



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologies
Département de Génie- Electrique

MÉMOIRE DE MASTER

Sciences et Technologies
Télécommunications
Réseaux et Télécommunications

Réf. : /2020

Présenté et soutenu par :
Dadoua Salima

Le : mercredi 30 septembre 2020

Compression des images fixes par la Norme JPEG2000

Jury :

M.	Baarir Zine-Edine	Pr	Université de Biskra	Encadreur
M.	Sbaa Salim	Pr	Université de Biskra	Président
Mme.	Barkat Aicha	MAA	Université de Biskra	Examinatrice

Remerciements

Je tiens à exprimer toute ma reconnaissance à mon directeur monsieur ,
ZINE-EDINE BAARIR. Je le remercie de m'avoir encadrée, orientée, aidée et conseillée.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.

Je remercie mes très chers parents, saad & saaida , qui ont toujours été là pour moi. Je remercie mes sœurs Salwa et Sara , et mon frère Yassine, pour leur aide et encouragements.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

RÉSUMÉ

Le traitement de l'image a connu plusieurs évolutions importantes au niveau des algorithmes et des architectures. L'une de ces évolutions est l'apparition du nouveau standard ISO/IEC de compression d'images JPEG2000 qui succède à JPEG.

Ce nouveau standard présente de nombreuses fonctionnalités et caractéristiques qui lui permettent d'être adapté à une large panoplie d'applications. Mais, ces caractéristiques se sont accompagnées d'une complexité algorithmique beaucoup plus élevée que JPEG. La partie la plus complexe dans le processus de compression JPEG2000 est le codeur entropique qui constitue à lui seul environ 70% du temps de traitement global pour la compression d'une image.

L'objectif de ce travail est l'étude de la compression des images numériques par la technique des ondelettes, plus précisément la norme de compression d'images JPEG2000 (Joint Photographic Expert Group) basée sur les ondelettes. La norme de compression JPEG à base de la DCT (Transformée en Cosinus Discrète) a beaucoup de succès et reste encore une méthode de compression d'images très performante dans le domaine de l'Internet et des appareils photos numériques. Pour les faibles valeurs du taux de bits, la qualité de l'image se dégrade rapidement. JPEG2000 avait comme objectifs principaux de répondre aux divers besoins, des nouvelles applications et d'apporter une nette amélioration par rapport à JPEG. Les expérimentations proposées dans ce travail sont réalisées sous l'environnement MATLAB.

MOTS-CLÉS : Compression d'images, JPEG2000, Ondelettes, JPEG, DCT, DWT, DCT

TABLE DE MATIÈRE

Introduction générale.....X

Chapitre I: Généralités sur Les Images Numériques et La Compression des Données

Partie 01 :(généralités sur les images Numériques)

I-1. Introduction.....	2
I-2.Définition de l'image.....	2
I-3.Image Numérique.....	3
I-4.Caractéristiques d'une image numérique.....	3
I-4-1. Pixel	3
I-4-2.Dimension.....	4
I-4-3. Histogramme de l'image.....	4
I-4-4. Images aux Niveaux de Gris.....	4
I-4-5. Images en Couleurs.....	5
I-4-5-1. Images 24 bits (ou couleurs vraies)	5
I-4-5-2.Images à palettes, images en 256 couleurs (8bits).....	6
I-4-5-3. Images en teintes (ou niveaux) de gris	7
I-5. Images Bitmap et Images Vectorielles	7

Partie 02 : (La Compression des Données)

I-1. Introduction :Compression d'images	8
I-2.Principe général de la Compression des images.....	9
I-3.Classification des méthodes de compression.....	10
I-3-1.Méthodes de compression sans perte.....	10
I-3-1-1. Méthode de codage RLE.....	10
I-3-1-2. Codage Huffman.....	11
I-3-1-2-1. Principe	
I-3-1-2-2. Exemple de codage Huffman.....	
I-3-1-3. Shanon-fano.....	12
I-3-2. Méthodes de compression avec perte	12
I-3-3-Différences clés entre la Compression avec perte et la Compression sans perte.....	14
I-3-5-Conclusion sur la compression avec et sans perte.....	14
I-4.Evaluation de la compression.....	15
I-5.Qualité de reconstruction des images	15
I-6.Conclusion	17

Chapitre II:Etude de la norme de compression d'images fixes JPEG

II-1. Introduction.....	20
II-2. Fonctionnement général duJPEG.....	20
II-3. Transformation des couleurs	22
II-4.Sous-échantillonnage de la chrominance (Cb et Cr)	23

II-5. Découpage en blocs	25
II-6. Transformation DCT	26
II-6-1. Avantage de la DC.....	29
II-7. La Quantification	29
II-7-1. Linéarisation des blocs	30
II-8. Le codage entropique.....	31
II-8-1. Codage HUFFMAN.....	31
II-8-2. Codage Arithmétique.....	32
II-9. Décompression JPEG.....	33
II.10. Avantages et inconvénients du JPEG.....	33
II-10. Conclusion.....	34

Chapitre III: Etude de la norme de compression d'images fixes JPEG2000

III-1. Introduction.....	36
III-2- Généralités	36
III-3- Principe général	36
III-2. Fonctionnement général du JPEG2000.....	37
III-4- Modèle d'image utilisé par JPEG2000	38
III-4-1. Prétraitement	39
III-5. Transformée en couleur	40
III-6. Transformée en ondelettes	41
III-6-1. Lifting shema	43
III-6.2. Le lifting 2D.....	45
III-7. Quantification	45
III-8. Règle de Déquantification.....	46
III-9. Régions d'intérêt	46
III-10. Codage entropique.....	46
III-10.1. Codage Tier1	47
III-10-1-1. Codeur EBCOT	47
III-10-2. Codage Tiers 2 (Organisation du Bitstream).....	48
III-11. Contrôle du taux de compression.....	49
III-12. Résistance aux erreurs	49
III-13. Résumé des étapes de la création d'un fichier JPEG2000	50
III-14. Applications	50
III-15. Avantages de la compression JPEG2000	50
III-16. Différence entre JPEG et JPEG2000	51
III-17. Conclusion.....	52

Chapitre IV : Simulation , Résultats et Discussion

IV.1. Introduction	54
IV.2. Mesure de la qualité de la restauration	54
IV.2.1 Erreur quadratique moyenne.....	
IV.2.2. Signal to noise ratio (SNR)	
IV.2.3. Peak Signal to Noise Ratio (PSNR).....	
IV.3. SNR et PSNR.....	55
IV.4. Exécution de notre projet	55
IV-4-1. Tests d'images au niveau de gris	55
IV.4.2. Test avec différents niveaux de décomposition en ondelette et code-blocs.....	57
IV.4.3. Test sur une image couleur.....	59

IV.4.4. Test Scalabilité en qualité sur une image au niveau de gris Lena	61
IV.4.5. Test Région d'intérêt (ROI) sur une image au niveau de gris Lena	62
IV.5.conclusion.....	63
Conclusion générale.....	64

LISTE DES FIGURES

Figure 1.1 : Point de l'image de coordonnées (x, y).

Figure 1.2 : Agrandissement d'un bloc de 8 pixels sur 8 pixels

Figure 1.3 : Représentation de la lettre A sous la forme d'un groupe de pixels.

Figure 1.4 : Représentation de dimension d'une image

Figure 1.5 : Image et histogramme Associés

Figure 1.6 : Image en mode niveau de gris.

Figure 1.7 : Image en mode couleur.

Figure 1.8 : Schéma général de la compression

Figure 1.9 : Types et méthodes de compression d'images

Figure 1.10 : Principes général de la compression

Figure 1.11 : Un exemple de codage par plage RLE.

Figure 1.12 : Un exemple d'arbre de Huffman, généré avec la phrase « this is an example of a huffman tree ».

Figure 2.1 : Principe général de la compression/décompression JPEG.

Figure 2.2.1 : RGB à YcrCb

Figure 2.2.2 : formule de passage de RGB à YCbCr.

Figure 2.2.3: formule de passage de YCbCr à RGB.

Figure 2.3.1 : Sous-échantillonnage 4:4:4

Figure 2.3.2 : Sous-échantillonnage 4:2:2

Figure 2.3.3 : Sous-échantillonnage 4:2:0

Figure 2.4: Types de formats de sous – échantillonnage dans JPEG

Figure 2.5 : Exemple de sous-échantillonnage de type 4:2:0.

Figure 2.6 : Equation de DCT.

Figure 2.7: DCT inverse.

Figure 2.8: DCT inverse.

Figure 2.8 : bloc DCT.

Figure 2.9: Schéma fonctionnel de l'algorithme JPEG.

Figure 2.10 : Image Originale.

Figure 2.11 : Bloc de 8×8 pixels uniforme.

Figure 2.12 : Représentation de la DCT d'un bloc de 8×8 pixels.

Figure 2.13 : Lecture en Séquence Zig-Zag.

Figure 2.14 :Schéma Lecture en Séquence Zig-Zag

Figure 2.15 : Schéma de codage.

Figure 3.1 : Fonctionnement général du JPEG2000

Figure 3.2 : Représentation d'une image composée de N composantes numérotées de 0 à N-1.

Figure 3.3: Division en tuiles, décalage en DC et DWT sur chaque composante.

Figure 3.4 : Codage des Composantes de Couleur.

Figure 3.5 : Transformée en ondelettes.

Figure 3.6: Décomposition d'une image en ondelettes en deux niveaux.

Figure 3.7 : Banc de filtres d'analyse pour la transformée en Ondelettes Lors du décodage.

Figure 3.8: Transformée ondelettes inverse

Figure 3.9 : Etapes de compression d'une image par DWT.

Figure 3.10 : Principe du schéma lifting.

Figure 3.11 : Schéma classique de décomposition-reconstruction.

Figure 3.12 : Représentation polyphasée.

Figure 3.13: Localisation de la ROI

Figure 3.14.1: Partitionnement des sous bandes en code-blocs.

Figure 3.14.2 : Représentation d'un code –bloc en plan de bits.

Figure 3.14.3 : Ordre de balayage d'un plan de bits.

Figure 3.14.4 : Illustration du concept de couches de qualité

Figure 3.15 : Résumé des étapes de la création d'un fichier JPEG2000

Figure 3.16 :Différence entre JPEG et JPEG2000

Figure 4.1 : Interface graphique et IDE de MATLAB R2013a.

Figure 4.2: PSNR (dB) en fonction des différents Tb (bpp) pour l'image LENA.

Figure 4.3 : Image 'Lena' reconstruite après une compression à 0.25 bpp à gauche p Figure 4.4 : Test avec différents niveaux de décomposition en ondelettes et code-blocsar le codeur JPEG et à droite par le codeur JPEG2000 une image couleur.

Figure 4.5 : Image originale 'Lena .Bmp'.

Figure 4.6 : PSNR (dB) pour chaque composante (Rouge, Vert, Bleu) en fonction des différents Tb (bpp) obtenus sur l'image Lena.

Figure 4.6.1 : Image 'Lena' reconstruite après une compression à 0.125.

Figure 4.6.2 : Image 'Lena' reconstruite après une compression à 0.25 bpp.

Figure 4.7.1 : Image 'Lena' reconstruite À 0.125 bpp par le codeur JPEG2000.

Figure 4.7.2 : Image 'Lena' reconstruite À 0.25 bpp par le codeur JPEG2000

Figure 4.7.3 : Image 'Lena' reconstruite À 0.5 bpp par le codeur JPEG2000

Figure 4.7.4 : Image 'Lena' reconstruite À 1.0 bpp par le codeur JPEG2000

Figure 4.8: Image 'Lena à 0.255 bpp de la région d'intérêt par le codeur JPEG2000.

LISTE DES TABLEAUX

Tableau (1.1) : couleurs vraies de l'image

Tableau (1.2) : Table de comparaison entre la compression avec et sans pertes

Tableau (2.3) : Tableau d'avantages et inconvénients des techniques de codage

Tableau (3.1) : Transformation directe et inverse des plans de couleur pour le mode ICT (irréversible) et pour le mode RCT (réversible) en (YUV).

Tableau (3.2) : Différence entre JPEG et JPEG2000

Tableau(4.1): Présentation des résultats obtenus en terme de PSNR (dB) sur l'image LENA de test.

Tableau (4.2) : Résultats du PSNR avec variation de la taille des code-blocs et avec différents niveaux de décomposition en ondelettes.

Tableau (4.3) : Résultats des PSNRs sur l'image Lena obtenus avec les deux codecs pour différents Tb (bpp).

LISTE D'ABRÉVIATIONS

- BPP : Bit Par Pixel.
- DCT : Transformée en Cosinus Discrète.
- DWT : Transformée en ondelettes.
- EQM : Erreur Quadratique Moyenne.
- EZW: Embedded Zero tree Wavelets.
- JPEG: Joint Photographic Expert Group.
- JPEG2000 Standard de compression d'images fixes récent, introduit par JPEG en 2000
- PSNR : Rapport Signal sur bruit crête.
- ROI : Region Of Interest.
- RC : Rapport de Compression.
- RLE : Codage de Longueur de séquence.
- SNR : Rapport Signal sur bruit.
- EBCOT : Embedded Block Coding with Optimized Truncation)
- EQM : Erreur Quadratique Moyenne
- ISO : International Standard Organization
- g: Filtre d'analyse passe-haut
- g_{\sim} : Filtre de synthèse passe-haut
- h : Filtre d'analyse passe-bas
- h_{\sim} : Filtre de synthèse passe-bas

INTRODUCTION GÉNÉRALE

L'objectif principal du travail présenté dans ce mémoire est l'étude du nouveau système de codage d'image fixe JPEG2000 fondée sur les ondelettes et leurs propriétés.

De nos jours l'image est un support d'information très performant, une image vaut plus que mille mots. Vu l'importance de l'image, et la grande quantité d'information qu'elle peut contenir, le monde s'intéresse de plus en plus à l'image et tend vers l'universalisation de son utilisation. En effet, l'image a touché plusieurs domaines de notre vie : la médecine, la météo, les télécommunications, la cartographie, la géologie, etc.....

Avec le développement de l'outil informatique, plusieurs techniques de traitement des images ont vu le jour [1]. Parmi les nombreux traitements qu'on peut effectuer sur les images, on trouve l'opération de compression des images. Cette opération devient plus que nécessaire vu le volume important d'informations mis en œuvre lors de l'utilisation des images numériques.

La contrepartie réside dans une quantité de données générées considérable qui peut rapidement saturer les systèmes conventionnels de transmission et de stockage ^[1]. A titre d'exemple, le CHU de Nancy-Brabois produit des examens de type Pet-Scan où les données sont parfois produites à partir d'une acquisition allant de la tête aux pieds du patient. Ainsi, un seul de ces examens peut à lui seul dépasser aisément le Giga octet. La nécessité de compresser les images apparaît donc aujourd'hui incontournable. De plus, la compression présente un intérêt évident pour la transmission des images qui peut s'avérer délicate du fait des bandes passantes existantes limitées.

L'idée de base de la compression des 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 pertes d'information. Au-delà, il est nécessaire d'élaborer des algorithmes de compression irréversibles (avec pertes) induisant une distorsion peu visible. Dans ce travail, nous allons étudier la norme de compression JPEG et nous montrerons ses avantages et limitations puis nous introduirons la nouvelle norme de compression JPEG2000 que nous comparerons à JPEG. Les expérimentations proposées dans ce travail sont réalisées sous l'environnement MATLAB.

Ce mémoire est composé de trois chapitres :

Le Premier chapitre présente les fondements de base de la compression, du codage utilisé et des critères d'évaluation d'une compression d'images.

Le Deuxième chapitre, décrit le fonctionnement général du JPEG.

Le Troisième chapitre, présente la compression des images numériques fixes en niveaux de gris et couleur par la technique des ondelettes, plus précisément la norme de compression d'images JPEG 2000 (Joint Photographic Expert Group) basée sur les ondelettes. Cette nouvelle méthode proposée en 2000, a reçu une attention particulière d'un grand nombre de chercheurs .

CHAPITRE I

Généralités sur Les Images Numériques et La Compression des Données

Partie 01 : Généralités sur Les Images Numérique.

I-1. Introduction :

L'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre.

C'est aussi le moyen le plus efficace pour communiquer ; chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

De ce fait, le traitement d'image est un ensemble de méthodes et techniques opérant sur l'image, dont le but est d'améliorer son aspect visuel. Il se définit comme un ensemble de tâches destinées à extraire de l'image des informations qualitatives et quantitatives.

L'image est un support d'information très performant, et comme on dit : une image vaut plus que mille mots. Vu l'importance de l'image, ^[1] et la grande quantité d'informations qu'elle peut contenir, le monde s'intéresse de plus en plus à l'image et tend vers l'universalisation de son utilisation. En effet, l'image a touché plusieurs domaines de notre vie : la médecine, la météo les télécommunications, la cartographie, la géologie,...etc.

I-2. Définition de l'image :

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture le dessin la photographie, le film,...etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain. ^[2]

Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique continue définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation ^[3]..

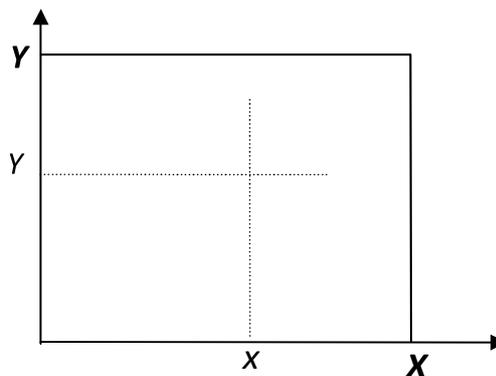


Figure 1.1 : Un point de l'image de coordonnées (x, y) .

I-2-1. Image Numérique :

Contrairement aux images obtenues à l'aide d'un appareil photo (analogique), ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits). L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés pixels.

- Une image numérique est constituée de points élémentaires (pixels) ayant chacun une couleur codée généralement sur 8 bits (entre 0 et 255) représentant l'intensité de la couleur.
- Dans une image couleur, pour chaque pixel, on représente l'intensité des 3 couleurs de base Rouge, Vert et Bleu (RGB pour les anglo-saxons).
- Une image couleur plein écran représente $1024 \times 768 \times 3$ soit près de 2 360 000 octets.
- Une page A4 scannée à 600 points par pouce peut dépasser 20 Méga octets...etc.
- Il faut donc compresser les images, c'est-à-dire réduire leur taille.
- Sur l'image ci-dessous on a isolé un bloc de 8 pixels sur 8 pixels et on l'a agrandi.

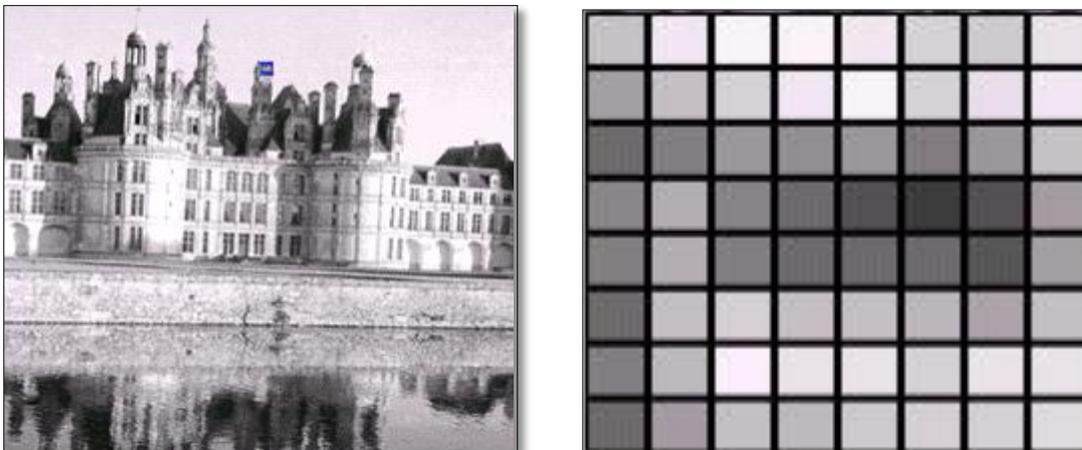


Figure 1.2 : Agrandissement d'un bloc de 8 pixels sur 8 pixels

- Ce bloc est marqué en bleu et correspond au sommet la cheminée au centre de l'image.
- Pour les images couleurs, on traite séparément chaque couleur, ce qui revient à traiter trois images noir et blanc.

