



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



UNIVERSITE ECHAHID HAMMA LAKHDAR – EL OUED
FACULTE DES TECHNOLOGIE
Département Gini électrique

Mémoire de fin d'études présenté
Pour l'obtention du diplôme de

MASTER ACADEMIQUE

Domaine : technologie
Filière : électronique
Spécialité : Télécommunications

Présenté par :
Zaouche saadeddine
Bekkouche nabil

Intitulé

Compression des images en utilisant la
transformation en bandelettes

Soutenu le 27 /09/2017

Devant le jury composé de:

M	HIMA Abdelkader	MMA	Directeur du mémoire
M	MADJOURI Abdelkader	MCB	President
M	BOULILA Mohamed	MMA	Examineur

2016 - 2017

REMERCIEMENT

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail. En second lieu, nous tenons à remercier notre encadreur M. HIMA et les membres jurys leurs précieux conseils et ses aides durant toute la période du travail

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Merci à tous et à toutes.

Résumé :

Dans ce travail, nous proposons une technique de compression sur des images fixes l'utilisateur ne s'intéresse qu'à certaines zones d'une même image. Ceci suggère que ses différentes zones peuvent être traitées par des approches différentes réversibles ou irréversibles. Nous proposons une approche de compression adaptative à l'aide de méthodes irréversibles. Après sélection des zones qualifiées d'intérêt.

Les bases de bandelettes sont construites à partir d'ondelettes bidimensionnelles déformées le long du flot géométrique dans chaque région de l'image cette méthode de compression associée à une technique de quantification uniforme, le rôle de cette étude est de l'amélioration des performances de compression pour obtenir une bonne qualité de compression de l'image.

Mots-Clés : Compression d'image , Bandelettes, codeur entropique, quantification

Abstract :

In this work, we propose a compression technique on still images the user is only interested in certain areas of the same image. This suggests that these areas can be treated by different approaches reversible or irreversible. We propose an adaptive compression approach using irreversible methods. After selection of qualified areas of interest.

The strips bases are constructed from two-dimensional wavelet distorted along the geometric flow in each region of the image compression method associated with a technique of uniform quantization, the role of this study is to improve the performance of get compression for a good compression image quality .

Key-words: Image compression,, Bandeletes, Entropy coding, quantization,

ملخص :

في هذه المذكرة، نقوم بدراسة تقنية ضغط على الصور الثابتة حيث أن المستخدم يركز فقط في مناطق معينة من نفس الصورة، وهذا يشير إلى أن هذه المناطق يمكن معالجتها من خلال مقاربات مختلفة عكسية أو غير عكسية . نقترح طريقة ضغط مضبوطة باستخدام أساليب غير عكسية بعد اختيار المناطق التي نريد الاستفادة منها . وتنشأ قواعد الشرائط من الموجات ثنائية الأبعاد المشوهة على طول التدفق الهندسي في كل منطقة من طريقة ضغط الصورة المرتبطة بتقنية تكميم موحدة، ودور هذه الدراسة هو تحسين خوارزمية للحصول على ضغط بجودة صورة مقبولة .

كلمات مفتاحية : ضغط الصور. باندلات. تشفير عدم الاحتمال . تكميم .

Liste des Tableaux

Tableaux 1.1 : codage de couleur	6
Tableaux 1.2 : nombre de bits et nombre de couleurs	6
Tableaux 4.1 : Résultat de l'image Peppers (pq fixe et T varie)	48
Tableaux 4.2 : Résultat de l'image baboon (pq fixe et T varie)	51
Tableaux 4.3 : Résultat de l'image peppers (T fixe et pq varie)	53
Tableaux 4.4 : Résultat de l'image baboon (T fixe et pq varie)	56

Liste des Figures

Figure 1.1 : image Matricielle et image Vectorielle	10
Figure 2.1 : Les étapes classiques de compression d'images.....	13
Figure 2.2 : Exemple de modèle d'horizon et déformation du domaine selon un flot géométrique	21
Figure 2.3 : Exemple de segmentation en quadtree et bande de taille minimale pour la construction de la base de bandelette	22
Figure 3.1 : Exemple de codage Huffman	29
Figure 3.2 : Étapes de codage arithmétique	31
Figure 3.3 : Caractéristique d'un quantificateur	33
Figure 3.4 : Caractéristique typique d'un quantificateur scalaire de $N = 6$ niveaux	34
Figure 3.5 : (a) et (b) quantification scalaire uniforme	35
Figure 3.6 : la Courbe Entrée-Sortie d'un quantificateur uniforme	35
Figure 3.7 : Erreur de quantification	36
Figure 3.8 : quantification scalaire non uniforme	37
Figure 3.9 : pas de quantification de taille variable	37
Figure: 3.10 : Quantification uniforme (à gauche) et non uniforme (à droite)	38
Figure 3.11 : Fonction de compression M-law ($\mu = 1, 10, 255$)	39
Figure 3.12 : (a)-Etapes de quantification par Companding. (b)-quantification μ -law à 8 niveaux. ($\mu=25$)	40
Figure 3.13 : quantification vectorielle (construire un dictionnaire)	41
Figure 3.14 : schéma d'un quantificateur vectoriel [38]	41
Figure 4.1 : Algorithme de compression.....	44
Figure 4.2 : Images de test utilisées	47
Figure 4.3 : Evolution du TC en fonction T pour l'image peppers (pq fixe et T varie)	49
Figure 4.4 : Evolution du PSNR en fonction T pour l'image peppers (pq fixe et T varie)	49

Figure 4.5 : Résultat pour l'image Peppers (pq fixe et T varie)	50
Figure 4.6 : Evolution du TC en fonction T pour limage baboon (pq fixe et T varie).....	51
Figure 4.7 : Evolution du PSNR en fonction T pour limage baboon (pq fixe et T varie).....	52
Figure 4.8 : Résultat pour l'image baboon (pq fixe et T varie)	52
Figure 4.9 : Evolution du TC en fonction du pq pour limage peppers (T fixe et pq varie)	54
Figure 4.10 : Evolution du PSNR en fonction de pq pour limage peppers (T fixe et pq varie)	54
Figure 4.11 : Résultat pour l'image peppers (T fixe et pq varie).....	55
Figure 4.12 : Evolution du TC en fonction de pq pour limage baboon (T fixe et pq varie)	56
Figure 4.13 : Evolution du PSNR en fonction de pq pour limage baboon (T fixe et pq varie)	57
Figure 4.14: Résultat pour l'image baboon (T fixe et pq varie)	57

Lexique d'abréviations

BMP	Bit Map Picture
BPP	Bits par Pixel.
CR	Taux ou Facteur de Compression (Compression Ratio)
CWT	Transformée en Ondelettes Continue
DCT	Discrete Cosine Transform
DFT	Transformation de Fourier Discrète
DFTI	Transformation de Fourier Discrète Inverse
DPI	Dots per Inch
DQS	Distorsion De Quantification Scalaire
DWT	Transformée en Ondelettes Discrètes (Discrète Wavelet Transform)
DXF	Drawing Exchange Format
EPS	Postscript / Encapsulated Postscript
EZW	Embedded Zerotree Wavelet
FFT2D	Transformée de Fourier Rapide a Deux Dimensions
GIF	Graphic Interchange Format
GIF	Graphics Interchange Format
IFFTD	Transformée de Fourier Rapide Inverse a Une Dmension.
JPEG	Joint Photographic Expert Group
$L^2(R)$	L'ensemble des Carrés Sommable
LBG	Linde, Buzo Et Gray
LP	Laplacien Modifié Pyramid
MSE	Erreur Quadratique Moyenne (Mean Square Error)
NB	Noire Et Blanche
PBM	Portable Bitmap File Format
PDFB	Pyramidal Directionnel Filtre-Bank
PGM	Portable Graymap File Format
PNG	Portable Network Graphics
PPM	Portable Pixmap File Format
PSNR	Rapport Signal Sur Bruit Crête.
PSNR	Rapport Signal a Bruit en Pic (Pic Signal to Noise Ratio)
Q(x)	La Fonction de Quantification

QNU	Quantificateur Non Uniforme
QS	La Quantification Scalair
QU	Quantificateur Uniforme
QV	Quantification Vectorielle
R,V,B	(R) Rouge, (V) Vert, (B) Bleu.
RLE	Codage de Longueur de Séquence
SPIHT	Set Partitioning in Hierarchical Trees.
SVG	Scalable Vector Graphics
TC	Taux de Compression.
TCD	Transformation en Cosinus Discret
TIFF	Tagged Image File Format
TOC	Transformation en Ondelettes Continue
TOD	Transformation en Ondelettes Discrètes
USFF	Unequally Spaced FFT
VHS	Système Visuel Humain (Visual Human System)
WT	Transformée en Ondelettes
Δ	Pas de Quantification

Sommaire

Remerciements	<i>i</i>
Résumé	<i>ii</i>
Listes des Tableaux	<i>iii</i>
Liste des Figures	<i>iv</i>
Lexique d'abréviations	<i>vi</i>
Sommaire	<i>viii</i>
Introduction général	1

Chapitre 1: Généralité sur les images

1.1 Introduction	3
1.2 Définition de l'image	3
1.3 Caractéristiques d'une image	3
1.3.1 Taille	3
1.3.2 Résolution	4
1.3.3 Poids	4
1.3.4 Codage de couleur	4
1.4 Types d'images	6
1.4.1 image matricielle	6
1.4.2 image vectorielle	9
1.5 Conclusion	10

Chapitre 2 : Généralités sur la compression

2.1 Introduction	12
2.2 Principes de la compression d'images	12
2.2.1 Schéma général	12
2.2.2 Les trois étapes classiques en compression	13
2.2.3 Mesures de performances et de distorsion	14

2.3 Méthodes de compression	16
2.3.1 Méthodes réversibles (sans perte)	16
2.3.2 Méthodes irréversibles (avec pertes)	16
2.4 Transformation	16
2.4.1 Transformée de Fourier discrète	17
2.4.2 Transformée de cosinus discret	17
2.4.3 La transformée en ondelettes (WT)	18
2.4.4 La transformée en bandelettes	21
2.5 Estimation rapide en bandelettes	23
2.5.1 Image discrétisée	23
2.5.2 L'algorithme	23
2.6 Conclusion	26

Chapitre 3 : Codage et quantification

3.1 Introduction	28
3.2 Codage	28
3.2.1 Codage de Huffman.....	28
3.2.2 Codage arithmétique.....	30
3.3 Quantification	32
3.3.1 Caractéristique d'un quantificateur	33
3.3.2 Quantification scalaire	33
3.3.2 Quantification vectorielle	40
3.4 Conclusion	42

Chapitre 4 : Simulation et résultats

4.1 Introduction	44
4.2 Algorithme de compression	44
4.3 Présentation des résultats	47

4.3.1 Images de test utilisées	47
4.3.2 Etapes de travail	48
4.3.3 Résultat	48
4.4 Conclusion	58
Conclusion générale	59
Bibliographies	60

Introduction général

La compression d'image est une application de la compression de données sur des images numériques. 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.

La compression d'image peut être effectuée avec perte de données ou sans perte. La compression sans perte est souvent préférée là où la netteté des traits est primordiale : schémas, dessins, techniques, icônes bandes dessinées. La compression avec perte, plus radicale, est utile pour les transmissions à bas débit, mais dégrade la qualité de l'image restituée. Les méthodes de compression sans perte sont également préférées là où la précision est vitale : balayages médicaux ou numérisations d'images pour archivage. Les méthodes avec perte restent acceptables pour des photos dans les applications où une perte mineure de fidélité (parfois imperceptible) est tolérée pour réduire les coûts de stockage ou d'envoi.

La transformée en ondelettes discrètes est la technique de transformation la plus populaire adoptée pour la compression d'image.

Ce travail de mémoire s'inscrit principalement dans l'un des aspects de ce domaine : la compression d'images fixes par un type d'ondelettes géométriques dites les bandelettes.

Le manuscrit est structuré de la façon suivante :

Dans le premier chapitre on présente des généralités sur l'imagerie bidimensionnelle comme la définition d'une image numérique, les caractéristique ,les types et quelque format de chaque type ainsi les avantages et les inconvénients .

Le deuxième chapitre permet de présenter des concepts généraux de compression d'images fixes en plus le principe de compression, ainsi qu'une brève présentation des ondelettes classiques et les bandelettes .

Dans le troisième chapitre, nous étudions quelques types de codage et les techniques de quantification scalaire les plus utilisés, dans la compression d'images,

Le quatrième chapitre est réservé pour l'exposition et la discussion de résultats obtenus par les schémas proposes.

Le mémoire s'achève par une conclusion générale et quelques perspectives.

A decorative frame resembling a scroll, with a vertical strip on the left and a horizontal strip at the top, both featuring rounded ends and a slight shadow.

Chapitre 1: généralités sur les images

1.1 Introduction :

Dans ce chapitre, nous allons présenter les différentes phases de la formation d'image ainsi que sa définition, son interprétation par la machine, ces différents types, ces formats, et caractéristiques de ces formats.

1.2 Définition de l'image :

Une image est une fonction de deux variables, par exemple $I(x,y)$ représentera une fonction d'amplitude (l'intensité) de deux variables réelles de position (x,y) dans le plan cartésien. Par exemple, une photographie aérienne, un profil d'élévation d'un terrain, et une carte topographique sont trois représentations différentes d'un même objet, trois images [1] .

- **l'image numérique :**

Le terme d'image numérique désigne, dans son sens le plus général, toute image qui a été acquise, traitée et sauvegardée sous une forme codée représentable par des nombres (valeurs numériques) [2] .

La numérisation est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal qu'elle représente (une infinité de valeur dans l'intensité lumineuse par exemple), à l'état d'image numérique qui est caractérisée par l'aspect discret (l'intensité lumineuse ne peut prendre que des valeurs quantifiées en un nombre fini de points distincts) [2] .

C'est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur [2] .

1.3 Caractéristiques d'une image :

1.3.1 Taille :

Pour connaître la taille d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre des cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. La taille de l'image est alors le nombre des pixels que multiplie la taille (en octet) de chacun de ces éléments.

Exemple : pour une image de 240 X 420 en True Color :

Nombre de pixels : $240 \times 420 = 100800$

Taille de chaque pixel : $24 \text{ bits} / 8 = 3 \text{ octets}$

Le poids de l'image est ainsi égal à : $100800 \times 3 = 302.400$ égal $302.400/1024 = 295$ Ko [3] .

1.3.2 Résolution :

La résolution est définie par un nombre de pixels par unité de longueur de l'image à numériser en dpi (dots per inch) ou ppp (points par pouce)]. On parle de définition pour un écran et de résolution pour une image

Plus le nombre de pixels est élevé par unité de longueur de l'image à numériser, plus la quantité d'information qui décrit l'image est importante et plus la résolution est élevée (et plus le poids de l'image est élevé).

La résolution d'une image correspond au niveau de détail qui va être représenté sur cette image. Pour la numérisation il faut considérer les 2 équations suivantes :

$$(X \times \text{résolution}) = x \text{ pixels}$$

$$(Y \times \text{résolution}) = y \text{ pixels}$$

où X et Y représentent la taille (pouce ou cm, un pouce = 2,54 centimètres) de la structure à numériser, où résolution représente la résolution de numérisation, et où x et y représentent la taille (en pixels) de l'image [3] .

1.3.3 Poids :

Pour connaître le poids (en octets) d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre de cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. Le poids de l'image est alors égal à son nombre de pixels que multiplie le poids de chacun de ces éléments [4] .

1.3.4 Codage de couleur :

Une image est donc représentée par un tableau à deux dimensions dont chaque case est un pixel. Pour représenter informatiquement une image, il suffit donc de créer un tableau de pixels dont chaque case contient une valeur. La valeur stockée dans une case est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur) [4] .

Usuellement on distingue 3 grands types de couleurs pour une image numérique :

- Le noir et blanc .
- Les niveaux de gris .
- La couleur.

Ces types sont généralement à choisir lors d'une numérisation par scanner ou lors de la configuration d'un appareil photographique.

✓ **Image noir et blanc :**

Le noir et blanc est le plus simple. Le contenu de chaque case de la matrice est soit un 0 (noir) soit 1 (blanc). Le nombre de couleurs n'est que de 2 et le rendu de l'image le moins performant mais parfois suffisant dans le cadre par exemple de documents scripturaux [5] .

✓ **Images aux Niveaux de Gris :**

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

Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise entre 0 et 255. Par convention, la valeur zéro représente le noir (intensité lumineuse null) et la valeur 255 le blanc (intensité lumineuse maximale).

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur" de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux [3] .

✓ **Image couleur :**

➤ **Principe :**

La couleur d'un pixel est obtenue, comme le ferait un peintre, par le mélange de couleurs fondamentales. Il ne s'agit pas ici de décrire toutes les techniques utilisées. Nous allons décrire un des principes les plus couramment utilisé qui est celui de la synthèse additive.

➤ **Codage RVB :**

Le principe consiste à mélanger les 3 couleurs : rouge, vert et bleu (noté RVB ou RGB en anglais). A l'aide de ces 3 couleurs, on obtient toute une palette de nuances allant du noir au blanc. A chaque couleur est associé un octet (donc 256 niveaux de luminosité) de chacune des couleurs fondamentales [5] .

Rouge	Vert	Bleu	Couleur
0	0	0	Noir
0	0	1	Nuance de noir
255	0	0	Rouge
0	255	0	Vert
0	0	255	Bleu
28	128	128	Gris
255	255	255	Blanc

Tableaux 1.1 : codage de couleurs [5] .

- Les images informatisées se présentent en plusieurs nombre de couleurs

nombre de bits	Nombre de couleurs possibles
1	2
4	16
8	256
16	65 536
24	16 777 216
32	4194967296

Tableaux 1.2 : nombre de bits et nombre de couleurs [7].

1.4 Types d'images :

Pour représenter une image, on peut la décrire à l'aide de fonctions mathématiques (représentation vectorielle) ou par l'ensemble des points qui la composent (représentation matricielle). Donc on a On distingue deux types d'images.

1.4.1 image matricielle :

✓ Définition :

Une image matricielle (de l'anglais bitmap), est une image constituée d'une matrice de points colorés. C'est-à-dire, constituée d'un tableau, d'une grille, où chaque case possède une couleur qui lui est propre et est considérée comme un point. Il s'agit donc d'une juxtaposition de points de couleurs formant, dans leur ensemble, une image. [7] .

✓ Quelques formats d'images matricielles :

➤ Le portable pixmap file format (PPM) :

Le format ppm concerne les image couleurs .chaque pixel a pour valeur un triplet (R.G.B) composé d'une composante rouge , vert et bleue .chaque composant est représentée

par un entier pouvant prendre ses valeurs entre 0 et 255 . le triplet (0,0,0) correspond au noir , (255,0,0) au rouge , (255,255,0) au jaune, ... , et (255,255,255) au blanc [8] .

➤ **Le portable graymap file format (PGM) :**

Le format PGM permet de représenter des image en niveaux de gris dont les pixels ont des valeurs entières comprises entre 0 (noire) et 255 (blanc) [8] .

