

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DE BATNA
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'INFORMATIQUE



MEMOIRE

Présenté au
DEPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de
Magister en informatique

OPTION
Ingénierie des systèmes informatique

Par
BOUCETTA ALDJIA

Thème :

**Etude de l'effet des Transformées de
Décorrélation en Compression des
Images Couleurs RGB**

Soutenu le :...../...../2010

Devant le jury constitué de :

Dr. Lahlouhi Ammar	M .C	U. Batna	Président
Dr. Babahenini Mohamed Chaouki	M.C	U. Biskra	Examineur
Dr. Gasmi AbdelKader	M.C	U. M'Sila	Examineur
Dr. Benzid Redha	M.C	U. Batna	Rapporteur
Dr. Seghir Rachid	M.C	U. Batna	Co-rapporteur

REMERCIEMENTS ET GRATITUDE

Ma haute gratitude, mes profonds respects et mes sincères remerciements et reconnaissances à mon rapporteur Monsieur Benzid Redha, maître de conférence à l'université de Batna qui m'a guidé avec grande patience tout au long de l'élaboration de ce travail et pour ses aides précieuses qui ont judicieusement éclairé mon chemin vers l'aboutissement et la réussite de la concrétisation de mon mémoire.

Je tiens à exprimer mes plus vifs remerciements à mon Co-rapporteur Monsieur Seghir Rachid, maître de conférence à l'université de Batna pour ses aides, son accompagnement et ses éclairages techniques. Je lui dois les remerciements les plus sincères.

Je remercie les membres de jury qui ont accepté de juger ce travail et d'y apporter leur caution :

Monsieur, **Dr. Lahlouhi Ammar**, maître de conférence à l'université de Batna, qui me fait le grand honneur d'accepter la présidence du jury.

Monsieur, **Dr. Babahenini Mohamed Chaouki**, maître de conférence à l'université de Biskra, pour l'honneur qu'il me fait en acceptant de participer à ce jury.

Monsieur, **Dr. Gasmi AbdelKader**, maître de conférence à l'université de M'Sila, pour l'honneur qu'il me fait en acceptant également de participer à ce jury.

J'adresse mes vifs remerciements à tous les enseignants qui, par leur enseignement, leur encouragement et leur aide, ont contribué à ma formation.

Table des matières

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GENERALE 1

Chapitre I : Généralités sur les images

I.1. Introduction	3
I.2. Définition de l'image.....	3
I.3. Image numérique	3
I.4. Les caractéristiques d'une image numérique	4
I.5. Les différents types d'images	6
I.6. Les formats standards d'image	8
I.7. Généralités sur les traitements d'image.....	10
I.8. Conclusion.....	12

Chapitre II : Transformation d'espaces de couleurs et étude énergétique

II.1. Introduction.....	13
II.2. Définition de l'espace de couleurs.....	13
II.3. Les espaces CIE.....	13
II.4. Les espaces luminance-chrominance	15
II.5. Les espaces répondant à la perception humaine.....	19
II.6. Les autres espaces de représentation	24
II.7. Etude énergétique.....	25
II.8. Conclusion	28

Chapitre III : Revue des méthodes de compression

III.1. Introduction.....	2
9	
III.2. Le but de la compression d'image	29
III.3. Mesures de performance de la compression d'image.....	29
III.4. Principe général de la compression des images.. ..	31
III.5. Classification des méthodes de compression.. ..	32
III.6. Méthodes réversibles ou sans perte.....	33
III.7. Méthodes avec perte ou irréversibles.....	38
III.8. Les normes de compression d'images.....	46
III.9. Particularité des images couleurs.....	50
III.10. Conclusion	50

Chapitre IV: Méthodologie proposée et résultats de simulation

IV.1. Introduction	51
IV.2. Présentation de la méthode	51
IV.3. Les Résultats de la simulation.....	54
IV.4. Etude comparative.....	64
IV.5. Conclusion	68
CONCLUSION GENERALE	69
BIBLIOGRAPHIE.....	70

