



République Algérienne Démocratique et
Populaire



Ministère de l'enseignement Supérieur
et de la Recherche scientifique

Université Echahid Hamma Lakhdar El-Oued

Faculté de Technologie

Mémoire de Fin d'Études

En vue de l'obtention du diplôme de

MASTER ACADEMIQUE

Domaine : Technologie

Filière : Télécommunications

Spécialité : Systèmes de Télécommunications

Thème

**Optimisation de l'algorithme de compression d'image à
base de SPECK**

Réalisé par :

- *Hamidatou Laila*
- *Lemmadi Meriem Manel*

Soutenu en 16 juin 2019 devant le jury composé de :

Dr. GHENDIR Said

MCB

Président

Dr. CHEMSA Ali

MCA

Examineur

Dr. HETTIRI Messaoud

MCB

Rapporteur

Année Universitaire : 2018/2019

Mémoire de Fin d'Études

Remerciements

Avant tous nous remercions « ALLAH » tout puissant qui nous a donné la force d'avoir accomplir ce travail et d'être vécu jusqu'à ce jour.

*Ce mémoire achève notre étude de master à l'université **Echahid Hamma Lakhdar d'El Oued** , cela représente pour nous l'occasion d'exprimer nos reconnaissances pour les personnes que nous ont aides à arriver jusqu'à cette étape.*

*Nous tiens à remercier très chaleureusement **Dr. HETTIRI Messaoud** qui nous a permis de bénéficier de son encadrement .*

Nous remercions les membres de jury qui ont accepté de juger ce travail et d'y apporter leur caution .



Hamidatou Laila

Je tiens à témoigner ma profonde gratitude ainsi que mes plus vifs remerciements à mes chères parents pour leurs patience, leurs conseils judicieux et leurs encouragements et pour ses soutient moral.

Je remercie mes chères collègues Nesrine , Zaïz Arbia , Manel , pour leurs aides , et leurs soutient permanent à la réalisation de ce travail et tous mes amis.



Lemmadi Meriem Manel

Je remercie mes chères parents pour leurs aides précieuse et leurs soutient moral au cours de mes travaux de thèse.

Je tiens également à remercier mes enseignants du département de Génie Electrique et tous mes amis.

Que toute personne qui d'une manière ou d'une autre, nous a aidés et encouragés pour l'aboutissement de ce travail, trouve ici l'expression de notre sincère reconnaissance .

Merci à Tous ..



Dédicace

Je dédie ce modeste travail :

À mes chères parents

À mes chères sœurs

À toute ma famille

À tous mes amis

À tous que je connaisse de près ou de loin ..



Lemmadi Meriem Manel



Dédicace

À mes parents

*Mon père **Saad** le bien aimé et*

*Ma chère mère **Abid Nadia***

qui a fait preuve de beaucoup de patience et de sacrifice,

qui m'a encouragé tout au long de mes études.

À toute ma famille, sans lesquels, je ne serai pas ici aujourd'hui.

*À mes chères frères **Adnane** , **Akram** , **ALi** , **Nader** , et mes chères
sœurs **Basma** et **Sara** et nos petite fille **Rahma**.*

*À tous les personnes qui est toujours à mes cotés , mes chères amis
Arbia , **Nesrine** , **Radja** , **Samia** , **wafa** , **chaima**.*

*À mon honorable binôme **Lemmadi Manel**.*

À tous mes camarades de promotion M 2 Télécommunication.

*À tous ceux qui m'ont soutenu et m'ont encouragé depuis mes premiers
pas dans cette vie.*

À tous ceux qui me sont chers et que je n'ai pas cité.

*Je leurs dédie ce modeste travail, en espérant qu'il soit à la hauteur de
leurs attentes.*



Hamidatou Laila

Résumé

La contrainte du temps est un facteur essentiel dans l'évaluation des performances de toute méthode de compression, elle revient à calculer le temps pris par la compression et la décompression des images. Cette contrainte est plus au moins imposée selon l'application visée par la compression (transmission ou archivage).

Dans le cadre de ce travail, nous proposons une optimisation de l'algorithme de compression d'image SPECK "Set Partitioning Embedded bloCK". L'approche proposée est caractérisée par sa rapidité par rapport à SPECK original car il manipule moins des coefficients que la première approche.

Mots-clés : compression, décompression, transmission, archivage, SPECK.

Abstract

The constraint of latency is an essential factor in evaluating the performance of any compression method, and it refers to calculating the time taken by the compression and decompression of images. This constraint is more or less imposed by the application targeted by the compression (transmission or archiving).

In this work we propose an optimization of the image compression algorithm SPECK " Set Partitioning Embedded bloCK ". The proposed approach is characterized by its speed compared to original SPECK because it manipulates fewer coefficients than the first one .

Keywords : compression, decompression, transmission, archiving, SPECK .

ملخص

بعد تقييد الوقت عاملاً أساسياً في تقييم أداء أي طريقة ضغط ، وهو بمثابة حساب الوقت الذي يستغرقه ضغط الصور وإزالة ضغطها . يتم اللجوء الى هذا القيد حسب التطبيق الذي يستهدفه الضغط (الإرسال أو الأرشفة) .

في إطار هذا العمل نقتراح خوارزمية تحسين ضغط الصور SPECK . تتميز هذه الطريقة المقترحة بسرعتها مقارنةً بـ SPECK الأصلي كونها تعالج معاملات أقل من الطريقة الأولى .

الكلمات المفتاحية : الضغط ، إزالة الضغط ، النقل ، الأرشفة ، SPECK .

Table des matières

Liste des figures.....	iv
Liste des tableaux.....	vi
Abréviations	viii
Introduction générale.....	1

Chapitre I : Généralités sur la compression d'image

I.1. Introduction.....	3
I.2. Image numérique.....	3
I.2.1. Pixel	4
I.2.2. Image à niveaux de gris	5
I.2.3. Image en couleur	5
I.3. Compression d'images.....	6
I.3.1. Définition de la compression d'image.....	6
I.3.2. Schéma fonctionnel de la compression.....	6
I.3.2.1. Décorrélation.....	7
I.3.2.2. Quantification	7
I.3.2.3. Codage.....	8
I.4. Méthodes de compression	8
I.4.1. Compression sans pertes	9
I.4.1.1. Codage de Shannon-Fano.....	9
I.4.1.2. Codage de Huffman.....	9
I.4.1.3. Codage arithmétique.....	11
I.4.1.4. Méthode des plages.....	11
I.4.2. Compression avec pertes	12
I.5. Compression d'images fixes par ondelettes.....	12
I.5.1. Compression basée sur les approches classiques.....	12
I.5.1.1. Quantification scalaire.....	12

I.5.1.2. Quantification vectorielle.....	13
I.5.2. Compression basée sur la similarité des coefficients d'ondelettes..	13
I.5.2.1. EZW de Shapiro.....	13
I.5.2.2. SPIHT de Said et Pearlman	14
I.5.2.3. Algorithme SPECK d' Islam et Pearlman.....	15
I.5.2.4. Algorithme EZBC.....	15
I.6. Conclusion	16

Chapitre II : Transformée en ondelettes

II.1. Introduction	17
II.2. Transformée en ondelettes	17
II.3. Analyse multi résolutions.....	19
II.3.1. Principe.....	19
II.3.2. Banc de filtres et analyse multi résolutions.....	20
II.3.2.1. Cas d'ondelettes orthogonales.....	20
II.3.2.2. Cas d'ondelettes bi-orthogonales.....	21
II.3.3. Extension à deux dimensions de l'analyse multi résolutions.....	22
II.4. Conclusion.....	25

Chapitre III : Codeur SPECK original et optimisé

III.1. Introduction	26
III.2. Principe de l'algorithme SPECK	26
III.3. Etapes de l'algorithme SPECK.....	27
III.3.1. Initialisation	27
III.3.2. Etape de sortie de bits	27
III.3.3. Phase de raffinement	28
III.3.4. Étape de la quantification	29
III.4. Exemple d'application du SPECK.....	29
III.5. Algorithme SPECK modifié	31
III.6. Exemple d'application du SPECK modifié	32
III.7. Comparaison entre les deux codeurs SPECK et SPECK rapide	34

III.8. Conclusion.....	35
------------------------	----

Chapitre IV : Résultats et Discussions

IV.1. Introduction.....	36
IV.2. Paramètres de validation.....	36
IV.2.1. Taux de compression (TC).....	36
IV.2.2. Mesure de la qualité d'image compressée.....	36
IV.2.3. Temps de calcul.....	37
IV.3. Images de test.....	37
IV.4. Niveaux de décomposition.....	41
IV.5. Approche modifiée	41
IV.6. Etapes détaillées de l'algorithme	42
IV.6.1. Initialisation	42
IV.6.2. Sortie des bits	43
IV.7. Résultats	45
IV.7.1. Image Lena (taille 512 x 512)	45
IV.7.2. Image Baboon (taille 512 x 512)	46
IV.7.3. Image Barbara (taille 512 x 512)	47
IV.7.4. Image Peppers (taille 512 x 512)	48
IV.7.5. Image Cameraman (taille 256 x 256)	49
IV.7.6. Image Goldhill (taille 256 x 256)	50
IV.7.7. Image Woman (taille 256 x 256)	51
IV.7.8. Image Hoed (taille 256 x 256)	52
IV.7.9. Image Penny (taille 128x128)	53
IV.8. Discussions.....	54
IV.8.1. Comparaison entre SPECK et SPECK rapide.....	54
IV.8.2. Comparaison entre le temps de calcul pour SPECK et SPECK modifié avec ondelette.....	55
IV.9. Conclusion	57
Conclusion générale.....	58
Bibliographies.	

Liste des figures

Figure I.1 : Des images numériques.....	4
Figure I.2 : Représentation de pixel.....	4
Figure I.3 : Synthèse Additive.....	5
Figure I.4 : Schéma de compression / décompression classique d'images....	7
Figure I.5 : Principe de la quantification.....	7
Figure I.6 : Arbre binaire de Huffman.....	10
Figure I.7 : Quantification scalaire (QS).....	12
Figure I.8 : Exemples de descendances parent-fils dans le cas SPIHT.....	14
Figure II.1 : Exemple d'ondelette mère (Daubechies 20).....	18
Figure II.2 : Représentation multi-résolution de l'image Lena	19
Figure II.3 : Décomposition 1D à l'aide de bancs de filtres.....	20
Figure II.4 : Analyse et synthèse par filtres orthogonaux.....	21
Figure II.5 : Analyse multi-résolution en 2D (cas d'ondelettes bi-orthogonales).....	23
Figure II.6 : TOD 2-D d'une image d' empreinte palmaire.....	24
Figure III.1 : Partitionnement de l'image X en deux sous-ensembles.....	26
Figure III.2 : Principe de partitionnement des ensembles I.....	28
Figure III.3 : Algorithme de l'étape de raffinement.....	28
Figure III.4 : Exemple de matrice de 8×8 coefficients d'ondelettes.....	29
Figure IV.1 : image originale « Lena » 512 × 512 pixels.....	38
Figure IV.2 : image originale « Baboon » 512 × 512 pixels.....	38
Figure IV.3 : image originale « Barbara » 512 × 512 pixels.....	38
Figure IV.4 : image originale « Peppers » 512 × 512 pixels.....	39
Figure IV.5 : image originale « Cameraman » 256 × 256 pixels.....	39
Figure IV.6 : image originale « Goldhill » 256 × 256 pixels.....	39
Figure IV.7 : image originale « Woman » 256 × 256 pixels.....	40
Figure IV.8 : image originale « Hoed » 256x256 pixels.....	40
Figure IV.9 : image originale « Penny » 128x128 pixels.....	40

Liste des figures

Figure IV.10 : Organigramme de l'algorithme de compression considéré.	42
Figure IV.11 : Succession de partition des blocs du plus grand bloc jusqu'au plus petit bloc (taille de bloc = $2^r * 2^r$).	43
Figure IV.12 : Image Lena reconstruite 512×512 par SPECK modifié avec ondelettes (TC=95% et PSNR=35.79 dB).	45
Figure IV.13 : Image Baboon reconstruite 512×512 par SPECK modifié avec ondelettes (TC=95% et PSNR=24.15dB).	46
Figure IV.14 : Image Barbara reconstruite 512×512 par SPECK modifié avec ondelettes (TC=94% et PSNR=31.24dB).	47
Figure IV.15 : Image Peppers reconstruite 512×512 par SPECK modifié avec ondelettes (TC=95% et PSNR=34.6 dB).	48
Figure IV.16 : Image Cameraman reconstruite 256×256 par SPECK modifié avec ondelettes (TC=94% et PSNR= 30.45dB).	49
Figure IV.17 : Image Goldhill reconstruite 256×256 par SPECK modifié avec ondelettes (TC=91% et PSNR=30.39dB).	50
Figure IV.18 : Image Woman reconstruite 256×256 par SPECK modifié avec ondelettes (TC=91% et PSNR=34.72dB).	51
Figure IV.19 : Image Hoed reconstruite 256×256 par SPECK modifié avec ondelettes (TC=91% et PSNR=32.00 dB).	52
Figure IV.20 : Image Penny reconstruite 128×128 par SPECK modifié avec ondelettes (TC=95% et PSNR=34.86dB).	53

Liste des tableaux

Tableau I.1 : Symboles avec leurs fréquence d'apparition et le code de Huffman	10
Tableau III.1 : Exemple de codage d'image par l'algorithme SPECK	29
Tableau III.2 : Bits de sortie de test de signifiante selon la position du coefficient.....	31
Tableau III.3 : sortie de bits de codage de la matrice (figure III.4) par SPECK modifié.....	32
Tableau III.4 : Comparaison entre les deux codeurs SPECK et SPECK modifié.....	34
Tableau IV.1 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Lena de taille 512x512.....	45
Tableau IV.2 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Baboon de taille 512x512.....	46
Tableau IV.3 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Barbara de taille 512x512.....	47
Tableau IV.4 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Peppers de taille 512x512.....	48
Tableau IV. 5 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Cameraman de taille 256x256.....	49
Tableau IV.6 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Goldhilde de taille 256x256.....	50

Liste des tableaux

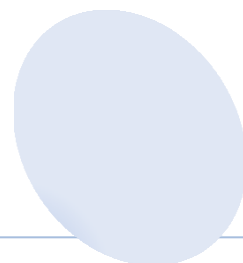
Tableau IV.7 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes et appliqués sur l'image Woman de taille 256x256.....	51
Tableau IV.8 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Hoed de taille 256x256.....	52
Tableau IV.9 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Penny de taille 128x128.....	53
Tableau IV.10 : (A) : Bloc de quatre coefficients ($C_i, C_{i+1}, C_{i+2}, C_{i+3}$), (B) : Bits de sortie de test de signifiante selon la position du coefficient signifiant dans le bloc de quatre coefficients dont trois insignifiants	54
Tableau IV.11 : Temps de calcul pour SPECK, SPECK modifié avec ondelettes sur l'image Barbara (512x512).....	55
Tableau IV.12 : Temps de calcul pour SPECK, SPECK modifié avec ondelettes sur l'image Peppers (512x512).....	55

Abréviations

2D	Deux Dimension.
AMR	Analyse M ulti- R ésolutions.
BF	B anc de F iltre.
DCT	D iscret C osine T ransform.
D	D iaagonale.
EZW	E Embedded Z ero –tree W avelet.
EZBC	E Embedded Z ero B locks coding based on C ontext modeling.
EQM	L' E rrreur Q uadratique M oyenne.
FC	F iltres C onjugués.
H	H orizontale.
JPEG 2000	J oint P hotographic E xperts G roup, Introduced in 2000 .
LSI	L iste de ensembles I nsignifiants.
LIP	L iste de coefficients I nsignifiants.
LSP	L iste de coefficients S ignifiants.
PSNR	P eak S ignal to N oise R atio.
QS	Q uantification S calaire.
QV	Q uantification V ectorielle.
RVB	R ouge , V ert , B leu.
RC	R atio C ompression.
SPIHT	S et- P artitioning H ierarchical T rees.
SPECK	S et- P artitioning E Embedded B lock.
TO	T ransformée en O ndelettes.
TC	T aux de C ompression.
V	V erticale.