➤ **Le portable bitmap file format (PBM) :**

Le format **BitMaP** (BMP) est un format d'image matricielle développe par Microsoft et IBM. C'est un format simple, ouvert et facilement utilisable par les logiciels mais il est peu utilise sur le web a cause de la taille volumineuse de ses fichiers, la couleur de chaque pixel étant codée sans effort de compression [9] .

ce format permet de stocker des images en noir et blanc ces-t-adire dont les pixels ne peuvent prendre que 2 valeurs 0 (noir) ou 1 (blanc) .

➤ **Le Portable Network Graphics (PNG) :**

Ce format est l'émanation de recommandations du consortium W3 (1996), il est donc recommandé de l'utiliser, il n'est cependant pas reconnu par les navigateurs d'ancienne génération.

Le format PNG étant très récent, son principal inconvénient est d'être inconnu de la plupart des logiciels.

Ce format est libre de droits. Il a été conçu pour remplacer GIF, Unisys ayant imposé en 1994 des royalties sur la compression LZW (ce qui ne sera plus en 2006) [10] .

➤ **JPEG Joint Photographic Expert Group :**

Ce format offre des taux de compression inégalés, même si la qualité de l'image s'en ressent au fur et à mesure que vous augmentez la compression.

Avec des taux de compression élevés donnant lieu à des fichiers images de petite taille, ce format est devenu le standard des formats d'image sur internet, que ce soit sur Facebook ou par courriel par exemple, se fait en peu de temps. Il est aussi instantanément imprimable et utilisable [11] .

➤ **JPEG2000 :**

apporte quelques amélioration au JPEG classique et notamment permet un réglage autorisant une compression sans perte ou encore la résistance aux erreurs de transmission.

➤ **GIF Graphics Interchange Format :**

C'est un format à l'origine adapté au transfert d'image sur le réseau Compuserve; il est très répandu pour sa facilité et sa rapidité de lecture.

Ce format était à l'origine uniquement compatible avec le monde IBM, puis a été étendu au monde Macintosh.

Ce format est sous brevet Unisys. Il s'agit donc d'un format « propriétaire », ce qui pose des problèmes de droits d'utilisation sur Internet.

Son faible volume est dû: d'une part à la compression systématique (LZW), d'autre part au codage de la couleur sur 8 bits/pixel.

Ce format ne gère donc que 256 couleurs ou niveaux de gris, on parle alors de couleurs indexées.

Le format GIF est assez bien adaptée aux images graphiques tels que les logos, qui ne nécessitent pas une palette de couleurs très étendue [10].

➤ **TIFF Tag Image File Format :**

Ce format permet une représentation matricielle en vraies couleurs (16 millions en RGB, YCrCb, CMYK, CIE Lab).

Ce format support de nombreuses informations additionnelles sur les couleurs, utiles pour le calibrage couleur (e.g. correction gamma).

Format « propriétaire » mis au point par Aldus corporation.

Ce format d'images peut être avec (compression LZW, JPEG, ...) ou sans compression (Brut). De plus, il existe différents modes de compressions (sans perte) possibles que l'on peut choisir (ou pas) lors de l'enregistrement du fichier. [10].

Avantages des images matricielles :

Le mode de codage des images bitmap (24 bits, codage RGB) les rend adaptées au fonctionnement des principaux périphériques, notamment les contrôleurs d'écran "true colors" (point allumé ou non, codé sur x bits).

Contrairement à image vectorielle, le format convient aux images complexes, principalement d'origine analogique, qui ne peuvent être codées qu'en mode point (photographie).

Elles se laissent manipuler et traiter par des opérations techniques "naturelles" pour un graphiste qui retrouve des outils et les manipulations très proches de ceux qui caractérisent son métier et sa pratique professionnelle de type analogique [12] .

Inconvénients des images matricielles :

Plus lourd que le format vectorielle. Une image matricielle prend plus de place en mémoire.

Elles supportent mal les opérations de redimensionnement, réduction ou agrandissement. Les deux opérations se traduisent par une perte d'information [12] .

Utilisations de l'image matricielles :

Le format matriciel a été utilisé pour envoyer un message dans l'espace à destination d'une intelligence extraterrestre .

1.4.2 image vectorielle :

✓ Définition:

Une image vectorielle est définie par un ensemble de données mathématiques : coordonnées, fonctions, attributs, etc , Composées de formes géométriques, plus adaptées à la création graphique. En général, le fichier correspondant est peu volumineux [9] .

✓ Quelques formats d'images vectorielles :

➤ Le format SVG (Scalable Vector Graphics) :

est un format ouvert d'image vectorielle, il est surtout utilise en cartographie et sur les téléphones portables [9] .

➤ Le format ODG (Open Document Graphic) :

est un format ouvert de dessin vectoriel, il est utilise par l'application (Draw d'Open Office) [9] .

Utilisations de l'image vectorielle :

L'image vectorielle est surtout utilisée pour réaliser la barre d'outils dessin des logiciels de bureautique ou le diagramme créé dans un logiciel de présentation [9] .

Avantages :

Ces images présentent 2 avantages : elles occupent peu de place en mémoire et peuvent être redimensionnées sans perte d'informations et sans effet dit : d'escalier [13] .

Inconvénients :

- Les images d'origine analogique comme les photographies sont difficilement transformées en images vectorielles du fait de leur complexité.
- L'affichage implique des calculs car il faut transformer l'image vectorielle en une image matricielle. Un fichier vectoriel est en fait un programme; une dégradation d'une partie du fichier est désastreuse (un fichier en mode " texte " comme un fichier Postscript permet d'atténuer cette difficulté) [15] .

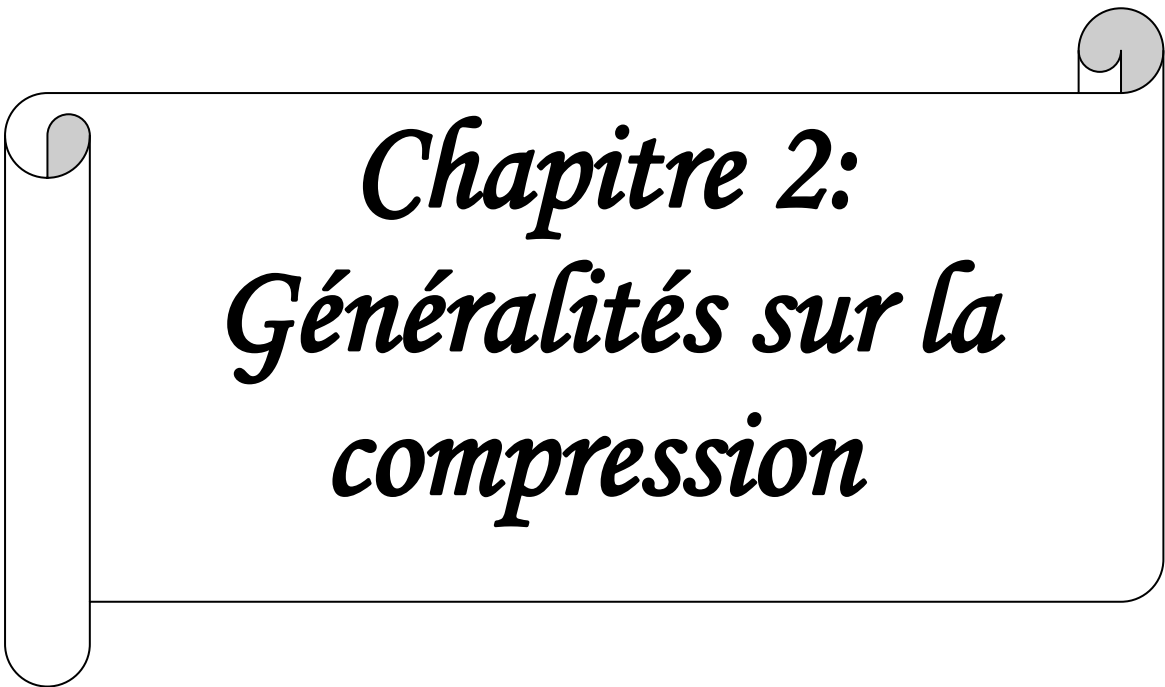
➤ Exemple d'une image Matricielle et Vectorielle :

Figure 1.1: image Matricielle et image Vectorielle [14] .

1.5 Conclusion :

Les images numériques sont de plusieurs types : couleurs, noir et blanc, niveaux de gris. Celles en couleurs peuvent être représentées sur des différentes espaces comme l’RVB .

Dans ce chapitre nous avons présenter en générale les images numériques, ils en existe plusieurs formats où chaque format a des caractéristiques spéciales, souvent ces formats servent à compresser et permettent de minimiser l'espace de stockage avec un taux de compression bien élevé .

A decorative graphic of a scroll with a light gray background and a black outline. The scroll is partially unrolled, with the top and bottom edges showing a slight curve. The text is centered on the scroll.

Chapitre 2: Généralités sur la compression

2.1 Introduction:

La compression d'image a été développée pour transmettre une quantité massive de données d'image en temps réel via des canaux de bande passante limitée. Les données sont sous forme de graphiques, audio, vidéo et image. Ces types de données doivent être comprimés pendant le processus de transmission. L'image numérique est essentiellement un tableau de valeurs différentes de pixels. Dans l'image numérique, les pixels de voisinage sont corrélés de sorte que ces pixels contiennent des bits redondants. En utilisant les algorithmes de compression, les bits redondants sont supprimés de l'image de sorte que la taille de l'image est réduite et que l'image est comprimée.

La compression d'image comporte deux composantes principales: réduction de la redondance et réduction des données non pertinentes. La réduction de redondance est obtenue en supprimant des bits supplémentaires ou des bits répétés. Dans une réduction non pertinente, les informations les plus petites ou les moins importantes sont omises, ce qui ne sera pas reçu par le récepteur. Les trois types de redondances, c'est-à-dire la redondance de codage, sont utilisés lorsqu'un nombre de mots de code inférieur est requis au lieu du symbole plus grand. La redondance entre pixels produit une corrélation des pixels d'une image et des données de redondance psycho-visuelle est ignorée par le système visuel normal. La compression d'image est appliquée pour réduire le nombre de bits qui représentent l'image.

2.2 Principes de la compression d'images :

2.2.1 Schéma général :

Le schéma d'un système de compression classique est représenté sur la figure 2.1. La première étape est la transformation. Elle permet d'exprimer l'image sous une autre forme redondante. Ensuite vient l'étape de quantification elle est l'étape qui augmente les redondances des pixels donc la perte d'information aura lieu. Après ça vient l'étape de codage pour la réduction de la quantité d'éléments binaires représentant l'information « image ».

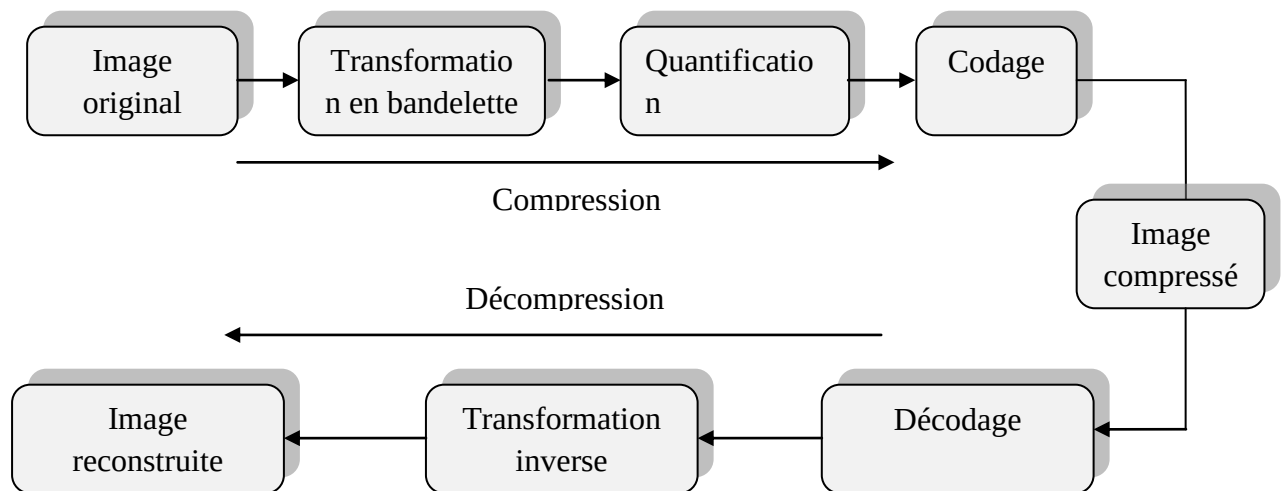


Figure 2.1 : Les étapes classiques de compression d'images [16].

Après codage on aura une image comprimée qui va subir un chemin inverse en vue de la décompresser. Un décodage pour inverser l'effet du codage puis une transformation inverse pour inverser la transformation en bandelettes et on aura une image reconstruite.

2.2.2 Les trois étapes classiques en compression:

Les méthodes de compression avec pertes d'images actuelles suivent les 3 étapes classiques de compression d'images [17]. Elles débutent pour la plupart par une réorganisation du contenu de l'image, afin de séparer les composantes importantes des composantes contenant peu d'informations. Cette tâche est remplie par une transformation mathématique.

Cette étape est suivie par la quantification qui dégrade de manière irréversible le signal puis vient la dernière étape; le codage (sans perte) qui produit un flux binaire.

✓ Transformation des données :

Le but de la transformée dans un schéma de compression est double. En effet, plus de réorganiser l'information, elle doit représenter les composantes importantes d'un signal avec le moins d'éléments possibles : c'est ce qu'on appelle une représentation creuse du signal ou, compacter l'énergie.

La réduction de la redondance a pour objectif de réduire les corrélations entre les pixels. Elle concentre les variations (énergies) de l'image sur quelques échantillons et répartie presque uniformément les pixels corrélés.

✓ Quantification :

L'étape de quantification est celle qui dégrade de manière irréversible le signal. Elle est cependant d'une importance capitale dans la réduction du débit binaire. La quantification est une opération qui transforme un signal continu en un signal discret à l'aide d'un ensemble appelé dictionnaire. Ce passage du continu au discret peut s'effectuer échantillon par échantillon. Malgré le fait que la quantification permet de gagner le taux de compression, est une étape où l'on perd de l'information. Comme l'opération de quantification consiste à enlever certaines informations, c'est ainsi une opération avec perte, qui donnera alors une perte de précision, qui est irréversible. Plus la précision de quantification est faible, plus on gagne en taux de compression, mais on perd également plus d'information [18].

✓ Codage :

Il y a deux grandes familles de codeurs qui sont: les codeurs entropiques et les codeurs par plages. Ils sont utilisés dans une chaîne de compression directement sur l'image de départ et ils sont aussi également employés à la dernière étape de la chaîne de compression afin d'exploiter les redondances présentes à la sortie du quantificateur [19].

Les codes entropiques sont basés sur la génération de mots dont la longueur dépend de la probabilité d'apparition des symboles de la source qu'il représente : un grand nombre de bits sera utilisé pour coder un symbole peu probable tandis qu'un symbole redondant sera codé sur très peu de bits.

Le codage par plages (ou run-length en anglais) consiste à coder la longueur d'une série d'échantillons nuls plutôt que de coder chaque échantillon indépendamment.

2.2.3 Mesures de performances et de distorsion :

Lorsqu'on utilise des méthodes non réversibles pour la compression d'images, il est utile de pouvoir mesurer la perte de qualité de l'image reconstruite par rapport à l'image initiale afin de pouvoir optimiser la compression.

La distorsion est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale.

✓ Taux ou Facteur de compression :

Le taux de compression donne une mesure de performance des méthodes de compression des images fixes [20] est donnée par (2.1).

$$CR = \frac{\text{nombre de bits de l'image compressée}}{\text{nombre de bits de l'image originale}} \dots\dots\dots (2.1)$$

CR : taux de compression (Compression Ratio).

Ce taux est une mesure de performance de l'algorithme de compression. Grâce à ce taux de compression on peut comparer les différents algorithmes.

Pour une même méthode de compression et un même CR réalisé sur des images distinctes, la qualité obtenue peut être très variable d'une image à l'autre. Les propriétés statistiques des images originales jouent un rôle prépondérant dans le résultat obtenu. Par exemple, dans le cas d'une image sur-échantillonnée qui est donc très redondante, il est facile d'obtenir des taux bas. La théorie de l'information donne une limite théorique au CR maximal qu'il est possible d'obtenir sans distorsion pour toute méthode de compression sur une image donnée.

✓ **Système visuel humain: VHS (Visual Human System) :**

L'œil est un outil essentiel pour apprécier la qualité d'une image. Il va permettre à l'utilisateur d'identifier le contenu des images, la netteté de celles-ci et la qualité des contours.

✓ **Erreur quadratique moyenne :**

Pour les méthodes avec pertes, la mesure de distorsion utilisée généralement en compression d'image, pour pouvoir comparer les différentes méthodes de compression, est l'erreur quadratique moyenne MSE (Mean Square Error) [20] :

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[f(i,j) - \hat{f}(i,j) \right]^2 \dots\dots\dots (2.2)$$

$f(i,j)$: Image originale.

$\hat{f}(i,j)$: Image reconstruite.

M.N : Taille de l'image.

Plus l'erreur quadratique moyenne est grande, plus il y a de différence de qualité entre l'image originale et l'image reconstruite. Pour la compression sans perte, comme il n'y a aucune différence entre l'image reconstruite et l'image originale, l'erreur quadratique moyenne est nulle.

✓ Rapport signal à bruit en pic :

La mesure la plus couramment utilisée par la communauté internationale est la mesure du rapport signal à bruit en pic PSNR (Pic Signal to Noise Ratio), il se mesure en décibel (dB) et il est défini par [20] :

$$\text{PSNR} = 10 \cdot \log_{10} \frac{(2^R - 1)^2}{MSE} \text{ dB} \dots\dots\dots(2.3)$$

où R représente le nombre de bits désignés pour un pixel. Quelques auteurs définissent

2.3 Méthodes de compression:

Selon les besoins, les techniques de compression d'images se différencient par le fait qu'elles permettent de compresser avec ou sans perte d'informations. Alors, on distingue deux grandes catégories.

2.3.1 Méthodes réversibles (sans perte) :

Quand on parle de compression sans perte (Lossless Compression) on fait souvent allusion aux opérations de codage. Dans ce cas les données compressées et décompressées sont tout à fait équivalentes. Ces techniques de compression sans pertes ou de codage sont qualifiées de compression réversible. Elles sont plus particulièrement utilisées pour les données informatiques dont l'exactitude est décisive (Exemple : données textuelles, programmes, images médicales...). Normalement, les facteurs de compression accomplis par ces algorithmes n'aboutissent qu'à des valeurs faibles de l'ordre de 2 à 3 au maximum [21].

2.3.2 Méthodes irréversibles (avec pertes) :

Quant aux méthodes de compression avec pertes (Lossy Compression) elles sont basées essentiellement sur l'élimination d'une partie de l'information jugée redondante. Ceci a pour effet l'obtention de meilleurs taux de compression, de plus en plus élevés, au détriment d'une certaine dégradation. Pour cette raison ces techniques de compression sont dites irréversibles. En effet, l'image originale et l'image reconstruite après compression ne sont pas exactement les mêmes. (Exemple : la DCT et les Ondelettes.) [21].