LISTE DES FIGURES

- Figure I.1 : Élément d'une image (le pixel).
- Figure I.2 : Echantillonnage d'une image.
- Figure I.3 : Résolution spatiale : échantillonnage.
- Figure I.4 : Résolution tonale : quantification.
- Figure I.5 : Image et histogramme associés.
- Figure I.6 : Image avec texture.
- Figure I.7 : Une image binaire.
- Figure I.8 : Image en niveaux de gris.
- Figure I.9 : Image en couleurs.
- Figure I.10: Exemple d'une image en 256 couleurs avec sa palette associée.
- Figure II.1 : Composition additive des couleurs.
- Figure II.2 : Présentation des couleurs dans l'espace RGB.
- Figure II.3 : Une image RGB, avec ses composantes séparées R, G et B.
- Figure II.4 : Une image RGB, avec ses composantes X, Y et Z.
- Figure II.5 : Une image RGB, avec ses composantes Y, U, et V.
- Figure II.6 : Une image RGB, avec ses composantes Y, Cb, et Cr.
- Figure II.7 : Une image RGB, avec ses composantes Y, I, et Q.
- Figure II.8 : Une image RGB, avec ses composantes Y, Db, et Dr.
- Figure II.9 : Une image RGB, avec ses composantes Y, Pb, et Pr .
- Figure II.10 : Une image RGB, avec ses composantes O1, O2 et O3.
- Figure II.11 : L'espace Lab.
- Figure II.12 : Une image RGB, avec ses composantes L, a et b.
- Figure II.13 : Présentation des couleurs dans l'espace HSL.
- Figure II.14 : Une image avec ses composantes H, S, et L.
- II.15 : Présentation des couleurs dans l'espace HSV.
- Figure II.16 : Une image RGB, avec ses composantes H, S, et V.
- Figure II.17 : Cube CMY.
- Figure II.18 : Une image avec ses composantes C, M et Y.
- Figure III.1 : Etapes principales de compression d'images.
- Figure III.3 : Un exemple de codage par plage RLE.
- Figure III.4 : Linéarisation en ligne, en colonne et en zig-zag.
- Figure III.5 : Arbre d'attribution des codes de Shannon.

Figure III.6 : Arbre de Huffman.

Figure III.7 : Génération de l'étiquette pour la séquence "acaab".

Figure III.8 : Schéma de principe de la compression / décompression par transformation.

Figure III.9 : Décomposition d'une image en sous-bandes.

Figure III.10 : Transformation en colonnes et en lignes.

Figure III.11 : Décompression progressive.

Figure III.12 : Principe de l'algorithme JPEG avec perte.

Figure III.13 : Le parcours d'un bloc en zigzag.

Figure III.14 : Exemple d'image compressée avec JPEG.

Figure III.15 : Schéma typique d'un codeur JPEG 2000.

Figure III.16 : Schéma du principe de la compression.

Figure IV.1 : Schéma du principe de la compression.

Figure IV.2 : Schéma de décompression.

Figure IV.3 : Les différentes images originales.

Figure IV.4 : Images reconstruites (Lena, Girl) pour différentes tailles de bloc: (a) 8x8, (b) 6x16, (c) 32x32.

Figure IV.5 : Images reconstruites (Lena, Girl) dans les deux espaces : (a) RGB, (b) YIQ.

Figure IV.6 : Résultats d'application de notre méthode sur les images "Peppers", "Airplane" et "Lena" dans (a) l'espace YCbCr, (b) l'espace YIQ.

Figure IV.7 : Les images reconstruites dans l'espace YIQ avec la taille de bloc 16x16.

LISTE DES TABLEAUX

Tableau II.1 : L'énergie dans l'espace RGB.

Tableau II.2 : L'énergie dans l'espace YCbCr.

Tableau II.3 : L'énergie dans l'espace YUV.

Tableau II.4 : L'énergie dans l'espace YPbPr.

Tableau II.5 : L'énergie dans l'espace YIQ.

Tableau II.6 : L'énergie dans l'espace O1O2O3.

Tableau II.7 : L'énergie dans l'espace YDbDr.

Tableau II.8 : L'énergie dans l'espace XYZ.

Tableau II.9 : L'énergie dans l'espace HSL.

Tableau II.10 : L'énergie dans l'espace HSV.

Tableau II.11 : L'énergie dans l'espace Lab.

Tableau II.12 : L'énergie dans l'espace CMY.

Tableau III.1 : Exemple du codage de Shannon.

Tableau III.2 : Code de Huffman.

Tableau III.3 : Probabilités des symboles.

Tableau IV.1 : Résultats obtenus du PNSR et du bpp pour les différentes images.

Tableau IV.2 : Les résultats moyens.

Tableau IV.3: Comparaison des résultats avec CBTC-PF.

