

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

Université Abou Bekr Belkaid
Tlemcen Algérie



جامعة أبي بكر بلقايد

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique



MEMOIRE DE PROJET DE FIN D'ETUDES



Pour l'obtention du diplôme de master en informatique

Thème

Synthèse et Etude comparative sur les méthodes de
compression d'images DCT et DWT

Réalisé par :

- Mr. Lasгаа Ismail.

Encadré par :

- Mr. Yaghmorasan Benzian.

Soutenu le : 02/05/2018 à 10h devant les Jurys :

Président: Mr Benamar Abdelkarim.

Examineur: Mr Benaissa Mohammed.

Année Universitaire : 2017-2018

Remerciements

Avant tout, nous remercions notre Dieu de nous avoir aidés à faire notre thème de fin d'étude et Merci à ma famille qui nous soutiennent bien.

Au terme de ce travail, nous tiens à exprimer notre profonde gratitude et nos sincères Remerciements à notre tuteur de notre projet de fin d'étude à l'université Abou BakrBelkaid. Monsieur *YaghmorasanBenzian* qui a accepté d'encadrer nos travaux.

Nous tenons à remercier vivement les jurys Monsieur *BenamarAbdelkarim* et Monsieur *Benaissa Mohammed* de leurs extrême empressement d'évaluer notre travail. Qu'ils ont accepté de venir à évaluer notre travail.

Nous voudrions remercier également tous nos amis de master nouveaux et DEUA.

Nos profonds remerciements vont à nos camarades.

Nos plus vifs remerciements s'adressent aussi à tout le cadre professoral et administratif de l'Université Abou BakrBelkaid Tlemcen.

Nos remerciements vont enfin à toute personne qui a contribué de près ou de loin à L'élaboration de ce travail.

Dédicace

Je dédie ce modeste travail à :

A MA CHÈRE MÈRE

A LA MÉMOIRE DE MON PÈRE

A ma femme qui a toujours été à mes côtés

A mon cher fils

A mes frères et sœurs, ainsi que leurs enfants

A tous mes amis et collègues

AU CADRE PROFESSORAL

*Et à tous ceux qui ont contribué de près ou de loin pour que ce
Projet soit possible, je vous dis merci.*

Liste des figures

Figure I.1 : représentation de pixel	03
Figure I.2 : L'image comme un groupe	03
Figure I.3 : Synthèse additive	05
Figure I.4 : image vectorielle	06
Figure I.5 : Image Bitmap	06
Figure II.1 : Arbre de Huffman après construction	13
Figure II.2 : Les étapes de la Compression	20
Figure II.3 : Les étapes de la décompression	20
Figure II.4 : Blocs constituant la chaîne de codage et décodage de JPEG2000	25
Figure II.5 : Division en tiles, décalage en DC et DWT sur chaque composante	26
Figure II.6 : Filtrage par des filtres passe-bas et passe-haut	27
Figure II.7 : Principe de décomposition 2D par filtrage passe - bas (H0) et passe - haut (H1) dans le sens horizontal (Ligne) et vertical (Colonne)	28
Figure II.8 : Diagramme de Mallat représentant les coefficients d'ondelettes de la transformée, classés par sous-bandes de filtrage et niveau de décomposition	28
Figure II.9 : Exemple d'une décomposition en ondelettes à 2 niveaux de résolution	29
Figure II.10: Le schéma général d'un codeur – décodeur	29
Figure III.1 : Interface graphique et IDE de MATLAB R2013a	32
Figure III.2 : Le résultat d'exécution de code source de fichier Image_compression.m	32
Figure III.3 : Le résultat d'exécution de bouton « Encoding »	35
Figure III.4 : L'image compresser par DWT.	35
Figure III.5 : L'image compresser par DCT	36
Figure III.6 : figure représentative illustre l'image originale (a gauche) et l'image compressée avec DWT ('nq=10', 'first level') et l'image compressée avec DWT ('nq=90' 'first level'(a droite))	37
Figure III.7 : figure représentative illustre l'image originale (a gauche) et l'image compressée avec DWT ('nq=10', 'second level') et l'image compressée avec DWT ('nq=90' 'second level'(a droite))	38
Figure III.8 : figure représentative illustre l'image originale (a gauche) et l'image compressée avec DWT ('nq=10', 'third level') et l'image compressée avec DWT ('nq=90' 'third level'(a droite))	39

Liste des tableaux

Tableau I.1 : Tableau comparatif entre les caractéristiques des différents formats de compression d'image	08
Tableau II.1: Symboles avec leurs probabilités	12
Tableau II.2 : Code des symboles	13
Tableau II.3: Exemple d'un technique codage utilise l'algorithme de Lempel Ziv	19
Tableau III.1: Comparaisons entre compression DCT et DWT avec $nq=10$,first level	36
Tableau III.2: Comparaison entre compression DWT avec $nq=10$ et $nq=90$ 'first level'	37
Tableau III.3: Comparaison entre compression DWT avec $nq=10$ et $nq=90$ 'second level'	38
Tableau III.4: Comparaison entre compression DWT avec $nq=10$ et $nq=90$ third	39

Listes des abreviations

BMP	Bitmap
CR	Compression Ratio
CYM	(C) Cyan, (Y) Yellow, (M) Magenta.
DC	Direct Current
DCT	Discrete cosine transform
DWT	Discrete wavelet transform
GIF	Graphics Interchange Format
ISO	International Standards Organisation
JPEG	Joint Photographic Experts Group
JPEG2000	Standard de compression d'images fixes récent, introduit par JPEG
LZW	Lempel–Ziv–Welch
MSE	Erreur quadratique moyenne
PCX	Picture Exchange Image Bitmap Zsoft
PNG	Portable Network Graphics
PSNR	Rapport signal sur bruit crête.
QMF	Quadrature mirrorfilter
RGB	(R) Red, (G) Green, (B) Bleu
RLE	Run Length Encoding
R,V,B	(R) Rouge, (V) Vert, (B) Bleu
TIFF	Tagged Image File Format

Sommaire

Introduction générale.....	1
<i>Chapitre I: Notions d'image</i>	
I.1 Introduction.....	2
I.2 Définition de L'image.....	2
I.3 Image numérique.....	2
I.4 Caractéristiques d'images.....	2
I.4.1 Pixel.....	2
I.4.2 Dimension.....	4
I.4.3 Résolution.....	4
I.4.4 Bruit.....	4
I.4.5 Histogramme.....	4
I.4.6 Contours et textures.....	4
I.4.7 Luminance.....	4
I.4.8 Contraste.....	4
I.4.9 Images à niveaux de gris.....	5
I.4.10 Images en couleurs.....	5
I.5 Types d'images.....	5
I.5.1 L'image vectorielle.....	5
I.5.2 L'image Bitmap (matricielle).....	6
I.6 Les types de formats d'image.....	6
I.6.1 BMP (bitmap).....	6
I.6.2 PNG (Portable Network Graphics).....	7
I.6.3 TIFF (Tagged Image File Format).....	7
I.6.4 GIF (Graphics Interchange Format).....	7
I.6.5 JPEG (Joint Photographic Experts Group).....	7
I.7 Conclusion.....	9
<i>Chapitre II : La différente méthode de compression</i>	
II.1.Introduction.....	10
II.2 Définition de la compression.....	10
II.2.1 Compression physique et logique.....	10
II.2.2 Compression symétrique et asymétrique.....	11
II.3 Concepts de compression.....	11
II.3.1 Compression avec pertes.....	11

II.3.2 Compression sans perte	11
II.4 Les différentes méthodes de compression	12
II.4.1 Codage Huffman.....	12
II.4.1.1 Principe de la méthode.....	12
II.4.1.2 Exemple de codage Huffman	12
II.4.2 L'algorithme de Shannon-Fano	13
II.4.3 Le codage arithmétique.....	14
II.4.3.1 Procédure de codage arithmétique.....	14
II.4.4 RLE (Run Length Encoding).....	16
II.4.4.1 Algorithme de compression.....	16
II.4.4.2 Algorithme de décompression	16
II.4.4.3 Caractéristiques de compression	16
II.4.5 La méthode LZW (Lempel Ziv Welch).....	16
II.4.6 La compression JPEG	18
II.4.6.1 Principe de compression du JPEG.....	18
II.4.6.2 Utilité et caractéristiques de la compression JPEG.....	23
II.4.7 La compression JPEG 2000.....	23
II.4.7.1 Fonctionnement général du JPEG2000.....	24
II.4.7.2 Prétraitement ou Traitements préliminaires.....	24
II.4.7.3 Transformée en ondelettes	25
II.4.7.4 Implémentation par banc de filtres	25
II.4.8 Compression par fractales.....	29
II.5 Conclusion.....	30
<i>Chapitre III : Implémentation et résultat</i>	
III.1 Introduction.....	31
III.2 Définition de Matlab	31
III.3 L'interface de programme d'application MATLAB (API)	31
III.4 Exécution de notre projet.....	32
III.5 Description de l'interface graphique	33
III.6 Analyse des résultats	33
III.6.1 Analyse des résultats à travers la table au-dessus.....	35
III.6.2 Analyse des résultats à travers du première table	38
III.6.3 Analyse des résultats à travers du deuxième table.....	39
III.6.4 Analyse des résultats à travers du troisième table	40

III.6.5 Analyse final.....	40
III.7 Conclusion.....	40
Conclusion générale	41
Références bibliographiques	42

Introduction générale

La compression des données est souvent appelée codage, où le codage est un terme très général englobant toute représentation spéciale des données qui satisfait une tâche ou un besoin. Comme la compression des fichiers, l'objectif de la compression des images est de réduire la taille et d'économiser de l'espace de stockage. Cependant, les algorithmes de compression d'images sont spécifiques à certains types de médias, tels que les fichiers images.

L'idée de base de la compression d'images est de réduire le nombre moyen de bits par pixel nécessaire à sa représentation. Il est possible dans une certaine limite de réduire ce nombre sans perte d'information. Au-delà, il est nécessaire d'élaborer des algorithmes de compression irréversibles (avec pertes) induisant une distorsion pas ou peu visible dans les conditions normales d'observation des images.

L'objectif de ce travail est la comparaison entre les différents types de compression d'images.

Plan du travail

Ce mémoire est structuré comme suit :

- Le premier chapitre, contient des généralités sur les notions d'image D'une manière générale.

- Dans le deuxième chapitre, on va présenter les méthodes de compression d'image.

« • Le troisième chapitre sera consacré aux résultats obtenus de l'opération de compression d'image par les méthodes JPEG et JPEG 2000, Ainsi qu'une interprétation des résultats.

- Enfin, nous clôturons notre manuscrit par une conclusion générale.

Chapitre I

Notions d'image

Chapitre I : Notions d'image

I.1 Introduction

La compression d'images est une nécessité plus vitale dans le domaine du multimédia, car les images numériques comme nous allons le découvrir, forment un énorme ensemble de données, même lorsqu'il s'agit d'une simple image fixe, d'où leur stockage et leur transmission qui deviennent très pénibles. Pour cela, nous observons la naissance de toute une branche de la science, celle qui traite les images fixe et mobile.

I.2 Définition de L'image

Une image, numérique ou fixe et est une représentation binaire d'informations visuelles telles que des dessins, des images, des graphiques, des logos ou des images vidéo individuelles. Les images numériques peuvent être sauvegardées électroniquement sur n'importe quel périphérique de stockage [s1].

I.3 Image numérique

L'image numérique est une matrice de $X \times Y$ pixels correspondant à l'échantillonnage et la quantification d'un signal acquis avec une caméra.

Une image numérique est constituée d'un tableau de valeurs entières. Pour pouvoir stocker et transmettre cette image comme n'importe quelle autre donnée informatique, il faut la coder en binaire, c'est-à-dire la décrire par une suite de 0 et de 1 [s2].

La numérisation d'une image peut s'effectuer selon deux procédés différents de codage, ayant chacun leurs applications propres et produisant deux modes d'images : en pixel ou 'vectoriel.

I.4 Caractéristiques d'images

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

I.4.1 Pixel

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

Le nom de "pixel", abrégé px, provient de l'expression anglaise Picture élément, qui signifie "élément d'image" ou "point élémentaire". Un pixel est généralement rectangulaire ou

Chapitre I : Notions d'image

pres'que carré et présente une taille comprise entre 0,18 mm et 0,66 mm de côté [s3].



