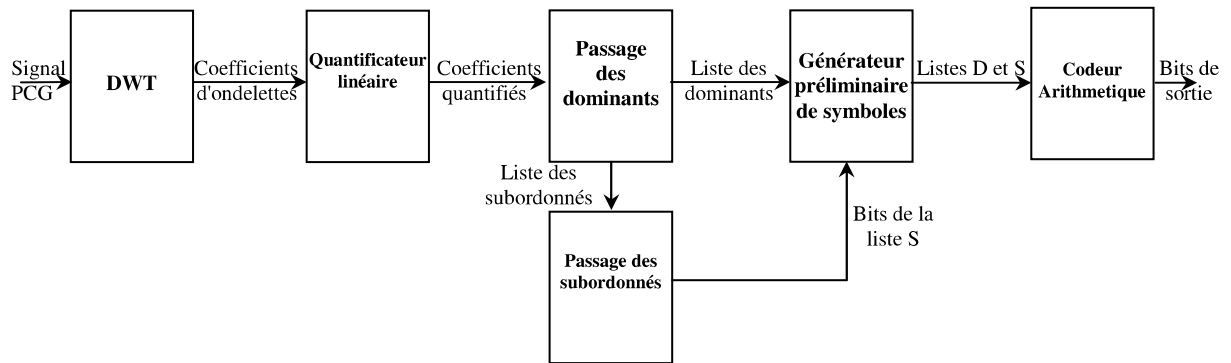


Figure 5.13 : Schéma bloc du codeur proposé

Dans l'algorithme de Shapiro [Sha93], un passage des dominants est suivi par un passage des subordonnés dans lequel tous les coefficients dans la liste des subordonnés seront vérifiés. Le balayage des coefficients est réalisé de telle façon qu'aucun enfant ne soit balayé avant son parent. La liste des subordonnés contient les coefficients significatifs. La séquence des seuils décroissants est $T_0 \ T_1 \ T_2 \ \dots \ T_{N-1}$ avec $T_i = T_{i-1}/2$ et $|coefficients| < 2T_0$ [Bou09c].

Pour simplifier les calculs et réduire le temps de traitement, seulement un seul passage des subordonnés est réalisé après tout les passages des dominant dans notre cas.

Finalement, un générateur préliminaire de symboles et un codeur arithmétique sont utilisés [Bou12]. Autrement dit, utilisant les coefficients quantifiés, le flot de bits est généré en suivant les étapes suivantes :

Etape 1: Initialisation du seuil, de la liste des dominants et de la liste des subordonnés.

Etape 2: Utilisant les passages des dominants pour créer le flot de symboles de la liste D et insérer les éléments significatifs dans la liste des subordonnés.

Etape 3: Un flot de bits de la liste S est généré après avoir appliqué un traitement simple et rapide utilisant un seul passage des subordonnés.

Etape 4: Un flot préliminaire de symboles est obtenu utilisant le diagramme de la figure 5.16, il est basé sur la technique RLE.

Etape 5: Un codeur arithmétique peut être utilisé pour obtenir des performances plus élevés avec notre approche.

5.3.1.1. Passage des dominants

Chaque coefficient ainsi que ses descendants sont balayés pour chaque passage. Si un coefficient est significatif, les symboles "P" ou "N" seront insérés dans le flot de symboles. Si un coefficient est insignifiant et tous ces descendants sont insignifiants, le symbole "T" sera insérer. Autrement, si un descendant est jugé significatif, le symbole "Z" est affecté.

Tous les coefficients signifiants sont placés sans leurs signes dans la liste des subordonnés. Un coefficient est codé si son amplitude (en valeur absolue) est inférieure au seuil $2T_i$. Ce qui nous permet d'éviter de coder ce coefficient une autre fois. Le passage des dominants est résumé en ce qui suit:

Tant que $(abs(coefficient) < 2 T_i)$

Début

Coder le coefficient balayé

Insérer le symbole

Si coefficient est codée en tant que "P" ou "N" alors

Début

Ajouter $abs(coefficient)$ dans la liste des subordonnés

Fin

Balayer le coefficient suivant

Fin

Considérons les coefficients du vecteur de test de la figure 5.14.

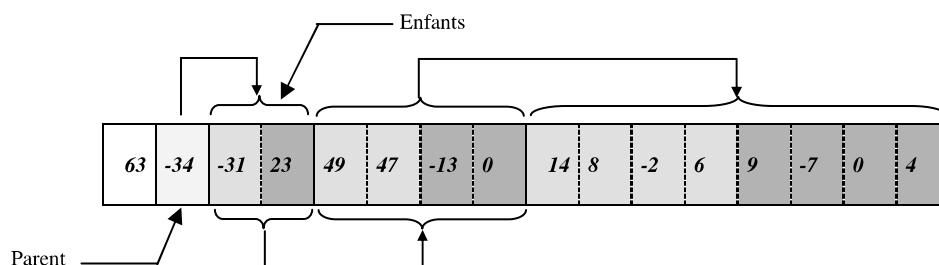


Figure 5.14 : Exemple de décomposition jusqu'au niveau 4 pour 16 cellules

D_0 est le premier flot de symboles obtenu dans la première itération, où $T_0=32$.

D_0 : P N Z T P P Z Z Z Z

L'algorithme est répété pour le reste des coefficients pour un seuil $T_i/2$. Pour $T_i=16$, la séquence D_i est obtenue, tel que :

$D_i: N P T T T T$

Ce processus est réitéré jusqu'à l'obtention de la qualité désirée du signal reconstruit ou jusqu'à ce que le nombre de bits ou la qualité désirés soient atteints.

5.3.1.2. Passage des subordonnés

Chaque passage des subordonnés dans la technique EZW envoie un seul bit pour chaque coefficient significatif. Durant le premier passage des subordonnés utilisant le seuil 32 (figure 5.14), l'intervalle d'incertitude pour tous les coefficients significatifs est $[32,64[$. Les coefficients appartenant à l'intervalle $[32,48[$, seront codés par le symbole "0" et ceux appartenant à l'intervalle $[48,64[$ seront codés par le symbole "1". On note que la limite de décision est la valeur 48. Dans le second passage de subordonnés (seuil 16) de nouveaux coefficients seront insérés avec les coefficients existants dans la liste des coefficients dominants. Dans ce cas, les intervalles d'incertitude $[48,64[$, $[32,48[$ et $[16,32[$ sont utilisés. Si un coefficient dans la liste des subordonnés appartient à la moitié inférieure d'un intervalle, le symbole "0" est envoyé; sinon le symbole "1" est émis si le coefficient appartient à la moitié supérieure de l'intervalle. Le processus est alterné entre les passages des dominants et les passages des subordonnés.