Tableau IV.4 : Comparaison des résultats avec la méthode DCT-Adaptive Scanning.

Introduction générale

Introduction générale

L'accroissement important de l'espace de stockage et/ou de transmission des informations visuelles pour de diverses applications telles que la télécopie, la vidéo-conférence, l'imagerie médicale et satellitaire, la télévision haute définition (HDTV), la télésurveillance, et les services d'informations sur l'Internet nous conduit à penser à élaborer des méthodes de compression de plus en plus performantes. En spécifiant les images en particulier, le stockage de ces dernières sur les mémoires de masse ainsi que leur transmission sur mobiles, ou sur Internet nécessitent leur acquisition sous format numérique (encore appelé format électronique) le plus compact que possible. Cependant, le format numérique des images est extrêmement coûteux en taille mémoire et/ou en temps de transmission. Pour résoudre ce problème qui peut limiter la faisabilité du stockage et de transmission, des techniques de compression d'images ont été élaborées pour compacter leur représentation numérique de la manière la plus fiable. A l'aide de ces techniques, le stockage et la transmission se sont rendus plus efficaces et plus rapides.

Ces techniques peuvent être classées en deux grandes catégories, suivant la nécessité de récupération parfaite ou non de l'information originale :

Nous avons d'abord la catégorie des techniques de *compression sans perte* d'information dites aussi réversibles, préservative ou à reconstruction exacte. Leur avantage majeur est la préservation de la qualité de l'image originale mais en dépit du taux de compression qui est toujours relativement faible.

L'autre classe regroupe les algorithmes dits avec perte. Ces derniers sont caractérisés par leur taux de compression assez élevé tout en préservant le plus possible la qualité originale.

Donc, si dans une application la perte en qualité quel quelle soit minime, après une compression, n'est pas tolérable, alors on opte pour la première catégorie. Sinon, l'utilisation des algorithmes avec perte est envisageable.

Il est à noter que beaucoup de techniques de compression d'images sont disponibles, telles que :

- Celles basées sur la DCT (Discrete Cosine Transform) [16], JPEG-DCT [15].
- Et les autres méthodes basées sur la transformée d'ondelette telles que : EZW (embedded coding using zerotrees of wavelet coefficients) ou SPIHT (set partitioning in hierarchical trees) [21], JPEG2000 [26] codage en sous-bande [23], etc., Le but commun de toutes ces techniques est d'obtenir un taux de compression élevé avec une qualité satisfaisante de l'image reconstruite.

Habituellement, les images couleurs sont stockées sous le format RGB (Red Green Blue : Rouge Vert Bleu) [12] [15]. Or le système RGB n'est pas le mieux adapté pour le processus de compression, puisqu'il y a une forte corrélation entre les plans RGB ; mais avec un changement d'espace de couleur RGB vers un autre espace moins corrélé, par exemple YCbCr. Cette transformation permet une compression plus efficace, car l'information est principalement condensée dans le plan Y. Nous pouvons, donc, comprimer plus efficacement Cb et Cr.

Il est clair que les résultats obtenus après application de la transformation YCbCr en terme de taux de compression et de distorsion sont mieux que ceux obtenus par application directe de la compression sur une image RGB, Mais la question qui mérite d'être posée : est-il de même pour les autres transformations YUV, YIQ, O1O2O3,...

Dans ce contexte, l'objectif principal, du travail présenté dans ce mémoire, est l'étude de l'effet des transformées de décorrélation en compression avec perte des images couleurs RGB. Pour cela, nous avons proposé une méthode de compression similaire à la norme JPEG mais au lieu de changer l'espace RGB vers YCbCr dans l'étape transformation des couleurs, nous choisissons et testons plusieurs espaces de couleurs par exemple (YCbCr [15] [16], YIQ [13], YUV [15], YDbDr [20], YPbPr [13], O1O2O3 [17], HSV [12], HSL [20], XYZ [18], Lab [14],...). Une autre différence par rapport à JPEG est l'utilisation d'un nouveau quantificateur [16], et enfin l'utilisation du codage entropique appelé TRE (Two-Role Encoder) [32].

L'organisation générale du mémoire est décrite ci-dessous :

Le chapitre I est consacré à quelques concepts sur l'image, et les principales techniques de traitement d'image, et par la suite nous présenterons quelques applications de traitement d'image.

Le chapitre II liste les différents espaces de couleurs existants et les relations existantes entre eux, associé à une étude énergétique de chaque espace.