I-3. Caractéristiques d'une image numérique :

I-3-1. Pixel :

Le pixel est la contraction de l'expression anglaise " picture element". Le pixel, étant le plus petit point de l'image, C'est une entité calculable qui peut recevoir une structure et une quantification.

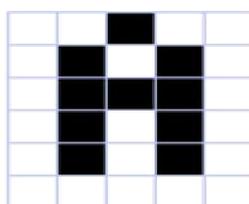


Figure 1.3 : Représentation de la lettre A sous la forme d'un groupe de pixels.

La quantité d'informations que véhicule chaque pixel donne des nuances entre images monochromes images en niveaux de gris et images couleurs

Dans chaque cas les pixels sont codés différemment

I-3-2. 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. [5]

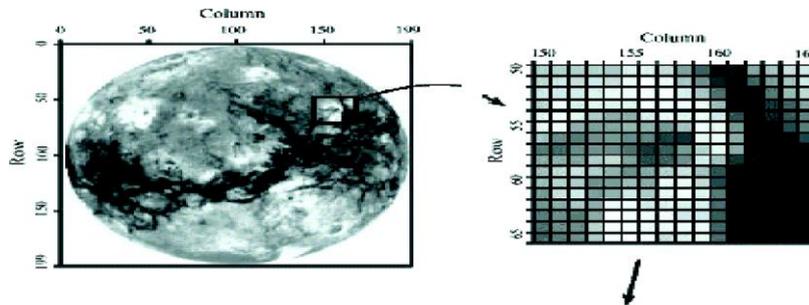


Figure 1.4 : Représentation de dimension d'une image.

I-3-3. Histogramme de l'image :

On veut souvent avoir une information sur la distribution des valeurs des pixels (niveaux) dans une image ; pour cela ;on utilise souvent une table qui donne le nombre de pixels de chaque niveau dans l'image. Cette table, souvent représentée graphiquement, est appelée Histogramme de l'image notée $h(v)$.

L'histogramme permet de donner un grand nombre d'informations sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée. [6]

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci

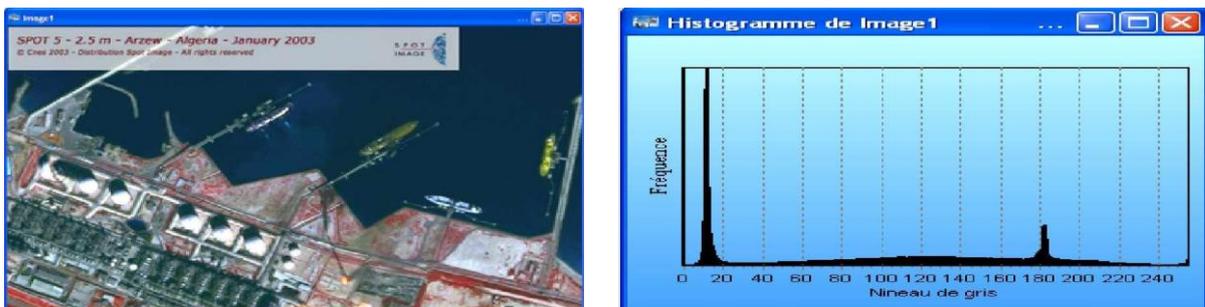


Figure 1.5 : Image et histogramme Associés

I-3-4. Images en 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.

Donc pour représenter les images en 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 nulle) 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. [7]

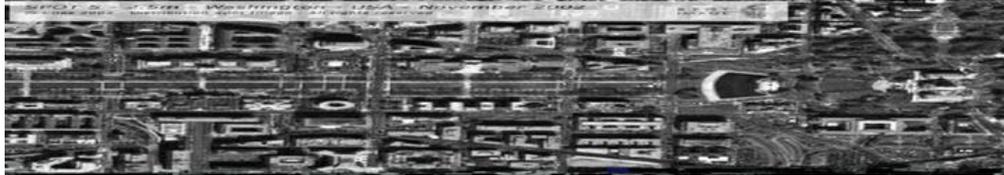


Figure 1.6 : Image en mode niveau de gris.

I-3-5. Images en Couleurs :

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc ou en niveaux de gris, les applications multimédias utilisent le plus souvent des images en couleurs (figure 1.6). 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.



Figure 1.7 : Image en mode couleur.

Il existe plusieurs types de représentation des images couleurs, parmi celle-ci nous trouvons :

I-3-5-1. Images 24 bits (ou « couleurs vraies ») :

Il s'agit d'une appellation trompeuse car le monde numérique (fini, limité) ne peut pas rendre compte intégralement de la réalité (infinie). Le codage de la couleur est réalisé sur trois octets, chaque octet représentant la valeur d'une composante couleur par un entier de 0 à 255. Ces trois valeurs codent généralement la couleur dans l'espace RVB. Le nombre de couleurs différentes pouvant être ainsi représenté est de $256 \times 256 \times 256$ possibilités, soit environ 16,7 millions de couleurs. Comme la différence de nuance entre deux couleurs très proches mais différentes dans ce mode de représentation est quasiment imperceptible pour l'œil humain, on considère commodément que ce système permet une restitution exacte des couleurs, c'est pourquoi on parle de « couleurs vraies » [8]

R	V	B	Couleur
0	0	0	Noir
0	0	1	nuance de noir
255	0	0	Rouge
0	255	0	Vert
0	0	255	Bleu
128	128	128	Gris
255	255	255	Blanc

Tableau 1.1 : couleurs vraies de l'image

I-3-5-2. Images à palettes, images en 256 couleurs (8 bits) :

Pour réduire la place occupée par l'information de couleur, on utilise une palette de couleurs « attachée » à l'image. On parle alors de couleurs indexées : la valeur associée à un pixel ne véhicule plus la couleur effective du pixel, mais renvoie à l'entrée correspondant à cette valeur dans une table (ou palette) de couleurs appelée look-up table ou LUT en anglais, dans laquelle on dispose de la représentation complète de la couleur considérée.

Selon le nombre de couleurs présentes dans l'image, on peut ainsi gagner une place non négligeable : on considère en pratique que 256 couleurs parmi les 16 millions de couleurs 24 bits sont suffisantes. Pour les coder, on aura donc une palette occupant $24 \text{ bits} \times 256 \text{ entrées}$, soit 3×256 octets, et les pixels de l'image seront associés à des index codés sur un octet. L'occupation d'une telle image est donc de 1 octet par pixel plus la LUT, ce qui représente un peu plus du tiers de la place occupée par une image en couleurs 24 bits (plus l'image contient de pixels, plus le gain de place est important, la limite étant le tiers de la place occupée par l'image en couleurs vraies).

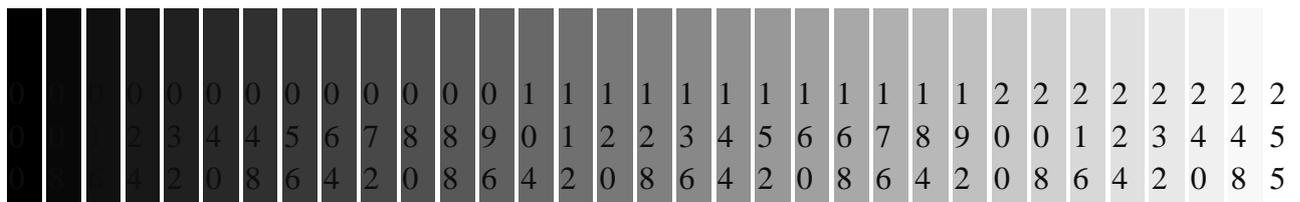
Une autre méthode existante consiste à se passer de palette, et de coder directement les trois couleurs en utilisant un octet : chaque composante couleur est codée sur deux bits, le bit restant peut servir soit à gérer plus de couleurs sur une des composantes, soit à gérer la transparence du pixel.

Avec cette méthode, on obtient des images bitmap avec un codage couleur effectivement limité à 8 bits, bien que la plage des couleurs possibles soit très réduite par rapport à celle qu'offre la méthode utilisant une palette.

Dans le cas des images en couleurs indexées, il est possible de spécifier que les pixels utilisant une des couleurs de la palette ne soient pas affichés lors de la lecture des données de l'image. Cette propriété de transparence est très utilisée (et utile) pour les images des pages web, afin que la couleur de fond de l'image n'empêche pas la visualisation de l'arrière-plan de la page. [9]

I-3-5-3. Images en teintes (ou niveaux) de gris :

On ne code ici plus que le niveau de l'intensité lumineuse, généralement sur un octet (256 valeurs). Par convention, la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale) :



Ce procédé est fréquemment utilisé pour reproduire des photos en noir et blanc ou du texte dans certaines conditions (avec utilisation d'un filtre pour adoucir les contours afin d'obtenir des caractères plus lisses).

Ce codage de la simple intensité lumineuse est également utilisé pour le codage d'images couleurs : l'image est représentée par trois images d'intensité lumineuses, chacune se situant dans une composante distincte de l'espace colorimétrique (par exemple, intensité de rouge, de vert et de bleu). [10]

I-4. Images Bitmap et Images Vectorielles :

D'un point de vue informatique, l'image est avant tout un fichier numérique qui va reconstituer un rendu visuel au travers de savants calculs. A la base, il existe deux technologies bien distinctes pour produire ce rendu, ayant chacune leurs avantages et inconvénients mais aussi leurs logiciels dédiés : d'un côté l'image **Bitmap**, de l'autre l'image **Vectorielle**.

Image Bitmap :

C'est certainement celle la plus répandue et la plus couramment utilisée. Elle est tout simplement constituée d'une grille de milliers de **Pixels** représentant les points de couleur successifs de l'image. On pourrait comparer cette technique à celle du pointillisme.

Chaque Pixel est alors un tout petit carré ayant une place bien définie avec sa propre couleur et l'image devient alors un quadrillage de Pixels qui, placés les uns à côté des autres, restituent le rendu visuel.

Image Vectorielle :

elle 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.

Autrement dit, pour afficher une ligne par exemple, le logiciel détermine le point de départ, le point d'arrivée puis la trajectoire à suivre. Ensuite, il calcule et positionne l'ensemble des pixels nécessaire pour afficher cette ligne. Il en va de même pour des formes et des couleurs plus complexes.

Partie 02 : Compression d'Image

I-1. Introduction :

La compression des données, d'une manière générale est l'ensemble des méthodes et règles qui permettent de réduire le volume d'une donnée sans perdre les informations essentielles.

L'enjeu de la recherche sur la compression d'images est de trouver un moyen de diminuer la taille d'une image, tout en essayant de limiter la dégradation due à la compression.

Le schéma général souvent utilisé pour décrire le fonctionnement des algorithmes de compression est celui présenté dans la figure 1.8 :

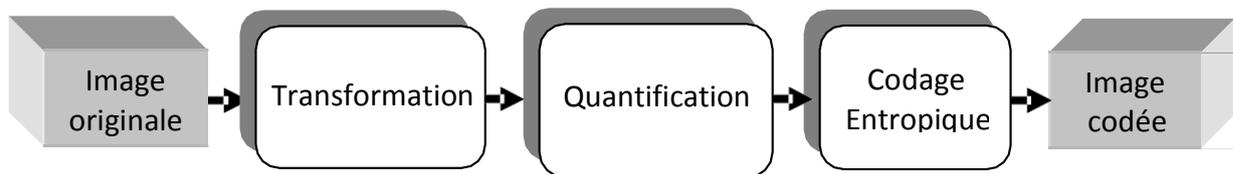


Figure 1.8 : Schéma général de la compression

Le but de la compression est de représenter les informations avec une forme plus compacte que l'original ou le résultat de la compression occupe moins d'espace que la donnée originale. Les données peuvent être compressées avec pertes ou sans pertes. [11]

Il existe deux types de compression :

- **Compression sans pertes** : la reconstruction des données est identique aux originales mais le taux de compression est très faible.
- Pas de modification de l'image.
- Ne change que la façon dont elle est codée sur disque.
- Affichage (= décompression) de l'image : identique à l'originale.
- **Compression avec pertes** : il ya suppression des informations les moins indispensables pour l'œil humain.
- Réduction du nombre de données

- Taux de compression plus élevé que dans la compression sans pertes.
Plus le taux de compression est élevé, plus le niveau de pertes est important et plus la qualité d'image est dégradée. [12]

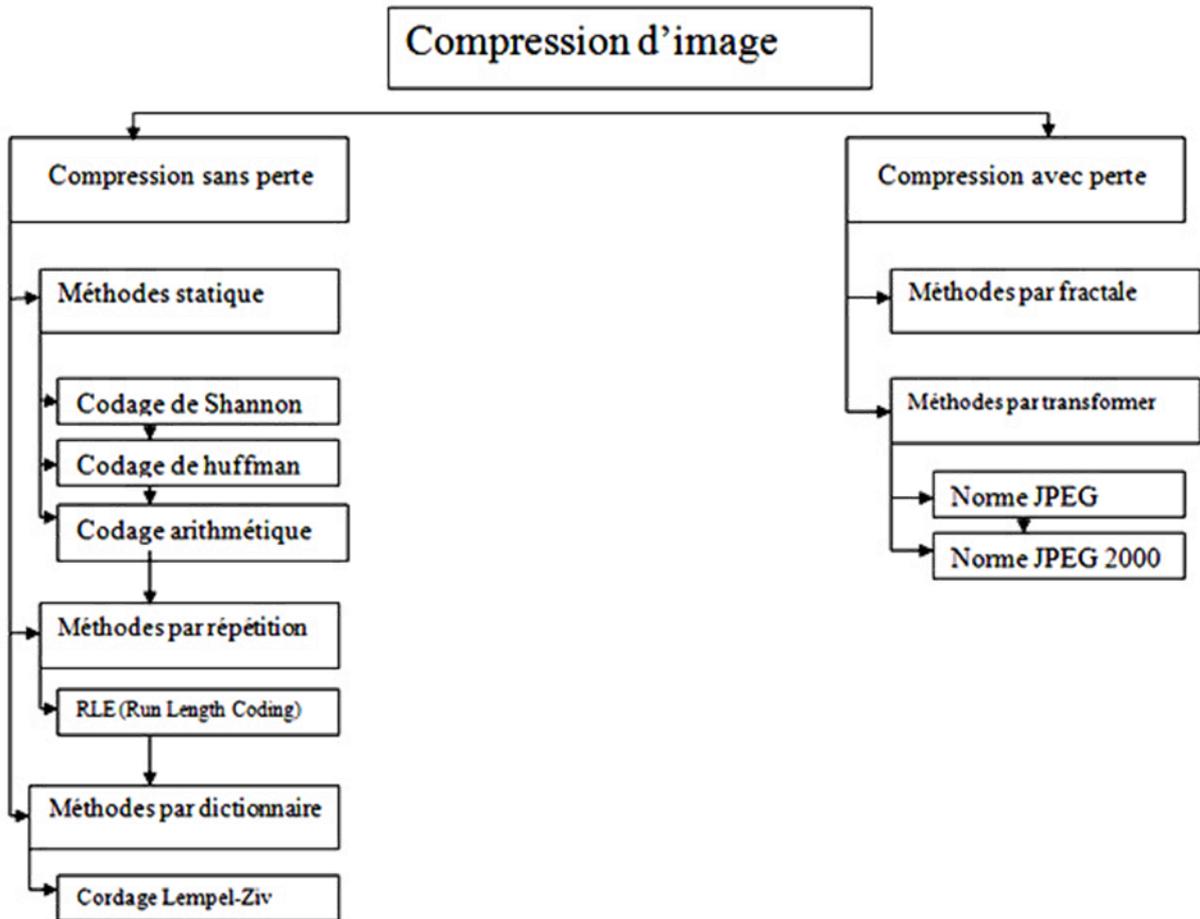


Figure 1.9 :Méthodes de compression d'image.

I-2- Principe général de la Compression des images

La compression d'une image se fait en général selon le schéma synoptique suivant. [13]



Figure 1.10 : Principes général de la compression.

Les méthodes de compression d'image ont des critères d'évaluation, les principaux critères sont [14] :

- _ La qualité de reconstitution de l'image.

- _ Le taux de compression.
- _ La rapidité du codeur.

1) Décorrélation :

La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte sur l'image. On essaie donc de tirer partie de cette corrélation, pour réduire le volume d'information en effectuant une opération de décorrélation des pixels. La décorrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible [15].

2) Quantification :

La quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. Elle représente une étape clé de la compression. Elle approxime chaque valeur d'un signal par un multiple entier d'une quantité q , appelée quantum élémentaire ou pas de quantification. Elle peut être scalaire ou vectorielle.

3) Codage :

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

- Unicité : deux messages différents ne doivent pas être codés de la même façon.
- Déchiffrabilité : deux mots de codes successifs doivent être distingués sans ambiguïté.

I-3- Classification des méthodes de compression :

Les méthodes de compression d'images peuvent être classées en deux grandes classes. Les méthodes de compression sans perte d'informations (réversibles) et les méthodes de compression avec perte d'informations (irréversibles).

I-3-1-Compression sans pertes :

Une compression est dite sans pertes si les données après décompression sont identiques aux données originelles.

Les méthodes de compression se basent toutes sur le même principe : la répétition d'une donnée est une répétition de trop.

L'objectif va être de supprimer le maximum de répétition pour obtenir une compression plus importante, moins connue du grand public, sont aussi très nombreux. Nous retrouvons par exemple le codage de Huffman ou les répétitions retirées. [16]

I-3-1-1-Compression RLE (Runlengthencoding) : [17]

L'utilisation de cette méthode a commencé depuis l'arrivée de la théorie de l'information.

La compression RLE est basée sur la répétition des données consécutives. Au lieu de coder les symboles consécutifs de la chaîne, on code le nombre des symboles consécutifs de la même valeur.

Cet algorithme est très simple et peut aboutir à des taux de compression plus élevés. Il existe des variantes dans lesquelles l'image est encodée par pavés de points, selon des lignes, ou bien même en zigzag.

Un exemple de traitement *RLE* est donné à la Figure 1.13 [18].

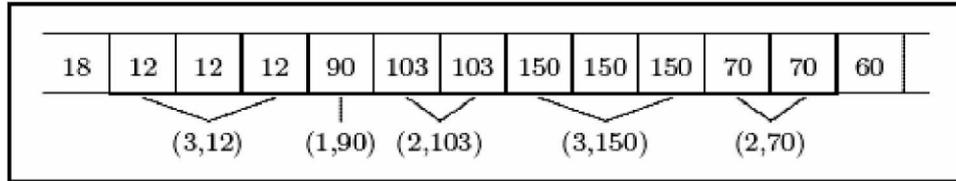


Figure 1.11 : Exemple de codage par plage RLE.

1.3.1.2. Codage de Huffman :

En 1952 David Huffman a proposé le codage de Huffman basé sur le remplacement de chaque symbole de la chaîne à compresser par un code binaire, Le symbole le plus fréquent est codé par moins de bits.

Pour compresser et/ou décompresser les données par l'algorithme de Huffman non a besoins de passer par 2 étapes : Le tableau de fréquence et L'Arbre de Huffman.

Un code de Huffman utilise un arbre binaire qui permet de coder et de comprimer des données. Pour coder les différents caractères existants on les représente par un mot binaire d'une longueur donnée. Le code ASCII utilise un mot binaire de longueur 7 pour représenter un caractère.

Plus grand sera le nombre de caractères codés plus il y aura de bits à stocker, manipuler et transmettre. Pour réduire la quantité de bits produite sans pertes d'information, on utilise la compression des données.

Certains caractères étant plus employés que d'autre, on les code avec des mots binaires plus courts, c'est le principe du codage de Huffman.

