

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'études

En vue de l'obtention du diplôme d'ingénieur d'état en Electronique

Option : contrôle

Thème :

Compression d'images par le codeur par plans de bits (BPE) du standard CCSDS

Proposé et dirigé par:
Mme. LAHDIR Leila

Présenté par :
M^{elle} HAMLAT Tassadit
M^{elle} MANSEUR Assia

Promotion : 2009/2010



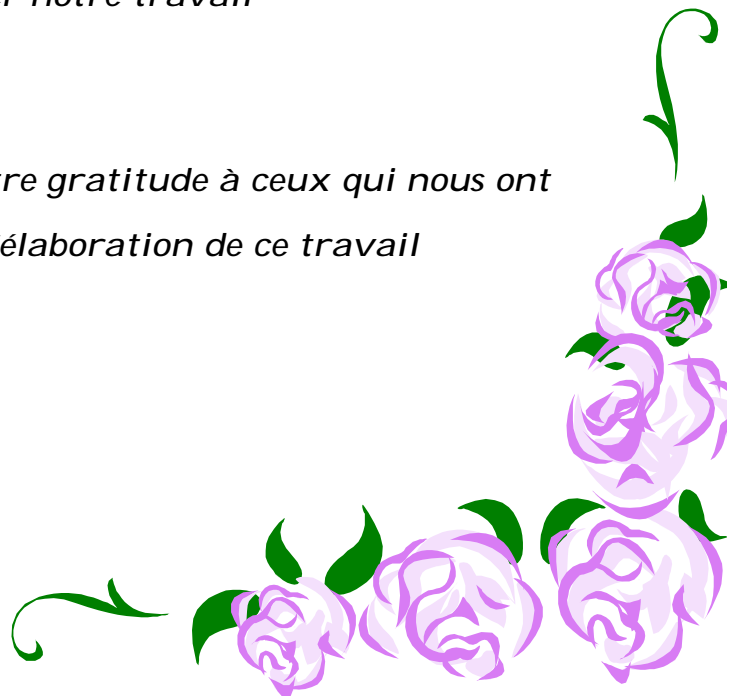
Remerciements

Nous tenons à remercier en premier lieu le dieu qui nous a montré le bon chemin et nous a donné du courage pour mener notre travail avec beaucoup de patience.

Nous exprimons nos respects et remerciements à Mr et Mme. Lahdir pour nous avoir encadrés.

Nous remercions également tous les membres du jury qui nous feront l'honneur de juger notre travail

En fin, nous exprimons toute notre gratitude à ceux qui nous ont aidés de près ou de loin à l'élaboration de ce travail





Je dédie notre modeste travail à :

✚ *A mes très chers et merveilleux parents, que j'ai la chance d'avoir. Qui m'ont soutenu tout au long de ma vie en m'offrant leur tendresse, leur amour, leur sagesse et surtout leur compréhension, je leur dis merci. Que dieu les protège.*

✚ *A mes très chers frères : Azzedine et Krimo.*

✚ *A ma très chère sœur Warda.*

✚ *A ma très chère sœur Wassila, son mari Alilou .*

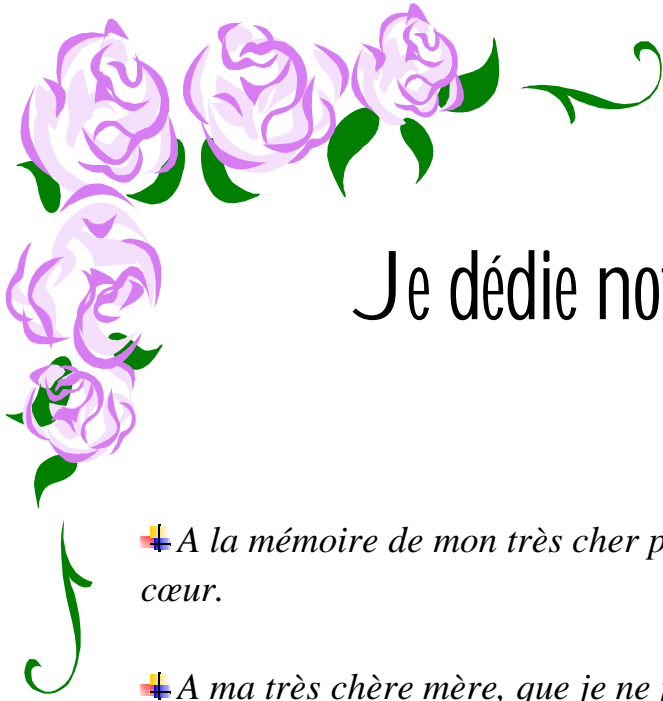
✚ *A mon adorable et très cher neveu Yacine.*

✚ *A tous mes oncles et tantes paternels et maternels.*

✚ *A tous mes cousins(es), tous mes amis(es), sans exception.*

ASSIA





Je dédie notre modeste travail à :

✚ *A la mémoire de mon très cher père qui sera toujours vivant dans mon cœur.*

✚ *A ma très chère mère, que je ne pourrais jamais remercier assez pour tout ce qu'elle a fait pour moi. Que dieu la protège.*

✚ *A mes frères TAHAR et KARIM et belles sœurs HAKIMA et HAYET.*

✚ *A mes chères sœurs HORIA et SALIHA.*

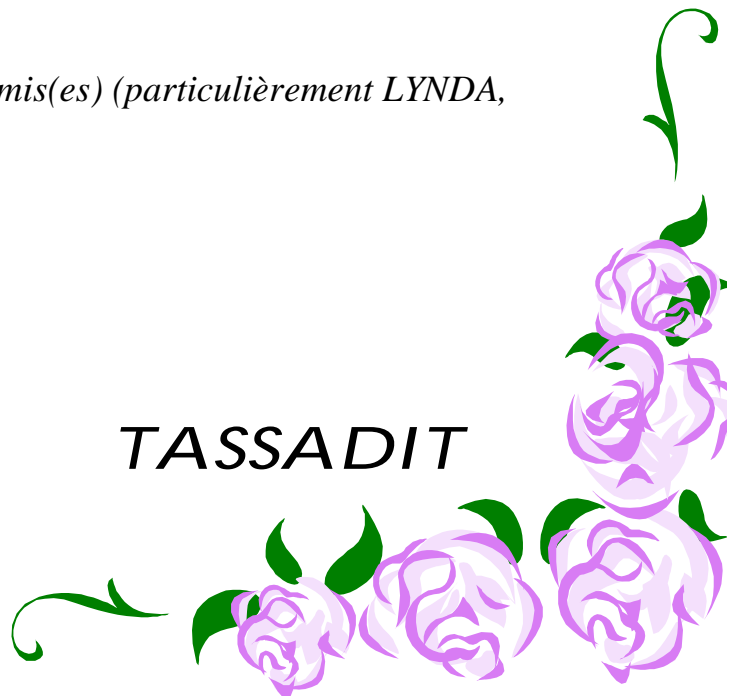
✚ *A ma sœur TAOUS et à la mémoire de son mari ARIZKI.*

✚ *A ma sœur NOUARA et son mari DJAMEL.*

✚ *A mes adorables neveux et nièces.*

✚ *A tous mes cousins(es), tous mes amis(es) (particulièrement LYNDA, DJIDJI et FADILA)*

TASSADIT



SOMMAIRE

Sommaire

Introduction	1
Chapitre I : Aperçu sur la compression d'image	
I. Préambule	3
I.1. Images et propriétés	3
I.1.1 Définition d'une image	3
I.1.2. Définition d'une image numérique	3
I.1.3. Définition de la compression d'image	4
I.1.4. Le but de la compression d'image	4
I.1.5. Principe général de la compression d'image	4
• Dé corrélation	5
• Quantification	5
• Codage	5
• La qualité de reconstitution de l'image :	5
I.1.6. Mesure des performances	5
• Le taux de compression	5
• L'entropie	6
• Mesure de distorsion	6
I.2. Principales méthodes de compression d'image	6
I.2.1. Méthodes réversibles	7
I.2.1.1.Codages statistiques (codages entropiques)	7
I.2.1.1.1. Codage de Shannon -Fano	7

I .2. 1.1.2. Codage de Huffman	9
I .2.1.1.3. Codage arithmétique	10
I .2.1.2. Cordage RLE (Run Length Encoding)	13
I .2.1.3. Méthodes à base de dictionnaire (LZW)	14
I .2.2. Méthodes irréversibles	14
I .2.2.1. Sous échantillonnage	14
I .2.2.2. Codage en sous bandes	15
I .2.2.3. Codage par dictionnaire	15
• La quantification scalaire	16
• La quantification vectorielle	16
I .2.2.4. Codage par fractale	17
I .2.2.5. Codage par transformée	17
I .2.2.6. Codage par DPC (Différentiel Prédictive Coding)	19
I .2.2.7. Méthodes hybrides	19
I .3. Les normes de la compression d'image	20
I .3.1. La norme JPEG	20
I .3.2. La norme JPEG 2000	21
I .3.3. Le standard CCSDS	22
I .4. Discussion	23

Chapitre II: Théorie des ondelettes

II.1.Préambule	24
II.2.Eléments de la théorie des ondelettes	25
II. 2.1.Définition	25

II.2.2.types d'ondelettes	25
• Les ondelettes orthogonales	25
• Les ondelettes biorthogonales	26
II.2.3.Propriété des ondelettes	26
II.3.Les types de la transformée en ondelettes	27
II.3.1.La transformée en ondelettes continue (TOC)	27
II.3.2.La transformée en ondelettes discrète (TOD)	29
II.4. Analyse multi résolution	30
II.4.1.Commentaires sur les propriétés de l'analyse multi résolution	31
II.4.2.Fonction d'échelle Φ et sous espaces V_j	31
II.4.3.Fonction ondelette et l'espace de détails W_j	32
II.4.4.Filtres associés aux ondelettes	32
II.4.5.Fonctions d'échelle et ondelettes à deux dimensions	33
II.4.6.Ondelettes et analyse multi résolution	34
II.4.7. Application de l'algorithme de S Mallat	34
II.4.7.1. Algorithme de décomposition	34
II.4.7.2.Algorithme de reconstruction	37
II.5.Discussion	39
 Chapitre III : Méthode adoptée	
III.1.Préambule	40
III.2. Schéma synoptique de la méthode adoptée.....	40
III.2.1.La transformée en ondelette discrète	41
III.2.2.Quantification à zone morte	41

III.2.3.Codage par plan de bits	41
• Les ensembles de coefficients :	44
• Les mots de transitions :	44
• Algorithme de base.....	46
III.2.4. Le décodage et la transformée en ondelette inverse	53
III.3.Discussion	53

Chapitre IV : Tests et résultats

IV.1 Préambule	55
IV.2. Représentation des données	55
IV.3. Les images tests	55
IV .4.Tests et interprétation des résultats obtenus	56
IV .5.Discussion	64
Conclusion	65
Annexes	66

Bibliographie

INTRODUCTION

L'image est devenue de nos jours le support d'informations le plus répandu et le plus performant vu son universalité, son accessibilité, son riche contenu et ces multiples applications dans tous les domaines des sciences.

L'évolution technologique a eu pour résultat l'introduction de l'électronique numérique dans le domaine de l'imagerie, ceci à un grand pas révolutionnaire pour le domaine de traitement d'image. Dès lors on a pu réaliser toutes sortes de manipulations sur les images, du moment où celles-ci étaient facilement exploitables par les machines informatiques.

Cependant l'énorme volume d'informations à la représentation de ces images pose de très sérieux problèmes quant leur stockage. Les solutions possibles à de tels problèmes sont soit perfectionner encore plus le matériel (support de stockage), afin de permettre une capacité de stockage supérieure, ou alors faire recours aux techniques de compression d'images.

Cette dernière était la plus évidente et la moins coûteuse. Les efforts menés dans ce sens par les organismes internationaux. L'**ISO** (**I**nternational **S**tandard **O**rganisation) et le **CCITT** (**C**omité **C**onsultatif **I**nternationale du **T**élégraphe et du **T**éléphone) et le **CCSDS** (**C**omité **C**onsultatif pour les **S**ystèmes des **D**onnées **S**pacials), ont conduit à l'apparition de plusieurs normes de compression d'images.

Deux catégories de techniques de compression sont proposées dans la littérature , à savoir les techniques réversibles et les techniques irréversibles.

Les techniques réversibles présentent l'avantage d'une reconstitution exacte du signal original et l'inconvénient d'un taux de compression insuffisant. Les techniques irréversibles autorisent une certaine perte d'information en exploitant les propriétés de l'œil. Ces méthodes permettent d'atteindre des taux de compression élevés tout en conservant la qualité de l'image.

Le travail effectué dans ce mémoire consiste à appliquer la transformée en ondelettes discrète sur l'image, suivie d'une quantification scalaire ; les coefficients issus de cette étape seront codés par le codeur BPE.

- Ø Dans le premier chapitre, on donne un aperçu général sur les différentes méthodes et quelques normes de compression d'images numériques.
- Ø Le second chapitre, aura pour contenu l'ensemble des notions mathématiques liées à la théorie des ondelettes et à l'analyse multi résolution.
- Ø Le troisième chapitre, sera consacré à la représentation de la méthode adoptée qui sera décrite d'une façon exhaustive.
- Ø Enfin dans le dernier chapitre, on va discuter les résultats issus des différents tests effectués en appliquant la méthode choisi.
- Ø On termine par une conclusion générale, tout en discutant les perspectives ouvertes par ce travail.

CHAPITRE I

*Aperçus sur la
compression d'images*

I.1. Préambule:

Plus la taille des informations ne cessent d'augmenter avec le temps ; la compression de données s'impose comme une solution très économique et simple d'utilisation. Aujourd'hui ; encore bien que la taille des unités de stockage ait connu un véritable débordement ; en simplifiant l'archivage par réduction du nombre de supports nécessaires.

La compression est basé sur une constatation simple .on peut distinguer la compression conservatrices (réversibles ou sans pertes), restituant l'intégralité du document (image originale) après décompression ; et la compression non conservatrices (irréversibles ou avec pertes), qui perdent une partie des informations de façon peu visible comme le standard JPEG et JPEG2000.La phase de décompression appelée aussi décodage où la restitution est nécessaire pour représenter l'image décompressée.

I.2. Image et propriétés :**I.2.1 Définition d'une image :**

D'une manière générale, on définit une image comme étant un support d'information. Elle présente une scène qui a été capté soit par un appareil photographique, soit par un satellite,...

Une image peut avoir d'autre définition. En traitement du signal, on défini une image comme étant un signal bidimensionnel. Mathématiquement parlant, une image est une application d'un sous ensemble $M \times N$ de $R \times R$ vers l'ensemble des réels R , qui à chaque couple de réels (x, y) associer le réel $f(x, y)$:

$$\begin{aligned} f(M, N) &\longrightarrow R \\ (x, y) &\longrightarrow f(x, y) \end{aligned}$$

I.2.2. Définition d'une image numérique :

Une image numérique, encore appelée image BITMAP est une image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleur prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de scène à représenter.

Il est très important de mentionner qu'une image numérique est obtenue par une conversion de celle-ci de son état analogique (signal électrique) vers un état numérique (binaire) pour qu'on puisse le traiter avec un système informatique.

I.2.3. Définition de la compression d'image :

La compression d'image est la réduction du nombre de bits par pixels à stocker ou à transmettre.

Cette compression a pour utilité de réduire la redondance des données d'une image afin de pouvoir l'emmagasiner sans occuper beaucoup d'espace ou la transmettre rapidement

I.2.4. Le but de la compression d'image :

Mettre les informations sous un format tel quelles occupent beaucoup moins de volume ; et de rendre la taille de l'information compatible au débit d'un canal de transmission, ou à la capacité de stockage d'un support afin de diminuer son temps de transmission.

I.2.5. Principe général de la compression d'image :

Le schéma fonctionnel de la compression est représenté dans la figure (I.1) :

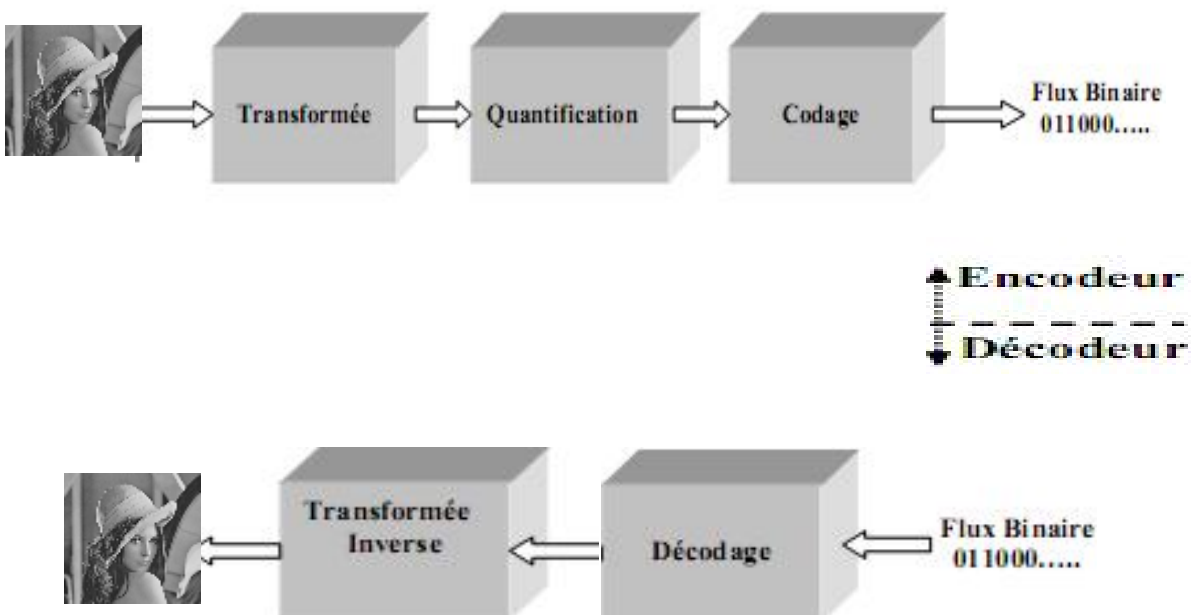


Figure (I.1) Schéma fonctionnel de la compression /décompression

- **Dé corrélation :**

Elle consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés. 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. Donc on essaie de réduire le volume d'information en effectuant une opération de dé corrélation des pixels. C'est une opération réversible.

- **Quantification :**

La quantification des coefficients obtenus a pour but de réduire le nombre de bits nécessaire pour leur représentation. Elle approxime chaque valeur d'un signal par un multiple entier d'une quantité « Q » appelé « quantum élémentaire » ou « pas de quantification ».

La quantification peut être scalaire ou vectorielle.

- **Le codage :**

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

§ Unicité : deux messages différents ne doivent pas être codés de la même façon.

§ Déchiffrabilité : deux mots de codes successifs doivent être distingués sans ambiguïté.

I.2.6. Mesure des performances :

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

- **Le taux de compression :**

Le taux de compression est le rapport entre la quantité de bits utilisé par l'image originale et celle utilisé par l'image comprimée

Rapport de compression :

R_c = nombre de bits de l'image originale / nombre de bits de l'image décompressée

$$\text{Taux de compression : } T_c = \left(1 - \frac{1}{R_c}\right) * 100 \quad \text{(I.1)}$$

- **L'entropie :**

C'est la mesure de quantité d'information contenue dans une image. Une image dont tous les pixels ont la même valeur, contient très peu d'information donc son entropie est très faible, par contre, une image dont tous les pixels ont une valeur aléatoire, contient beaucoup d'informations, son entropie est élevée.

$$H(p) = - \sum_{k=0}^m \{P(k) \log P(k)\} \quad (\text{I.2})$$

k : les niveaux de gris que peut revêtir un point particulier d'une image.

m : nombre totale de p .

$P(k)$: probabilité d'apparition de niveaux de gris.

- **Mesure de distorsion :**

La distorsion est l'erreur introduite lors de compression d'une image.

$$MSE = \frac{1}{m \times n} \sum_{i=0}^m \sum_{j=0}^n [I(i, j) - I'(i, j)]^2 \quad (\text{I.3})$$

$I(i, j)$: pixel de l'image originale.

$I'(i, j)$: pixel de l'image reconstruite.

m : nombre de ligne de la matrice image.

n : nombre de colonne de la matrice image.

- **La qualité de reconstitution de l'image :**

Le rapport signal sur bruit crête est noté.

$PSNR$: est la mesure de la performance d'un codeur en termes de qualité visuelle de l'image reconstruite.

$$PSNR = 10 \log_{10} \left[\frac{(255)^2}{MSE} \right] (dB) \quad (\text{I.4})$$

255= dynamique de l'image.

Lorsque la reconstitution est parfaite :

$MSE \rightarrow 0$.

$PSNR \rightarrow \text{l'infini}$.

Ces mesures sont très utilisées car elles sont très simples à calculer, cependant elles ne rendent pas compte de la qualité visuelle engendrée par la compression.

- **Temps d'exécution (tc)** : temps du processus de compression /décompression.