INTRODUCTION

Générale



Introduction Générale

L'utilisation massive des images numériques engendre aujourd'hui des volumes de données de plus en plus importants. La compression de ces images numériques devient alors une nécessité afin d'assurer leur archivage d'une part et de faciliter leur transmission d'autre part .

Les méthodes de compression de ce type de données peuvent être classées suivant la nécessité de récupération parfaite ou non de l'information originale en deux grandes catégories La compression sans pertes d'information ou réversible qui a l'avantage de préserver la qualité de l'image originale, mais avec un taux de compression relativement faible et la compression avec pertes qui regroupe des algorithmes caractérisés par leur taux de compression assez élevé tout en gardant le mieux possible une acceptable qualité de l'image originale. Elle est basée généralement sur une phase de transformation qui sert à compacter l'information utile dans un nombre minimum de coefficients non nuls.

La transformation la plus courante est celle par ondelettes classiques , Cependant, elle présente des limites dans la représentation des régularités géométriques. Ceci étant dû au fait que les bases de transformée en ondelettes bidimensionnelle sont obtenues par le produit tensoriel de deux ondelettes unidimensionnelles appliquées suivant les lignes et les colonnes de l'image .

Il y a des algorithmes de compression par ondelettes dites codages imbriqués. Ces codeurs sont basés sur la notion d'arbre de zéros (zerotree) proposée par Shapiro en 1993 .. Nous citerons quelques algorithmes comme EZW (Embedded Zéro –tree Wavelet) , SPIHT (Set-Partitioning In Hierarchical Trees) et SPECK (Set-Partitioning Embedded bloCK) exploitent les corrélations entre les coefficients d'une image transformée en ondelettes. les deux premiers algorithmes sont basées sur les corrélations d'un groupe de coefficients coordonnées dans les différentes sous-bandes comme un ensemble nommé arbre . Par contre , le dernier est basé sur la partition successive de l'image transformée en ondelettes en blocs de tailles différentes selon les significances de

Introduction Générale

coefficients d'ondelettes .

Comparé à d'autres codeurs, l'algorithme SPECK est exploite des structures d'ensembles de coefficients non significatifs en blocs plutôt qu'en arbres. Ces structures de blocs permettent de s'affranchir efficacement du non stationnarité (d'ordre 1) des coefficients en adaptant localement la statistique utilisée pour le codage .

Pour atteindre les objectifs de ce travail, nous présentons ce mémoire subdivisé en quatre chapitres, chaque chapitre aborde une thématique bien définie.

Le premier chapitre décrit tout d'abord les principes généraux de la compression d'image (décorrélation, quantification et codage), ensuite nous décrivons la classification des méthodes de codage qui sont divisées en deux classes :

- Les méthodes avec pertes d'informations (quantification vectorielle, codage par transformée et codage imbriqué).
- Les méthodes sans pertes d'informations (codage de Shannon-fano, codage Huffman et codage arithmétique).

Dans le deuxième chapitre décrit la transformation en ondelettes et la transformation de Fourier et l'analyse multi-résolutions en particulier . L'idée étant de montrer son intérêt lorsqu'elle est utilisée pour le codage imbriqué .

Le troisième chapitre décrit l'algorithme SPECK avec un certain nombre d'exemples d'utilisation et détaille notre contribution appelée SPECK modifié , qui consiste en l'optimisation de l'algorithme SPECK .

Enfin, le quatrième chapitre sera consacré à l'évaluation de notre méthode en effectuant les différents tests sur des images références ainsi que le déploiement et l'analyse des différents résultats obtenus ainsi qu'une étude comparative en nous basons sur la métrique suivante : le taux de compression et le PSNR.

On termine notre thèse par une conclusion générale tout en soulignant les limites de notre algorithme et en discutant les perspectives ouvertes par ce travail.

Chapitre I

Généralités sur la compression d'image

I.1. Introduction

La compression est l'action utilisée pour réduire la taille physique d'un bloc d'information.

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 d'un réseau téléinformatique.

Parce que généralement les images requièrent une place importante, la compression est devenue part intégrante des données graphiques. Presque tous les formats de fichiers graphiques utilisent l'une ou l'autre méthode de compression. On rencontre souvent la compression de données comme étant une partie de l'encodage de données au même titre que le cryptage de données et la transmission de données [1].

I.2. L'image numérique

Une image numérique est une image dont le support est stocké sous forme binaire dans un fichier informatique. Celle-ci peut être obtenue soit à partir de capteurs optiques (appareil photo, caméra, scanner..) ou créée à partir de logiciels (Paintbrush, libre office...). Chaque image numérique est constituée d'un nombre donné de lignes. Chaque ligne comporte un nombre de point donnés. L'ensemble constitue une matrice. Ces points sont dénommés pixel (de l'anglais Picture élément et noté souvent px). Chaque « case » de cette matrice contient des nombres caractéristiques à la couleur attribuée au pixel [2] [3].



Figure I.1 : Des images numériques [3].

I.2.1. Pixel

En informatique et en photo numérique, une image est constituée d'un ensemble de points appelés pixels qui sont les plus petits éléments constitutifs d'une image numérique.

Le nom de "pixel", abrégé px, provient de l'expression anglaise Picture élément, qui signifie "élément d'image" ou "point élémentaire". Un pixel est généralement rectangulaire ou presque carré et présente une taille comprise entre 0,18 mm et 0,66 mm de côté [4].



Figure I.2 : Représentation de pixel [4].

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleur. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image.

Dans une image couleur (R V B), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B).

I.2.2. Image à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux Intermédiaires [5].

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux .

I.2.3. Image en couleur

La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle RVB) [6].

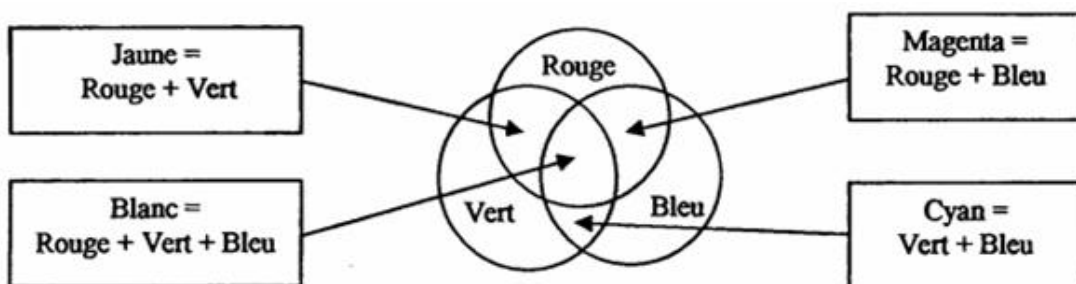


Figure I.3 : Synthèse Additive[6].

I.3. Compression d'image

I.3.1. Définition de la Compression d'image

La compression consiste à la réduction de la taille physique d'un bloc d'information (réduction de nombre de bit par pixel à stocker) en exploitant la redondance informationnelle dans l'image [7].

Trois sortes de redondances sont exploitées dans la compression d'image :

- La redondance spatiale entre pixels ou bloc voisins dans l'image.
- La redondance temporelle entre images successives dans une séquence vidéo.
- La redondance spectrale entre plans de couleur ou bandes spectrales [8].

Les principaux critères d'évolution de toute méthode de compression sont :

- La mesure de qualité : la qualité d'un système se mesure par la qualité de reconstruction de l'image.
- La rapidité du codeur et décodeur.
- La réduction des débits : le taux de compression.

I.3.2. Schéma fonctionnel de la compression

Nous donnons un schéma dont les éléments sont couramment utilisés dans la communauté scientifique de la compression d'images. Ce schéma général souvent utilisé pour décrire le fonctionnement des algorithmes de compression est présenté dans la figure I.4 . Les méthodes de compression d'images avec perte suivent les 3 étapes classiques de compression d'images naturelles. Elles débutent pour la plupart par une réorganisation du contenu de l'image, afin de séparer les composantes importantes (au sens visuel), des composantes contenant peu d'information. Cette tâche est remplie par une transformation mathématique. Cette étape est suivie par la quantification qui dégrade de manière irréversible le signal. La dernière étape de codage (sans perte) produit le flux binaire [9].

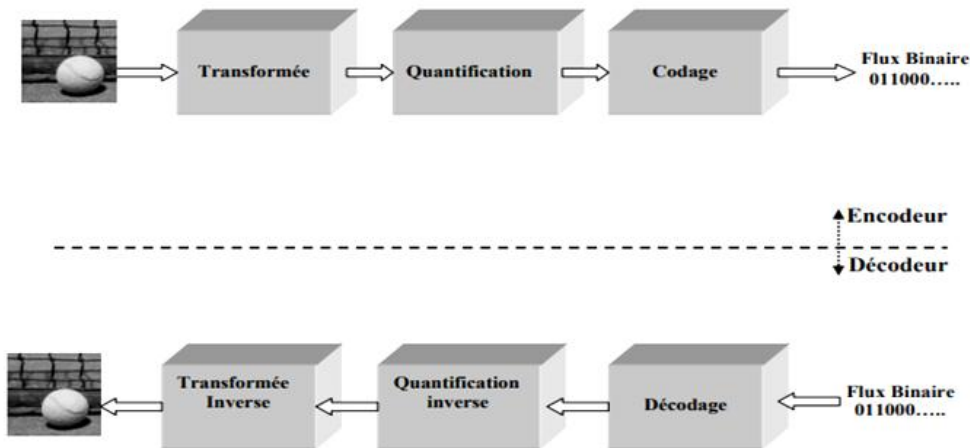


Figure I.4 : Schéma de compression / décompression classique d'images[9] .

I.3.2.1. Transformation ou dé-corrélation

La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte sur l'image. On essaie donc de tirer profit de cette corrélation, pour réduire le volume d'information en effectuant une opération de dé-corrélation des pixels.

La dé-corrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible [10].

I.3.2.2. Quantification

Cette étape permet de réduire le nombre de bits nécessaires à la représentation des coefficients issus de l'étape de transformation.

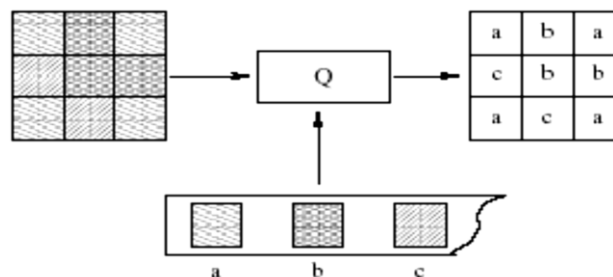


Figure I.5 : Principe de la quantification.

I.3.2.3. Codage entropique

C'est la phase finale du processus de compression, qui permet de générer le flot binaire à stocker ou à transmettre. Notons que le codage entropique est un codage sans pertes [11].

I.4. Méthodes de compression

Les méthodes de compression d'images sont divisées en deux classes, à savoir, les méthodes sans pertes et celles avec pertes . Le choix d'une méthode de compression dépend des images à traiter et de leurs entropies. Rappelons que l'entropie est une grandeur qui caractérise la quantité d'information que contient une image .L'entropie est donnée par l'équation:

$$H = - \sum_{i=0}^{N-1} P_i \log_2 P_i \quad [\text{bits}] \quad (\text{I.1})$$

où P_i est la probabilité de présence d'un niveau de gris i quelconque dans une image quantifié sur N niveaux .

A titre d'exemple, une image dont tous les pixels ont la même valeur (forte redondance) contient peu d'informations, son entropie est donc faible. Dans de tels cas, on utilise les méthodes sans pertes. A l'inverse, si la valeur de l'entropie est grande, on utilise les méthodes avec pertes .

Notons que dans certains cas, on utilise des méthodes hybrides qui consistent à coder différemment des zones déterminées de l'image en fonction de leurs intérêts. Il s'agit alors de coder les zones d'intérêts par des techniques sans pertes, et le reste de l'image par des techniques avec pertes .

Comme indiqué sur la figure I.4 , la seule étape qui différencie les méthodes de compression d'images avec pertes de celles sans pertes est l'étape de quantification qui n'intervient pas dans ces dernières. En effet, c'est le processus de quantification qui introduit les distorsions qui apparaissent sur les images décompressées [12].