Utilisant notre approche, un seul passage des subordonnés est effectué. On peut ainsi aboutir aux mêmes résultats de l'algorithme original de Shapiro [Sha93]. Le flot de bits de la liste S est généré par utilisation de l'algorithme suivant:

Pour tous les éléments de la liste des subordonnés faire

Début

élément_binaire = forme binaire de l'élément sans le MSB

Placer la valeur binaire en position verticale

Fin

Générer le flot de bits de la liste S ($S=\{S_0, S_1, \dots\}$) par balayage des bits dans les positions horizontales

Cette étape est basée sur la figure 5.15. Dans chaque itération, tous les coefficients significatifs doivent être convertis en binaire. Pour mieux illustrer ce principe, le même exemple est utilisé:

Donc $(63)_{10}$ est codé en binaire par $(111111)_2$, $(34)_{10}=(100010)_2$, ..., jusqu'à $(2)_{10}=(10)_2$. A l'exception du MSB pour chaque nombre (toujours égale à 1), les autres bits seront placés dans la position verticale (comme l'indique la figure 5.15). Les séquences résultantes qui sont respectivement $S_0(1010)$, $S_1(100110)$, $S_2(1001111100)$, $S_3(1101110100110)$ et $S_4(10111110010100)$ représentent le flot de bits généré dans un seul passage des subordonnés.

	63	34	49	47	31	23	13	14	8	9	6	7	4	2
$T_0=32$	63	34	49	47										
$T_1=16$	31	23												
$T_2=8$	13	14	8	9										
$T_3=4$	6	7	4											
$T_4=2$	2													

	63	34	49	47	31	23	13	14	8	9	6	7	4	2
$S_0 :$	1	0	1	0										
$S_1 :$	1	0	0	1	1	0								
$S_2 :$	1	0	0	1	1	1	1	1	0	0				
$S_3 :$	1	1	0	1	1	1	0	1	0	0	1	1	0	
$S_4 :$	1	0	1	1	1	1	1	0	0	1	0	1	0	0

Figure 5.15 : Passage des subordonnés pour toutes les itérations du vecteur de test

Il est clair que pour une compression sans perte, tous les flots de bits sont pris en considération. En contrepartie, pour une compression avec perte, si le processus s'arrête au seuil T_i , seulement S_0 , S_1 , ..., S_i sont nécessaires pour le décodeur. Les valeurs reconstruites sont calculés au niveau i utilisant l'expression (5.8) :

$$T_0 + \sum_{j=0}^i b_j T_{j+1} + \frac{T_{i+1}}{2} \quad (5.8)$$

Où b_j pour $j=0..i$ sont les bits générés à partir du niveau 0 au niveau i . Par exemple, le coefficient 63 (commençant à partir du niveau 0) est approximé au niveau $i=1$ par :

$$T_0 + b_0 \cdot T_1 + b_1 \cdot T_2 + \frac{T_2}{2} = 32 + 1 \cdot 16 + 1 \cdot 8 + 4 = 60$$

Similairement, le coefficient 31 (commençant à partir du niveau 1) est approximé pour $i=1$ par:

$$\frac{T_0}{2} + \sum_{j=1}^i b_j T_{j+1} + \frac{T_{i+1}}{2} \quad (5.9)$$

$$\frac{T_0}{2} + b_1 T_2 + \frac{T_2}{2} = 16 + 1 \cdot 8 + 4 = 28$$

En général un coefficient commençant au niveau k jusqu'au niveau d'approximation i est calculé par la relation (5.10) :

$$\frac{T_0}{2^k} + \sum_{j=k}^i b_j T_{j+1} + \frac{T_{i+1}}{2} \quad (5.10)$$

La reconstruction est progressive et la phase de raffinement est assurée par l'addition des séquences S_i .

A la réception de S_0 par le décodeur, l'information récupérée est :

63	34	49	47	31	23	13	14	8	9	6	7	4	2	63	34
56	40	56	40	-	-	-	-	-	-	-	-	-	-	-	-

En ajoutant la séquence S_1 , l'information restituée est approximée par :

63	34	49	47	31	23	13	14	8	9	6	7	4	2	63	34
60	36	52	44	28	20	-	-	-	-	-	-	-	-	-	-

Par la suite, les bits "0" et "1" des listes S seront remplacés par "R" et "U" respectivement.

5.3.1.3. RLE et codeur arithmétique

Le flux de symboles préliminaires est codé utilisant le schéma suivant:

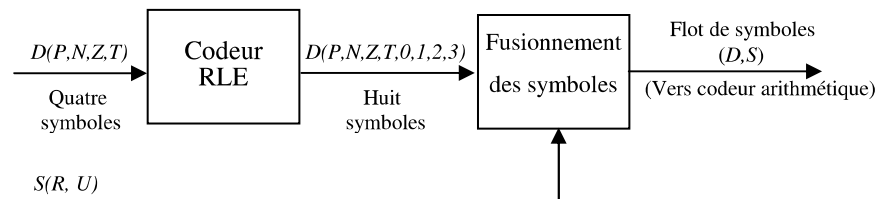


Figure 5.16 : Codeur de symboles

Notre approche distribue l'entropie différemment que l'algorithme de Shapiro [Sha93] par utilisation des quatre symboles employés dans la technique EZW et quatre autres symboles supplémentaires. Afin d'augmenter les performances de notre algorithme, un prétraitement (RLE) est réalisé avec la compression arithmétique. Le codeur RLE converti les symboles identiques consécutifs en un code. Ce dernier est composé du caractère et du nombre indiquant la longueur de la séquence en cours. Chaque symbole est codé en deux champs, le premier représente le nombre de répétitions et le deuxième champ

représente le caractère lui-même. Si le nombre de répétitions égale à 1 (pas de répétitions), le symbole est codé sur un seul champ qui représente le caractère. Cette phase est très simple et rapide pour l'implémenter. On note que le nombre de répétitions est converti en base 4. Dans ce cas, seulement les nombres 0, 1, 2 and 3 sont obtenus. Par conséquent, quatre nouveaux symboles viennent d'être ajoutés avec les quatre symboles existants.

On considère le flot de symboles issus du passage des dominants D_0 et D_1 de l'exemple de la figure 5.14 tels que :

$D_0 : NPZPPPTTTPPPPPZZZZ$ (18 symboles)

$D_1 : PZZTTTNPTTZZZZP$ (15 symboles)

A la sortie du codeur RLE on récupère les symboles suivants :

$D_0 : NPZ2P3T12P10Z$ (13 symboles)

$D_1 : P2Z3TNP2T10ZP$ (13 symboles)

Chaque séquence de la liste D est fusionnée avec le flot de bits de la liste S qui contient les symboles "R" et "U". On aura la séquence : $D_0 S_0 D_1 S_1 \dots$ Cette dernière est codée avec un codage rapide et efficace à l'aide du codeur arithmétique.