Le chapitre III consiste en un tour d'horizon portant sur les principales méthodes de compression d'images avec et sans perte ainsi que des notions générales telles que les mesures de performance et de distorsion.

Le chapitre IV présente la méthode de compression proposée tout en validant sa performance en effectuant une comparaison avec deux méthodes récentes [16] [17].

Le résumé du travail effectué ainsi que quelques perspectives envisageables sont donnés en conclusion générale.

Chapitre I

Généralités sur les images

I.1. Introduction

Ce chapitre, va présenter les notions de base de l'image et du traitement de l'information. Ces notions fondamentales et nécessaires sont passées en revue afin d'enchaîner d'une manière logique avec le chapitre concernant la compression.

On donnera quelques concepts de l'image et les principales techniques de traitement d'image, et par la suite nous présenterons quelques applications de traitement d'image.

I.2. Définition de l'image

Une image est une représentation visuelle voire mentale de quelque chose (objet, être vivant et/ou concept) par la peinture, le dessin, la photographie, le film, etc. [1] [5].

Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique continue, définie dans un domaine borné, où 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 [1].

I.3. Image numérique

Quand on parle d'image numérique, on parle de pixels. *Pixel est la contraction de l'expression anglaise "Picture éléments"*: éléments d'image. C'est tout simplement le plus petit élément d'une image, et on peut le figurer comme un carré élémentaire [1].

Chaque pixel est caractérisé par sa position qui peut être exprimée par deux coordonnées sur l'axe horizontal X et l'axe vertical Y et sa couleur qui représente sa valeur (*Pixel* (i, j)) (valeur unique, puisque le pixel est considéré comme indivisible) comme le montre la figure ci-dessous.

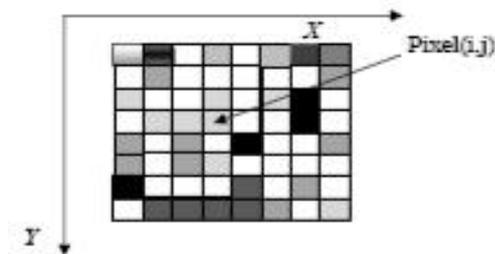


Figure I.1 : Élément d'une image (le pixel).

Donc, *Une image numérique* est une collection de pixels qui représentent chacun un point de l'image, codé par des valeurs numériques, ces valeurs peuvent être scalaires (images en niveaux de gris), ou bien vectorielles (images couleurs) [2].

Pour des raisons de commodité de représentation pour l'affichage et l'adressage, les images sont généralement rangées sous forme de tableau (noté I par exemple) de n lignes et p colonnes. Chaque élément $I(x, y)$ représente un pixel de l'image et sa valeur est associée à un niveau de gris codé sur m bits. Dans le cas d'une image couleur, on code celle-ci le plus souvent sur 3 couleurs : rouge, vert, bleu (8 bits pour chaque couleur).

La numérisation d'image peut se décomposer en 2 opérations principales :

a) L'échantillonnage (résolution spatiale) [2] : est le procédé de discrétisation spatiale d'une image, consistant à associer à chaque zone rectangulaire (pixel) $R(x, y)$ d'une image continue une unique valeur $I(x, y)$ (voir Figure I.2).

On parle de sous-échantillonnage lorsque l'image est déjà discrétisée et qu'on diminue le nombre d'échantillons.

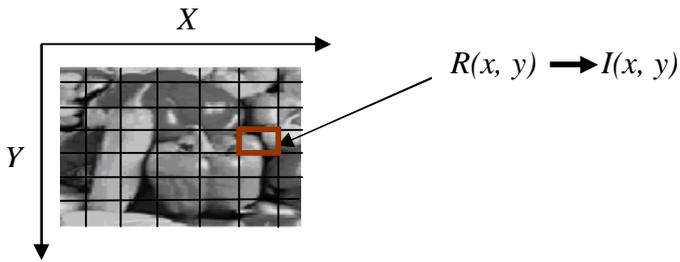


Figure I.2 : Echantillonnage d'une image.

Plus la taille des pixels est petite, plus l'image numérique est proche de l'image réelle. La figure I.3 montre un exemple d'une même image acquise à des résolutions différentes.

b) **La quantification (résolution tonale)** [2] : désigne la discrétisation tonale correspondant à la limitation du nombre de valeurs différentes que peut prendre chaque pixel.