Figure I.1 représentation de pixel

La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci dessous :



Figure I.2 : L'image comme un groupe de pixels

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

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

I.4.2 Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

I.4.3 Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre).

I.4.4 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

I.4.5 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image.

I.4.6 Contours et textures

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

I.4.7 Luminance

C'est le degré de luminosité des points de l'image.

I.4.8 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image.

I.4.9 Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires [1].

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

I.4.10 Images en couleurs

La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.). [s4], [2], [1].

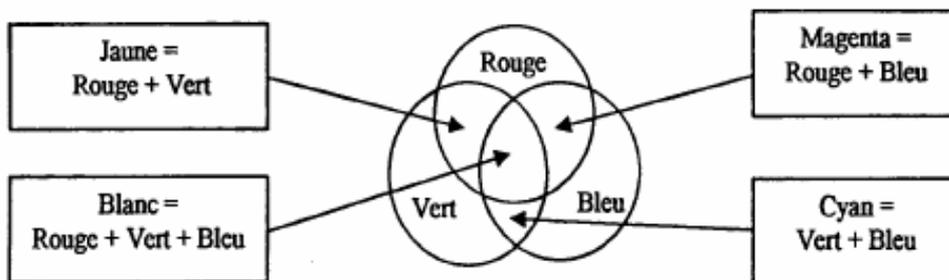


Figure I.3 : Synthèse Additive

I.5 Types d'images

I.5.1 L'image vectorielle

L'image vectorielle est une représentation conceptuelle de forme calculée par des formules mathématiques, (exemple, un cercle n'est pas déterminé par des pixels mais par une formule mathématique qui détermine sa forme, sa taille et son emplacement).

L'image Vectorielle utilise également la technique du Pixel, mais cette fois, leur position et leur couleur ne sont pas figées puisqu'elles sont calculées dynamiquement par le

logiciel. [s5]

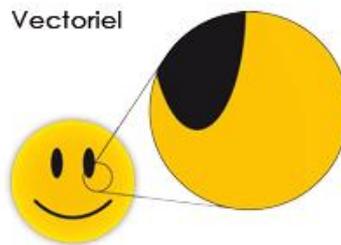


Figure I.4: image vectorielle

I.5.2 L'image Bitmap (matricielle)

L'image bitmap est représentée par une trame de points que l'on appelle pixels. Ce ne sont pas des formules mathématiques qui définissent les formes, mais un ensemble de pixels qui agissent comme un tableau pointilliste [s6].

Exemple

Une image (comme un cercle par exemple) de petite taille, que l'on agrandit dix fois, Cette opération peut entraîner une perte dans la qualité des couleurs ou la netteté. Alors l'image devient déformée (figure I.3).

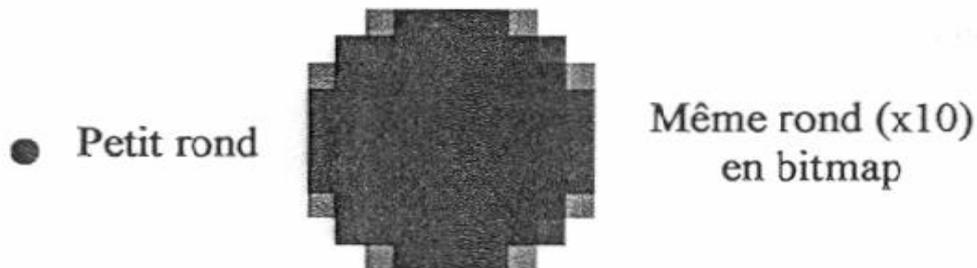


Figure I.5 : image Bitmap

I.6 Les types de formats d'image

I.6.1 BMP (bitmap):

Bitmap (BMP) est un format non compressé de fichier image qui peut être utilisé pour créer et stocker des graphiques informatiques. Un fichier bitmap affiche un petit point dans un motif qui, lorsqu'il est vu de loin, crée une image globale. Une image bitmap est une grille composée de lignes et de colonnes où une cellule spécifique reçoit une valeur qui la remplit ou la laisse vierge, créant ainsi une image hors des données [s7].

Chapitre I : Notions d'image

I.6.2 PNG (Portable Network Graphics):

Le format PNG fournit une norme portable, légalement inutilisée, bien compressée et bien définie pour les fichiers d'images bitmap sans perte.

Bien que la motivation initiale pour le développement de PNG soit de remplacer GIF, la conception offre de nouvelles fonctionnalités utiles non disponibles dans GIF, avec un coût minimal pour les développeurs [3].

Transparence

La transparence est supportée avec la possibilité d'utiliser des masques (fusion de deux images).

I.6.3 TIFF (Tagged Image File Format):

Le format de fichier d'image marqué (TIFF) est un autre format de fichier d'image populaire. Développé par Aldus Corporation dans les années 1980, il a ensuite été soutenu par Microsoft. Son support pour l'ajout d'informations complémentaires (dénommé "tags") offre une grande flexibilité physique [4].

I.6.4 GIF (Graphics Interchange Format) :

Le format GIF est un type de bitmap, mais contrairement au format JPEG ou PNG, les fichiers GIF sont limités à une palette maximale de 256 couleurs. Essentiellement, chaque image GIF contient une «boîte de crayons» prédéfinie et il n'y a aucun moyen de mélanger ces couleurs pour créer de nouvelles couleurs. Alors que GIF est généralement un choix médiocre pour les images avec une grande variation de couleur, cette limite de 256 couleurs peut aider à réduire les dimensions des fichiers, ce qui est idéal même pour les vitesses d'Internet les plus lentes. Pendant de nombreuses années, GIF a fourni la seule option de transparence du Web - bien que PNG et SVG offrent maintenant cela aussi [s8].

I.6.5 JPEG (Joint Photographic Experts Group):

C'est un format de fichier graphique permettant des taux de compression impressionnants mais au détriment de la qualité de l'image : la compression se fait avec perte d'information. L'extension de fichier correspondante est JPG. Ce format ne supporte pas la transparence contrairement au GIF. La norme JPEG utilise l'ADCT (Transformée en cosinus discrète) [s9].

Chapitre I : Notions d'image

Le tableau et l'image au-dessous illustrent la différence entre les formats de compression d'image.

Format	Nom	L'extension de fichier	type de compression	méthodes	L'usage
JPEG	Joint Photographic Experts Group	.jpg ou .jpe ou .jpeg	Compression avec perte	DCT/RLE	Pour naturel images
JPEG 2000	Joint Photographic Experts Group 2000	.jp2 ou .j2k Ou .jpf ou .jpg2	Compression avec perte	DWT	Pour meilleure qualité d'image (20% meilleur que jpeg)
GIF	Graphics Interchange Format	.gif ou .gfa ou .giff	Compression sans perte	LZW	Pour les images animées
PNG	Portable Network Graphics	.png	Compression sans perte	DEFLATE	Pour les webcomics et pour les logos et c'est un meilleur format de compression
TIFF	Tagged Image File Format	.tif ou .tiff	Compression sans perte	Zip/LZW/RLE /DEFLATE	Pour multi images dans un fichier
BMP	bitmap	.bmp	Non compressé format	Zip	Pour les bitmap

Tableau I.1 : Tableau comparatif entre les caractéristiques des différents formats de compression d'image.

I.7 Conclusion

Dans ce chapitre, nous avons essayé de faire un récapitulatif sur les notions élémentaires d'image numérique (vectorielle et matricielle) comme pixel, dimension, résolution, bruit, histogramme, contours et textures, luminance, contraste, niveau de gris et la couleur et ses différents types formats (Bitmap, jpeg,png,tiff,gif.).

Le chapitre suivant est consacré aux objectifs de la compression et les différentes méthodes de compression

Chapitre II

Les différentes méthodes de compression

Chapitre II : Les différentes méthodes de compression

II.1.Introduction

Avec le développement de l'outil informatique, on effectue des échanges de volumes importants d'information. Or la gestion d'une telle masse pose des problèmes de stockage et de transfert. Pour cela, des études ont été menées afin de mettre en évidence des algorithmes de compression et décompression de données. Leur but est de changer le format des informations de telle sorte qu'elles occupent moins de volume. Une fois compressées, les données ne sont plus accessibles en tant que données cohérentes; pour les récupérer, il suffit de les décompresser.

Dans ce chapitre, nous donnons quelques notions essentielles sur la compression d'image.

II.2 Définition de la compression :

En général, la compression des données consiste à prendre un flux de symboles et à les transformer en codes [5]. La compression est effectuée par un programme qui utilise une formule ou un algorithme pour déterminer comment réduire la taille des données pour gagner l'espace de stockage.

La compression de données est également appelée codage source ou réduction de débit est le processus de modification, d'encodage ou de conversion de la structure des bits de données de manière à consommer moins d'espace sur le disque. Il permet de réduire la taille de stockage d'une ou plusieurs instances ou éléments de données.

L'objectif de la compression des données est de représenter une source d'information aussi précisément que possible en utilisant le plus petit espace de stockage.

II.2.1 Compression physique et logique :

Dans le cas de compression physique on considère la compression comme un algorithme capable de comprimer énormément de données dans un minimum d'espace mémoire mais la compression logique est un algorithme de compression qui a pour but de recoder les données dans une représentation différente plus compacte contenant la même information physique [s6].

Une distinction doit cependant être faite entre la compression physique et la compression logique [6], [7] :

- **La compression physique** : s'applique uniquement aux données de l'image. Il s'agit de traduire les trains de bit d'un motif à un autre.

Chapitre II : Les différentes méthodes de compression

- **La compression logique** : est effectuée par un raisonnement logique en substituant une information par une information équivalente.

II.2.2 Compression symétrique et asymétrique :

Les algorithmes de compression peuvent être divisés en deux groupes : les algorithmes symétriques et les algorithmes asymétriques [s10] .

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

La **Compression asymétrique** Demande cependant plus de travail pour l'une des deux opérations; si par exemple, on désire stocker une image dans une base de données, il est intéressant de la stocker de manière à ce qu'elle tienne un minimum de place sur le disque, et qu'elle soit le plus rapidement possible décompressée. Les algorithmes nécessitant plus de travail à la décompression ne sont pas courants.

II.3 Concepts de compression

Il y a deux concepts de compression importants sont une compression à perte et sans perte :

II.3.1 Compression avec pertes :

Les techniques de compression avec perte impliquent une certaine perte d'information, et les données qui ont été compressées à l'aide de techniques à perte ne peuvent généralement pas être récupérées ou reconstruites exactement. En contrepartie de l'acceptation de cette distorsion dans la reconstruction, nous pouvons généralement obtenir des rapports de compression beaucoup plus élevés que ce qui est possible avec une compression sans perte [8].

II.3.2 Compression sans perte :

Avec la compression sans perte, les données sont compressées sans perte de données. Les techniques de compression sans perte, comme leur nom l'indique, ne comportent aucune perte d'information. Si les données ont été compressées sans perte, les données d'origine peuvent être récupérées exactement à partir des données compressées. La compression sans

Chapitre II : Les différentes méthodes de compression

perte est généralement utilisée pour les applications qui ne tolèrent aucune différence entre les données originales et les données reconstruites [8].

Exemples de méthodes sans perte:

- codage en longueur, codage Huffman, méthode Lempel-Ziv-Welche (LZW).

II.4 Les différentes méthodes de compression

II.4.1 Codage Huffman :

Le codage Huffman [9], crée des codes à longueurs variables sur un nombre entier de bits.

L'algorithme considère chaque message à coder comme étant une feuille d'un arbre qui reste à construire. L'idée est d'attribuer aux messages de plus faibles probabilités, les mots codés les plus longs et les mots codés les plus courts pour les messages de fortes probabilités.

II.4.1.1 Principe de la méthode :

1. Calculer la probabilité associée à chaque symbole dans la chaîne à coder.
2. Classer les probabilités d'apparition des symboles par ordre de probabilités décroissantes.
3. Sommer les deux plus faibles probabilités correspondantes aux deux symboles de la séquence (la plus petite probabilité prene un 0 et celle plus grande prene un 1) ;
Ce qui donne une nouvelle probabilité.
4. Répéter les opérations 2 et 3, jusqu'à ce qu'il ne reste que deux probabilités (la racine).
5. Lire le code binaire de chaque symbole de haut vers le bas.