I.3. Principales méthodes de compression d'image :

Méthodes de compression peuvent se regrouper, en deux classes :

- Les méthodes sans perte d'information (sans distorsion ou réversibles).
- Les méthodes avec perte d'information (avec distorsion ou irréversibles).

Les expérimentations menées montrent que généralement les méthodes qui atteignent des taux de compression très élevés sont les méthodes avec distorsion (irréversibles) par contre, les méthodes sans distorsion (réversible) engendrent des taux de compression très faibles et ne sont utilisées que dans des applications sensibles telles que les images médicales et des images satellitaires.

I.3.1. Méthodes réversibles :

Elles permettent de retrouver exactement les pixels de l'image originale.

I.3.1.1. Codages statistiques (codages entropiques) :

Le but de ces codages est de s'approcher le plus possible de l'entropie de l'image, en affectant les codes les plus courts possibles aux pixels de probabilités élevées et vice versa. Parmi les codages statistiques, on a :

I .3.1.1.1. Codage de Shannon -Fano :

C. Shannon du laboratoire BELLS et R. M. Fano du MIT ont développé à peu près en même temps une méthode de codage, qui après avoir classé par ordre de probabilités décroissantes les symboles, on les partage en deux parties dont la somme des probabilités est comparable. La partie inférieure prend la valeur 1, la partie supérieure prend la valeur 0.

On divise ensuite chaque partie de nouveau en deux et on rajoute de nouveau un 0 ou un 1, et ainsi de suite jusqu'à avoir des parties avec un seul symbole. Pour un symbole, le code résultant est constitué de succession de 0 et de 1 qui ont été affectés.

Exemple :

Voici l'image de la figure :

```

4 4 4 4 4 4 4 0
4 5 5 5 5 5 4 0
4 5 6 6 6 5 4 0
4 5 6 7 6 5 4 0
4 5 6 6 6 5 4 0
4 5 5 5 5 5 4 0
4 4 4 4 4 4 4 0
4 4 4 4 4 4 4 0
    
```

Figure (I.2) Exemple d'un bloc d'image de 8*8pixel sur 8 niveau de gris.

Niveau de gris	Nombre d'occurrences	Probabilités respectives
(a) 0	8	0.125
(b) 4	31	0.484
(c) 5	16	0.250
(d) 6	8	0.125
(e) 7	1	0.016

Table (I.1) Calcule de probabilités de chaque niveau de gris.

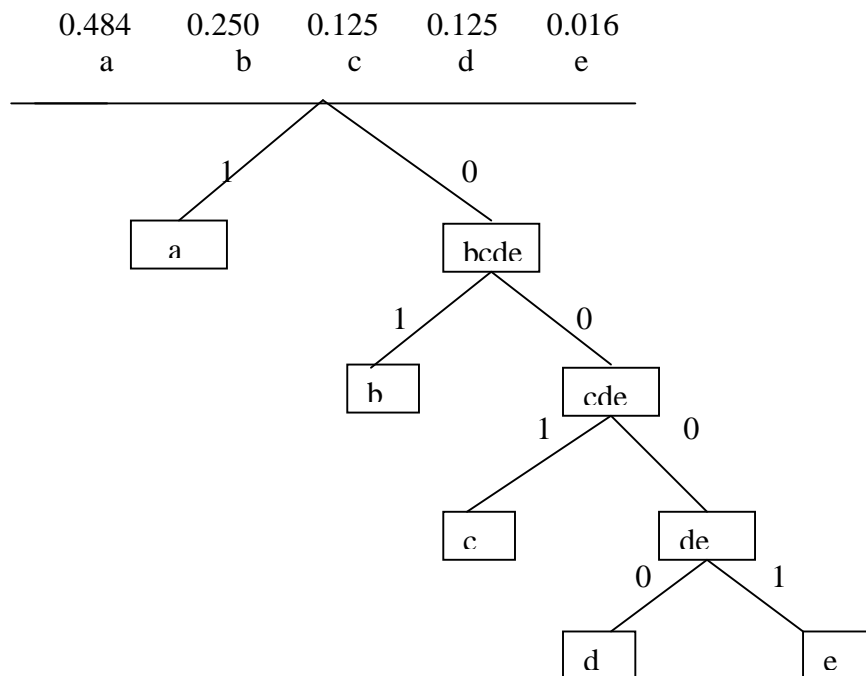


Figure (I.3) Arbre de Shannon Fano.

(a :1, b :01, c :001, d:0000, e:0001)

Bien que le codage de Shannon- Fano soit un grand pas en avant, il a été rapidement remplacé par un système de codage encore plus efficace : le codage de Huffman.

I .3. 1.1.2. Codage de Huffman :

Le codage de Huffman a la particularité de produire des codes de préfixe unique, ce qui permet de les décoder sans aucune ambiguïté. La façon de construire les codes binaires est différente de Shannon- Fano .L’algorithme de Huffman comprend les étapes suivantes :

- On ordonne la suite de probabilités sans prendre en compte la probabilité nulle.
- On combine les deux probabilités les plus faibles, puis on réordonne la nouvelle suite
- On itère le processus jusqu’à l’obtention d’une suite à deux éléments.

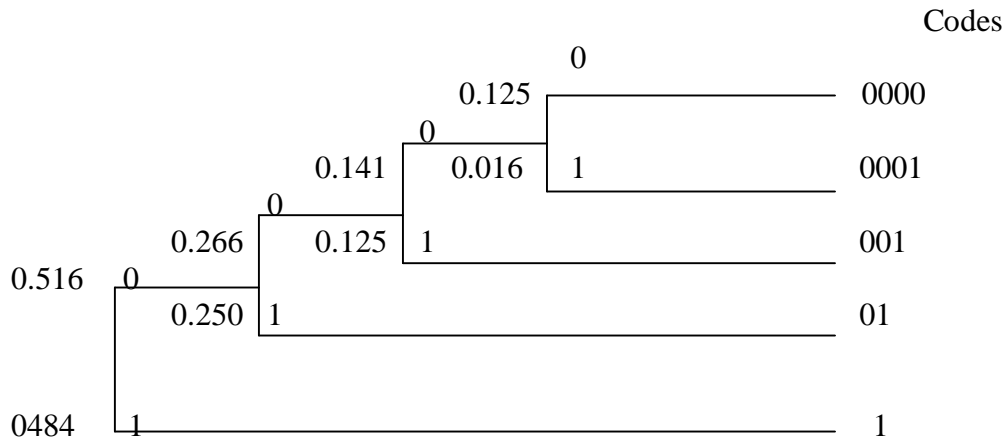


Figure (I.4) Exemple de codage par Huffman.

I.3.1.1.3. Le codage arithmétique :

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.

L'algorithme du codage arithmétique est le suivant:

1. Soit X une séquence de m symboles $X=x_1x_2\dots x_m$
 2. Calculer la probabilité associée à chaque symbole p_i
 3. Associer à chaque symbole un sous intervalle $[L_i, H_i)$ proportionnel à sa probabilité, avec $H_i - L_i = p_i$.
 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. pour chaque symbole dans la séquence à coder on calcule :
 - largeur = $H - L$
 - $L = L + \text{largeur} \times L_i$
 - $H = L + \text{largeur} \times H_i$
 6. A la fin, n'importe quelle valeur de l'intervalle $[L, H)$ représente d'une manière unique la séquence d'entrée. Généralement, on choisi comme représentant de la séquence la limite inférieure (L) de l'intervalle, ou bien on peut choisir la valeur moyenne: $(L+H) / 2$.
- On appelle $\text{code}(X)$ la valeur choisie comme représentant de la séquence à coder.

Après le codage de la séquence X, on peut facilement décoder cette séquence en utilisant la table des probabilités (ou bien la table des fréquences) et en appliquant l'algorithme suivant :

1. D'abord on doit initialiser la limite inférieure (L) de l'intervalle de travail à la valeur 0 et la limite supérieure (H) à la valeur 1,

2. pour chaque symbole à décoder :

- largeur = H - L

- Chercher le sous intervalle [Li, Hi] du symbole à décoder sachant que :

$$Li \leq (\text{code}(X) - L) / \text{largeur} \leq Hi$$

- décoder le symbole xi.

$$-L = L + \text{largeur} \times Li$$

$$-H = L + \text{largeur} \times Hi$$

3. A la fin, on arrive à décoder la séquence X.

Il est à signaler que le codage arithmétique statique utilise une table des probabilités fixe durant les processus de codage et de décodage.

Considérons donc un exemple pratique: la chaîne X= "compressee"

On calcule la probabilité de chaque symbole :

Symbole (xi)	occurrences	Probabilités (pi)
c	1	0.1
e	3	0.3
m	1	0.1
o	1	0.1
p	1	0.1
r	1	0.1
s	2	0.2

Table (I.2) Calcule de probabilités de chaque symbole.

Le nombre de caractères de l'exemple dans ce cas est 10, ce qui simplifie grandement les manipulations numériques de l'exemple puisque les nombres auront un nombre fini de

décimales et ce procédé s'applique à des chaînes de longueur quelconque car il repose sur des intervalles et non sur les représentations exactes de nombres.

On décide maintenant d'associer à chaque symbole un domaine 'd' dans l'espace des probabilités, le but étant d'associer un symbole à un domaine. La table d'associations sera transmise avec le message encodé.

Symbole (xi)	Probabilités (pi)	Domaine $L_i \leq d < H_i$
c	0.1	$0.0 \leq d < 0.1$
e	0.3	$0.1 \leq d < 0.4$
o	0.1	$0.4 \leq d < 0.5$
m	0.1	$0.5 \leq d < 0.6$
p	0.1	$0.6 \leq d < 0.7$
r	0.1	$0.7 \leq d < 0.8$
s	0.2	$0.8 \leq d < 1.0$

Table (I.3) Associations d'un domaine pour chaque probabilité

Symbole (xi)	borne inférieure (L)	borne supérieure (H)
	0	1
c	0.0	0.1
o	0.04	0.05
m	0.045	0.046
p	0.0456	0.0457
r	0.04567	0.04568
e	0.045671	0.045674
s	0.0456734	0.045674
s	0.04567388	0.045674
e	0.045673902	0.045673968
e	0.045673908	0.045673946

Table (I.4) Associations d'un sous intervalle pour chaque probabilité

Ainsi $\text{code}(X) = 0.045673908$ est la représentation arithmétique du message "compressee". En pratique, n'importe quel nombre compris entre 0.045673908 et 0.045673946 représente le bon message.

Le décodage : sachant que le message encodé est entre 0.0 et 0.1, la première lettre est un c.

On applique $L_i \leq (\text{code}(X) - L) / \text{largeur} \leq H_i$

On soustrait ensuite la borne inférieure 0.0 pour (c) et on divise par la largeur pour (c); $[0.045673908 - 0.0] / 0.1 = 0.45673908$ on obtient ainsi qui est compris entre 0.4 et 0.5 donc la prochaine lettre sera un o etc.

I.3.1.2. Cordage RLE (Run Length Encoding) :

Le codage RLE est très répandu dans les algorithmes de compression d'images. Ce codage est beaucoup moins efficace que d'autres, il est facile à implémenter, rapide à exécuter. Le codage RLE travaille sur la réduction de la taille physique des chaînes de caractères répétitives. Ces répétitions se nomment en anglais un « run ».

-Le premier octet représente le nombre de répétition ; en pratique cette valeur est comprise le plus souvent entre 1 et 128 ou 256.

-L'octet suivant représente l'octet à itérer.

L'application du codage du RLE sur la séquence d'image de la figure (I.3) selon une direction horizontale donne le résultat suivant :

(7,4) (1,0)
 (1,4) (5,5) (1,4) (1,0)
 (1,4) (1,5) (3,6) (1,5) (1,4) (1,0)
 (1,4) (1,5) (1,6) (1,7) (1,6) (1,5) (1,4) (1,0)
 (1,4) (1,5) (3,6) (1,5) (1,4) (1,0)
 (1,4) (5,5) (1,4) (1,0)
 (7,4) (1,0)
 (7,4) (1,0)

Par contre pour une direction verticale, le RLE donne le codage suivant :

(9,4) (5,5) (3,4) (1,5) (3,6)
 (1,5) (3,4) (1,5) (1,6) (1,7)
 (1,6) (1,5) (3,4) (1,5) (3,6)
 (1,5) (3,4) (5,5) (10,4) (8,0)

L'efficacité du RLE dépend du nombre de transition des niveaux de gris de l'image à coder. Plus le nombre de niveaux gris est faible, plus le codage est meilleur. Le taux de compression atteint dépend de la complexité de l'image.

I .3.1.3. Méthodes à base de dictionnaire (LZW) :

Cet algorithme est un des plus répandu. Le premier élément de la famille des compresseurs LZ a été créé par A BRAHAM lempel et Jacob Ziv.

Ce compresseur nommé LZ77 est utilisé dans les programmes d'archivage de données. Il est spécialisé dans les données textuelles alors que LZ78 est plus efficace pour les données binaires comme les images.

Une amélioration des algorithmes précédents a été introduite par Terry Welch. Le résultat fut l'algorithme LZW. Le LZW est capable de travailler avec n'importe quel type de données. Il est rapide en compression et décompression et ne nécessite pas d'opération à virgule flottante. De par le fait qu'il encode au niveau du bit et non au niveau de l'octet. Il ne se soucie pas du processeur et de la manière dont il code les informations.

La méthode LZW consiste à remplacer par quelques bits, un mot, une phrase ou même un paragraphe entier. Ces bits sont constitués d'une manière unique à l'aide d'un dictionnaire créé au fur et à mesure des besoins. Bien entendu, il n'est pas question de transférer ou de sauvegarder le dictionnaire. Il doit être créé dynamiquement.

Le flux d'informations à compresser est découpé en chaîne d'octets. Chaque chaîne est comparée au dictionnaire, si elle n'est pas présente, elle est stockée. Elle est ensuite écrite dans le flot de sortie compressé. Quand une chaîne, déjà rencontrée, elle est codée et transmise si elle a une longueur inférieure au plus grand mot du dictionnaire.

Pour le décodage, le dictionnaire est reconstruit dans le sens inverse.

I .3.2. Méthodes irréversibles :

I .3.2.1. Sous échantillonnage (décimation) :

Le sous échantillonnage consiste à éliminer une moitié des données.

L'application de cette méthode sur un bloc d'image se fait de manière suivante :

- 1-on néglige une ligne de la matrice sur deux.
- 2-on néglige une colonne de la matrice sur deux.

Exemple :

Soit la matrice M de dimension 4*4 :

$$M = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}$$

Après sous échantillonnage on obtient la matrice M' :

$$M' = \begin{bmatrix} a & c \\ i & k \end{bmatrix}$$

Dans le cas de la reconstruction, l'image sera obtenue par interpolation en remplaçant chacun des pixels manquant par la moyenne des deux pixels adjacents.

Dans cette méthode la distorsion ne sera pas contrôlée.

I .3.2.2. Codage en sous bandes :

Le codage en sous bandes consiste à découper l'image en différentes bandes de fréquences spectrales pour les codées.

La bande spectrale de l'image originale est d'abord divisée en deux parties à l'aide de deux filtres numériques (passe bas et passe haut).

On obtient ainsi deux sous images de bandes de fréquences limitées, elles sont à leur tour sous échantillonnées puis filtrées.

Le résultat obtenue consiste une structure arborescente composée d'un nombre entier de sous bandes.

L'image est reconstruite par combinaison des sous – bandes à l'aide des filtres d'interpolation.

I .3.2.3. Codage par dictionnaire :

Les algorithmes de compression à base de dictionnaires codent les chaînes de symboles de longueurs variables comme des prototypes uniques. Ces prototypes forment un index dans un dictionnaire. On peut citer :

- **La quantification scalaire :**

La quantification scalaire consiste à remplacer un nombre très grand de symboles par un nombre restreint de codes. Par exemple, le fait d'arrondir est une quantification scalaire. Elle est moins complexe. On distingue la quantification scalaire uniforme et la quantification scalaire non uniforme. Dans le cas de la quantification scalaire uniforme les intervalles de quantification sont tous de la même taille, par contre la quantification non uniforme les intervalles de quantification sont de tailles différentes. Parmi les types de quantification scalaire, la quantification à zone morte qui est un type spécial de quantification, où l'intervalle autour de zéro est plus large. La zone morte ou "dead-zone" en anglais, qualifie par conséquent cet intervalle autour de zéro, autorise la totalité des valeurs de source reconnues comme petites, d'être quantifiées à une seule même valeur (généralement zéro).

Ce type de quantification est par conséquent non-uniforme (ou asymétrique). Cependant, si la totalité des intervalles sont uniformes, on qualifie le plus souvent ce type de quantification d'uniforme à zone morte.

Ce type de quantification est particulièrement utilisé en compression d'image, où suite à une transformation de l'image par ondelette ou DCT, il existe de très nombreuses valeurs autour de zéro, non-significatives, qui pénaliseraient la suite du processus de codage. Typiquement, les valeurs comprises dans la zone morte sont quantifiées à zéro, et ne sont par conséquent pas reconnues par le codage entropique.

- **La quantification vectorielle :**

La quantification vectorielle découpe la matrice de symboles en petits blocs, chaque bloc est codé par son indice dans un dictionnaire de bloc type, donc elle permet la quantification simultanée de plusieurs coefficients. Elle est coûteuse en termes de complexité calculatoire.

La quantification vectorielle développée par Greshu et Gray (1980, fait aujourd'hui l'objet de plusieurs publications dans la compression d'image numérique. La Figure(I.5) schématise un exemple d'organisation structurelle d'un quantificateur vectoriel.

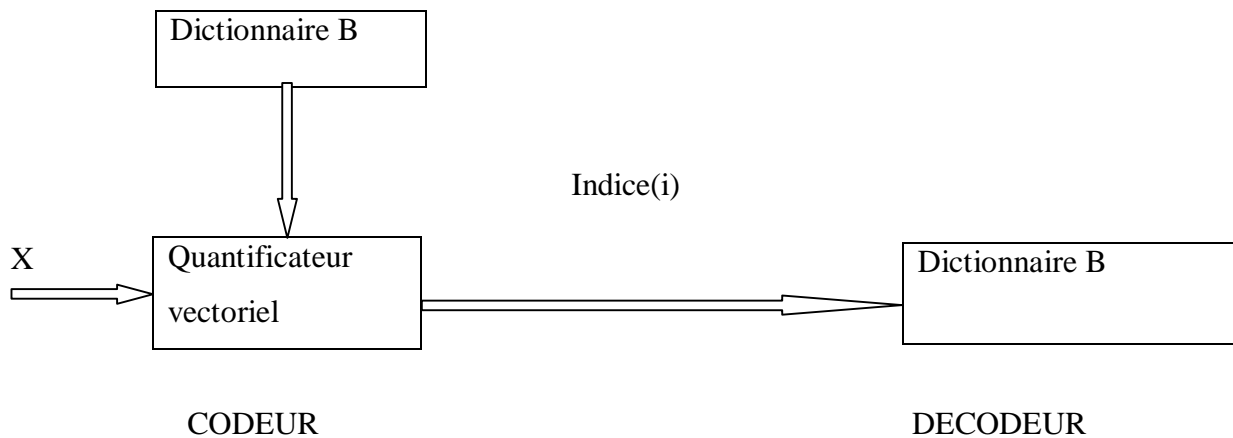


Figure (I.5) principe de la quantification vectorielle

I .3.2.4. Le codage par fractale :

Dans la technique précédemment citées, l'information pour chaque pixel est mémorisé ; par contre dans cette technique, ce sont les instructions ou des formules qui seront mémorisées pour la création de l'image. On démontre mathématiquement que pour chaque image il existe une formule de calcul.

A l'inverse de la quantification vectorielle, la compression fractale est asymétrique. Elle prend du temps pour la compression mais la décompression est rapide, elle consiste à lire en une simple lecture les formules mathématiques et puis à reconstituer l'image .La partie la plus difficile est de générer les formules qui représentent correctement l'image.la compression fractale, considère l'image composée de plusieurs petites images. Cette technique cherche les relations reliant les petites images composantes de l'image initiale puis les représenter par des formules mathématiques. Actuellement, la qualité des images reconstruites suite à une compression fractale est nettement inférieure à celle obtenue avec la norme JPEG.

I .3.2.5. Codage par transformée :

Dans ce type de codage, l'image de dimension $N \times N$ est subdivisée en sous images ou blocs de taille réduite chaque bloc subit une transformation mathématique orthogonal inversible linéaire du domaine spectral vers le domaine fréquentiel indépendamment des autres blocs (transformée en un ensemble de coefficients plus au moins indépendants).