2.4 Transformation :

La transformation permet de réorganiser l'information et de représenter les composantes importantes d'un signal avec le moins d'éléments possible. Les pixels passent d'un espace où ils sont fortement corrélés dans un autre espace où leur corrélation est moindre, le but de

codage par transformée est donc de trouver une base dans laquelle, d'une part les échantillons du signal se trouvent naturellement décorrélés, et d'autre part les informations pertinentes sont facilement identifiables. Parmi les principales transformées utilisées en codage d'images, citons :

- Transformation en cosinus discret (TCD).
- Transformée en ondelettes discrètes (DWT) et les décompositions par bancs de filtres.
- On mesure l'efficacité de ces transformations par : l'efficacité de décorrélation, la concentration de l'énergie, et l'existence d'algorithmes rapides pour calculer les transformations.

2.4.1 Transformée de Fourier discrète :

La transformation de Fourier discrète (DFT) est un outil mathématique de traitement du signal numérique, qui est l'équivalent discret de la transformation de Fourier continue qui est utilisée pour le traitement du signal analogique. En effet, un tel opérateur ne peut traiter que des nombres et de plus en quantité par la taille de la mémoire.

Soit un signal discret $x(n)$, la DFT et la DFTI sont données par les formules suivantes :

La transformation direct : (DFT)
$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{nk}{N}} \dots\dots\dots (2.4)$$

La transformation inverse : (DFTI)
$$X(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(k) e^{j2\pi \frac{nk}{N}} \dots\dots\dots (2.5)$$

2.4.2 Transformée de cosinus discret :

L'étape suivante, après la conversion de coordonnées de couleur est de diviser les trois composantes de couleur de l'image en plusieurs blocs 8×8 . La définition mathématique du TCD avant et le TCD inversé sont les suivantes:

Vers l'avant DCT

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \cos \left[\frac{\pi(2y+1)v}{2N} \right] \dots\dots (2.6)$$

Pour $u = 0, \dots, N-1$ et $v = 0, \dots, N-1$

$$\text{où } N = 8 \text{ et } C(k) = \begin{cases} 1/\sqrt{2} & \text{pour } k = 0 \\ 1 & \text{sinon} \end{cases}$$

Inverse DCT

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \cos \left[\frac{\pi(2y+1)v}{2N} \right] \dots\dots(2.7)$$

Pour $x = 0, \dots, N-1$ et $y = 0, \dots, N-1$ où $N = 8$

Le $f(x, y)$ est la valeur de chaque pixel dans le bloc sélectionné 8×8 , et le $F(u, v)$ est le coefficient DCT après la transformation. La transformation du bloc 8×8 est également un bloc 8×8 composés de $F(u, v)$. La DCT est étroitement liée à la DFT. Les deux prenant un ensemble de points du domaine spatial et les transformer en une représentation équivalente dans le domaine de la fréquence. Cependant, pourquoi DCT est plus approprié pour la compression d'image que DFT? Les deux principales raisons sont:

1- la DCT peut concentrer l'énergie du signal transformé en basse fréquence, alors que la DFT ne peut pas. Selon le théorème de Parseval, l'énergie est la même dans le domaine spatial et dans le domaine fréquentiel. Parce que les yeux humains sont moins sensibles à la composante basse fréquence, nous pouvons nous concentrer sur la composante basse fréquence et réduire la contribution de la composante haute fréquence après la prise de DCT.

2- Pour la compression d'image, le DCT peut réduire l'effet de blocage que le DFT.

Après transformation, l'élément le plus à gauche correspondant à une fréquence nulle dans les deux directions est le «coefficient DC» et le reste est appelé «coefficients AC» [22].

2.4.3 La transformée en ondelettes (WT):

✓ La transformée en ondelettes continue (CWT)

Cette transformation consiste à des translations et des dilatations d'une seule fonction fixée, l'ondelette $\psi \in L^2(\mathbb{R})$. Dans la transformée en ondelette continue, les paramètres de translation et de dilatation varient continûment. Cela signifie que nous utilisons les fonctions :

$$\Psi_{ab}(x) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right) \text{ avec } a \in \mathbb{R} - \{0\}, b \in \mathbb{R} \quad (2.8)$$

Ces fonctions sont mises à l'échelle de telle sorte que leur norme dans $L^2(\mathbb{R})$ soit indépendante de a . La CWT d'une fonction $f \in L^2(\mathbb{R})$ est défini par :

$$W(a,b) = \langle f, \Psi_{a,b} \rangle \quad (2.9)$$

En utilisant l'identité de Parseval, on peut aussi l'écrire sous forme :

$$W(a,b) = \frac{1}{\sqrt{2\pi}} \langle F, \Psi_{a,b} \rangle \quad (2.10)$$

$$\text{Ou } \Psi_{ab}(\omega) = \frac{1}{\sqrt{|a|}} e^{-i\omega} \Psi(a\omega) \quad [23].$$

On notera que la CWT convertit une fonction à une variable en une fonction à deux variables. La représentation d'une fonction par sa CWT est redondante et la transformée inverse n'est donc pas toujours unique. De plus, toutes les fonctions $W(a,b)$ ne sont pas forcément la CWT de la fonction f . Si l'ondelette Ψ satisfait la condition d'admissibilité

$$C_\Psi = \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$$

Alors, la transformée d'ondelette continue $W(a,b)$ admet un inverse

$$f(x) = \frac{1}{C_\Psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} w(a,b) \Psi_{ab}(x) \frac{da db}{a} \dots \dots \dots (2.11)$$

D'après la condition d'admissibilité, nous observons que $\Psi(0)$ et par conséquent Ψ doit osciller.

Cette transformée est souvent représentée par une image 2D (en couleurs ou en niveaux de gris) correspondant en module et phase de la transformée $w(a,b)$. L'avantage de cette caractérisation par rapport à la transformée de Fourier est qu'elle ne donne pas seulement une information sur le type de singularité mais aussi sa localisation dans le temps [23].

✓ La transformée en ondelettes discrète (DWT) :

La transformée en ondelette continue est très redondante. Il suffit en pratique de quelques coefficients pour couvrir entièrement le plan temps échelle. Les paramètres a et b sont fondamentalement continus. Leur discrétisation ne s'effectue pas selon un critère bien défini. Quant à la transformée en ondelettes discrète le concept est différent. En effet, numériquement nous devons savoir comment discrétiser cette transformée de telle manière à conserver les propriétés intéressantes. Nous donnons donc que les paramètres de dilatation a et de translation b puissent être discrétisés sans perte d'information. Nous pouvons donc formuler le problème de l'existence d'une famille $\{\Psi_{a_m, b_n}\}$ dénombrable, formant une base de L^2 et permettant une reconstruction exacte du signal en question. Cette famille redondante d'ondelettes discrètes, a été introduite dans le cadre des frames ou structures obliques par Ingrid Daubechies [25]. La redondance d'informations contenues dans cette transformée peut être généralement intéressante dans le cadre de l'analyse des signaux. En effet, elle est

intéressante dans la mesure où les conditions que doivent vérifier les ondelettes ne sont pas trop contraignantes [26]. La reconstruction numérique du signal est stable. D'autre part, avec l'introduction des bases orthonormées d'ondelettes [27], la théorie des ondelettes a pris un tournant décisif dans le sens où cette notion de base orthonormée constitue un outil clef en traitement du signal ; elle a des contraintes de rapidité nécessitant le minimum de données possible [24].

Dans la cadre de la théorie des frames, I. Débauchies a proposé une discrétisation des facteurs d'échelle a et de translation b telle que :

$$a_m = a_0^m, m \in Z \text{ et } a_0 > 1$$

$$b_n = nb_0 a_0^m, n \in Z \text{ et } b_0 > 1$$

Alors la famille d'ondelettes est de la forme:

$$\Psi_{m,n}(x) = a_0^{-m/2} \Psi(a_0^{-m}x - nb_0) \dots \dots \dots (2.12)$$

Une telle discrétisation permet à la famille, à chaque échelle a_m , de couvrir tout le support de signal sans trop de redondance, chaque ondelette étant essentiellement concentrée sur l'intervalle $[a_0^m nb_0, a_0^m (n+1)b_0[$. Par contre la transformée ne sera plus invariante en translation comme la transformée continue, raison de la dépendance des paramètres. Pour exprimer les coefficients de la transformée en ondelette discrète, nous les notons par d_n^m

$$d_n^m = \langle \Psi_{m,n}, f \rangle = a_0^{-m/2} \int \bar{\Psi}(a_0^{-m}x - nb_0) f(x) dx \dots \dots \dots (2.13)$$

Nous montrons dans la littérature spécialisée [26] que si la famille $\{\Psi_{m,n}, m, n \in Z\}$ est une frame, alors nous aurons :

$$\frac{b_0 \log(a_0)}{\pi} A \leq \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega \leq \frac{b_0 \log(a_0)}{\pi} B \dots \dots \dots (2.14)$$

Où A et B sont deux constantes positives d'un frame donnée.

Cette expression nous montre simplement que la condition d'admissibilité est vérifiée et l'ondelette $\Psi_{m,n}$ est analysante. Dans ce cas, la construction des algorithmes numériquement stable pour reconstruire le signal à partir des ses coefficients ondelettes d_n^m est possible:

$$f(x) = \frac{2}{A+B} \sum_{m,n} \langle \Psi_{m,n}, f(x) \rangle \Psi_{m,n} + R \dots \dots \dots (2.15)$$

Avec : $\|R\| \leq o\left(\frac{B}{A} - 1\right) \|f(x)\|$, il exprime un terme d'erreur.

Si la base est orthonormée alors le terme R sera nul. Cependant, il est préférable de réduire au maximum la redondance de cette représentation. Dans ce cas, on choisit des valeurs de $a_0 = 2, b_0 = 1$ pour lesquelles les ondelettes $\Psi_{a_0 b_0}$ constituent une base orthonormée. Nous concluons que l'orthonormalité est une condition nécessaire pour le passage de la transformée en ondelettes continue vers la transformée en ondelettes discrète [29].

2.4.4 La transformée en bandelettes :

La transformée en bandelettes, introduite par Le Pennec et Mallat dans [30], construit une base adaptée au contenu géométrique d'une image. Pour des fonctions avec une régularité géométrique, l'approximation dans une meilleure base de bandelettes permet d'avoir une décroissance asymptotique de l'erreur optimale. La construction originale de [30] effectue d'abord une déformation locale sur des sous-carrés pour aligner la singularité avec la direction horizontale ou verticale. Une étape de bandelétisation utilisant des produits tensoriels d'ondelettes est ensuite appliquée pour absorber la régularité le long de la géométrie.

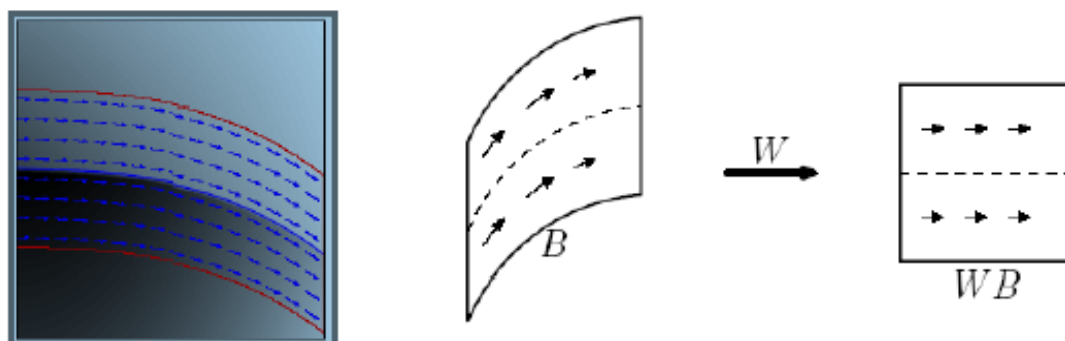


Figure 2.2 : Exemple de modèle d'horizon et déformation du domaine selon un flot géométrique [30].

La construction d'une base de bandelettes s'effectue sur un domaine B où la fonction f considérée est un modèle d'horizon, comme montré à la figure 2.2. La bande B contient une unique courbe de singularité, que l'on peut approcher par une courbe paramétrée (par exemple horizontalement) par $x_2 = \gamma(x_1)$. L'opérateur de déformation

$$\forall x \in B, w(x) = (x_1, x_2 - \gamma(x_1)) \dots\dots\dots (2.16)$$

permet de définir le domaine déformé $WB \stackrel{\text{def.}}{=} w(B)$ (supposé carré), ainsi que la fonction warpée $Wf(x) = f(w^{-1}(x))$.

À partir d'une fonction d'ondelette 1D ψ associée à une fonction d'échelle ϕ on peut définir une base d'ondelettes déformées de $L^2(B)$

$$\{\psi_{j,n}^k(w(x)) \mid j \leq j_{WB}, n \in I_{WB} \text{ et } k \in \{V, H, D\}\}$$

où $2^{j_{WB}}$ est la taille du carré WB et I_{WB} est un ensemble d'indices de positions.

Les ondelettes verticales ψ_{jn}^V et diagonales ψ_{jn}^D possèdent des moments nuls le long de γ , elles sont donc adaptées pour l'approximation de f qui est régulière le long de γ . Ce n'est pas le cas des ondelettes horizontales ψ_{jn}^H , qu'il faut remplacer par de nouvelles fonctions anisotropes définies par

$$\forall j_1, j_2 \leq j_{WB}, \psi_{j_1 j_2 n}^H(w(x)) \stackrel{\text{def.}}{=} \psi_{j_1 j_1}(x_1) \psi_{j_2 j_2}(x_2 - \gamma(x_1)) \dots \dots \dots (2.17)$$

On obtient ainsi la base orthonormée de bandelettes de $L^2(WB)$.

Pour approcher une fonction f avec une régularité géométrique, il faut segmenter le domaine $[0, 1]$ en carrés à l'intérieur desquels la fonction f est un modèle d'horizon. La figure 2.3 gauche, montre une telle segmentation utilisant un quadtree.

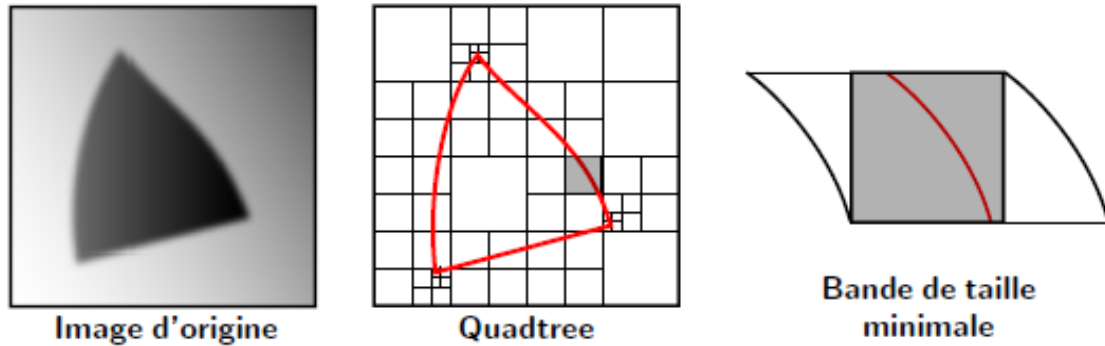


Figure 2.3 : Exemple de segmentation en quadtree et bande de taille minimale pour la construction de la base de bandelette [30].

Pour pouvoir appliquer la construction précédente, il faut agrandir chaque petit carré de la segmentation pour obtenir une bande B que l'on peut ensuite warper, comme le montre la figure 2.3 droite. On obtient une base de bandelettes de $L^2([0, 1]^2)$ en restreignant l'ensemble des fonctions obtenues sur chaque bande aux carrés de la segmentation.

2.5 Estimation rapide en bandelettes:

Dans cette section, nous décrivons chaque étape du calcul de l'estimateur en bandelettes.

2.5.1 Image discrétisée:

Les calculs qui suivent sont effectués non pas sur des fonctions $f \in L^2$ mais sur des images discrétisées $\bar{f} \in \ell^2$, ce qui correspond à la pratique numérique. Une image discrétisée $\bar{f}$ de $N \times N$ pixels est obtenue par la projection d'une fonction $f \in L^2([0, 1]^2)$ sur un ensemble de fonctions d'échelle $\{\Phi_{Jn}\}_n$ à une résolution $2^J = N^{-1}$

$$\forall n \in \{0, \dots, N-1\}^2, \bar{f}[n] \stackrel{\text{def}}{=} \langle f, \Phi_{Jn} \rangle \dots \dots \dots (2.18)$$

$$\Phi_{Jn}(x) \stackrel{\text{def}}{=} 2^{-J} \Phi(2^{-1}x - n) \dots \dots \dots (2.19)$$

2.5.2 L'algorithme :

Le calcul de l'estimateur en bandelettes inclut la transformée en bandelettes, un seuillage adapté suivant l'échelle et enfin la transformée en bandelettes inverse.

La transformée en bandelettes est calculée avec un algorithme rapide qui comprend à la fois le calcul de la meilleure base de bandelettes, $B^*(I_j^K)$ pour chaque échelle et le calcul de la décomposition de l'image inversée sur cette base.

A .Données de l'algorithme :

L'utilisateur dispose de données bruitées et discrétisées $\bar{y} = R\bar{f} + \bar{w}$ où w est une réalisation d'un bruit blanc gaussien de variance ε . On commence par calculer le signal discrétisé inversé $\bar{g} \stackrel{\text{def}}{=} R^{-1} \bar{y} = \bar{f} + R^{-1} \bar{w}$ de taille $N \times N$ pixels.

On souhaite donc estimer le signal discrétisé d'origine $\bar{f}$ à partir de l'image bruitée $\bar{g}$. La variance du bruit ε peut être estimée directement à partir du signal corrompu $\bar{y}$ à l'aide de techniques standards [31].

B .Transformée en ondelettes 2D :

On calcule la transformée en ondelettes discrètes 2D des données d'origine $\bar{g}$. Ceci donne une série de coefficients $\{\bar{g}_j^k\}_{j>J, k}$. Ces nouvelles images $\bar{g}_j^k$, pour chaque échelle 2^j et orientation $k \in \{V, H, D\}$, peuvent être stockées dans une unique image de la même taille que

l'image l'originale g . Les étapes suivantes (C)-(G) implémentent la bandelétisation et sont répétées pour chaque échelle et orientation [32].

C. Sélection des carrés dyadiques :

Un carré dyadique S est, par définition, obtenu en subdivisant de façon récursive le carré $[0, 1]^2$ en quatre sous-carrés de même taille.

Ceci permet de regrouper les coefficients $\{\bar{g}_j^k[n]\}_n$ par carrés dyadiques formés par les n vérifiant $2^j n \in S$. Les étapes suivantes (D)-(G) sont répétées pour chaque carré dyadique S à une échelle fixée 2^j et orientation k de la transformée en ondelettes. On note $gs \stackrel{def}{=} \{\bar{g}_j^k[n]\}_{(2^j n) \in S} \in \ell_j^2(S)$ l'ensemble des coefficients en ondelettes à l'intérieur de S [32].

D. Quantification de la géométrie :

On doit maintenant trouver la meilleure géométrie quantifiée possible $\tilde{\gamma}_S$ à l'intérieur du carré S . On va donc tester, pour chaque type de géométrie $\Omega \in \{H, V, \Xi\}$ un ensemble suffisant de courbes géométriques (sauf bien sûr pour le cas $\Omega = \Xi$ où l'on ne retransforme pas gs).