II.4.1.2 Exemple de codage Huffman :

Soit une source composée d'un alphabet de 8 symboles avec les probabilités données sur le tableau (2.1).

Symbole	A	B	C	D	E	F	G	H
Probabilité	0.01	0.02	0.05	0.09	0.18	0.19	0.21	0.25

Tableau II.1 : Symboles avec leurs probabilités.

Chapitre II : Les différentes méthodes de compression

L'arbre de Huffman est alors donné par la figure (II.1).

Le Codage des symboles est alors donné par le tableau (II.2).

Symbole	A	B	C	D	E	F	G	H
Code	110000	110001	11001	1101	111	00	01	10

Tableau II.2 : Code des symboles.

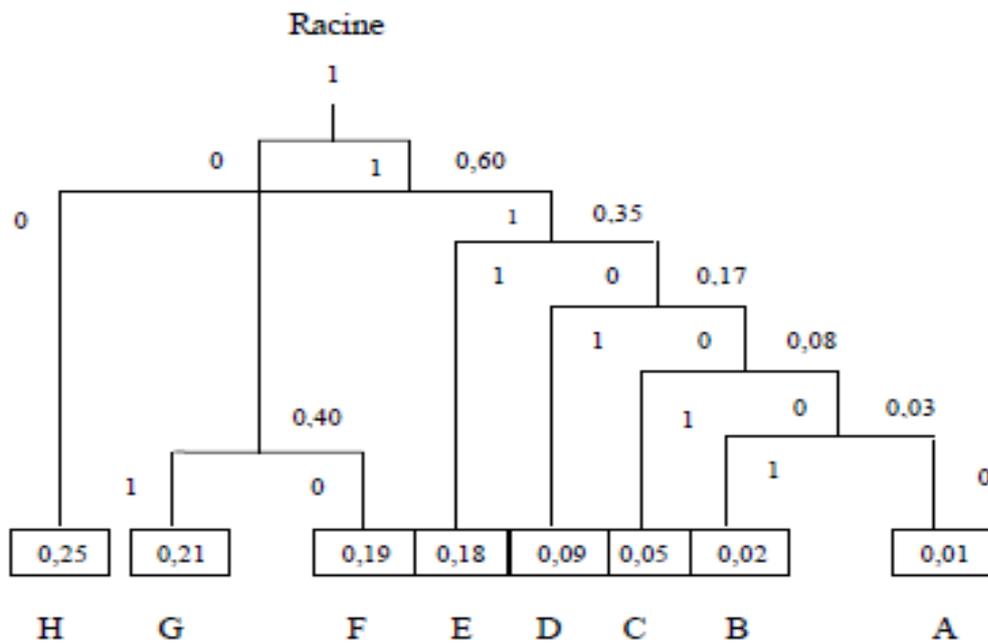


Figure II.1 : Arbre de Huffman après construction.

II.4.2 L'algorithme de Shannon-Fano

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

-Les symboles sont triés et classés en fonction de leur fréquence en commençant par le plus fréquent.

Chapitre II : Les différentes méthodes de compression

- La liste des symboles est ensuite divisée en deux parties de manière à ce que le total des fréquences de chaque partie soit aussi proche que possible.

- Le chiffre binaire 0 est affecté à la première partie de la liste, le chiffre 1 à la deuxième partie.

- Chacune des deux parties fait à son tour l'objet des démarches 2 et 3.

Et ainsi de suite jusqu'à ce que chaque symbole soit devenu une feuille de l'arbre correspondant à un code déterminé.

Sauf dans des cas extrêmement simples, cette méthode ne permet pas d'approcher efficacement l'entropie. On lui préfère l'algorithme de Huffman [6].

II.4.3 Le codage arithmétique

Il a été prouvé que le codage de Huffman est la meilleure méthode de codage à codes de longueur fixée. Mais les codes de Huffman doivent avoir une longueur entière de bits, ce qui ne permet pas toujours de réaliser une compression optimale. Le codage dit arithmétique est alors une méthode plus performante [8].

II.4.3.1 Procédure de codage arithmétique [10] :

- Calculer la probabilité associée à chaque symbole dans la chaîne à coder.

- Associer à chaque symbole un sous intervalle proportionnel à sa probabilité, dans l'intervalle [0.1] (l'ordre de rangement des intervalles sera mémorisé car il est nécessaire au décodeur)

- Initialiser la limite inférieure de l'intervalle de travail à la valeur 0 et la limite supérieure à la valeur 1.

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

* largeur = limite supérieure - limite inférieure.

* limite inférieure = limite inférieure + largeur * (limite basse du sous intervalle du symbole).

* limite supérieure = limite inférieure + largeur * (limite haute du sous intervalle du symbole).

- La limite inférieure code la chaîne de manière unique.

On remarque que le premier symbole de la chaîne fixe est le premier chiffre après la virgule du code final.

Décrivons le principe du codage et de décodage sur un exemple :

Considérons le codage du message ATLAS. Les probabilités des caractères sont les suivantes :

Chapitre II : Les différentes méthodes de compression

Caractère	Probabilité
A	1/10
L	1/5
S	1/10
T	1/10

Dans l'intervalle général de probabilité [0,1], chaque symbole se voit affecter un intervalle de probabilité (la manière d'affecter cet intervalle n'ayant pas d'importance).

Caractère	Probabilité	Intervalle
A	1/10	$0.10 \leq r < 0.20$
L	1/5	$0.20 \leq r < 0.40$
S	1/10	$0.40 \leq r < 0.50$
T	1/10	$0.50 \leq r < 0.60$

Le codage s'effectue progressivement à partir de la première lettre du message. Puisque c'est A, le codage du message doit être compris entre 0,10 et 0,20. La deuxième lettre est T; dans le sous intervalle 0,10 à 0,20, le codage sera compris entre 50 et 60 %. Autrement dit, à ce stade (AT) sera compris entre $0,10+(0,20-0,10) \times 0,50$ et $0,10+(0,20-0,10) \times 0,60$, soit entre 0,15 et 0,16.

Au niveau de la troisième lettre L, il doit être compris entre $0,15+(0,16-0,15) \times 0,20$ et $0,15+(0,16-0,15) \times 0,40$ soit entre 0,152 et 0,154 .

En continuant, on arrive au tableau suivant :

Caractère successif	Limite inférieure	Limite supérieure
A	01	0.2
T	0.15	0.16
L	0.152	0.154
A	0.1522	0.1524
S	0.15228	0.1523

Le codage du message est constitué par la dernière limite inférieure, 0,15228.

Le décodage s'effectuera de manière unique à partir de ce nombre, suivant le

Chapitre II : Les différentes méthodes de compression

mécanisme inverse.

Cette technique se montre un peu plus lente que celle de Huffman mais elle présente des taux de compression supérieurs [1].

II.4.4 RLE (Run Length Encoding)

But : cet algorithme élide les répétitions successives de caractères.

II.4.4.1 Algorithme de compression [8]

1. Recherche des caractères répétés plus de n fois (n fixé par l'utilisateur)
2. Remplacement de l'itération de caractères par :
 - un caractère spécial identifiant une compression.
 - le nombre de fois où le caractère est répété.
 - le caractère répété.

II.4.4.2 Algorithme de décompression

Durant la lecture du fichier compressé, lorsque le caractère spécial est reconnu, on effectue l'opération inverse de la compression tout en supprimant ce caractère spécial [7].

Exemple :

AAAAARRRRRROLLLLBBBBBUUTTTTT

On choisit comme caractère spécial : @ et comme seuil de répétition : 3

Après compression : @5A@6RO@4L@5BUU@6T gain : 11 caractères soit 38%.

Utilité : essentiellement pour la compression des images (car une image est composée de répétitions de pixels, de couleur identique, codés chacun par un caractère).

II.4.4.3 Caractéristiques de compression [11]

- algorithme très simple.
- taux de compression relativement faible (40%).

C'est une méthode utilisée par de nombreux formats d'images (BMP, PCX, TIF).

II.4.5 La méthode LZW (Lempel Ziv Welch)

LZW (Lempel-Ziv-Welch), est un algorithme de compression adaptatif qui n'assume aucune connaissance des probabilités de symboles. [12]

La compression LZW fonctionne en lisant une séquence de symboles, en regroupant les symboles en cordes et en convertissant les chaînes en codes. Parce que les codes occupent moins d'espace que les chaînes qu'ils remplacent, nous obtenons une compression. Les fonctionnalités caractéristiques de LZW incluent.

Chapitre II : Les différentes méthodes de compression

Exemple : on a la liste de donnée suivante :

a b b a a b b a a b a b b a a a b a a b b a

Algorithme de codage :

Etape 1 : Initialisation du dictionnaire pour contenir tous les blocs de longueur un ($D = \{a, b\}$).

Etape 2 : Recherche du bloc le plus long W qui est apparu dans le dictionnaire.

Etape 3 : Encodage W par son index dans le dictionnaire.

Etape 4 : Ajout de W suivi du premier symbole du prochain bloc au dictionnaire.

Etape 5: Aller à l'étape 2.

Nous commençons par appliquer l'algorithme de LZW et nous obtenons le résultat suivant



dictionnaire			
index	entrée	index	entrée
0	a	7	b a a
1	b	8	a b a
2	a b	9	a b b a
3	b b	10	a a a
4	b a	11	a a b
5	a a	12	b a a b
6	a b b	13	b b a

Tableau II.3 : Exemple d'un technique codage utilise l'algorithme de Lempel Ziv

Chapitre II : Les différentes méthodes de compression

II.4.6 La compression JPEG

Une des meilleures compressions actuellement sur le marché est la compression JPEG (JPEG signifiant Joint Photographie Experts Group).

En 1982, l'ISO format un groupe d'expert de la photographie (PEG) afin de réfléchir sur les méthodes permettant de transmettre des images fixes, des animations et du texte sur le réseau RNIS.

En 1986, un groupe de l'IUT-T débuta une étude sur les méthodes de compression de documents fax en niveau de gris ou en couleur. La méthode de compression élaborée était très proche de celle mise au point par PEG.

Ainsi, en 1987, l'ISO et RUT-T unirent leurs forces et formèrent un comité conjoint d'expert de la photographie (JPEG) [6].

II.4.6.1 Principe de compression du JPEG

La norme JPEG de 91 décrit le format des données compressées et le schéma de codage et de décodage. Les algorithmes de compression sont proposés mais n'ont pas de valeur normative [6].

Voyons d'abord le schéma de principe

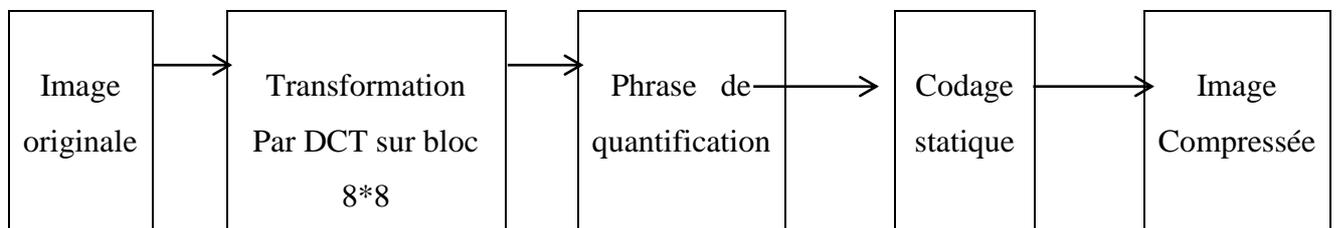


Figure II.2 :Les étapes de la Compression

Pour retrouver l'image décompressée, il faut faire le chemin inverse, mais on ne pourra pas avoir la même image [s6]. Voir le schéma de principe en dessous:

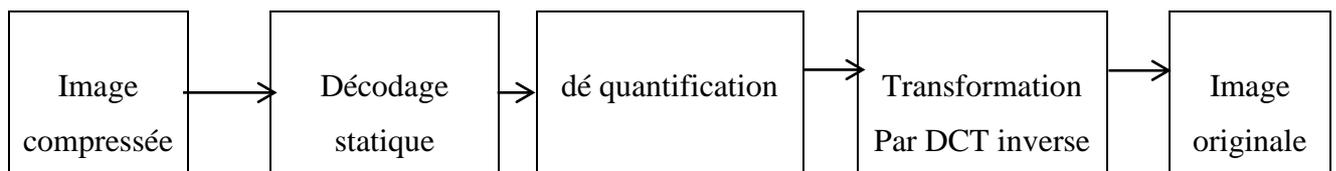


Figure II.3 : Les étapes de la décompression

Chapitre II : Les différentes méthodes de compression

Exemple

Soit un bloc de 8 x 8 pixels à 256 niveaux de gris:

Matrice de pixels d'entrée

140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	162
147	167	140	155	155	140	136	162
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	136

Ensuite la Transformée en Cosinus Discrète (DCT) est appliquée sur les pixels de chaque bloc.