Les coefficients obtenus sont alors quantifiés et codés en vue de leur transmission ou de leur stockage. Parmi ces transformations linéaires existantes :

1) Transformation de Fourier discrète (TFD)

La transformée de Fourier est la plus connue et la plus utilisée dans des nombreuses applications qui réalise de traitement d'image. C'est un outil mathématique qui réalise une transformation de signal dans le domaine fréquentiel. Elle permet de compacter grandement l'énergie moyenne sur un petit nombre de composant

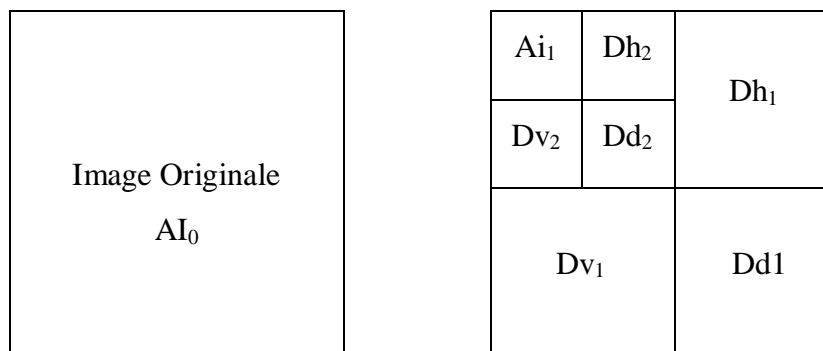
2) La transformation en cosinus discrète (DCT)

La transformée en cosinus discrète est une technique qui a été proposé par Ahmed Natarajan et Reo en 1974. Elle est l'une des transformations les plus utilisées pour la compression d'image adoptée dans nombreux standards de compression. Elle est très similaire à la transformée de Fourier, excepté le fait qu'elle n'utilise pas de valeurs complexes.

3) La transformation en ondelettes discrète (DWT)

La transformée en ondelettes est une méthode de la décomposition en sous bandes avec une manière particulière de choisir les filtres.

La décomposition en ondelettes aboutit à une image de détails et une image basse résolution. Cette dernière est à nouveau décomposée, et ainsi de suite, les filtres utilisés possédant la propriété de régularité qui rend ces filtres plus avantageux par rapport à ceux utilisée en sous bandes, la figure ci-dessous illustre le principe de décomposition en ondelettes basée sur une analyse multi résolution dyadique.



Figure(I.6) Schéma de représentation d'une image décomposée en ondelettes

I .3.2.6. Codage par DPC (Différentiel Prédictive Coding) :

Cette méthode exploite la corrélation qui existe entre un pixel de l'image et ses voisins. Dans son principe, elle établit une fonction de prédiction qui permet d'estimer la valeur d'un pixel en fonction de son voisin (la valeur des pixels voisins). On code alors, la différence (l'erreur) de prédiction qui représente l'écart entre la vraie valeur du pixel et la valeur prédite. L'écart étant en général faible, sa représentation nécessite moins de bits que le pixel lui-même. Les méthodes prédictives permettent une mise en œuvre facile et conduisent à des meilleurs taux de compression. Elles sont efficaces pour les images dont l'évolution temporelle ou spatiale est petite.

I .3.2.7. Méthodes hybrides :

Les méthodes hybrides sont une combinaison de plusieurs techniques de compression (spatiales et /ou transformées) visant à jumeler les avantages et inconvénients de celles –ci, afin d'avoir un taux de compression élevé, une meilleure qualité de restitution de l'image et un temps de calcul minimal lors du processus de codage. Nous présentons ci après ; quelques méthodes hybrides qui ont été développées au sein de notre laboratoire LAMPA (laboratoire d'analyse et de modélisation des phénomènes aléatoires).

-Compression d'images par des fractales dans le domaine de la DCT :

C'est l'association de la transformée en cosinus discrète et le formalisme fractal, les coefficients issus de la transformée en cosinus discrète ont été quantifiés puis codés par des fractales selon JPEG.

-Compression d'image par un codage fractale multi résolution domaine de la transformée en ondelettes :

C'est la combinaison entre le codage fractal et l'approche multi résolution de la transformée en ondelettes.

-Compression d'images par paquets d'ondelettes :

C'est l'application de la transformée en ondelette discrète pour la décomposition, puis les informations seront quantifiés et codés par le codeur LZ

Les résultats obtenus sur l'image test Lena sont répertoriés dans la table (I.5) pour quelque types de codeurs :

Types d'algorithmes	PSNR (dB)	Taux de compression C (%)
Ondelettes et fractales sans itération	32,90	93,75
DCT et QV en sous bandes	32,20	93,75
Ondelettes et fractales multirésolutions	30,08	72,36
Fractales et les algorithmes	34,15	85.76

Table (I.5) Comparaison des résultats obtenus sur l'image Lena.

I .4. Les normes de la compression d'image :

Des algorithmes de compression ont donné naissance à des normes avec des économies de 10 à 90% mais qui sont insuffisantes devant le problème de stockage (banques d'images ou l'utilisation de séquences vidéo sur ordinateur) ; ce qui a rendu nécessaire la mise en place sur le plan international de groupe de coordination et d'étude chargé de mettre aux points des standard adaptés à ces applications afin de rendre cohérent et compatible les échanges d'informations sur les canaux de communication . Les différentes normes de compression sont :

I .4.1. La norme JPEG :

La norme JPEG est une technique utilisée dans de nombreuses applications, notamment, dans le domaine de la photographie numérique et de la transmission de fichiers images sur Internet (80% des images disponibles sur le Web sont au format JPEG). Cette norme qui a été conçue par le groupe ISO (International Standards Organisation) et le groupe CEI (Commission Electronique International), est une méthode avec perte associant la DCT, la quantification scalaire et le codage entropique. Elle a pour objectif de répondre aux exigences suivantes :

- Performances équivalentes à celles des nouvelles techniques de compression, (taux de compression, qualité de restitution et temps de calcul) ;
- Possibilité de compresser tous types d'images ;
- Facilité d'implémentation ;
- Possibilité d'utiliser différents types de codeurs (séquentiel, progressif, hiérarchique)

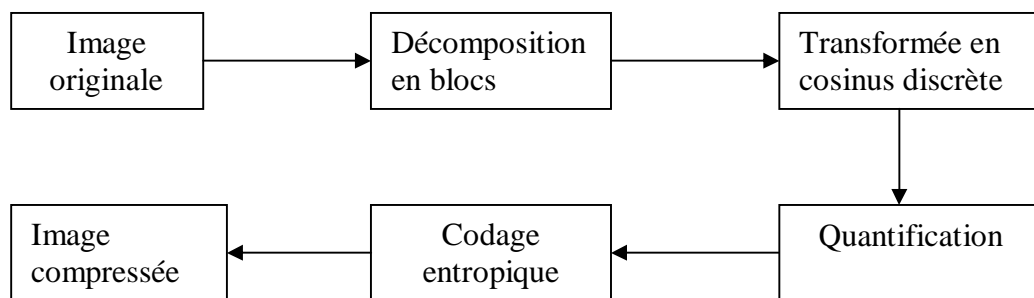


Figure (I.7) Chaîne de compression JPEG

I .4.2. La norme JPEG 2000 :

Remarquons toutefois que lorsqu'on fixe un taux de compression trop élevé les images après décompression présentent des effets de blocs très visibles. Pour palier cet inconvénient, une nouvelle norme, JPEG2000, a été mise au point. La version définitive du standard a pris forme en décembre 2000. Ce nouveau standard a pour objectif d'offrir de nouvelles fonctionnalités permettant de répondre à une demande croissante, cette méthode associe une transformée en ondelettes discrètes (DWT) qui est appliquée à l'image entière (traitement global) et un codage spécifique des sous bandes. Notons qu'un prétraitement est souvent appliqué à l'image originale pour mettre en évidence les régions les plus pertinentes et rendre ainsi le codage plus efficace.

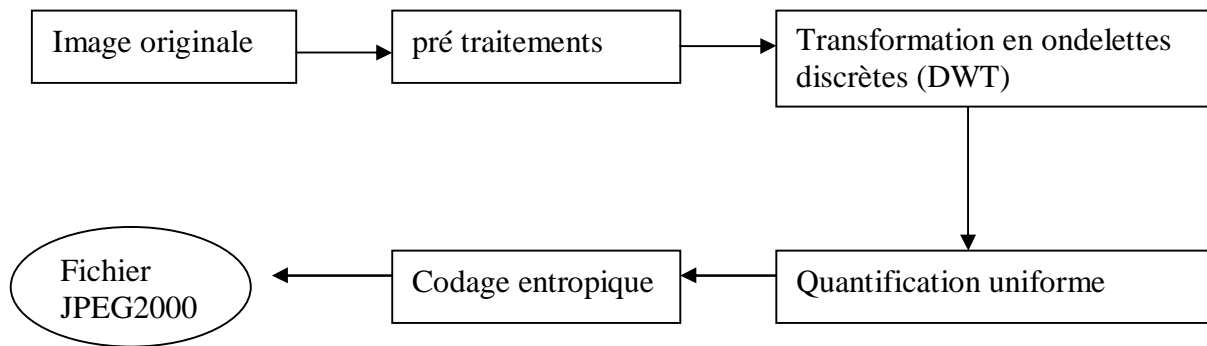


Figure (I.8) Chaîne de compression JPEG2000

I.4.3. Le standard CCSDS :

Le CCSDS est un comité consultatif pour les systèmes de données spatiales. Le groupe de travail a été engagé dans la recommandation des normes de compression de données pour les espaces d'applications.

En 1997, la première recommandation CCSDS de la compression de données a été adoptée, et en 1998 le groupe a commencé à évaluer la faisabilité d'établir une recommandation de compression d'image, il a récemment adopté une recommandation pour la compression des données avec un communiqué final attendu en 2005. L'algorithme adopté est la transformée en ondelettes bidimensionnelles suivie par le codage par plan de bits BPE progressif. Cet algorithme peut fournir à la fois une compression avec ou sans perte ; et permet de contrôler le volume comprimé de données et la fidélité avec laquelle les données ont été transformées.

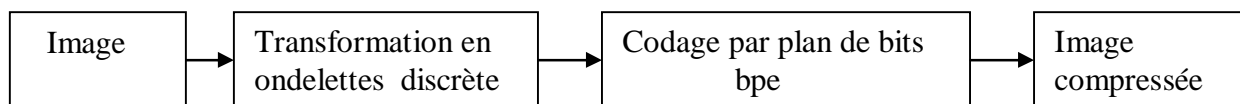


Figure (I.9) Chaîne de compression du standard CCSDS

I.5. Discussion :

L'image numérique, est un domaine dont l'application ne cesse de s'intensifier et de s'élargir, de ce fait la quantité d'information doit être réduite et pour cela diverses techniques de compression d'image ont été présentées dans ce chapitre.

Par la suite nous allons spécifier notre travail qui consistera en la compression des images par la transformée en ondelettes qui sont l'un des outils les plus efficaces actuellement dans le traitement d'images, donc dans le chapitre suivant ; on va présenter, avec profondeur, cette nouvelle alternative de traitement d'image qui est la théorie des ondelettes.

CHAPITRE II

La théorie des ondelettes

II.1.Préambule:

La transformée en ondelettes a été introduite au début des années 1980, c'est une méthode avancée dans l'analyse des signaux, elle a fait preuve de supériorité par rapport à ces voisines (transformée de Fourier, transformée de Gabor)

La transformation intermédiaire entre les domaines temporel et fréquentiel s'est basée jusqu'aux années 80 sur l'analyse de Fourier. Cette dernière permet la décomposition d'un signal en une série défini par l'équation ci-dessous:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt \quad (\text{II.1})$$

Les inconvénients de la transformée de **Fourier** sont liés à l'impossibilité de décrire les propriétés locales des fonctions à partir de leurs propriétés spectrales. Cette analyse nous donne une information globale sur le signal, Pour pallier à ses inconvénients, en 1946, **Denis Gabor** a élaboré la transformée de Fourier à fenêtre glissante (**TFG**). Cette transformation associe à un signal une suite de transformations de Fourier sur de petits intervalles de temps, dans le but de localiser l'analyse de Fourier, en utilisant une 'fenêtre spatiale glissante' dans l'intégrale de Fourier.

la fenêtre est représentée par une fonction gaussienne $G(t)$:

$$G_{a,b}(t) = e^{-iat} G(t-b) \quad a, b \in \mathcal{R} \quad (\text{II. 2})$$

Et sa représentation est donnée par la figure si dessous avec un décalage de temps t :

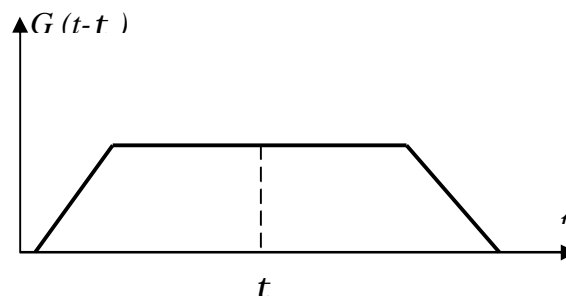


Figure (II.1) fenêtre de Gabor décalée du temps t

La transformée de Fourier à fenêtre glissante est de taille fixe, ce qui ne convient pas à l'analyse des signaux non stationnaire (signaux ayant des composantes de tailles différentes de la fenêtre).

J.Morlet et Gossmann ont introduit une transformée qui s'adapte au contenu fréquentiel du signal à analyser pour l'étude haute résolution des signaux sismiques ; elle permet une analyse plus précise dans les deux domaines : fréquentiel et temporel. Elle est appelée la transformée en ondelettes.

II.2.Eléments de la théorie des ondelettes :

II. 2.1.Définition :

Les ondelettes sont une famille de fonctions engendrées par translation en temps et dilatation en fréquence à partir d'une même fonction ψ , appelée ondelette mère. Elles sont normalisées suivant l'équation (II.3) :

$$y_{a,b}(t) = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right) \quad (\text{II.3})$$

Avec : $\psi(t)$: fonction bien localisée en temps et en fréquence

a : paramètre de dilatation.

b : paramètre de translation

II.2.2.types d'ondelettes :

Ü Les ondelettes orthogonales :

Les ondelettes constituent des familles de fonctions normalisées et oscillantes, générées à partir d'une fonction unique appelée ondelette mère. Ces dernières sont caractérisées par des ondes localisées. Cette propriété de localisation lui permet d'agir localement sur le signal. On la désigne souvent par la lettre ψ .

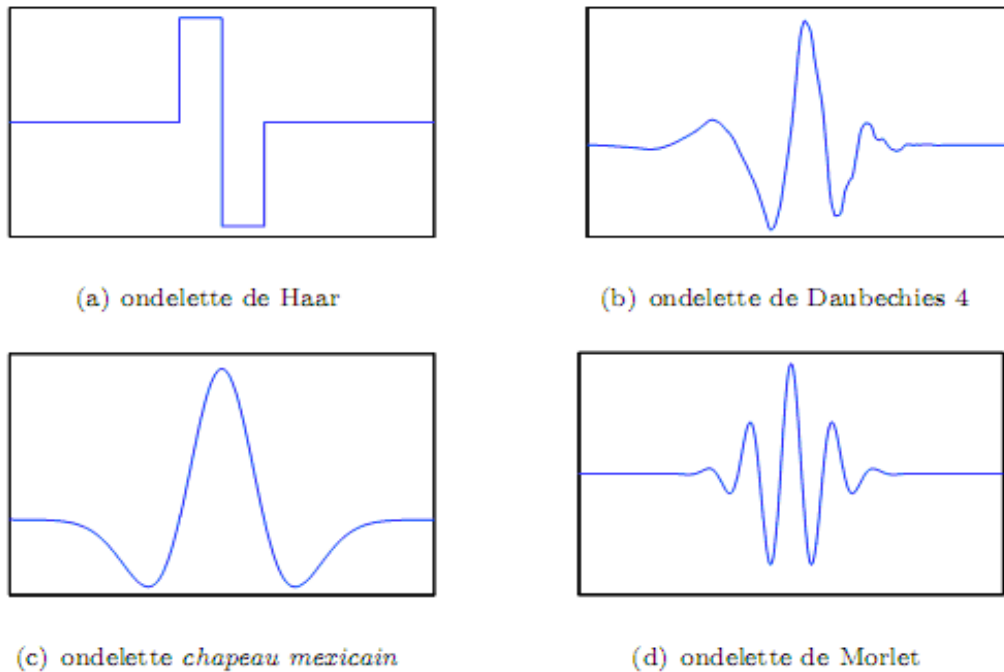


Figure (II.2) quelques types d'ondelettes

ü Les ondelettes biorthogonales :

Cette famille d'ondelettes présente la propriété de la phase linéaire, qui est nécessaire pour la reconstruction de signal et d'image. En employant deux ondelettes, une pour la décomposition et l'autre pour la reconstruction.

II.2.3. Propriété des ondelettes :

La grande variété d'ondelettes conduit à l'étude de leurs propriétés, et leurs influences sur le traitement de signal, en particulier sur les signaux bidimensionnels. L'ondelette est caractérisée par les propriétés suivantes :

- **La symétrie :**

La symétrie est utile pour éviter les déphasages non linéaires provocateurs de distorsion de l'image, pour cela il est préférable d'utiliser les bases d'ondelettes bi orthogonales.

- **Oscillation :**

Ceci se traduit par plusieurs passages par 0 de l'ondelette ψ . Elle peut être exprimée par le nombre de moments nuls de l'ondelette. En pratique, les ondelettes qui possèdent N moments nuls conduit à annuler tous les coefficients d'ondelettes des polynômes de degré inférieur à N .

- **Régularité :**

Est une propriété importante dans la phase de synthèse à cause de la sensibilité de l'œil à l'introduction de nouvelles singularités.

- **Orthogonalité :**

Si les ondelettes sont orthogonales, la transformée en ondelettes est stable et réversible.

- **Localisation :**

Cette propriété est très importante dans l'imagerie, car elle permet de caractériser assez précisément les phénomènes locaux.

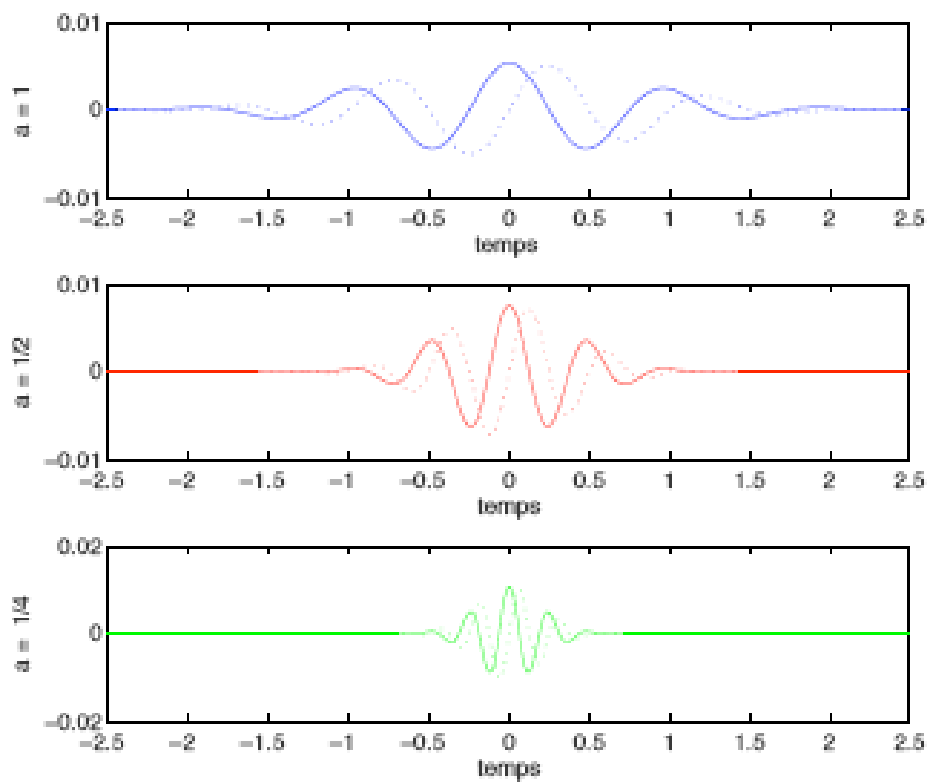
II.3. Les types de la transformée en ondelettes :

II.3.1. La transformée en ondelettes continue (TOC) :

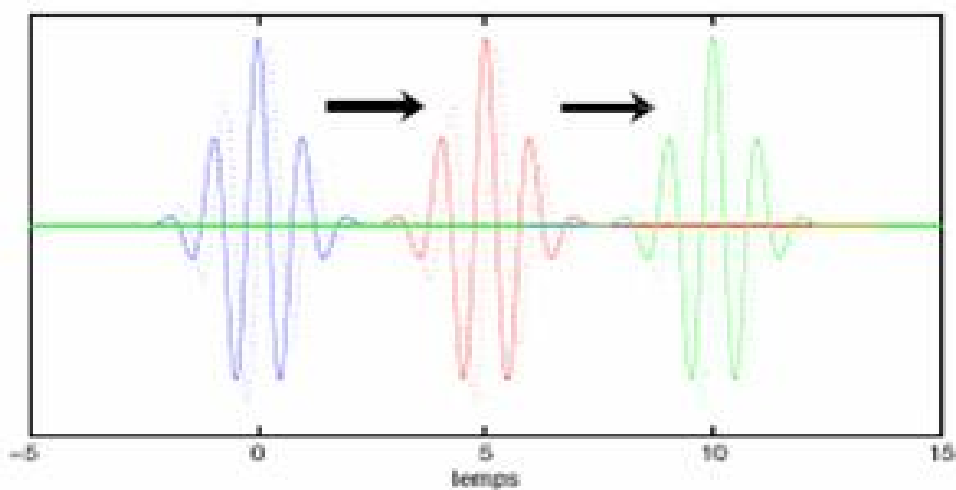
La transformée en ondelette est dite continue car les paramètres de dilatation de translation prennent des valeurs continues. La transformée en ondelettes d'une fonction f est une projection de celle-ci sur la famille d'ondelettes choisie ; elle est définie par l'équation (II.4) :

$$W(a, b) = CWT(f)(a, b) = \langle \psi_{a,b}, f \rangle = \int_{-\infty}^{+\infty} |a|^{-1/2} \left(\frac{x-b}{a} \right) f(x) dx \quad (\text{II.4})$$

Les figures (II.3)[a] et (II.3)[b] illustrent les effets de dilatation et de la translation d'une ondelette analysante.