5.3.2. Résultats expérimentaux

Les bases de données sont disponibles dans [DB2], [DB3], [DB4] et [DB5]. Les signaux "Normal", "4hs", "Sumg" [DB2], "As-early", "Lateas", "Mr", "Ps", "Ar", "Ms" [DB3], "Late", "Split", "Dia", "S3", "S4", "NL" [DB4] et "A_regurg" [DB5] ont été utilisés. Leur résolution est de 16 bits pour chaque échantillon. Chaque signal possède sa propre fréquence d'échantillonnage. L'ondelette mère utilisée est "bior4.4" pour dix niveaux de décompositions. Les performances sont mesurées en utilisant les expressions (5.5) et (5.11).

$$PRD = \sqrt{\frac{\sum_{k=1}^n (X_{ori}[k] - X_{rec}[k])^2}{\sum_{k=1}^n (X_{ori}[k])^2}} \times 100 \quad (5.11)$$

Où X_{ori} et X_{rec} représentent le signal original et le signal reconstruit respectivement.

Pour évaluer les performances de notre technique, une étude comparative entre le codeur EZW original [Sha93], MEZW [Oua08] et EEZW (méthode proposée) utilisant le signal PCG "Normal" a été élaborée.

Dans [Oua08], Ouafi a développé la méthode MEZW afin de réduire la quantité totale d'information. Six symboles ont été utilisés par Ouafi au lieu des quatre symboles utilisés par l'algorithme EZW, les nouveaux symboles insérés sont "*Pt*" et "*Nt*". Cette méthode optimise le codage avec un groupement des éléments avant leur codage. Dans notre approche (EEZW), huit symboles (*P*, *N*, *T*, *Z*, *0*, *1*, *2* et *3*) ont été utilisés sans citer les deux symboles (*U* et *R*) récupérés de la liste des subordonnés.

Le Tableau 5.7 montre la fréquence d'apparition, la probabilité et l'entropie des symboles utilisés par les techniques citées. Dans notre approche, une redistribution des probabilités offre des résultats améliorés que les codeurs EZW et MEZW. La quantité totale d'informations obtenue par notre algorithme (28855 bits) est inférieure à celles obtenues par les algorithmes EZW (39688 bits) et MEZW (35583 bits), ce qui rend notre technique plus performante.

Sachant que l'expression de l'entropie du symbole *i* est donnée par :

$$H_i = -\log_2(p_i) \quad (5.12)$$

La quantité d'informations totale L est calculée par :

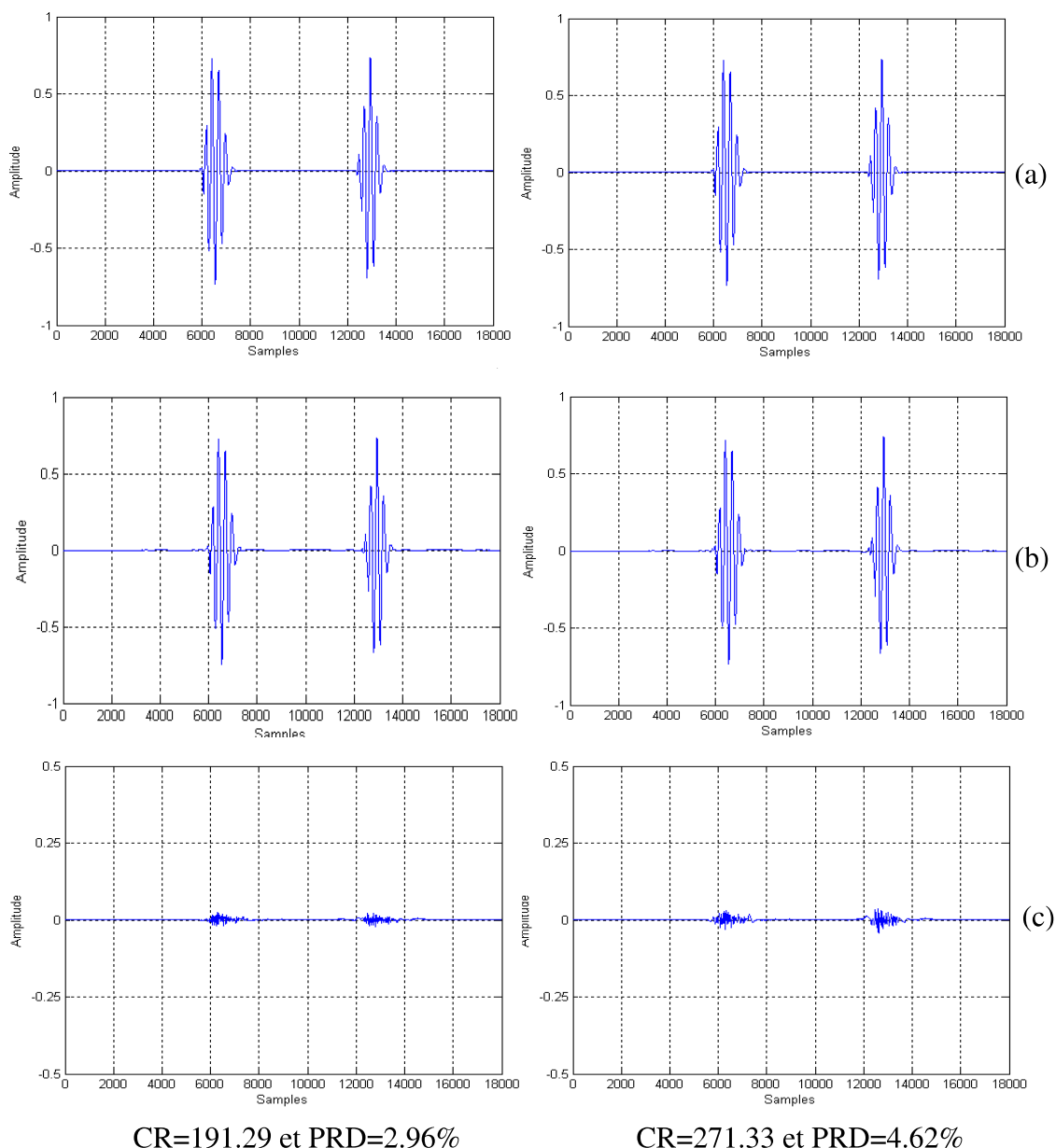
$$L = \sum_{i=1}^n f_i H_i \quad (5.13)$$

Tel que *n* représente le nombre total des symboles fournies par chaque technique. *n*=6, 8 et 10 pour les codeurs EZW, MEZW et pour EEZW respectivement.