La formule pour DCT est donnée ci-dessous:

$$\text{DCT}(k_1, k_2) = \frac{2}{N} C(k_1) C(k_2) \sum \sum \text{Pixel}(i,j) \cos((2i+1) i\pi/2N) \cos((2j+1) j\pi/2N)$$

$$C(k) = \begin{cases} \sqrt{1/2} & \text{si } k = 0 \\ 1 & \text{sinon} \end{cases}$$

N=la taille de la matrice d'image exemple : 2x2, 4x4, 8x8 ...etc.

$i, j = \{0 \dots N-1\}$.

Pixel (i,j)= Valeur de pixel.

$C(k) = \sqrt{1/2}$ si $k=0$, $C(k)=1$ si $k \geq 1$

Chapitre II : Les différentes méthodes de compression

Matrice DCT

1210	-18	15	-9	23	-9	-14	-19
21	-34	26	-9	-11	11	14	7
-10	-24	-2	6	-18	3	-20	-1
-8	-5	14	-15	-8	-3	-3	8
-3	10	8	1	-11	18	18	15
4	-2	-18	8	8	-4	1	-7
9	1	-3	4	-1	-7	-1	-2
0	-8	-2	2	1	4	-6	0

Les valeurs de la matrice DCT ont été arrondies à l'entier le plus proche. Le coefficient continu 1210 sur la composante (0,0) représente "la moyenne" de la grandeur d'ensemble de la matrice d'entrée. Cette moyenne n'est pas une moyenne statique mais un nombre proportionnel à la somme de toutes les valeurs du signal. Les autres valeurs de la DCT sont les "écarts" par rapport à cette moyenne. Lorsque l'on monte dans les hautes fréquences, les valeurs de la matrice ont tendance à s'approcher de 0 quand on s'éloigne du coin supérieur gauche de la matrice.

L'étape de quantification de chaque bloc regroupe les ensembles de valeurs proches. Ensuite, chaque amplitude originale sera remplacée par la valeur moyenne de l'intervalle, c'est à dire l'étape de quantification est de diminuer la précision du stockage de sentiers de la matrice DCT pour diminuer le nombre de bits occupés par chaque entier. C'est la partie non conservative de la méthode. Les basses fréquences sont conservées, la précision des hautes fréquences est diminuée. La perte de précision est plus grande lorsqu'on s'éloigne de la position (0,0). Les valeurs de la matrice DCT seront divisées par la matrice de quantification.

Matrice de quantification

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Chapitre II : Les différentes méthodes de compression

Matrice DCT quantifiée

403	-4	2	-1	2	-1	-1	1
4	-5	3	-1	-1	1	1	0
-1	-3	0	0	-1	0	-1	0
-1	0	1	-1	0	0	0	1
0	1	1	0	-1	1	1	0
0	0	-1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Lors de la décompression, il suffira de multiplier la valeur de la matrice DCT quantifiée par l'élément correspondant de la matrice de quantification pour obtenir une approximation de la DCT. La matrice obtenue est appelée matrice DCT dé quantifiée.

Matrice DCT dé quantifiée

1209	-20	14	-9	22	-13	-15	-17
20	-35	27	-11	-13	15	17	0
-7	-27	0	0	-15	0	-19	0
-9	0	13	-15	0	0	0	0
0	13	15	0	-19	21	23	25
0	0	-17	0	0	0	0	0
15	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

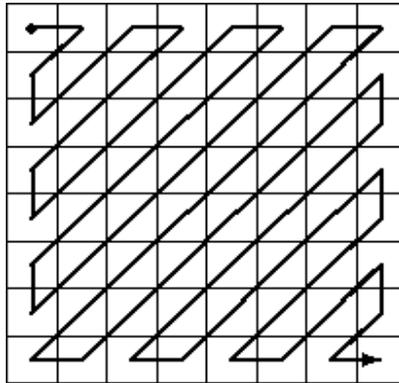
Matrice de pixels de sortie (décompression)

142	143	154	141	133	153	179	179
139	152	129	151	144	154	163	181
150	156	139	166	162	163	154	172
163	145	160	153	151	153	145	154
168	150	156	145	140	139	141	159
148	164	133	164	158	140	136	163
130	159	123	164	165	140	134	145
148	156	140	148	159	146	153	141

Chapitre II : Les différentes méthodes de compression

Le codage de la matrice DCT quantifiée se fait en parcourant les éléments dans l'ordre imposé par une séquence appelée Séquence zigzag. Les éléments sont parcourus en commençant par les basses fréquences puis ensuite en traitant les fréquences de plus en plus élevées.

Parcours en zigzag



Etant donné qu'il y a beaucoup de composantes de hautes fréquences qui sont nulles dans la matrice DCT, la séquence zigzag engendre de longues suites de 0 consécutifs. D'une part, les suites de valeurs nulles sont simplement codées en donnant le nombre de 0 successifs.

D'autre part, les valeurs non nulles seront codées en utilisant une méthode statique.

Les avantages de DCT

- Computationnel efficace (facile à mettre en œuvre dans le matériel).
- Performance de compression élevée.
- Sortie réel.
- Transformation rapide (surtout quand il est utilisé dans une image compressée).
- JPEG est destiné à des images de type photographique.

Les inconvénients de DCT

- Elle besoin de diviser le codage d'entrée en blocs 2d non chevauchants.
- Mauvaise transformation de l'ensemble de l'image.
- Le ratio de compression est bas.

Chapitre II : Les différentes méthodes de compression

II.4.6.2 Utilité et caractéristiques de la compression JPEG [6], [1]

- La norme JPEG doit être très proche des techniques nouvelles de compression, en terme de taux de compression (un excellent taux de compression sur des images ayant les points codés de 6 à 24 bits.), de qualité de restitution et temps de calculs.
- JPEG doit pouvoir compresser n'importe quel type d'images réelles. (peut gérer différentes formes d'expression des couleurs comme RGB, CYM ou les niveaux de gris).
- L'algorithme doit être implémenté sans trop de problèmes sur une grande gamme de CPUs, et sur des cartes spécialisées.
- Le codage doit pouvoir être séquentiel, progressif et sans pertes.
- JPEG est destiné à des images de type photographique. Les images composées de vecteurs, de droite ou d'objet géométrique, ne seront pas compressées avec une excellente efficacité.

JPEG est prévu pour supprimer certaines informations, invisible à l'œil nu, dans une image.

II.4.7 La compression JPEG 2000

JPEG 2000 est une norme basée sur des ondelettes pour la compression d'images encore numériques. Il a été développé par le comité ISO JPEG pour améliorer les performances de JPEG tout en ajoutant de nouvelles fonctionnalités et fonctionnalités importantes pour permettre de nouvelles applications d'imagerie [13].

Norme de compression. Il a été créé par le comité mixte du Groupe d'experts photographiques dans le but de remplacer leur norme JPEG originale transformée en cosinus discrète à l'ondelette (Discrete Wavelet Transform).

Chapitre II : Les différentes méthodes de compression

II.4.7.1 Fonctionnement général du JPEG2000

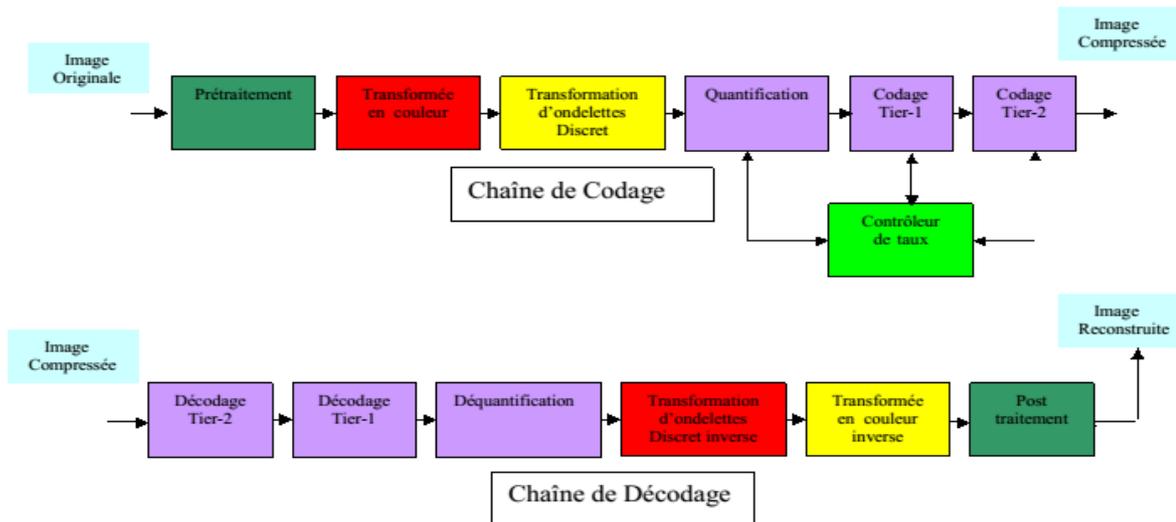


Figure II.4 : Blocs constituant la chaîne de codage et décodage de JPEG2000 [14].

II.4.7.2 Prétraitement ou Traitements préliminaires

Deux opérations sont réalisées au cours de cette phase [15]. La première est le partitionnement de l'image en rectangles de même taille (sauf exception faite pour les zones situées aux extrémités de l'image) que l'on appelle Tiles (figure II.5) et qui ne se recouvrent pas. Les Tiles seront ensuite chacune codée séparément, avec leurs propres paramètres. Ce Partitionnement est particulièrement utile dans les applications qui possèdent des Ressources mémoire limitées.

La seconde opération : DC level shifting convertit les valeurs des composantes non signées de L'image en valeurs signées. Elle consiste à ramener ces coefficients en logique non signée de [0 255] en logique signée à [-128 127], c'est-à-dire centrés autour de zéro en soustrayant 128.

Ce décalage DC level shifting est opéré afin de simplifier certains traitements ultérieurs.

Chapitre II : Les différentes méthodes de compression

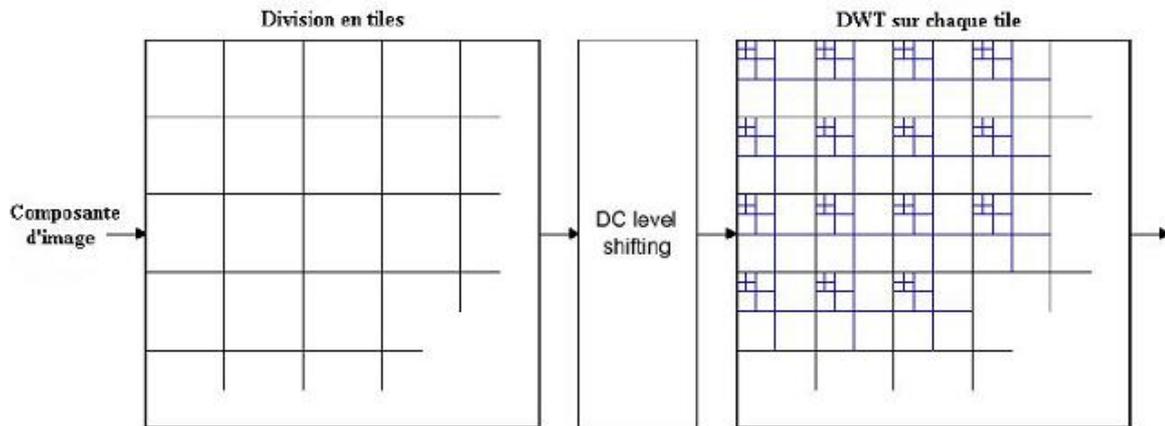


Figure II.5 : Division en tiles, décalage en DC et DWT sur chaque composante.