(a) Changement d'échelle.



(b) Changement de position.

Figure (II.3) La dilatation et translation d'ondelette

La première présentation dans la figure ((II.3) [a]) est l'ondelette de référence, elle correspond à une ondelette non dilatée et non translatée : $a=1$ et $b=0$,

La deuxième et la troisième correspondent à des ondelettes comprimées avec $a=1/2$ et $a=1/4$, non translatées de $b=0$, représentées par la figure ((II.3) [a]). Dans la dernière figure (II.3. [b]), nous avons présenté l'ondelette de référence translatée dans les valeurs positive avec $b=0$, $b=5$, $b=10$.

La transformée inverse de la TOC n'est pas unique du fait de la redondance de l'information ; c'est pour cela que l'ondelette y doit vérifier la condition d'admissibilité suivante :

$$C_y = \int_{-\infty}^{+\infty} \frac{|\hat{y}(w)|^2}{w} dw < \infty \tag{II.5}$$

La transformée inverse est donnée par l'équation (II.6)

$$f(x) = \frac{1}{C_y} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} w(a,b) y_{a,b}(x) \frac{dad b}{a^2} \tag{II.6}$$

II.3.2. La transformée en ondelettes discrète (TOD) :

La transformée en ondelettes discrète est une transformation bien localisée en temps et en fréquence, dont les paramètres sont discrétisés en vue d'une analyse spectrale des signaux numériques. Les paramètres a et b prennent des valeurs discrètes, selon l'expression suivante :

$$(a, b) = (a_0^j, kb_0 a_0^j) \text{ ou } (j, k) \in \mathbb{Z}^2 \tag{II.7}$$

Où : $a_0 \neq 1$ (le pas de dilatation)

$b_0 \neq 0$ (le pas de translation)

Ainsi, la nouvelle famille d'ondelettes s'établit par :

$$y_{j,k}(t) = a_0^{-j/2} \cdot y(a_0^{-j}t - kb_0) \quad (\text{II.8})$$

La transformée en ondelettes discrètes d'une fonction est donnée par l'équation suivante :

$$W(a,b) = \langle y_{j,k}, f \rangle = a_0^{-j/2} \int_{-\infty}^{+\infty} (a_0^{-j}t - kb_0) f(t) dt \quad (\text{II.9})$$

Y.MEYER a montré qu'en choisissant des opérateurs de dilatation et de translation dyadique ($a=2$ et $b=1$), la réduction de redondance de la représentation en ondelettes est maximale. Les coefficients y_{jk} forment une base Hilbertienne de $L^2(\mathbb{R})$

Pour $a=2^{-j}$ et $b=2^{-j} \cdot k$, on obtient :

$$\{y_{j,k}\}_{j,k \in \mathbb{Z}} = \{2^{j/2} \cdot y(2^j t - k)\}_{j,k \in \mathbb{Z}} \quad (\text{II.10})$$

La reconstitution de la fonction $f(x)$ est obtenue grâce à l'équation suivante :

$$f(x) = \sum_{j,k \in \mathbb{Z}} \langle f(x), y_{j,k}(x) \rangle y_{j,k}(x) \quad (\text{II.11})$$

II.4. Analyse multi résolution :

Elle consiste à représenter un signal comme une limite de ses approximations successives, ou chaque approximation est une version de la précédente. Les approximations successives sont présentées à différentes résolutions, d'où le nom multi résolution.

Y.MEYER et S. MALLAT, définissent l'analyse multi résolution comme étant une suite de sous espaces vectoriels $(V_j) \in L^2(\mathbb{R}^n)$ vérifiant les propriétés suivantes :

1. $\forall j \in \mathbb{Z} \ V_j \subset V_{j+1}$.
2. $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$.
3. $\bigcup_{j \in \mathbb{Z}} V_j$ est dense dans $L^2(\mathbb{R}^n)$.

4. $\forall f \in L^2(\mathbb{R}^n), \forall j \in \mathbb{Z}, \text{ on a : } f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$
5. $\forall f \in L^2(\mathbb{R}^n), \forall k \in \mathbb{Z}^n, \text{ on a : } f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0.$
6. Il existe une fonction $g(x)$ dans V_0 telle que $\{g(x-k)\}_{k \in \mathbb{Z}^n}$ soit une base de Riesz pour l'espace V_0 .

II.4.1. Commentaires sur les propriétés de l'analyse multi résolution :

- L'information contenue dans V_j est également incluse dans V_{j+1} , cela signifie que plus on monte en résolution plus le signal tend vers le signal original (propriétés 1 et 3).
- En approximant une fonction f à des résolutions 2^j , des informations sont de plus en plus perdues quand j tend vers $-\infty$ i.e. $\lim_{j \rightarrow -\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\}$.
- Un signal appartenant à un sous espace vectoriel V_j , a ses détails dilatés d'un facteur de 2 quand il est considéré dans V_{j+1} (propriété 4).
- Un signal translaté reste invariant par rapport à la résolution (propriétés 5).

II.4.2. Fonction d'échelle F et sous espaces V_j :

On construit une analyse multi résolution à l'aide de sous espaces V_j emboîtés les un dans les autres, tels que le passage de l'un à l'autre soit le résultat d'un changement d'échelle. Pour réaliser les approximations d'une fonction $f \in L^2$ on fait appel à une fonction d'échelle ou père d'ondelettes $\Phi(x) \in L^2(\mathbb{R}^n)$ telle que :

$$\Phi_{j,n} = 2^j \Phi(2^j x - n) \quad (j, n) \in \mathbb{Z} \times \mathbb{Z} \tag{II.12}$$

La fonction Φ constitue une base orthogonale pour (V_0) .

La projection de $f(x)$ sur V_j décomposée sur les bases orthonormales donnée par l'équation suivante :

$$P_j f(x) = \sum_{n=-\infty}^{+\infty} (f(u), \Phi_j(u - 2^j n)) \Phi_j(u - 2^j n) \tag{II.13}$$

Les coefficients $A_j^d = \{(f(u), \Phi_j(u - 2^j n))\}_{n \in \mathbb{Z}}$ décrivent l'approximation discrète de $f(x)$ à la résolution 2^j . La projection $P_j f(x)$ définie ci-dessus est caractérisée uniquement par l'ensemble des coefficients des produits internes $A_j^d f$

II.4.3.Fonction ondelette et l'espace de détails W_j :

On a vu que si $(V_j)_{j \in \mathbb{Z}}$ est une approximation multi résolution de $L^2(\mathbb{R})$ alors :

$$V_j \subset V_{j-1} \quad \forall j \in \mathbb{Z}$$

On peut alors déterminer un espace W_j qui est le complément orthogonal de V_j dans V_{j-1} tel que :

- $V_{j-1} = V_j \oplus W_j$ et $V_j \perp W_j$
- $\forall J, j \in \mathbb{Z}, J \neq j \Rightarrow W_J \perp W_j$
- $\bigoplus_{j \in \mathbb{Z}} W_j = L^2(\mathbb{R})$

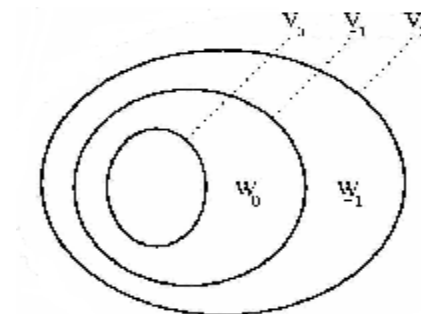


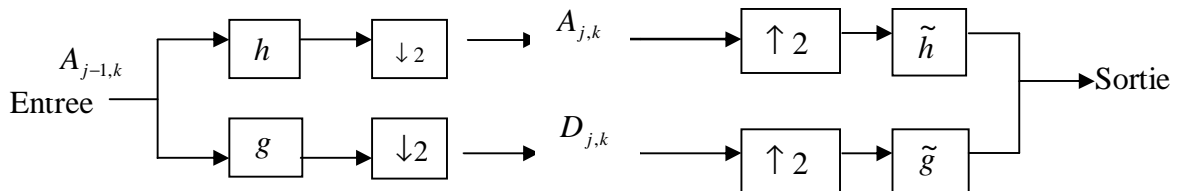
Figure (II.4) Les espaces emboîtés de l'analyse multi-résolution

Pour j fixé, les coefficients $(D_{j,n}(f) = (f, \Psi_{j,n}))_{n \in \mathbb{Z}}$ représente la différence entre deux approximations de f à des résolutions différentes, soit l'information perdue quand on passe d'une approximation de f d'une résolution $2^{-(j-1)}$ à une résolution 2^{-j}

II.4.4.Filtres associés aux ondelettes :

Les signaux qu'on manipule dans la pratique, sont des signaux échantillonnés. Le traitement spatio-fréquentiel de ses signaux est effectué à l'aide de filtres numérique qui

engendrent une analyse multi résolution sous certaines conditions. et bancs de filtres associés aux fonctions d'échelles et ondelettes sont donnés par le schéma suivant :



$\downarrow 2$ Décimation par 2

$\uparrow 2$ Zéros entre 2 échantillons (interpolation)

Figure (II.5) décomposition et reconstruction à l'aide de bancs de filtres

h et g : sont respectivement les filtres passe bas et passe haut utilisés dans l'étape de la décomposition.

\tilde{h} et \tilde{g} : sont respectivement les filtres passe bas et passe haut utilisés dans l'étape de la reconstitution.

Le signal d'entrée du banc de filtre est décomposé par filtrage avec décimation, donc on obtient deux sous bandes, une sous bande basse fréquence et une sous bande haute fréquence.

La reconstruction du même signal est obtenue en sommant les deux signaux filtrés par \tilde{h} et \tilde{g} après avoir effectué l'opération d'interpolation.

II.4.5. Fonctions d'échelle et ondelettes à deux dimensions :

La fonction d'échelle $\Phi(x,y)$ dans le cas bidimensionnel s'écrit comme le produit de deux fonctions d'échelles monodimensionnelles :

$$\Phi(x,y) = \Phi(x) \Phi(y) \tag{II.14}$$

Le calcul des différentes approximations ainsi que les coefficients d'ondelettes, sont réalisés en effectuant des convolutions à l'aide de filtres bidimensionnels séparables. Les filtrages sont appliqués de manière indépendante sur les lignes et les colonnes de l'image suivant l'algorithme mentionné dans le cas monodimensionnel.

Pour un niveau de résolution m donné, l'image originale est représentée par un ensemble de 3^{m+1} sous images.

La notion de multi résolution nous permet de considérer l'image sur plusieurs échelles. Les échelles consécutives sont séparées par un facteur de résolution 2

II.4.6. Ondelettes et analyse multi résolution :

La multi résolution hiérarchique permet de représenter une image sous un format réduit. L'ensemble des images aux différentes résolutions a l'aspect d'une pyramide, d'où le nom souvent utilisé « approche pyramidale ». Ceci présente un grand avantage en traitement d'images du fait que l'image réduite occupe moins d'espace en mémoire, ce qui permet d'accélérer les traitements. Plusieurs études sur l'analyse multi résolution des images ont été effectuées. Les ondelettes permettent de représenter la différence d'informations existant entre le signal de départ et son approximation à une résolution plus grossière.

II.4.7. Application de l'algorithme de S Mallat :

II.4.7.1. Algorithme de décomposition :

En utilisant l'algorithme de décomposition de Mallat en 2D, une valeur sur deux de l'image originale est supprimée selon les lignes et selon les colonnes à chaque passage de l'ondelette.

Les résolutions des approximations découlant de la décomposition sont donc toujours réduites par rapport à l'image précédente. Chaque passage d'ondelette a pour effet de produire une nouvelle image d'approximation et trois images de détail, dont la résolution vaudra : $r_i = r_0 2^j$
avec

j : niveau de décomposition

r_0 : résolution de l'image originale

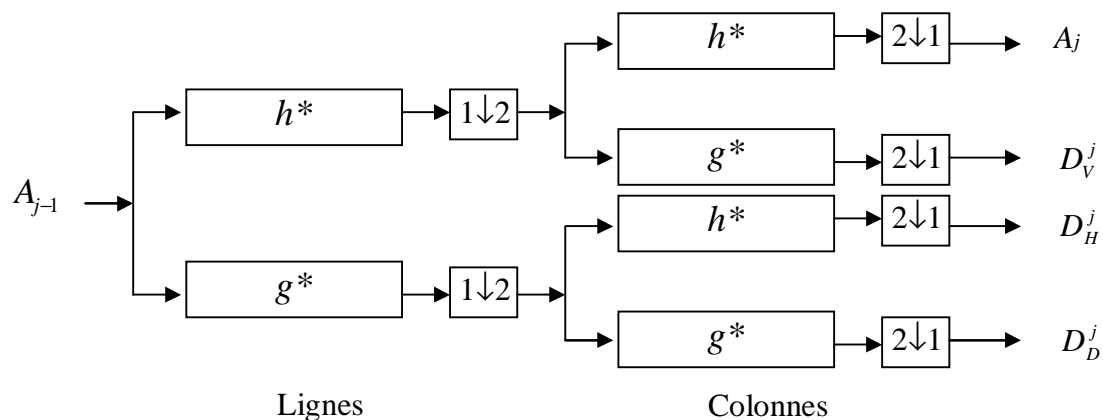
r_1 : résolution à l'approximation en question

1- Nous appliquons une transformation suivant les lignes. Ce qui donne deux images, la première correspond aux coefficients sous échantillonnés du filtrage passe-bas (figure (II.7).a) et la deuxième aux coefficients sous-échantillonnés du filtrage passe-haut (figure(II.7).a.)

2- Puis nous appliquons une transformation pour les deux images précédentes suivant les colonnes, ceci donne quatre types de coefficients :

- Les coefficients issus de la convolution avec le filtre passe-bas h suivant les lignes et les colonnes ;

-Les coefficients issus de la convolution avec le filtre passe-bas suivant les colonnes et les lignes ; (figure (II.7).b et c)



Avec h et g les filtres passe bas et passe haut respectivement.

X^* Convolution avec le filtre X^*

$1 \downarrow 2$ Décimation d'une colonne sur deux

$2 \downarrow 1$ Décimation d'une ligne sur deux

Figure (II.6) algorithme de décomposition

La procédure de décomposition est itérée sur l'imagette d'approximation. Les figures suivantes montrent les imagettes obtenues par une décomposition en TOD de niveau 2 de l'image Lena. Cette décomposition donne une pyramide d'images (Figure (II.8)) de détails à différentes échelles et orientations et une approximation (Figure (II.7)).

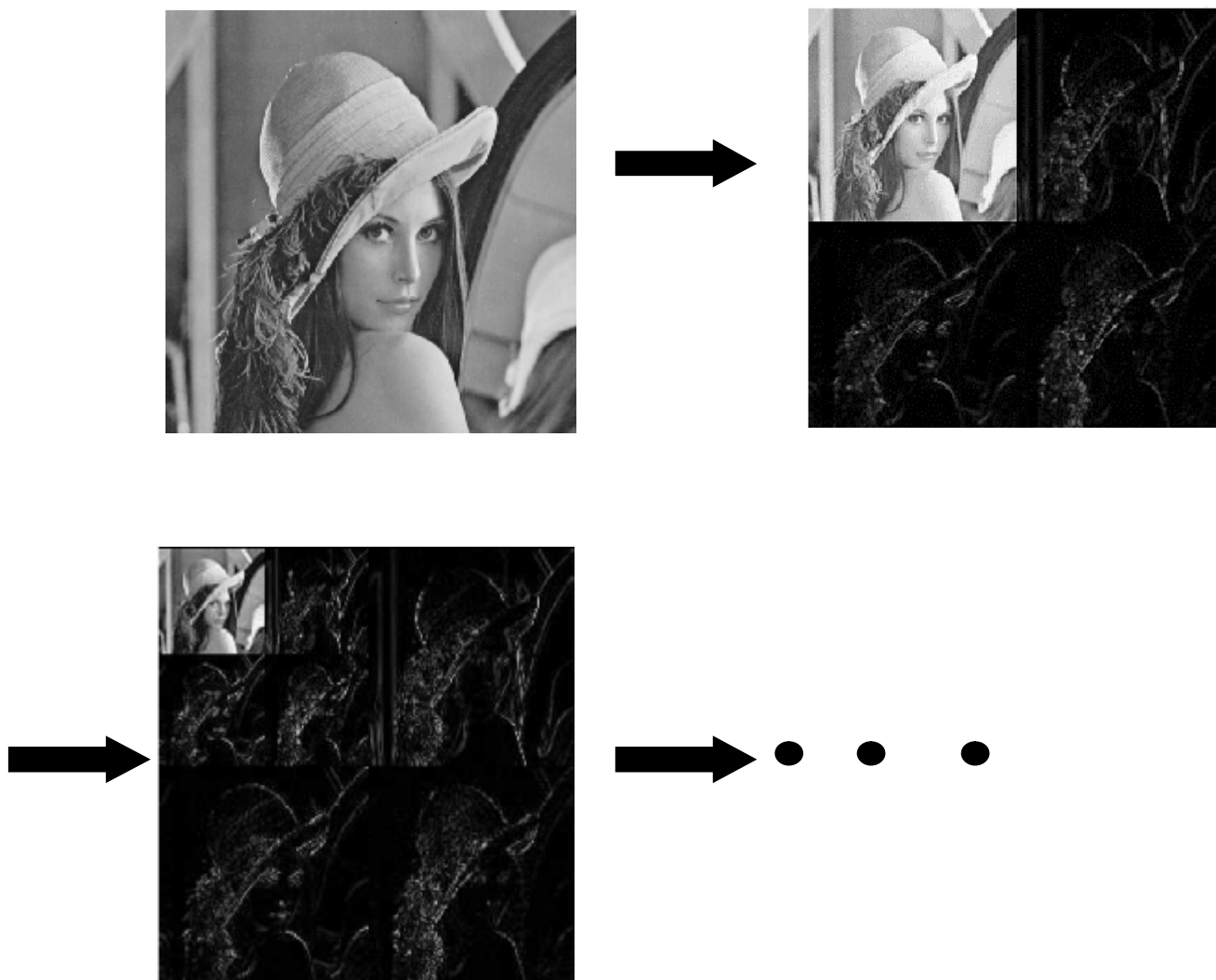


Figure (II.7) Exemple de décomposition en multi-résolution de l'image niveaux 1 et 2.