Dans un codage de Huffman, on associe les caractères aux feuilles d'un certain arbre binaire.

Le mot binaire qui code un caractère devient donc le numéro binaire de sa feuille. [19]

- Exemple de codage huffman :

Le principe du codage de Huffman repose sur la création d'une structure d'arbre composée de nœuds.

Supposons que la phrase à coder est « *this is an example of à huffman tree* ». On recherche tout d'abord le nombre d'occurrences de chaque caractère. Dans l'exemple précédent, la phrase contient 2 fois le caractère *h* et 7 espaces. Chaque caractère constitue une des feuilles de l'arbre à laquelle on associe un poids égal à son nombre d'occurrences.

L'arbre est créé de la manière suivante, on associe chaque fois les deux nœuds de plus faibles poids, pour donner un nouveau nœud dont le poids équivaut à la somme des poids de ses fils. On réitère ce processus jusqu'à n'en avoir plus qu'un seul nœud : la racine. On associe ensuite par exemple le code 0 à chaque embranchement partant vers la gauche et le code 1 vers la droite.

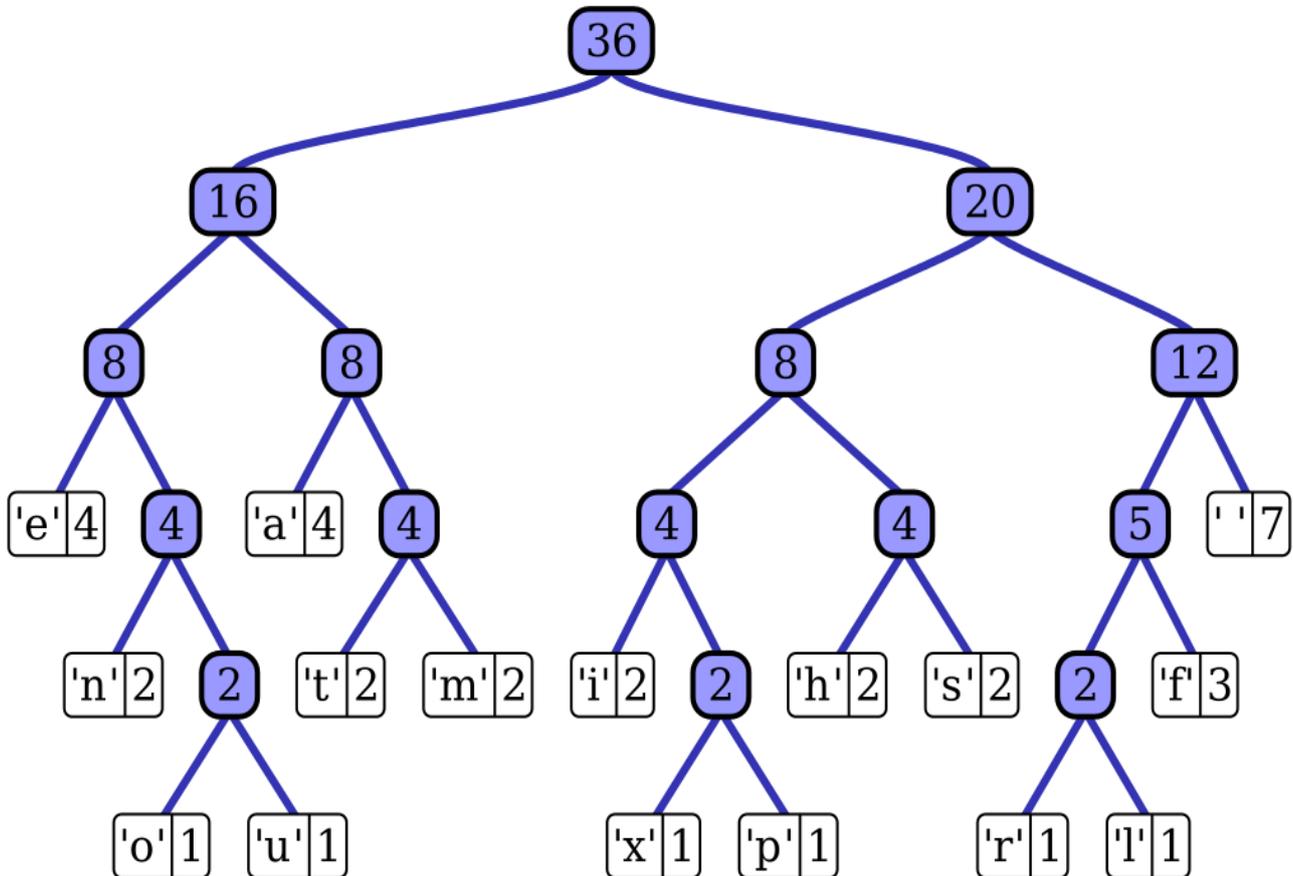


Figure 1.12 : Un exemple d'arbre de Huffman, généré avec la phrase « this is an example of a huffman tree ». [20]

1.3.1.3.Codage Shanon-Fano :

C'est le même principe de Huffman mais dans la 2^{ème} étape, au lieu de construire l'arbre de Huffman, on construit le tableau de Shano-Fano.

Notre contribution n'étant pas sur le codage, nous invitons le lecteur intéressé de consulter les références suivantes : [1,2,3,7]

I-3-2-Compression avec pertes :

La compression avec pertes est une méthode de compression de données qualifiée d'irréversible. Il est donc impossible de retrouver les données originelles après une telle compression. C'est pour cela que l'on emploie les méthodes de compression avec pertes uniquement qu'avec des données "perceptibles" humainement, soit des sons ou des images. En effet, ces types de données peuvent subir des modifications, parfois importantes, sans que l'on ne ressente une quelconque différence par rapport aux données de départ.

Ces techniques de compression avec perte se basent sur le principe que seul un sous ensemble très faible des données que l'on souhaite compresser possède un caractère exploitable pour notre œil, pour notre ouïe... C'est donc cet ensemble là que l'on compressera de manière courte, de sorte que même après avoir réduit la quantité de données, le résultat soit toujours très ressemblant à l'original, voir presque identique, toujours pour une perception humaine.

Avec un algorithme de compression avec perte, la suite de bits obtenue après les opérations de compression et de décompression est donc différente de l'originale, mais l'information restituée est en revanche très voisine. Les algorithmes de compression avec perte sont utiles pour les images, le son et la vidéo. Les fichiers exécutables par exemple, ne peuvent donc pas être compressés selon ce type de méthode car ils ont besoin de conserver leur code intact pour fonctionner. En effet, on ne peut pas espérer faire fonctionner un programme correctement s'il est reconstruit approximativement en supprimant des bits en en ajoutant.

- On peut résumer les 3 principes de la compression avec pertes :

1/ La compression avec pertes est généralement utilisée pour la vidéo et le son, où une certaine quantité de pertes d'informations ne sera pas détectée par des utilisateurs.

2/Une compression est dite avec pertes si les données après décompression sont différentes des données originelles. Elles sont appliquées à des données perceptibles, c'est-à-dire à des images, des sons ou des vidéos.

3/Le principe consiste à supprimer les informations là où les sens de la vue et de l'ouïe ne les perçoivent que très peu. Par exemple, l'œil humain ne distingue que très peu les zones de contraste. Aussi, nous pouvons retirer des détails à ces zones sans trop impacter sur la qualité de l'image.

Ici, nous allons plutôt parler de techniques telles que le sous-échantillonnage ou la quantification. De plus, le nom de format représente directement le type de compression employé : JPEG, MP3, DIVX, MPEG, etc... [21]

I-3-3-Comparaison entre la compression avec et sans pertes :

Le table 1.2 résume la comparaison entre la compression avec et sans perte

Compression avec pertes	Compression sans pertes	
Définition	La compression avec perte est la méthode de codage qui utilise des estimations imprécises pour représenter le contenu.	La compression sans perte est un groupe d'algorithmes de compression de données qui permet de reconstruire avec précision les données d'origine à partir des données compressées.
Utilisé dans	Images, audio et vidéo.	Texte ou programme, images et son.
Application	JPEG, GUI, MP3, MP4, OGG, H-264, MKV, etc.	RAW, BMP, PNG, WAV, FLAC, ALAC etc.
Capacité de stockage de données	Grande	Faible

Table 1.2 : Table de comparaison entre la compression avec et sans perte

I-3-4-Différences clés entre la Compression avec perte et la Compression sans pertes :

La compression avec pertes supprime une partie non utile des données, ce qui est indétectable tandis que la compression sans pertes reconstruit les données exactes.

La compression sans pertes peut réduire la taille des données dans une faible mesure. D'autre part, la compression avec pertes peut réduire la taille du fichier dans une plus grande mesure.

La qualité des données se dégrade en cas de compression avec perte tandis que la compression sans pertes ne dégrade pas la qualité des données.

Dans la compression avec pertes, le canal reçoit plus de données. Inversement, le canal détient une plus petite quantité de données en cas de compression sans pertes.

I-3-5-Conclusion :

La compression avec pertes peut atteindre un niveau élevé de compression de données par rapport à la compression sans pertes. La compression sans pertes ne dégrade pas la qualité des données, au contraire, la compression avec pertes dégrade la qualité des données. La compression avec pertes ne peut pas être implémentée dans tous les types de fichiers car elle supprime une partie des données (redundantes) ce qui n'est pas possible dans le cas d'un texte. [22]

Qualité d'image

Des recherches approfondies ont montré que les images obtenues avec des méthodes basées sur des ondelettes donnent une très bonne qualité visuelle. Au début, il a été montré que même les méthodes de codage simples produisaient de bons résultats lorsqu'elles étaient combinées avec des

ondelettes et constituaient la base de la norme JPEG2000. Cependant, SPIHT appartient à la prochaine génération de codeurs à ondelettes, utilisant un codage plus sophistiqué. En fait, SPIHT exploite les propriétés des images transformées en ondelettes pour augmenter son efficacité.

Les applications SPIHT exploitent les propriétés présentes dans une grande variété d'images. Il a été testé avec succès sur des images naturelles (portraits, paysages, mariages, etc.) et médicales (rayons X, tomodensitométrie, etc.). En outre, son processus de codage intégré s'est avéré efficace dans un large éventail de qualités de reconstruction. Par exemple, il peut coder aussi bien des portraits de bonne qualité que des images médicales de haute qualité (par rapport à d'autres méthodes dans les mêmes conditions).

Le codage SPIHT a également été testé à des fins moins habituelles, comme la compression de cartes d'altitude, de données scientifiques et autres.

I-4-Evaluation de la compression :

L'évaluation de la compression peut être effectuée par plusieurs paramètres. Parmi eux, on peut citer :

$$I-4-1 \text{ Quotient} = \frac{\text{Taille des données originales}}{\text{Taille des données compressées}} \quad (I-1)$$

Plus une compression sera forte, plus le quotient de compression sera lui aussi élevé.

I-4-2. Gain de compression = $(1 - 1/\text{Quotient}) \times 100 \%$, exprimé en pourcentage.

Plus le gain de compression est élevé, plus la taille du fichier compressé résultant est faible.

$$I-4-3. \text{ Taux de bits (Tb) en bits/pixel (bpp)} = \frac{\text{Nombre original de bits/pixel}}{\text{Quotient}} \quad (I-2)$$

Remarque :

Le taux de bits est aussi appelé débit de compression. ^[23]

I-4-5. Qualité de reconstruction des images :

Mesure de la qualité de la restauration La performance d'une méthode de restauration peut être calculée à l'aide d'un des indicateurs suivants : → Erreur quadratique moyenne (mean square error) → Rapport signal-a-bruit → PSNR

Le problème de la qualité se pose pour les méthodes de compression avec pertes

d'informations. En effet, l'image reconstruite n'est pas exactement l'image originale, ainsi il nous faut une mesure de la qualité de l'image reconstruite. Parmi ces critères, nous citerons :

I-4-5-1) EQM (Erreur quadratique moyenne) :

Elle permet de mesurer la dégradation entre une image X origine et l'image X reconstruite après compression de coordonnées (n, m) et de dimension (N,M).

$$EQM = \frac{1}{M \times N} \times \sum_{n=1}^N \sum_{m=1}^M (X_{origine}(n, m) - X_{reconstruite}(n, m))^2$$

Avec :

X origine (n, m) et X reconstruite (n, m) : représentent les valeurs des pixels d'image d'origine et reconstruite.

I-4-5-2. SNR (Signal to Noise Ratio) : Il permet de mesurer le signal original sur bruit d'une image.

$$SNR = 10 \times \log_{10} \left(\frac{\frac{1}{M \times N} \times \sum_{n=1}^N \sum_{m=1}^M (X_{origine}(n, m))^2}{EQM} \right) \text{ dB}$$

I-4-5-3. PSNR (Peak Signal to Noise Ratio) : C'est le rapport signal crête sur L'erreur d'une image.

$$PSNR = 10 \times \log_{10} ((X_{max})^2 / EQM) \text{ dB} \quad (I.3)$$

Avec :

X max : désigne la valeur du pixel maximale possible dans l'image qui prend une valeur 255 pour une image en niveau de gris codée sur 8 bits. Une valeur de PSNR infini correspond à une image non dégradée, et cette valeur décroît en fonction de la dégradation. [24]

I.6.CONCLUSION :

Ce chapitre a été divisé en deux parties et paramètres ; dans la première, nous avons défini l'image avec ses différents types et dans la deuxième partie nous avons présenté les objectifs de la compression et les différentes solutions apportées, comme la compression sans pertes d'informations (réversible) et avec pertes d'informations (irréversible). Enfin nous avons définie les critères d'évaluation de la compression et de la qualité des images reconstruites.

CHAPITRE II

Etude de la norme de compression d'Images fixes JPEG

II-1. Introduction :

Le format JPEG (Joint Photographic Experts Group) est un format d'image compressée qui offre une bonne compression pour une qualité très correcte. Ces deux avantages en font l'un des formats d'image les plus répandus, particulièrement sur le web où les problématiques de transfert et de stockage sont importantes.

Les spécifications de ce format ont commencé entre 1978 et 1980 mais c'est en 1991 que fut déposée la norme JPEG et en 1992 qu'elle fut adoptée. La paternité de ce format est communément attribuée à IBM mais celle-ci fut revendiquée par Forgent en 2006 puis Global Patent Holdings en 2007.

Le format JPEG est représenté par les fichiers .jpeg ou .jpg et offre des taux de compressions allant de 3 à 100 (autrement dit, le fichier est 3 à 100 fois plus petit que l'image d'origine).

Son successeur existe déjà : il s'agit du format JPEG 2000. Ce dernier offre une meilleure compression pour une qualité bien plus importante. Cependant, le format JPEG 2000 est un format propriétaire ce qui freine sa diffusion. Il n'est donc utilisé que dans un cadre professionnel restreint. A titre d'exemple, l'illustration ci-dessous compare une même image au format JPEG et au format JPEG2000. Les deux images ont exactement la même taille. [2]

II-2. Fonctionnement général du JPEG :

JPEG définit deux types de compression :

- Avec pertes ou compression irréversible. C'est le JPEG « classique ». Il permet des taux de compression de 3 à 100 [26,28].
- Sans pertes ou compression réversible. Dans ce cas, le taux de compression n'est que de l'ordre de 2 à 8 [26,28]. Pour cette raison, ce type de compression est très peu utilisé.

Les étapes de la compression JPEG avec pertes sont les suivantes (Voir Figure 2.1) :

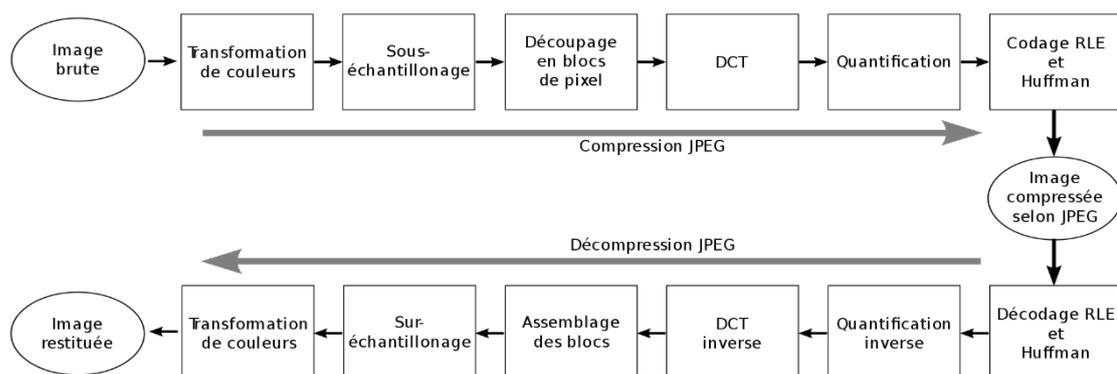


Figure 2.1 : Principe général de la compression/décompression JPEG.

Le format JPEG, commence par traiter séparément chaque composante de couleur. Il est possible d'utiliser directement le modèle RVB, mais JPEG est plus efficace sur le modèle YCbCr (après conversion éventuelle en luminance / chrominance). Comme l'œil humain est moins sensible aux changements de teinte (chrominance) que changements de luminosité (luminance).

L'algorithme JPEG est basé sur cette différence de perception. D'ailleurs, il est même possible de sous – échantillonner ces composantes de chrominance, afin d'aboutir à un modèle prédéfini. Ensuite le format JPEG, commence par découper chaque composante YCbCr en blocs carrés de 64 (8×8) pixels (L'algorithme de compression est calculé pour chaque bloc séparé, ce qui explique pourquoi ces blocs ou groupes de blocs deviennent visibles lorsque la compression est trop appliquée). L'étape suivante dans le processus de compression consiste à appliquer une transformée en cosinus discrète (DCT) à chaque bloc 8×8 pixels, Cette méthode permet de décrire chaque bloc en une carte de fréquences et en amplitudes plutôt qu'en pixels et couleurs, C'est l'étape qui coûte le plus de temps et de ressources dans la compression et la décompression , mais la plus importante car elle nous a permis de séparer les basses fréquences et les hautes fréquences présentes dans l'image. L'étape DCT lui-même est sans pertes. La quantification est l'étape dans laquelle on perd réellement des informations (et donc de la qualité visuelle), mais c'est celle qui fait gagner beaucoup de place. Le but est ici est de garder les basses fréquences, c'est-à-dire celles auxquelles l'œil humain est très sensible et d'atténuer et de vouloir annuler les hautes fréquences, c'est à dire celles auxquelles l'œil humain est très peu sensible. JPEG utilise des tables de quantification, pour la luminance et pour la chrominance. Une fois que ces tables ont été construites, les constantes des tables sont utilisées pour quantifier les coefficients DCT. Chaque coefficient DCT est divisé par sa constante correspondante dans la table de quantification et arrondie au plus proche entier. Le résultat de la quantification des coefficients DCT sont parcourus en zig zag avant d'appliquer le codeur entropique qui est ici, Le codeur de Huffman qui compresse sans pertes. Pour pouvoir reconstruire l'image originale à partir des données compressées, il faut appliquer l'étape de décompression donc de décodage, cette étape s'obtient en appliquant les procédés inverses

de l'étape du codage

Dans la suite ,nous verrons les différentes étapes de la compression/décompression JPEG en détails. [27]

3. Etapes principales :

- DCT (Discrete Cosine Transformation).
- Quantification.
- Lecture Zigzag.
- Codage direct de la composante DC (~ moyenne du bloc 8×8).

- RLE sur les composants AC (Run Length Encoding: on associe les n-uplets de coefficients nuls avec le premier coefficient non nul suivant).
- Codage entropique de type VLC (Variable Length Coding).

II-3. Transformation des couleurs :

La première étape consiste à passer du modèle initial des couleurs de l'image (souvent RVB) en modèle de type chrominance/luminance (YCrCb).