II.4.7.3 Transformée en ondelettes

L'idée de base de la DWT (Discrete Wavelet Transform) est de séparer les basses et hautes fréquences d'une image [16]. Intuitivement, basses et hautes fréquences peuvent être comprises de la manière suivante. Les basses fréquences correspondent à une version grossière de l'image originale dans laquelle les valeurs des pixels ont été moyennées et où aucune variation brusque n'est observée d'un pixel à l'autre. Les hautes fréquences quant à elles contiennent toute l'information sur les détails de l'image. On comprend aisément que plus d'information est contenue dans la version basse fréquences que dans celle ne fournissant que les détails de l'image. On dit que l'énergie de l'image est concentrée dans les basses fréquences.

L'objectif de la DWT est donc de concentrer l'information de l'image en une zone très localisée de manière à pouvoir ensuite compresser fortement les zones ne contenant que peu d'information.

En pratique, la transformée DWT consiste en l'application successive de paires de filtres passe-bas (H_0) et passe-haut (H_1), suivis d'un sous-échantillonnage de facteur deux (suppression d'un échantillon sur deux), comme l'illustre la figure (II.6). La transformée inverse consiste à appliquer une autre paire de filtres passe-bas (G_0) et passe-haut (G_1), précédés par un sur-échantillonnage de facteur 2 (insertion d'un zéro (0) entre chaque échantillon), ensuite une addition (figure (II.6)).

II.4.7.4 Implémentation par banc de filtres

Une méthode équivalente et plus efficace pour calculer la transformée en ondelette est de convoluer le signal avec une paire de filtres miroirs en quadratures (QMF) convenablement choisis suivi par un sous-échantillonnage de facteur 2 ou décimation.

Les QMF qui décomposent ainsi le signal sont constitués d'un filtre passe-bas H_0 et

Chapitre II : Les différentes méthodes de compression

un filtre passe-haut H_1 . Ils divisent ainsi la bande passante du signal exactement en son milieu. Les coefficients sont recombinaés pour synthétiser le signal) (tx par la transformée en ondelette inverse. Elle est obtenus à l'aide d'une opération de sur-échantillonnage de facteur 2 suivie par la paire des filtres QMF G_0 et G_1 [17].

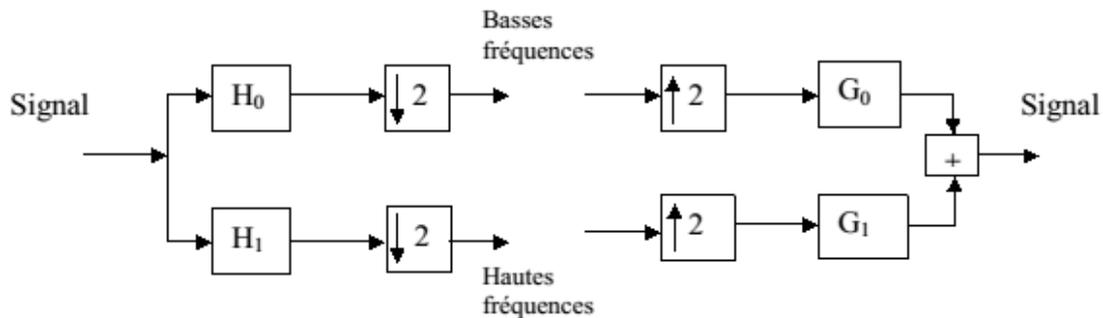


Figure II.6 : Filtrage par des filtres passe-bas et passe-haut

Décomposition en une dimension (1D) et deux Dimensions (2D) et niveaux de résolution

Pour le cas monodimensionnel (1D), les coefficients d'ondelettes à l'échelle $m+1$ sont calculés par application récursive de la transformée en ondelettes du signal passe bas à l'échelle m . Après k niveaux de décomposition on obtient k ensembles de coefficients hautes fréquences $n m C$, et une de bases fréquences, notée $n d$ [18].

Dans le cas bidimensionnel (2D), les coefficients d'ondelettes sont obtenus par application de la transformée en ondelettes (1D) sur les deux dimensions du signal à chaque niveaux de décomposition (lignes puis colonnes pour les images numériques).

Alors, après k niveaux de décomposition on obtient $3k$ parties (sous bandes) de coefficients hauts fréquences ou détails (à chaque niveau trois détails sont obtenus horizontales, verticales et diagonales) et une partie de basses fréquences appelée approximation ou bande de base.

C'est ce que l'on appelle habituelle l'analyse multi résolution. Il s'agit de l'application de l'algorithme de Mallat.

Le principe est de décomposer un signal en une approximation, appelée également la bande de base ou basse fréquence (L en anglais de "low"), et en ses détails ou composantes hautes fréquences. La même opération est répétée, toujours sur l'approximation uniquement, jusqu'à l'obtention de la résolution souhaitée. S'il s'agit d'une image nous aurons un signal 2D. Dans ce cas chaque niveau de décomposition sera obtenu en deux étapes.

Chapitre II : Les différentes méthodes de compression

La première étape concerne les lignes et la seconde étape les colonnes. Un niveau de décomposition sur une image nous donnera une approximation (Bande de base ou LL1 pour low-Low niveau1) et trois détails respectivement LH1, HL1 et HH1 (pour Low-High niveau1, High-Low niveau1 et High High niveau1) :

- LH1 : une décomposition basse fréquence selon lignes suivie par une décomposition haute fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition.
- HL1 : une décomposition haute fréquence selon lignes suivie par une décomposition basse fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition.
- HH1 : une décomposition haute fréquence selon lignes suivie par une décomposition haute fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition

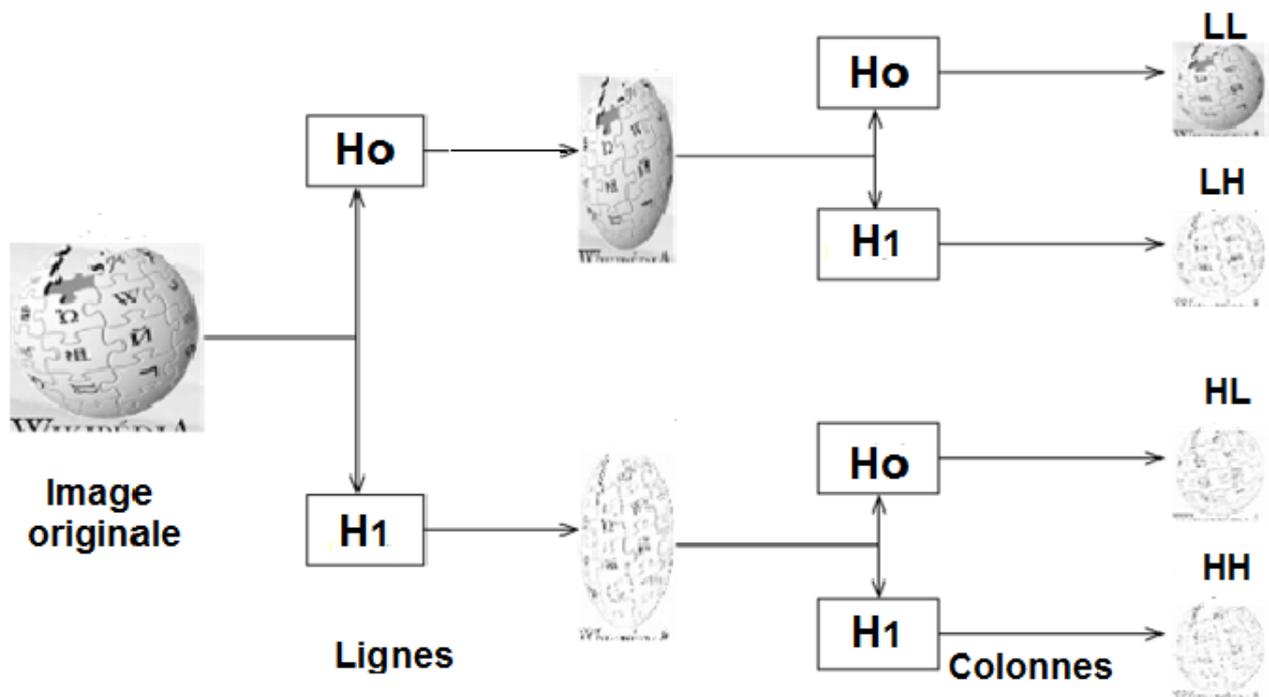


Figure II.7 : Principe de décomposition 2D par filtrage passe - bas (H_0) et passe - haut (H_1) dans le sens horizontal (Ligne) et vertical (Colonne).

Chapitre II : Les différentes méthodes de compression

3LL	3HL	Sous-bande Image détails Horizontaux	Sous-bande Image détails horizontaux 1HL
3LH	3HH	2HL	
Sous-bande Image détails verticaux 2LH		Sous-bande Image détails diagonaux 2HH	Sous-bande Image détails diagonaux 1HH
Sous-bande Image détails verticaux 1LH			

Figure II.8 : Diagramme de Mallat représentant les coefficients d'ondelettes de la transformée, classés par sous-bandes de filtrage et niveau de décomposition.

Les avantages de DWT

- Transformation de l'ensemble de l'image (introduise une mise à l'échelle inhérente).
- Meilleure identification des données pertinentes pour la perception humaine (taux de compression plus élevé).
- Calcul très rapide.
- Transformation rapide (surtout quand il est utilisé dans une image compressée).
- Elle ne besoin pas de diviser le codage d'entrée en blocs 2d non chevauchants.

Les inconvénients de DWT

- Temps de compression plus long.
- Le coût de calcul de DWT par rapport à DCT peut être plus élevé.
- Mauvaise sélectivité directionnelle pour les caractéristiques diagonales car les filtres d'ondelettes sont séparables et réel.

Chapitre II : Les différentes méthodes de compression

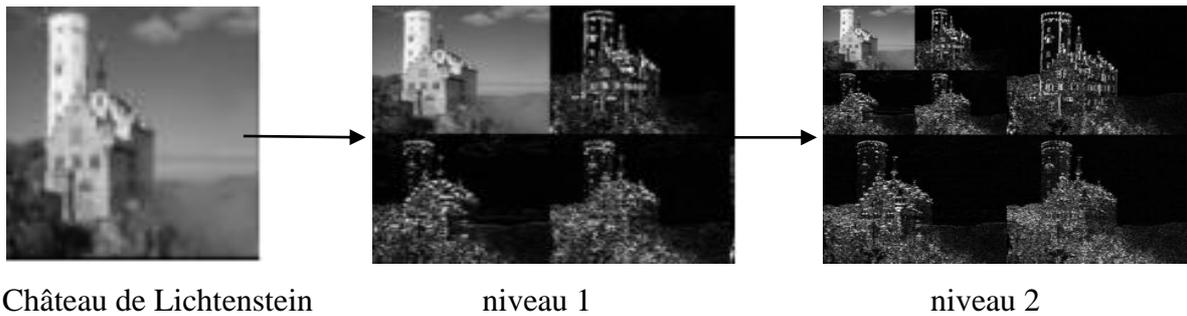


Image originale

Figure II.9 : Exemple d'une décomposition en ondelettes à 2 niveaux de résolution.

II.4.8 Compression par fractales :

Cette technique a été initialement promue par M.Barnsley, qui a fondé une société basée sur la technologie de compression d'image par fractale mais qui n'a pas publié de détails sur son schéma. Le premier schéma public a été attribué à E.Jacobs et R.Boss du Naval Ocean Systems Center à San Diego qui ont utilisé le partitionnement et la classification réguliers des segments de courbes afin de compresser les courbes fractales aléatoires en deux dimensions [19].