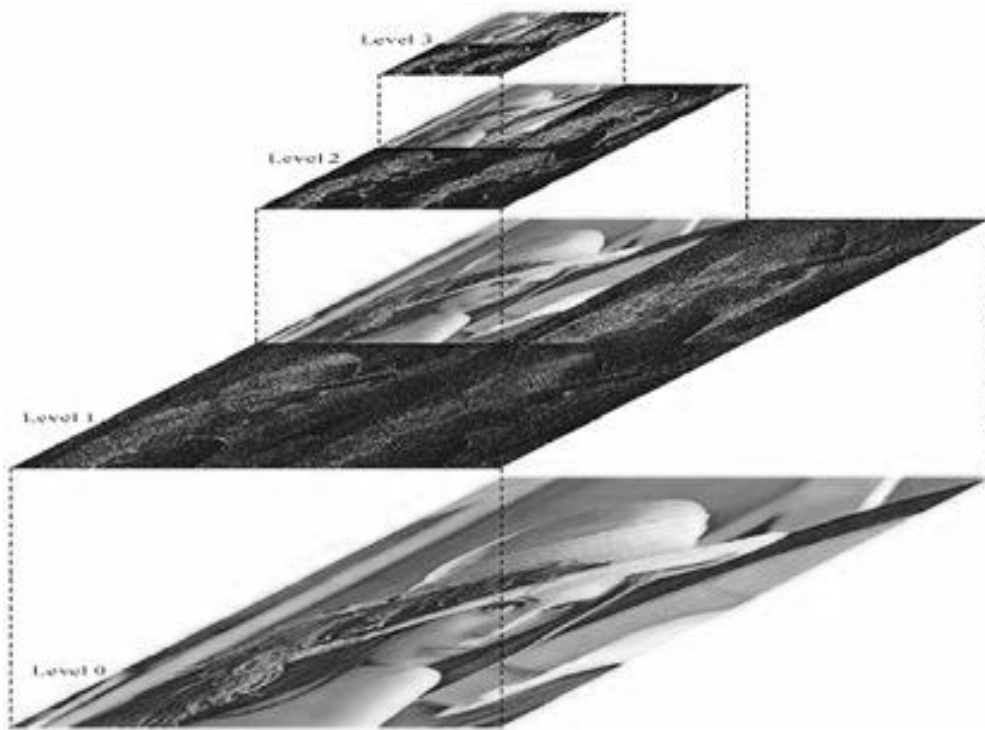
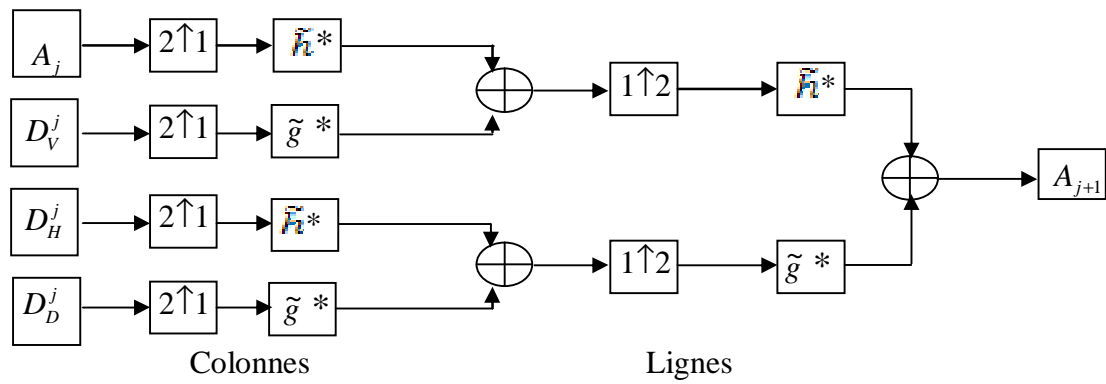


Figure (II .8) Représentation Pyramidale de l'image Lena

II.4.7.2. Algorithme de reconstruction :

La transformée en ondelettes inverse permet de reconstruire parfaitement le signal. En 2D, on peut déterminer l'image de niveau de décomposition $(j-1)$ à partir de l'image d'approximation et des trois images de détail. La reconstruction des approximations bidimensionnelles est schématisée par la figure (II.9)



X^*

Convolution avec le filtre X^*

$2 \uparrow 1$

Interpolation d'une ligne de zéros entre deux

$1 \uparrow 2$

Interpolation d'une colonne de zéros entre deux

Figure (II.9) Algorithme de reconstruction 2D

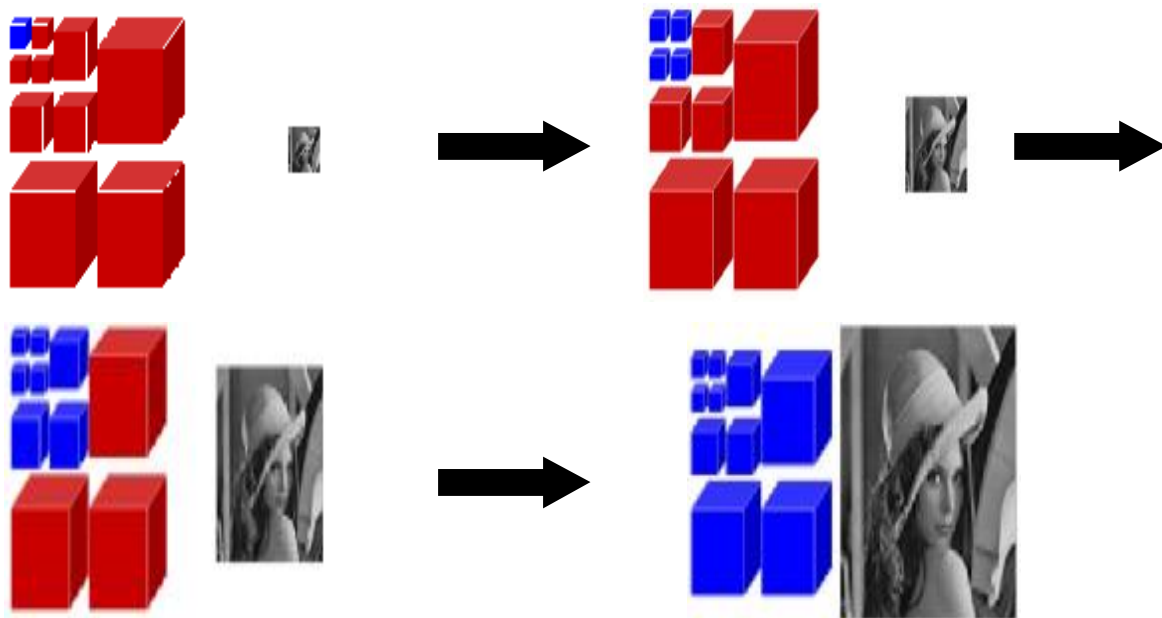


Figure (II.10) Schéma de reconstruction d'une image à 2D

II.5.Discussion :

Dans ce chapitre, nous avons fait une analyse plus au moins détaillée de la théorie des ondelettes, en commençant par la TOC(Transformation en ondelettes continue), puis la TOD(Transformation en ondelette discrète), et enfin la multi résolution. Nous avons donc pu conclure, que l'utilisation des ondelettes permet de nombreuses applications au niveau de la compression d'image, et elles apportent de bien meilleurs résultats que les algorithmes utilisés en imagerie multimédia pour la compression.

Dans le prochain chapitre, nous allons illustrer la méthode adoptée pour la compression des images, on utilisant le codeur par plan de bits BPE du standard CCSDS.

CHAPITRE III

Méthode adoptée

III.1.Préambule:

Après avoir présenté tous les outils nécessaires pour définir notre travail au cours des chapitres précédents, nous allons voir dans ce chapitre le développement de la méthode choisie qui combine la transformée en ondelettes bidimensionnelles et la quantification scalaire à zone morte suivi du codage par plan de bits BPE progressif. Cet algorithme peut fournir à la fois une compression avec ou sans perte, selon l'ondelette choisie, et permet de contrôler le volume comprimé de données et la fidélité avec laquelle les données ont été transformées. Ici on s'intéresse à la compression avec perte.

III.2.Schéma synoptique de la méthode adoptée:

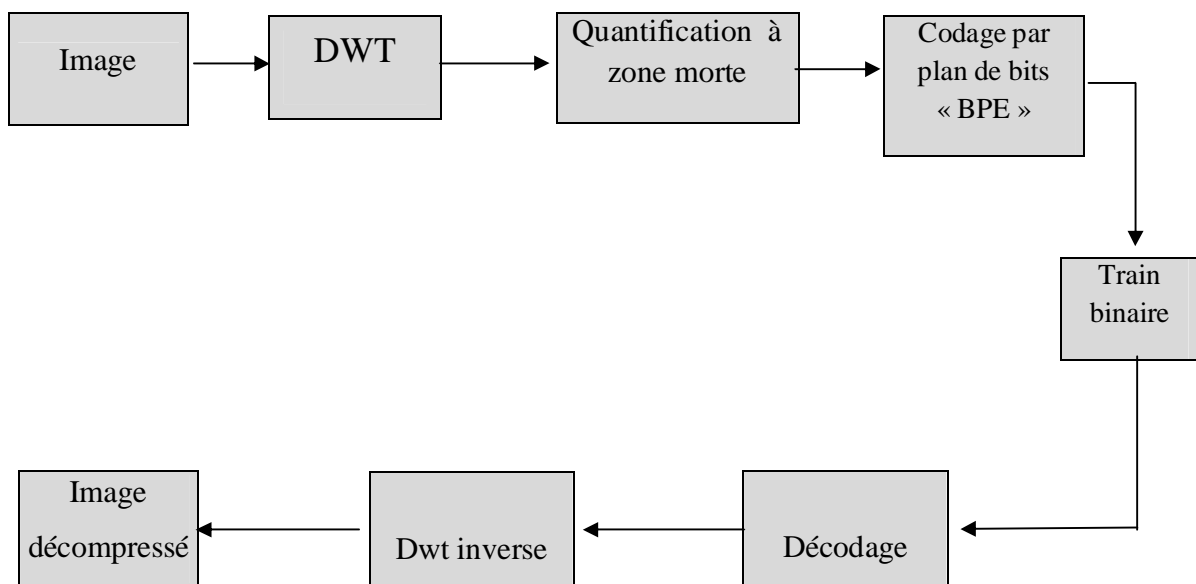


Figure (III.1) Schéma synoptique de la méthode adoptée

III.2.1. La transformée en ondelette discrète:

La transformée en ondelettes retenue pour la compression d'images dans le standard du CCSDS est la transformée en ondelettes biorthogonale 9/7, c'est celle qui permet d'atteindre les meilleures performances en compression, c'est aussi la transformée en ondelettes utilisée dans la norme JPEG2000. Le nombre de niveaux de décomposition en ondelettes a été fixé à 3. Ce nombre de niveaux de décomposition est un compromis entre performances en compression et complexité calculatoire. De plus, un faible niveau de décomposition réduit le nombre de pixels affectés par la perte de données compressées.

III.2.2. Quantification à zone morte:

Pour la quantification des coefficients de la transformée en ondelettes d'une image, il est montré que la quantification scalaire à zone morte est un très bon choix au sens de la distorsion engendrée.

Cette quantification consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur appelée seuil. Dans notre travail on a choisi comme seuil de quantification la moyenne de chaque sous bande, il est calculé de la façon suivante :

$$seuil = \frac{1}{N} \sum_{i=0}^n \sum_{j=0}^m C_{ij}$$

C_{ij} : Coefficient d'une sous bande de coordonnées i et j

N : Nombre des coefficients d'une sous bande ; $N=1 \dots n$

III.2.3. Codage par plan de bits BPE :

Après une transformée en ondelettes sur trois niveaux, notre algorithme traite des segments de S blocs.

Un bloc est un ensemble de 64 coefficients d'ondelettes correspondant à une région de taille 8×8 pixels dans l'image originale. Chaque bloc comporte:

- 1 coefficient de la sous-bande LL_3 ;
- 3×1 coefficients des sous-bandes HL_3, LH_3 et HH_3 : les parents ;
- 3×4 coefficients des sous-bandes HL_2, LH_2 et HH_2 : les enfants ;
- 3×16 coefficients des sous-bandes HL_1, LH_1 et HH_1 : les petits-enfants ;

	Famille 0	Famille 1	Famille 2
Parents	HL_3	LH_3	HH_3
enfants	HL_2	LH_2	HH_2
Petits enfants	HL_1	LH_1	HH_1

Table (III.1) les sous bandes des coefficients AC d'un bloc

Groupes de Coefficients d'une famille	Cordonnés
Parent P_i	(r, c)
Enfants C_i	$(2r, 2c), (2r, 2c+1), (2r+1, 2c), (2r+1, 2c+1)$
Petits enfants Hi_0	$(4r, 4c), (4r, 4c+1), (4r+1, 4c), (4r+1, 4c+1)$
Petits enfants Hi_1	$(4r, 4c+2), (4r, 4c+3), (4r+1, 4c+2), (4r+1, 4c+3)$
Petits enfants Hi_2	$(4r+2, 4c), (4r+2, 4c+1), (4r+3, 4c), (4r+3, 4c+1)$
Petits enfants Hi_3	$(4r+2, 4c+2), (4r+2, 4c+3), (4r+3, 4c+2), (4r+3, 4c+3)$

Table (III.2) la répartition des coefficients d'ondelettes d'une famille i

La figure (III.2) illustre cette répartition dans un bloc. Le codage des coefficients d'ondelettes est effectué par plans de bits sur ces blocs de coefficients. Les coefficients de la sous-bande LL_3 sont codés séparément des coefficients des autres sous-bandes.

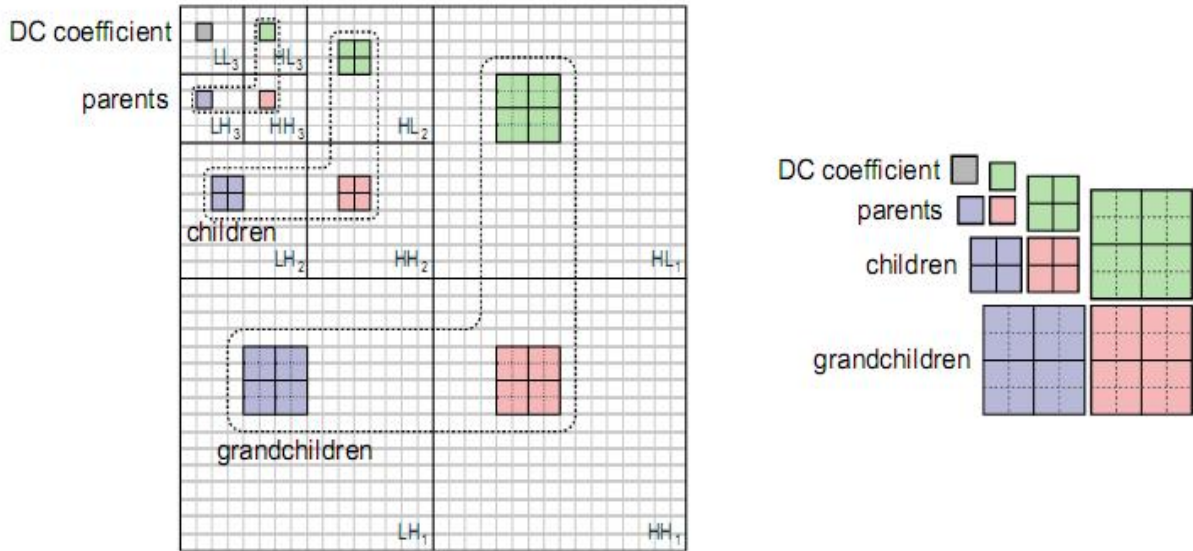


Figure (III.2) Bloc de 64 coefficients d'ondelettes tel que défini dans le standard du CCSDS

Pour décrire les techniques de codage par plans de bits, il faut introduire la notion de coefficients significants.

On dit que le coefficient d'ondelettes x est significatif au plan de bit b si : $|x| \geq 2^b$.

Soit un coefficient d'ondelettes x tel que $2^b \leq |x| < 2^{b+1}$.

Le processus de codage est effectué du plan de bits le plus significatif vers les plans de bits les moins significatifs. Le coefficient x n'est pas significatif à tous les plans de bits supérieurs à b : il n'y a que des bits 0 à coder. Ce coefficient x devient significatif au plan de bits b : le bit à coder est 1, dès lors on sait que $2^b \leq |x| < 2^{b+1}$ donc c'est la fin de la partie codage de la signifiante pour ce coefficient. Le bit de signe du coefficient x est codé dès que ce coefficient devient significatif, c'est-à-dire dès que le premier bit 1 a été codé pour ce coefficient. On commence ensuite la partie de raffinement de l'amplitude.

Le codeur BPE du CCSDS exploite des plages de zéros par l'utilisation des mots de transition. Aussi, afin de localiser ces plages de zéros, plusieurs ensembles de coefficients sont définis sur la figure (III.2)

- **Les ensembles de coefficients :**

- les coefficients parents sont notés P_i avec $i \in [0, 2]$;
- les ensembles de 4 coefficients enfants de la même sous-bande C_{ij} avec $j \in [0, 3]$ sont notés C_i ;
- les ensembles de 16 coefficients petits-enfants de la même sous-bande sont notés G_i ;
- chaque ensemble G_i est partitionné en 4 sous-ensembles de 4 coefficients notés H_{ij} ;
- la concaténation des ensembles C_i et G_i forme l'ensemble des descendants noté D_i ;
- la concaténation de tous les ensembles de descendants D_0, D_1 et D_2 forme l'ensemble B ;

- **Les mots de transitions :**

A un plan de bits b , après le codage des coefficients parents, des mots de transition sont utilisés pour signaler les plages de zéros, c'est-à-dire les ensembles dans lesquels tous les coefficients sont non significatifs au plan de bits b . Ainsi, le premier mot de transition $\text{tran}B$ indique si tous les coefficients de l'ensemble B sont encore non significatifs au plan de bits b . Sinon, le mot de transition $\text{tran}D$ permet d'indiquer les ensembles D_i dans lesquels tous les coefficients sont encore non significatifs. Dans les ensembles D_i contenant des coefficients significatifs, les coefficients enfants des ensembles C_i sont codés et le mot de transition $\text{tran}G$ indique les ensembles G_i dans lesquels tous les coefficients petits-enfants sont encore non significatifs. Enfin, dans les ensembles G_i contenant des coefficients significatifs, le mot de transition $\text{tran}H_i$ indique les ensembles H_{ij} dans lesquels tous les coefficients sont encore non significatifs. Les petits-enfants des groupes H_{ij} significatifs sont ensuite codés. Ce processus de codage est détaillé par suite.

Pour limiter les effets de perte de données qui peuvent se produire sur le canal de communication les coefficients d'ondelettes d'une image sont divisés en segments.

Un segment est défini comme un groupe de S blocs consécutifs, chaque segment est codé indépendamment des autres ; le S peut être affecté à n'importe quelle valeur donc : $16 \leq S \leq 2^{20}$ sauf pour le dernier segment pour laquelle S peut être attribué à une valeur qui est : $1 \leq S \leq 2^{20}$ La valeur de S peut être choisie en fonction de la mémoire disponible pour stocker le segment. Voila un exemple de la division d'une image en segments figure (III.3) :

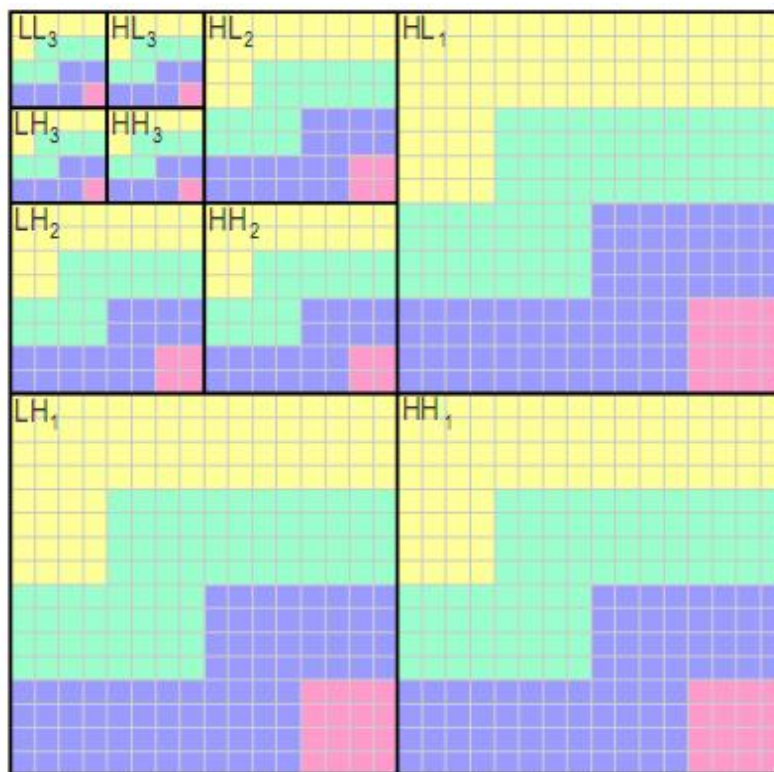


Figure (III.3) répartition des coefficients de la DWT en segments

Chaque ombre d'une couleur représente un segment avec la taille de chacun est $S = 5$ sauf pour le dernier segment qui a $S=1$.

N.B : notée que ce chiffre est purement à titre d'illustration, en faite la valeur minimale de S est 16 à part le dernier segment d'une image qui peut prendre une valeur moins de 16.

• **Algorithme de base**

Le plan de bits le plus significatif est déterminé par le coefficient d'ondelette de plus grande amplitude dans le bloc (mis à part le coefficient de la sous-bande LL3).

La figure (III.4) illustre la structure de codage d'un plan de bits :

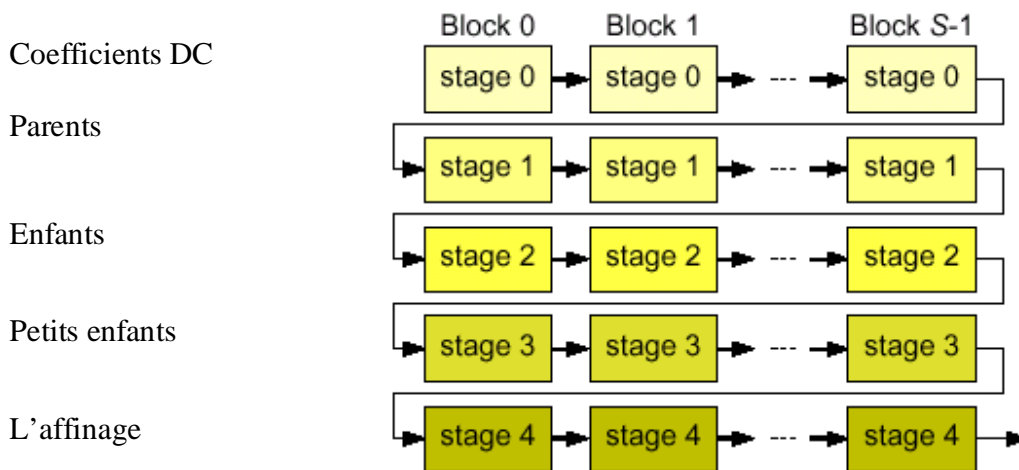


Figure (III.4) Structure de codage d'un plan de bits

1.Etape0: Codage des coefficients DC:

Le codeur BPE code la version quantifiée des coefficients DC dans le segment .cette première étape de codage de coefficients DC se fait séparément du codage des coefficients AC afin qu'un simple codage peut être utilisé pour exploiter la corrélation inter-blocs entre les coefficients DC quantifiées.