Figure 2.2.1 : RGB à YCrCb

JPEG est capable de coder les couleurs sous n'importe quel format, toutefois les meilleurs taux de compression sont obtenus avec des codages de couleur de type luminance/chrominance car l'œil humain est assez sensible à la luminance (la luminosité) mais peu à la chrominance (la teinte) d'une image. Afin de pouvoir exploiter cette propriété, l'algorithme convertit l'image d'origine depuis son modèle colorimétrique initial (en général RVB) vers le modèle de type chrominance/luminance YCbCr. Dans ce modèle, Y est l'information de luminance, et Cb et Cr sont deux informations de chrominance, respectivement le bleu moins Y et le rouge moins Y. [28]

La formule de passage de RGB ou RVB (rouge vert bleu) vers YCbCr est la suivante :

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.2220 & 0.7067 & 0.0713 \\ -0.1195 & -0.3810 & 0.5 \\ 0.5 & -0.4542 & -0.0458 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Figure 2.2.2 : formule de passage de RGB à YCbCr.

La conversion inverse fait ainsi :

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.5560 \\ 1 & -0.1866 & -0.48886 \\ 1 & 1.8580 & 0.0001 \end{pmatrix} \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix}$$

Figure 2.2.3: formule de passage de YCbCr à RGB.

II-4. Sous-échantillonnage de la chrominance :

Pour exploiter cette faible sensibilité de l'œil humain à la chrominance, on peut procéder à un sous-échantillonnage des signaux. Le principe de l'opération est de réduire la taille de plusieurs blocs de chrominance en un seul bloc. Un traitement identique est appliqué aux blocs Cb et aux blocs Cr, tandis que les blocs de luminance (blocs Y) ne sont pas modifiés. Le sous-échantillonnage peut être réalisé selon plusieurs modes différents. Le type de mode utilisé dans une opération de compression est spécifié par la notation « J:a:b ».

L'interprétation de la notation est la suivante :

J : représente la largeur de la plus petite matrice de pixels considérée (généralement 4)

a : le nombre de composantes de chrominance dans la première ligne

b : le nombre de composantes de chrominance supplémentaires dans la deuxième ligne

On distingue 3 types majeurs de sous-échantillonnages.

Le **4:4:4** est un sous-échantillonnage qui consiste à ne rien faire. Aucune compression n'est effectuée et aucune perte de qualité ne peut être enregistrée.

Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							

Figure 2.3.1 : Le sous-échantillonnage 4:4:4

Le **4:2:2** est un sous-échantillonnage qui consiste à supprimer les informations de la chrominance sur une colonne sur deux. La colonne restante possède alors la moyenne des chrominances des deux colonnes. Lors de la décompression, il suffira de recopier cette information sur la colonne vide.

Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y

Figure 2.3.2 : Le sous-échantillonnage 4:2:2

Le **4:2:0** est un sous-échantillonnage qui effectue le même processus que le 4:2:2 mais en effectuant aussi cette opération sur les lignes. On divise donc ici par 4 l'information de la chrominance.

Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y	Y	Y	Y	Y	Y	Y	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y	Y	Y	Y	Y	Y	Y	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y	Y	Y	Y	Y	Y	Y	Y
Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y	Y Cb Cr	Y
Y	Y	Y	Y	Y	Y	Y	Y

Figure 2.3.3 : Le sous-échantillonnage 4:2:0

Ce type de sous-échantillonnage permet de réduire de manière importante la taille des images. Cependant, il conduit à une perte d'informations qui ne peut être récupérée dans la phase de décodage de l'image ; il s'agit donc d'une opération de compression irréversible. À l'issue de cette phase, les données se présentent sous la forme de blocs de coefficients Y, Cb et Cr, le nombre de

blocs Cb et Cr étant égal au quart, à la moitié ou à 100 % du nombre de blocs Y selon le mode sous-échantillonnage appliqué [29]

Lors de la décompression de l'image JPEG, il suffit de sur-échantillonner les plans Cb et Cr retrouvés après le décodage pour retrouver l'image d'origine.

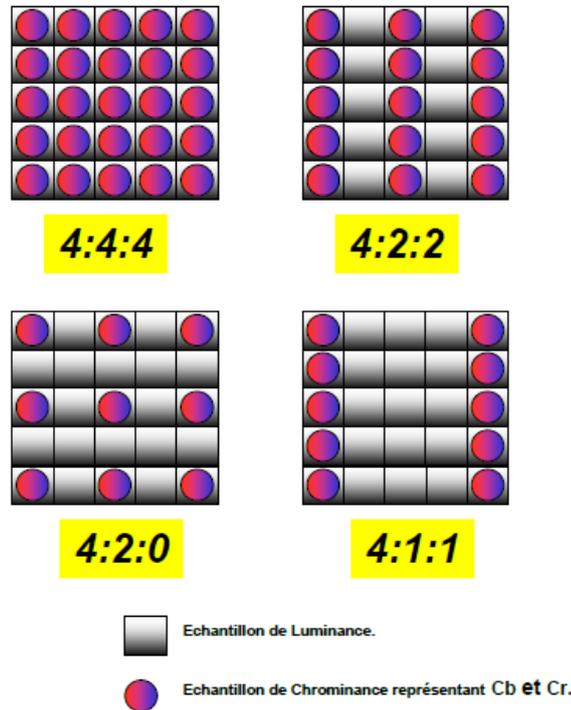


Figure 2.4: types de formats de sous – échantillonnage dans JPEG.

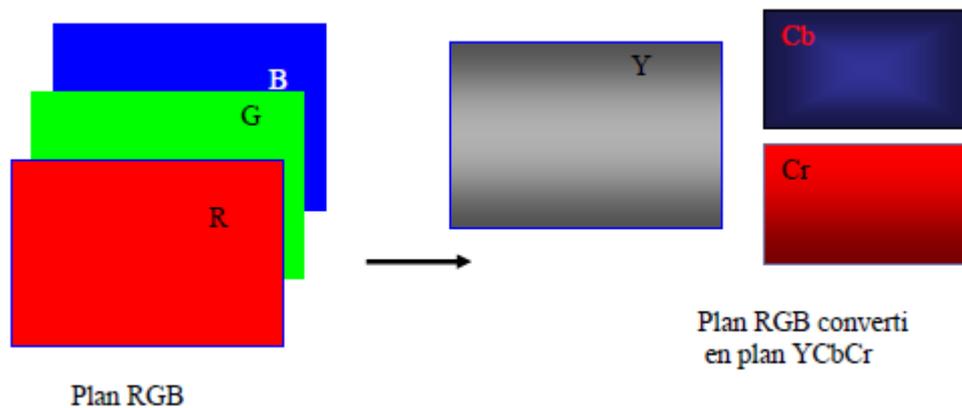


Figure 2.5 : Exemple de sous-échantillonnage de type 4:2:0.

II-5. Découpage en Blocs :

Pour faciliter les calculs, une division pour chaque bloc (Y, Cb, Cr) doit être appliquée pour chaque sous-bloc Y, Cr, Cb, chaque bloc contient 8*8 case. Les opérations qui suivent sont appliquées pour chaque bloc indépendamment des autres. [30]

L'image est découpée en blocs de 64 (8×8) sous-pixels. Les informations de luminance n'étant pas sous-échantillonnées, chaque bloc correspond à 8×8 pixels dans l'image d'origine. Les informations de chrominance, par contre, correspondent en fonction du sous-échantillonnage à 8×8, 16×8 ou 16×16 pixels dans l'image d'origine.

La compression est effectuée sur des blocs de pixels seulement. Ces blocs sont des matrices de pixels 8*8 soient 64 pixels.

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

II-6. Transformation DCT :

Pour chaque bloc, on applique une transformation de cosinus discrète, chaque valeur du résultat obtenue reflète la rapidité, l'écart et l'importance d'un changement d'un pixel à un autre.

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix} \rightarrow F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

Pour un bloc de taille N*N, la DCT s'exprime mathématiquement [31] :
La transformée DCT s'exprime mathématiquement par [20], [21]

$$F(u,v) = \left(\frac{1}{4}\right) \cdot C(u) \cdot C(v) \sum_{i=0}^7 \sum_{j=0}^7 f(i,j) \cdot \cos\left((2i+1) \cdot u \cdot \frac{\pi}{16}\right) \cdot \cos\left((2j+1) \cdot v \cdot \frac{\pi}{16}\right)$$

Figure 2.6 : Equation de DCT.

Et la transformée DCT inverse s'exprime par :

$$f(i, j) = \left(\frac{1}{4}\right) \sum_{u=0}^7 \sum_{v=0}^7 C(u) \cdot C(v) \cdot F(u, v) \cdot \cos\left((2i + 1) \cdot u \cdot \frac{\pi}{16}\right) \cdot \cos\left((2j + 1) \cdot v \cdot \frac{\pi}{16}\right)$$

Figure 2.7: DCT inverse.

Avec:

$$C(u) = c(v) = 1/\sqrt{2} \quad \text{pour } u=v=0.$$

$$C(u) = c(v) = 1 \quad \text{autres } u \neq 0, v \neq 0.$$

F (u,v) : représente la valeur de la DCT au point de coordonnées (u, v) dans le bloc résultat de 8 × 8 pixels.

f (i,j) : représente la valeur du pixel de coordonnées (i, j) dans le bloc de l'image originale de 8× 8 pixels.

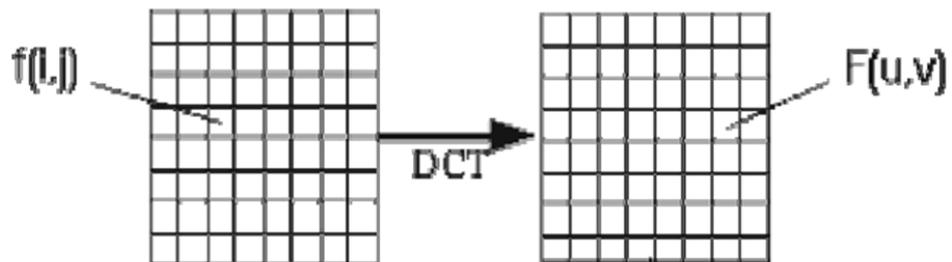


Figure 2.8: DCT inverse.

Les étapes principales de tout bloc basé sur l'algorithme DCT sont :

- Diviser l'image en blocs de 8x8 pixels
- Appliquer DCT sur chacun de ces blocs.

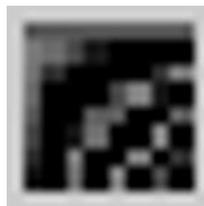


Figure 2.9 : bloc DCT.

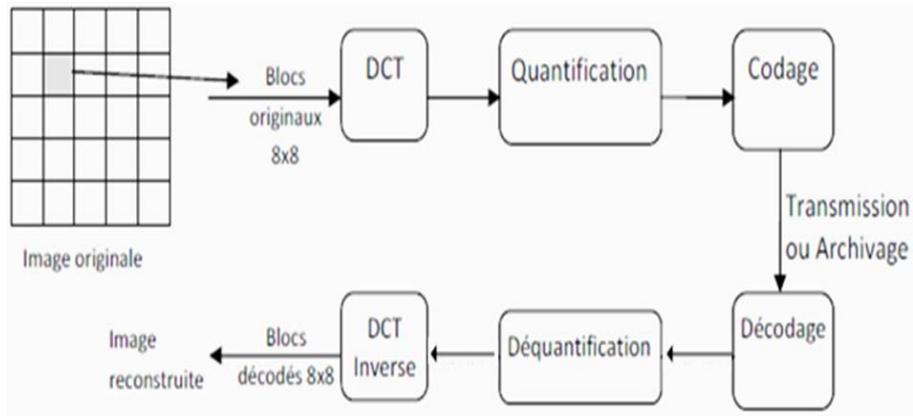


Figure 2.10: Schéma fonctionnel de l'algorithme JPEG

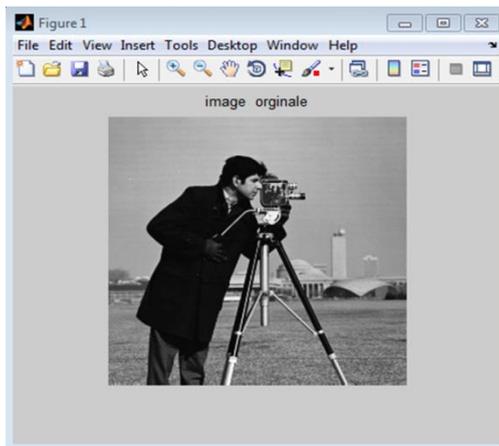


Figure 2.12 : Image Originale

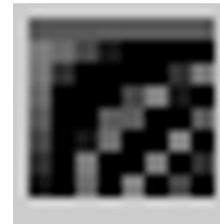


Figure 2.11 : Bloc de 8 x 8 pixels uniforme.

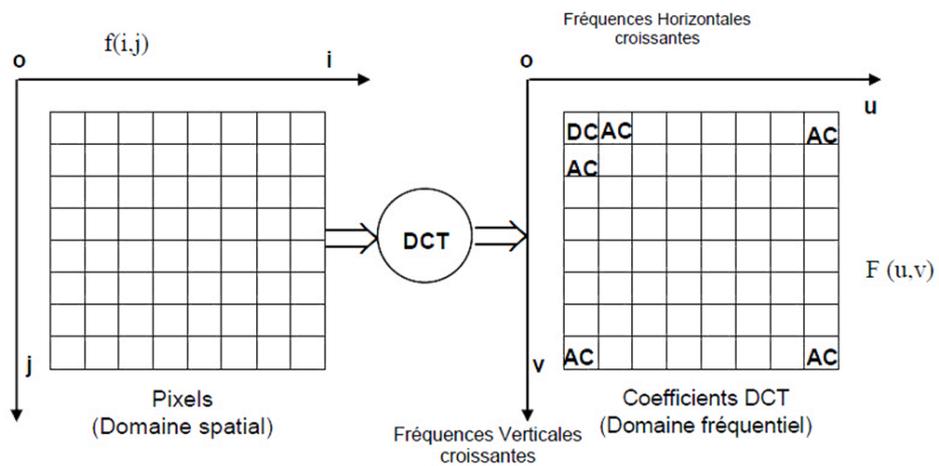


Figure 2.13 : Représentation de la DCT d'un bloc de 8 x 8 pixels.

III.6.1. Avantages de la DCT :

Les spécialistes du codage s'accordent à la DCT les avantages suivants :

- la DCT permet d'atteindre des performances meilleures à celles des techniques différentielles (10 à 30 MBPP en télévision) et de très bas débits pour des images de communications de qualité réduite (de 2 MBPS 64 KBPS selon la qualité recherchée).
- Elle offre actuellement le meilleur compromis puissance complexité.
- La complexité des équipements augmente brutalement lorsque l'on cherche à descendre en dessous d'un seuil situé vers 50 MBPS.
- Elle possède un ensemble de fonctions de base stable (les fonctions cosinus).
- Elle répond aux exigences des télécommunications et de l'audiovisuel grand public et permet sinon une totale compatibilité, du moins un minimum de synergie tant pour les études que pour le développement des composants et le volume de leur marché qui fixera le coût des circuits VLSI

Le codeur DCT permet d'inclure facilement la réponse en fréquence de l'œil humain dans le pas de quantification.^[32]

II-7. Quantification :

La quantification de chaque blocannule ou diminue les hautes fréquences, afin d'atténuer les détails en parcourant le bloc intelligemment avec un codage RLE (en zig-zag pour enlever un maximum de valeurs nulles).

Il est à noter que les coefficients de la DCT les plus « énergétiques » (c'est-à-dire de plus haute énergie et donc ceux qui sont porteurs d'informations) se trouvent autour de la fréquence zéro (composante continue) et donc il s'agit d'éliminer les coefficients en dessous d'un certain seuil (forcément éloignés de l'origine) car ils sont peu porteurs d'information.

La technique est simple. Il suffit de diviser notre matrice des fréquences avec la matrice de quantification pour obtenir notre matrice quantifiée.

Cette matrice a un avantage qui sera utilisée dans la prochaine étape : les valeurs non nulles ne sont toutes situées dans le bord haut-gauche de la matrice.

$$F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix} \quad \longrightarrow \quad F^* = \begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Cette quantification est donc réalisée par une matrice de quantification Q de la taille du bloc de 8×8 pixels. Elle s'applique sur un bloc de la DCT de telle sorte que : [33], [34]

$$F^*(u,v) = \text{entier le plus proche de } \frac{F(u,v)}{Q(u,v)}$$

Avec :

$F^*(u,v)$ est la matrice quantifiée.

$F(u,v)$ est la matrice des coefficients à quantifier.

$Q(u,v)$ est la table de quantification.

Et pour la quantification inverse :

$$F(u,v) = F^*(u,v) \times Q(u,v) \quad (\text{II-1})$$

Avec

$F(u,v)$ représente la matrice de quantifiée

Dans les traitements d'images actuels, l'attable de quantification est définie par : [33]

$$Q(i,j) = 1 + (1 + i + j)Fq \quad (\text{II-2})$$

Fq : Représente le facteur de qualité de la matrice de quantification.

II-7-1. Linéarisation des blocs :

Lecture Zigzag

- Pourquoi ? Pour rassembler les faibles niveaux DCT.
- Transforme un bloc 8×8 en un bloc 1×64 .

Après la quantification et avant le codage entropique, la lecture des blocs se fait d'une manière en Zig-Zag (balayage depuis les basses fréquences jusqu' aux hautes fréquences) afin d'obtenir un maximum de zéros à la fin de la séquence (figure (2.13)). [35, 33]

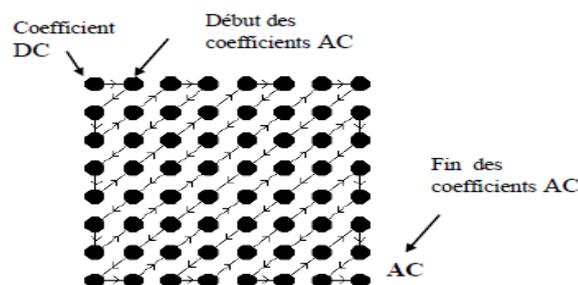


Figure 2.14 : Lecture en Séquence Zig-Zag.

3. Le nœud créé remplace désormais les deux symboles dans la suite du processus. A ces derniers sont affectés respectivement les chiffres binaires 0 (pour le plus fréquent) et 1 (pour le plus rare).
4. La même démarche est reprise en considérant les deux symboles ou nœuds de poids le plus faible. Elle est renouvelée tant qu'il reste plus d'un nœud de libre.

II.8.2. Codage Arithmétique :

Le codage arithmétique ne suit pas l'idée de remplacer un symbole par un code spécifique. Il remplace un flot de symboles par un seul nombre en virgule flottante. Ce dernier est compris entre zéro et un. Il est décodé de façon unique pour recréer le flot exact de symboles qui conduit à sa génération [31, 38, 39].

Avec le schéma de codage très simplifié suivant, on remarque que le codage nous délivre deux tables (quatre pour une image couleur). Ces tables étant enregistrées dans le fichier final peuvent être choisies par le compresseur.

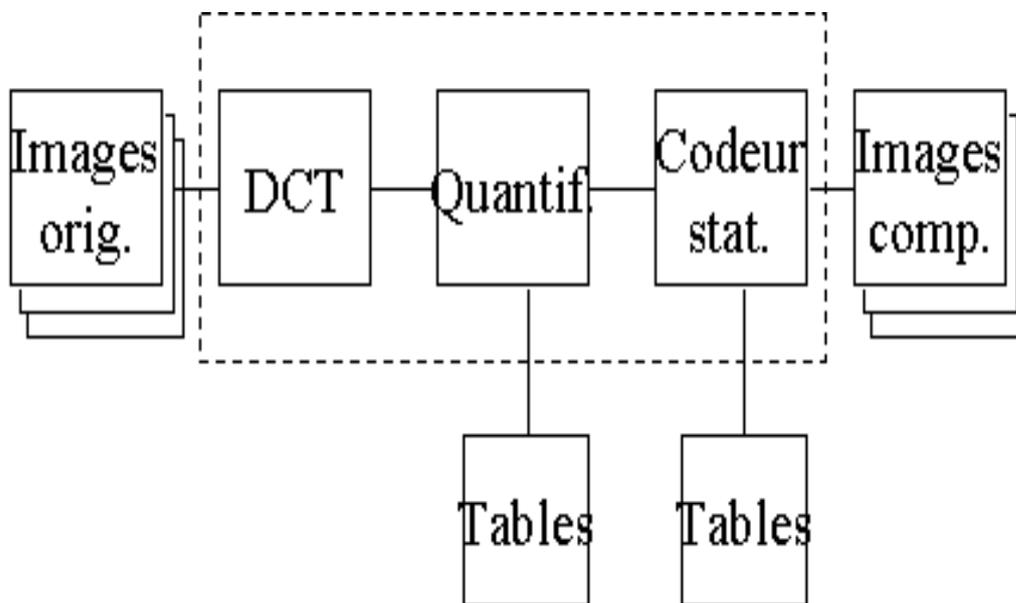


Figure 2.16: Schéma de codage.