I.4.1. Les méthodes sans pertes

Ces méthodes permettent après le processus de compression / décompression de reconstituer une image identique à l'image originale. L'avantage de ce type de chaîne est d'avoir une image reconstruite identique, mais l'inconvénient réside dans le faible taux de compression que l'on peut atteindre. Les méthodes sans pertes peuvent être employées directement dans une chaîne de compression.

I.4.1.1. Codage de Shannon-Fano

Il utilise des codes de longueur variable, comportant d'autant plus de bits que la probabilité du symbole est faible. Les codes sont définis à l'aide d'un algorithme spécifique selon un arbre de Shannon-Fano :

- Les symboles sont triés et classés en fonction de leur fréquence en commençant par le plus fréquent.
- La liste des symboles est ensuite divisée en deux parties de manière à ce que le total des fréquences de chaque partie soit aussi proche que possible.
- Le chiffre binaire 0 est affecté à la première partie de la liste, le chiffre 1 à la deuxième partie.
- Chacune des deux parties fait à son tour l'objet des démarches 2 et 3. Et ainsi de suite jusqu'à ce que chaque symbole soit devenu une feuille de l'arbre correspondant à un code déterminé.

Sauf dans des cas extrêmement simples, cette méthode ne permet pas d'approcher efficacement l'entropie. On lui préfère l'algorithme de Huffman [13].

I.4.1.2. Codage de Huffman

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

Le principe est le suivant :

- Les probabilités d'occurrence de chaque message sont placées dans une liste dans un ordre décroissant. Nous dirons que la liste est composée d'enfants.
- Les deux probabilités les plus faibles sont identifiées en fin de liste.
- La somme des deux probabilités est placée à sa place dans la liste triée. Elle constitue un nœud parent. Les deux enfants sont retirés de la liste.
- Le chemin « enfant de plus faible probabilité, parent » est codé par un 1, l'autre par un 0.

La procédure reprend à l'étape 2 jusqu'à ce qu'il ne reste plus qu'une probabilité dans la liste [14] .

Exemple

Soit un message a codé "ABBBBAAC". La fréquence d'apparition ainsi que le code Huffman correspondant est donné dans le tableau suivant est représentés par la figure I.6 :

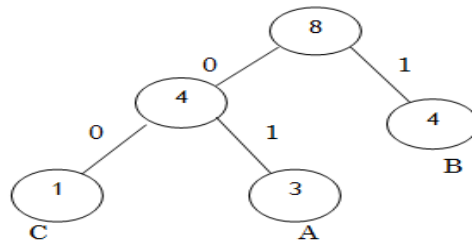


Figure I.6 : Arbre binaire de Huffman .

Symbole	Fréquence d'apparition	Code Huffman
a	3	01
b	4	1
c	1	00

Tableau I.1 : Symboles avec leurs Fréquence d'apparition et le code de Huffman.

I.4.1.3. Codage arithmétique

Contrairement à l'algorithme de Huffman qui associe aux symboles des motifs binaires dont la taille dépend de leur distribution. Le codeur arithmétique traite le vecteur de données dans son ensemble, en lui associant un unique nombre décimal rationnel varie entre 0 et 1 [15].

❖ Procédure de codage arithmétique

- Calculer la probabilité associée à chaque symbole dans la chaîne à coder.
- Associer à chaque symbole un sous intervalle proportionnel à sa probabilité, dans l'intervalle [0,1] (l'ordre de rangement des intervalles sera mémorisé car il est nécessaire au décodeur).
- Initialiser la limite inférieure de l'intervalle de travail à la valeur 0 et la limite supérieure à la valeur 1.

Tant qu'il reste un symbole dans la chaîne à coder :

- $\text{largeur} = \text{limite supérieure} - \text{limite inférieure}$.
- $\text{limite inférieure} = \text{limite inférieure} + \text{largeur} * (\text{limite basse du sous intervalle du symbole})$.
- $\text{limite supérieure} = \text{limite inférieure} + \text{largeur} * (\text{limite haute du sous intervalle du symbole})$.
- La limite inférieure code la chaîne de manière unique.
- On remarque que le premier symbole de la chaîne fixe est le premier chiffre après la virgule du code final [16].

I.4.1.4. Méthode des plages

Lorsqu'on considère une ligne de la matrice représentant une image numérique, plusieurs échantillons successifs sur cette ligne peuvent posséder la même valeur. L'ensemble de ces échantillons est appelé "plages". Cette méthode consiste donc à décrire les suites des pixels identiques par leurs longueurs et leurs valeurs. Par exemple, une plage de vingt pixels noirs équivaut à la donnée de 2 nombres : 20 et 0.

I.4.2. Les méthodes avec pertes

Lors de la phase de quantification, des modifications sont apportées aux valeurs de l'image initiale. L'avantage de ce type d'approche est qu'il est possible d'atteindre des taux de compression importants, mais au détriment de la qualité de l'image reconstruite. Cependant, la majorité des applications grand public s'est orientée vers ce type de compression : appareil photo numérique , images naturelles , transmission d'images sur les différents réseaux, stockage d'images, etc ... [17].

I.5. Compression d'images fixes par ondelettes

I.5.1. Compression basée sur les approches classiques

I.5.1.1. Quantification scalaire (QS)

La quantification scalaire est l'approximation de chaque valeur du signal aléatoire $\mathbf{x}(t)$, par une valeur \mathbf{q} qui appartient à un ensemble fini de codes $\{y_1, y_2, \dots, y_l\}$. A toute amplitude \mathbf{x} comprise dans l'intervalle $[x_{i-1}, x_i]$, on fait correspondre une valeur quantifiée y_i située dans cet intervalle. On l'appelle le pas de quantification Δ . Il représente l'écart entre chaque valeur \mathbf{q} .

Arrondir la valeur \mathbf{x} provoque une erreur de quantification, appelé le bruit de quantification [18] .

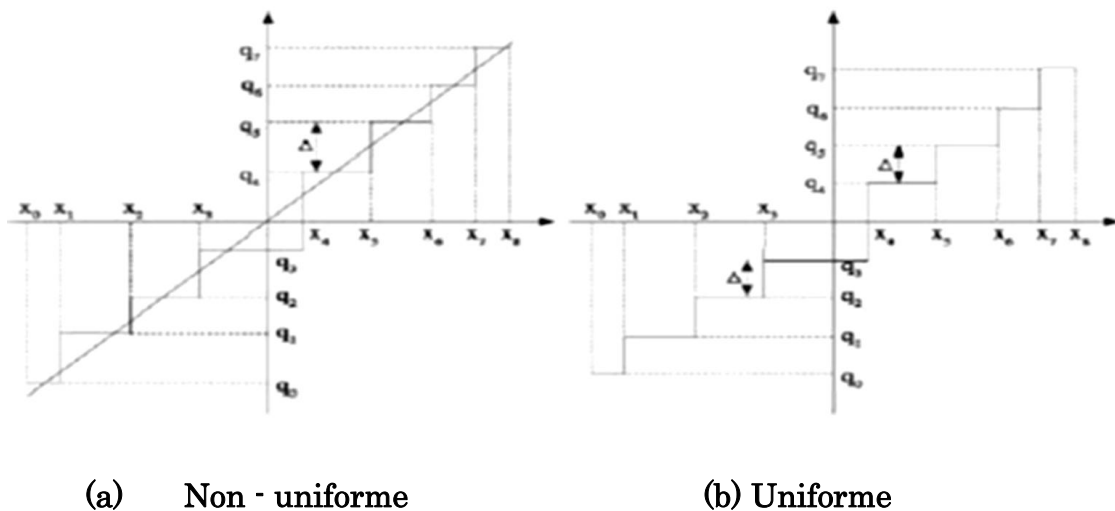


Figure I.7 : Quantification scalaire[18] .

I.5.1.2. Quantification vectorielle (QV)

La quantification scalaire associe à une variable continue, une variable discrète pouvant prendre un nombre plus faible et fini de valeurs. Ces valeurs ne sont jamais totalement décorrélées, ou indépendantes. Shannon a montré qu'il est toujours possible d'améliorer la compression de données en codant des vecteurs plutôt que des scalaires.

La compression des images fixes par ondelettes utilisant la quantification scalaire est rapide par rapport à la quantification vectorielle, mais cette dernière donne un meilleur taux de compression vu sa grande capacité de mieux décorréler les coefficients d'ondelettes [19].

I.5.2. Compression basée sur la similarité des coefficients d'ondelettes

I.5.2.1. EZW de Shapiro

C'est le premier codeur en sous-bande par 'zerotree' à avoir été introduit par Shapiro. Il procède au regroupement des coefficients non significatifs sous forme d'arbre de zéros (zerotree). La structure zerotree permet de détecter les zones de l'image qui ne contiennent pas d'information significative et sont codées ensuite en arbre.

On définit un niveau de seuil. Un coefficient d'ondelettes est dit significatif si son amplitude est plus grande que ce seuil. la valeur initiale du seuil est déterminée par :

$$T_0 = 2 \lceil \log_2 (C_{max}) \rceil \quad (I.2)$$

Où C_{max} est l'amplitude maximale des coefficients d'ondelettes. on construit alors une liste de coefficients significatifs et une liste de coefficients non significatifs. A chaque passe, on code les coefficients significatifs. les bits non significatifs deviennent les bits de départ de la seconde passe sur lesquels un nouveau calcul de seuil est appliqué par division par deux du seuil précédent, et ainsi de suite par récurrence jusqu'au codage globale déterminé par un nombre maximal de passes.

L'algorithme EZW à été ensuite amélioré par l'algorithme SPIHT et SPECK [20].

I.5.2.2. SPIHT de Said et Pearlman

L'algorithme SPIHT, proposé par Said et Pearlman est une amélioration du schéma de codage par l'algorithme EZW. Il repose sur les mêmes concepts : codage progressif par plan de bits et utilisation des dépendances hiérarchiques entre les coefficients des différents sous-bandes. Cependant, un nouveau protocole de dépendance entre les coefficients est défini. Les coefficients de la sous-bande de plus basse fréquence (A2 en figure I.8) sont regroupés par quatre, et pour chaque groupe, la descendance est comme suit :

Un des quatre coefficients (marqué par '*' en figure I.8) n'admet pas de descendants, alors que les trois autres (en gris en figure I.8) ont quatre descendants chacun. Pour les autres sous-bandes, comme dans le cas de l'EZW, chaque coefficient admet quatre descendants.

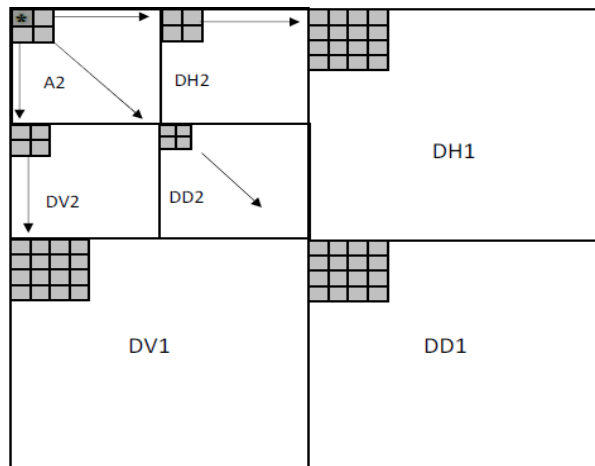


Figure I.8 : Exemples de descendance parent-fils dans le cas SPIHT, le pixel désigné par (*) n'accepte pas de descendants.

Trois ensembles de coordonnées sont utilisés pour présenter la nouvelle méthode de codage :

- $D(i,j)$: Ensemble de tous les descendants du pixel (i,j) .
- $O(i,j)$: Ensemble de coordonnées des descendants directs du pixel (i,j) .
- $L(i,j)$: Ensemble défini comme étant $D(i,j)-O(i,j)$.

❖ Algorithme

L'algorithme est cependant plus sophistiqué : contrairement à l'algorithme EZW qui n'utilise qu'un seul ensemble décrivant la signifiante des coefficients, le SPIHT utilise trois listes :

- LIS : Liste des ensembles insignifiants.
- LIP : Liste des coefficients insignifiants.
- LSP : Liste des coefficients signifiants.

Les entrées de chacune des listes sont les coordonnées (i,j) dans la matrice image, qui représentent des pixels individuels dans le cas des listes LIP et LSP et l'ensemble de descendants $D(i,j)$ ou $L(i,j)$ dans le cas de la liste LIS. L'algorithme SPIHT permet de générer directement un flux binaire à la place des symboles de signifiante utilisés dans l'EZW. Cet algorithme, même dans sa version non codée entropiquement, dépasse les performances obtenues par l'EZW [21] .

I.5.2.3. Algorithme SPECK d'Islam et Pearlman

Offrant des performances comparables à l'algorithme SPIHT , l'algorithme SPECK exploite des structures d'ensembles de coefficients non significatifs en bloc plutôt qu'en arbres. Ces structures de blocs permettent de s'affranchir efficacement de la non-stationnarité (d'ordre 1) des coefficients en adaptant localement la statistique utilisée pour le codage [22].

I.5.2.4. Algorithme EZBC

Le principe de codage par l'algorithme EZBC (Embedded ZeroBlocks coding based on Context modeling), est similaire à l'algorithme SPECK. L'innovation de ce codeur provenant principalement de l'exploitation de la dépendance entre les noeuds du quad-tree de signifiante[23].

Le partitionnement en quad-tree est de plus réalisé indépendamment dans chaque sous-bande permettant une meilleure séparation des statistiques de signifiante et un apprentissage plus efficace à l'aide de contextes plus étendus [24] .

I.6. Conclusion

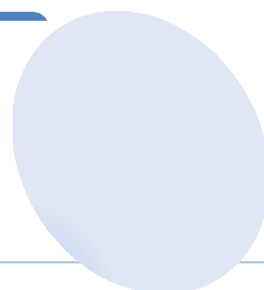
Le but de ce chapitre était d'introduire les notions générales de l'image numérique et son processus de compression. Il cherche aussi parfois à comprimer l'image, afin de gagner en vitesse lors de la transmission de l'information, et en capacité de stockage, tout en dégradant le moins possible les images considérées.