2.Etape1: Codage des parents:

on a deux mots à définir $types_b[P]$, $signes_b[P]$

Soit x coefficients d'ondelette $\in P$ tel que $P= \{P_0, P1, P2\}$

$$x \text{ significatif} \Leftrightarrow |x| \geq 2^b$$

– $\text{types}_b[\mathbf{P}]$ est le mot correspondant à l'amplitude des coefficients $x \in \mathbf{P}$ qui n'étaient pas significants aux plans de bits précédents:

$$\text{types}_b[\mathbf{P}] = \begin{cases} 1 & \text{pour les } |x| \geq 2^b \\ 0 & \text{pour les } |x| < 2^b \end{cases}$$

– $\text{signe}_b[\mathbf{P}]$ correspond aux bits de signe des coefficients $x \in \mathbf{P}$ qui sont devenus significants à ce plan de bits c'est-à-dire tels que $2^b \leq |x| < 2^{b+1}$

$$\text{signe}_b[\mathbf{P}] = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Nb : le signe est désigné pour les x significants.

3. Etape2: Codage des enfants:

on a quatre mots à définir tranB , tranD , $\text{types}_b[\mathbf{Ci}]$, $\text{signe}_b[\mathbf{Ci}]$

-**tranB** ne sera pas généré s'il a déjà pris la valeur 1 dans le plan précédent

$$\text{tranB} = \begin{cases} 1 & \text{s'il existe au moins un } D_i \text{ significatif} \\ 0 & \text{pas } D_i \text{ significatif} \end{cases}$$

-**tranD** n'est généré que si $\text{tranB} \neq 0$ (existe des descendants significants)

tranD un mot de transition servant à indiquer quels sont les ensembles de descendants D_i qui deviennent significants à ce plan de bits.

-Codage de la significiance des coefficients enfants dans les ensembles D_i trouvés significants

$$-\text{types}_b[\mathbf{Ci}] = \begin{cases} 1 & \text{pour les } |x| \geq 2^b \\ 0 & \text{pour les } |x| < 2^b \end{cases}$$

$$-\text{signe}_b[\mathbf{Ci}] = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

4. Etape3 : Codage des petits-enfants :

on a quatre mots à définir tranG , tranHi , $\text{types}_b[\mathbf{Hij}]$, $\text{signe}_b[\mathbf{Hij}]$

Si $\text{tranB}=0$, cette étape est vide (pas de petits enfants signifiants)

-tranG un mot de transition servant à indiquer quels sont les ensembles de petits enfants G_i qui deviennent signifiants à ce plan de bits.

-tranHi un mot de transition servant à indiquer quels sont les groupes de petits enfants H_{ij} qui deviennent signifiants à ce plan de bits.

-Codage de la signifiante des coefficients petits enfants dans les groupes H_{ij} trouvés signifiants

$$-\text{types}_b[\mathbf{H}_{ij}] = \begin{cases} 1 & \text{pour les } |x| \geq 2^b \\ 0 & \text{pour les } |x| < 2^b \end{cases}$$

$$-\text{signe}_b[\mathbf{H}_{ij}] = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Mots binaire	Taille max. (nombre de bits)	Valeurs impossibles
$\text{types}_b[\mathbf{P}]$	3	-
$\text{types}_b[\mathbf{C}_i]$	4	-
$\text{types}_b[\mathbf{H}_{ij}]$	4	0000
tranD	3	000
tranG	3	-
tranHi	4	0000

Table (III.3) les tailles maximales des mots binaire et valeurs impossibles :

5. Etape4: Raffinement:

Dans cette partie, l'amplitude des coefficients trouvés significatifs aux plans de bits précédents est affinée en mettant pour ces coefficients significatifs un bit « 0 » ou « 1 » non signé, « 1 » pour le coefficient le plus significatif et « 0 » pour les coefficients les moins significatifs.

Les mots binaires générés précédemment sont codés entropiquement à l'aide des tables de codage. Dans un premier temps, ils sont convertis en symboles en utilisant des tables différentes en fonction de la longueur du mot binaire et du type de mot.

mots	symboles
00	0
01	2
10	1
11	3

Table (III.4) Symboles pour les mots binaires de 2 bits

mots	Symboles	Symboles
	(types _b [P], types _b [Ci], types _b [Hi] tranG, tranHi)	(tranD)
000	1	-
001	4	3
010	0	0
011	5	4
100	2	1
101	6	5
110	3	2
111	7	6

Table (III.5) Symboles pour les mots binaires de 3 bits

mots	Symboles (types $b [C_i]$)	Symboles (types $b [H_{ij}]$, tranHi)
0000	10	-
0001	1	1
0010	3	3
0011	6	6
0100	2	2
0101	5	5
0110	9	9
0111	12	11
1000	0	0
1001	8	8
1010	7	7
1011	13	12
1100	4	4
1101	14	13
1110	11	10
1111	15	14

Table (III.6) Symboles pour les mots binaires de 4 bits

D'autres tables de codages ont ensuite utilisées pour convertir ces symboles en mots binaires de longueur variable qui sont inclus dans le train binaire.

-Tous les mots binaires de deux bits sont codés en utilisant l'un des codes binaires de longueurs variables donnés dans le tableau.

-Tous les mots binaires de trois bits sont codés en utilisant l'un des codes binaires de longueurs variables donnés dans le tableau.

- Tous les mots binaires de quatre bits sont codés en utilisant l'un des codes binaires de longueurs variables donnés dans le tableau.

symboles	Code 0	Code1
0	1	00
1	01	01
2	001	10
3	000	11

Table (III.7) Les codes de taille variable pour les mots binaires de 2bits

symboles	Code 0	Code 1	Code 2
0	1	10	000
1	01	11	001
2	001	010	010
3	0000	011	011
4	00001	0010	100
5	00010	0011	101
6	000110	0000	110
7	000111	0001	111

Table (III.8) Les codes de taille variable pour les mots binaires de 3 bits

symboles	Code 0	Code 1	Code 2	Code 3
0	1	10	100	0000
1	01	11	101	0001
2	001	010	110	0010
3	0001	011	111	0011
4	0000000	0010	0100	0100
5	0000001	0011	0101	0101
6	0000010	000000	0110	0110
7	0000011	000001	0111	0111
8	00001000	000010	00100	1000
9	00001001	000011	00101	1001
10	00001010	000100	00110	1010
11	00001011	000101	00111	1011
12	00001100	0001100	00000	1100
13	00001101	0001101	00001	1101
14	00001110	0001110	00010	1110
15	00001111	0001111	00011	1111

Table (III.9) Les codes de taille variable pour les mots binaires de 4 bits

En fin, on arrive à générer le train binaire suivant ces étapes:

- 1- Les bits des coefficients AC de chaque bloc sont inclus dans le flux de bits en sortie
- 2- Pour chaque bloc, la chaîne de bits de sortie est composée de bits des coefficients dans l'ordre suivant :
 - Pi pour chaque $i=0, 1, 2$
 - les C_i , pour chaque $i=0, 1, 2$
 - les H_{ij} , pour chaque $i=0, 1, 2$ et chaque $j=0, 1, 2, 3$
- 3- Les ensembles C_i , H_{ij} sont codés dans l'ordre indiqué dans la table (III.2)
- 4- les chaînes résultantes pour tous les blocs dans les segments sont concaténées pour produire une chaîne de sortie pour le segment.

N.B :

- Les mots binaire de 1 bit ne sont pas codés mais inclus directement dans le train binaire.
- Le mot tranB a toujours au plus 1 bit donc il n'est jamais codé entropiquement.
- les bits des signes ne sont jamais codés car les coefficients AC sont généralement négatifs et positifs avec une même probabilité

III.2.5. Le décodage et la transformée en ondelette inverse :

On applique le chemin inverse de codage en utilisant les tables (III.4), (III.5), (III.6), (III. 7), (III.8) et (III.9) pour avoir les mots binaires générés pour chaque plan de bit et puis le chemin inverse pour chacun de ces plan jusqu'à l'obtention des coefficients en ondelettes sous forme entière. Ensuite, on applique la transformée en ondelettes inverse, cette opération consiste en l'introduction des lignes et des colonnes de zéros entre les échantillons du l'image décodée (c'est le principe d'interpolation), enfin, on obtient l'image compressée.

III.4.Discussion :

Dans ce chapitre, on a développé la méthode de codage par plan de bits BPE du CCSDS qui se base sur la transformée en ondelettes suivi de la quantification scalaire à zone morte.

Une fois on a défini le schéma complet de notre méthode adoptée, place aux différents tests appliqués sur celui-ci, décelant ses performances. Ces tests et résultats émanant seront présentés dans le prochain chapitre.

CHAPITRE IV

Testes et résultats

IV.1 Préambule :

Dans le chapitre précédent, on a proposé une nouvelle méthode de compression d'images utilisant le codeur BPE. Dans le but d'évaluer notre algorithme de compression, des images tests seront utilisées et des critères seront pris en considération pour discuter les résultats obtenus.

IV.2. Présentation des données :

Afin d'interpréter les résultats obtenus, les deux critères d'évaluation des performances choisis sont :

- Le rapport signal sur bruit crête (*PSNR*) en décibels (dB). Ce dernier mesure la distorsion de l'image traitée par rapport à l'image originale, selon l'équation suivante :

$$PSNR(i) = 10 \log_{10} \left[\frac{(255)^2}{MSE} \right]$$

- Le taux de compression défini, par la formule ci-dessous :

$$Tc = \left(1 - \frac{1}{Rc} \right) * 100$$

L'algorithme de compression que nous avons développé, a été mis en œuvre en utilisant le langage de programmation MATLAB 2007b.

IV.3. Les images tests :

Pour valider la méthode que nous avons mise en œuvre et montrer son efficacité, nous l'avons appliqué aux images tests Lena et Goldhill. Celles-ci sont formées de 512x512 pixels codés sur 256 niveaux de gris. Elles représentent respectivement un portrait et un paysage, qui sont très utilisées en compression d'images. L'image Goldhill présente plus de détails par rapport à l'image Lena.



a) Image Lena



b) Image Goldhill

Figure (IV .1) Images Tests**IV .4. Tests et interprétation des résultats obtenus :**

Nous avons effectué des tests sur nos images références en faisant varier le taux de compression en bpp et en pourcentage (T_c) ainsi que le rapport de compression R_c pour obtenir des variations en PSNR ; tout en faisant varier le temps d'exécution (t_c) en fonction du taux en bpp pour les deux images. Les résultats obtenus sont répertoriés respectivement dans les tables (IV .1), (IV .2), (IV .3), (IV .4).

(bit par pixel)(bpp)	0.16	0.17	0.2	0.22	0.26	0.32	0.4	0.53	0.8	1.6
PSNR (dB) l	29.90	30.58	31.33	32.29	32.93	33.78	35.21	36.42	38.60	41.92
PSNR (dB) G	28.29	28.66	29.02	29.61	30.24	30.77	31.42	32.71	34.57	38.82

Table (IV.1) Résultats du PSNR en fonction du taux en bpp des deux images

L'évolution du PSNR (dB) en fonction du débit binaire (bpp) des deux images Lena et Goldhill est représenté par la figure suivante :

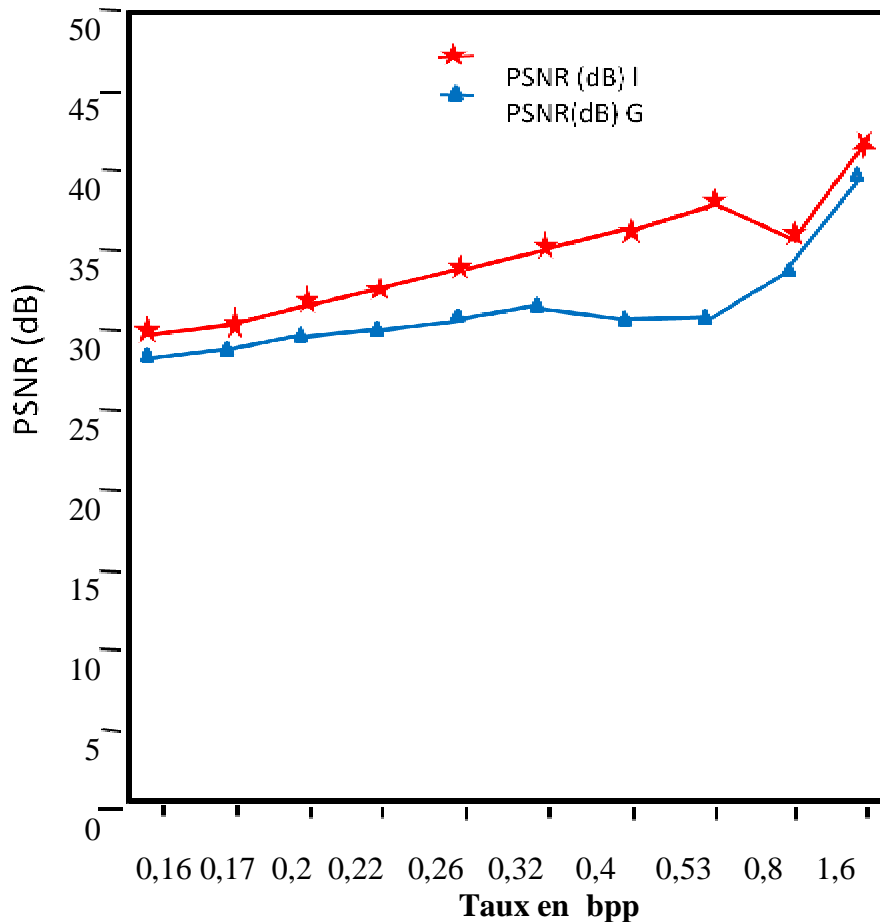


Figure (IV.2) Evolution du PSNR en fonction du taux en bpp pour les deux images

D'après la figure obtenue, on remarque que PSNR varie proportionnellement en fonction du débit binaire. En comparant les évolutions des PSNR en fonction du taux, on constate que l'image Lena atteint les meilleurs PSNR par rapport à l'image Goldhill pour les mêmes valeurs du débit binaire.

Rc	5	10	15	20	25	30	35	40	45	50
PSNR (dB)l	41.29	38.60	36.42	35.21	33.78	32.93	32.29	31.33	30.58	29.90
PSNR (dB)g	38.82	34.57	32.71	31.42	30.77	30.24	29.61	29.02	28.66	28.29

Table (IV.2) Résultats du PSNR des deux images en fonction de Rc

L'évolution du PSNR (dB) en fonction du rapport de compression (RC) des deux images Lena et Goldhill est représenté par la figure qui suit :

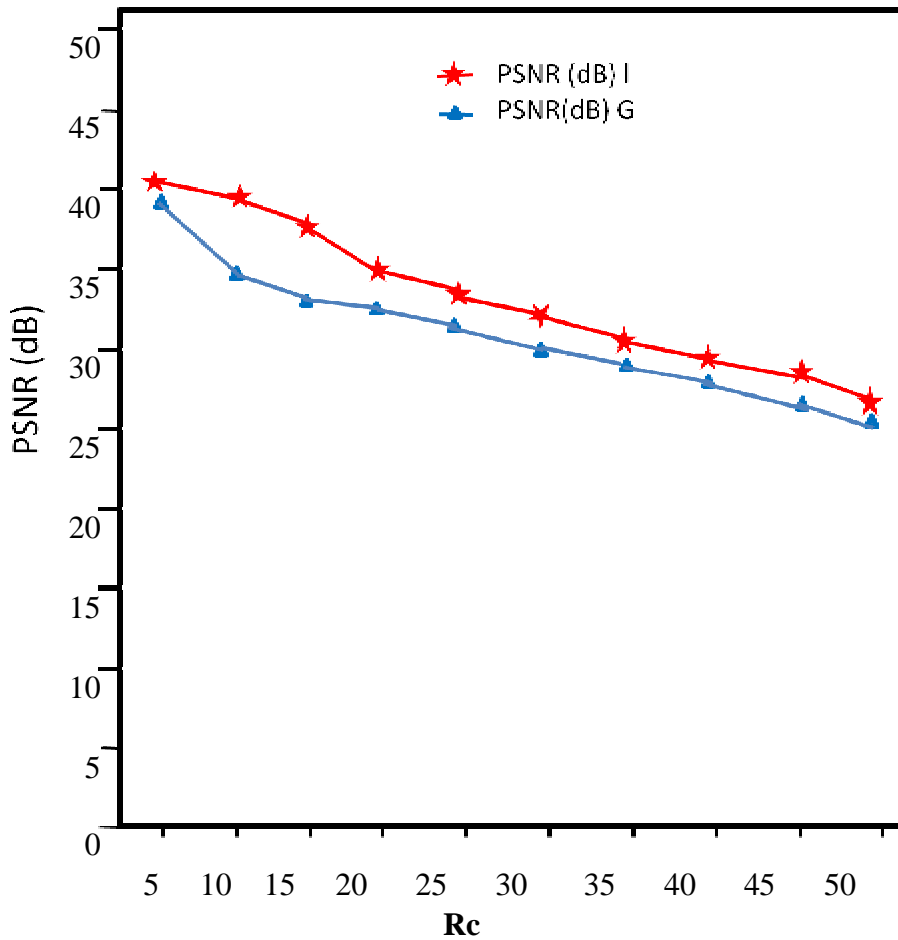


Figure (IV.3) Rapport signal sur bruit (PSNR) en fonction de Rc

La figure (IV.3) montre que le PSNR est inversement proportionnelle au rapport de compression (R_c), c'est à dire de plus en plus le rapport de compression augmente la qualité de l'image se dégrade, et en comparant entre les deux images, on remarque que l'image Lena atteint les meilleurs PSNR par rapport à l'image Goldhill pour les mêmes valeurs du rapport de compression.

Tc(%)	80	90	93.33	95	96	96.66	97.14	97.50	97.77	98
PSNR (dB)l	41.29	38.60	36.42	35.21	33.78	32.93	32.29	31.33	30.58	29.90
PSNR (dB)G	38.82	34.57	32.71	31.42	30.77	30.24	29.61	29.02	28.66	28.29

Table (IV.3) Résultats du PSNR des deux images en fonction de Tc

L'évolution du PSNR (dB) en fonction du taux de compression (Tc) des deux images Lena et Goldhill est représenté par la figure suivante

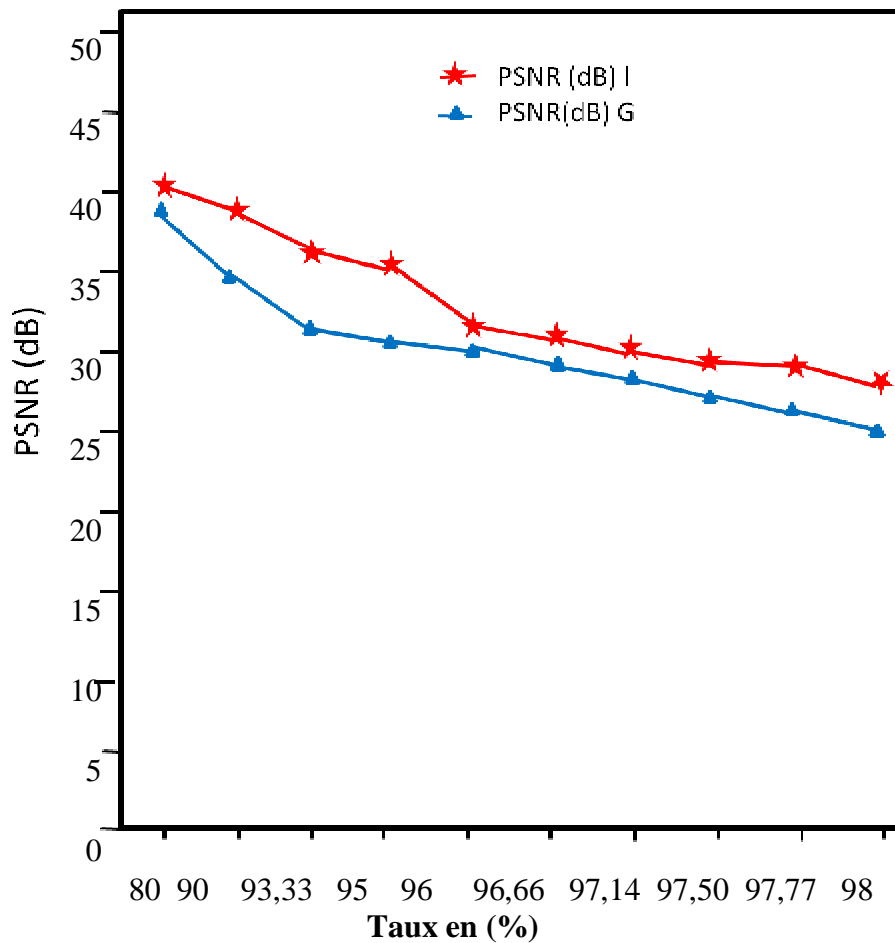


Figure (IV.4) Rapport signal sur bruit (PSNR) en fonction de Tc

La figure (IV.4) montre que le PSNR est inversement proportionnelle au taux de compression(Tc) ; la qualité de l’image se dégrade de plus en plus que la taux de compression croix. L’image Lena atteint des meilleurs résultats par rapport à l’image Goldhill pour les mêmes valeurs de taux compression.