Tableau d'avantages et inconvénients :

Des techniques	Avantages	Inconvénients
Encodage de longueur (RLE)	Cet algorithme est facile à mettre en œuvre et ne nécessite pas beaucoup de puissance processeur [20].	La compression RLE n'est efficace qu'avec des fichiers contenant beaucoup de données répétitives [20].
Codage LZW	Compression simple, rapide et bonne [21]. Table de mots de code dynamique construite pour chaque fichier [21]. La décompression recrée la table de mots de code afin qu'il ne soit pas nécessaire de la passer [23].	La compression réelle est difficile à prédire Jin dal [21]. Il occupe plus d'espace de stockage que le taux de compression optimal [21]. L'algorithme LZW ne fonctionne que lorsque les données d'entrée sont suffisamment grandes et que la redondance des données est suffisante [20].
Codage de Huffman	Cet algorithme de compression très simple et efficace compressant des fichiers texte ou programme [21]. Cette technique montre des séquences plus courtes pour des caractères plus fréquents [20]. Sans préfixe : aucun codage de séquence de bits d'un caractère n'est le préfixe de tout autre codage de séquence de bits [20].	Une image compressée par cette technique est mieux compressée par d'autres algorithmes de compression [21]. L'arbre de code doit également être transmis ainsi que le message (sauf si une table de code ou une table de prédiction est convenue entre l'expéditeur et le destinataire) [20]. Données entières corrompues par un bit corrompu [20]. La performance dépend d'une bonne estimation si l'estimation n'est pas meilleure que la performance est mauvaise [20].
Codage Arithmétique	Sa capacité à garder le codage et le modeler séparés [22]. Aucune arborescence de code ne doit être transmise au récepteur [22]. Son utilisation les valeurs fractionnaires [22].	le codage arithmétique a des opérations complexes car il consiste en additions soustractions multiplications et divisions [22]. codage arithmétique nettement plus lent que le codage de Huffman, il n'existe aucune précision infinie [22]. deux structures de problèmes pour stocker les nombres et la division constante de l'intervalle peuvent entraîner un chevauchement de code [22].

Tableau 2.3 : Tableau comparatif des techniques de codage

II.9. Décompression JPEG

Les étapes de la décompression s'effectuent dans l'ordre inverse de la compression suivant les méthodes définies précédemment (en même temps que la compression).

II.10. Avantages et inconvénients du JPEG :

Le format JPEG comme tous les formats photo cumule défauts et avantages. Voyons cela en détails :

Avantages du JPEG :

Les avantages du format JPEG sont la simplicité, l'universalité et la légèreté.

- Simplicité car, le format Jpeg est en quelque sorte « développé » par l'appareil, et ne nécessite pas d'être travaillé systématiquement en post-traitement sur ordinateur comme le fichier Raw.
- Universalité : le format Jpeg peut être lu par tous les programmes, ordinateurs, tablettes, etc. Il est reconnu par tous les labos, sites web,...etc.
- Légèreté : le format Jpeg est un format compressé et à ce titre il prend évidemment très peu de place sur la carte mémoire ou le disque dur. Mais cet argument est moins valable aujourd'hui, du fait du prix très bas des cartes et du disque dur.

Inconvénients :

Le format JPEG possède les inconvénients suivants :

- La compression des données : la légèreté due à la compression amène une perte de qualité
 - Développement définitif : les images au format Jpeg sont en quelque sorte « développées » par l'appareil et les réglages tels que la balance couleur ou les styles d'images appliqués définitivement^[34]

Le fait que la compression soit appliquée sur des sous-matrices produit un effet de pixellisation pour un taux de compression trop élevé. Ce phénomène introduit des effets de bords entre les blocs (artéfacts)

II.10.CONCLUSION :

Dans ce chapitre, nous avons parlé de la compression des images fixes selon la norme JPEG, Qui s'effectue en plusieurs étapes qu'on peut résumer ici en deux étapes essentielles :

1) La première étape (DCT + Quantification) : C'est une compression qui va engendrer des pertes. En effectuant une transformation DCT sur une image, on va modifier sa représentation.

C'est en tirant partie des propriétés de cette représentation (qui va concentrer l'information utile dans les basses fréquences) et en tolérant certaines erreurs dans nos calculs, que la première compression va se faire.

2) La deuxième étape Codage de Huffman : Qui s'effectue sans pertes. Un codage qui se base sur des caractéristiques statistiques ; il s'agit du codage de Huffman. Dans cette partie, les informations codées seront reconstituées lors des décodages tels qu'elles sont, c'est-à-dire sans aucune perte.

CHAPITRE III

**Etude de la norme de compression
d'Images fixes**

JPEG2000

III.1- Introduction :

JPEG 2000 ou ISO/CEI 15444-1, abrégé JP2 (quelquefois J2K), est une norme de compression d'images commune à l'ISO, la CEI et l'UIT-T, développée entre 1997 et 2000, et créée par le groupe de travail *Joint Photographic Experts Group*. Depuis mai 2015, il est officiellement reconnu par l'ISO / CEI et l'UIT-T sous le code ISO/IEC CD 15444.

JPEG 2000 peut travailler avec ou sans perte, en utilisant des transformées en ondelettes (méthode d'analyse mathématique du signal), dont la décomposition est similaire à la transformée de Fourier à court terme. Ses performances en compression (avec et sans perte) sont supérieures à celle de JPEG ISO/CEI 10918-1 (JPEG Baseline). On obtient donc des fichiers d'un poids inférieur à qualité d'image égale. De plus, les contours nets et contrastés sont mieux rendus

III-2- Généralités :

La norme JPEG 2000 comprend 16 parties (en 2019), c'est cependant à la première que l'on fait référence comme « cœur » du système. JPEG normalise uniquement l'algorithme et le format de décodage. La méthode d'encodage est laissée libre, du moment que l'image produite est décodable par un décodeur standard. La norme propose à cette fin un jeu de fichiers de test, appelés fichiers de "conformance" ou conformité. Un décodeur est dit conforme s'il est capable de décoder tous les fichiers dits de "conformance".^[38]

Outre ses performances en compression, JPEG 2000 apporte une multitude de nouvelles caractéristiques telles la scalabilité, les régions d'intérêt, la résistance aux erreurs de transmission, le codage sans pertes, la polyvalence de l'organisation des données, ainsi que les diverses extensions visant une application (interactivité, sécurité, sans fil, etc.) qui font l'intérêt de cette norme.

Par ses fonctionnalités avancées, sa capacité à gérer les images de grande taille, ainsi que d'excellentes performances à haut débit, JPEG 2000 s'adresse aux professionnels de l'image, mais n'a pour l'instant que peu d'applications grand public. En particulier, sa présence sur le World Wide Web est marginale.

Un codec JPEG2000 en source ouverte, Open JPEG, a été développé par l'Université de Louvain, afin de promouvoir l'utilisation de JPEG 2000.^[39]

III-3- Principe général :

Le processus de codage suit un schéma classique de modification des propriétés statistiques des données source par un changement de modèle (synthèse) colorimétrique par une transformée, avant quantification des coefficients issus de cette transformée puis codage entropique. Les nouveautés par rapport à JPEG du point de vue compression sont l'utilisation d'une transformée

en ondelettes, qui offre une scalabilité naturelle, mais surtout d'un algorithme de codage entropique très sophistiqué. Celui-ci est fortement basé sur l'algorithme EBCOT de David Taubman. Il consiste en un regroupement et une modélisation des coefficients ondelettes qui fournissent à un codeur arithmétique adaptatif un train binaire possédant les propriétés statistiques adéquates.

Il s'ensuit une étape d'allocation de débit qui permet de respecter le débit cible, et dont le travail est facilité par le partitionnement du train binaire formé par EBCOT. La dernière étape est la mise en forme syntaxique du codestream JPEG2000, avec la formation des paquets, puis la syntaxe haut niveau, particulièrement abondante dans JPEG2000.

Dans la norme JPEG2000, un codestream est l'ensemble des données formées par les données images compressées regroupées dans des paquets ainsi que la syntaxe de haut niveau : en-têtes de pavés, en-tête principal. Les métadonnées du format de fichier JP2 ne font pas partie du codestream. JP2 encapsule le codestream JPEG2000 dans un format de fichier.^[40]

III-4- Fonctionnement général du JPEG2000 :

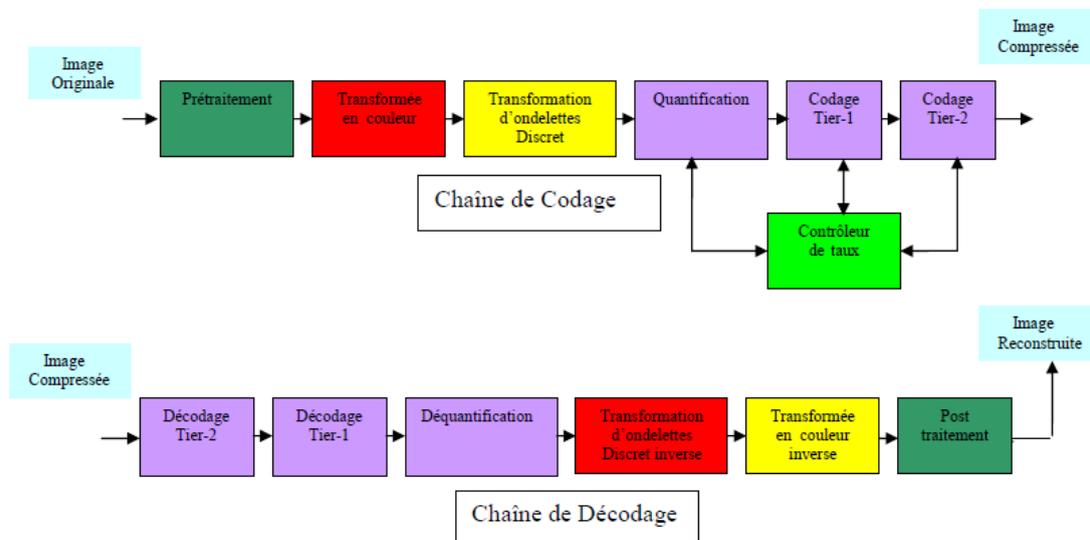


Figure 3.1 : Fonctionnement général du JPEG2000

La chaîne de codage de JPEG-2000, commence par un prétraitement de l'image originale (figure (III-1)). L'image d'origine est segmentée en petites images appelées tuiles. Une tuile inclut tous les plans de couleur (RGB). Chaque plan d'une tuile est codé et décodé de façon indépendante. Cette méthode permet de faire du traitement parallèle matériel ou logiciel. Chaque plan de tuile est décorrélé par une transformée en ondelettes. Les coefficients d'ondelettes sont regroupés par niveau de décomposition. Chaque niveau de décomposition contient quatre sous-bandes. Cette représentation correspond au diagramme de Mallat^[41]. Deux types de filtres sont proposés dans la première partie de la norme : les filtres (5,3) pour la transformation en ondelettes dite réversible et les filtres (9,7) pour la

transformation dite irréversible qui fournit les meilleurs résultats en termes de décorrélation de l'image pour les faibles valeurs de taux débits [42].

Une sous-bande regroupe les coefficients correspondant à une bande de fréquences spatiales horizontales et verticales d'un plan d'image. Pour l'instant, il n'y a pas encore de compression mais seulement une concentration de l'information sur un nombre réduit de coefficients.

Après quantification, l'information est encore plus concentrée sur un nombre réduit de coefficients. Les sous-bandes de « hautes fréquences » peuvent être quantifiées plus 36 fortement sans pertes visibles pour l'œil [43]. Chaque sous-bande d'un plan est divisée en blocs de coefficients. Ce sont des tableaux de coefficients rectangulaires qui pourront être extraits de façon indépendante afin de subir un codage entropique.

Le codage d'un bloc consiste à parcourir les coefficients plans de bits par plan de bits en allant du plan de bits de poids le plus élevé (MSB) vers le moins élevé (LSB). Le codage commence seulement lorsqu'un plan de bits devient significatif. La position du plan de bits significatif est alors stockée dans un en-tête du fichier et le codage des plans de bits commence. Chaque plan de bits est codé en trois passes successives excepté le premier plan significatif. Chacune de ces trois passes récupère des informations contextuelles à propos des données du plan de bits, ces contextes qui serviront au codeur entropique de type arithmétique. Un codeur arithmétique utilise cette information contextuelle et son état interne pour coder les informations résultant du balayage de chaque plan de bits. Des paquets de données compressées sont générés. Un processus de terminaisons est mis en œuvre après chaque bloc de coefficients codés ou après chaque passe de plan de bits. Ce dernier mode de terminaison permettra de mieux résister aux erreurs de transmission. Dans la suite, nous verrons les différentes étapes du codage et décodage JPEG2000 en détails.

III-4- Modèle d'image utilisé par JPEG2000 :

Avant de présenter en détails le mécanisme de codage utilisé dans JPEG2000, il est nécessaire de bien comprendre le modèle d'image utilisé. Une image est composée de plusieurs composantes (limitées à 214). Chaque composante est constituée d'une matrice rectangulaire d'échantillons. Les valeurs des échantillons de chaque composante sont des valeurs entières (avec ou sans signe) de précision pouvant aller de 1 à 38 bits/échantillons.

Le signe et la précision des échantillons sont spécifiés individuellement par composante. Ces composantes peuvent être utilisées pour représenter les informations spectrales de l'image originale. Par exemple, une image en couleur RGB est composée de 3 composantes : une pour représenter la couleur rouge, une autre pour la couleur verte, et une autre pour la couleur bleue. Dans le cas simple d'une image à nuance de gris, on ne retrouve qu'une seule composante correspondant à l'information

de luminance de l'image. Les composantes entre elles peuvent avoir des tailles différentes. Par exemple lorsque l'on représente une image constituée des informations de luminance et de chrominance, les échantillons de luminance sont souvent plus finement échantillonnés que les échantillons de chrominance^[44]. De plus, JPEG2000 gère le fait que les instants d'échantillonnage puissent être non coïncidents les uns avec les autres, comme c'est le cas pour des images vidéo de type 4 :2 :0 par exemple.

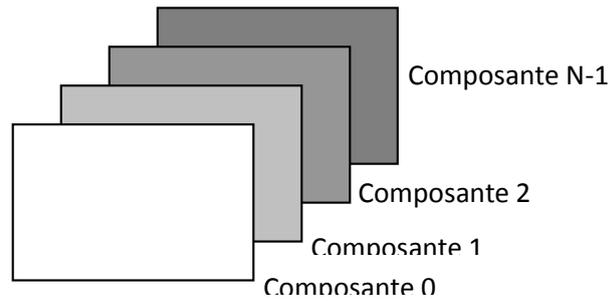


Figure 3.2 : Représentation d'une image composée de N composantes numérotées de 0 à N-1.

L'image est repérée selon un système de coordonnées.

4.1. Prétraitement ou Traitements préliminaires :

Deux opérations sont réalisées au cours de cette phase ^[45].

La première est le partitionnement de l'image en rectangles de même taille que l'on appelle tuiles (figure (3.2)) et qui ne se recouvrent pas. Les tuiles seront ensuite codées 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] à [128, 127],

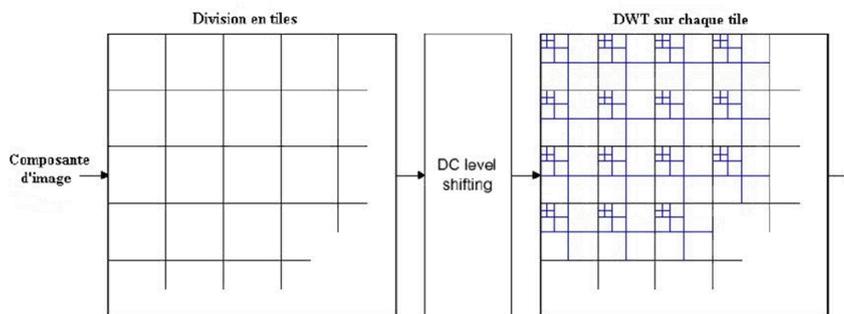


Figure 3.3: Division en tuiles , décalage en DC et DWT sur chaque composante.^[45]

III-5. Transformée en couleur :

La transformée couleur est optionnelle. Elle consiste à passer du modèle colorimétrique de l'image d'origine (RVB en général) au modèle YUV (1 luminance, 2 chrominances) plus adapté pour la compression car les 3 composantes sont beaucoup moins corrélées.

Deux transformées sont spécifiées : une irréversible (ICT pour Irréversible Component Transform), à coefficients réels, et une réversible (RCT pour Réversible Component Transform) à coefficients entiers. L'intérêt de la RCT est d'être utilisée en combinaison avec la transformée en ondelettes réversible 5/3 pour la compression sans pertes.

Bien que la RCT puisse être utilisée pour du codage avec pertes, c'est généralement l'ICT qui est utilisée car elle donne de meilleurs résultats.

Le tableau (III-1) donne un exemple des transformées directe et inverse de l'espace de couleur RV B dans l'espace de couleur(YUV) dans le cas de la transformée réversible et irréversible

MODE	ICT	RCT
Codage	$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$	$\begin{pmatrix} Y_r \\ U_r \\ V_r \end{pmatrix} = \begin{pmatrix} \frac{R+2G+B}{4} \\ R-G \\ B-G \end{pmatrix}$
Décodage	$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & -0.71414 \\ 1.0 & 1.772 & 0 \end{pmatrix} \cdot \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$	$\begin{pmatrix} G \\ R \\ B \end{pmatrix} = \begin{pmatrix} Y_r - \left[\frac{U_r + V_r}{4} \right] \\ U_r + G \\ V_r + G \end{pmatrix}$

Tableau 3.1 : Transformation directe et inverse des plans de couleur pour Le Mode ICT (irréversible) et pour le mode RCT (réversible) en (YUV).

Le codage des trois composantes obtenues se fera alors indépendamment comme le montre la figure suivant

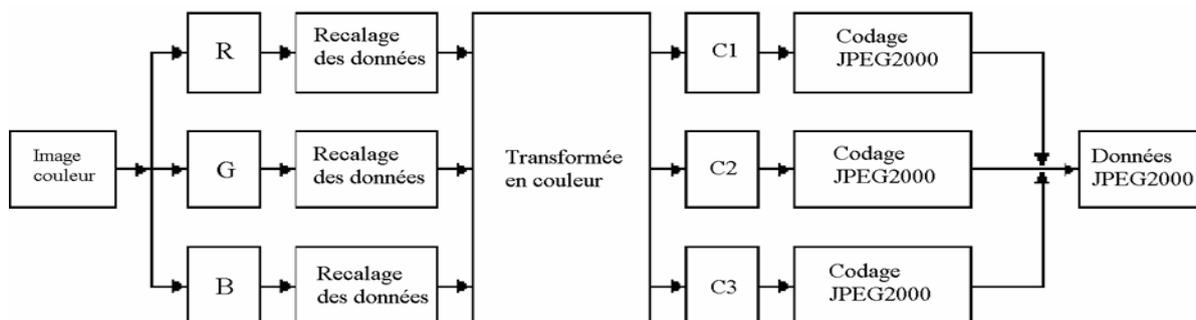


Figure 3.4 : Codage des Composantes de Couleur.