Symbole	EZW			MEZW			EEZW		
	Fréquence d'apparition	Probabilité	Entropie	Fréquence d'apparition	Probabilité	Entropie	Fréquence d'apparition	Probabilité	Entropie
	F_i	p_i	H_i (bits)	f_i	p_i	H_i (bits)	$\bar{f}_i$	p_i	H_i (bits)
<i>T</i>	7602	0.4103	1.2851	4501	0.2918	1.7770	2691	0.2032	2.2989
<i>Z</i>	5100	0.2753	1.8610	5100	0.3306	1.5967	2101	0.1587	2.6560
<i>P</i>	789	0.0426	4.5534	51	0.0033	8.2406	649	0.0490	4.3508
<i>N</i>	789	0.0426	4.5534	53	0.0034	8.1851	645	0.0487	4.3597
<i>Pt</i>	-	-	-	738	0.0478	4.3855	-	-	-
<i>Nt</i>	-	-	-	736	0.0477	4.3894	-	-	-
<i>0</i>	-	-	-	-	-	-	283	0.0214	5.5482
<i>1</i>	-	-	-	-	-	-	665	0.0502	4.3156
<i>2</i>	-	-	-	-	-	-	1461	0.1103	3.1801
<i>3</i>	-	-	-	-	-	-	501	0.0378	4.7242
<i>U</i>	1920	0.1036	3.2704	1920	0.1245	3.0061	1920	0.1450	2.7859
<i>R</i>	2326	0.1256	2.9936	2326	0.1508	2.7293	2326	0.1757	2.5092
Information totale			Information totale			Information totale			
L= 39688 bits			L= 35583 bits			L= 28855 bits			

Tableau 5.7 : Fréquence d'apparition, probabilité et entropie des symboles utilisés par les algorithmes EZW, MEZW et EEZW appliqués au signal "Normal" pour un seuil Th = 500

La figure 5.17 montre les performances visuelles de la technique proposée utilisant le signal PCG “Normal”, qui compare le signal original avec deux signaux PCG reconstruits pour deux seuils différents.



(a) Echantillons du signal original “Normal” (b) Signaux reconstruits (c) signaux erreurs

Figure 5.17 : Compression du signal PCG “Normal” pour deux seuils

Les signaux reconstruits de la figure 5.17.b ont été obtenus avec : (CR=191.29:1 et PRD=2.96%) et (CR=271.33:1 et PRD=4.62%). Cependant, la figure 5.17.c présente les erreurs résultantes. Le rapport de compression obtenu est très élevé et la qualité de reconstruction est très bonne. Ce qui signifie que l’information clinique du signal reconstruit a été conservée.

Les figures 5.18 et 5.19 montrent les résultats quantitatifs des techniques EEZW, MEZW et EEZW. Notre approche prouve son efficacité significative par rapport aux deux autres techniques.

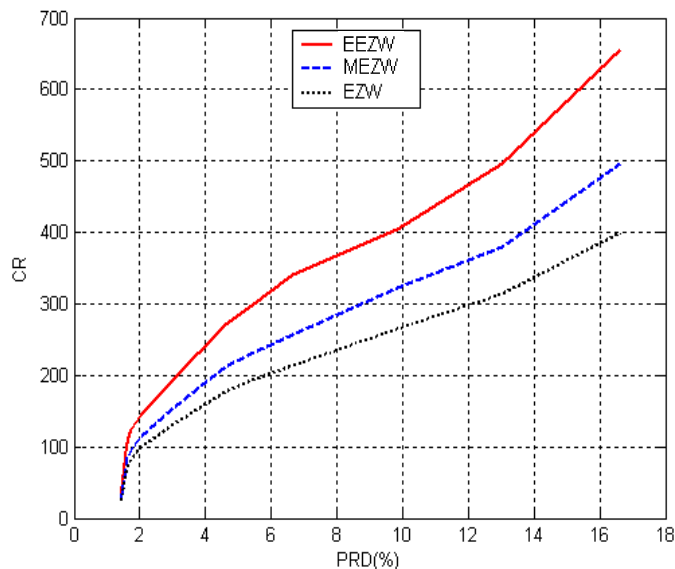


Figure 5.18 : Comparaison des différentes méthodes de compression appliquées au signal PCG “Normal”

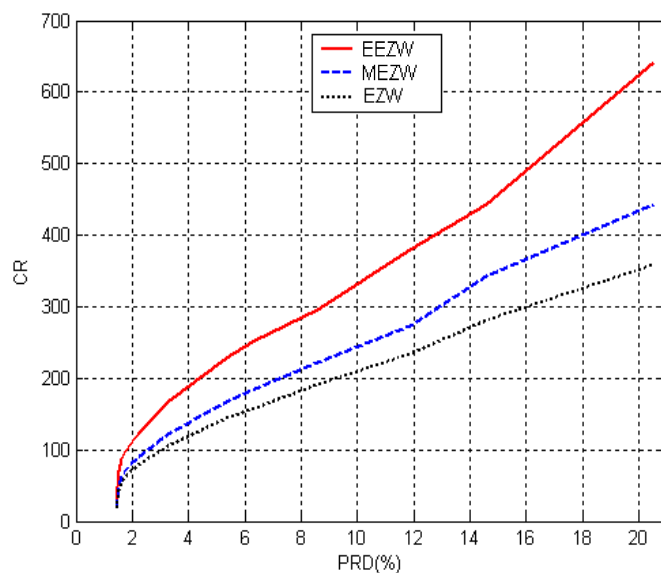


Figure 5.19 : Comparaison des différentes méthodes de compression appliquées au signal PCG “4hs”

Notre technique a été aussi comparée avec le codeur populaire MP3 (Motion Picture experts group audio layer 3) utilisant les signaux déjà cités. Les résultats obtenus dans le tableau 5.8 montrent que les CR obtenus par notre technique sont encore très élevés par

rapport à ceux du codeur MP3 pour un débit de 32kbps. Par exemple CR=167.35:1 et PRD=5.35% pour le signal PCG “S3” utilisant notre technique contre CR=5.30:1 avec le même PRD pour le même signal PCG utilisant le codeur MP3. Notre approche EEZW est très performante pour tous les signaux PCG cités. Cela confirme que notre approche dépasse largement la puissance du codeur MP3.

Enregistrement	Fréquence	EEZW	MP3 32kbps	PRD
PCG	d'échantillonnage	CR	CR	(%)
S3	11025 Hz	167.35	5.30	5.35
S4		192.97		5.43
NL		126.00		5.49
Dia		66.18		5.49
Late		67.60		5.80
Ar	22050 Hz	45.38	10.90	5.24
As_early		149.02		5.05
Lateas		130.01		5.19
Mr		56.15		5.31
Ms		140.84		5.07
Ps		87.43		5.15
Sumg		214.92		5.18
4hs		226.90		5.31
Normal	44100 Hz	286.32	21.97	5.08
A_regurg		117.76		10.43

Tableau 5.8 : Résultats expérimentaux pour quelques signaux PCG

Le processus de recherche des coefficients dans chaque intervalle dans chaque passage des subordonnés dans la technique EZW (détaillé dans le paragraphe 5.3.1.2.) nécessite un nombre considérable d'opérations de comparaisons pour fournir les symboles ("0" ou "1").

Dans notre approche, seulement avec un seul passage des subordonnés, le plan de bits (figure 5.15) constituée par des valeurs binaires des coefficients est généré en n'exigeant que moins de temps de traitement par rapport à l'algorithme EZW.

Différents tests ont été effectués pour évaluer la rapidité de notre algorithme. Ils ont été appliqués à 16384 échantillons pour les signaux PCG suivants : “Normal”, “S3”, “Late”, “Ps” et “A_regurg”. Les tests ont été exécutés par un Pentium4 3.4 GHz. Le temps moyen d'exécution utilisant l'algorithme EZW est entre 2 secondes et 5 secondes. Notre technique a besoin seulement de 1 à 2 secondes utilisant un seul passage des subordonnés.