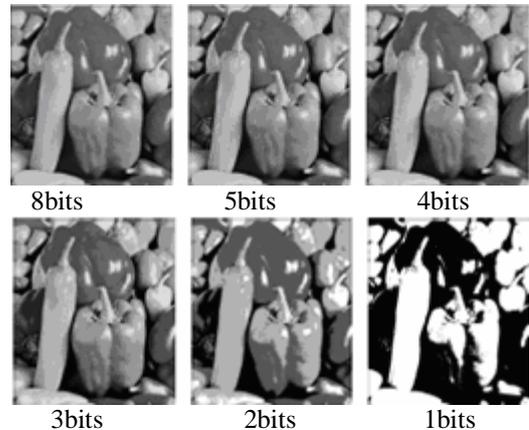
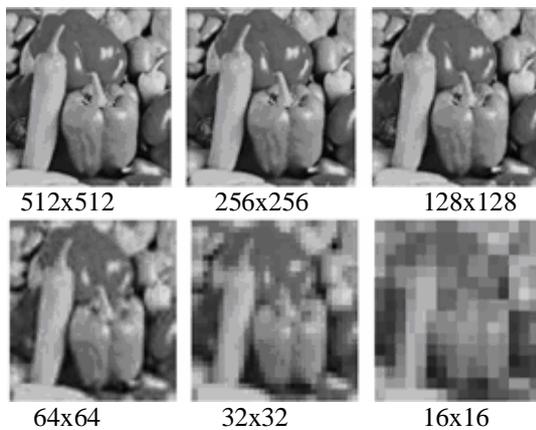


Figure I.3: Résolution spatiale: échantillonnage. **Figure I.4:** Résolution tonale: quantification.

Ce qui mène à dire que la qualité de l'image numérique dépend de :

- La qualité des périphériques de numérisation de l'image, du nombre de niveaux de gris ou de couleurs enregistrées, etc.
- La qualité de l'affichage à l'écran : définition de l'écran, nombre de teintes disponibles simultanément, calibrage de l'écran, etc.

I. 4. Les caractéristiques d'une image numérique

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

I.4.1. Dimension (définition) [1] : C'est la taille de l'image, ou le nombre de points (pixels) la composant. Cela correspond au nombre de pixels qui compose l'image en hauteur (axe vertical) et en largeur (axe horizontal): Une image possédant 640 pixels en largeur et 480 en hauteur aura une définition de 640 pixels par 480, notée 640x480.

I.4.2. Résolution : La résolution d'une image est définie par un nombre de pixels par unité de longueur classiquement en dpi (dots per inch) ou ppp (points par pouce) [3]. Ce paramètre est défini lors de la numérisation et dépend principalement des caractéristiques du matériel utilisé lors de la numérisation.

Augmenter la résolution peut entraîner des temps de visualisation et d'impression plus longs, et conduire à une taille trop importante du fichier contenant l'image.

I.4.3. Histogramme : Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse (Figure I.5) [2]. Par convention, un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite).

Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant [5] [6].

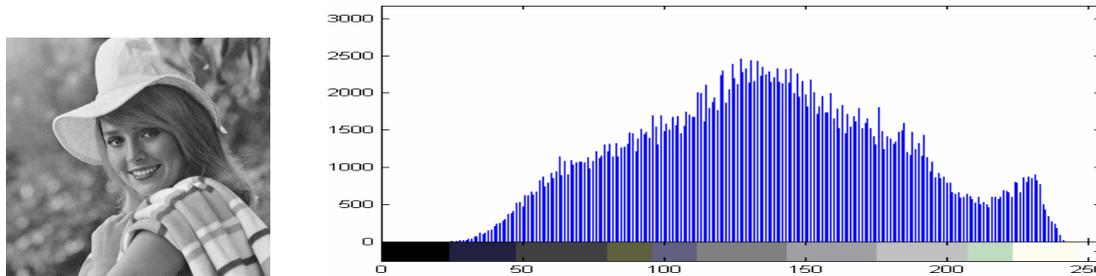


Figure I.5 : Image et histogramme associés.

I.4.4. Luminance : C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet [5].