III-6. Transformée en ondelettes :

Tout d'abord, on procède à une transformation de l'image en ondelettes selon un schéma à plusieurs niveaux, processus relativement complexe qui est détaillé ci-dessous.

- Ensuite, on réalise une quantification des informations. Lors de cette phase, les détails qui se situent au-dessous d'un certain seuil sont purement et simplement éliminés. C'est donc à ce niveau que se produit la perte d'informations.
- Enfin, on termine en codant les informations.

La décompression des images s'opère par le schéma inverse : les informations sont tout d'abord décodées pour fournir un ensemble à plusieurs niveaux d'ondelettes qui permet la reconstitution progressive de l'image.

La transformée en ondelettes est utilisée pour l'analyse des composantes imagerie pour différents niveaux de décomposition [46,13,30,31]. Ces niveaux de décomposition contiennent un nombre de sous-bandes, composées chacune de coefficients qui décrivent les caractéristiques horizontales et verticales de l'imagerie originale. Dans la première partie du standard JPEG2000 sont permises seulement des décompositions de puissance 2 sous la forme d'une décomposition dyadique. On passe d'un niveau N-1 à N par un filtrage passe-bas et passe-haut sous échantillonnés d'un facteur 2 sur les lignes puis sur les colonnes. Chaque sous-bande représente l'activité du signal dans les différentes bandes de fréquence à différentes résolutions spatiales. Sur un niveau, on obtient une décomposition de l'image en sa composante principale (bande LL) et trois composantes de détails qui représentent les détails horizontaux, verticaux et diagonaux de l'image. Si le nombre de niveaux de résolutions spatiales est NL, on a alors $3*NL+1$ sous-bandes différentes.

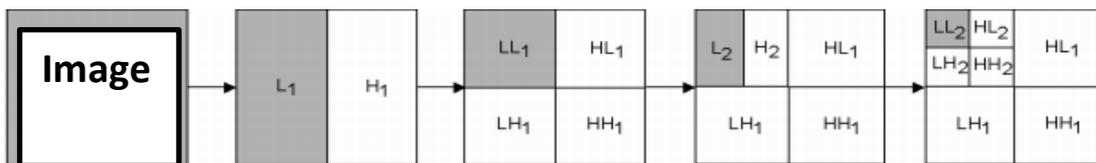


Figure 3.5 : Transformée en ondelettes.

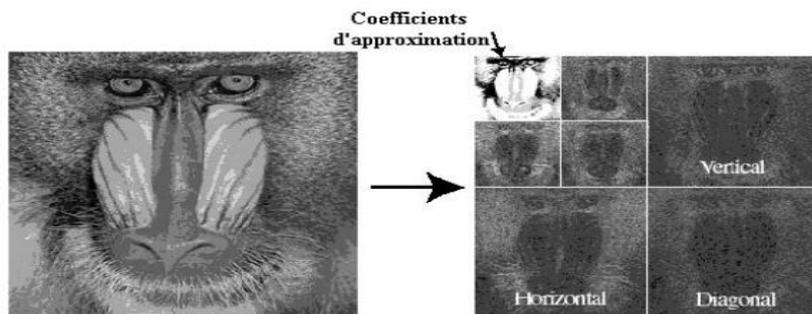


Figure 3.6: Décomposition d'une image en ondelettes en deux niveaux.

Pour réaliser la transformée en ondelettes discrète, le standard utilise une décomposition unidimensionnelle (1-D) de la sous-bande pour les deux directions horizontales et verticales, avec des filtres passe-bas et passe-haut. Les coefficients donnés par le filtre passe-haut représentent une version résiduelle des coefficients originaux, nécessaires pour la reconstruction parfaite de l'image originale à partir des coefficients passe-bas. La transformée en ondelettes discrètes peut être irréversible ou réversible. La transformée irréversible de base est calculée avec le filtre 9/7 de Daubechies [45]. La transformée réversible est réalisée avec le filtre Le Gall 5/3 [47]. Le signal doit être d'abord élargi périodiquement [30]. Cette extension est utilisée pour assurer la correspondance entre les coefficients du masque de filtrage et les coefficients qui se trouvent au bord du signal. La dimension de l'extension dépend de la taille du masque de filtrage.

La transformée est appliquée indépendamment sur chaque tuiles et sur chaque composante. Elle est réalisée en deux étapes, à l'aide d'une décomposition en sous-bandes à une dimension, appliquée tout d'abord de façon horizontale puis verticale et cela de façon itérative jusqu'à ce que le nombre de niveaux de décomposition ait été atteint. On décompose suivant N niveaux de résolution, ces niveaux de résolution contiennent alors les coefficients d'ondelettes qui décrivent les caractéristiques spatiales locales pour un certain facteur d'échelle la tuile . Chacune se décompose en sous-bandes qui contiennent les contributions fréquentielles verticales, horizontales et diagonales de l'image pour un niveau de résolution donné.

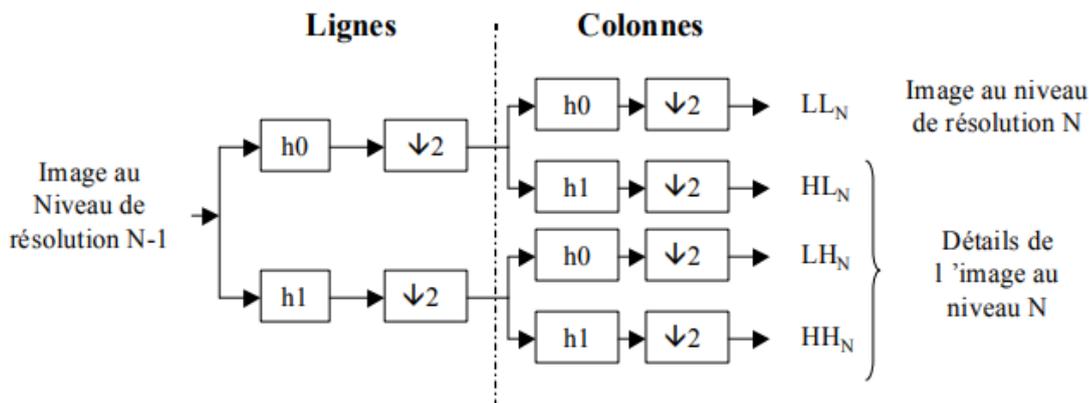


Figure 3.7 : Banc de filtres d'analyse pour la transformée en Ondelettes Lors du décodage.

L'image sera reconstruite en opérant la transformée en ondelettes inverse selon le schéma suivant :

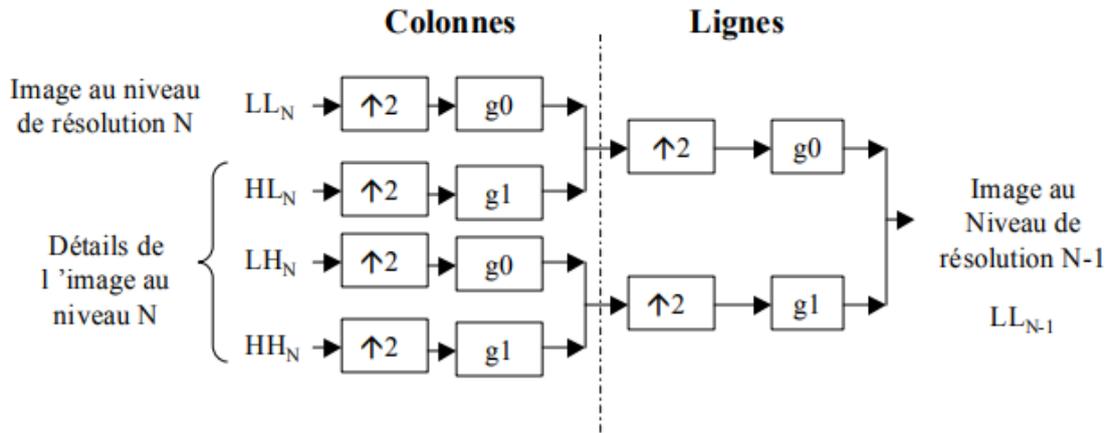


Figure 3.8: Transformée ondelettes inverse.

Pour réduire la complexité algorithmique, JPEG2000 propose d'implanter la transformée en ondelettes en utilisant la méthode du lifting (lifting-scheme) [28,29,33]. Celle-ci permet de minimiser le nombre d'opérations ainsi que la taille de la mémoire, en réalisant simultanément les filtrages passe-bas et passe-haut. Le filtrage convolutif réalise une série de produits entre les deux filtres et le signal. Le filtrage basé sur le lifting est constitué d'une séquence très simple d'opérations de filtrage, pour lesquelles les valeurs paires alternatives du signal sont remplacées par la sommation des valeurs impaires, et les valeurs impaires du signal sont remplacées par la sommation des valeurs paires [2], [14], [6], [28], [29]. Pour le cas réversible (non destructif), les résultats sont arrondis à une valeur entière.

La figure suivante représente un exemple de la compression DWT

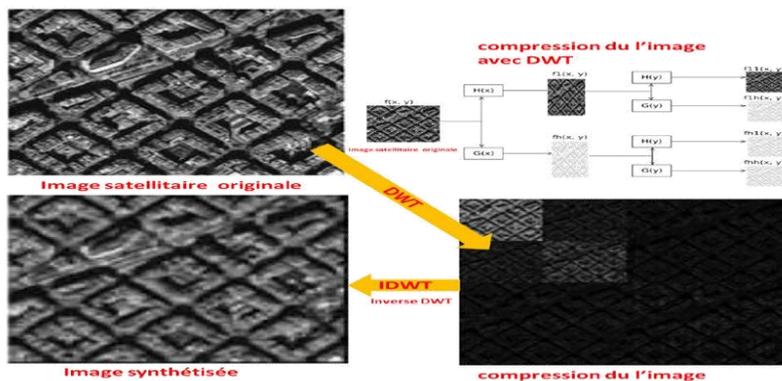


Figure 3.9 : étapes de compression d'une image par DWT.

III-6.1. Le lifting schéma :

Il a été introduit en 1994 par Wim Sweldens. Cette méthode d'implantation de la transformée en ondelettes s'affranchit totalement de la notion de dilatation et de translation, et ne repose donc plus sur la transformée de Fourier.

Le schéma lifting [13, 18, 35, 46, 48] s'effectue en trois étapes présentées dans la figure suivante :

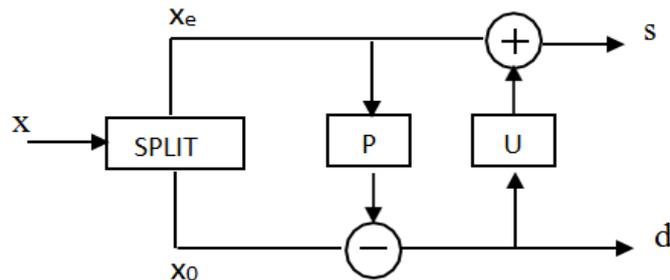


Figure 3.10 : Principe du schéma lifting.

Le signal original, noté $\{x\}$, est partitionné (opérateur " SPLIT" dans le schéma) en deux : $\{xe\}$ (l'ensemble des échantillons d'indice pair) et $\{x0\}$ (l'ensemble des échantillons d'indice impair). Une opération de prédiction P est appliquée au sous-ensemble xe . Nous posons donc :

$$d = x0 - P(xe) \quad \text{(III-1)}$$

L'erreur d de prédiction sur $\{x0\}$ représentera alors l'ensemble des coefficients d'ondelettes, c'est à dire les détails de l'image. Cette étape définit ce qu'on appelle un pas " lifting ".

De même, un opérateur de mise à jour est appliqué à $\{x0\}$ afin de pallier aux problèmes d'effets de bord (conservation d'énergie ou conservation de la moyenne des intensités lumineuses). Soit :

$$s = xe + U(d) \quad \text{(III-2)}$$

Suivant le filtre de transformée en ondelettes utilisé, nous pouvons avoir plusieurs étages de pas lifting et de pas lifting dual.

L'avantage de cette méthode est que l'algorithme est simple et l'inversion facile.

Factorisation de la matrice polyphasée

On peut obtenir un schéma lifting à partir d'un schéma classique de filtrage (figure 3.11).

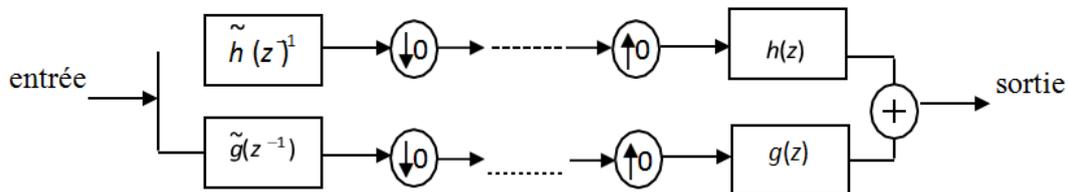


Figure 3.11 : Schéma classique de décomposition-reconstruction.

La transformation classique utilise deux filtres h (passe-bas) et g (passe-haut) pour l'analyse, chaque filtrage étant suivi d'un sous-échantillonnage. La transformation inverse consiste en premier lieu à échantillonner les entrées, puis à appliquer deux filtres h (passe-bas) et g (passe-haut) pour la synthèse. La représentation polyphasé (figure 3.12) de la transformée en ondelettes est obtenue par un sous-échantillonnage en éléments d'indices pair et impair, puis par l'application de la matrice polyphasé duale.

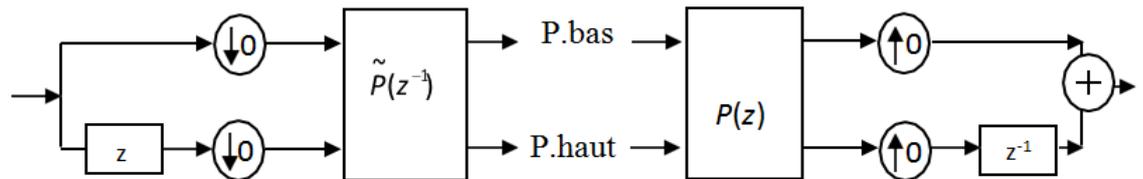


Figure 3.12 : Représentation polyphasé.

III-6.2. Lifting 2D :

La façon la plus directe est d'étendre au cas bidimensionnel est de considérer le produit tensoriel. On a alors deux étages de lifting successivement sur les lignes et les colonnes.

La compression par ondelettes, particulièrement appropriée pour les images, est vraisemblablement la méthode de compression la plus performante à l'heure actuelle et est la base du JPEG2000. Ses avantages sont multiples : elle permet une compression fois plus importante que le JPEG, offre une meilleure définition des détails et permet un téléchargement progressif de l'image.^[49]

III-7. Quantification :

La transformation en ondelettes est suivie d'une quantification scalaire uniforme permettant de réduire la dynamique des données. Chaque coefficient de la transformation $c_b(u,v)$, de la sous-bande b est quantifié à valeur $q_b(u,v)$ suivant l'équation ^[47,49] .

$$q_b(u,v) = \text{signe}(c_b(u,v)) \left\lfloor \frac{|c_b(u,v)|}{\Delta_b} \right\rfloor$$

Où Δ_b est le pas de quantification en fonction de la dynamique et du type de la sous-bande, du nombre de niveaux de décomposition. La dimension du pas de quantification Δ_b est représentée par rapport à la dimension dynamique de la sous-bande b .

Δ_b sera représenté sous une forme (ϵ_b, μ_b) correspondant à :

$$\Delta_b = \left(1 + \frac{\mu_b}{2^{11}} \right) \cdot 2^{Rb - \epsilon_b}$$

III-8. Règle de Déquantification :

La valeur de déquantification se calcule comme suit ^[50] :

Règle de Déquantification :

$$z = [q + r \text{ sign}(q)] \Delta_b, \text{ pour } q \neq 0$$

$$z = 0, \text{ autrement. (III-3)}$$

Où q est l'index de quantification, Δ_b est le pas de quantification, z est la valeur reconstruite de signal, le $\text{sign}(q)$ dénote le signe de q , et r est la polarisation de reconstruction.

- $r = 0.5$ résultat dans la reconstruction de midpoint (aucune polarisation).
- $r < 0.5$ polarisation de la reconstruction vers zéro. Une valeur populaire pour r est 0.375.

III-9. Régions d'intérêt :

Une région d'intérêt (ROI pour Region of Interest) est une région de l'image codée avec une plus grande précision car elle présente un intérêt particulier (ex : visage, plaque d'immatriculation, ... etc.). Cette plus grande précision se fait au détriment des autres zones de l'image qui sont alors compressées à un taux inférieur et donc dégradées ^[44, 45].

La forme de la région d'intérêt n'est pas signalée au décodeur, seulement la présence d'une région d'intérêt et la valeur du décalage opéré sont signalées. Au décodage, les données seront redécalées de façon à retrouver la dynamique d'origine.

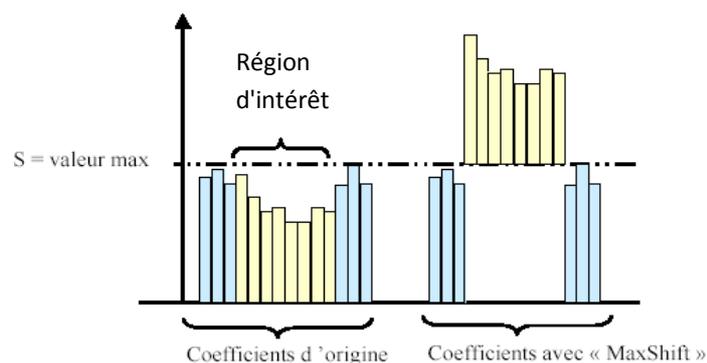


Figure 3.13: Localisation de la ROI ^[30].

III-10. Codage entropique :

Le codage entropique dans JPEG2000 est divisé en deux étapes de codage : Tier-1 et codage Tier-2. L'algorithme de codage dans cette norme est très complexe.

III-10.1. Codage tier-1 :

Cette partie est divisée en deux étapes de codage :

1) Codeur EBCOT : Il s'agit de l'entité qui parcourt les bits d'un code-bloc et qui envoie au codeur arithmétique binaire adaptatif MQ une succession de couples (bit à encoder (décision) et contexte du bit (CX) associé).

2) Codeur arithmétique binaire adaptatif MQ : Appelé aussi codeur MQ ; il est chargé de coder ces couples (bit à encoder (décision) et contexte du bit (CX)).

III-10-1-1. Codeur EBCOT :

Après la quantification scalaire, les coefficients des différentes sous-bandes sont codés avec un algorithme EBCOT (Embedded Block Coding with Optimized Truncation) [38, 39].

Cet algorithme a été créé en majorité par David Taubman, le principe de base de l'EBCOT est le suivant : Après la quantification scalaire, les coefficients issus des différentes sous-bandes de la transformée en ondelettes sont rangés en blocs, appelés code-blocs (figure (3.14.1)), de forme rectangulaire, de taille paramétrable, leurs hauteurs et largeurs correspondant à une puissance de deux, et le produit largeur hauteur ne devant pas dépasser 4096 (64 x 64) avec une hauteur et largeur minimale de 4. Chaque code-bloc est codé indépendamment, sans aucune référence aux autres code-blocs de la même sous-bande ou d'une autre. Ce codage indépendant offre des avantages importants, comme un accès spatial aléatoire au contenu de l'image, calcul parallèle durant le codage ou décodage [40]

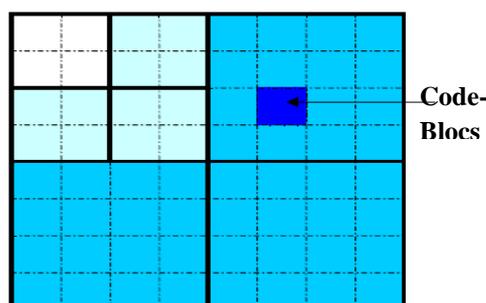


Figure 3.14.1: Partitionnement des sous bandes en code-blocs.