5.4. Comparaison entre les techniques développées

Une étude comparative entre les trois techniques développées a été introduite. Le tableau 5.9 résume les résultats obtenus pour la compression des signaux ECG 103, 117 et 119 de la base de données [DB1].

Algorithme	103		117		119	
	CR	PRD	CR	PRD	CR	PRD
RR-DWT3	2.86	1.92	6.29	1.30	1.82	2.35
SPIHT-VKTP	5.00	1.36	8.11	1.18	4.00	1.42
EEZW	4.82	1.36	7.83	1.18	3.91	1.42

Tableau 5.9 : Résultats obtenus pour les signaux de tests 103, 117 et 119

Les résultats montrent que la technique à base de RR-DWT3 est efficace seulement si le nombre de faux positifs et le nombre de faux négatifs soient nuls ($D=0$) (voir le tableau 5.1 pour les cas de 100, 117 et 121).

L'approche SPIHT-VKTP dépasse légèrement les performances du codeur EEZW. On peut conclure que les performances du codeur EEZW sont presque équivalentes à ceux de l'algorithme de [Lu00].

La compression de 10 minutes d'enregistrements du signal 117 a été exécutée par le même processeur en utilisant l'approche EEZW nécessite environ trois minutes, par contre on a besoin d'environ quatre minutes pour SPIHT-VKTP. On peut dire aussi que le codeur EEZW est plus simple à implémenter.

5.5. Conclusion

Dans ce chapitre, nous avons testé les performances de nos algorithmes de compression et de décompression sur les signaux ECG et PCG en utilisant les ondelettes comme outil de transformation. Cette dernière permet d'améliorer considérablement le taux de compression tout en gardant une meilleure qualité de signal.

On remarque que plus le PRD est plus petit, plus la qualité de signal reconstruite sera la meilleure, c'est-à-dire le signal reconstruit est plus fidèle au signal original.

Notre approche EEZW est simple, rapide et efficace par rapport aux autres techniques développées et aussi par rapport à quelques techniques existantes dans la

littérature. On peut dire que le codeur EZW et ses variantes (MEZW et EEZW) prouvent leurs efficacités surtout dans la compression des signaux PCG.

Les différents tests et résultats obtenus dans ce chapitre nous ont permis de choisir l'une des trois techniques développées soit en termes d'efficacité, de simplicité ou de rapidité. On a constaté que notre approche EEZW a offert les avantages suivants: simple à implémenter, rapide, qualité et taux de compression élevés.

CONCLUSION GENERALE ET PERSPECTIVES

CONCLUSION GENERALE ET PERSPECTIVES

La compression des signaux ECG et PCG est un domaine de recherche étudié depuis des dizaines d'années. De très nombreuses techniques ont abordé cette problématique par un grand nombre d'algorithmes différents, comme nous l'avons vu à l'introduction générale. La majorité des algorithmes récents se concentrent sur utilisation des techniques de transformées dans le but de localiser le maximum d'énergie dans un nombre limitée de coefficients. Cependant, les ondelettes demeurent un outil essentiel qui permet de garder une très bonne localisation temporelle et fréquentielle. Les techniques de codage hiérarchiques par plan de bits, à savoir EZW et SPIHT, sont basées sur les transformées en ondelettes discrètes, ils sont classés parmi les techniques les plus performantes dans la compression des signaux ECG et PCG. Ces deux approches exploitent la relation entre les coefficients dans les différentes sous bandes. Ce qui permet d'effectuer un codage imbriqué des coefficients d'ondelettes. Avec une telle représentation progressive du signal, on peut avoir d'excellentes performances en termes de débit-distorsion par rapport à des codeurs bien plus complexes et non progressifs de l'époque.

Dans le cadre de cette thèse, nous avons proposé en premier lieu une méthode de compression des signaux ECG basée sur la détection des pics R utilisant l'algorithme de Pan et Tompkins [Pan85]. Ce qui a permis de former un volume constitué de milliers de voxels. Ces derniers représentent les échantillons du signal ECG. Une transformation d'ondelettes discrète 3D est ensuite utilisée. Finalement un seuillage, un quantificateur scalaire uniforme et un codeur RLE optimisé ont été appliqués. Cette technique apporte un gain supplémentaire comparable avec ceux des méthodes [Djo95], [Hil97] et [Tch07b] sous condition d'avoir un nombre de faux positifs et un nombre de faux négatifs nuls.

Notre deuxième méthode a été fondée sur la technique SPIHT adaptée aux signaux ECG et combinée avec l'encodeur VKTP. Les résultats ont été améliorés par rapport aux codeurs [Ben03], [Bla04], [Che08], [Lu00] et [Hil97], mais malheureusement, nous avons accru aussi la complexité (difficile pour l'implémentation et lente lors de l'exécution).

Notre nouvelle approche notée EEZW consiste à effectuer une redistribution des probabilités des symboles, c'est à dire utiliser huit symboles au lieu de quatre employés

dans l'algorithme de Shapiro, ce qui a permis de minimiser le nombre de symboles redondants d'une part et à optimiser le codage par développement d'une nouvelle variante RLE. Les résultats obtenus par cette nouvelle approche en termes de PRD et de taux de compression obtenus par Shapiro ont été améliorés surtout pour les moyens et hauts débits tout en réduisant le temps de calcul par utilisation d'un seul passage des subordonnés au lieu de plusieurs passages alternés (cas du codeur EZW).

La méthode EEZW a été appliquée aux signaux PCG et ECG. Elle a été comparée avec les deux techniques élaborées dans cette thèse et avec d'autres techniques récentes déjà existantes. Elle nous a permis d'obtenir de hautes performances (débit élevé, qualité conservée, vitesse de compression réduite). On peut conclure aussi que notre technique EEZW a largement dépassé la puissance du codeur MP3 le plus connu et le plus utilisé.

Finalement on peut dire que l'approche EZW et ses variantes MEZW et surtout EEZW sont très efficaces dans la compression des signaux PCG (rapports de compression très élevés avec de très bonnes qualités de reconstruction).