Enfin, on explique quelques méthodes de codages qui se basent sur la transformation en ondelettes qui gagné un intérêt considérable ces dernières années. Cette dernière offre une plus grande finesse au niveau de l'analyse du signal et permet de mieux s'adapter aux propriétés locales de l'image. Les codeurs d'images basés sur la transformation en ondelettes (JPEG2000, EZW, SPIHT, SPECK ...etc.), permettent d'avoir des résultats meilleurs.

Dans le chapitre suivant, nous étudierons la base des ondelettes que permettent en effet d'obtenir simultanément une qualité de compression supérieure aux autres méthodes et un taux de compression progressif.

Chapitre II

Transformée en ondelettes



II.1. Introduction

La représentation efficace de l'information visuelle est au cœur de nombreux problèmes en traitement d'images incluant la compression, le filtrage, et l'extraction de primitives pertinentes. Par représentation efficace on entend la description compacte et précise, par un nombre restreint d'échantillons, de l'information significative contenue dans une image . En effet, une transformée efficace peut capturer l'essence d'un signal donné par un nombre minimal de fonctions de base.

En pratique, cette description doit être générée par des transformations structurelles et des algorithmes rapides . La transformée en ondelettes , récemment intégrée dans le standard de codage d'images JPEG2000, constitue un exemple typique pour ces représentations. Les ondelettes sont utilisées dans les deux catégories de techniques de compression que sont la compression sans perte et avec perte .

Dans ce chapitre , Il existe autres transformations capables de fournir des décompositions multi-résolution mieux adaptées à la représentation des images .

II.2. Transformée en Ondelettes (TO)

Une ondelette est une fonction qui oscille comme une onde mais qui est rapidement atténuée d'où son nom ondelette qui veut dire petite onde. Elle est localisée à la fois en temps et en fréquence et permet de définir par translation en temps et dilatation en échelle, une famille de fonction analysantes. Les ondelettes sont considérées comme un "zoom" mathématique permettant de décrire les propriétés d'un signal à plusieurs échelles de temps simultanément [25] .

D'une façon plus formelle, une fonction $\Psi(\mathbf{t})$ est appelée ondelette si elle vérifie la condition suffisante d'admissibilité :

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \quad (\text{II.1})$$

On peut obtenir une famille d'ondelettes formant une base orthogonale de $L^2(\mathbb{R})$:

$$\Psi_{m,n}(x) = \frac{1}{\sqrt{2^m}} \Psi\left(\frac{1}{2^m}x - n\right) \quad (\text{II.2})$$

Où l'indice m représente un facteur d'échelle et l'indice n est un facteur de translation.

Par conséquent, tout signal $s(x)$ pourra être décomposé sur cette base :

$$s(x) = \sum_m \sum_n d_{m,n} \Psi_{m,n}(x) \quad (\text{II.3})$$

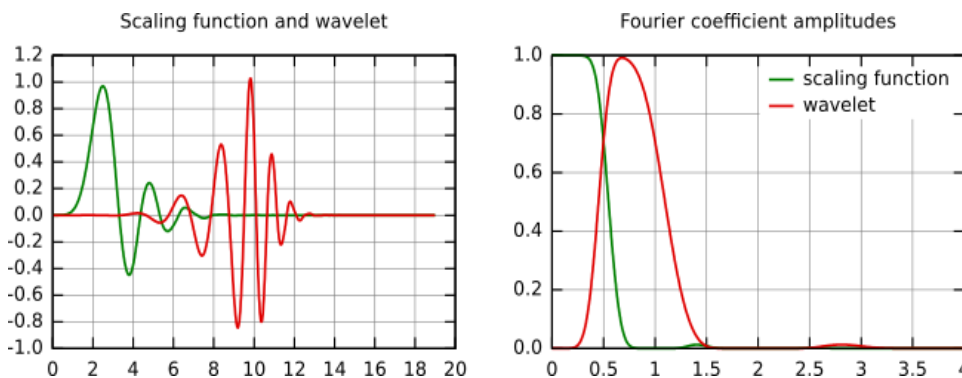
Avec :

$d_{m,n}$: sont les coefficients d'ondelettes obtenus par la discrétisation de la fonction suivante :

$$d_{m,n} = \int s(x) \Psi_{m,n} dt \quad (\text{II.4})$$

L'intérêt d'une décomposition est que les fonctions $\Psi_{m,n}$ possèdent la propriété remarquable d'être à la fois bien localisées en temps et en fréquence (filtre passe bande à support temporel étroit).

Ainsi , le coefficient " m " correspond à la résolution d'analyse , tandis que " n " représente la localisation temporelle (figure II.1).



(a) fonction d'échelle et ondelette Daubechies (b) leur contenu fréquentiel

Figure II.1 : Exemple d'ondelette mère (Daubechies 20).

II.3. Analyse multi-résolutions (AMR)

II.3.1. Principe

Pour une ondelette mère $\Psi(x)$ donnée, on peut trouver une fonction particulière $\varphi(x)$ dite fonction d'échelle, ayant les propriétés d'un filtre passe-bas, et telle que l'approximation $S_m(x)$ du signal $s(x)$ à la résolution "m" s'écrive [26]:

$$S_m(x) = \sum_n a_{m,n} \varphi_{m,n}(x) \quad (\text{II.5})$$

Et $a_{m,n}$ sont les coefficients d'approximation obtenus par la discrétisation de la fonction suivante :

$$a_{m,n} = \int s(x) \varphi_{m,n}(x) dx \quad (\text{II.6})$$

Où :

$$\varphi_{m,n}(x) = \frac{1}{\sqrt{2^m}} \varphi\left(\frac{1}{2^m} x - n\right) \quad (\text{II.7})$$

Un exemple de représentation multi-résolution d'une image est donné par la figure suivante (II.2) :



Figure II.2 : Représentation multi-résolution de l'image Lena à la résolution trios .

II.3.2. Banc de filtres (BF) et analyse multi-résolutions

Le but de la représentation multi-résolution est qu'elle peut être calculée à partir d'une transformée pyramidale mise en œuvre à l'aide de filtres numériques. le principe de la transformée pyramidale consiste en la décomposition du signal à analyser à l'aide d'une paire de filtres (g et h), l'un de ces filtres (g) fournira les coefficients d'ondelettes (ou détails), le second (h) les coefficients d'approximation [27].

Les fonctions de transfert des filtres " g " et " h " étant calculées à partir des transformée de Fourier des fonctions $\Psi(x)$ et $\phi(x)$:

$$\phi(2w) = H(w) \phi(w) \quad \text{et} \quad \Psi(x) = G(w) \phi\left(\frac{w}{2}\right) \quad (\text{II.8})$$

Avec : $H(w) = \text{TF} \{ k(x) \}$ et $G(w) = \text{TF} \{ g(x) \}$

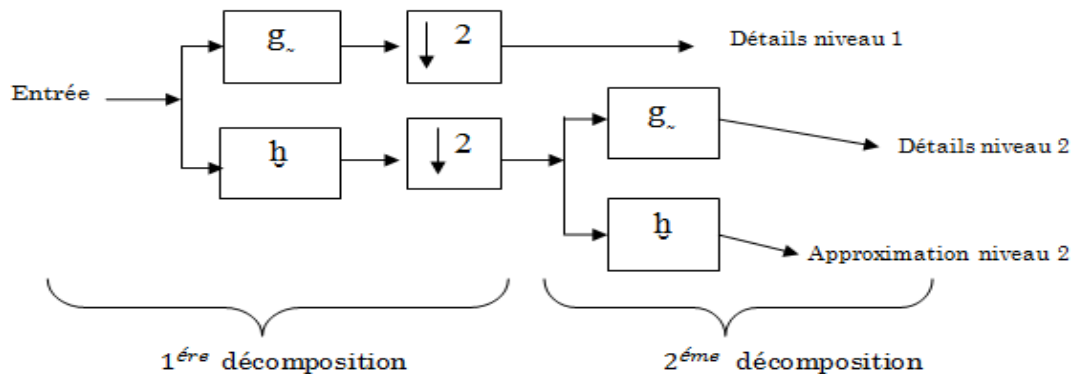


Figure II.3 : Décomposition 1D à l'aide de bancs de filtres.

II.3.2.1. Cas d'ondelettes orthogonales

Soit V un espace vectoriel de dimension n et $\{\phi_j / j=1..,n\}$ une base de cette espace. Cette base est qualifiée de orthogonale si est seulement si [28] :

$$\langle \phi_j, \phi_i \rangle = 0 \quad \forall \quad i \neq j ; i, j = 1, \dots, n$$

Normée si est seulement si :

$$\langle \phi_j, \phi_j \rangle = 1 \quad \forall \quad j = 1, \dots, n$$

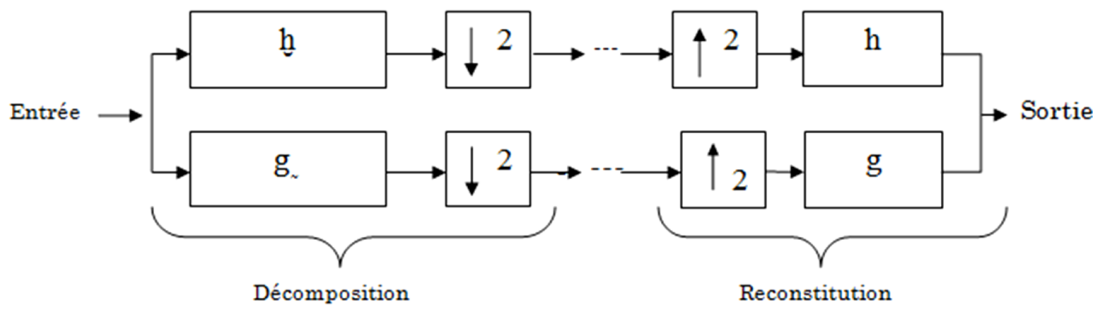


Figure II.4 : Analyse et synthèse par filtres orthogonaux.

La reconstitution est obtenue de la manière suivante :

$$S_{m-1}(n) = \sum_k h(2n-k) S_m(k) + \sum_k g(2n-k) d_m(k) \quad (\text{II.9})$$

Où : S représentent l'approximation et d représentent les détails .

II.3.2.2. Cas d'ondelettes bi-orthogonales

Deux familles de fonctions duales sont considérées, l'une sera utilisée pour l'analyse (transformée directe) et l'autre pour la reconstruction ou synthèse (transformée inverse). Ces familles sont orthogonales entre elles mais ne sont pas orthogonales en elles-mêmes.

Nous avons besoin de deux familles duales de sous-espaces emboîtés définies par les deux familles de fonctions d'échelle, chacune formant une base non orthogonale. Les propriétés à respecter sont l'indépendance linéaire et l'existence et l'unicité de la décomposition. Si on note V_j et \tilde{V}_j les deux familles de sous-espaces, on a [29] :

$$V_j \subset V_{j-1} \quad \text{et} \quad \tilde{V}_j \subset \tilde{V}_{j-1}$$

Les sous-espaces complémentaires sont définis différemment car ils sont complémentaires au sous-espace dual :

$$W_j \perp \tilde{V}_j \quad \text{et} \quad W_j \sim \perp V_j$$

Dans le cas bi-orthogonal, l'analyse est effectuée de la même manière que dans le cas orthogonale :

$$S(j,k) = \sum_n h(n-2k) s(j-1,n) \quad (\text{II.10})$$

$$d(j,k) = \sum_n g(n-2k) s(j-1,n) \quad (\text{II.11})$$

Par contre, la reconstruction est assurée par les filtres conjugués (FC), soit :

$$s(j-1,k) = \sum_{l \in \mathbb{Z}} \check{h}(n-2k) s_{j,l} + \sum_{l \in \mathbb{Z}} \check{g}(n-2k) d_{j,l} \quad (\text{II.12})$$

La relation imposée entre les filtres étant :

$$\check{g}_n = (-1)^n h_{1-n} \quad \text{et} \quad g_n = (-1)^n \check{h}_{1-n} \quad (\text{II.13})$$

- **Intérêt :**

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

- **Inconvénient :**

Cet transformé nécessite deux ondelettes. De plus, elles peuvent introduire des régularités très différentes [30].

II.3.3. Extension à deux dimensions de l'analyse multi résolutions

Il existe plusieurs façons pour étendre la transformation en ondelettes pour un signal 2D, tel que l'image. Le procédé classique consiste à obtenir la fonction d'échelle 2D $\phi(x, y)$ et la fonction d'ondelette $\psi(x, y)$ par le produit tensoriel des fonctions d'ondelettes 1D $\psi(x)$, $\psi(y)$ et les fonctions d'échelle $\phi(x)$, $\phi(y)$.

La fonction d'ondelette mère 2D correspondante permet d'obtenir les détails dans trois directions, horizontale, verticale et diagonale [31] :

$$\psi(x, y)^V = \phi(x) \psi(y) \quad (\text{II.14})$$

$$\psi(x, y)^H = \psi(x) \phi(y) \quad (\text{II.15})$$

$$\psi(x, y)^D = \psi(x) \psi(y) \quad (\text{II.16})$$

On obtient alors, pour une résolution donnée " m ", quatre images distinctes (figure II.5) : l'approximation de l'image original à cette résolution (A_m ou LL_m), et les images de détails dans les trois directions horizontale (DA_m ou HL_m), diagonale (DD_m ou HH_m), verticale (AD_m ou LH_m). En réitérant cette opération pour des résolutions successives sur l'image de l'approximation, on obtient une structure pyramidale en sous bandes.