Dans la suite, on note $\{\theta_i\}_{i=0}^{\alpha-1}$ la base orthogonale des polynômes de Lagrange sur l'intervalle couvert par S (d'abord horizontalement, puis verticalement suivant la valeur de Ω). On teste l'efficacité de la base de bandelettes $B(S, \tilde{\gamma}_m^\Omega)$ de $\ell_j^2(S)$ pour les géométries

$$\tilde{\gamma}_m^\Omega \stackrel{def}{=} \sum_{i=0}^{p-1} (m_i \varepsilon^2) \theta_i \quad \text{Ou} \quad |m_i| \leq A_\theta \varepsilon^{-2} \quad \text{et} \quad \Omega \in \{V, H\} \dots\dots\dots (2.20)$$

pour chaque catégorie $\Omega \in \{V, H\}$ et pour chaque ensemble de coefficients

$$m \stackrel{def}{=} \{m_i\}_i \text{ vérifiant } |m_i| \leq A_\theta \varepsilon^{-2}.$$

L'étape suivante (E) est répétée pour chaque géométrie potentielle $\tilde{\gamma}_m^\Omega$, ainsi que pour le cas particulier $\Omega = \Xi$ [32].

E. Calcul de la transformée de Alpert :

Pour une géométrie donnée $\tilde{\gamma} = \tilde{\gamma}_m^\Omega$, on calcule l'ensemble des produits scalaires

$$A_\gamma(gs)[l, i] \stackrel{def}{=} \langle gs, \Psi_{li} \rangle \quad \text{ou} \quad B(S, \tilde{\gamma}) \stackrel{def}{=} \{\Psi_{li}\}_{li} \dots\dots\dots (2.21)$$

Cette transformée A_γ peut être calculée avec l'algorithme de transformée de Alpert discrète comme expliqué dans [35]. Pour le cas particulier $\tilde{\gamma} = \Xi$ la transformation A_Ξ est l'identité.

F. Sélection de la meilleure géométrie :

Il faut maintenant sélectionner la géométrie $\tilde{\gamma}S = \tilde{\gamma}_m^\Omega$ qui minimise le Lagrangien [32]

$$\mathcal{L}(gs, B(S, \tilde{\gamma}S), t_j) \stackrel{def}{=} \sum_{(l,i) \in J_\sigma} |A_\gamma(gS)[l, i]|^2 + (Card(J_\sigma) + M_G) \varepsilon^2 \dots\dots\dots (2.22)$$

où t_j est le seuil qui dépend de l'échelle 2^j

$$\text{et } J_\sigma \stackrel{def}{=} \{(l, i) \mid |A_\gamma(gS)[l, i]| > \varepsilon\} \dots\dots\dots (2.23)$$

et $M_G = 1$ si $\Omega = \Xi$ et $M_G = \alpha$ si non. On est ainsi capable de définir la géométrie optimale à l'intérieur de S par

$$\gamma S = \operatorname{argmin} (\mathcal{L}(gs, B(S, \tilde{\gamma}_m^\Omega), \varepsilon)) \dots\dots\dots (2.24)$$

G. Construction du quadtree :

Une fois que l'on a calculé les coefficients transformés $A_{\gamma_s}(gs)$ pour chaque carré dyadique, on choisit la meilleure segmentation en carrés. Grâce à l'additivité du Lagrangien et à la structure hiérarchique des quadtree, la minimisation de L peut être effectuée avec une procédure rapide comme introduit dans [33] et déjà employé dans [30].

Lors de l'étape précédente (F), nous avons enregistré, pour chaque carré dyadique S , la Valeur $\mathcal{L}(S) \stackrel{def}{=} \mathcal{L}(gs, B(S, \gamma S), t_j)$ du Lagrangien restreint à S , en même temps que la meilleure géométrie quantifiée $\tilde{\gamma}S$. Alors, pour chaque échelle 2^j et orientation k , on calcule la structure de quadtree optimale :

- Initialisation du quadtree : chaque petit carré S de largeur $b = 2^j$ est une feuille de l'arbre. On enregistre dans les feuilles de l'arbre la géométrie optimale $\tilde{\gamma}S$ et on initialise $\mathcal{L}_0$ le Lagrangien cumulatif sur le sous-arbre, à $\mathcal{L}_0(S) = \mathcal{L}(S)$.
- On démarre avec des carrés S de taille $b = 2 \times 2^j$.

- Pour chaque carré S , on note (S_1, S_2, S_3, S_4) ses 4 sous-carrés et

$$\mathcal{L}'(S) \stackrel{def}{=} \mathcal{L}_0(S_1) + \mathcal{L}_0(S_2) + \mathcal{L}_0(S_3) + \mathcal{L}_0(S_4) + t_j^2 \dots\dots\dots(2.25)$$

est le Lagrangien du sous-arbre (le facteur additionnel t_j^2 est dû au coût du codage de la subdivision, évalué à un coefficient). Les sous-carrés sont regroupés si $\mathcal{L}(S) < \mathcal{L}'(S)$.

Si c'est le cas, on déclare S comme une feuille et on enregistre la meilleure géométrie $\tilde{y}_S$ dans l'arbre. On met à jour $\mathcal{L}_0(S) = \min(\mathcal{L}(S), \mathcal{L}'(S))$.

- Tant que $b < 1$, faire $b \leftarrow 2 \times b$ et répéter l'étape précédente.

H. Seuillage des coefficients :

Pour chaque carré S du quadtree, on enregistre les coefficients de la transformée de Alpert $A_{\tilde{y}_S}(gs)$. Ces coefficients peuvent être stockés à la place des coefficients de la transformée en ondelettes. On met ensuite à zéro les coefficients dont la valeur absolue est en dessous du seuil t_j [32].

I. Calcul de la transformée en bandelettes inverse :

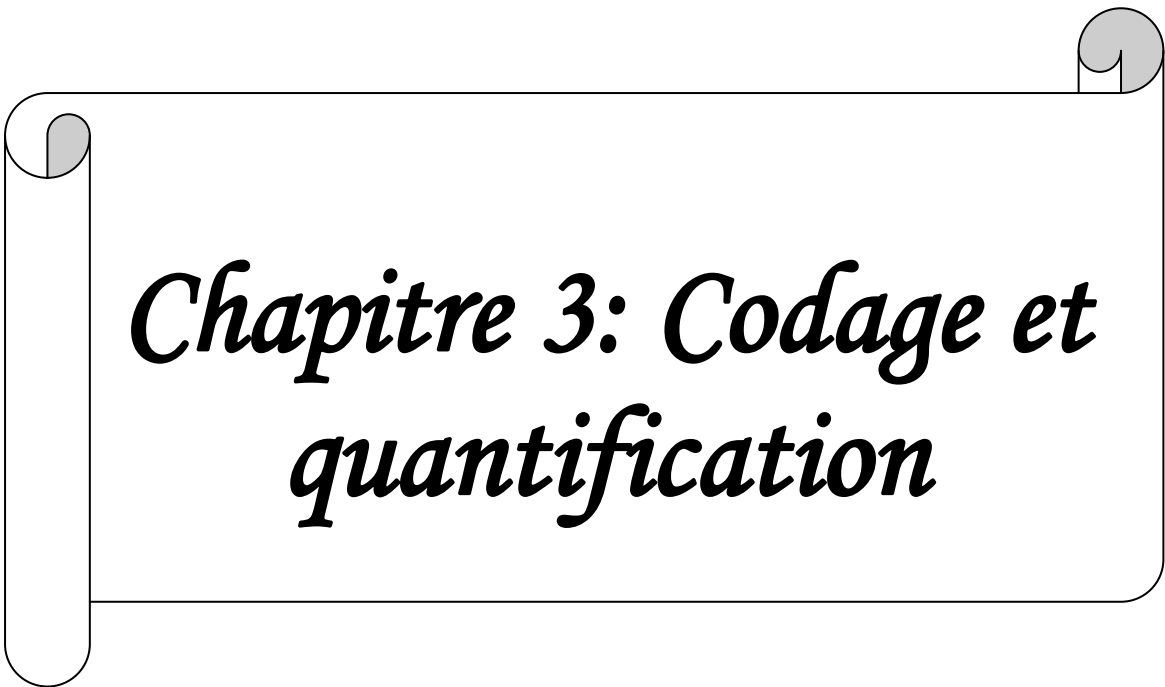
Le calcul de la transformée inverse est implémenté en inversant d'abord la transformée de Alpert sur chaque carré dyadique S faisant parti du quadtree à chaque échelle et chaque orientation, puis en inversant la transformée en ondelettes. Ces calculs ne demandent aucune recherche de meilleure base, puisque l'on utilise les paramètres géométriques (quadtrees et géométries quantifiées), trouvés lors de la transformée directe [32].

2.6 Conclusion :

Dans ce chapitre nous avons présentée en général les deux techniques de compression avec pertes et sans perte en plus la transformation en ondelettes et bandelette.

Pour la compression d'images, les ondelettes apparaissent donc comme un outil extrêmement efficace. Elles permettent en effet d'obtenir simultanément une qualité de compression supérieure aux méthodes précédentes et une compression progressive de l'image.

Le chapitre suivant présente les autres étapes de compression : quantification, codage est décodage.

A decorative graphic of a scroll with a black outline and a light gray fill. The scroll is partially unrolled, with the top and bottom edges showing a slight curve. The text is centered on the unrolled portion.

Chapitre 3: Codage et quantification

3.1 Introduction

La représentation numérique d'un image implique la quantification de chaque échantillon selon un nombre fini de valeurs discrètes. L'objectif visé est, soit une transmission, soit un traitement déterminé.

- Dans le premier cas, chaque échantillon du signal est quantifié, codé, puis transmis; à la réception, il est décodé, puis converti en amplitude continue.
- Dans le second cas, la loi de quantification est imposée par le système de traitement ; une contrainte importante pour un système de traitement numérique consiste à commettre des erreurs de calcul qui soient négligeables vis-à-vis de l'incertitude sur le signal lui-même.

3.2 Codage :

L'opération du codage entropique est parfaitement réversible c-à-d elle n'introduit aucune distorsion comme c'est le cas avec la quantification. Après la phase de quantification, les coefficients prennent des valeurs dans un ensemble fini de symboles $\{\hat{y}_i\}$. L'idée est de trouver une sorte de bijection de l'ensemble $\{\hat{y}_i\}$ dans un nouvel ensemble $\{\hat{z}_i\}$ de manière à minimiser le nombre moyen de bits/coefficient [34].

3.2.1 Algorithme de Huffman :

Dans l'algorithme de Huffman, on procède différemment pour un résultat comparable. On commence par choisir les deux symboles qui ont le moins d'occurrences, on leur donne les codes 0 et 1 et on les regroupe dans un arbre binaire auquel on attribue un nombre d'occurrences : la somme des deux symboles qu'il regroupe. Ensuite, on continue : on choisit deux symboles ou arbres qui ont le moins d'occurrences et on les regroupe. On regroupe ainsi deux par deux des symboles ou des arbres en fonction du nombre d'occurrences qu'ils représentent [36].

✓ Exemple de codage Huffman :

Soit le signal $S = \text{"CITRONTRESCONTRIT"}$.

On commence par l'ensemble : $\{ C/2, I/2, T/4, R/3, O/2, N/2, E/1, S/1 \}$.

L'algorithme groupe les deux symboles ou arbres les moins représentés : $E/1$ et $S/1$ et crée un arbre pour eux : $\{ E, S \}/2$. Dans cette notation, on met la branche 0 à gauche, la branche 1 à droite.

On continue donc avec $\{ C/2, I/2, T/4, R/3, O/2, N/2, \{E, S\}/2 \}$. On a le choix pour grouper les moins représentés : le C et le I, le O avec le ES, etc. Le premier choix donne : $\{ \{C, I\}/4, T/4, R/3, O/2, N/2, \{E, S\}/2 \}$.

On continue et on obtient $\{ \{C, I\}/4, T/4, R/3, \{O, N\}/4, \{E, S\}/2 \}$.

Un nouveau groupement du R et ES donne $\{ \{C, I\}/4, T/4, \{R, \{E, S\}\}/5, \{O, N\}/4 \}$.

On continue : $\{ \{ \{C, I\}, T \}/8, \{R, \{E, S\}\}/5, \{O, N\}/4 \}$

Encore : $\{ \{ \{C, I\}, T \}/8, \{ \{R, \{E, S\}\}, \{O, N\} \}/9 \}$.

Et la dernière étape car il ne reste plus que deux sous-groupes : $\{ \{ \{C, I\}, T \}, \{ \{R, \{E, S\}\}, \{O, N\} \} \}/17$.

A lire ainsi :

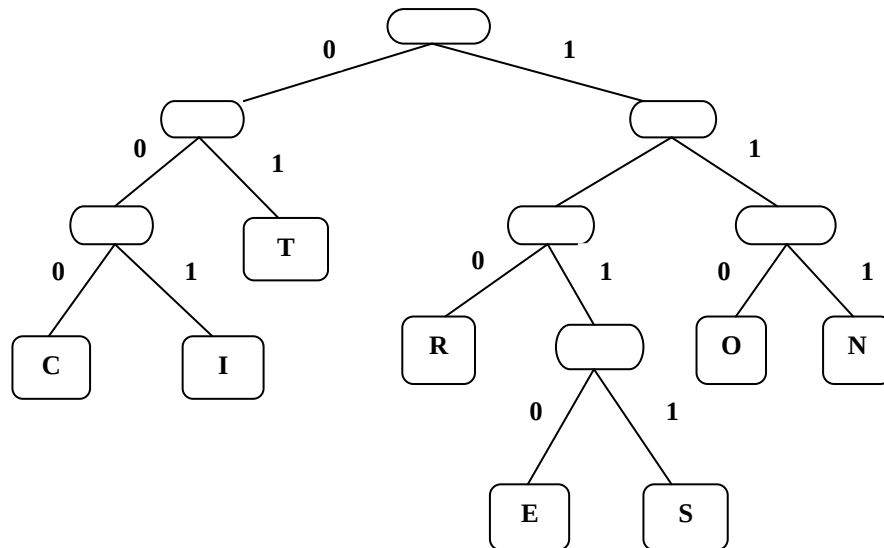


Figure 3.1 : Exemple de codage Huffman [36].

On obtient donc les codes suivants :

$C = 000, I = 101, T = 01, R = 100, O = 110, N = 111, E = 1010, S = 1011$.

Le signal S est codé = "0001010110011011101100101010110001101110110010101" sur 49 bits [36].

3.2.2 Codage Arithmétique (Entropique):

Principes de base :

Contrairement à l'algorithme précédent, le code est associé à la séquence et non chaque symbole pris individuellement.

On associe à chaque symbole à coder un intervalle $[a_k, b_k[$ de $[0,1[$

On itère ce processus jusqu'à ce que toute la séquence soit traitée

La séquence est alors codée par un réel de l'intervalle $[0,1[$

Méthode : Algorithme de codage :

Cet algorithme comporte 5 étapes successives :

1) On initialise l'intervalle de codage $[a_c, b_c[$ avec les valeurs $a_c=0$ et $b_c=1$, cet intervalle a une largeur $L=b_c-a_c=1$

2) cet intervalle est partitionné en N sous-intervalles (N nombre de symboles de l'alphabet de la source) proportionnellement aux probabilités $p(S_k)$ de chaque symbole S_k . cette partition est constituée de sous-intervalles $[a_k, b_k[$ tels que :

$$b_k - a_k = p(s_k) \text{ avec } a_k = a_c + \text{largeur} \times \sum_{i=1}^{k-1} P(S_i) \text{ et } b_k = a_c + \text{largeur} \times \sum_{i=1}^k P(S_i)$$

3) On choisit le sous-intervalle correspondant au prochain s_k à coder dans la séquence et on met à jour les valeurs a_c et b_c de la manière suivante :

$$a_c = a_c + \text{largeur} \times a_k \text{ et } b_c = a_c + \text{largeur} \times b_k$$

4) Avec le nouvel intervalle $[a_c, b_c[$ on recommence le processus de l'étape 2.

5) Les étapes 2,3 et 4 sont répétées jusqu'à épuisement des symboles de la séquence et obtention du dernier intervalle $[a_c, b_c[$

La représentation binaire de tout réel X_c de l'intervalle $[a_c, b_c[$ est un code de la séquence[37].

Exemple : Codage arithmétique d'une source .

On considère la source $S = \{a, b, c, d, e\}$ avec les probabilités respectives d'occurrence des symboles suivantes :

$$P(a)=0.3, P(b)=0.25, P(c)=0.20, P(d)=0.15, P(e)=0.1$$

On souhaite coder la séquence bdcea

Pour coder cette séquence on divise l'intervalle $[0,1[$ en 5 sous-intervalles, puis on se place sur le sous-intervalle correspondant au premier symbole de la séquence à coder, il s'agit du

symbole "b". Pour le symbole suivant de la séquence "d" on subdivise le sous intervalle de b, $[0.3, 0.55[$ en 5 sous-intervalles correspondant au nombre de symboles de l'alphabet de la source S.

On procède ainsi récursivement pour toute la séquence (voir figure ci-dessous).

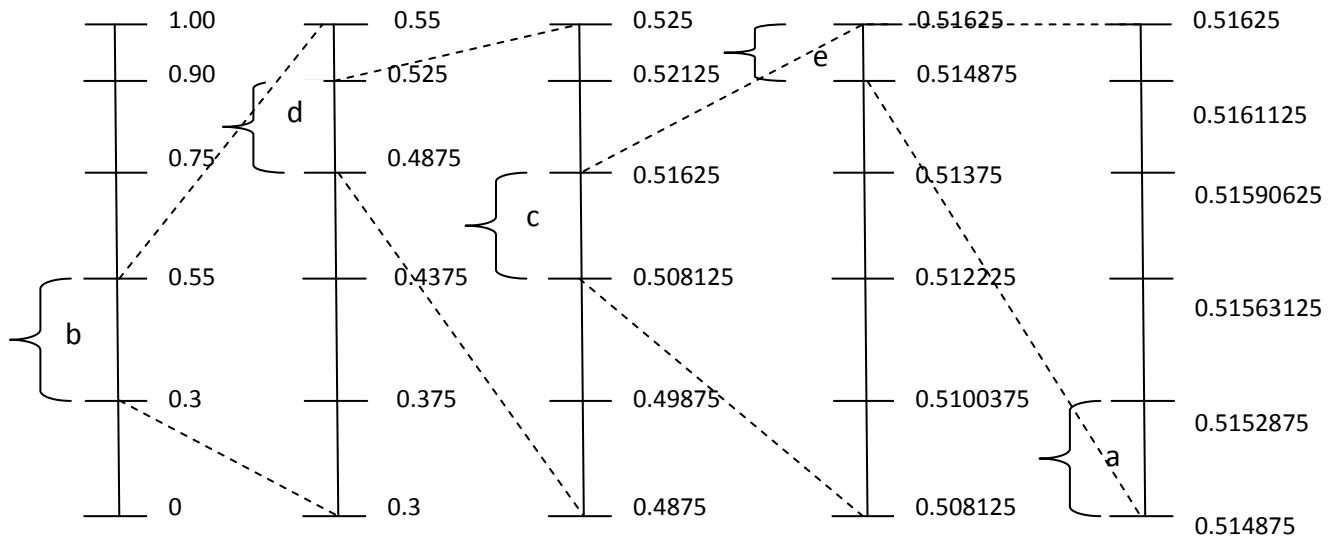


Figure 3.2 : Étapes de codage arithmétique [37].