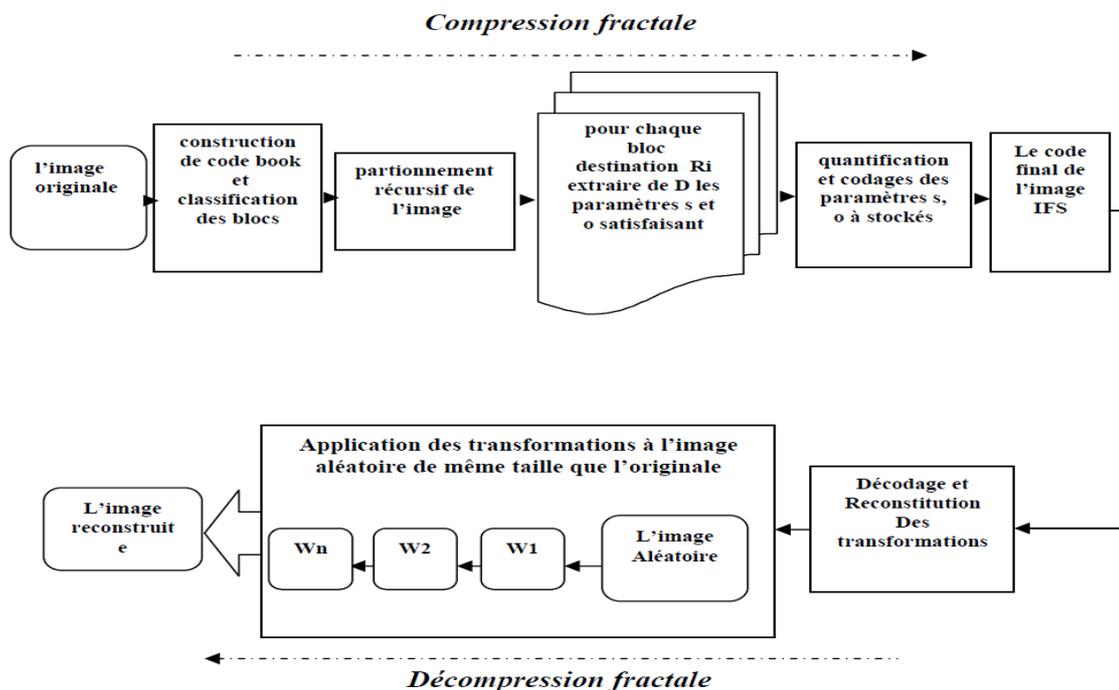


Figure II.10 : Le schéma général d'un codeur – décodeur

Chapitre II : Les différentes méthodes de compression

II.5 Conclusion

Après avoir présenté les techniques et les algorithmes de compression d'image, nous pouvons tenir les points suivants :

Nous distinguons deux types de compression avec pertes et compression sans perte suivant leur usage :

- Les données informatiques qui doivent rester identiques à leur original (Textes, programmes informatiques, ...) utilisent la compression sans perte (RLE, LZW, Huffman, ...).
- Les données dont la qualité se limite aux perceptions humaines (images,) utilisent la compression avec pertes ou sans perte (JPEG, JPEG2000, ...).

Chapitre III

Implémentation et résultat

Chapitre III: Implémentation et résultat

III.1 Introduction

Nous présentons dans ce chapitre, une étude comparative entre les algorithmes de la compression d'image JPEG et JPEG 2000 que nous avons implémenté avec différentes méthodes de compression (DWT, DCT) avec différents conditions , pour réaliser cette étude, nous utilisons MATLAB

III.2 Définition de Matlab

MATLAB est un langage de haute performance pour l'informatique technique. Il intègre le calcul, la visualisation et la programmation dans un environnement facile à utiliser où les problèmes et les solutions sont exprimés dans une notation mathématique familière. Les utilisations typiques comprennent:

- Mathématiques et calcul.
- Développement d'algorithme.
- Modélisation, simulation et prototypage.
- Analyse, exploration et visualisation des données.
- Graphiques scientifiques et techniques.
- Développement d'applications, y compris le bâtiment d'interface utilisateur graphique.

III.3 L'interface de programme d'application MATLAB (API).

Nous allons utiliser la version R2016a dans notre projet

Dans le premier, nous devons télécharger ou acheter le logiciel Matlab à partir du site officiel de Mathworks, et l'installer.

Après l'installation de Matlab, nous l'ouvrons, l'image au-dessous illustre l'interface de Matlab.

Chapitre III: Implémentation et résultat

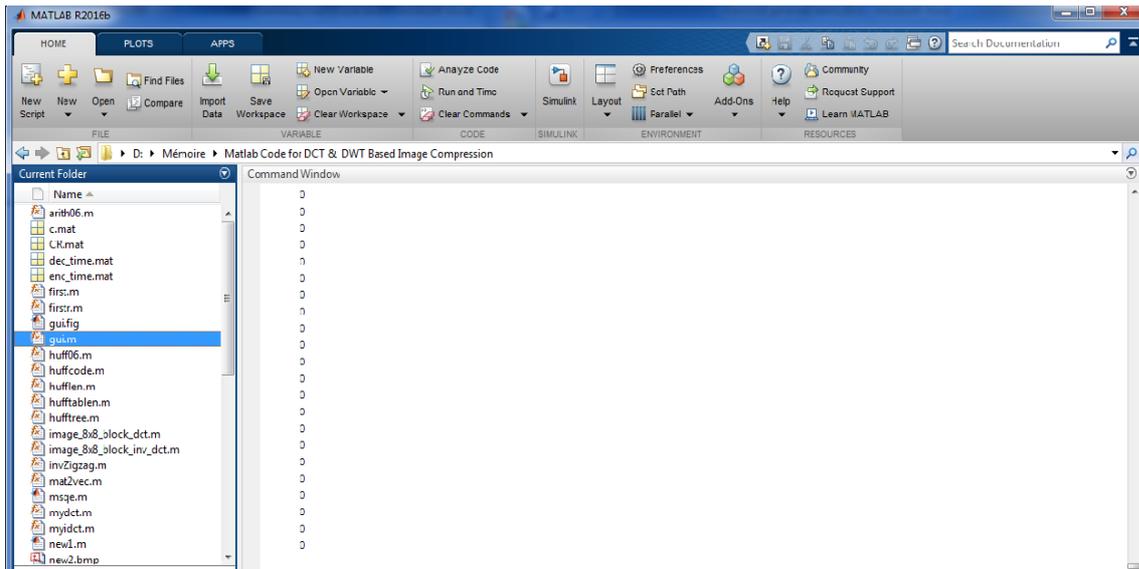


Figure III.1 : Interface graphique et IDE de MATLAB R2013a

III.4 Exécution de notre projet

On exécute notre fichier Image_compression.m de notre projet contient des fonctions qu'elle utilise la DCT (cosinus discret) et DWT (ondelette discrète) comme méthodes de compression d'image.

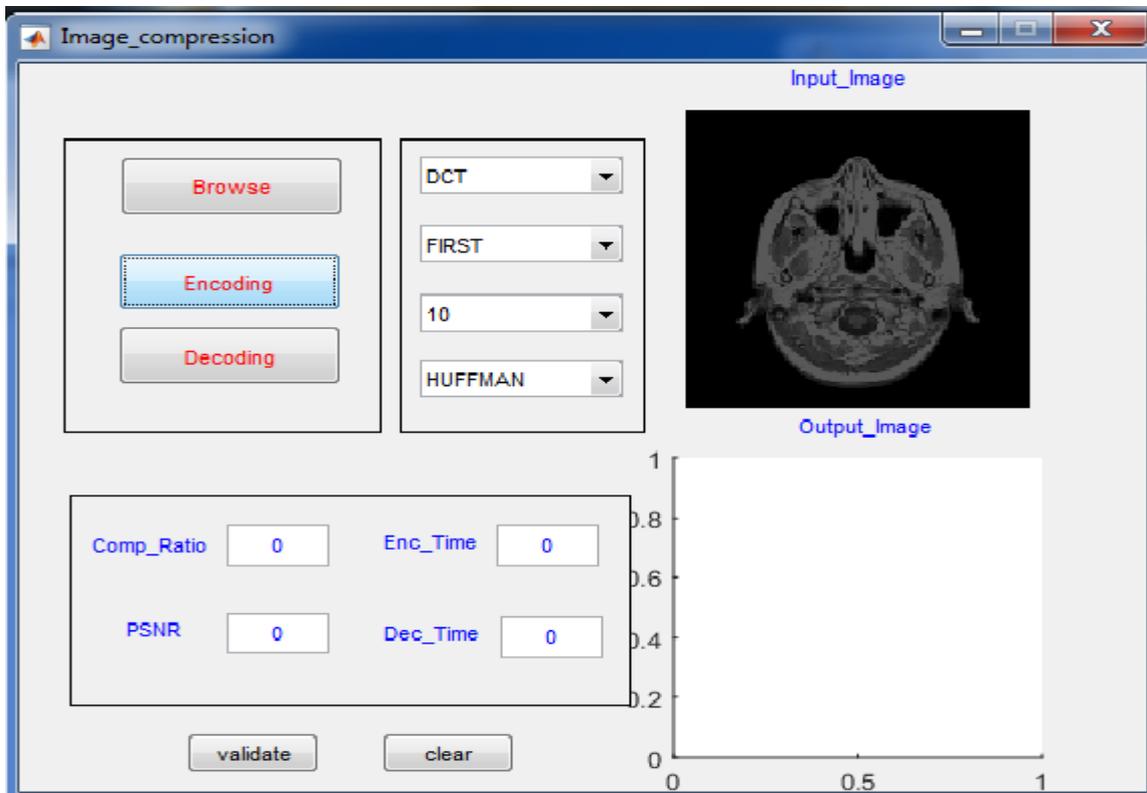


Figure III.2 : Le résultat d'exécution de code source de fichier Image_compression.m.

Chapitre III: Implémentation et résultat

III.5 Description de l'interface graphique

La photo au-dessus représente l'interface graphique de notre application de compression d'image. Au-dessus, on a trois boutons : le premier « Browse » pour parcourir et choisir une image qui existe déjà, le deuxième bouton « Encoding » pour compresser l'image choisie et le bouton « Decoding » pour décompresser. Et quatre menus déroulants : le premier pour choisir la méthode de compression (DCT ou DWT), le deuxième signifie le niveau de compression pour la DWT et le troisième pour choisir le niveau de quantification et le dernier il contient les deux algorithmes (Huffman, Arithmétique).

Au-dessous, on a quatre étiquettes et des zones de texte :

La première étiquette « comp_ratio » qui affiche le ratio de compression.

La deuxième PSNR qui affiche le rapport signal à bruit.

La troisième « Enc_Time » qui affiche le temps de compression d'image.

La quatrième « Dec_Time » qui affiche le temps de décompression d'image.

Et deux boutons : la première « Validate » pour afficher quatre résultats et la deuxième « Clear » pour effacer les données.

Et on a deux axes : le premier « Input_image » pour l'affichage de l'image originale et le deuxième « Output_image » pour afficher l'image compressée.

III.6 Analyse des résultats

Pour l'évaluation de la qualité des images après compression, on mesure la qualité de reconstruction de l'image par rapport à l'image originale (PSNR : rapport signal sur bruit) en fonction du taux de compression (CR) et de l'erreur quadratique moyenne (MSE), aussi le paramètre du temps de compression joue un rôle important dans cette évaluation, ainsi que l'aspect visuel de l'image.

Notre but c'est de trouver un bon compromis entre ces mesures, c'est à dire un fort PSNR et un CR minimal, et bien sûr sans pour autant perdre de la qualité visuelle de l'image originale, tout ça en respectant la contrainte du temps, plus le temps est court meilleur est le système.

Chapitre III: Implémentation et résultat

Performance de la compression :

➤ Taux de compression :

Etant donné que l'objectif de la compression est de minimiser la quantité de l'information indispensable pour la représentation d'une image, on définit le paramètre taux de compression CR (Compression Ratio) :

$$CR = \frac{\text{Nombre de bits avant la compression}}{\text{Nombre de bits après la compression}}$$

➤ Mesures de distorsion :

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.