(bit par pixel)(bpp)	0.16	0.17	0.2	0.22	0.26	0.32	0.4	0.53	0.8	1.6
tc(s)I	1.38	1.40	1.39	1.41	1.54	1.65	1.75	1.67	2.29	2.87
tc(s)G	1.45	1.48	1.5	1.46	1.69	1.65	1.64	1.64	1.93	3.50

Table (IV .4) Résultats des temps d’exécution (tc)en fonction du taux en bpp pour deux images

L'évolution du temps d'exécution en fonction du débit binaire (bpp) des deux images Lena et Goldhill est représenté par la figure suivante :

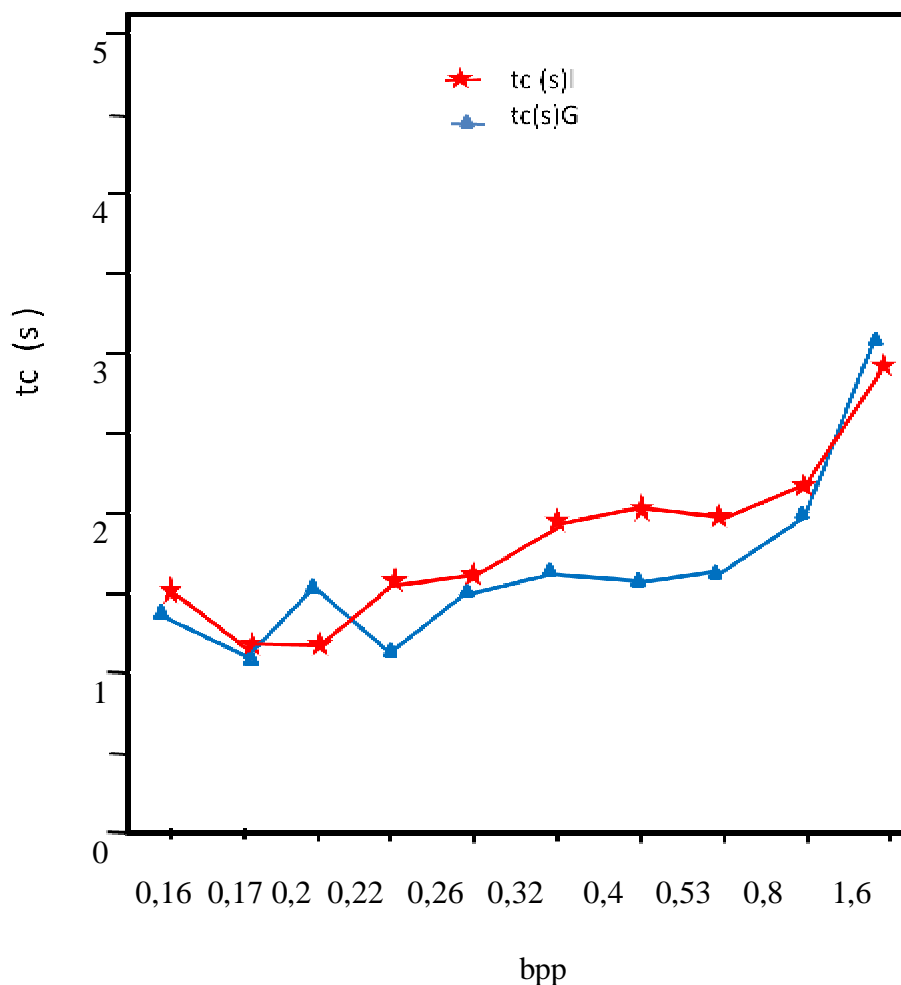


Figure (IV.5) Temps d'exécution (tc) en fonction du taux en bpp pour les deux images

De la figure (IV.5), on déduit que le temps d'exécution varie proportionnellement en fonction du débit binaire. En comparant entre les deux images, on constate que le temps d'exécution pour l'image Lena varie plus rapidement par rapport à l'image Goldhill.



a) Image Lena, $R_c=5$

PSNR=41.29(db)

b) Image Goldhil , $R_c=5$

PSNR=38.82 (db)

Figure (IV.6) Images reconstruites pour un taux = 80%



a) Image Lena, $R_c=20$

PSNR=35.21 (db)

b) Image Goldhil, $R_c=20$

PSNR= 31.42(db)

Figure (IV.7) Images reconstruites pour un taux = 95%

a) Image Lena, $R_c=35$

PSNR=32.29 (db)

b) Image Goldhill, $R_c=35$

PSNR= 29.61(db)

Figure (IV.8) Images reconstruites pour un taux = 97.14%a) Image Lena, $R_c=50$

PSNR= 29.90 (db)

b) Image Goldhill, $R_c=50$

PSNR= 28.29(db)

Figure (IV.9) Images reconstruites pour un taux = 98%

Après les tests effectués sur les différentes images références, les résultats obtenus montrent que l'image Lena atteint les meilleures performances par rapport à l'image Goldhill concernant la meilleure qualité visuelle (PSNR élevé), temps d'exécution plus petit avec des taux de compression importants. On constate qu'il existe une relation étroite entre les paramètres facteur de qualité et complexité de l'image d'une part et le rapport signal sur bruit et taux de compression d'autre part donc plus l'image moins complexe (contient moins de détails) plus elle atteint des meilleures performances à la reconstitution.

IV.5. Discussion :

L'algorithme de compression et de décompression des images, ainsi, développé est capable de compresser toute image avec de forts taux de compression et de la restituer avec une qualité appréciable. Ainsi, les tests effectués sur l'image Lena et Godhill prises pour références ont montré la supériorité de cet algorithme vis à vis d'autres méthodes de compression, telle que: celles basées sur les ondelettes. En effet, notre algorithme réalise un très bon compromis entre le taux de compression, la qualité de restitution d'une image et la limitation du nombre d'altérations. De telles performances ont pu être atteintes grâce à la combinaison de l'ondelette bi orthogonale 9/7 et le codeur BPE que nous avons adopté pour réaliser le processus de compression et de décompression des images, sachant que, le PSNR a atteint 41.92dB pour un débit de 1.6 bits/pixel dans le cas de l'image référence Lena. Concernant l'image Goldhill, notre algorithme de compression a pu atteindre, pour un débit de 1.6bits/pixel, un PSNR de 38.82 dB.

Visuellement, nous avons constaté que les images reconstruites sont d'un aspect visuel appréciable car pratiquement, tous les détails des images originales sont bien restitués. Par conséquent, notre algorithme a apporté de bien meilleurs résultats par rapport à d'autres algorithmes déjà utilisés.

CONCLUSION

Dans ce mémoire, nous avons exposé, le fait que la compression dite "sans perte" ne permettait pas une réduction significative du volume de ces données. Nous avons ensuite investigué la compression avec pertes maîtrisées, à savoir, des pertes n'affectant pas la qualité des images. Une nouvelle méthode basée sur une transformée en ondelettes, qui possède des propriétés intéressantes du point de vue de la compression a été élaborée et implémentée. En particulier, nous avons utilisé une transformation en ondelette biorthogonale 9/7 ainsi nous avons fixé le nombre de résolution à trois. Une fois la décomposition effectuée, nous nous sommes intéressés au codage des coefficients d'ondelettes.

Les tests ont montré la supériorité de cet algorithme vis à vis d'autres méthodes de compression. En effet, notre algorithme réalise un très bon compromis entre le taux de compression et la qualité de restitution d'une image

Des améliorations sont toujours possibles, en appliquant la transformée en bandelettes par blocs, qui est à l'origine des post-transformées, avec modification du critère de sélection des bases pour adapter la post-transformée à une compression progressive. Nous employons alors la post-transformée dans le codeur du CCSDS le tout ayant une faible complexité calculatoire

Nous tirons la conclusion que ce travail a été très bénéfique pour nous. Il nous a permis d'enrichir nos connaissances dans un domaine aussi vaste qui est la compression d'image en général.

ANNEXES

Quelques types d'ondelettes

1. Les ondelettes de Haar :

Le mérite revient à Alfred Haar d'avoir construit en 1909 des bases considérées aujourd'hui comme le fondement de la théorie des ondelettes. En effet, Haar a défini une fonction $h(x)$ telle que :

$$h(x) = \begin{cases} 1 & \text{pour } 0 \leq x \leq \frac{1}{2} \\ -1 & \text{pour } \frac{1}{2} \leq x \leq 1 \\ 0 & \text{ailleurs} \end{cases} \quad (\text{A.1})$$

Pour $n \geq 1$, il a construit une base orthonormée de $L^2 [0 ; 1]$ avec des fonctions définies par l'équation suivante :

$$h_n(x) = 2^{\frac{j}{2}}(2^j x - K) \quad (\text{A.2})$$

Avec :

$$n = 2^j x + K$$

$$j \geq 0 \quad \text{et} \quad 0 \leq K \leq 2^j$$

Dans l'équation ci-dessus, $L^2 [0 ; 1]$ est l'espace des fonctions de carré intégrable sur l'intervalle $[0 ; 1]$. Le support de $h_n(x)$ est l'intervalle dyadique défini par l'équation:

$$I_n = [K2^{-j}; (K + 1)2^{-j}] \subset [0; 1[\quad (\text{A.3})$$

$h_n(x)$ peut aussi s'écrire selon l'équation :

$$h_n(x) = 2^{\frac{j}{2}} h(2^j x - K) = \frac{1}{\sqrt{2^{-j}}} h\left(\frac{x - K2^{-j}}{2^{-j}}\right) \quad (\text{A.4})$$

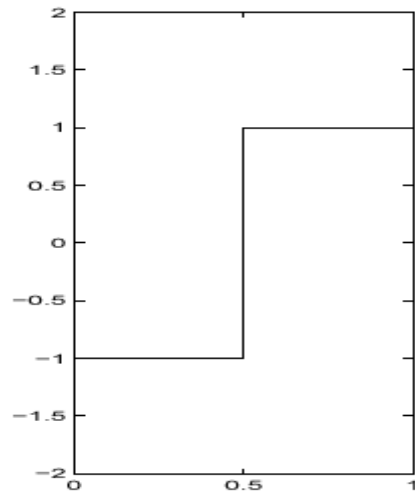


Fig1: Ondelette de Haar

2. Ondelettes de Daubechies :

Ingrid Daubechies (1990) a complété l'œuvre de Haar en 1987. Pour toute valeur r , Daubechies construit une base orthonormée de $L^2(\mathbb{R})$ de la forme suivante :

$$2^{\frac{j}{2}} \Psi_r(2^j x - K) \quad (\text{A.5})$$

Où :

$$j \in \mathbb{Z} \text{ et } K \in \mathbb{Z}$$

Elle vérifie les propriétés énoncées plus haut. En effet, Ψ est définie sur un support compact $[0 ; 2r + 1]$ et satisfait l'équation ci-dessous:

$$\int_{-\infty}^{+\infty} \Psi_r(x) dx = \int_{-\infty}^{+\infty} x^r \Psi_r(x) dx \quad (\text{A.6})$$

Où :

$\Psi_r(x)$ possède r dérivées continues.

Lorsque $r = 0$, $\Psi_r(x)$ est définie sur $[0 ; 1]$, on retombe sur le système de Haar (Daubechie1 « db1 »).

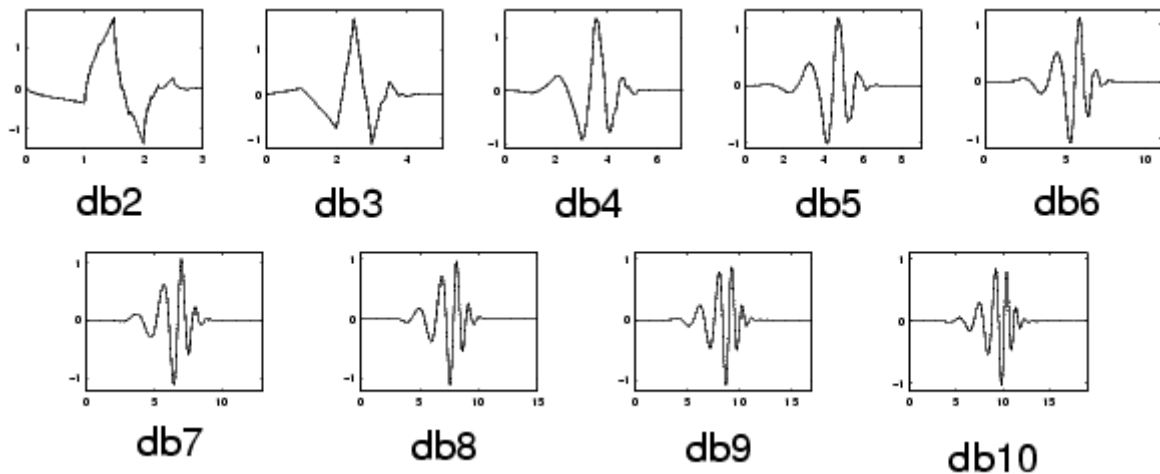


Fig.2 : Exemples d'ondelettes de Daubechies

NB : Les ondelettes de Haar et de Daubechies font partie du groupe des ondelettes appelées ondelettes orthogonales. Ces ondelettes sont des fonctions de base vérifiant les conditions présentées par l'équation :

$$\left\{ \begin{array}{l} \langle \phi_k^j \setminus \Psi_l^j \rangle = \delta_{k,l} \\ \langle \Psi_k^j \setminus \Psi_l^j \rangle = \delta_{k,l} \end{array} \right. \quad \text{Pour tout } j, k, l \quad (\text{A.7})$$

Où :

$\delta_{k,l}$: est défini par l'équation :

$$\delta_{k,l} = \begin{cases} 1 & \text{pour } l = k \\ 0 & \text{ailleurs} \end{cases}$$

$\phi(x)$ et $\Psi(x)$ sont des fonctions permettant de définir respectivement les coefficients des filtres d'analyse passe-bas et passe-haut de l'ondelette.

3. Ondelettes bi-orthogonales :

Pour définir les ondelettes bi-orthogonales, il est nécessaire d'introduire les fonctions duales $\tilde{\Psi}(x)$ et $\tilde{\phi}(x)$, de $\phi(x)$ et $\Psi(x)$, respectivement. Les conditions présentées par l'équation ci-dessous sont alors vérifiées :

$$\left\{ \begin{array}{l} \langle \phi_k^j \setminus \tilde{\phi}_l^j \rangle = \delta_{k,l} \\ \langle \Psi_k^j \setminus \tilde{\Psi}_l^j \rangle = \delta_{k,l} \end{array} \right. \quad \text{Avec} \quad \delta_{k,l} = \begin{cases} 1 \text{ pour } l = k \\ 0 \text{ ailleurs} \end{cases} \quad (\text{A.8})$$

Une ondelette bi-orthogonale est telle que la fonction d'échelle originale est orthogonale à l'ondelette duale et l'ondelette originale est orthogonale à la fonction d'échelle duale. Autrement dit, les ondelettes bi-orthogonales vérifient les conditions présentées par l'équation :

$$\left\{ \begin{array}{l} \langle \phi_k^j \setminus \tilde{\phi}_l^j \rangle = 0 \\ \langle \Psi_k^j \setminus \tilde{\Psi}_l^j \rangle = 0 \end{array} \right. \quad \text{Pour tout } j, k, l \quad (\text{A.9})$$

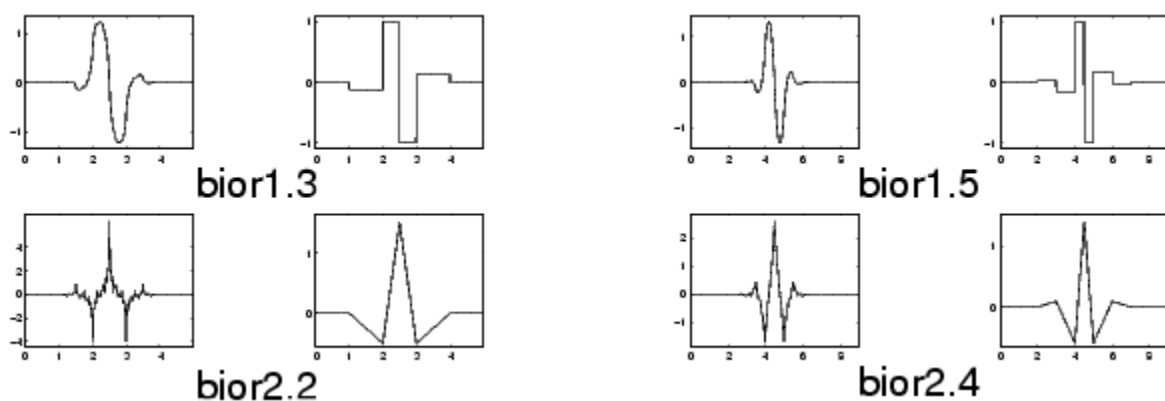


Fig.3 : Exemples d'ondelettes bi-orthogonales

Exemple de codage avec le BPE

Dans cet exemple on s'intéresse au codage des coefficients d'ondelettes dits <AC>, c'est-à-dire les coefficients en dehors de la sous-bande LL_3 . Le codage de l'imagette basse fréquence <DC> n'est pas pris en compte.

Le BPE utilise une structure d'arbre pour coder les plans de bits. Trois familles sont définies par bloc et numérotées par l'indice $i \in \{0, 1, 2\}$. Il y a trois générations; chacune correspondant à l'un des trois niveaux de la décomposition en ondelettes :

1. Les *parents* correspondent aux coefficients du 3^{em} niveau de décomposition en ondelettes. Il y a 3 coefficients parents par bloc et on note le groupe $P = \{P_0, P_1, P_2\}$.

Dans l'exemple $P = \{-34, -31, 23\}$.

2. Les enfants (children) correspondent aux coefficients du 2^{em} niveau de décomposition en ondelettes. Il y a 3 groupes de 4 coefficients enfants par bloc notés C_i avec $i \in \{0, 1, 2\}$.

3. Les petits-enfants (grandchildren) correspondent aux coefficients du 1^{er} niveau de décomposition en ondelettes. Il y a 3 ensembles de 16 coefficients petits-enfants par bloc notés G_i avec $i \in \{0, 1, 2\}$. Chaque ensemble G_i est divisé en 4 groupes de 4 coefficients d'ondelettes notés H_{ij} avec $j \in \{0, 1, 2, 4\}$.

On définit aussi la liste des descendants d'une famille i notée $D_i = \{C_i, G_i\}$ et la liste de tous les descendants dans un bloc notée $B = \{D_0, D_1, D_2\}$.

L'exemple de coefficients d'ondelettes est représenté sur la figure (III.1). Ceci constitue un bloc de 64 coefficients tel que défini dans le standard du CCSDS. Le coefficient DC est représenté dans la case grisée. Les autres sont des coefficients AC. Le plus fort coefficient AC en valeur absolue vaut 49. Il y a donc 5 plans de bits à coder.

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

Figure (III.1) Exemple de coefficients d'ondelettes utilisés

A/- Codage du premier plan de bits : b=5

Le plan de bits $b = 5$ est représenté sur la figure (III.1)

Étape1: Codage des parents

Deux mots de codes ont générés à cette étape: $\text{types}_b[\mathbf{P}]$ et $\text{sign}_b[\mathbf{P}]$.

– $\text{types}_b[\mathbf{P}]$ est le mot correspondant à l'amplitude des coefficients $x \in \mathbf{P}$ qui n'étaient pas significants aux plans de bits précédents: $\text{tb}(x)=0$ si $|x| < 2^b$ ou $\text{tb}(x)=1$ si $2^b \leq |x| < 2^{b+1}$. Comme $b=5$ est le premier plan de bits codé, aucun coefficient n'est encore significatif. On a $\mathbf{P} = \{-34, -31, 23\}$ et donc $\text{types}_5[\mathbf{P}] = 1, 0, 0$.

– $\text{sign}_b[\mathbf{P}]$ correspond aux bits de signe des coefficients $x \in \mathbf{P}$ qui sont devenus significatifs à ce plan de bits c'est-à-dire tels que $2^b \leq |x| < 2^{b+1}$ ($\text{tb}(x)=1$). Ici, seul le coefficient $p_0 = -34$ est devenu significatif. Il est négatif. On a donc $\text{sign}_5[\mathbf{P}] = 1$.