Méthode : Algorithme de décodage.

Cet algorithme comporte six étapes successives :

- 1) On initialise $a_c=0$ et $b_c=1$
- 2) On calcule la largeur du sous-intervalle du code : $\text{largeur} = b_c - a_c$
- 3) On trouve le sous-intervalle $[a_k, b_k[$ du symbole S_k avec $1 \leq k \leq N$ tel que :

$$a_k \leq \frac{(x_c - a_c)}{\text{largeur}} < b_k$$

On rappelle que X_c est le réel codant la séquence.

- 4) On obtient le symbole S_k
- 5) On met à jour le sous-intervalle de codage : $a_c = a_c + \text{largeur} \times a_k$ et $b_c = a_c + \text{largeur} \times b_k$
- 6) On répète les 2,3,4 et 5 jusqu'à obtenir le décodage de tous les symboles de la séquence.

Exemple : Décodage de l'exemple précédent.

On applique l'algorithme de décodage à l'exemple précédent :

On considère la valeur $x_c=0.51508125$ codant la séquence

Etape 1 :

On initialise $a_c=0$ et $b_c=1$

Etape 2 :

On calcule la largeur du sous-intervalle du code : $\text{largeur} = b_c - a_c = 1$

Etape 3 :

On calcule le nombre $\frac{(x_c - a_c)}{\text{largeur}}$ dont la valeur est 0.51508125 et on cherche k tel que ce nombre soit compris dans la partition initiale.

Etape 4 :

$k=2$, Il s'agit du sous-intervalle $[0.3, 0.55[$ qui correspond au symbole b .

Etape 5 :

On met à jour le sous-intervalle de codage $a_c = a_c + \text{largeur} \times a_k$ et $b_c = a_c + \text{largeur} \times b_k$

$$a_c = 0 + 1 \times 0.3 = 0.3$$

$$b_c = 0 + 1 \times 0.55 = 0.55$$

On répète l'étape 2 : $\text{largeur} = 0.55 - 0.3 = 0.25$

$$\text{Etape 3 : } (0.51508125 - 0.3) / 0.25 = 0.860325$$

Etape 4 : $k=4$, il s'agit du sous-intervalle $[0.75, 0.90[$ qui correspond au symbole d .

On revient à l'étape 5 et ainsi de suite ... [37].

3.3 quantification :

Dans le schéma de compression, l'étape de quantification est celle qui dégrade de manière irréversible le signal. Elle est cependant d'une importance capitale dans la réduction du débit binaire. La quantification est une opération qui transforme l'image d'entrée à l'aide d'un ensemble de valeurs appelé dictionnaire.

3.3.1 Caractéristique d'un quantificateur :

- Un quantificateur prend un signal u en entrée et rend un signal quantifié v .
- La qualité du signal quantifié v dépend du pas de quantification noté généralement Δ .
- Si le pas est constant, on parle de quantification uniforme. Si le pas est variable Alors la quantification non uniforme.

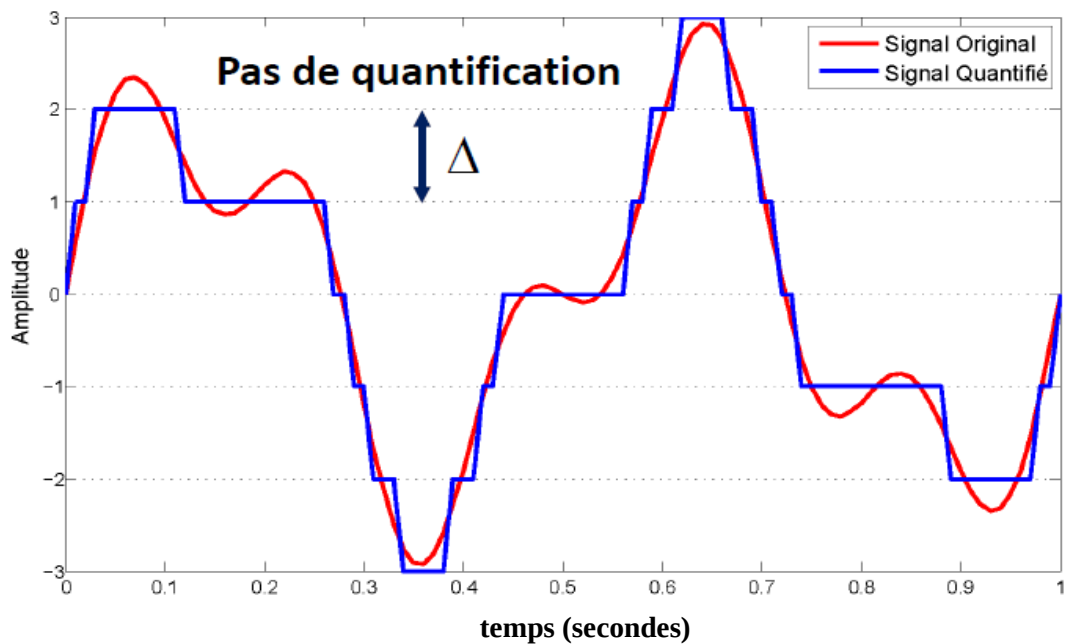


Figure 3.3 : Caractéristique d'un quantificateur [41].

3.3.2 Quantification scalaire :

La quantification scalaire [38] (Scalar Quantization) est la forme la plus couramment utilisée en compression d'images, c'est une procédure qui associe à une variable continue X une variable discrète x . Pour cela on associe à x la valeur quantifiée $x_q = Q(X)$, où Q est une fonction (non linéaire) de quantification. Il s'agit d'une entrée divisée en une famille finie d'intervalles $p_i = [x_{i-1}, x_i[$ et la valeur de sortie y_i est choisie dans l'intervalle P_i . Le codage consiste à trouver l'intervalle auquel appartient une valeur d'entrée $x(n)$ et à lui associer le numéro de cet intervalle $i(n) \in 1, \dots, L$ qui sera transmis ou stocké.

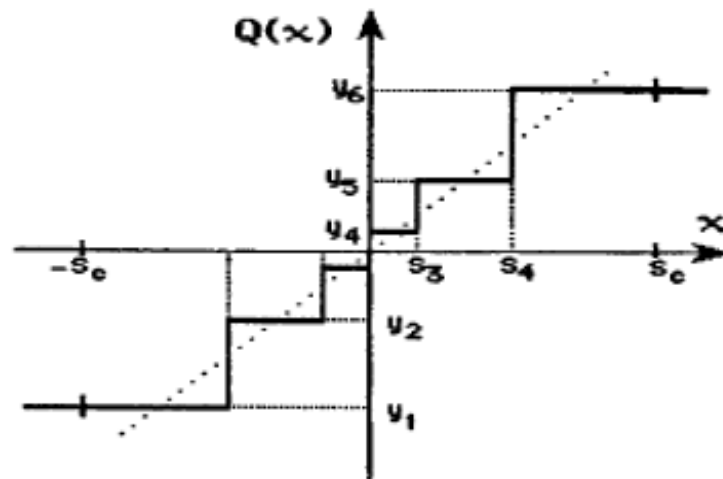


Figure 3.4 : Caractéristique typique d'un quantificateur scalaire de $N = 6$ niveaux [39].

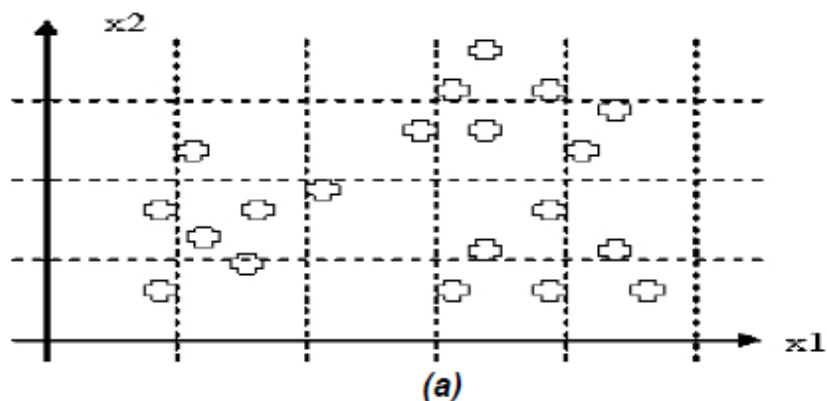
Il existe plusieurs formes de quantifications scalaires dont :

o **Quantification scalaire uniforme (linéaire) :**

La quantification est dite uniforme (ou parfois «linéaire») si la largeur de tous les intervalles de quantification est la même, Ceci implique que les niveaux de quantification sont équidistants.

Pour définir un quantificateur scalaire uniforme, il faut :

1. partitionner l'intervalle $[a, b]$ en $L = 2^R$ intervalles distincts de même longueur $q = (b - a) / 2^R$.
2. associer un numéro à chaque intervalle "i"
3. définir un représentant par intervalle "y_i", par exemple le milieu de l'intervalle.



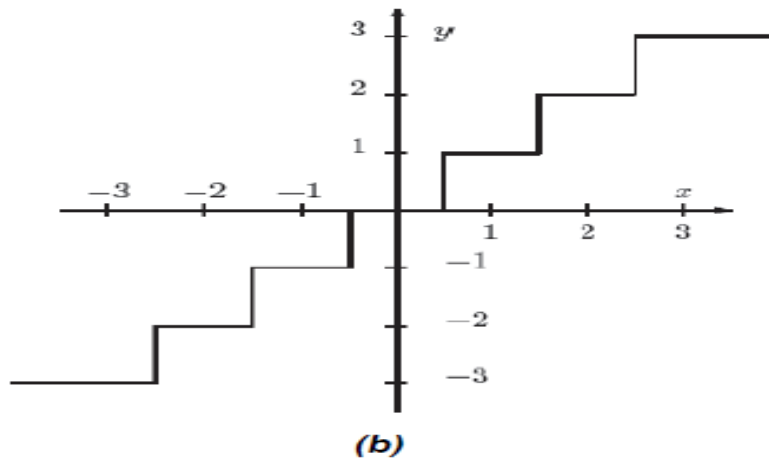


Figure 3.5 : (a) et (b) quantification scalaire uniforme [40].

La quantification scalaire jouit d'une simplicité qui permet des traitements temps réel dans des applications difficiles comme les radars ou la télévision [40].

❖ **Caractéristique d'un quantificateur uniforme :**

La quantificateur uniforme est Caractérisé par la Courbe Entrée-Sortie $v = Q(u)$ (figure 3.6)

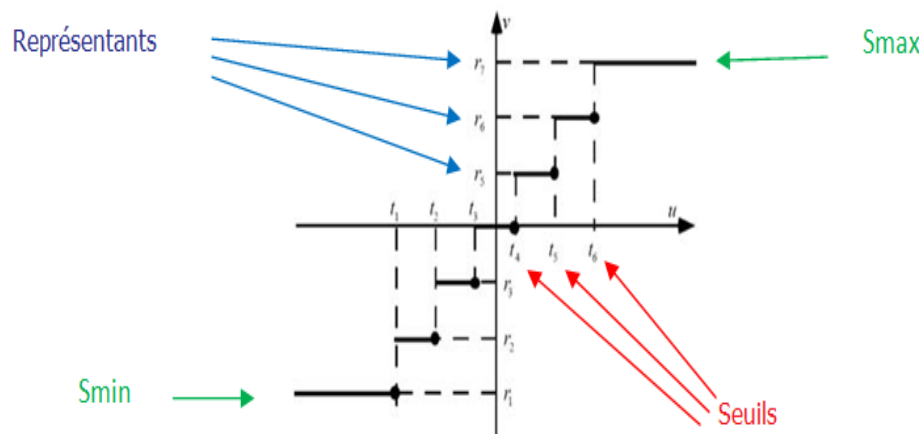


Figure 3.6 : la Courbe Entrée-Sortie d'un quantificateur uniforme [41].

Erreur de quantification $\varepsilon = u - v$

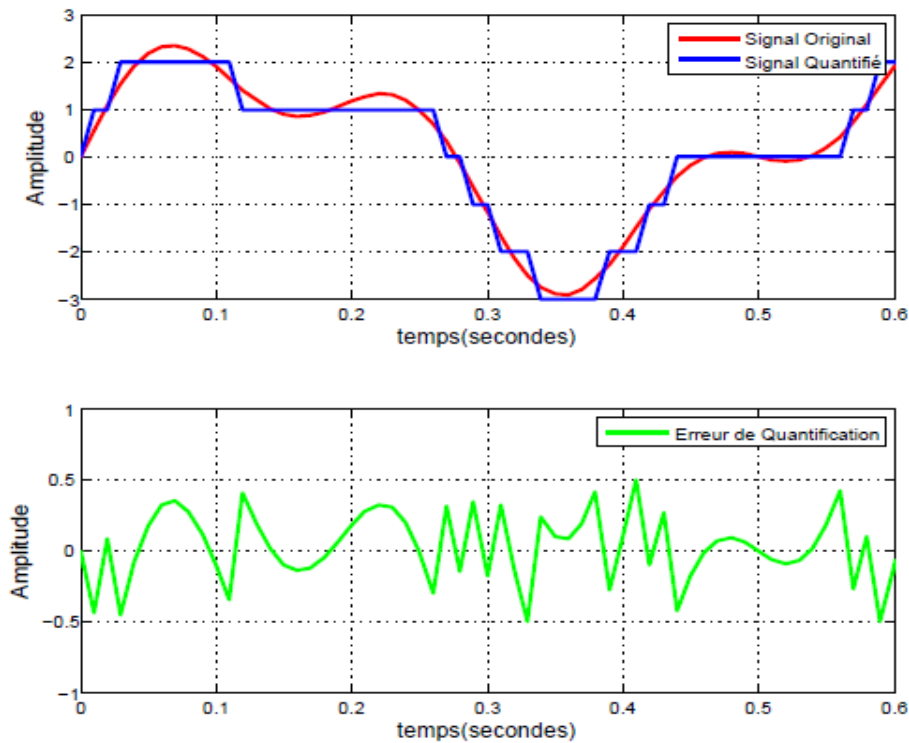


Figure 3.7 : Erreur de quantification [41].

❖ **Avantage d'un quantificateur uniforme :**

- ✓ ne dépend que du paramètre $\Delta = (t_{k+1} - t_k) = (r_{k+1} - r_k)$
- ✓ les représentants sont au milieu de chaque cellule,
- ✓ Le nombre de bits N_b pour coder les représentants est donné par

$$N_b = \log_2(Nb_{\text{rep}}) \text{ (en bits)}$$

Nb_{rep} : nombre de représentants

❖ **Inconvénients d'un quantificateur uniforme :**

L'erreur de quantification ne dépend pas de l'amplitude du signal d'entrée.

❖ **Exemple 1 :** quantification de la voix en téléphonie.

1. La différence entre puissances du signal original et puissance du bruit de quantification (SNR) évalue la qualité du signal perçu par l'interlocuteur.

Le SNR est proportionnel à la puissance du signal émis . Donc on a Conclu que le quantificateur uniforme est inadaptée en téléphonie .

2. Elle est efficace pour les signaux ayant des distributions uniformes. Or ce n'est pas le cas pour les signaux multimédia.

❖ **Exemple 2:** quantification d'une image en noir et blanc.

- L'erreur de quantification est relative à Δ .
- La valeur des pixels dépend de la photo.

Le SNR serait meilleur si le pas de quantification était variable [41].

○ **Quantification Scalair non uniforme (non linéaire) :**

La quantification non uniforme est motivée par le constat qu'un quantificateur uniforme n'est pas optimal que dans le cas où le signal d'entrée a une distribution uniforme, ce qui n'est pas le cas pour la plupart des signaux, elles suivent des distributions différentes et des fois aléatoires : Les signaux de parole ont une densité de probabilité proche d'une loi de Laplace, tandis que les signaux de musique ont souvent une distribution gaussienne. La distribution des valeurs des pixels d'une image peut beaucoup varier d'une image à l'autre, et elle est rarement uniforme. Ainsi, certaines valeurs de quantification correspondant à des échantillons peu probables sont gaspillées utilisées, ce qui représente un gaspillage de bits.

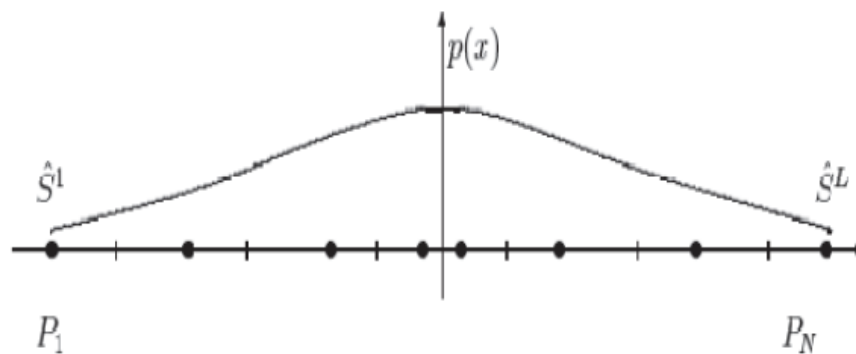


Figure 3.8: quantification scalaire non uniforme [40].

❖ **Caractéristique d'un Quantificateur Non Uniforme :**

Un quantificateur non uniforme (QNU) a un pas de quantification de taille variable.

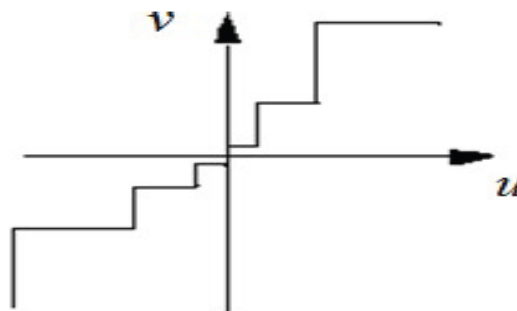


Figure 3.9 : pas de quantification de taille variable [41].