La mesure de distorsion utilisée généralement en compression d'image est l'erreur quadratique moyenne MSE (Mean Square Error). Cette grandeur est définie par la moyenne des écarts aux carrés entre le pixel (i, j) de l'image originale est celui de l'image reconstruite suivant l'équation suivante :

$$MSE = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [I(i,j) - \tilde{I}(i,j)]^2}{N \times N}$$

Avec :

N x N : Taille de l'image.

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

$\tilde{I}(i, j)$: 3Pixel de l'image résultante.

On référence cette erreur par rapport à la dynamique de l'image en décibels. On obtient un rapport signal sur bruit crête pour une image (dans notre cas R=8), appelé PSNR (Peak Signal to Noise Ratio) qui se traduit par l'équation suivante :

$$PSNR = 10 \log_{10} \frac{[2^R - 1]^2}{MSE}$$

Ou: $2^R - 1$ représente la dynamique de l'image.

Chapitre III: Implémentation et résultat

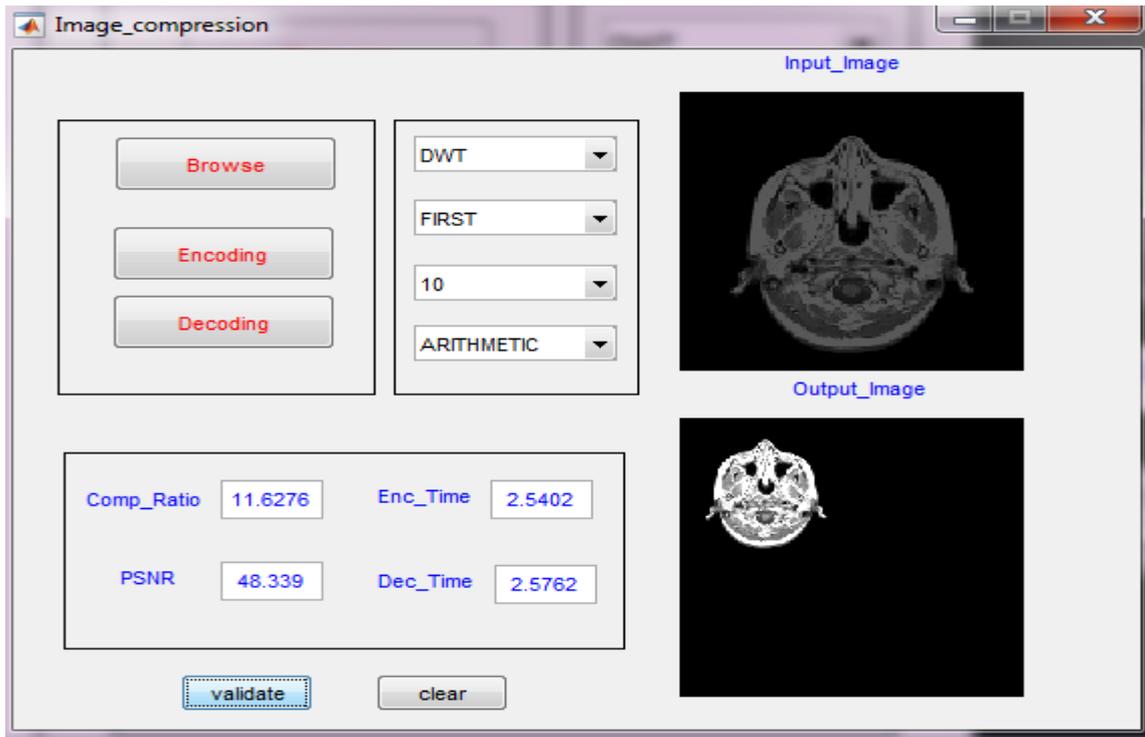


Figure III.3 : Le résultat d'exécution de bouton « Encoding »

Si on clique au bouton « Encoding » donc notre programme il nous donne l'image dans le premier niveau.

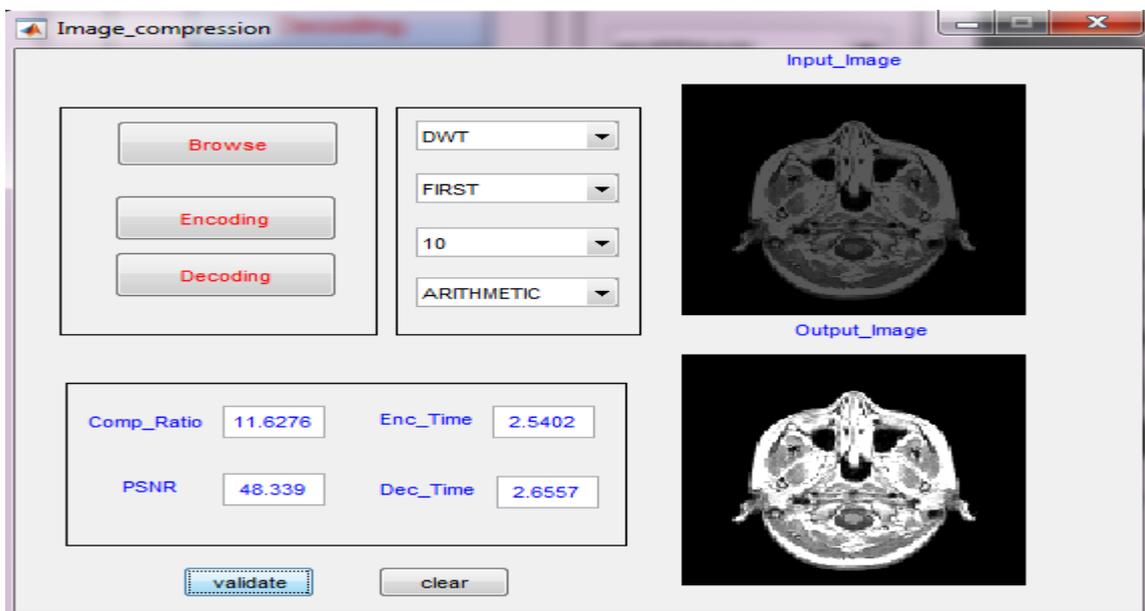


Figure III.4 : L'image compresser par DWT

Si on clique au bouton « Decoding » donc notre programme il nous donne l'image compressée par DWT .

Chapitre III: Implémentation et résultat

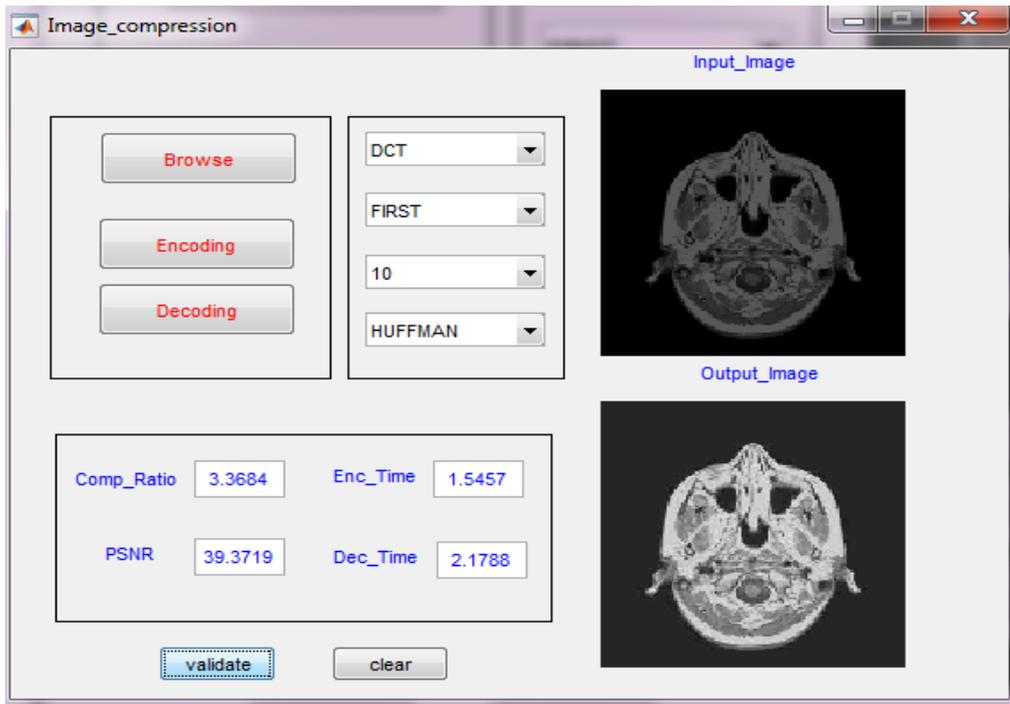


Figure III.5 : L'image compresser par DCT

Si on clique au bouton « Encoding » après « Decoding » donc notre programme il nous donne l'image compressée par DCT.

Comparaison visuel entre la photo compressée par DWT et par DCT

D'après la Comparaison visuel entre la photo compressée par DWT et par DCT avec l'algorithme de Huffman et le niveau de quantification 10 nous remarquons que l'image compressée par la DWT elle donne une belle qualité par rapport a l'image compressée par DCT.

	Compression DCT	Compression DWT avec nq=10, 'first level'
Comp_ratio (%)	3.3684	11.6276
PSNR (db)	39.3719	48.339
Enc_time (s)	1.5457	2.5402
Dec_time (s)	2.1788	2.6557

Tableau III.1 : Comparaisons entre compression DCT et DWT avec nq=10,first level

Chapitre III: Implémentation et résultat

III.6.1 Analyse des résultats à travers la table au-dessus

Nous remarquons que le CR de l'image compressée par la DCT est plus petit que le CR de l'image compressée par DWT et que PSNR du premier cas inférieur par rapport au deuxième cas et ça signifie que l'image compressée par DWT avait une bonne qualité que l'image compressée par la DCT.



Figure III.6 : figure présentative illustre l'image originale (a gauche) et l'image compressée avec DWT ('nq=10', 'first level') et l'image compressée avec DWT ('nq=90' 'first level' (a droite)).

Comparaison visuel entre la photo compressée par DWT avec différents mesures

D'après la Comparaison visuel entre la photo compressée par DWT avec les paramètres first level et niveau de quantification 10 et algorithme de Huffman et par DWT avec le niveau de quantification 90 et même mesures précédents nous remarquons que l'image compressée par les premiers mesures elle donne une belle qualité par rapport a l'image compressée par DWT avec les deuxièmes mesures.

	Compression DWT avec nq= 10' first level'	Compression DWT avec nq=90 'first level'
Comp_ratio (%)	11.6276	62.267
PSNR (db)	48.339	27.0211
Enc_time (s)	2.5402	0.35024
Dec_time (s)	2.6557	0.33854

Tableau III.2 : Comparaison entre compression DWT avec nq=10 et nq=90 'first level'.

Chapitre III: Implémentation et résultat

III.6.2 Analyse des résultats à travers du première table

Nous remarquons que le CR de l'image compressée par la DWT avec les paramètres first level et niveau de quantification 10 et algorithme de Huffman est plus petit que le CR de l'image compressée par DWT avec le niveau de quantification 90 et que PSNR du premier cas supérieur par rapport au deuxième cas et ça signifie que l'image compressée par DWT avec niveau 10 avait une bonne qualité que l'image compressée par la DWT avec niveau 90.



Figure III.7 : figure présentative illustre l'image originale (a gauche) et l'image compressée avec DWT ('nq=10', 'second level') et l'image compressée avec DWT ('nq=90', 'second level') (a droite)).

Comparaison visuel entre la photo compressée par DWT avec différents mesures

D'après la Comparaison visuel entre la photo compressée par DWT avec les paramètres second level et niveau de quantification 10 et algorithme de Huffman et par DWT avec le niveau de quantification 90 et même mesures précédents nous remarquons que l'image compressée par les premiers mesures elle donne une belle qualité par rapport a l'image compressée par DWT avec les deuxièmes mesures.

	Compression DWT avec nq=10' second level'	Compression DWT avec nq= 90 'second level'
Comp_ratio (%)	8.2386	47.5588
PSNR (db)	49.5739	28.6058
Enc_time (s)	1.2047	0.35636
Dec_time (s)	1.755	0.65155

Tableau III.3 : Comparaison entre compression DWT avec nq=10 et nq=90 'second level'.