Les résultats de notre expérimentation sont très encourageants, on peut donc suggérer dans le futur les perspectives suivantes :

- ✓ Elaboration d'algorithmes permettant d'éliminer les fausses détections positives et négatives des ondes QRS.
- ✓ Implémentation de la technique EEZW sur des cartes à base de DSP ou FPGA.
- ✓ Adaptation et application de la technique EEZW dans la compression des signaux audio, images, images biomédicales volumétriques, images satellitaires, séquences vidéo.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [Add00] P. S. Addison, "Evaluating arrhythmias in ECG signals using wavelet transforms", *Engineering in Medicine and Biology Magazine (IEEE)*, Vol. 19, No. 5, pp. 104-109, 2000.
- [Ahm00] S. M. Ahmed, A. Al-Shrouf and M. Abo-Zahhad, "ECG data compression using optimal non-orthogonal wavelet transform", *Medical Engineering & Physics*, Vol. 22, pp. 39-46, 2000.
- [Ala07] J. M. Alajarín, J. L. Candel and R. R. Merino, "Classification and diagnosis of Heart Sounds and Murmurs Using Artificial Neural Networks", *Bio-inspired Modeling of Cognitive Tasks, Lecture Notes in Computer Science*, pp. 303-312, 2007.
- [Ant92] M. Antonini, M. Barlaud, I. Daubechies and P. Mathieu, "Image coding using wavelet transforms", *IEEE Trans. on Image Processing*, Vol. 1, No. 2, pp. 205-220, 1992.
- [Baa04] Z. E. Baarir et A. Ouafi, "Etude de la transformée en ondelettes dans la compression d'images fixes", *Courrier du Savoir*, No. 5, pp. 69-74, 2004.
- [Bea08] V. Beaudoin, "Développement de nouvelles techniques de compression de données sans perte", *Thèse de Doctorat*, Université Laval, Quebec, 2008.
- [Bea97] P. Beaurepaire, "Compression d'images appliquée aux angiographies cardiaques : aspects algorithmes, évaluation de la qualité diagnostique", *Thèse Doctorat*, Université de Technologie de Compiègne, 1997.
- [Ben03] R. Benzid, F. Marir, A. Boussaad, M. Benyoucef and D. Arar, "Fixed percentage of wavelet coefficients to be zeroed for ECG compression", *Electron. Lett.*, Vol. 39, pp. 830-831, 2003.
- [Ben08] R. Benzid, A. Messaoudi and A. Boussaad, "Constrained ECG compression algorithm using the block-based discrete cosine transform", *Digit. Signal Process.*, Vol. 18, pp. 56-64, 2008.
- [Ben09] M. S. Bensegueni, "Analyse et compression du signal ECG par méthodes utilisant la transformée en ondelettes", *Mémoire de Magister*, Université Mentouri Constantine, 2009.

- [Bena07] M. Benabdellah, “Outils de compression et de crypto compression : Applications aux images fixes et vidéo”, *Thèse Doctorat en Télécommunications et Multimédia*, Université de Mohammed Agdal, 2007.
- [Benz07] R. Benzid, F. Marir and N. E. Bouguechal, “Electrocardiogram compression method based on the adaptive wavelet coefficients quantization combined to a modified two-role encoder”, *IEEE Signal Processing Letters*, Vol. 14, pp. 373-376, 2007.
- [Bij95] A. Bijaoui and F. Rue, “A multiscale vision model adapted to the astronomical images”, *Elsevier Signal Process.*, Vol. 46, No. 3, pp. 345–362, 1995.
- [Bla04] M. Blanco-Velasco, F. Cruz-Roldan, J. I. Godino-Llorente and K. E. Barner, “ECG compression with retrieved quality guaranteed”, *Electron. Lett.*, Vol. 40, pp. 1466–1467, 2004.
- [Bou04] H. Boukerbout, “Analyse en ondelettes et prolongement des champs de potentiel, développement d'une théorie 3-D et application en géophysique”, *Edition Géosciences-Rennes*, 2004.
- [Bou09a] N. Boukhennoufa, K. Benmahammed, M. A. Abdi and F. Djeflal, “Wavelet-Based ECG signals compression using SPIHT technique and VKTP coder”, *The IEEE International Conference on Signals, Circuits and Systems*, Djerba, Tunisia, pp. 156-160, Nov, 6-8, 2009.
- [Bou09b] N. Boukhennoufa et K. Benmahammed, “Compression des signaux ECG utilisant la technique EZW adaptée”, *International Conference on Applied Informatics*, Bordj Bou Arréridj, Algeria, pp. 47-51, Nov, 2-4, 2009.
- [Bou09c] N. Boukhennoufa, K. Benmahammed et A. Bouchelaghem, “Application de la méthode de compression à encapsulation d’ondelettes par arbres de zéros aux images IRM”, *2nd International Conference on Electronics Systems CISE’09*, Batna, Algeria, Oct, 25-26, 2009.
- [Bou12] N. Boukhennoufa, K. Benmahammed and R. Benzid, “Effective PCG Signals Compression Technique Using an Enhanced 1-D EZW”, *International Journal of Advanced Science and Technology*, Vol. 48, pp. 89-102, 2012.

- [Cab08] A. Cabasson, “Estimation et analyse des intervalles cardiaques”, *Thèse Doctorat en Traitement du Signal et des Images*, Université Nice Sophia Antipolis, 2008.
- [Che08] D. Chen and S. Yang, “Compression of ECG signal using video codec technology-like scheme”, *Journal of Biomed. Science and Eng.*, Vol. 1, No.1, pp. 22-26, 2008.
- [Che98] J. Chen and S. Itoh, “A wavelet transform-based ECG compression method guaranteeing desired signal quality”, *IEEE Trans. Biomed. Eng.*, Vol. 45, pp. 1414–1419, 1998.
- [Chr06] E. Christophe, “Compression des images hyperspectrales et son impact sur la qualité des données”, *Thèse de Doctorat, Ecole Nationale Supérieure de l’Aéronautique et de l’espace*, Paris, 2006.
- [Coi91] R. R. Coifman et Y. Meyer, “Remarques sur l’analyse de Fourier à fenêtre”, *C. R. Acad. Sci. Paris*, pp. 259–261, 1991.
- [Coi93] R. R. Coifman, Y. Meyer, S. Quake and M.W. Wickerhauser, “Signal processing and compression with wavelet packets, Progress in wavelet analysis and applications”, *Ed. Frontières*, France, 1993.
- [Cox68] J. R. Cox, F. M. Nolle, H.A. Fozzard and G. C. Olivier, “AZTEC a processing program for real time ECG rhythm analysis”, *IEEE Trans. on Biomed. Eng.*, Vol. 15, pp. 128-129, 1968.
- [Czi99] A. Cziho, “Quantification vectorielle et compression d’image. Application à l’imagerie médicale”, *Thèse Doctorat*, Université de Rennes 1, 1999.
- [Dau92] I. Daubechies, “Ten lectures on wavelets”, *CBMS-NSF Regional Conference Series in Applied Mathematics*, 1992.
- [DB1] Arrhythmia Database, PhysioBank, physiologic signal archives for biomedical research
<http://www.physionet.org/physiobank/database/mitdb/>
- [DB2] University of Dundee, Ninewells Hospital & Medical School, Dundee, Scotland
<http://www.dundee.ac.uk/medther/cardiology/3hs.htm>
- [DB3] University of Washington, Department of medicine
<http://depts.washington.edu/physdx/heart/demo.html>