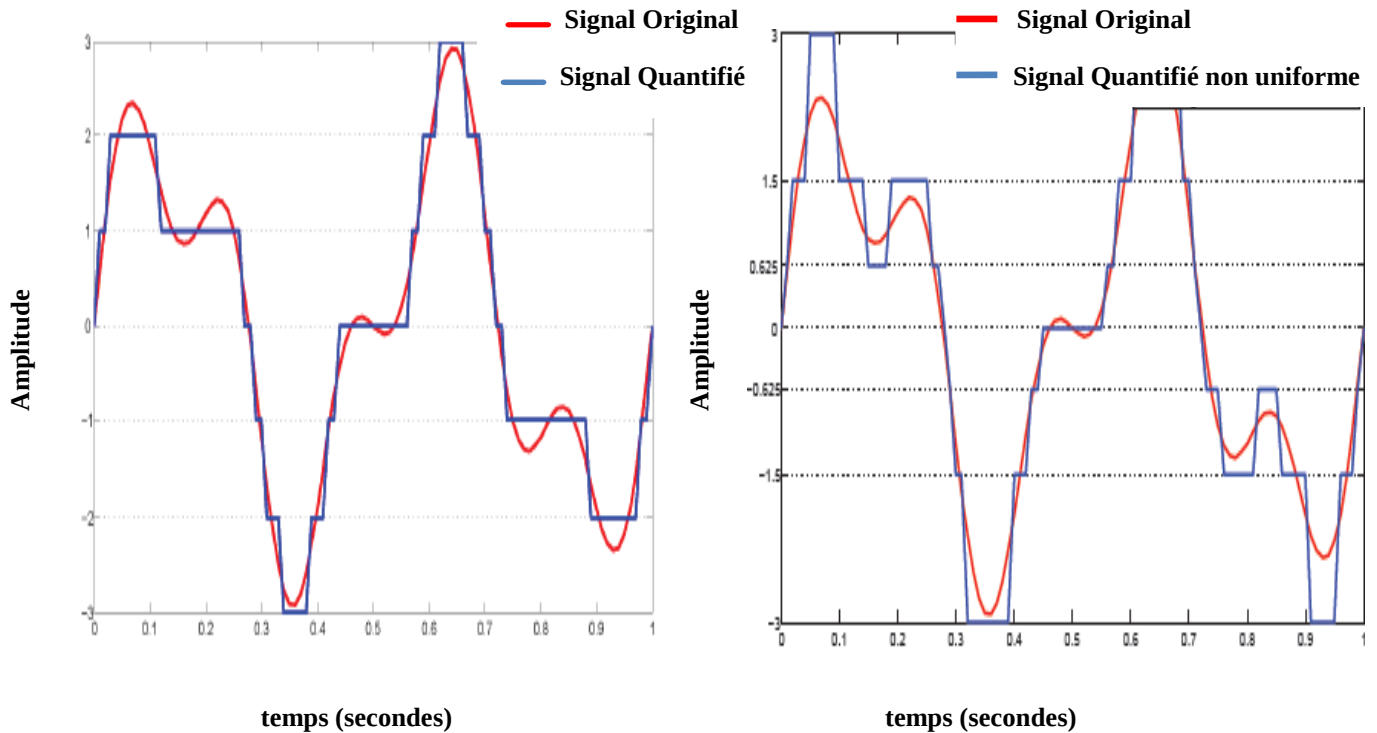


Figure: 3.10 : Quantification uniforme (à gauche) et non uniforme (à droite) [41].

❖ **L'algorithme Loi μ (ou Loi Mu, en anglais μ -law ou mu-law) :**

L'algorithme Loi μ est une forme de quantification par compression logarithmique, elle été introduit pour la première fois dans l'industrie téléphonique au Japon et les États-Unis (le standard G.711) pour les données audio. Elle exploite le fait que le système de perception auditive humaine est un processus logarithmique, dans laquelle les signaux sonores de hautes amplitudes ne nécessitent pas la même résolution que celles de faibles amplitudes.

L'oreille humaine est plus sensible au bruit de quantification dans les petits signaux que dans les signaux forts. « μ -law » applique une fonction de quantification logarithmique $F(x)$ pour ajuster la résolution de données en proportion du niveau du signal d'entrée. Les petits signaux sont représentés avec plus de précision que les grands signaux.

La fonction $F(x)$ de compression μ -law est définie par l'équation (3.1), où μ est le coefficient de compression ($\mu = 255$ aux États-Unis et le Japon) et x le vecteur de données à quantifier [42].

$$F(x) = \frac{\text{sign}(x) \times \ln(1 + \mu|x|)}{\ln(1 + \mu)} \quad 0 \leq |x| \leq 1 \quad (3.1)$$

La Figure 3.10 présente cette fonction pour les trois valeurs de $\mu = 1, 10$, et 255. Plus la valeur de μ est grande plus les valeurs de faibles amplitudes sont amplifiées.

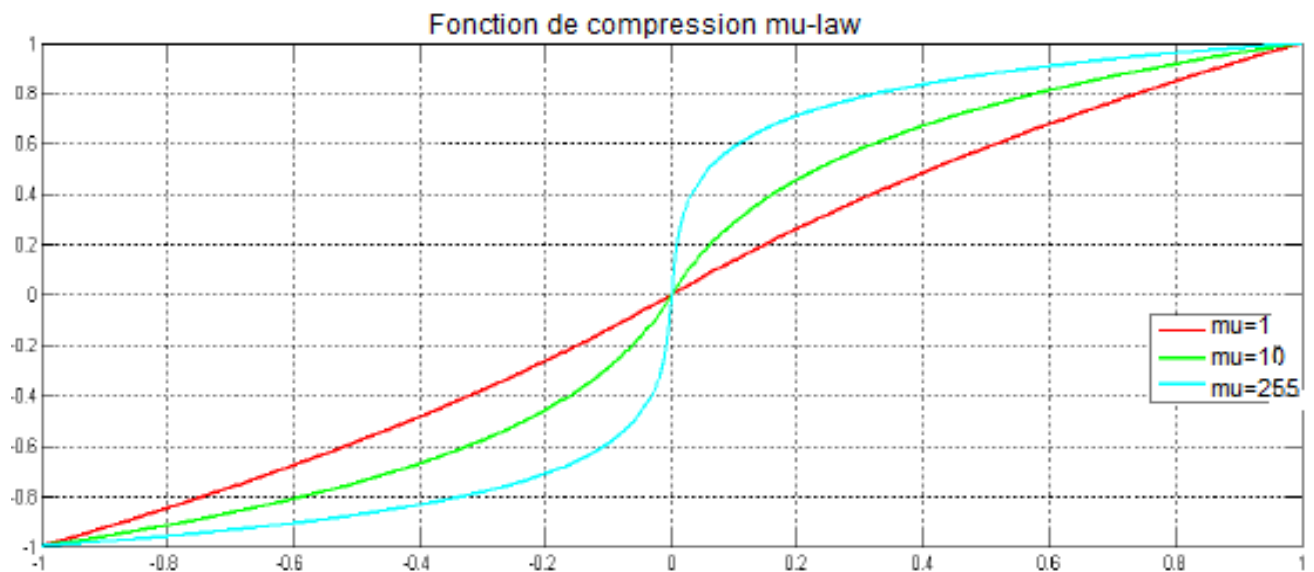


Figure 3.11 : Fonction de compression M-law ($\mu = 1, 10, 255$) [42].

Le décodage des données codées n'est ensuite qu'une question d'inverser les étapes de l'encodage. Les étapes de quantification par compression μ -law peuvent être représentées par le schéma de la Figure 3.11.

Il est clair que les grandes valeurs de μ provoquent une quantification grossière pour les grandes amplitudes et plus précis pour les faibles amplitudes.

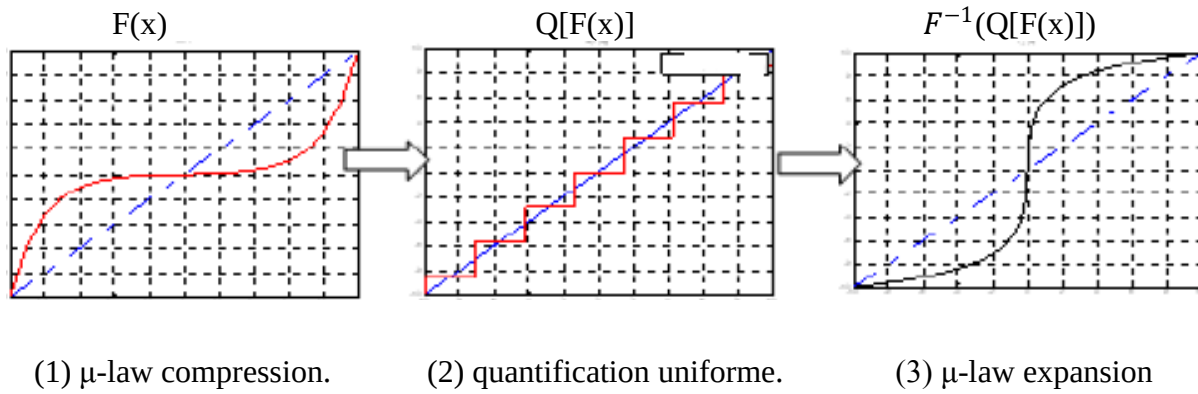
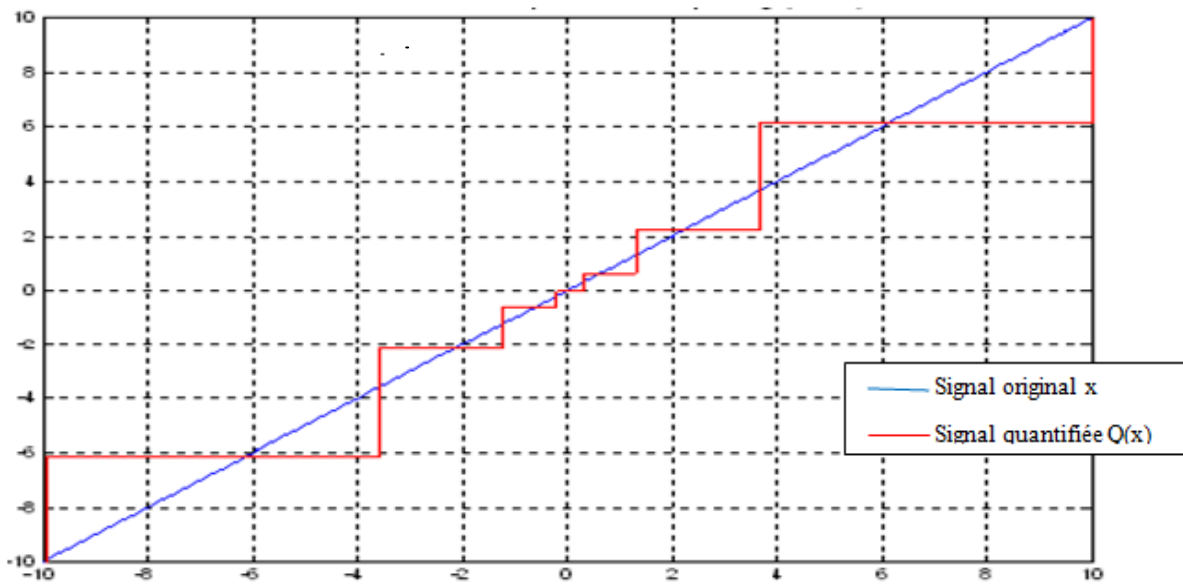


Figure (a)



Quantification à 8 niveaux par mu-law companding

Figure (b)

Figure 3.12: (a)-Etapas de quantification par Companding. (b)-quantification μ -law à 8 niveaux. ($\mu=25$) [42].

3.3.2 Quantification vectorielle :

La quantification vectorielle (Vector Quantization) est une opération irréversible développée par Gersho et Gray [43], Elle n'est qu'une généralisation de la quantification scalaire où son sens le plus général est l'approximation d'un signal d'amplitude continue par un signal d'amplitude discrète. la QV consiste à remplacer des vecteurs de l'image (bloc de pixels) par des vecteurs appartenant à un ensemble fini appelé dictionnaire, ce dernier est caractérisé par deux paramètres importants :

- Nombre de vecteurs représentants.
- Dimension des vecteurs représentants.

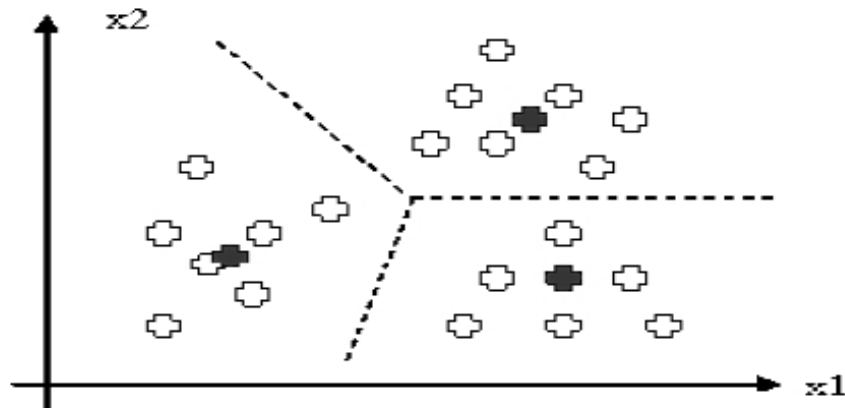


Figure 3.13 : quantification vectorielle (construire un dictionnaire) [40]..

Le principe de la QV est d'associer par un quantificateur vectoriel à chaque vecteur d'entrée $X_i = (x_j, j=1 \dots k)$ un vecteur $Y_i = (y_j, j=1 \dots k) = Q(X_i)$, ce vecteur Y_i étant choisi parmi un dictionnaire (*codebook*) de taille finie .

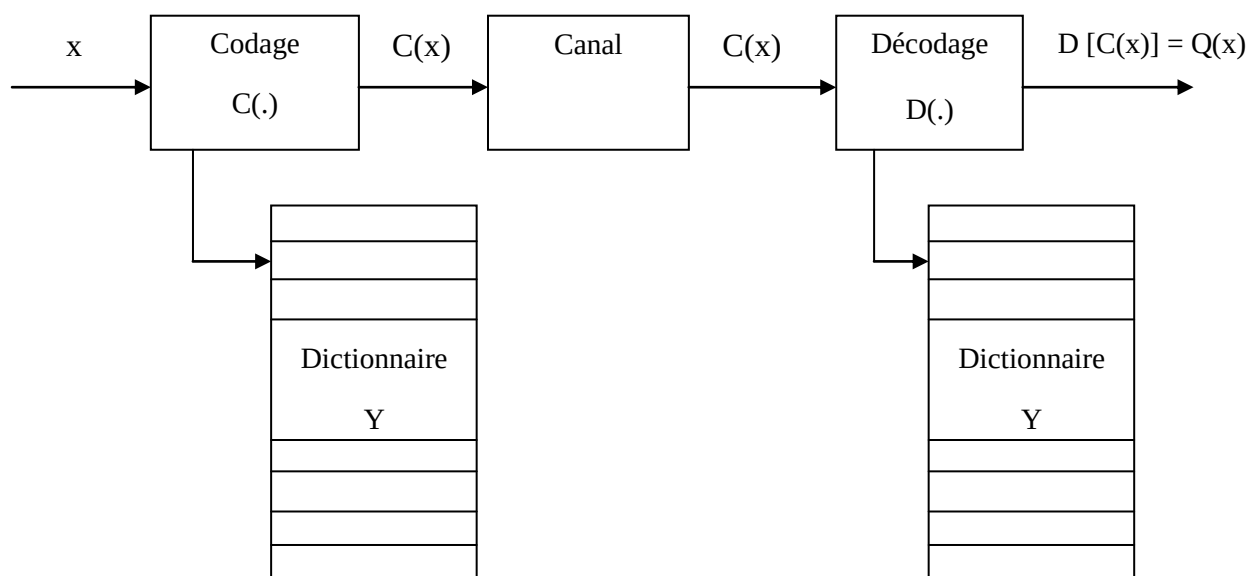


Figure 3.14 : schéma d'un quantificateur vectoriel [44].

Si le codage par QV est simple, la conception d'un dictionnaire est plus compliquée et a donné lieu à de nombreux algorithmes. Un des plus utilisés l'algorithme LBG (Linde, Buzo et Gray) ou algorithme de Lloyd généralisé. Le succès d'un quantificateur vectoriel dépend de son dictionnaire, si le codebook contient des blocs qui sont bien représentatifs des images ,la qualité de la compression sera bonne , inversement [45].

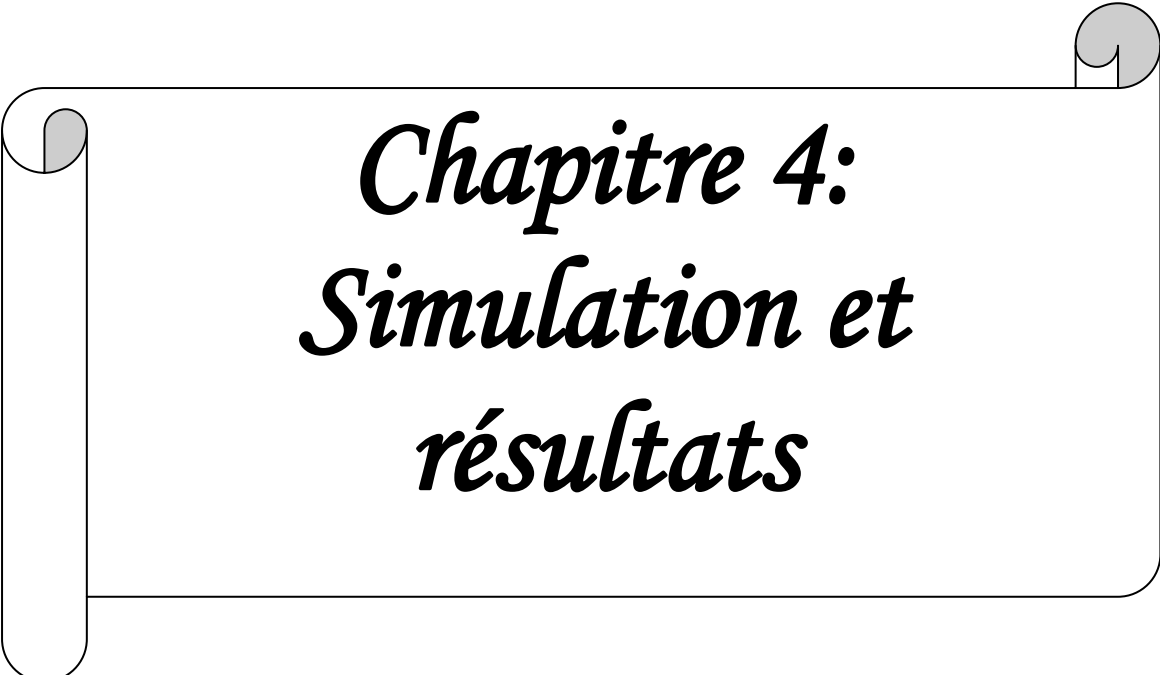
L'inconvénient de la quantification vectorielle est sa complexité, qui limite la taille des vecteurs utilisés. Pour contourner ce problème le dictionnaire est structuré de façon à

simplifier la recherche du représentant, Ceci est réalisable avec la quantification vectorielle arborescente ou d'utiliser des transformations linéaires (telle que la décomposition en sous - bandes et les transformées en ondelettes) et d'appliquer la quantification vectorielle sur les coefficients [46].

3.4 Conclusion :

Le processus de quantification est le principal responsable de la dégradation de l'image. C'est lors de cette étape que l'image va être réellement compressée. L'utilisateur choisit au début la qualité de la compression.

On a présenté dans ce chapitre les différents types de quantification et codage , la quantification scalaire est plus utile que la quantification vectorielle si elle est précédée par une transformation qui génère des coefficients fortement dé corrélés.



Chapitre 4: Simulation et résultats

4.1 introduction :

Dans ce chapitre, nous appliquons l'algorithme de transformée en bandelettes décrite dans chapitre 2 ainsi que la quantification et codage proposé dans le chapitre 3 à la compression d'images.

Lorsque la compression avec pertes éliminer quelques informations en gardant la meilleure PSNR possible, pour obtenir le meilleur résultat en parallèle de la variation du seuil T et pas de quantification pq .