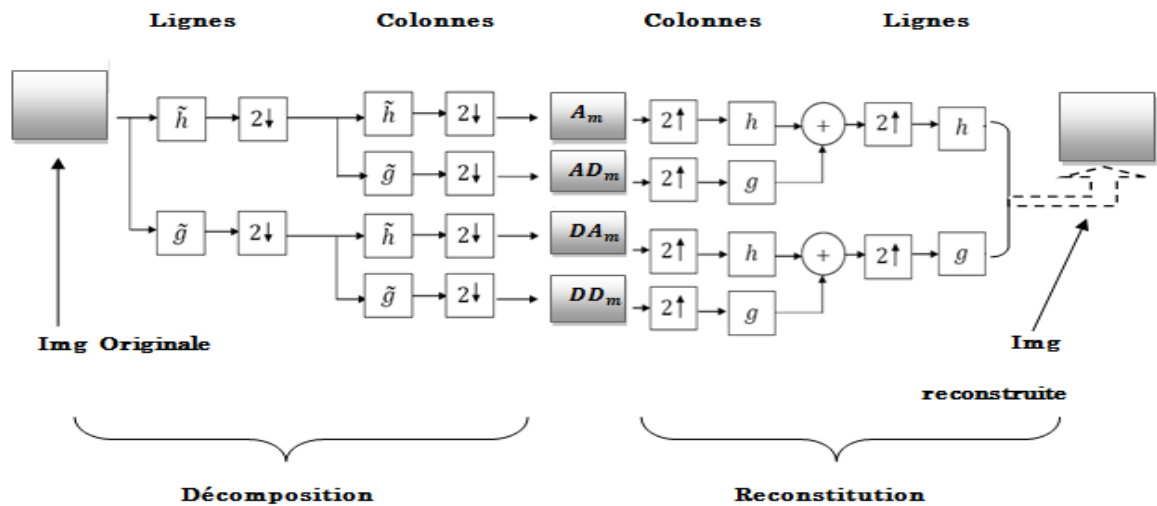
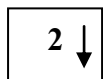
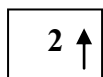


Figure II.5 : Analyse multi-résolution en 2D (cas d'ondelettes bi-orthogonales) [32].

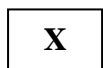
Avec :



Décimation d'une ligne ou d'une colonne sur deux .



Interpolation des lignes ou colonnes , Mettre une ligne ou une colonne de zéros entre deux lignes ou colonnes .



Convolution (ligne ou colonne) avec le filtre (h , g) .

Un exemple de la décomposition à un niveau de résolution $m = 2$ d'une image « Empreinte palmaire » est présenté en (figure II.6) [33] .

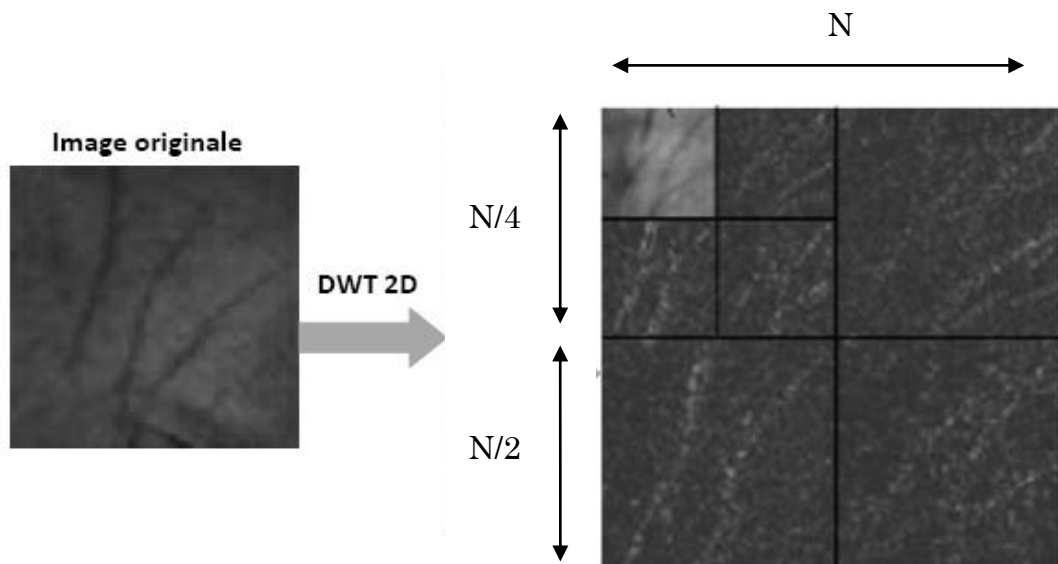


Figure II.6 : TOD 2-D d'une image d' empreinte palmaire.

II.4. Conclusion

Dans ce chapitre nous avons rappelé les transformées directionnelles et on constate que un nombre très restreint de ces transformée directionnelles a été appliqué à la compression. En effet, la plupart de ces transformées ont le désavantage d'être redondantes.

Par contre la transformée en ondelettes peut être assimilée à une segmentation fréquentielle de l'information contenue dans l'image numérique à la manière d'un filtrage horizontale et verticale présentant une structure pyramidale.

Les principes de base de l'analyse multi-résolution et des ondelettes ont été posés par Meyer (Meyer 1986,1988) pour les aspects mathématiques, puis Stéphane Mallat (Mallat 1989) pour les aspects signaux et images.

Nous avons examiné dans ce chapitre comment les principes de l'analyse multi-résolution multidimensionnels présentés précédemment peuvent être étendus aux images numériques

Notre travail, est consacré au codeurs SPECK et notamment à son optimisation notée SPECK rapide (voir chapitre III).

Chapitre III

Codeur SPECK original et optimisé

III.1. Introduction

Le but de ce chapitre est de présenter l'algorithme SPECK et ensuite notre algorithme SPECK modifié afin de montrer l'amélioration apportée .

L' utilisation de la transformée en ondelettes dans l'étape de transformation offre une analyse multi-résolutions de l'image . cette propriété est très intéressante pour la compression progressive des images numériques .

Après un bref rappel sur la méthode originale de quantification développée par Islam et Pearlman (SPECK) , nous proposons une optimisation de l'algorithme SPECK nommée SPECK modifié .

III.2. Principe de l'algorithme SPECK

Les coefficients sont initialement séparés en deux ensembles, l'un noté **S** contenant les coefficients de basses fréquences et l'autre, noté **I** contenant le reste des coefficients (cf. figure III.1). De la même manière que dans SPIHT, deux listes sont tenues à jour, l'ensemble LSP pour représenter les coefficients significatifs et l'ensemble de coefficients non significatifs LIS . La liste d'ensembles non significatifs LIS contient des blocs de coefficients de taille variable ' **y** ' compris les coefficients isolés vus comme des blocs de 1 x 1 (stockés dans la liste LIP dans le cas de SPIHT) [34].

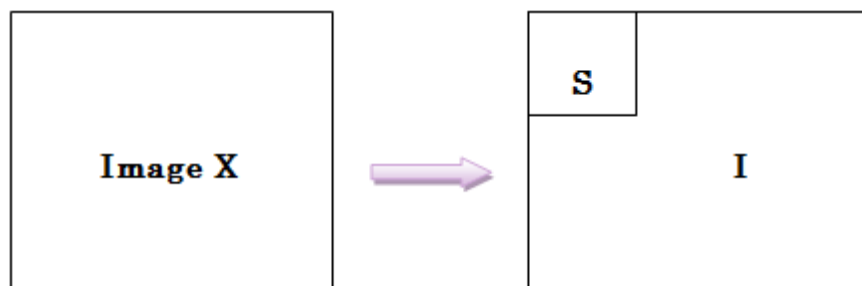


Figure III.1 : Partitionnement de l'image **X** en deux sousensembles **S** et le reste **I**[34].

Lorsque tous les coefficients d'ondelettes ont des valeurs inférieures à un seuil donné, on obtient un bloc de zéro (bloc insignifiant) . on calcule le seuil T_n comme suit : $T_n = 2^n$, n étant la partie entière de logarithme népérien du maximum de toutes les valeurs des coefficients d'ondelettes $C_{i,j}$ de l'image de taille $N \times M$.

III.3. Etapes de l'algorithme SPECK

L'algorithme **SPECK** peut être résumé comme suit :

III.3.1. Initialisation

- Partitionner l'image \mathbf{X} en deux ensembles : " \mathbf{S} " qui prend les coefficients de la sous-bande la plus basse , et " \mathbf{I} " qui prend le reste de l'image tel que : $\mathbf{I} = \mathbf{X} - \mathbf{S}$ (cf. figure III .1).
- Envoyer $n_{max} = \lfloor \log_2 (\max_{(i,j) \in X} \{ | C_{i,j} | \}) \rfloor$ (III.1)
- Ajouter \mathbf{S} à LIS et mettre $LSP = \varnothing$.

III.3.2. Etape de sortie de bits :

❖ Dans l'ordre croissant de taille des listes \mathbf{S} (les ensembles les plus petits d'abord) :

Pour chaque ensemble $\mathbf{S} \in LIS$ faire :

- Si l'ensemble est significatif et non réduit à un seul coefficient alors, Retirer l'ensemble de la liste, le partitionner récursivement en quatre sous-blocs, sur lequel ce test est effectué à nouveau.
- Si le bloc est réduit à un seul coefficient significatif, alors, Ajouter le coefficient à LSP.
- Sinon, l'ensemble est laissé dans la LIS.

❖ **Test de l'ensemble I :**

- Si \mathbf{I} est significatif, alors , Le bloc \mathbf{I} est séparé en trois blocs correspondants au sous-bandes de plus basse-fréquences et un ensemble \mathbf{I} contenant le reste des coefficients (cf. figure III.2). Les trois nouveaux blocs sont traités comme précédemment.
- Répéter le processus de séparation de l'ensemble \mathbf{I} jusqu'à ce qu'il soit insignifiant.

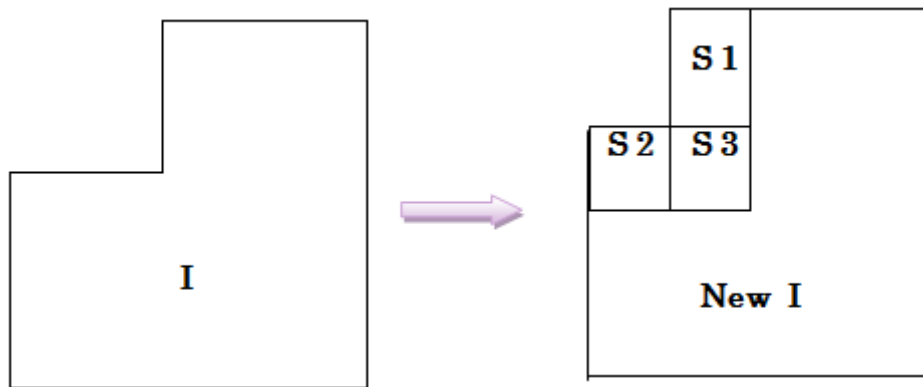


Figure III.2 : Principe de partitionnement des ensembles I.

III.3.3. Phase de raffinement

Pour chaque $(i,j) \in \text{LSP}$, à l'exception de ceux testés par la passe précédente, envoyer le $n^{\text{ième}}$ bit de poids le plus fort de $|C_{i,j}|$.

L'algorithme correspondant est le suivant :

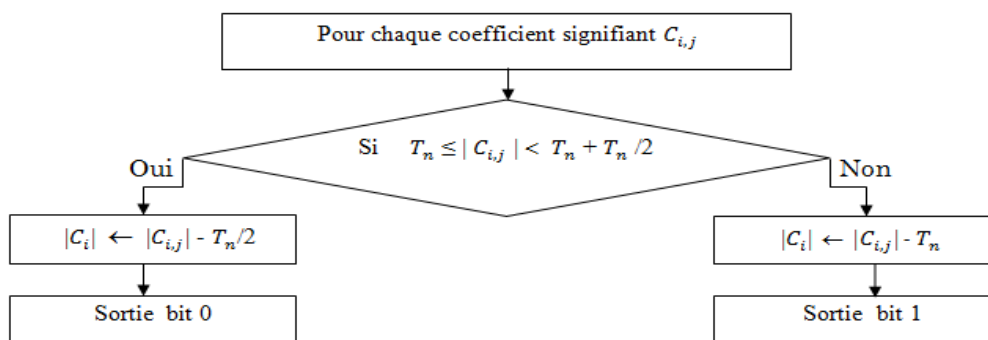


Figure III.3 : Algorithme de l'étape de raffinement des coefficients significants .

III.3.4. Étape de la quantification

Décrémenter n par 1, et aller à l'étape 2.

L'algorithme SPECK, donne généralement des résultats similaires à ceux obtenus par SPIHT [35].

III.4. Exemple d'application du SPECK

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

Figure III.4 : Exemple de matrice de 8×8 coefficients d'ondelettes.