Chapitre III: Implémentation et résultat

III.6.3 Analyse des résultats à travers du deuxième table

Nous remarquons que le CR de l'image compressée par la DWT avec les paramètres second level et niveau de quantification 10 et algorithme de Huffman est plus petit que le CR de l'image compressée par DWT avec le niveau de quantification 90 et que PSNR du premier cas supérieur par rapport au deuxième cas et ça signifie que l'image compressée par DWT avec niveau 10 avait une bonne qualité que l'image compressée par la DWT avec niveau 90.

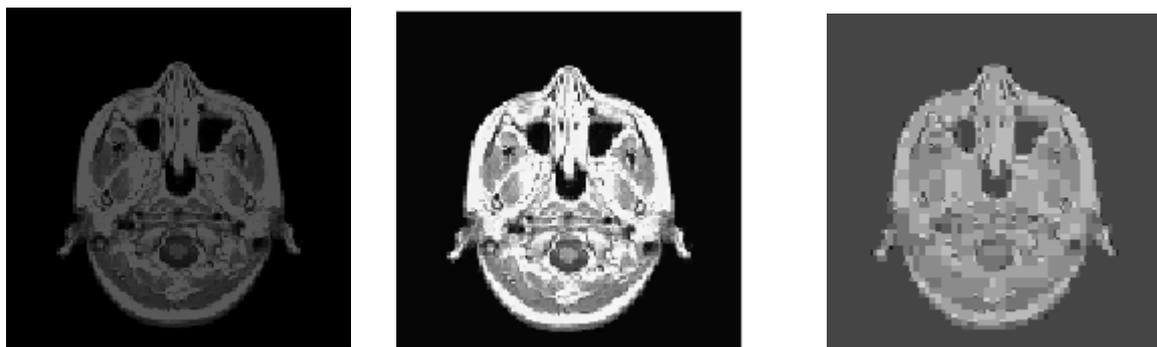


Figure III.8 : figure présentative illustre l'image originale (a gauche) et l'image compressée avec DWT ('nq=10', 'third level') et l'image compressée avec DWT ('nq=90' 'third level' (a droite)).

Comparaison visuel entre la photo compressée par DWT avec différents mesures

D'après la Comparaison visuel entre la photo compressée par DWT avec les paramètres third level et niveau de quantification 10 et algorithme de Huffman et par DWT avec le niveau de quantification 90 et même mesures précédents nous remarquons que l'image compressée par les premiers mesures elle donne une belle qualité par rapport a l'image compressée par DWT avec les deuxièmes mesures.

	Compression DWT avec nq=10 'third level'	Compression DWT avec nq=90 'third level'
Comp_ratio (%)	7.8315	43.9176
PSNR (db)	49.5106	28.7266
Enc_time (s)	1.224	0.43817
dec_time (s)	1.1075	0.31755

Tableau III.4 : Comparaison entre compression DWT avec nq=10 et nq=90 third

Chapitre III: Implémentation et résultat

III.6.4 Analyse des résultats à travers du troisième table

Nous remarquons que le CR de l'image compressée par la DWT avec les paramètres third level et niveau de quantification 10 et algorithme de Huffman est plus petit que le CR de l'image compressée par DWT avec le niveau de quantification 90 et que PSNR du premier cas supérieur par rapport au deuxième cas et ça signifie que l'image compressée par DWT avec niveau 10 avait une bonne qualité que l'image compressée par la DWT avec niveau 90.

III.6.5 Analyse final

D'après les comparaisons différentes entre les deux méthode (DWT,DCT) soit visuel ou résultantes nous concluons que la compression avec la méthode DWT est meilleure par rapport à la méthode DCT i.e. la méthode JPEG2000 meilleur que la méthode JPEG.

III.7 Conclusion

Dans notre projet expérimental nous concluons que la compression par l'ondelette est une meilleure solution de compression que la DCT avec autre manière la JPEG2000 respect bien le bute principale de la compression (minimisé le maximum de la taille originale sans perdre la qualité).

Nous avons surtout évalué les performances en termes de qualité visuelle de l'image après décompression entre le codec JPEG2000 par rapport à JPEG pour différents conditions et pour niveau de gris.

Les résultats obtenus montrent que la norme JPEG2000 est meilleure que la norme classique JPEG. Mais cette supériorité de la performance de JPEG2000 diminue à mesure qu'on augmente la valeur du taux de bits. On peut dire que la compression d'images par ondelettes permet d'atteindre de très bons résultats.

Conclusion générale

L'objectif de la compression d'image est de réduire la taille et de l'économiser l'espace disque. Dans ce travail, nous avons étudié différentes méthodes de compression d'images. Cependant, les algorithmes de compression sont spécifiques à certains types d'image, tels que les fichiers image, que nous avons vus dans notre projet de fin d'étude.

Dans le premier chapitre nous avons étudié les notions d'image. Nous avons cité les caractéristiques d'image et leurs différents types comme PNG, GIF et Tiff et jpeg .

Dans le deuxième chapitre nous avons étudié les différents méthodes de compression sans et avec perte d'informations nous avons présenté les algorithmes de compression suivante : algorithme de Huffman, Shannon Fano, arithmétique, RLE, LZW, jpeg, jpeg2000 et fractal.

Dans le dernier chapitre grâce aux différentes fonctions prédéfinies sur MATLAB nous avons fait une étude de performance de la technique de compression JPEG 2000 par rapport à JPEG avec différents conditions de compression.

Enfin, d'après les études qu'on a fait dans notre projet expérimental de fin d'étude nous déduisons que le domaine de la compression reste encore un sujet de recherche, et le but est d'arriver à une compression optimale en terme de temps de compression, de taux de bits et en terme de qualité de l'image reconstruite est toujours en course. La transformation en ondelettes reste une méthode à développer et à améliorer afin d'atteindre de meilleurs résultats par rapport à ce qui existe déjà.

Référence bibliographiques

- [1] C. Diltsm, Numériser vos collections, RCIP Octobre 2001.
http://www.chin.gc.ca/Francais/Contenu_Numerique/Guide_Gestionnaires/
- [2] C. Lepec, Les Graphiques Sur Ordinateurs, N. RIMOUX & A. COLIN Editeurs. Paris, 1991.
- [3] Paeth, A.W., “Image File Compression Made Easy”, in Graphics Gems II, James Arvo, editor. Academic Press, San Diego, 1991. ISBN 0-12-064480-0.
- [4] Ze-Nian Li and Mark S. Drew, ”Fundamentals of Multimedia”, Pearson Education, United States of America, 2006, p 560
- [5] Mark Nelson and Jean-loup Gailly, “The data compression book”_second edition, League for Programming Freedom, Cambridge MA 02139, 1995
- [6] R. Sylvain, "La Compression de Données", Club Photoshop de Nantes.
Conférence du 14 octobre 1999.
- [7] C. Benoit & A. Dusson, "La compression de données informatiques". Juin 1999. <http://www.esil.univ-mrs.fr/~cbenoit/projets/comp/>
- [8] Khalid Sayood. “Introduction to data compression”, 3rd edition, 500 Sansome Street, Suite 400, San Francisco, CA 94111 the, 2006, p680.
- [9] D. A. Huffman, “ A method for the construction of minimum - redundancy codes” , Proceedings of the Institute of Radio Engineers, vol. 40, pages 1098–1101. sept. 1952.
- [10] F. Davoine, Compression d'Images par Fractales Basée sur la Triangulation de Delaunay, Thèse l'INPG, Institut National Polytechnique de Grenoble, Décembre 1995.
- [11] J. F. Pillou, "Comment ça marche? L'informatique – Compression sSAsaqd'images", Année 1998. <http://www.commentcamarche.net/video/>
- [12] Hari M, “Compression Algorithms: Huffman and Lempel-Ziv-Welch (LZW)”, 6.02 Introduction to EECS II: Digital Communication Systems Fall 2012.
- [13] Robert Buckley, Ph.D, “JPEG 2000 - a Practical Digital Preservation Standard?”, Xerox Research Center Webster, New York, February 2008.
- [14] Min-Mo Sung, and Hee-Joung Kim, “ Clinical Evaluation of JPEG2000 Compression for Digital Mammography ” , IEEE Trans on Nuclear Science, vol .49, no. 3, June 2002.
- [15] M.D. Adams, “ The JPEG2000 Still Image Compression Standard”, ISO/IEC JTC1 / SC29/ WG1 (ITU-TSG16), December 2005.
- [16] Y. Kumar Jain, S. Jain, “ Performance Evaluation of Wavelets for Image Compression”, Asian Journal of International Technology 5(10) , pp. 113–130, 2002.
- [17] A. Boukaache ‘‘Compression d’images hautes définitions en utilisant une transformation hybride’’ Thèse de doctorat 2012 Guelma algérie.

Référence bibliographiques

[18] M.L. Hilton, "Wavelet and Wavelet Packet Compression of Electrocardiograms," IEEE Trans. Biomed. Eng., vol. 44, pp. 394-402, May 1997.

[19] R.D. Boss, E.W. Jacobs, "Fractals-Based Image Compression," NOSC Technical Report 1315, Sept. 1898. Naval Ocean Systems Center, San Diego CA 92152-5000.

Sites web

[s1] <https://www.computerhope.com/jargon/i/image.htm>

[s2] https://www.sites.univ-rennes2.fr/arts-spectacle/cian/image_numFlash/pdf/chap3_tout.pdf

[s3] <https://www.gralon.net/articles/photo-et-video/photo-et-video/article-les-pixels---definition-et-caracteristiques-2014.htm>

[s4] <http://www.kaddour.com/>

[s5] <http://arts-appliques.ac-orleans-tours.fr/php5/docs/gimp-poutaredy.pdf>

[s6] "http://membres.lycos.fr/compressions/menua.html", Octobre 2002.

[s7] site officiel de technopedia <https://www.techopedia.com/definition/792/bitmap-bmp> ,Mai 2017

[s8] https://fr.wikipedia.org/wiki/Graphics_Interchange_Format

[s9] <http://www.futura-sciences.com/tech/definitions/informatique-jpeg-1232/> ,Mai 2017

[s10] <thesis.univ-biskra.dz/2408/5/CHAPITRE%20III.pdf>

ملخص :

يشمل التطور الملحوظ في مجال تكنولوجيا المعلومات وتنوع تطبيقات الوسائط المتعددة في السنوات الأخيرة تطوير تقنيات ضغط الصور الأكثر كفاءة والتي تهدف إلى تحسين قدرة الإرسال وتخزين البيانات.

نقترح في هذا العمل أساليب الضغط المختلفة للصور، من أجل تحسين أداء الضغط بفضل المقارنة بين تحويل الموجات و تحويل جيب التمام المنفصل ، من وجهة نظر الجودة ومعدل ضغط الصورة

(PSNR et RC).

هذه الأساليب أسفرت عن نتائج مشجعة مع توقعات مثيرة للاهتمام.

كلمات مفتاحية : ضغط الصور والبيانات وتحويل الموجات ، تحويل جيب التمام منفصلة ، PSNR ، RC.

Résumé :

Le développement remarquable dans le domaine des technologies de l'information et la diversité des applications multimédias dans les dernières années implique le développement des techniques de compression d'image plus efficaces dont l'objectif est d'améliorer la capacité de transmission et le stockage des données.

Nous proposons dans ce travail les différentes méthodes de compression, afin d'améliorer les performances de compression grâce a la comparaison entre l'ondelette par rapport à la transformée en cosinus discrète, du point de vue qualité et taux de compression de l'image (PSNR et RC).

Ces méthodes a donné des résultats encourageants avec des perspectives intéressantes.

Mots-Clés : Compression d'image, l'ondelette, transformée en cosinus discrète, PSNR, RC.

Abstract:

The remarkable development in the field of information technology and the diversity of multimedia applications in recent years involves the development of more efficient image compression techniques whose objective is to improve the transmission capacity and the storage of data.

We propose in this work the various compression methods, in order to improve the compression performance thanks to the comparison between the wavelet with respect to the discrete cosine transform, from the point of view of quality and compression rate of the image (PSNR and RC).

These methods yielded encouraging results with interesting prospects.

Keywords: Image compression, data, wavelet, discrete cosine transform, PSNR, RC.