4.2 Algorithme de compression :

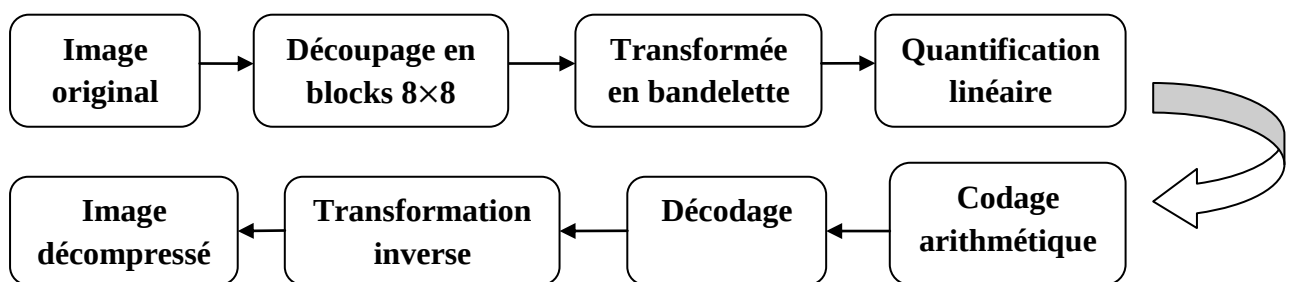


Figure 4.1 : Algorithme de compression

☒ Les fonctions de base utilisées dans l'algorithme de compression:

```

clear all, close all, clc;
T = 100;
options.use_single_qt = 0;
n = 8;
s = Inf;
M1 = double(imread('baboon.bmp'));
MW = perform_wavelet_transform(M1, log2(n), 1);
  
```

Ici, nous changeons la valeur de seuil

Ici nous choisissons l'image que nous voulons compresser

```

Jmin = 2;
Jmax = log2(n) - 1;
j_min = 2;
j_max = 4;
  
```

```

psnr_band = [];
bit_band = [];
tauxdec = [];
Mbt = [];
  
```

```

disp('Computing quadtree.');
```

```

[QT, Theta] = compute_wavelet_quadtree(MW, Jmin, T, j_min, j_max, s, options);
  
```

```

% direct transform
disp('Computing forward bandelet transform. ');
[MB,Rbandg] =
perform_wavelet_bandelet_transform(Ml,Jmin,QT,Theta,1,
options);

MBt =reshape(MB,1,n*n);
MBtm=min(MBt);
MBtM=max(MBt);
pq=10;
part = floor(MBtm):floor(MBtM);
part = ceil(MBtm):pq:ceil(MBtM);
code = floor(MBtm):floor(MBtM)+1 ;
code = ceil(MBtm):pq:ceil(MBtM)+pq ;
[ indx, quant]=quantiz(MBt, part, code);

perform_wavelet_bandelet_transform(musig1,Jmin,QT,Theta,-1,
options);

quant1 = reshape(quant,n,n);

% partie codage
disp('arith code sig');

[Store_Byte,Counts,Table]=Arith_Code(quant1);
disp('arith code Q');

[Store_ByteQ,CountsQ,TableQ]=Arith_Code(QT);
disp('arith code Theta');

[Store_ByteT,CountsT,TableT]=Arith_Code(Theta);

%bandelet inverse

disp('Computing inverse bandelet transform. ');
[Mb,Rbandg] =
perform_wavelet_bandelet_transform(quant1,Jmin,QT,Theta,-1,
options);

disp 'taux de compression'
tauxdec = (length(Store_Byte)*8+Rbandg)/(n*n)*100;

disp 'psnr'
psnr_band =[psnr_band psnr_saad( Mb,Ml )];

Mbt = Mb;
sprintf('Le taux de compression est : %.5f', tauxdec)
sprintf('Le PSNR est : %.5f', psnr_band)

```

Ici, nous changeons la valeur de
pas de quantification

☒ Exemple d'application :

Une image 512×512



Découpage en blocks 8×8



Un Block 8×8

La matrice correspondante:

44	38	44	37	43	49	58	105
61	42	38	43	40	50	78	137
99	46	40	36	44	53	101	142
137	58	37	40	39	57	130	141
157	91	45	37	42	88	144	145
156	132	52	43	54	136	147	131
160	150	77	46	63	141	139	139
160	158	114	47	105	139	125	137

Après transformation :

699,875	-87,875	54,750	-132,750	-8,131	-35,250	-29,698	18,384
-141,067	-107,657	177,750	-86,000	-43,840	-42,750	-16,617	-6,625
-56,568	0	3,181	-54,800	-26,870	-52,000	48,083	-0,530
-30,052	-47,376	-41,719	12,020	35,001	-57,250	41,012	-4,125
13,000	-2,828	-11,000	-10,750	-3,535	-9,899	-2,121	-19,091
16,500	5,303	-10,000	4,065	-13,435	7,071	-4,949	1,414
-3,000	2,121	1,000	-18,000	-21,213	-7,071	-30,405	0,707
-16,000	7,778	16,000	-30,935	-8,485	2,121	4,242	1,414

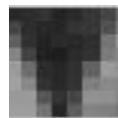
Après Quantification :

709	-81	59	-131	-1	-31	-21	19
-141	-101	179	-81	-41	-41	-11	-1
-51	9	9	-51	-21	-51	49	9
-21	-41	-41	19	39	-51	49	-1
19	-1	-1	-1	-1	-1	-1	-11
19	9	-1	9	-11	9	-1	9
-1	9	9	-11	-21	-1	-21	9
-11	9	19	-21	-1	9	9	9

Après Transformation inverse :

40,838	44,841	50,844	37,261	42,736	54,304	71,174	106,369
69,237	46,255	41,243	38,675	52,550	55,718	82,987	121,925
96,224	50,473	35,520	42,893	40,090	47,258	118,025	148,821
132,695	66,029	33,990	30,165	43,832	48,672	123,767	136,093
157,031	107,568	44,610	39,988	49,254	98,924	154,615	147,202
161,137	137,266	50,717	41,402	50,632	128,623	137,994	134,474
153,349	154,590	87,497	51,655	70,927	145,947	143,862	130,585
158,455	156,004	114,604	38,927	93,306	133,219	128,241	117,857

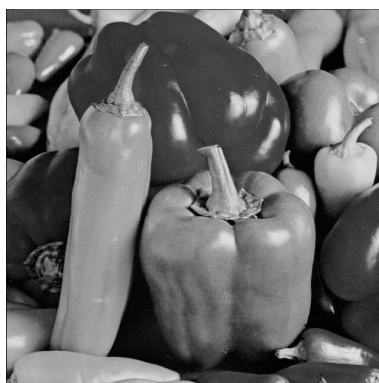
Image reconstruite :



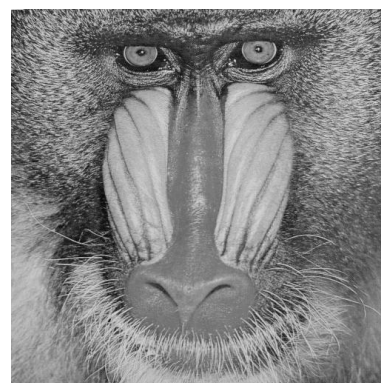
4.3 Présentation des résultats :

4.3.1 Images de test utilisées :

Deux images de test naturels aux niveaux de gris (512×512, 8 bits/pixel) sont utilisées pour montrer l'efficacité des transformées ainsi que les matrices de quantification proposées: Peppers.bmp et Baboon.bmp qui sont donnés dans la figure (4.2).



Peppers 512×512



Baboon 512×512

Figure 4.2 : Images de test utilisées

4.3.2 Etapes de travaille :

Pour appliquer cet algorithme, nous avons utilisé le programme MATLAB.

On doit fixer un paramètre et changer l'autre

- ✚ Dans le premier cas on a fixé le pas de quantification($pq = 10$) et varie le seuil et marquer parallèlement les valeurs de taux de compression et le PSNR.
- ✚ Dans le deuxième cas on a fixé le seuil ($T=100$) et varie le pas de quantification et marquer parallèlement les valeurs de taux de compression et le PSNR.
- ❖ Pour les deux cas on a Simuler les figures suivant:

Taux de compression (TC) en fonction de T ou pq et le PSNR (dB) en fonction de T ou pq

- ❖ Est ensuite on a Simuler quelques images pour défirent valeurs de T ou pq.

4.3.3 Résultat:

A. fixer le pas de quantification ($pq = 10$) et varie le seuil:

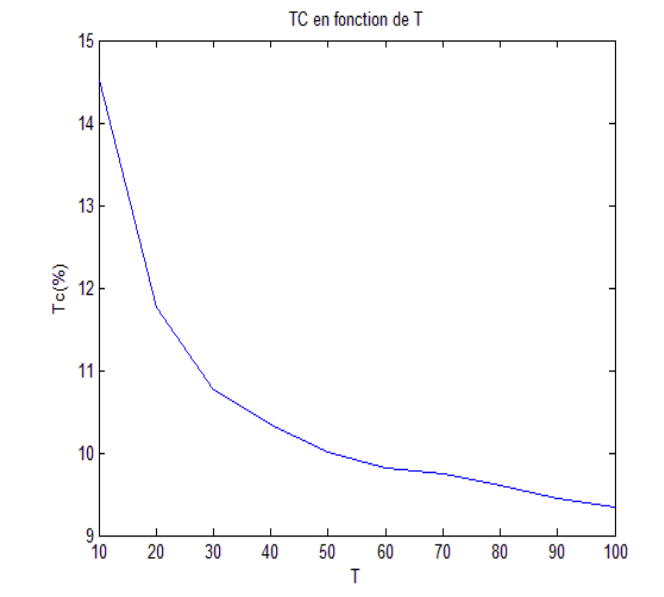
✓ **Résultat pour limage peppers :**

Le tableau suivante présente le PSNR et le TC de limage peppers si en a varie la seuil et fixer le pas de quantification .

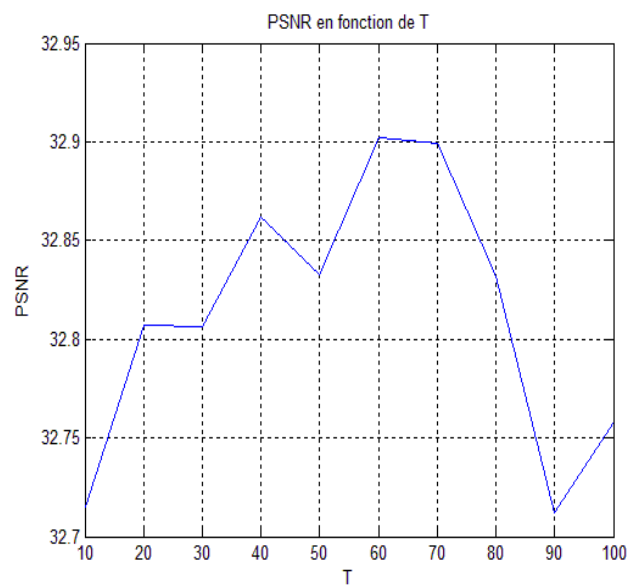
T	10	20	30	40	50	60	70	80	90	100
TC(%)	14.51	11.761	10.77	10.349	10.008	9.8198	9.7435	9.5993	9.4471	9.3468
PSNR(dB)	32.715	32.807	32.806	32.862	32.833	32.902	32.899	32.832	32.712	32.758

Tableaux 4.1 : Résultat de l'image Peppers (pq fixe et T varie)

❖ Les Figures :

**Figure 4.3:**

Evolution du TC en fonction T pour l'image peppers (pq fixe et T varie)

**Figure 4.4:**

Evolution du PSNR en fonction T pour l'image peppers (pq fixe et T varie)

❖ Les images :



Image original



Image pour T=10

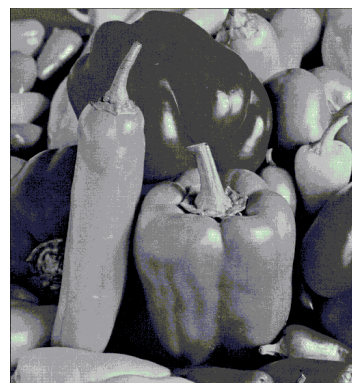


Image pour T=60

Figure 4.5: Résultat pour l'image Peppers (pq fixe et T varie)✓ **Interprétation des résultats :**

Après l'application de l'algorithme de bandelette à l'image peppers pour différentes valeurs de seuil on n'a remarqué que:

Pour la figure du taux de compression (TC) en fonction de le seuil (T) lorsqu'on augmente le seuil, le taux de compression diminue par ce que l'augmentation de seuil augmente L'information qui a été éliminer par la Fonctionnement de seuillage .

Pour la figure du PSNR en fonction de le seuil (T) le graph est presque constant ($32.715 \text{ dB} < \text{PSNR} < 32.902 \text{ dB}$) parce que L'information qui a été éliminer n'a pas d'impact significatif sur la valeur de PSNR .

Il n'a pas Une grande différence de La qualité des photos que nous avons affichée, Mais avec l'augmentions de la taille des images, Nous pouvons observer que la qualité de

l'image qui caractérisé par un PSNR = 32.902 dB mieux que l'image qui caractérisé par un PSNR = 32.715 dB.

✓ **Résultat pour l'image baboon :**

Le tableau suivante présente le PSNR et le TC de l'image baboon si on varie le seuil et fixe le pas de quantification .

T	10	20	30	40	50	60	70	80	90	100
TC(%)	17.528	16.232	15.018	13.966	13.2	12.475	11.891	11.46	10.942	10.633
PSNR(dB)	33.087	32.958	33.035	32.98	33.086	32.968	33.032	32.958	32.949	33.001

T	110	120	130	140
TC(%)	10.315	10.101	9.9228	9.7446
PSNR(dB)	32.922	32.992	32.985	33.008

Tableaux 4.2 : Résultat de l'image baboon (pq fixe et T varie)

❖ **Les Figures :**

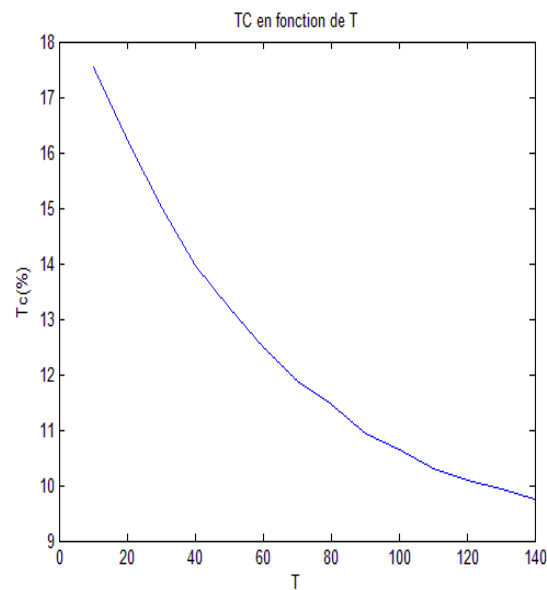
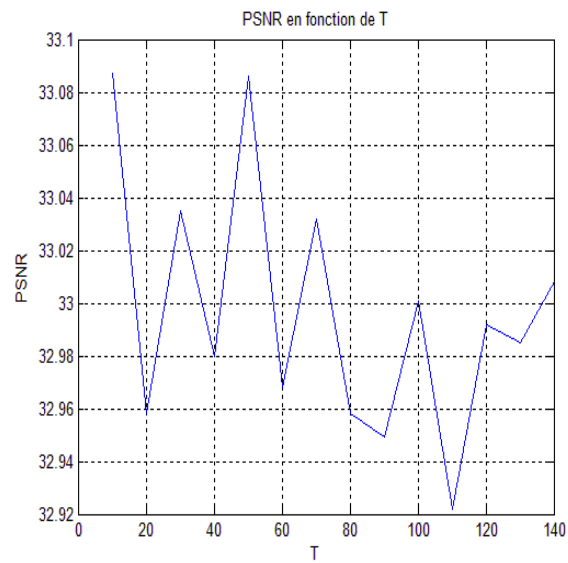


Figure 4.6:

Evolution du TC en fonction T pour l'image baboon (pq fixe et T varie)

**Figure 4.7:**

Evolution du PSNR en fonction T pour l'image baboon (pq fixe et T varie)

❖ Les images :

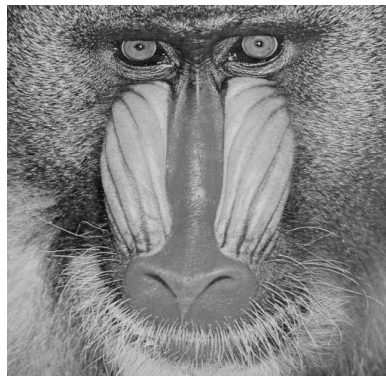


Image original



Image pour T=10

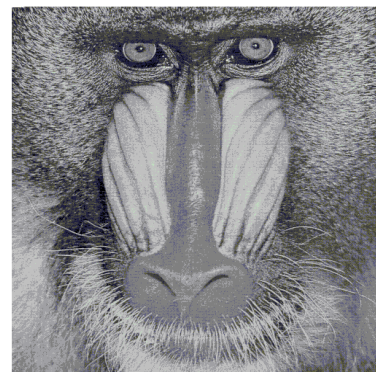


Image pour T=110

Figure 4.8: Résultat pour l'image baboon (pq fixe et T varie)

✓ **Interprétation des résultats :**

Après l'application de l'algorithme de bandelette à l'image baboon pour différentes valeurs de seuil on n'a remarqué que:

Pour la figure du taux de compression (TC) en fonction de le seuil (T) lorsqu'on augmente le seuil, le taux de compression diminue par ce que l'augmentation de seuil augmente L'information qui a été éliminer par la Fonctionnement de seuillage .

Pour la figure du PSNR en fonction de le seuil (T) le graph est presque constant ($32.922 \text{ dB} < \text{PSNR} < 33.087 \text{ dB}$) parce que L'information qui a été éliminer n'a pas d'impact significatif sur la valeur de PSNR .

Il n'a pas Une grande différence de La qualité des photos que nous avons affichée, Mais avec l'augmentions de la taille des images, Nous pouvons observer que la qualité de limage qui caractérisé par un PSNR = 33. 087 dB mieux que l'image qui caractérisé par un PSNR = 32.922 dB.

B . fixer le seuil (T=100) et varie le pas de quantification :

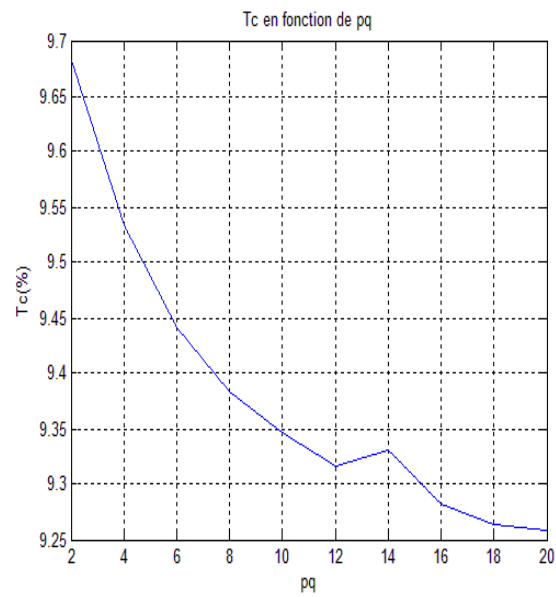
✓ **Résultat pour limage Peppers :**

Le tableau suivante présente le PSNR et le TC de limage peppers si en varie le pas de quantification et fixer la seuil.