Le seuil T_n est calculé comme suite :

$$N (\text{partie entière de } \log_2 (63)) = 5 \quad ; \quad T_5 = 2^5 = 32$$

Etape	Coefficient ou ensemble testés	Bit de sortie	Action	Listes de controle
N=5 S= $S^1(0,0)$ I=reste				LIS= $((0,0))$ LSP=
	$S^1(0,0)$	1	Partitionnement en quatre Ajout à LIS	LIS= $((0,0)0,(0,1)0,(1,0)0,(1,1)0)$ LSP=
	(0,0)	1+	(0,0) à LSP	LIS= $((0,1)0,(1,0)0,(1,1)0)$ LSP= $((0,0))$
	(0,1)	1-	(0,1) à LSP	LIS= $((0,0)0,(1,1)0)$ LSP= $((0,0),(0,1))$
	(1,0)	0	NON	
	(1,1)	0	NON	

Test I	S(I)	1	Partitionnement en 3S new I	
	$S^1(0,2)$	1	Partitionnement en quatre. Ajout à LIS(0)	LIS= $((1,0)0,(1,1)0,(0,2)0,(0,3)0,(1,2)0,(1,3)0)$
	(0,2)	1+	(0,2) à LSP	LIS= $((1,0)0,(1,1)0,(0,3)0,(1,2)0,(1,3)0)$ LSP= $((0,0)(0,1)(0,2))$
	(0,3)	0	NON	
	(1,2)	0	NON	
	(1,3)	0	NON	
	$S^1(2,0)$	0	Ajout à LIS(1)	LIS= $((1,0)0,(1,1)0,(0,3)0,(1,2)0,(1,3)0,(2,0)1)$
	$S^1(2,2)$	0	Ajout à LIS(1)	LIS= $((1,0)0,(1,1)0,(0,3)0,(1,2)0,(1,3)0,(2,0)1,(2,2)1)$
Test I	S(1)	1	Partitionnement en 3S	
	$S^2(0,4)$	0	Ajout à LIS(1)	LIS= $((1,0)0,(1,1)0,(0,3)0,(1,2)0,(1,3)0,(2,0)1,(2,2)1)(0,4)2)$
	$S^2(4,0)$	1	Partitionnement en quatre. Ajout à LIS(1)	LIS= $((1,0)0,(1,1)0,(0,3)0,(1,2)0,(1,3)0,(2,0)1,(2,2)1,(4,0)1,(4,2)1,(6,0)1,(6,2)1,(0,4)2)$
	$S^1(4,0)$	0	NON	LIS= $((1,0)0,(1,1)0,(0,3)0,(1,2)0,(1,3)0,(4,2)0,(4,3)0,(5,2)0,(5,3)0,(2,0)1,(2,2)1,(4,0)1,(4,2)1,(6,0)1,(6,2)1,(0,4)2)$
	$S^2(4,2)$	1	Partitionnement en quatre. Ajout à LIS(0)	
	(4,2)	0	NON	LIS= $((1,0)0,(1,1)0,(0,3)0,(1,2)0,(1,3)0,(4,2)0,(4,3)0,(5,2)0,(5,3)0,(2,0)1,(2,2)1,(4,0)1,(4,2)1,(6,0)1,(6,2)1,(0,4)2)$ LSP= $((0,0),(0,1),(0,2)(4,3))$
	(4,3)	1+	Déplacer (4,3) à LSP	
	(5,2)	0	NON	
	(5,3)	0	NON	
	$S^1(6,0)$	0	NON	
	$S^1(6,2)$	0	NON	
Fin Sortie n=5	$S^2(4,4)$	0	Ajout à LIS(2)	LIS= $((1,0)0,(1,1)0,(0,3)0,(1,2)0,(1,3)0,(4,2)0,(4,3)0,(5,2)0,(5,3)0,(2,0)1,(2,2)1,(4,0)1,(4,2)1,(6,0)1,(6,2)1,(0,4)2,(4,4)2)$ LSP= $((0,0),(0,1),(0,2)(4,3))$

Tableau III.1 : Exemple de codage d'image par l'algorithme SPECK.

III.5. Algorithme SPECK modifié

Le mêmes étapes que le SPECK original sauf que l'étape de sortie de bits, cette dernière est représentée comme suit :

- ❖ Dans l'ordre croissant de taille des listes S (les ensembles les plus petits d'abord) :

Pour chaque ensemble $S \in LIS$ faire :

- Si l'ensemble est significatif et non réduit à un seul coefficient alors, Retirer l'ensemble de la liste, le partitionner récursivement en quatre sous-blocs, sur lequel ce test est effectué à nouveau.
- Si le bloc est réduit à un bloc de quatre coefficients . Si la significance des quatre coefficients successifs $C_i, C_{i+1}, C_{i+2}, C_{i+3}$ selon le tableau suivant alors (S le coefficient est significatif et I le coefficient est insignifiant) .

C_i	C_{i+1}	C_{i+2}	C_{i+3}	Bits de sortie		
S	I	I	I	00	$C_i \in LSP$	Coder le signe de C_i
I	S	I	I	01	$C_{i+1} \in LSP$	Coder le signe de C_{i+1}
I	I	S	I	10	$C_{i+2} \in LSP$	Coder le signe de C_{i+2}
I	I	I	S	11	$C_{i+3} \in LSP$	Coder le signe de C_{i+3}

Tableau III.2 : Bits de sortie de test de significance selon la position du coefficient.

- Si le bloc est réduit à un seul coefficient significatif, alors, coder son signe et l'ajouter à LSP.
- Sinon, l'ensemble est laissé dans la LIS.

❖ Test de l'ensemble I :

- Si I est significatif, alors , Le bloc I est séparé en trois blocs correspondants au sous-bandes de plus basse-fréquences et un ensemble I contenant le reste des coefficients (même figure III.2). Les trois nouveaux blocs sont traités comme précédemment.
- Répéter le processus de séparation de l'ensemble I jusqu'à ce qu'il soit insignifiant.

III.6. Exemple d'application du SPECK modifié

Le codage des blocs et des coefficients par l'algorithme de SPECK modifié de la matrice de la figure III.4 se fait avec une seule itération. le tableau III.3 résume l'opération de sortie des bits.

Etape	Coefficient ou ensemble testés	Bit de sortie	Action	Listes de controle
N=5 $S=S^1(0,0)$ I=reste				LIS=((0,0)) LSP= ϕ
	$S^1(0,0)$	1	Partitionnement en quatre Ajout à LIS	LIS=((0,0)0,(0,1)0,(1,0)0,(1,1)0) LSP= ϕ
	(0,0)	1+	(0,0) à LSP	LIS=((0,1)0,(1,0)0,(1,1)0) LSP=((0,0))
	(0,1)	1-	(0,1) à LSP	LIS=((0,0)0,(1,1)0) LSP=((0,0),(0,1))
	(1,0)	0	NON	
	(1,1)	0	NON	

Test I	S(I)	1	Partitionnement en 3S new I	
	$S^1(0,2)$	+00	(0,2) à LSP	LIS=((1,0)0,(1,1)0) LSP=((0,0)(0,1)(0,2))
	$S^1(2,0)$	0	Ajout à LIS(1)	LIS=((1,0)0,(1,1)0,(2,0)1)
	$S^1(2,2)$	0	Ajout à LIS(1)	LIS=((1,0)0,(1,1)0,(2,0)1,(2,2)1)
Test I	S(1)	1	Partitionnement en 3S	
	$S^2(0,4)$	0	Ajout à LIS(1)	LIS=((1,0)0,(1,1)0,(2,0)1,(2,2)1, (0,4)2)
	$S^2(4,0)$	1	Partitionnement en quatre. Ajout a LIS(1)	LIS=((1,0)0,(1,1)0,(2,0)1,(2,2)1, (4,0)1, (4,2)1,(6,0)1,(6,2)1,(0,4)2)
	$S^1(4,0)$	0	NON	LIS=((1,0)0,(1,1)0,,(2,0)1,(2,2)1, (4,0)1, (4,2)1,(6,0)1,(6,2)1,(0,4)2)
	$S^2(4,2)$	+01		
			Déplacer (4,3) à LSP	LIS=((1,0)0,(1,1)0,(2,0)1,(2,2)1, (4,0)1, (4,2)1,(6,0)1,(6,2)1,(0,4)2) LSP=((0,0),(0,1),(0,2),(4,3))
	$S^1(6,0)$	0	NON	
	$S^1(6,2)$	0	NON	
Fin Sortie n=5	$S^2(4,4)$	0	Ajout à LIS(2)	LIS=((1,0)0,(1,1)0,(2,0)1,(2,2)1, (4,0)1, (4,2)1,(6,0)1,(6,2)1,(0,4)2,(4,4)2) LSP=((0,0),(0,1),(0,2)(4,3))

Tableau III.3 : sortie de bits de codage de la matrice de la figure III.4 par SPECK modifié.

III.7. Comparaison entre les deux codeurs SPECK et SPECK modifié

En comparant les sorties de bits de codage de la matrice des coefficients d'ondelettes (figure III.4) par les deux codeurs SPECK et SPECK modifié, on obtient les valeurs illustrées dans le tableau III.4.

	Seuil	SPECK	SPECK modifié
Nombre de bits de sortie	32	29	28
	16	53	52
	8	118	114
	4	210	208
	2	292	290

Tableau III.4 : Comparaison entre les deux codeurs SPECK et SPECK modifié.

III.8. Conclusion



Dans ce chapitre, nous avons proposé un algorithme de compression d'image le SPECK modifié , basé sur le même principe que l'algorithme SPECK.

Cet algorithme est basé sur l'utilisation de deux bits au lieu de quatre employés dans l'algorithme SPECK original pour coder les blocs significants de quatre coefficients dont trois insignifiants. A cet effet , nous avons obtenu une quantité d'informations inférieure par rapport à l'algorithme SPECK original.

Chapitre IV

Résultats et Discussions



IV.1. Introduction

Nous présentons dans ce chapitre, une étude détaillée des résultats obtenus par les deux algorithmes (SPECK et SPECK modifié) appliqués sur un ensemble d'images références très utilisées en compression d'images sera défini pour évaluer notre algorithme.

IV.2. Paramètres de validation

IV.2.1. Taux de compression

Sachant que, le but d'une compression est de minimiser la quantité d'informations nécessaires à la représentation d'une image, on définit le rapport de compression R_c par :

$$R_c = \frac{\text{Nbre de bits de l' image originale}}{\text{Nbre de bits de l' image compressée}} \quad [\text{bpp}] \quad (\text{IV.1})$$

Par conséquent, on peut définir la quantité T_c appelée taux de compression par :

$$TC(\%) = 100 - \left(\frac{\text{Taille_image_compressée}}{\text{taille_image_originale}} \right) * 100 \quad (\text{IV.2})$$

IV.2.2. Mesure de la qualité d'image compressée

Le taux de compression n'est pas le seul critère de performance d'un système de compression. Dans le cas d'une compression avec pertes, la qualité de l'image reconstruite doit aussi être prise en compte. Il y a donc un compromis à trouver entre le taux de compression et la qualité. Nous parlons de compromis débit-distorsion. La distorsion est la mesure de l'erreur commise entre l'image originale et l'image reconstruite. Pour cette mesure, nous utiliserons l'Erreur Quadratique Moyenne (EQM) entre l'image originale et l'image compressée de taille $M \times N$ [36] :

$$EQM = \frac{\sum_{m=1}^M \sum_{n=1}^N [I(m,n) - I(m,n)^{\wedge}]^2}{M \times N} \quad (IV.3)$$

Où :

$M \times N$: Taille de l'image à coder

$I(m,n)$: Un pixel de l'image originale

$I(m,n)^{\wedge}$: Un pixel de l'image reconstruite

Le Rapport Signal sur Bruit (Crête Peak Signal to Noise Ratio " PSNR ") en dB est défini par :

$$PSNR = 10 \log_{10} \frac{(255)^2}{EQM} \quad (IV.4)$$

IV.2.3. Temps de calcul (Complexité)

Un troisième critère de performance d'un système de compression est la complexité. La complexité calculatoire peut être mesurée par le temps d'exécution du processus de compression ou en nombre d'opérations par pixel : c'est le nombre moyen d'opérations qui sont nécessaires à la compression de l'image [37].

IV.3. Images de test

Pour notre application, nous avons utilisé les images testées en niveaux de gris :

- Lena codée sur 8 bits, de taille 512x512 pixels (figure IV.1).
- Baboon codée sur 8 bits, de taille 512x512 pixels (figure IV.2).
- Barbara codée sur 8 bits, de taille 512x512 pixels (figure IV.3).
- Peppers codée sur 8 bits, de taille 512x512 pixels (figure IV.4).
- Cameraman codée sur 8 bits, de taille 256x256 pixels (figure IV.5).
- Goldhill codée sur 8 bits, de taille 256x256 pixels (figure IV.6).
- Woman codée sur 8 bits, de taille 256x256 pixels (figure IV.7).
- Hoed codée sur 8 bits, de taille 256x256 pixels (figure IV.8).
- Penny codée sur 8 bits, de taille 128x128 pixels (figure IV.9).



Figure IV.1 : image originale « Lena » 512 × 512 pixels.

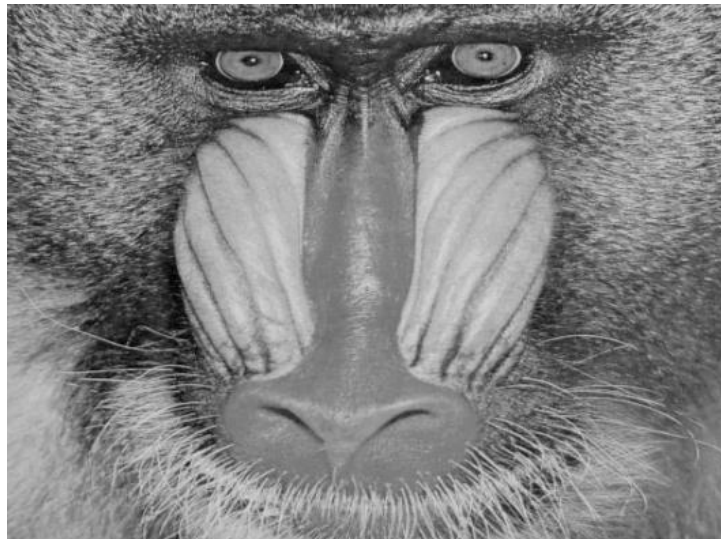


Figure IV.2 : image originale « Baboon » 512 × 512 pixels.



Figure IV.3 : image originale « Barbara » 512 × 512 pixels.



Figure IV.4 : image originale « Peppers » 512 × 512 pixels.



Figure IV.5 : image originale « Cameraman » 256 × 256 pixels.



Figure IV.6 : image originale « Goldhill » 256 × 256 pixels.



Figure IV.7 : image originale « Woman » 256 × 256 pixels.

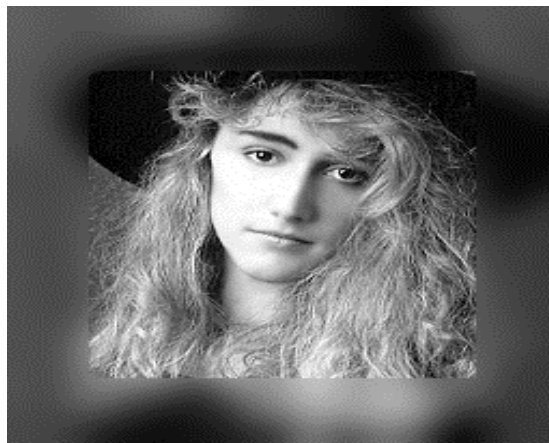


Figure IV.8 : image originale « Hoed » 256x256 pixels.



Figure IV.9 : image originale « Penny » 128x128 pixels.

IV.4. Niveaux de décomposition

La qualité de l'image compressée par ondelettes dépend essentiellement du choix de l'ondelette. En effet, ce choix est basé sur le fait de trouver des bancs de filtres qui compactent le maximum d'énergie, tout en garantissant une bonne qualité de reconstruction de l'image [38].