Les code-blocs sont en fait des tableaux de coefficients qui peuvent être représentés par un tableau tridimensionnel binaire constitué de plusieurs plans de bits. La figure (3.14.2) illustre comment les coefficients dans chaque code-bloc sont rangés en plans de bits.

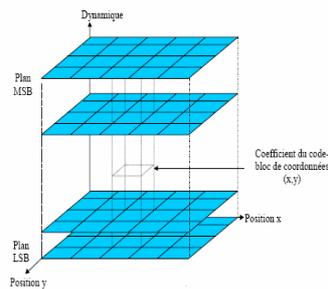


Figure 3.14.2 : Représentation d'un code –bloc en plan de bits.

Pour le codage des bits des plans de bits. Chaque plan de bits est balayé par bande de quatre bits de haut de gauche vers la droite et stripe par stripe de haut en bas comme le montre la figure (3.14.3) :

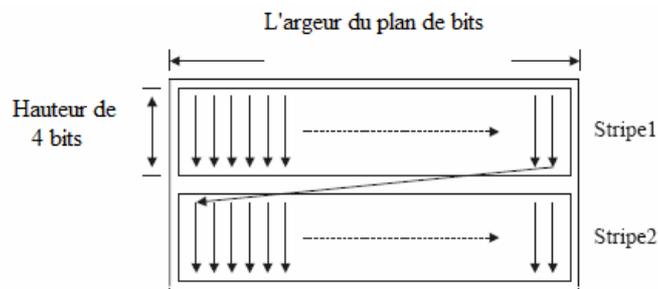


Figure 3.14.3 : Ordre de balayage d'un plan de bits.

III-10-2. Codage Tiers 2 (Organisation du Bitstream) :

Une particularité du JPEG 2000 est que plusieurs débits peuvent être spécifiés au codage. Dans ce cas, la séquence binaire (bitstream) de chaque code-bloc sera coupée en plusieurs endroits.

La réunion des premières portions des séquences binaires de chaque code-bloc forme une première couche (layer) de qualité. L'ajout des secondes portions de chaque code-bloc forme la deuxième couche apportant ainsi un incrément de qualité à la totalité de l'image (ou tuile), et ainsi de suite [33]. On peut remarquer que dans certaines couches le bitstream d'un code bloc peut être vide. La totalité du flux binaire codé est ordonnée pour former le code-stream [42]. Ce dernier est ensuite encapsulé dans un fichier au format JP2 selon les spécifications de la norme.

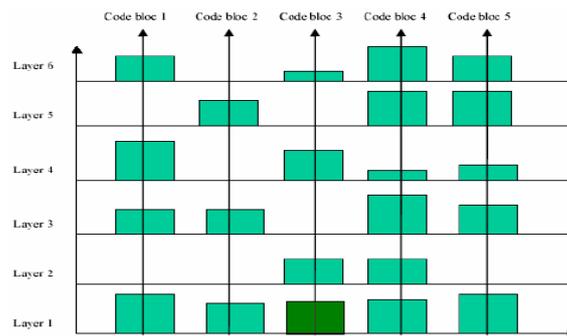


Figure 3.14.4 : Illustration du concept de couches de qualité.

III-11. Contrôle de taux : [52]

Dans le codeur, le contrôle du taux ^[2, 3] est accompli par deux mécanismes distincts :

-Choix des pas de quantification : Il fixe des pas de quantification plus ou moins grands afin d'arriver au débit binaire voulu. Cependant cette méthode n'est pas la meilleure parce qu'à chaque fois que l'on modifie le pas de quantification il faut refaire intégralement le premier étage de codage.

- sélection des passes de codage qui seront inclus dans le flux de données final

(codstream) : Cette méthode est la meilleure car elle permet d'enlever rapidement des données superflues et elle n'affecte que le deuxième étage de codage.

III-12. Résistance aux erreurs :

la norme propose plusieurs outils, dont la philosophie est essentiellement de compartimenter les mots de codes ou de réduire leur longueur afin d'éviter la propagation des erreurs ou de limiter leurs effets. Les outils proposés sont les suivants :

Marqueur de segment : Ces marqueurs sont insérés après chaque plan de bits et sont codés arithmétiquement. Leur bon décodage indique que le plan de bit courant a été correctement décodé. Inversement, si le marqueur n'est pas trouvé, le plan de bits sera considéré comme erroné et donc supprimé.

Terminaison à chaque passe : C'est un moyen de limiter la propagation des erreurs, en compartimentant les données de façon fine (à chaque passe). Le décodeur arithmétique peut ainsi continuer le décodage en cas d'erreur.

Marqueur de resynchronisation (SOP/EPH) : Ces deux marqueurs indiquent le début et la fin de chaque paquet et permettent au décodeur de se synchroniser grâce au numéro de paquet inclus dans ces marqueurs. Il s'agit ici de marqueurs de syntaxe (non codés arithmétiquement) et dont la gestion est propre à chaque décodeur.

III.13. Résumé des étapes de la création d'un fichier JPEG 2000 :

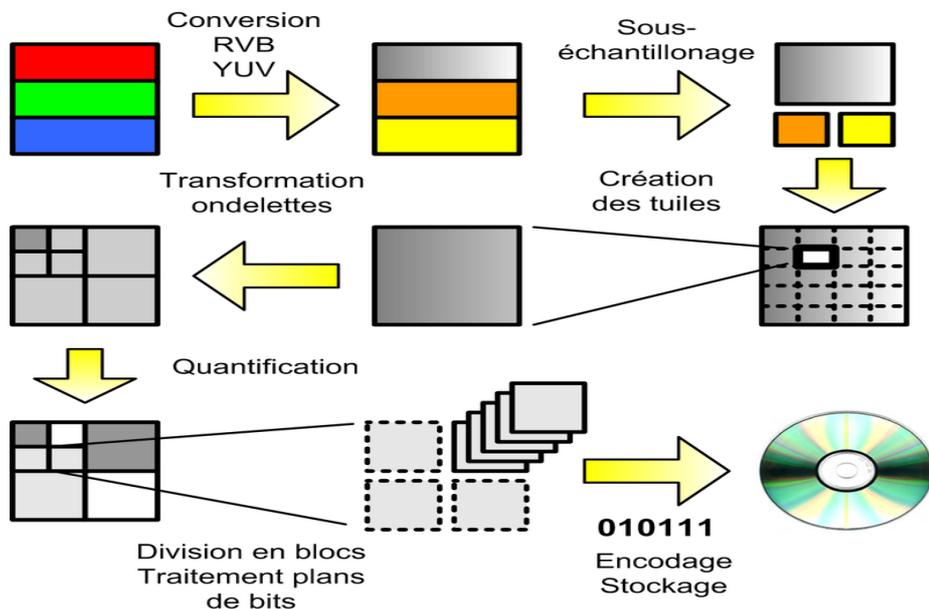


Figure 3.15 : Résumé des étapes de la création d'un fichier JPEG 2000 ^[52]

III-14. Applications :

Certains marchés et applications destinés à être desservis par cette norme sont énumérés ci-dessous :

- Les applications grand public comme les appareils multimédia (par exemple, les appareils photo numériques, assistants numériques personnels, les téléphones mobiles 3G, fac-similé couleur, imprimantes, scanners, etc.)
- communication client / serveur (par exemple, Internet, base de données de l'image, le streaming vidéo, serveur vidéo, etc.)
- Militaire / surveillance (par exemple, des images satellite HD, détection de mouvement, la distribution et le stockage réseau, etc.)
- Imagerie médicale, esp. les DICOM spécifications pour l'échange de données médicales.
- Biométrie.
- Télédétection
- Cinéma numérique

III-15. Avantages de la compression JPEG 2000 : ^[53]

La compression JPEG 2000 permet une plus grande efficacité de compression, en particulier à bas débit, pour des applications telles que la transmission d'image sur réseaux et la télédétection

(remotesensing). Par exemple, moins de 0,25 bpp (bit per pixel) pour des images en niveaux de gris avec beaucoup de détails.

D'autres caractéristiques requises concernent :

La description basée sur le contenu, pour retrouver une image dans de grandes banques de données ; de meilleures performances lors de multiples cycles de compression-décompression ;

Un format de fichier plus flexible.^[54]

III-16. Différence entre JPEG et JPEG 2000 :

JPEG signifie Joint Photographique Expert Group. Le nom de fichier d'une image JPEG est .jpg ou .jpeg. JPEG est le format le plus couramment utilisé pour les photographies. JPEG2000 est un autre format du Joint Photographique Expert Group. Il a été publié en 2000. C'était la tentative du groupe de mettre à niveau le format JPEG. Le plus récent, JPEG 2000 mettrait à jour la norme JPEG à 16 bits, à partir de la norme 8 bits d'origine.



Figure 3.16 :Différence between JPEG and JPEG 2000

Ce nouveau format est basé sur des ondelettes. Le nom des fichiers JPEG 2000 est «.jp2 pour les fichiers conformes à la norme ISO / CEI 15444-1 et .jpx pour les spécifications de la partie 2 étendue, publiées sous la cote ISO / CEI 15444-2».

JPEG.org, le site Web officiel du Joint Photographic Expert Group, répertorie les fonctionnalités suivantes dans le cadre du JPEG 2000 :

- Meilleure efficacité en compression
- Possibilité de compression sans pertes
- Décodage avec différentes résolutions de sortie
- Codage ROI
- Amélioration de la résilience au bruit
- Accès au débit binaire compressé à tout moment pour accéder directement à l'image
- Meilleures performances de codage / décodage à travers de nombreux cycles différents
- Format de fichier plus flexible

De plus, la technologie des ondelettes, sur laquelle repose le JPEG2000, est beaucoup plus complexe que le code utilisé pour les images JPEG, ce qui signifie que la lecture et l'écriture d'images

JPEG2000 prendront plus de temps. Il faudra également un logiciel plus complexe et plus de mémoire pour exécuter le logiciel complexe. Par conséquent, compresser avec des images JPEG2000 prend généralement plus de temps que travailler avec des images JPEG.^[55]

format	Nom	L'extension De fichier	Type de compression	Méthode	L'usage
JPEG	Joint photographic Experts Group	JPG ou JPE ou JPEG	Compression avec pertes	DCT/RTE	Pour nature Images
JPEG 2000	Joint photographic Experts Group	Jp2 ou j2k ou JPF ou jpg2	Compression avec pertes	DWT	Pour meilleure qualité d'image (20%. Meilleur que jpeg)

Tableau 3.2 : Différence ente JPEG et JPEG 2000

III-17. CONCLUSION

Dans ce chapitre nous avons présenté les principales fonctionnalités proposées dans JPEG 2000. Nous avons pu voir que JPEG2000 possède tous les outils nécessaires pour devenir une norme universelle, par pallier aux limitations de la norme JPEG

CHAPITRE IV

Simulation, Résultats

et Discussion

IV.1.Introduction :

Notre travail consiste à compresser une image fixe ‘‘LENA’’ en niveau de gris de résolution 256×256 pixels en utilisant l’outil de programmation MATLAB version R2015a

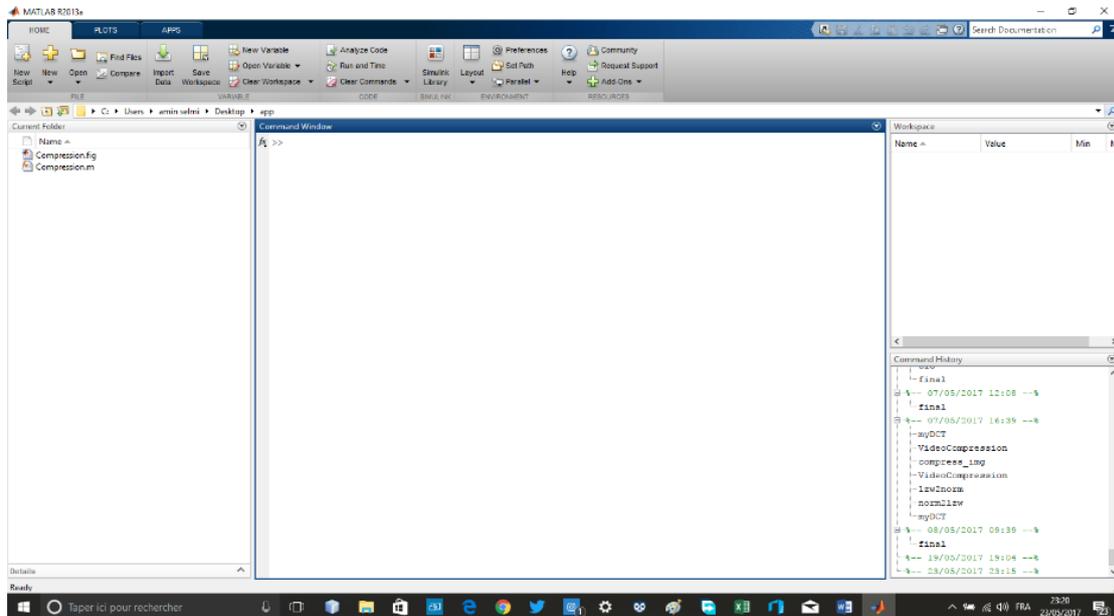


Figure 4.1 : Interface graphique et IDE de MATLAB R2015a.

Qu’est-ce qu’une bonne compression ?

Un algorithme performant de compression possède un gain de compressions maximales et une erreur quadratique moyenne minimale

IV.2.Mesure de la qualité de la restauration :

La performance d’une méthode de restauration peut être calculée à l’aide d’un des indicateurs suivants :

- 1- Erreur quadratique moyenne (mean square error).
- 2- Rapport signal_à-bruit.
- 3- PSNR.

IV.2.1 Erreur quadratique moyenne :

L’erreur quadratique moyenne de la reconstruction \hat{l} de l’image l est définie par :

$$EQM = \frac{1}{MN} \sum_{i,j} (l(i,j) - \hat{l}(i,j))^2$$

IV.2.2.Rapport Signal à Bruit (RSB) :

Le rapport signal à bruit est défini par :

$$\text{RSB} = \frac{\sum_{i,j} \hat{I}(i,j)^2}{\sum_{i,j} (I(i,j) - \hat{I}(i,j))^2}$$

(Ou formule équivalente en échelle logarithmique).

La différence $I(i,j) - \hat{I}(i,j)$ est considérée comme un bruit.

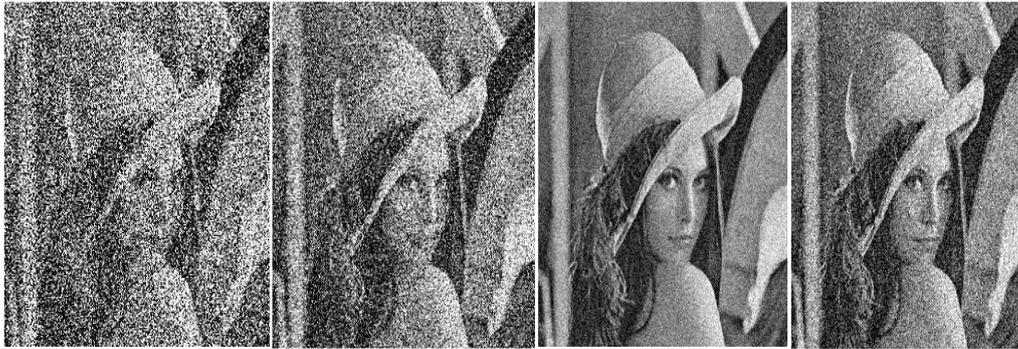
IV.2.3.Peak Signal to Noise Ratio (PSNR)

Le peak signal to noise ration est défini par :

$$\text{PSNR} = \frac{\max_{i,j} \hat{I}(i,j)^2}{\frac{1}{MN} \sum_{i,j} (I(i,j) - \hat{I}(i,j))^2}$$

(Ou formule équivalente en échelle logarithmique).

Ce critere est souvent utilise pour mesurer la qualite de la reconstruction d'une image qui a subit une compression avec perte d'information (ex format JPEG).

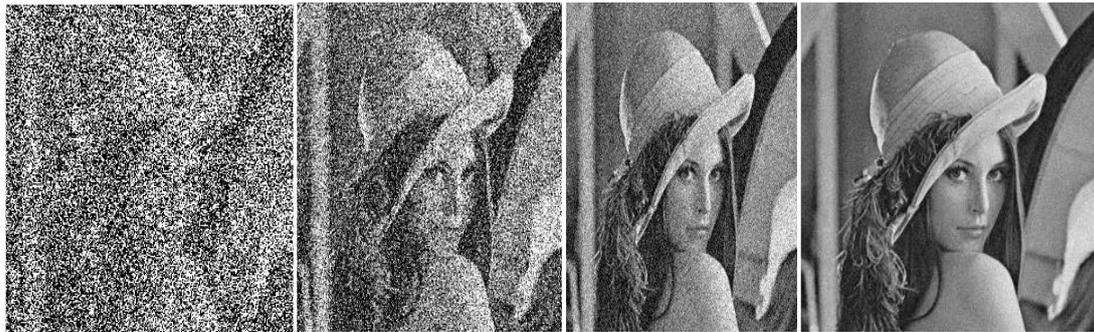
IV.3.RSB et PSNR

RSB = 0dB

RSB = 10dB

RSB = 20dB

RSB = 30dB



PSNR = 0dB

PSNR = 10dB

PSNR = 20dB

PSNR = 30dB

IV.4. Exécution de notre projet :***IV-4-1. Tests d'images au niveau de gris :***

IV.4.1.1. Images de test :



Le tableau (4-1) présente les résultats obtenus en terme de PSNR (dB) sur l'images de tests : "Lena" avec différents taux de bits (Tb) en bits par pixel (bpp) : 0.125 bpp, 0.25 bpp, 0.5 bpp, 1.0 bpp.

Taux de bits (Tb) en bpp	JPEG		JPEG2000	
	PSNR(en dB)	Quotient	PSNR(en dB)	Quotient
0.125	21.92	64	30.16	64
0.25	30.77	32	33.20	32
0.5	34.75	16	36.36	16
1.0	38.25	8	39.31	8

Tableau 4.1: Présente les résultats obtenus en terme de PSNR (dB) sur l'images de tests.

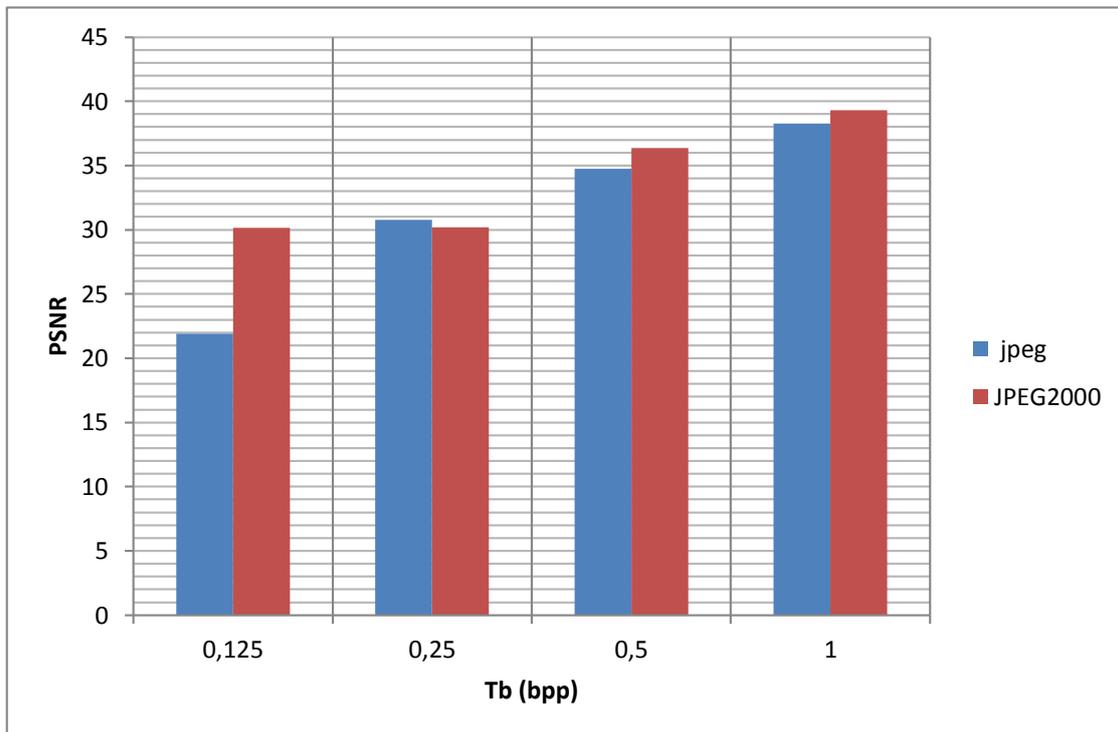


Figure 4.2: PSNR (dB) en fonction des différents Tb (bpp) pour l'image LENA.