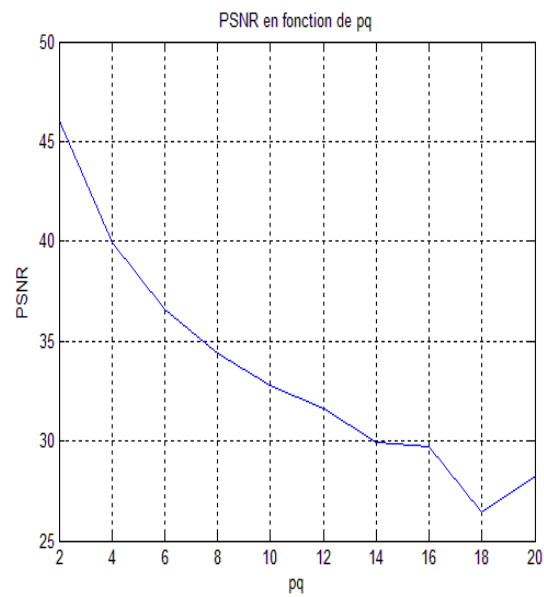
pq	2	4	6	8	10	12	14	16	18	20
TC(%)	9.6825	9.5329	9.4414	9.3834	9.3468	9.3163	9.3315	9.2827	9.2644	9.2583
PSNR (dB)	46.023	39.943	36.571	34.375	32.758	31.603	29.899	29.686	26.399	28.262

Tableaux 4.3 : Résultat de l'image peppers (T fixe et pq varie)

❖ Les Figures :

**Figure 4.9:**

Evolution du TC en fonction du pq pour l'image peppers (T fixe et pq varie)

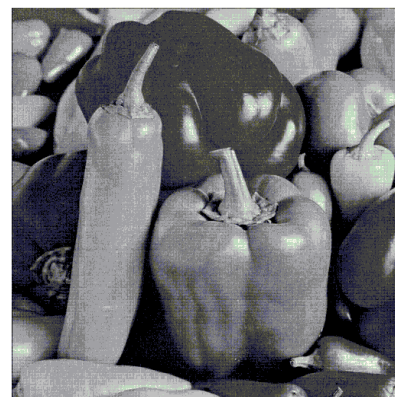
**Figure 4.10:**

Evolution du PSNR en fonction de pq pour l'image peppers (T fixe et pq varie)

❖ Les images :



Image original

Image pour $pq=2$ Image pour $pq=20$ **Figure 4.11:** Résultat pour l'image peppers (T fixe et pq varie)✓ **Interprétation des résultats :**

Après l'application de l'algorithme de bandelette à l'image peppers pour différentes valeurs de pas de quantification (pq) on n'a remarqué que:

Pour la figure du taux de compression (TC) en fonction du pas de quantification (pq) lorsqu'on augmente le pas de quantification, le taux de compression diminue par ce que l'augmentation de pas de quantification augmente L'information qui a été éliminer par la Fonctionnement de quantification .

Aussi pour la figure du PSNR en fonction de pas de quantification ,le graph est Décroissant parce que l'augmentation de pas de quantification augmente L'information qui a été éliminer .

Pour les deux photos que nous avons affichées, La qualité de photo qui caractérisé par un PSNR = 46. 023 dB mieux que l'image qui caractérisé par un PSNR = 28. 262 dB Parce que leur PSNR est plus grande .

✓ Résultat pour limage baboon :

Le tableau suivante présente le PSNR et le TC de limage baboon si en varie le pas de quantification et fixer la seuil.

pq	2	4	6	8	10	12	14	16	18	20
TC(%)	11.033	10.868	10.761	10.697	10.633	10.587	10.548	10.523	10.477	10.456
PSNR (dB)	46.92	40.828	37.504	34.986	33.001	31.544	30.084	29.327	28.068	27.595

Tableaux 4.4 : Résultat de l'image baboon (T fixe et pq varie)

❖ Les Figures :

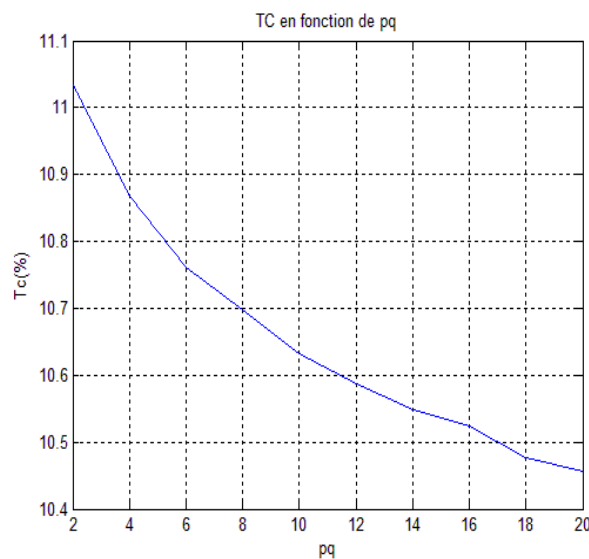
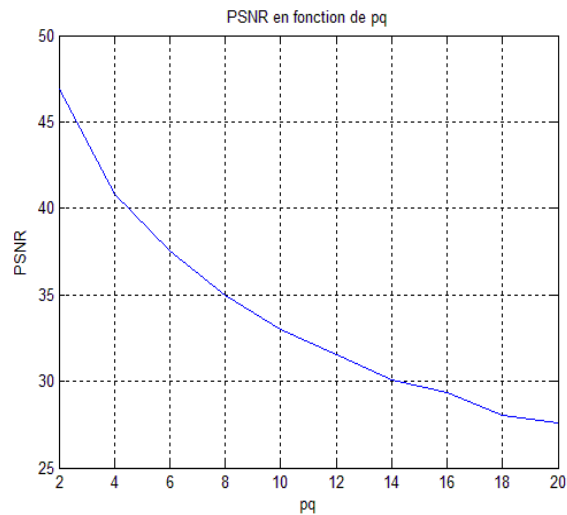


Figure 4.12:

Evolution du TC en fonction de pq pour limage baboon (T fixe et pq varie)

**Figure 4.13:**

Evolution du PSNR en fonction de pq pour l'image baboon (T fixe et pq varie)

❖ Les images :

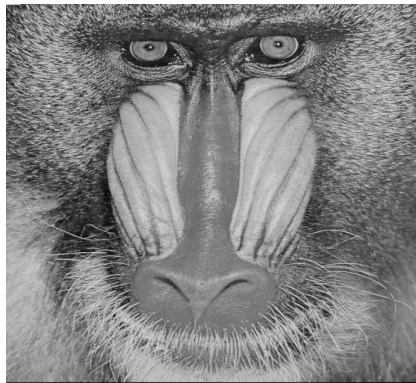


Image original

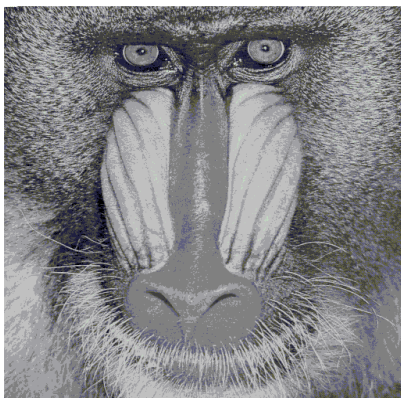


Image pour pq=2

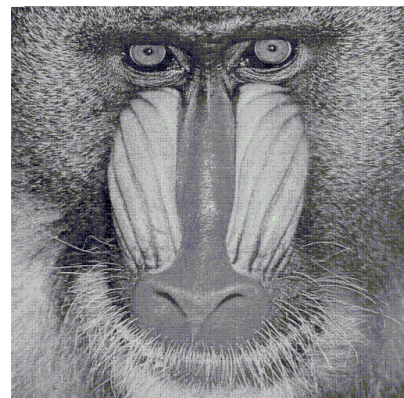


Image pour pq=20

Figure 4.14: Résultat pour l'image baboon (T fixe et pq varie)

✓ Interprétation des résultats :

Après l'application de l'algorithme de bandelette à l'image baboon pour différentes valeurs de pas de quantification (pq) on n'a remarqué que:

Pour la figure du taux de compression (TC) en fonction du pas de quantification (pq) lorsqu'on augmente le pas de quantification, le taux de compression diminue par ce que l'augmentation de pas de quantification augmente L'information qui a été éliminer par la Fonctionnement de quantification .

Aussi pour la figure du PSNR en fonction de pas de quantification ,le graph est Décroissant parce que l'augmentation de pas de quantification augmente L'information qui a été éliminer .

Pour les deux photos que nous avons affichées, La qualité de photo qui caractérisé par un PSNR = 46. 92 dB mieux que l'image qui caractérisé par un PSNR = 27. 595 dB Parce que leur PSNR est plus grande .

4.4 Conclusion :

Dans ce chapitre, nous avons appliqué un algorithme de compression d'images fixes basée sur une transformation en Bandelettes et une quantification scalaire uniforme avec un codage arithmétique.

Après l'analyse et discussions on peut conclure que :

- ✓ Les meilleurs résultats obtenus correspondent au valeurs de PSNR >30 db et TC <10 .
- ✓ L'augmentation de T et pq améliore le taux de compression mais diminue la qualité de l'images.
- ✓ Les propriétés de compression varient selon le contenu de l'image original .

Conclusion générale

La nécessité de compresser des données de plus en plus volumineuses et difficiles à transmettre ou à stocker est accrue avec le développement des techniques de communications. Les performances d'un système de compression sont évaluées par sa complexité calculatoire, son coût de stockage, le débit (ou le taux de compression) qu'il permet d'atteindre et la dégradation qu'il engendre sur les données compressées. Les techniques de compression ont fait l'objet de nombreuses recherches pour tenter d'optimiser ces différents aspects. Il demeure, cependant, difficile d'optimiser tous ces critères en même temps.

Le travail réalisé dans ce mémoire, a pour objectif l'application d'une méthode de représentation des images par la transformée de bandelette dans le domaine de compression d'image .

Les bases de bandelettes décomposent une image selon des vecteurs multi échelles allongés le long d'un flot géométrique indiquant des directions de régularité de l'image. L'optimisation de cette géométrie est faite par un algorithme rapide de meilleur base qui permet d'obtenir des résultats de compression.

La quantification est la phase la plus importante dans le schéma de compression avec perte, La quantification scalaire est plus utile que la quantification vectorielle, si elle est précédée par une transformation qui génère des coefficients fortement dé-corrélés. Elle peut être plus performante en adaptant le dictionnaire de quantification aux caractéristiques de signal à traité, ainsi on obtient des coefficients statistiquement plus redondants.

Les codeurs entropiques sont souvent utilisés dans la compression d'image, comme le codage de Huffman et le codage arithmétique. Pourtant ce dernier semble plus compliqué et plus lent, il fera toujours un meilleur ou aussi bon travail que le codage de Huffman en terme de taux de compression.

Dans ce mémoire nous avons étudié d'une façon générique les étapes de compression d'une image en bandelettes tout en tenant en considération les principales contraintes suivantes :

- avoir un taux de compression le plus petit que possible.
- avoir un PSNR le plus grand que possible.

Nous avons constater que plus T augmente plus TC diminue et PSNR diminue. Un suivi du ce travail doit avoir lieu pour mener plus d'amélioration sur le côté temps d'exécution et le côté taux de compression toute en gardant un bon PSNR, ainsi que utilisation d'autres types de quantificateur et d'autres types de codage .

Bibliographies

- [1] A. Gagalowicz Traitement des images et vision par machine (travail de rédaction collégial initié au début des années 1990 dans le cadre de cours dispensés à l'ESIEA).
- [2] Maou Hadia - Classification des feuilles de plantes avec la carte de KOHONEN- Mémoire de Master Académique - UNIVERSITE KASDI MARBAH OUARGLA -2015
- [3] Jean-marie MUNGUAKONKWA BIRINGANINE - La liaison automatique des plusieurs images perçues sur un scanner - ISP(Institut Supérieur Pédagogique de Bukavu) - licencié en pédagogie; Option : Informatique de Gestion 2008 .
- [4] Mohammed BENABDELLAH- THESE DE DOCTORAT (OUTILS DE COMPRESSION ET DE CRYPTOCOMPRESSION : APPLICATIONS AUX IMAGES FIXES ET VIDEO) UNIVERSITE MOHAMMED V-AGDAL - Juin 2007.
- [5] Christophe.alleau. Transmettre et stocker de l'information (Images numériques) ENSEIGNEMENT SPECIFIQUE TERMINALE S SCIENCES PHYSIQUES Académie Poitiers .
- [6] Achraf HARRAG – article (Le codage de l'information) - École Supérieure d'Informatique de Paris 2006 .
- [7] Arghyro Paouri, responsable de la cellule audio-vidéo du Centre Edgar-Morin / IIAC & membre du comité technique du consortium ArcMC, 2015 – 2016.
- [8] Lingrand Diane Cours de traitement d'images (Projet RAINBOW) Université Nice Sophia Antipolis, rapport de recherche ISRN I3S/RR-2004-05-FR . 22 Janvier 2004.
- [9] Karine Silini Travailler dans un environnement numérique évolutif (Maître de conférences en informatique - Université du Littoral Côte d'Opale) Version du 17 décembre 2013.
- [10] Alain Trémeau - Introduction au Format d'Images Fixes Partie 1 - Laboratoire d'Informatique Graphique et d'Ingénierie de la Vision - Univ_Saint-Etienne – 2005.
- [11] Julie-Caroline Gagné, chargée de projet Avril 2015
- [12] Tatouage numérique robuste (watermarking) par LSB (Least Significant Bit) Mémoire MASTER PROFESSIONNEL Université de Kasdi Merbah Ouargla Année universitaire : 2015-2016.
- [13] Développement d'algorithmes de tatouage d'images basés sur la SVD et les transformées discrètes(mémoire magister UNIVERSITE FERHAT ABBAS-SETIF 2014).

- [14] Paul Murrell - Importing Vector Graphics: The grImport Package for R - Department of Statistics - The University of Auckland - Private Bag 92019.
- [15] Gérard-Michel COCHARD et Sébastien CHOPLIN - Infographie et Image Informatique - Université de Picardie Jules Verne.
- [16] Carlos Fajardo, Oscar M. Reyes and Ana Ramirez - Seismic Data Compression Using 2D Lifting Wavelet Algorithms - Industrial University of Santander, Bucaramanga, Colombia (Received: 06-10-2014 | Accepted: 02-12-2014 | Online: 30-01-2015).
- [17] David Salomon ,Data Compression. The Complete Reference. Springer, 920p. (2004).
- [18] S. Mallat, « A Wavelet tour Of Signal Processing (2nd Ed.) », 1999.
- [19] Y. Gaudeau, J.M. Moureaux, « Un schéma de compression avec perte efficace pour les images médicaux volumiques », Faculté de Sciences et Technique, Article (SPIHT3D), 2006.
- [20] P.Beaurepaire, “Compression d’images appliquée aux angiographies cardiaques : aspects algorithmiques, evaluation de la qualité diagnostique,” thèse doctorat, d’ordre 07 ISAL0107, Lyon, France, 1997.
- [21] Compression des Images en Couleurs Fixes en Utilisant la DWT (UNIVERSITE BADJIMOKHTAR ANNABA) Memoire de MAGISTER 2013 .
- [22] Wei-Yi Wei - An Introduction to Image Compression - Graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan, ROC
- [23] P.RAVIER, « Cours d’ondelettes », Espeo-université d’Orléans 2000.
- [24] BELADGHAM.M, « Segmentation, Analyse et Compression d’image par Ondelettes: Application aux IRM », Mémoire de magister en électronique option : Signaux et systèmes, université de tlemcen. Octobre 2003.
- [25] B.DELYON, « ondelettes orthogonales et bi orthogonales », RR, N° 1985, 1993.
- [26] P.G.LEMARIE, « les ondelettes en 1988 », Lecture Notes in Mathematics, No 1438 springer – Verlag Publisher.
- [27] I.DAUBECHIES, « orthonormal bases of Compactly Supported Wavelet », com.pure Appl.Math. Vol.41, pp.909, Nov 1988.
- [28] E. B .BOUCHEREAU, « Analyse d’images par transformées en ondelettes : Application aux images sismiques. », Thèse de doctorat de l’université Joseph Fourier, gronoble 1, Spécialité : « Mathématiques Appliquées », Mars 1997.

- [29] P.J.BRUT, E.H.ADELSON, « the LAPLACIEN pyramid as a compact image code », IEEE Trans. On com. Vol. COM-31 (4), pp.337-345, April 1983.
- [30] E. Le Pennec and S. Mallat. Bandelet Image Approximation and Compression. SIAM Multiscale Modeling and Simulation, page to appear, 2005.
- [31] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. Biometrika, 81 :425–455, Dec 1994.
- [32] Gabriel Peyré. Géométrie multi-échelles pour les images et les textures. Mathématiques [math].Ecole Polytechnique X, 2005. Français.
- [33] D. Donoho. Wedgelets : Nearly-minimax estimation of edges. Ann. Statist, 27 :353–382, 1999.
- [34] Moussa Ammar. Optimisation d'un schéma de codage d'image à base d'une TCD. Application a un codeur JPEG pour l'enregistrement numérique à bas débit. Traitement du signal et de l'image. Telecom ParisTech, 2002. Français .
- [35] B.K. Alpert. Wavelets and Other Bases for Fast Numerical Linear Algebra, pages 181–216. C. K. Chui, editor, Academic Press, San Diego, CA, USA, 1992.
- [36] Pierre Nerzic, Codage et compression d'images et de vidéo, IUT de Lannion - dept Informatique - univ-rennes1 automne/hiver 2007-2012
- [37] Dr. Mvogo Ngono Joseph - Cours : Compression d'images (Principes généraux de codage entropique d'une source) Master II: IASIG.
- [38] Z. Brahimi and K.A. Saadi. Color image coding based on embedded wavelet zerotree and scalar quantization. In International Conference on Pattern Recognition, volume 1, pages 504-507, 2004.
- [39] Jean-Pierre ADOUL , La quantification vectorielle des signaux approche algébrique, ANN.Telecomm,n°34,1986.
- [40] La compression des images 3D memoire de magister Universite Des Sciences Et De La Technologie D'oran-Mohammed Boudiaf- Usto-Mb 2011.
- [41] Codage et Compression de données multimédia - Département R&T IUT Nice Côte d’Azur Université de Nice - Sophia Antipolis , Frédéric Payan (MCF UNS) 2014.
- [42] Pu, Ida Mengyi. Fundamental Data Compression. Oxford : Elsevier's Science and Technology, 2006. pp 180-181. ISBN-10:0-7506-6310-3.
- [43] A. Gersho and R. M. Gray. Vector quantization and signal compression. Kluwer, Boston, 1992.

- [44] Etude d'algorithmes de quantification vectorielle arborescente pour la compression d'images fixes, rapport de recherche n° 2241, Unité de recherche INRIA Rennes IRISA France janvier 1994 .
- [45] S.A Elhannachi, Compression des images fixes en couleur par hybridation d'un réseau de neurones et fractale, thèse de doctorat de l'université MB-USTO, 2005.
- [46] Hocine Bekkouche. Synthèse de bancs de filtres adaptés, application à la compression des images, thèse de doctorat de l'université paris-sud XI, 2007.