Un autre critère très important dans la compression est le niveau de décomposition de la transformée par ondelettes. Nous avons choisi quatre résolutions pour pouvoir montrer l'efficacité de notre algorithme.

IV.5. Approche modifiée

Au début, nous appliquons la transformation en ondelettes sur l'image d'entrée. Les coefficients de ondelettes sont indexés par un scan en zigzag (Morton Scan).

Ensuite, nous calculons le seuil de compression " T_n " afin de tester la signifiante des blocs et des coefficients de ondelettes par rapport à ce seuil; le codage se fait par '0' si les coefficients ou les blocs désignés sont insignifiants et par '1' s'ils sont signifiants par rapport au seuil calculé.

Enfin, nous terminons par une étape de raffinement des coefficients significatifs. Après la décrémentation du seuil " T_n ", le programme se déroule jusqu'à l'obtention du taux de compression désiré.

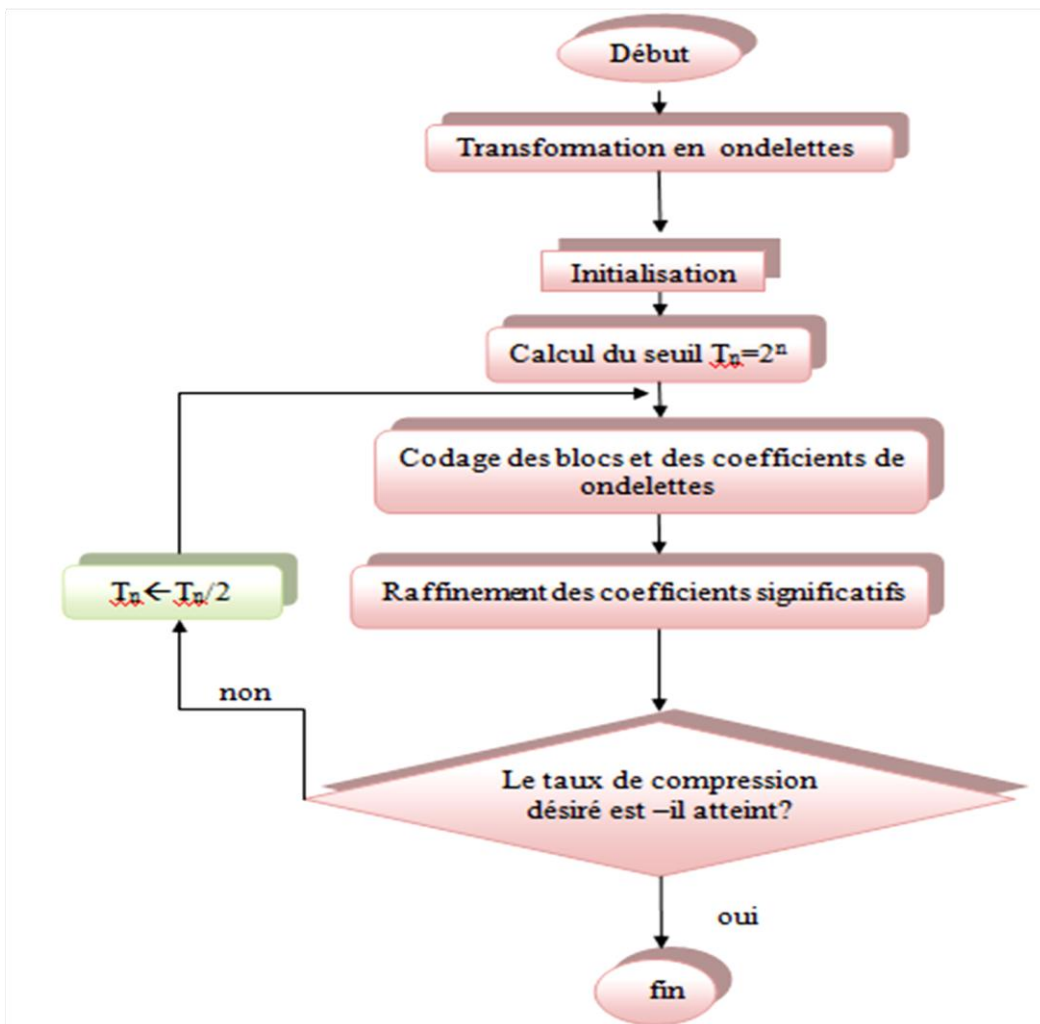


Figure IV.10 : Organigramme de l'algorithme de compression considéré.

IV.6. Etapes détaillées de l'algorithme

IV.6. 1. Initialisation

- 1) Nous calculons le seuil $T_n=2^n$ tel que n : représente la partie entière du logarithme base 2 de la valeur absolue du maximum des coefficients de l'image transformée en ondelettes .

$$n = \text{fix}(\log_2(\max(\text{abs}(\text{image_transformée_ondelettes})))) \quad (\text{IV.5})$$

- 2) Nous initialisons la liste des coefficients significatifs $LSP = \emptyset$.

IV.6. 2. Sortie des bits

Nous appliquons le codage suivant (figure IV.11) sur les coefficients de ondelettes C_{ij} qui ne se trouvent pas dans la liste des coefficients signifiants (LSP).

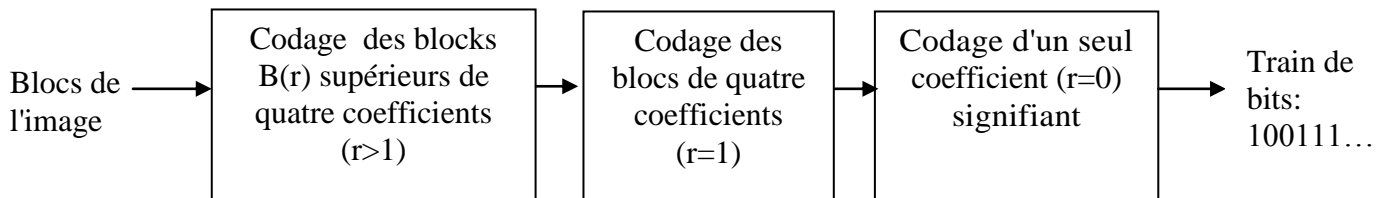


Figure IV.11 : Succession de partition des blocs du plus grand bloc jusqu'au plus petit bloc (taille de bloc = $2^r * 2^r$).

a- Codage d'un bloc $B(r)$ ($r > 1$) :

Si $B(r)$ est insignifiant alors

- sortie du bit '0'
- on passe au bloc suivant $B(r)$

Sinon

- sortie du bit '1'
- partition $B(r)$ en 4 blocs: $r \leftarrow r-1$

Fin

b- Codage d'un bloc B(r) de quatre coefficients (r=1):

Si B(1) est insignifiant **alors**

- sortie du bit '0'
- on passe au bloc suivant B(1)

Sinon

Si B(1) contient un seul coefficient significatif parmi le quatre **alors**

- sortie du bit '1'
- sortie bit '1'

Si la signifiante des quatre coefficients successifs $C_i, C_{i+1}, C_{i+2}, C_{i+3}$ selon le tableau suivant **alors**
(S le coefficient est significatif et I le coefficient est insignifiant)

C_i	C_{i+1}	C_{i+2}	C_{i+3}	Bits de sortie	
S	I	I	I	00	$C_i \in LSP$
I	S	I	I	01	$C_{i+1} \in LSP$
I	I	S	I	10	$C_{i+2} \in LSP$
I	I	I	S	11	$C_{i+3} \in LSP$

Si S est négatif **alors** Sortie bit '1'

Fin

Sinon sortie bit '0'

Sinon

- sortie bit '0'
- partition B(1) en 4 coefficient:

Fin

Fin

c- Codage d'un seul coefficient (r=0) :

Si C_{ij} est insignifiant **alors**

- sortie du bit '0'

Sinon

- sortie du bit '1'
- $C_{ij} \in LSP$

si C_{ij} est négatif **alors** Sortie bit '1'

Sinon sortie bit '0'

Fin

Fin

IV.7. Résultats

Dans ces tableaux, nous commençons par comparer notre méthode SPECK modifié avec le codeur SPECK en variant le PSNR en fonction du taux de compression TC .

Les tableaux suivants illustrent les résultats obtenus avec différentes images de différentes tailles :

IV.7.1. Image Lena (taille 512 x 512)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	209715	38.86	38.96
91	0.72	188743	38.21	38.36
92	0.64	167772	37.61	37.74
93	0.56	146800	36.92	37.01
94	0.48	125829	36.35	36.44
95	0.40	104857	35.70	35.79

Tableau IV.1 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Lena de taille 512x512.



Figure IV.12 : Image Lena reconstruite 512x512 par SPECK modifié avec ondelettes (TC=95% et PSNR=35.79 dB).

IV.7.2. Image Baboon (taille 512 x 512)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	209715	27.36	27.49
91	0.72	188743	26.77	26.95
92	0.64	167772	26.12	26.30
93	0.56	146800	25.22	25.41
94	0.48	125829	24.60	24.68
95	0.40	104857	24.07	24.15

Tableau IV.2 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Baboon de taille 512x512.

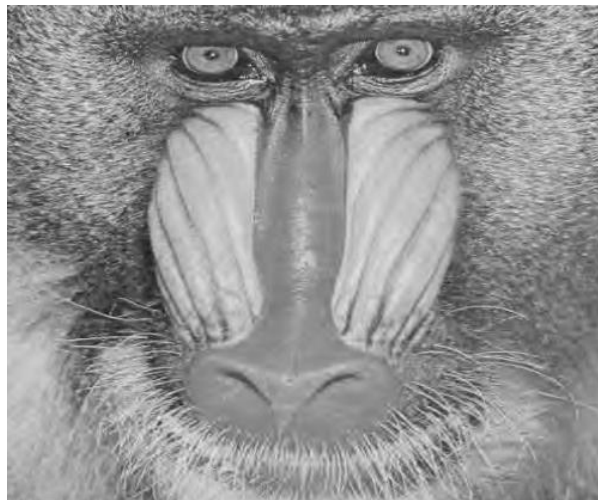


Figure IV.13 : Image Baboon reconstruite 512×512 par SPECK modifié avec ondelettes (TC=95% et PSNR=24.15dB).

IV.7.3. Image Barbara (taille 512 x 512)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	209715	35.07	35.10
91	0.72	188743	34.45	34.50
92	0.64	167772	33.43	33.52
93	0.56	146800	32.20	32.24
94	0.48	125829	31.20	31.24
95	0.40	104857	30.47	30.49

Tableau IV.3 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Barbara de taille 512x512.



Figure IV.14 : Image Barbara reconstruite 512×512 par SPECK modifié avec ondelettes (TC=94% et PSNR=31.24dB).

IV.7.4. Image Peppers (taille 512 x 512)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	209715	36.80	36.94
91	0.72	188743	36.38	36.51
92	0.64	167772	36.01	36.10
93	0.56	146800	35.64	35.71
94	0.48	125829	35.21	35.28
95	0.40	104857	34.46	34.60

Tableau IV.4 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Peppers de taille 512x512.



Figure IV.15 : Image Peppers reconstruite 512×512 par SPECK modifié avec ondelettes (TC=95% et PSNR=34.6 dB).

IV.7.5. Image Cameraman (taille 256 x 256)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	52428	33.71	33.96
91	0.72	47185	32.73	32.99
92	0.64	41943	31.86	32.04
93	0.56	36700	31.11	31.23
94	0.48	31457	30.33	30.45
95	0.40	26214	29.24	29.39

Tableau IV.5 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Cameraman de taille 256x256.



Figure IV.16 : Image Cameraman reconstruite 256x256 par SPECK modifié avec ondelettes (TC=94% et PSNR= 30.45dB).

IV.7.6. Image Goldhill (taille 256 x 256)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	52428	30.82	30.98
91	0.72	47185	30.25	30.39
92	0.64	41943	29.77	29.85
93	0.56	36700	29.36	29.45
94	0.48	31457	28.83	28.91
95	0.40	26214	28.12	28.21

Tableau IV.6 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Goldhillde taille 256x256.



Figure IV.17 : Image Goldhill reconstruite 256×256 par SPECK modifié avec ondelettes (TC=91% et PSNR=30.39dB).

IV.7.7. Image Woman (taille 256 x 256)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	52428	34.97	35.12
91	0.72	47185	34.60	34.72
92	0.64	41943	34.20	34.31
93	0.56	36700	33.52	33.73
94	0.48	31457	32.81	32.93
95	0.40	26214	32.00	32.12

Tableau IV.7 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes et appliqués sur l'image Woman de taille 256x256.



Figure IV.18 : Image Woman reconstruite 256×256 par SPECK modifié avec ondelettes (TC=91% et PSNR=34.72dB).

IV.7.8. Image Hoed (taille 256 x 256)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	52428	32.61	32.88
91	0.72	47185	31.87	32.00
92	0.64	41943	30.99	31.19
93	0.56	36700	30.28	30.40
94	0.48	31457	29.66	29.80
95	0.40	26214	28.78	28.91

Tableau IV.8 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Hoed de taille 256x256.



Figure IV.19 : Image Hoed reconstruite 256×256 par SPECK modifié avec ondelettes (TC=91% et PSNR=32.00 dB).

IV.7.9. Image Penny (taille 128x128)

TC (%)	Débit (bpp)	Nombre de bits	PSNR (dB)	
			SPECK	SPECK modifié avec ondelettes
90	0.8	13107	39.63	39.88
91	0.72	11796	38.79	38.97
92	0.64	10485	37.98	38.12
93	0.56	9175	37.08	37.25
94	0.48	7864	35.73	36.03
95	0.40	6553	34.77	34.86

Tableau IV.9 : Variation du PSNR en fonction du TC pour SPECK, SPECK modifié avec ondelettes appliqués sur l'image Penny de taille 128x128.



Figure IV.20 : Image Penny reconstruite 128×128 par SPECK modifié avec ondelettes (TC=95% et PSNR=34.86dB)

IV.8. Discussions

IV.8.1. Comparaison entre SPECK et SPECK modifié

Le taux de compression augmente dans le cas du SPECK modifié, vu que l'exploitation des insignifiants de coefficients dans les blocs de quatre coefficients ($C_i, C_{i+1}, C_{i+2}, C_{i+3}$), tel que si on trouve trois coefficients insignifiants parmi les quatre, on les code par deux bits (représentant la position du coefficient signifiant dans le bloc) au lieu de quatre bits dans le codeur SPECK original (tableau IV.10).