- [DB4] EGeneral Medical Inc. Raleigh, USA
<http://www.egeneralmedical.com/listohearmur.html>
- [DB5] Cardiothorasic Surgery of Savannah, Goeorgia
http://www.openheartsurgery.com/heart_sounds.html
- [Deb01] S. R. Debbal, J. Agzarian and D. Abbott, "Optimal wavelet denoising for phonocardiograms", *Microelectronics Journal*, Vol. 32, pp. 931-941, 2001.
- [Deb05] S. M. Debbal et F. Bereksi-Reguig, "Choix de l'ondelette analysante et classification des signaux phonocardiogrammes en fonction des souffles surajoutés", *Afrique Science*, Vol. 1, No. 1, pp.1-13, 2005.
- [Deb07] S. M. Debbal and F. Bereksi-Reguig, "Automatic measure of the split in the second cardiac sound by using the wavelet transform technique", *Computers in Biology and Medicine*, Vol. 37, pp. 269-276, 2007.
- [Deb08] S. M. Debbal and F. Bereksi-Reguig, "Computerized heart sounds analysis", *Computers in Biology and Medicine*, Vol. 38, pp. 263-280, 2008.
- [Del91] J. P. Delmas, "Eléments de théorie du signal : les signaux déterministes", *Ellipses, Collection Pédagogique de Télécommunication*, 1991.
- [Djo95] A. Djohan, T. Q. Nguyen and W. J. Tompkins, "ECG Compression using discrete symmetric wavelet transforms", *International Conference, IEEE Medicine and Biology Society*, pp. 167-168, 1995.
- [Ele07] M. D. M. Elena, J. M. Quero and I. Borrego, "Optimal selection of wavelet coefficients for electrocardiograph compression", *ETRI Journal*, Vol. 29, pp. 285 - 287, 2007.
- [Efi11] V. G. Efimov, "Elaboration of continuous wavelet transform parameters for problems of ultrasonic thickness gauging", *International Conference and Seminar of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, pp. 530-532, Jun, 30-Jul, 4, 2011.
- [Gau12] A. Gautam and M. Kaur, "ECG Analysis using Continuous Wavelet Transform (CWT)", *IOSR Journal of Engineering*, Vol. 2, No. 4, pp. 632-635, 2012.
- [Gra95] A. Graps, "An introduction to wavelets", *IEEE Institute of Electrical and Electronics Engineers*, Vol. 2, No. 2, 1995.

- [Gua06] Z. Guangzhi , “Using Wavelet Transform to Extract Seismic Attributes”, *8th International Conference on Signal Processing*, Vol. 4, Nov, 16-20, 2006.
- [Gue12] M. Guermoui, “Analyse spectrale du signal Phonocardiogramme (PCG)”, *Mémoire de Magister*, Université de Batna, 2012.
- [Ham04] J. R. Hampton, “L’ECG facile”, *Sixième édition*, Lavoisier, France, 2004.
- [Hea92] D. M. Healy, D. Coll and J. Weaver, “Two applications of wavelet transforms in magnetic resonance imaging”, *IEEE Trans. on Information Theory*, Vol. 38, No. 2, pp. 840-860, 1992.
- [Hil97] M. L. Hilton, “Wavelet and Wavelet Packet Compression of Electrocardiograms”, *IEEE Trans. Biomed. Eng.*, Vol. 44, pp. 394-402, 1997.
- [Huf52] D. A. Huffman, “A method for the construction of minimum-redundancy codes”, *In Proceedings of the Institute of Radio Engineers*, Vol. 40, pp. 1098–1101, 1952.
- [Ish83] M. Ishijima, S. B. Shin, G. H. Hostetter and J. Sklansky, “Scan-Along Polygonal Approximation for Data Compression of Electrocardiograms”, *IEEE Trans. on Biomed. Eng.*, Vol. 30, No. 11, pp. 723-729, 1983.
- [Ist00] R. S. H. Istepanian and A.A. Petrosian, “Optimal zonal wavelet-based ECG data compression for a mobile telecardiology system”, *IEEE Trans. on Information Technology in Biomedecine*, Vol. 4, No. 3, pp. 200-211, 2000.
- [Jon05] S. A. Jones, “ECG Notes: Interpretation and Management Guide”, *F. A. Davis Company, Philadelphia*, 2005.
- [Kha06] A. Khawaja, “Automatic ECG analysis using principal component analysis and wavelet transformation”, *Karlsruhe Transactions on Biomedical Engineering*, University of Karlsruhe, Allemagne, 2006.
- [Kha09] M. A. Khaled, “Réseaux de neurones appliqués à l’analyse et à la modélisation non linéaire du signal ECG”, *Mémoire de Magister*, Université de Constantine, 2009.
- [Kum06] V. Kumar, S. C. Saxena and V. K. Giri, “Direct data compression of ECG signal for telemedicine”, *International Journal of System Science*, Vol. 37, No. 1, pp. 45-63, 2006.

- [Kun82] M. Kunt, H. Rey and A. Lingtenberg, "Preprocessing of electrocardiograms by digital techniques", *Signal Processing, North-Holland Publishing Company*, 1982.
- [Lee99] H. Lee and K. M. Buckley, "ECG data compression using cut and align beats approach and 2-D transforms", *IEEE Trans. Biomed. Eng.*, Vol. 46, No. 5, pp. 556–564, 1999.
- [Lem78] A. Lempel and J. Ziv, "Compression of individual sequences via variable-rate coding", *IEEE Transactions on Information Theory*, Vol. 24, No. 5, pp. 530–536, 1978.
- [Lou97] P. M. A. K. Louis and A. Rieder, "Wavelets theory and applications", *John Wiley & Sons*, 1997.
- [Lu00] Z. Lu, D. Y. Kim and W. A. Pearlman, "Wavelet compression of ECG signals by the Set Partitioning in Hierarchical Trees Algorithm", *IEEE Trans. Biomed. Eng.*, Vol. 47, pp. 849-856, 2000.
- [Maa08] M. Maamoun, M. Neggazi, A. Meraghni and D. Berkani, "VLSI design of 2D discrete wavelet transform for area efficient and high speed image computing", world academy of science", *Engineering and Technology*, Vol. 45, pp. 538-543, 2008.
- [Mal89] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674-693, 1989.
- [Mar04] J. Martinez-Alajarin and R. Ruiz-Merino, "Wavelet and wavelet packet compression of phonocardiograms", *Electronics Letters*, Vol. 40, pp. 1040-1041, 2004.
- [Mar08] J. Martinez-Alajarin, J. Martinez-Rosso and R. Ruiz-Merino, "Encoding technique for binary sequences using vector tree partitioning applied to compression of phonocardiographic signals", *Electron. Lett.*, Vol. 44, pp. 84-85, 2008.
- [Mar95] X. Marsault, "compression et cryptage des données multimédia", *Traité des nouvelles technologies, Hermes*, Paris, 1995.
- [Mas07] A. Masmoudi et M. S. Bouhlef, "Un nouvel algorithme de compression exploitant le codage arithmétique en lui introduisant de nouveaux paramètres de codage", *Journées d'étude et d'échange*, Université Montpellier II, 2007.