PSNR=21.92 db, $T_b=0.125$ bpp,
Quotient = 64. JPEG.



PSNR = 30.94 db, $T_b= 0.125$ bpp,
Quotient = 64. JPEG2000.



PSNR=30.77 db, $T_b= 0.25$ bpp,
Quotient = 32. JPEG



PSNR= 34.12 db, $T_b= 0.25$ bpp,
Quotient = 32. JPEG2000

Figure 4.3 : Image ‘Lena ’reconstruite après une compression à 0.25 bpp à gauche par le codeur JPEG et à droite par le codeur JPEG2000 une image couleur.

IV.4.2. Test avec différents niveaux de décomposition en ondelette et code-blocs :

Pour obtenir des résultats qui permettent un meilleur PSNR / taux de bits (T_b), nous avons varié la taille des code-blocs utilisés pour la compression des images et le niveau de résolution de la décomposition en ondelettes.

Le test est effectué sur l'image ' Lena .Bmp ' pour un T_b de (0.5 bpp). Les résultats obtenus sont dans le tableau suivant (4.2) : avec PSNR en (dB), $T_b = 0.5\text{bpp}$.

Taille des blocs	2 ^{ème} résolution	3 ^{ème} résolution	4 ^{ème} résolution	5 ^{ème} résolution
8*8	35.77	36.38	36.45	36.46
16*16	36.44	36.94	36.98	36.99
32*32	36.70	37.18	37.23	37.23
64*64	36.76	37.28	37.30	37.30

Tableau 4.2 : Résultats du PSNR avec variation de la taille des code-blocs et avec différents niveaux de décomposition en ondelettes.

Après l'analyse des résultats du tableau (4-2) nous avons trouvé que le PSNR augmente avec l'augmentation de la taille des code-blocs et cela pour tous les niveaux de résolution de la décomposition en ondelettes.

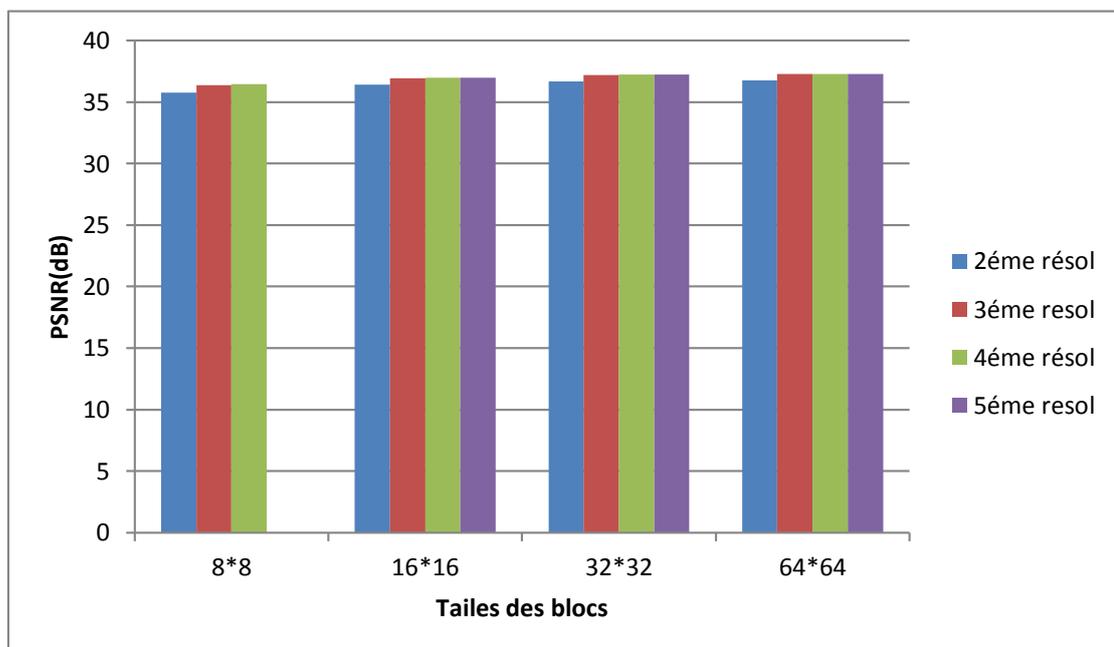


Figure 4.4 : Test avec différents niveaux de décomposition en ondelette et code-blocs

IV.4.3. Test sur une image couleur :

Test effectué de la compression sur l'image couleur Lena de résolution 512 * 512 pixels codée sur 24 bits / pixel (figure (4.5)) avec différents taux de bits (Tb) pour les trois composantes R(rouge) V(vert) B(bleu)



Figure 4.5 : Image originale 'Lena .Bmp'.

Taux de bits (bpp)	Quotient	JPEG			JPEG 2000		
		R	V	B	R	V	B
0.125	192	11.7	15.95	15.53	27.79	27.95	26.10
0.25	96	25.41	26.7	24.00	29.57	30.10	27.43
0.5	48	29.22	30.75	27.25	30.66	31.40	28.42
1.0	24	30.50	32.89	28.58	31.51	33.01	29.15

Tableau 4.3 : Résultats des PSNRs sur l'image Lena obtenus avec les deux codecs pour différents Tb (bpp).

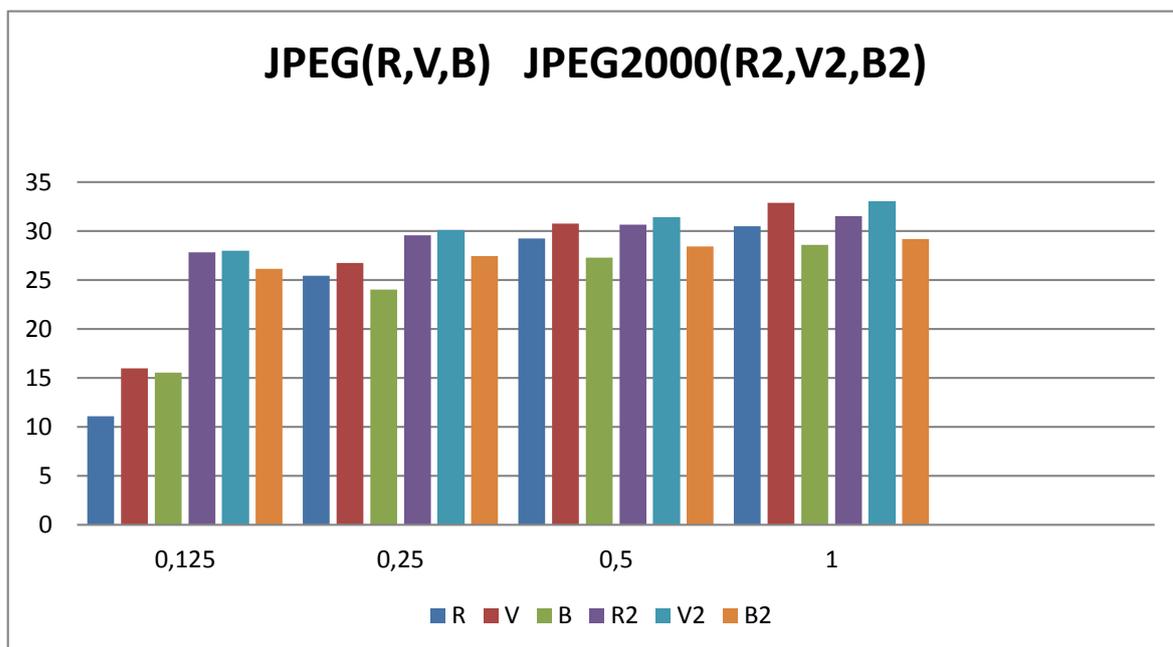


Figure (4.6) : PSNRs (dB) pour chaque composante (Rouge, Vert, Bleu) en fonction des différents Tb (bpp) obtenus sur l'image Lena.

Après avoir calculé le PSNR des deux normes de compression de l'image couleur lena pour les trois composantes R V B rouge vert et bleu nous avons remarqué que le PSNR de JPEG2000 de chaque composante est meilleur.



Tb= 0.125 bpp.

Quotient =192. JPEG



Tb = 0.125 bpp.

Quotient =192. JPEG2000

Figure 4.6.1 : Image 'Lena' reconstruite après une compression à 0.125.



Tb = 0.25 bpp,

Quotient = 96. JPEG



Tb=0.25bpp

Quotient = 96. JPEG2000

Figure 4.6.2 : Image 'Lena' reconstruite après une compression à 0.25 bpp.

IV.4.4. Test Scalabilité en qualité sur une image au niveau de gris Lena :

L'image de test Lena est compressée avec trois taux de bits (0.125 bpp, 0.25 bpp, 0.5 bpp) en même temps et les résultats des images de reconstructions sont cités dans les figures suivantes :



**Figure (4.7.1) : Image 'Lena' reconstruite
À 0.125 bpp par le codeur JPEG2000.**



**Figure (4.7.2) : Image 'Lena' reconstruite
À 0.25 bpp par le codeur JPEG2000.**



**Figure (4.7.3) : Image ‘Lena ’reconstruite
À 0.5 bpp par le codeur JPEG2000.**



**Figure (4.7.4) : Image ‘Lena ’reconstruite
À 1.0 bpp par le codeur JPEG2000.**

IV.4.5. Test Région d'intérêt (ROI) sur une image au niveau de gris Lena :

La qualité d'une image se diffère selon le poids d'une région à l'autre la figure suivante (4.8) montre une image Lena compressée avec un taux de bits de 0.25bpp dans une zone de Région d'intérêt rectangulaire.



Figure 4.8 : Image ‘Lena à 0.255 bpp de la région d'intérêt par le codeur JPEG2000.

IV.5.CONCLUSION

Après avoir obtenu les résultats ci-dessus sur l'image de lena, nous remarquons que la norme de compression JPEG2000 basée sur la transformée en ondelettes et pour les faibles taux de bits est meilleure en comparant avec JPEG et même si nous augmentons la valeur de taux de bits la différence entre les deux normes n'est pas visuellement perceptible.

Nous avons remarqué aussi après avoir effectué une étude comparative JPEG2000 /JPEG sur la même image lena que la norme JPEG2000 donne de meilleurs résultats de points de vue qualité d'image et taux de compression par rapport à JPEG

ET nous avons trouvé que cette nouvelle norme JPEG2000 pour la compression d'images fixes offre de nouvelles fonctionnalités (ROI, progressivité en qualité et en résolution) ces fonctionnalités permettent de couvrir les besoins actuels dans plusieurs domaines tel que le domaine de l'imagerie médicale et l'internet et la photographie numérique.

CONCLUSION GÉNÉRALE ET PERSPECTIVE

Dans ce travail, nous avons étudié différentes méthodes de compression d'images fixes en niveau de gris.

Vu que le JPEG2000 donne de meilleurs résultats que le JPEG, il serait intéressant d'étudier éventuellement les performances de cette norme pour différents types d'images de tests et pour différents types d'ondelettes.

Une étude des performances de la technique de compression JPEG 2000 par rapport à JPEG a été faite. Des Programmes et des fonctions sous MATLAB ont été développés afin de faciliter la compression et la décompression des images et l'évaluation des tests. Nous avons surtout évalué les performances en termes de qualité visuelle de l'image après décompression entre le codec JPEG2000 par rapport à JPEG dans le cas du mode avec pertes pour différents taux de bits.

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.

Les résultats obtenus montrent l'efficacité du codage JPEG2000 par rapport au codage JPEG des points de vue quantitatif et qualitatif.

Bibliographie

- [1,10,24,51] M. SAHIR, « Compression des images numériques par la technique des ondelettes », Mémoire de Magister, soutenu le 19 juin 2011 à l'Université de Sétif.
- [2] E. LE PENNEC, « Compression d'image » — Images des Mathématiques, CNRS, 2006.
- [3] J.-N. GOUYET, C. NELSON et M. LEGER, « JPEG 2000, format de compression d'image numérique par ondelettes – Principes », Techniques de l'Ingénieur, 2016.
- [4] M. CAGNAZZO, « Ondelettes et compression d'images », PPT TELECOM-Paris Tech, Paris, 2012.
- [5] A. OUAHABI, « Analyse multirésolution pour le signal et l'image », Hermes-Lavoisier, Paris, 2012.
- [6] R. C. GONZALEZ et R. E. WOODS, « Digital image processing », Prentice-Hall, N. J., 2008.
- [7,8] https://fr.wikipedia.org/wiki/Image_num%C3%A9rique
- [9] <https://waytolearnx.com/2018/07/difference-entre-la-compression-avec-perte-et-la-compression-sans-perte.html> visitée le 08/05/2020
- [11] <http://d.nouchi.free.fr/TER/c6.html>, visitée le 20/05/2020.
- [12] <http://www-igm.univ-mlv.fr/lecroq/cours>, visitée le 25/05/2020
- [13] [http://www-igm.univ-mlv.fr/dr/XPOSE2013/La compression de donnees/jpeg.html](http://www-igm.univ-mlv.fr/dr/XPOSE2013/La%20compression%20de%20donnees/jpeg.html), visitée le 20/06/2020.
- [14] Jean-Michel DOUDOUX. D_eveloppons en Java. <Http://www.jmdoudoux.fr>, visitée le 01/08/2020
- [21] Rao, K.R., and Yip, P, “ Discrete Cosine Transform - Algorithms, Advantages ” , Applications. Academic Press, Inc. London, 1990.
- [22] C. Huang, S. Ravi , A. Raghunathan , and N. K. Jha , “ Eliminating Memory Bottlenecks for a JPEG Encoder Through Distributed Logic-Memory Architecture and Computation-unit Integrated Memory ” , IEEE Custom Integreted Circuits Conference , 20005.
- [23] Léger, A., J. Mitchell, and Y. Yamazaki, “ Still Picture Compression Algorithms Evaluated for International Standardization ” , Proceedings of the IEEE Global Telecommunications Conference, pp. 1028-1032, November 1988.
- [25] Wallace, Gregory K, “ The JPEG still picture compression standard ” ,IEEE Transactions on Consumer Electronics. vol 38, no .1, pp. 18-34. 1992.
- [26] Gregory K. Wallace , “ Information Technology – Digital Compression and Coding of Continuous-Tone Still Image – Requirements and Guidelines”, Copyright by ITU, 1993.
- [27] W. B. Pennebaker, J. L. Mitchell, “ JPEG Still Image data compression standard ” , VanNostrand Reinhold, NY, 1993.

Bibliographie

- [28] Klaus Jung and Ruedi Seiler, “ Segmentation and Compression of Documents with JPEG2000 ”, *IEEE Trans. On Consumer Electronics*, vol. 49, no. 4, November 2003.
- [29,31] S.G. Mallat, “ A theory for multiresolution signal decomposition : the wavelet representation ”, *IEEE trans.Pattn.Anal. Mach. Intell.* vol. 11, no7. pp.674-693. July 1989.
- [30] Klaus Jung and Ruedi Seiler, “ Segmentation and Compression of Documents with JPEG2000 ”, *IEEE Trans. On Consumer Electronics*, vol. 49, no. 4, November 2003.
- [32] D. Santa-Cruz, R. Grosbois, T. Ebrahimi, “ JPEG 2000 performance evaluation and assessment ”, *Signal Processing: Image Communication* 17 .pp.113–130,2002.
- [33] B. E. Usevitch, “ A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000 ”, *IEEE Signal. Processing Mag.*, vol. 18, pp .22-35, Sept. 2001.
- [34] M.D. Adams, “ The JPEG2000 Still Image Compression Standard ”, *ISO/IEC JTC1 /SC29/ WG1 (ITU-TSG16)*, December 2005.
- [35] C. Christopoulos, A. Skodras, T. Ebrahimi, “ The JPEG2000 Still Image Coding System : An Overview ”, *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 4,pp.1103- 1127, November 2000.
- [36] Y. Kumar Jain, S. Jain, “ Performance Evaluation of Wavelets for Image Compression ”, *Asian Journal of International Technology* 5(10) , pp. 113–130,2002.
- [37] D. Le Gall and A.Tabatabai, “Subband Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Techniques”, *Proc IEEE Int. Conf. ASSP, NY*, pp.761-765, 1988.
- [38] https://commons.wikimedia.org/wiki/Image:Huffman_tree.svg consulte le 20/08/2020 a 11pm
- [39] M. Antonini, M. Barlaud, P. Mathieu and I.Daubechies: “Image Coding Using theWavelet Transform”, *IEEE Trans. Image Proc.*, pp. 205-220, April 1992.
- [40] D. Taubman, “ High Performance Scalable Image Compression with EBCOT ”, *IEEE Transactions on Image Processing*, vol. 9, no. 7, July 2000.
- [41] D. Taubman, E. Ordentlich, M. Weinberger, G.Seroussi, I. Ueno, and F. Ono, “ Embedded block coding in JPEG2000 ”, in *Proc. IEEE Int. Conf. Im- age Processing, Vancouver, Canada*, vol. 2, pp. 33-36, Sept. 2000.
- [42] A. Skodras, C. Christopoulos, and T. Ebrahimi, “ The JPEG 2000 still image compression standard ”, *IEEE Signal. Processing Mag.*, vol. 18, pp 36-58, Sept. 2001.
- [43] M. Boliek, C. Christopoulos, et E. Majani, “ JPEG 2000 Part I Final Committee Draft Version 1.0 ”, *FCD15444-1, ISO/IECJTC1/SC29WG1 N1523* –décembre 1999.
- [44] M. Boliek, C. Christopoulos, et E. Majani, “ JPEG 2000 Part I Final Committee Draft Version 1.0 ”, *FCD15444-1, ISO/IEC JTC1/SC29 WG1 N1646R*, april 2000.
- [45] M. Boliek, C. Christopoulos, et E. Majani, “ JPEG 2000 Part I Final Committee Draft Version 1.0 ”, *FCD15444-1, ISO/IEC JTC1/SC29 WG1 N164R*, juillet 2001.

Bibliographie

[46] C. Christopoulos, J. Askelof, and M. Larsson, “ Efficient methods for encoding regions of interest in the upcoming JPEG 2000 still image coding standard”, IEEE Signal Processing Letters, vol. 7, no. 9, pp. 247–249, Sept. 2000.

[47] J. Askelof, M. Larsson Carlander, and C. Christopoulos, “ Region of interest coding in JPEG 2000 ”, Signal Processing: Image Communication, vol. 17, no. 1, pp. 105–111, Jan. 2002.

[48] <http://www.kakadusoftware.com/>

[49] <http://www.ijg.org/>

[50] <http://www.med.univ-rennes1.fr/cerf/iconocerf/idx/T/THORAX.html>.

[52] G.K. Wallace, The JPEG Still Picture Compression Standard, IEEE Transactions on Consumer Electronics, December 1991.

[53] L. H. Viêt, Etude de la méthode de la transformation en ondelette et l'application a la compression des images, Rapport final de TIPE ,15 juillet 2005.

[54] D. Taubman, M. W. Marcellin, JPEG2000: Standard for Interactive Imaging ,Proceedings of the IEEE, Vol. 90, No. 8, August 2002.

Les sites internet.

<http://www.jpeg.org/JPEG2000.html>

http://fr.wikipedia.org/wiki/JPEG_2000

www.electronique.biz/

<http://www.ruses.com>

www.ieee.com