(A)	(B)																																				
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 5px;">C_i</td> <td style="padding: 5px;">C_{i+1}</td> </tr> <tr> <td style="padding: 5px;">C_{i+2}</td> <td style="padding: 5px;">C_{i+3}</td> </tr> </table>	C_i	C_{i+1}	C_{i+2}	C_{i+3}	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th rowspan="2" style="padding: 5px;">C_i</th> <th rowspan="2" style="padding: 5px;">C_{i+1}</th> <th rowspan="2" style="padding: 5px;">C_{i+2}</th> <th rowspan="2" style="padding: 5px;">C_{i+3}</th> <th colspan="2" style="padding: 5px;">Bits de sortie</th> </tr> <tr> <th style="padding: 5px;">SPECK modifié</th> <th style="padding: 5px;">SPECK original</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">S</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">00</td> <td style="padding: 5px;">1000</td> </tr> <tr> <td style="padding: 5px;">I</td> <td style="padding: 5px;">S</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">01</td> <td style="padding: 5px;">0100</td> </tr> <tr> <td style="padding: 5px;">I</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">S</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">10</td> <td style="padding: 5px;">0010</td> </tr> <tr> <td style="padding: 5px;">I</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">I</td> <td style="padding: 5px;">S</td> <td style="padding: 5px;">11</td> <td style="padding: 5px;">0001</td> </tr> </tbody> </table>	C_i	C_{i+1}	C_{i+2}	C_{i+3}	Bits de sortie		SPECK modifié	SPECK original	S	I	I	I	00	1000	I	S	I	I	01	0100	I	I	S	I	10	0010	I	I	I	S	11	0001
C_i	C_{i+1}																																				
C_{i+2}	C_{i+3}																																				
C_i	C_{i+1}	C_{i+2}	C_{i+3}	Bits de sortie																																	
				SPECK modifié	SPECK original																																
S	I	I	I	00	1000																																
I	S	I	I	01	0100																																
I	I	S	I	10	0010																																
I	I	I	S	11	0001																																

Tableau IV.10 : (A) : Bloc de quatre coefficients ($C_i, C_{i+1}, C_{i+2}, C_{i+3}$)
 (B) : Bits de sortie de test de signifiante selon la position du coefficient.
 Signifiant dans le bloc de quatre coefficients dont trois insignifiants.
I : coefficient insignifiant, **S** : coefficient signifiant.

IV.8.2. Comparaison entre le temps de calcul pour SPECK et SPECK modifié avec ondelettes

TC (%)	Temps de calcul (secondes)			
	SPECK		SPECK modifié	
	Compression	Décompression	Compression	Décompression
90	12.95	2.21	12.10	1.77
91	10.80	2.01	10.49	1.73
92	8.81	1.95	8.49	1.60
93	8.06	1.91	8.46	1.60
94	6.02	1.85	5.72	1.33
95	5.53	1.72	4.18	1.28

Tableau IV.11 : Temps de calcul pour SPECK, SPECK modifié avec ondelettes sur l'image Barbara (512×512).

TC (%)	Temps de calcul (secondes)			
	SPECK		SPECK modifié	
	Compression	Décompression	Compression	Décompression
90	16.18	2.23	14.28	2.03
91	15.09	1.98	13.15	1.90
92	11.82	1.92	10.96	1.75
93	9.28	1.65	8.91	1.51
94	7.51	1.45	7.20	1.35
95	5.63	1.28	5.40	1.19

Tableau IV.12 : Temps de calcul pour SPECK, SPECK modifié avec ondelettes sur l'image Peppers (512×512).

- Le temps de compression est plus supérieur que le temps de décompression car dans le premier nous manipulons un nombre de bits plus grand que dans la décompression, par exemple le nombre de bits codés dans la compression sont $512 \times 512 \times 8$ bits = 2097152 bits par contre dans la décompression si TC= 90% (débit=0.8 bpp) le nombre de bits décodés sont 209715 bits .On peut conclure que le décodage est rapide que le codage.
- De les tables précédente, on remarque que le temps de compression par SPECK modifié avec ondelettes est un peu plus petit que le temps du codeur SPECK original.

IV.9. Conclusion

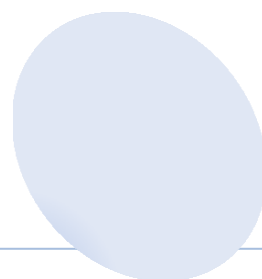


Le SPECK modifié emploie deux bits au lieu de quatre employés dans l'algorithme SPECK original pour coder les blocs significants de quatre coefficients dont trois insignifiants.

L'ajout de la transformée en ondelettes améliore aussi le PSNR parce qu'elle donne des décorrélations supplémentaires dans l'exploitation de la géométrie de l'image.

En se basant sur l'utilisation des pointeurs des coefficients de ondelettes scannés en Morton scan, l'algorithme SPECK modifié devient plus simple du point de vu de la programmation.

CONCLUSION Générale



Conclusion Générale

La compression à bas débit repose sur la construction d'opérateurs d'approximation non linéaires qui permettent d'approcher efficacement le signal à l'aide d'un faible nombre de coefficients. Dans ce cadre, les ondelettes sont connues pour représenter de manière optimale les signaux possédant un nombre fini de discontinuités.

Dans ce contexte, nous avons considéré les algorithmes de compression par ondelettes et par codages imbriqués (EZW, SPIHT et SPECK). Le codeur imbriqué offre la propriété de la transmission progressive de l'image codée .

Dans le cadre de cette mémoire , nous avons proposé un nouvel algorithme pour optimiser le codage SPECK . Notre méthode dite " SPECK modifié " a deux spécificités :

- D'une part, elle est caractérisée par simplicité et rapidité pour minimiser le nombre de bits redondants .
- Et d'autre part, elle est donnée des meilleurs résultats par rapport à l'algorithme du SPECK original .

Dans le SPECK original , chaque coefficient insignifiant dans le bloc de quatre coefficients dont trois insignifiants est codé indépendamment par un bit " 0 " , mais dans le SPECK modifié , nous codifions les trois coefficients insignifiants dans ce bloc particulier par deux bits seulement.

Après avoir observé nos résultats , on peut conclure que le temps de compression par SPECK modifié avec ondelettes est un peu plus petit que le temps du codeur SPECK original parce que dans le premier nous testons les blocs significants qui contiennent trois coefficients insignifiants au même temps , mais dans le deuxième le test se fait coefficient par coefficient dans ces blocs particuliers .

Conclusion Générale

En perspectives, nous proposons d'associer notre algorithme SPECK modifié avec une transformation qui utilise les cervelets ou les contourlets.

L'association de ces transformations avec notre algorithme pourrait réduire en plus le nombre de coefficients à coder.



Bibliographies



- [1]. OUAFI Abdelkrim, "Compression d'images sans pertes JPEG-Ls ", , Thèse de Master en Sciences en Electronique , Université Mohamed Khider – Biskra , 2013.
- [2]. Frédéric Dufaux, " Traitement du Signal et des Images " ,TELECOM ParisTech, 26 janvier 2011.
- [3]. Patrick Gros," Traitement de l'image et de la vidéo" , 2014.
- [4]. Site : <https://www.gralon.net/articles/photo-et-video/photo-et-video/article-les-pixels-definition-et-caracteristiques-2014.htm> .
- [5]. C. Diltsm, " Numérisez vos collections", RCIP Octobre 2001.
Site : http://www.chin.gc.ca/Francais/Contenu_Numerique/Guide_Gestionnaires/ .
- [6]. C. Lepec, N. RIMOUX & A. COLIN Editeurs. "Les Graphiques Sur Ordinateurs", Paris, 1991. Site: <http://www.kaddour.com/> .
- [7]. Cédric Valade. Thèse " Compression d'images complexes avec pertes: application à l'imagerie Radar " P 36 , domain_ether. Télécom ParisTech, Décembre 2006.
- [8]. Franck Davoine. Thèse " Compression d'images par fractales basée sur la triangulation de Delaunay" . Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 1995. Français.
- [9]. Gaudeau, Y. "Contributions en compression d'images médicales 3D et d'images naturelles 2D", Thèse de doctorat de l'Université Henri Poincaré, Nancy 1, Spécialité Automatique, Traitement du Signal et Génie Informatique, France(2006).
- [10]. N. Morceau, " Techniques de compression des signaux ", Edition Masson,1995.
- [11], [12] , [17]. M. Lahdir, "Nouvelle approche de compression d'images basée sur les ondelettes et les fractales : Application aux images Météosat ", thèse de Doctorat en électronique : Option Télédétection, faculté de génie Electrique et Informatique, Université Mouloud Mammeri de Tizi-Ouzou UMMTO, p. 120, 2007.
- [13]. R. Sylvain, "La Compression de Données", Club Photoshop de Nantes. Conférence du 14 octobre 1999.
- [14]. D.A. Huffman, "A method for the construction of minimum-redundancy codes", Proc. of the IRE, vol.40, pp. 1098-1101, September 1952 .
- [15]. D. Taubman, W.M. Marcellin "JPEG2000 image compression fundamentals, standards and practice", Kluwer Academic Publishers, 2002.
- [16]. F. Davoine, Compression d'Images par Fractales Basée sur la Triangulation de Delaunay, Thèse l'INPG, Institut National Polytechnique de Grenoble, Décembre 1995 .

- [18] . *Z. BAARIR, *N. DOGHMANE*, *A. OUAFI*, *N. TERKI *," compression d'images fixes biomédicales par Transformée en ondelettes, quantification Vectorielle et codage entropique " , Courrier du Savoir – N°02, Juin .
- [19]. Vivien Chappelier, " codage progressif d'images par ondelettes orientées " . Thèse de doctorat en traitement du signal . Université de Rennes 1,2005.
- [20]. OUAFI Abdelkarim. "compression d'images avec pertes par codages imbriqués, proposition d'une optimisation de l'algorithmme EZW " . Thèse de doctorat en Electronique .Université Mohamed Khider Biskra,2009.
- [21]. A. Said and W.A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits and Systems for Video Technology, vol. 6, pp. 243–250, Juin 1996.
- [22]. Pragya Tiwari , Mohd Ahmed, S.G.Kerhalkarm , " Transmission of Images Using SPECK " , IJCSNS , VOL.14, No.11, November 2014.
- [23]. J. Mainai beye, "Codage et compression d'images par ondelettes“, thèse de doctorat,Université de Tunis, ELMANAR, E.N.I.T, Juillet 2002 .
- [24]. V. Chappellie, "Codage progressif d'images par ondelettes orientées". Thèse de doctorat, Université de Rennes 1. Déc 2005 .
- [25]. DAHRAOUI N., " Traitement numérique du signal secondaire de l'analyse par spectrométrie de masse des ions secondaires. Quantification des profils en profondeur et mise au point d'une procédure de déconvolution. Cas des structures TiSi2/ Si(B) et TiSi2/Si(As)" , Thèse de Magister, Université Kasdi Merbah de Ouargla, 2014, 87 p.
- [26]. M.HETTIRI , " Etude d'algorithmes de codage imbriqué appliques à la compression de séquences d'image " , Mémoire de magister en électronique . Université Mohamed khider Biskra , 2006.
- [27]. Vivien Chappelier, " codage progressif d'image par ondelettes orientées ". Thèse de doctorat en traitement du signal . Université de Rennes 1, 2005.
- [28]. G. Donovan, J. Geronimo, D. Hardin, P. Massopust, "Construction of orthogonal wavelets using fractal interpolation functions", SIAM J. Math. Anal., vol. 27, N_4, pp. 1158-1192,1996.
- [29]. A.Cohen, I.Daubechies, J.C.Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets", Comm. On Pure and Applied Mathematics, , vol.XLV, pp.485-560,1992.
- [30]. Marc ANTONINI , " COMPRESSION DES IMAGES ET DES VIDEOS NUMERIQUES DIX ANNEES DE RECHERCHES AU CNRS" , Université de Nice-Sophia Antipolis, le 20 octobre 2003.
- [31]. KADRI Oussama . " Compression d'images fixes par Ondelettes géométriques par utilisation des Curvelets et différents types d'interpolation dans la quantification scalaire " , Mémoire de magister en électronique. Université Mohamed khider Biskra. 2014.

- [32]. M. OUKALI Salim , " Compression d'images par bandelettes : Application à des images de télédétection " . Mémoire de magister en électronique. Université Mouloud Mammeri Tizi-Ouzou ,2011.
- [33]. BOUCHEMHA .Amel , " Etude et Application des transformées géométriques à la Compression des images hautes résolutions et à la Biométrie (Authentification Vérification de l'empreinte palmaire " , Thèse de doctorat en électronique , Université Badji Mokhtar – Annaba , p 10 , 2015.
- [34]. A. Islam and W. A. Pearlman, "An embedded and efficient low-complexity hierarchical image coder," Visual Communications and Image Processing '99, Proceedings of SPIE, vol. 3653, pp. 294–305, Jan. 1999.
- [35]. William. A. Pearlman and A. Said. "Set Partition Coding: Part I of Set Partition Coding and Image Wavelet Coding Systems". Foundations and Trends in Signal Processing .Vol. 2, No. 2., p137, 2008.
- [36]. R.Lakkundi , M.V.Latte, D.K.Deshpande "Reduced memory listless speck image " . International Conference on intelligent Signal processing and Robotics , Allahabad, INDIA,February 2013.
- [37]. M. Lahdir, "Nouvelle approche de compression d'images basée sur les ondelettes et les fractales : Application aux images Météosat " , Thèse de Doctorat en électronique : Option Télédétection, faculté de génie Electrique et Informatique, Université Mouloud Mammeri de Tizi-Ouzou UMMTO, 2007.
- [38]. J.D. Villasenor, B.Belzer and J.Liano "Wavelet Filter Evaluation for image Compression", Proc. In IEEE International conference on Image Processing ICIP'97, Vol.1, pp.624-627,1997.