Étape2: Codage des enfants

Plusieurs mots de code peuvent être générés à cette étape : tranB , tranD , et les $\text{types}_b[\mathbf{C}_i]$ et $\text{sign}_b[\mathbf{C}_i]$.

– **tranB** prend la valeur 0 si aucun descendant n'est significatif dans le bloc au plan de bits b codé. Il prend la valeur 1 si au moins un descendant du bloc est significatif : $\exists x \in b, 2^b \leq |x| < 2^{b+1}$ ($t_b(x)=1$)

Dans l'exemple, deux descendants sont significatifs au plan de bits $b = 5$. On a donc $\text{tranB} = 1$.

– **tranD** dans l'exemple, comme $b = 5$ est le premier plan de bits codé, aucun ensemble de descendants D_i a déjà été trouvé significatif. tranD sera donc un mot de 3 bits. De plus, comme $\text{tranB} = 1$, le mot tranD est généré. Au plan de bits $b = 5$, deux descendants sont significatifs. Ils font partie des ensembles de descendants D_0 et D_1 . On a donc $\text{tranD} = 1, 1, 0$.

– **types_b[Ci]** et **signs_b[Ci]** sont les mots correspondant respectivement à l'amplitude des enfants $x \in C_i$ qui n'étaient pas significatifs aux plans de bits précédents et aux signes des coefficients $x \in C_i$ qui sont devenus significatifs à ce plan de bits. Le mot tranD a déjà indiqué qu'il existe des descendants significatifs uniquement dans les familles $i = 0$ et $i = 1$. Seuls les groupes d'enfants C_0 et C_1 sont donc codés.

On a $\text{types}_5[C_0] = 1, 0, 0, 0$, $\text{signs}_5[C_0] = 0$ ($C_0[0] = 49$) et $\text{types}_5[C_1] = 0, 0, 0, 0$.

Puisque aucun coefficient du groupe C_1 n'est significatif, $\text{signs}_5[C_1]$ est vide.

Étape 3 : Codage des petits-enfants

Si le mot $\text{tranB} = 0$, cette étape est vide puisque cela signifie qu'aucun descendant, petits-enfants compris, n'est significatif à ce plan de bit. Plusieurs mots de code peuvent être générés à cette étape : tranG , tranHi , et les $\text{types}_b[\text{Hij}]$ et $\text{signs}_b[\text{Hij}]$.

– **tranG** indique dans quelle famille on peut trouver des petits-enfants significatifs. Au plan de bits $b = 5$, le mot tranD a indiqué que des descendants des familles $i = 0$ et $i = 1$ sont significatifs. Seuls les petits-enfants de ces deux familles sont considérés. Seul le petit-enfant $H_{11}[1] = 47$ est significatif. Il fait partie de la famille $i = 1$. On a donc $\text{tranG} = 0, 1$.

– **tranHi** est aussi un mot de transition servant à indiquer quels sont les groupes de petits-enfants H_{ij} de la famille i qui deviennent significatifs à ce plan de bits. tranG indique déjà qu'il existe des

Petits-enfants significatifs uniquement dans la famille $i = 1$. Seul le mot tranH_1 est donc généré. Seul le petit-enfant $H_{11}[1] = 47$ est significatif. On a donc $\text{tranH}_1 = 0, 1, 0, 0$.

– **types_b[Hij]** et **signs_b[Hij]** sont les mots correspondant respectivement à l'amplitude des petits-enfants dans H_{ij} qui n'étaient pas significatifs aux plans de bit précédents et aux signes des coefficients $x \in H_{ij}$ qui sont devenus significatifs à ce plan de bits.

Le mot tranH₁ a déjà indiqué qu'il n'existe des petits-enfants signifiants que dans le groupe H₁₁. Seul le groupe de petits-enfants H₁₁ est donc codé. On a types₅[H₁₁] = 0, 1, 0, 0 et signs₅[H₁₁] = 0 (H₁₁[1] = 47)

	-1	+1
.
.
.
.	.	.	+1	.	.	.
.
.
.

Figure (III .2) Plan de bits b=5

Les coefficients devenant signifiants à ce plan de bits sont marqués par un < 1 > précédé du signe du coefficient. Les coefficients non signifiants sont marqués par un < . >.

Étapes	Mots binaires	Commentaires
Parents	$\text{types}_5[\text{P}] = 1, 0, 0$ $\text{signs}_5[\text{P}] = 1$	$P_0 = -34$ devient significatif au plan de bits $b = 5$ p_0 négatif..
Enfants	$\text{tranB} = 1$ $\text{tranD} = 1, 1, 0$ $\text{types}_5[\text{C}_0] = 1, 0, 0, 0$ $\text{signs}_5[\text{C}_0] = 0$ $\text{types}_5[\text{C}_1] = 0, 0, 0, 0$	Première fois qu'un descendant est significatif. Descendants significatifs dans D_0 et D_1 . Pas de codage des coefficients de D_2 . $C_0[0] = 49$ devient significatif au plan de bits $b = 5$. $C_0[0]$ est positif. Pas de coefficient significatif dans C_1 .
Petits-enfants	$\text{tranG} = 0, 1$ $\text{tranH1} = 0, 1, 0, 0$ $\text{types}_5[\text{H11}] = 0, 1, 0, 0$ $\text{signs}_5[\text{H11}] = 0$	Petits-enfants significatifs dans la liste G_1 . Pas de codage des petits-enfants de G_0 . Petits-enfants significatifs dans le groupe H_{11} . $H_{11}[1] = 47$ devient significatif au plan de bits $b = 5$. $H_{11}[1]$ est positif.

Table (III.1) Codage du plan de bits $b = 5$.

B/-Codage du deuxième plan de bits : $b=4$

Le plan de bits $b = 4$ est représenté sur la figure (III.3)

Étape 1 : Codage des parents

Deux mots de code sont générés à cette étape : $\text{types}_b[\text{P}]$ et $\text{sign}_b[\text{P}]$.

– $\text{types}_4[\text{P}] = 1, 1$. Comme p_0 était déjà significatif au plan de bits précédent, il n'est pas codé ici.

– $\text{signs}_4[\text{P}] = 1, 0$ car p_1 est négatif ($p_1 = -31$) et p_2 est positif ($p_2 = 23$).

Étape 2 : Codage des enfants

- tranB n'est pas généré parce qu'il vaut déjà 1 au plan de bits précédent. On sait donc déjà qu'il y a des coefficients significatifs parmi les descendants.
 - tranD ne prend plus qu'un bit. On sait déjà qu'il y a eu des descendants significatifs au plan de bits précédent dans les ensembles D_0 et D_1 (tranD au plan de bits $b = 5$). Au plan de bits $b = 4$, il n'y a pas de descendant significatif dans l'ensemble D_2 . On a donc $\text{tranD} = 0$.
 - Les mots $\text{types}_4[C_0]$ et $\text{types}_4[C_1]$ sont générés. En effet, les mots tranD au plan de bits $b = 5$ et $b = 4$ ont indiqué qu'il n'existe des descendants significatifs que dans les familles $i = 0$ et $i = 1$. Comme le coefficient $C_0[0] = 49$ était déjà significatif au plan de bits précédent $t_4(C_0[0]) = 2$, le mot $\text{types}_4[C_0]$ est un mot de trois bits.
- On a donc $\text{types}_4[C_0] = 0, 0, 0$ et $\text{types}_4[C_1] = 0, 0, 0, 0$. Aucun coefficient enfant n'est nouvellement significatif au plan de bits $b = 4$. Les mots $\text{signs}_4[C_i]$ ne sont donc pas générés.

Étape 3 : Codage des petits-enfants

- tranG est maintenant un mot de 1 bit puisque, d'après le mot tranD, il ne peut pas y avoir de petits-enfants significatifs dans la famille $i = 2$ et on sait déjà qu'il y a des petits-enfants significatifs dans la famille $i = 1$. Ici, il n'y a pas de petit-enfant significatif dans la famille $i = 0$ au plan de bits $b = 4$. On a donc $\text{tranG} = 0$.
- Comme le mot tranG au plan de bits $b = 5$ a indiqué au moins un coefficient significatif parmi les petits-enfants de la famille $i = 1$, le mot tranH_1 précise leur localisation. De plus, au plan de bits $b = 5$, le mot tranH₁ a déjà indiqué des coefficients significatifs dans le groupe 1. Il n'est donc pas nécessaire de signaler ce groupe. Le mot tranH_1 est donc de trois bits au plan de bits $b = 3$: $\text{tranH}_1 = 0, 0, 0$.
- Seul le groupe H_{11} est codé puisque selon le mot tranH_1 c'est le seul groupe qui peut posséder des coefficients significatifs. Comme le coefficient $H_{11}[1] = 47$ était déjà significatif au plan de bits précédent, le mot $\text{types}_4[H_{11}]$ est un mot de trois bits codant la signification des autres coefficients du groupe. On a donc $\text{types}_4[H_{11}] = 0, 0, 0$.

	0	1
-1	+1
.
.
.	.	.	0	.	.	.
.
.
.

Figure (III .3) Plan de bits b=4

Les coefficients devenant significants au plan de bit $b = 4$ sont marqués par un $\langle 1 \rangle$ précédé du signe du coefficient. Les coefficients non significants sont marqués par un $\langle \cdot \rangle$. Les coefficients significants aux plans de bits précédents sont marqués par un bit $\langle 0 \rangle$ ou $\langle 1 \rangle$ non signé.

Étapes	Mots binaires	Commentaires
Parents	types ₄ [P] = 1, 1 signs ₄ [P] = 1, 0	p1 = - 31 et p2 = 23 deviennent significants. p0 déjà significatif au plan de bits précédent. p1 négatif et p2 positif.
enfants	tranD = 0 types ₄ [C0] = 0, 0, 0 types ₄ [C1] = 0, 0, 0, 0	Pas de descendant significatif dans la liste D2. C0[0] déjà significatif au plan de bits précédent. Pas de coefficient significatif dans C1.
Petits-enfants	tranG = 0 tranH1 = 0, 0, 0 types ₄ [H11] = 0, 0, 0	Pas de petit-enfant significatif t significatif dans G0. Pas de petit-enfant significatif dans les groupes autre que H11. H11[1] déjà significatif au plan de bits précédent.
Affinage	0, 1, 0	Bits des coefficients P0, C0[0] et H11[1] au plan de bits b = 4.

Table (III.2) Codage du plan de bits b = 4.

Étape 4 : Affinage de l'amplitude des coefficients déjà significants :

Dans cette partie, l'amplitude des coefficients trouvés significants aux plans de bits précédents est affinée. Les bits de ces coefficients au plan de bits b = 4 sont émis en commençant par les parents, puis les enfants et enfin les petits-enfants et en suivant la numérotation des familles et des groupes. Trois coefficients étaient significants avant le plan de bits b = 4 : p0, C₀[0] et H₁₁[1]. Le train binaire généré à l'étape 4 est donc 0, 1, 0.

C/-Codage du troisième plan de bits : $b=3$

Le plan de bits $b = 3$ est représenté sur la figure (III.4)

Étape 1 : Codage des parents

Tous les parents ont déjà été trouvés significatifs aux plans de bits précédents. Leur amplitude sera affinée à l'étape 4.

Étape 2 : Codage des enfants

- Il y a des descendants significatifs dans l'ensemble D_2 . On a donc $\text{tran}D = 1$.
- Tous les $\text{types}_3[C_i]$, $i \in \{0, 1, 2\}$ sont générés. En effet, les mots binaires $\text{tran}D$ aux plans de bits $b = 5$, $b = 4$ et $b = 3$ ont indiqué qu'il existe des descendants significatifs dans trois familles $i \in \{0, 1, 2\}$. On a donc $\text{types}_3[C_0] = 1, 1, 1$ et $\text{sign}_3[C_0] = 0, 0, 1$.
 $\text{types}_3[C_1] = 1, 1, 1, 0$ et $\text{sign}_3[C_1] = 0, 0, 1$. $\text{types}_3[C_2] = 0, 1, 1, 1$ et $\text{sign}_3[C_2] = 1, 1, 0$.

Étape 3 : Codage des petits-enfants

- $\text{tran}G$ est maintenant un mot de deux bits puisqu'on sait déjà qu'il y a des petits-enfants significatifs dans la famille $i = 1$. Ici, il y a des petits-enfants significatifs dans la famille $i = 0$ et aucun dans la famille $i = 2$ au plan de bits $b = 3$. On a donc $\text{tran}G = 1, 0$.
- $\text{tran}H_0 = 1, 1, 0, 1$ précise la localisation des petits-enfants significatifs dans l'ensemble G_0 .
 $\text{tran}H_1 = 1, 1, 0$ est un mot de trois bits puisqu'on savait déjà qu'il y a au moins un coefficient significatif dans le groupe H_{11} .
- Les mots $\text{types}_3[H_{ij}]$ et $\text{sign}_3[H_{ij}]$ générés sont donnés dans Table (III.3)

	0	0 +1	. +1 -1 .
1	0	+1 -1
+1 +1		. -1	. . . +1
-1 .		-1 +1
. +1 . 1		
.
.
. +1

Figure (III.4) Plan de bits $b=3$

Les coefficients devenant significants au plan de bits $b = 3$ sont marqués par un $\langle 1 \rangle$ précédé du signe du coefficient. Les coefficients non significants sont marqués par un $\langle \cdot \rangle$. Coefficients significants aux plans de bits précédents sont marqués par un bit $\langle 0 \rangle$ ou $\langle 1 \rangle$ non signé.

Étapes	Mots binaires	Commentaires
Parents		
Enfants	$\text{tranD} = 1$ $\text{types3}[C0] = 1, 1, 1$ $\text{signs3}[C0] = 0, 0, 1$ $\text{types3}[C1] = 1, 1, 1, 0$ $\text{signs3}[C1] = 0, 0, 1$ $\text{types3}[C2] = 0, 1, 1, 1$ $\text{signs3}[C2] = 1, 1, 0$	<p>Descendant significants dans la liste D2. C0[0] déjà significatif au plan de bits précédent</p> <p>Trois nouveaux coefficients significants dans C1.</p> <p>Trois nouveaux coefficients significants dans C2.</p>
Petits-enfants	$\text{tranG} = 1, 0$ $\text{tranH0} = 1, 1, 0, 1$ $\text{tranH1} = 1, 1, 0$ $\text{types3}[H00] = 0, 1, 0, 0$ $\text{signs3}[H00] = 0$ $\text{types3}[H01] = 1, 0, 0, 0$ $\text{signs3}[H01] = 1$ $\text{types3}[H03] = 0, 1, 0, 0$ $\text{signs3}[H03] = 0$ $\text{types3}[H10] = 0, 1, 0, 0$ $\text{signs3}[H10] = 0$ $\text{types3}[H11] = 0, 0, 0$ $\text{types3}[H12] = 0, 0, 0, 1$ $\text{signs3}[H12] = 0$	<p>Petit-enfants significants dans G0 Pas de petit-enfant significatif dans G2.</p> <p>Petit-enfants significants dans les groupes H00, H01 et H03.</p> <p>Petit-enfants significatif dans les groupes H10, H12.</p>
Affinage	0, 1, 0, 0, 1	

Table (III.3) Codage du plan de bits $b = 3$.

Étape 4 : Raffinement de l'amplitude des coefficients déjà significants

Cinq coefficients étaient significants avant le plan de bits $b = 3$: P_0 , P_1 , P_2 , $C_0[0]$ et $H_{11}[1]$. Le train binaire généré à l'étape 4 est donc 0, 1, 0, 0, 1.

On suit ses étapes jusqu'au dernier plan de bit ($b=0$) et finalement on applique le codage entropique sur les les mots binaires générés à l'aide de tables de codage citées précédemment.

- **Redondance identifiée :**

Une redondance peut être identifiée dans cet algorithme. Lors du codage du plan de bits $b = 5$, le mot *tranD* indique qu'il existe des descendants significants dans la famille $i = 1$. Lors du codage de la signifiante des enfants, on s'aperçoit qu'aucun enfant de l'ensemble C_1 n'est significatif. On en déduit qu'il existe forcément au moins un petit-enfant significatif dans la famille $i = 1$. Pourtant, cette information est retransmise dans le mot *tranG*. Heureusement, la probabilité de cette configuration des coefficients d'ondelettes est faible. Peu de bits sont donc redondants. De plus, l'exploitation de cette redondance complique l'implantation de l'algorithme.

Notations

- DWT : discrete wavelet transform (transformée en ondelette discrète)
- DWT 9/7: ondelette discrète à deux filtres : 9 (passe bas) et 7(passe haut).
- BPE : bit plane encoder (codage par plan de bit en anglais)
- AC : coefficients de l'image détail
- DC : coefficients de l'image d'approximation
- LLn : sous bande obtenue après filtrage des lignes et colonnes d'une image par le filtre passe bas.
- LHn : sous bande obtenue après filtrage des lignes et colonnes d'une image par les filtres passe bas et passe haut respectivement.
- HLn : sous bande obtenue après filtrage des lignes et colonnes d'une image par les filtres passe haut et passe bas respectivement.
- HHn : sous bande obtenue après filtrage des lignes et colonnes d'une image par le filtre passe haut.



Bibliographie

[1] : S.BELAMIRI, T.SAIDI, R.MEDDANE « Compression d'image par paquet d'ondelette » mémoire d'ingénieur en électronique, UMMTO, 2003.

[2] : E.INCERTI « compression d'image : algorithmes et standards » IA 101, Vol 02, paris 2003

[3] : M.BENBAHLOUL et T.MOKHTAR « compression d'image fixes selon JPEG 2000 » mémoire d'ingénieur en électronique, UMMTO, 2003

[4] : L. AKROUR « compression d'images numériques par fractales dans le domaine DCT » mémoire d'ingénieur d'état en électronique UMMTO, 2003

[5] A.LIORET, G.SAVART, R.ROY, Guide d'utilisation de programmes MATLAB en analyse numérique. Département mathématiques et Génie industriel école polytechnique de Montréal. 10 janvier 2002.

[6] : O.LE CADET « Méthodes d'ondelettes pour la segmentation d'images. Applications a l'imagerie médicale et au tatouage d'images » .thèse doctorat en math, Institut National Polytechnique de Grenoble, 28 Sept 2004

[7]ATEF MASMOUDI MOHAMED SALIM BOUHLEL, Un nouvel algorithme de compression exploitant le codage arithmétique en lui introduisant de nouveaux paramètres de codage, Institut Supérieure de Biotechnologie de Sfax (ISBS)-TUNIS

[8] :N.HAMRENE, D.IDIR, L.HAMOUDI « codages d'images en sous bande par fractales appliqués aux images médicales » thèse d'ingénieur en électronique, UMMTO, 2005

[9] :N.NAIT AMARA ET R.CHOUCHANE « compression d'images fixes par classification de régions en associant les ondelettes et les fractales » thèse d'ingénieur en électronique, UMMTO, 2008

[10] : O. MESSAOUI, « Etude comparative des méthodes de compression d'image : application aux images MSG » Thèse magister, Télédétection, L'UMMTO, 2008.

[11] :M.IDIR « compression irréversible d'image numérique a base d'ondelettes entières » mémoire d'ingénieur en électronique, UMMTO, 2007

[12] : M.LAHDIR « Nouvelle approche de compression d'images basée sur les ondelettes et les fractales : application aux images méteosat» thèse de doctorat, UMMTO, 2007

[13] BAARIR, Z.E.; OUAFI, A. Etude de la transformée en ondelettes dans la compression d'images fixes, Courrier du Savoir, 2004.

[14]LANGLET, R. Transformée en ondelettes : Application aux propriétés diélectriques et mécanique de nanostructures carbonées, Thèse de doctorat, Spécialité Physique, Université de Franche Comté, (2004)

[15] AKROUR, L. Compression d'images par SPIHT appliqué dans le domaine des ondelettes entières: Application aux images MSG .Thèse magister, Télédétection, L'UMMTO, 2010.

[16] A,CZIHO. Quantification vectorielle et compression d'image, application à l'imagerie médicale. Thèse de doctorat, école doctorale en informatique, traitement du signal et télécommunications, université de Rennes 1, France, (1999)

[17] LAHDIR, M., AMEUR, S. ET AKROUR, L .Compression d'images numériques par fractal dans le domaine DCT. Internationale Conférence: Sciences of Electronic,

Technologies of Information and Télécommunication SETIT2005, 27-31 Mars 2005, Sousse, Tunisie.

[18] X. DELAUNAY. Compression d'images satellite par post-transformées dans le domaine ondelettes .Doctorat de l'université de Toulouse, novembre 2008.

[19] P.-S. YEH, P. ARMBRUSTER, A. KIELY, B. MASSCHELEIN, G. MOURY, C. SCHAEFER ET C. THIE-BAUT. The new CCSDS (the Consultative Committee for Space Data System Standards image compression recommendation). Aerospace Conference, 5(12), p. 4138–4145, 2005.

[20] CCSDS the Consultative Committee for Space Data System Standards. Image Data Compression Informational Report CCSDS 120.1-G-1 Green Book, juin 2007.

[21] CCSDS the Consultative Committee for Space Data System Standards. Image Data Compression Recommended Standard - CCSDS 122.0-B-1 Blue Book, novembre 2005.