- [Mat02] B. Matei, “Methodes multiresolutions non-lineaires applications au traitement d’image”, *Thèse de doctorat*, Université de Paris VI, 2002.
- [Mia02] S. G. Miaou and C. L. Lin, “A quality-on-demand algorithm for wavelet-based compression of Electrocardiogram signals”, *IEEE Trans. on Biomed. Eng.*, Vol. 49, No. 3, pp. 233-239, 2002.
- [Mor09] N. Moreau, “Outils pour la compression Application à la compression des signaux audio”, *Lavoisier*, 2009.
- [Mor94] N. Moreau, “Techniques de compression des signaux”, *Collection technique et scientifique des télécommunications*, Masson, Paris, 1994.
- [Nel95] M. Nelson and J. L. Gailly, “The Data Compression Book”, *John Wiley & Sons*; 2nd Edition, 1995.
- [Oua08] A. Ouafi, A. Taleb Ahmed, Z. Baarir and A. Zitouni, “A modified embedded zerotree wavelet (MEZW) algorithm for image compression”, *Journal of Math Imaging Vis*, Vol. 30, pp. 298–307, 2008.
- [Pan85] J. Pan and W. J. Tompkins, “A real time detection algorithm”, *IEEE Transactions on Biomed. Eng.*, Vol. 32, No. 3, pp. 230-236, 1985.
- [Pea98] W. A. Pearlman and A. Said. “Data compression using set partitioning in hierarchical trees”, *United States Patent 5,764,807*, 1998.
- [Poo05] M. Pooyan, A. Taheri, M. Moazami-Goudarzi and I. Saboori, “Wavelet compression of ECG signals using SPIHT algorithm”, *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 2, pp. 284-287, 2005.
- [Rab91] M. Rabbani and P. W. Jones, “Digital image compression techniques”, *Georgia Institute of Technology*, USA, 1991.
- [Rol00] D. L. Rollins, C. R. Killingsworth, G. P. Walcott, R. K. Justice, R. E. Ideker and W. M. Smith, “A telemetry system for the study of spontaneous cardiac arrhythmias”, *IEEE Trans. on Biomed. Eng.*, Vol. 47, No. 7, pp. 887-892, 2000.
- [Sah00] S. Saha, “Image Compression – from DCT to Wavelets: A review”, *Springer ACM Cross words students magazine*, Vol. 6, No. 3, 2000.
- [Sal04] D. Salomon, “Data compression, the complete reference”, *Springer 3. Edition*, New York, 2004.

- [Sav04] G. Savaton, E. Gasseau and E. martin, “Design of a flexible: reusable hardware component for the 2D discrete wavelet transform”, *CNRS Centre de Recherche UBS Rue de Saint Maudé*, Vol. 21, No. 2, pp. 157-178, 2004.
- [Sen00] N. Sendrier, “Introduction la théorie de l'information”, *Document cours, Université de Bourgogne –Département IEM*, 2000.
- [Sha48] C. Shannon, “A mathematical theory of communication”, *Bell System Technical, Journal*, 1948.
- [Sha93] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients”, *IEEE Trans. on Signal Processing*, Vol. 41, No. 12, pp. 3445-3462, 1993.
- [Sim05] V. Simard, “Transformée en ondelettes pour un système d'acquisition de signaux corticaux implantable”, *Thèse Maître ès sciences en Génie Électronique, École Polytechnique de Montréal*, 2005.
- [Sin10] S. Singh and N. N. Gandhi, “Pattern analysis of different ECG signal using Pan-Tompkin's algorithm”, *International Journal on Computer Science and Engineering*, Vol. 2, No. 7, pp. 2502-2505, 2010.
- [Som10] K.P. Soman, K.I. Ramachandran and N.G. Resmi, “Insight Into Wavelets: from theory to practice”, *Prentice-Hall Edition*, 3rd Edition, 2010.
- [Ste05] M. Steinbuch and M. J. G. van de Molengraft, “Wavelet Theory and Applications: A literature study”, *Eindhoven University of Technology*, 2005.
- [Tai05] S. C. Tai, C. C. Sun and W. C. Yan, “A 2-D ECG compression method based on wavelet transform and modified SPIHT”, *IEEE Trans. on Biomed. Eng.*, Vol. 52, No. 6, pp. 999–1008, 2005.
- [Tao04] T. Tao and A. Mukherjee, “LZW based compressed pattern matching”, *Proc. Data Compression Conference (DCC'04)*, pp. 568-576, 2004.
- [Taq11] J. Taquet, “Techniques avancées pour la compression d'images médicales”, *Thèse de Doctorat en Traitement du signal et Télécommunications, Université de Rennes1*, 2011.
- [Tau02] D. S. Taubman, “ Embedded block coding in JPEG 2000”, *Elsevier, Signal Processing: Image Communication*, Vol. 7, No. 1, pp. 49–72, 2002

- [Tau04] D. S. Taubman and M. Marcellin, “JPEG2000: Image Compression Fundamentals, Standards and Practice”, *The International Series in Engineering and Computer Science*, USA, 2004.
- [Tch07a] D. Tchiotsop, “Modélisations polynomiales des signaux ECG, Application à la compression”, *Thèse de Doctorat*, Institut polytechnique de Lorraine, 2007.
- [Tch07b] D. Tchiotsop, D. Wolf, V. Louis-Dorr and R. Husson, “ECG data compression using Jacobi polynomials”, *International Conference IEEE Eng. Med. Biol. Soc.*, Lyon, Aug, 23-26, 2007.
- [Tho05] J. Thornton, “Wavelet Packet Correlation Methods in Biometrics”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 81–84, Mar, 18-23, 2005.
- [Toh07] G. Tohumoglu and K. E. Sezgin, “ECG signal compression by multi iteration EZW coding for different wavelets and thresholds”, *Computers in Biology and Medicine*, Vol. 37, pp. 173–182, 2007.
- [Van11] K. Vanisree and J. Singaraju, “Automatic detection of ECG RR interval using discrete wavelet transformation”, *International Journal on Computer Science and Engineering*, Vol. 3, No. 3, pp. 1599- 1605, 2011.
- [Wel84] T. Welch, “A technique for high-performance data compression”, *IEEE Computer*, pp. 8–19, 1984.
- [Wic92] M. V. Wickerhauser, “Acoustic signal compression with wavelet packets”, *Ed. New York Academic*, Vol. 2, pp. 679–700, 1992.
- [Wu96] Y. Wu , “Nonparameter density estimation using wavelet transformation and scale-space zero-crossing reconstruction”, *3rd International Conference on Signal Processing*, pp. 319 - 322, Oct, 14-18, 1996.
- [Xia08] W. Xiang , “A study on the image processing of nano scale using wavelet”, *2nd IEEE International Conference on Nanoelectronics*, pp. 787-790, Mar, 24-27, 2008.
- [Xio98] Z. Xiong, K. Ramchandran and M. T. Orchard, “Wavelet packet image coding using space-frequency quantization”, *IEEE Trans. on Image Processing*, Vol. 7, No. 6, pp. 892-898, 1998.