I.4.5. Contraste : C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport [1] :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (I.1)$$

I.4.6. Profondeur : C'est le nombre de bits par pixel, cette valeur reflète le nombre de couleurs ou de niveaux de gris d'une image, par exemple :

- 32 bits/pixel = 1,07 milliards de couleurs
- 24 bits = 16,7 millions de couleurs
- 16 bits = 65 536 couleurs
- 8 bits = 256 couleurs

I.4.7. Le poids de l'image : Le poids d'une image se détermine en fonction de ces deux paramètres : dimensions, profondeur. Le poids de l'image est alors égal à sa dimension multipliée par sa profondeur. Par exemple, pour une image 640x480 en vraies couleurs (True colors) :

- Nombre de pixels (dimension) : $640 \times 480 = 307200$
- Poids de chaque pixel (profondeur) : 24 bits = 3 octets
- Le poids de l'image est ainsi égal à : $307200 \times 3 = 921600$ octets.

I.4.8. Contours et textures : Les contours représentent les frontières entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative [5]. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

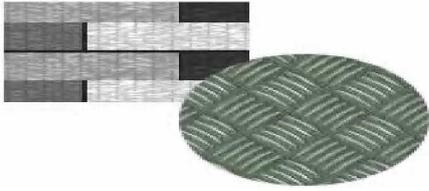


Figure 1.6: Image avec texture.

I.5. Les différents types d'images

Il existe différentes catégories d'image selon le nombre de bit sur lequel est codée la valeur de chaque pixel.

I.5.1. L'image monochrome (binaire) : C'est le plus simple type d'image où chaque pixel peut prendre uniquement la valeur noire ou blanche (chaque pixel est codé sur un seul bit) [5]. C'est typiquement le type d'image que l'on utilise pour scanner du texte quand celui ci est composé d'une seule couleur.

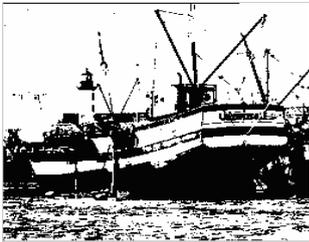


Figure I.7: Une image binaire.

I.5.2 L'image en niveaux de gris : Le niveau de gris est la valeur de l'intensité lumineuse à 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 à niveaux de gris (Figure I.8), on peut attribuer à chaque pixel de l'image une valeur correspondante à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet [4].



Figure I.8: Image en niveaux de gris.

I.5.3 L'image en couleurs : Même s'il est parfois utile de pouvoir représenter des images en noir et blanc ou en niveau de gris, les applications multimédias utilisent le plus souvent des images en couleurs [5] [2] (Figure I.9).

La représentation des couleurs s'effectue de la même manière que les images en niveaux de gris avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation.



Figure I.9: Image en couleurs.

Il existe plusieurs modes de représentation de la couleur, le plus utilisé pour les images numériques est le mode RGB [4]. Ce mode est basé sur la synthèse additive des couleurs, c'est à dire que le mélange des trois composantes (R, G, B) donne une couleur.

Il existe différents types d'images couleurs en fonction du nombre de bits utilisés pour le stockage de l'information couleur.

- *Image en "vraies couleurs" (True colors) (ou 24 bits)*

Il s'agit d'une appellation trompeuse car on est dans un monde numérique (discret, fini) qui ne peut pas rendre compte de la réalité (infinie) [4].

Chaque pixels peut prendre une valeur dans le RGB comprise entre 0 et 255 (soit $256 \times 256 \times 256$ possibilités= plus de 16 millions de possibilités).

L'information couleur de chaque pixel est donc codée par 3 octets, par conséquent les images en vraies couleurs sont très "lourdes" en termes de l'espace de stockage.

- *Image indexée*

Afin de diminuer la charge de travail nécessaire pour manipuler des images en 24 bits (et/ou gagner l'espace de stockage), on peut utiliser le mode de représentation en couleurs indexée. Dans ce cas on attache une palette de 256 couleurs (usuellement) à l'image.

Ces 256 couleurs sont choisies parmi les 16 millions de couleurs de la palette RGB. Pour chaque image le programme recherche les 256 couleurs les plus pertinentes.

Chaque pixel va non plus véhiculer le code couleur RGB qui lui est affecté, mais simplement un chiffre compris entre 0 et 255. A chacun de ces chiffres va correspondre une couleur, définie par son code RGB et stockée dans une palette (Look Up Table (LUT)) avec les données de l'image [4].

Lors de la visualisation de l'image, la correspondance se fait entre le numéro de la couleur affecté à chaque pixel (compris entre 0 et 255) et le code couleur RGB correspondant issu de la LUT.



Figure I.10: Exemple d'une image en 256 couleurs avec sa palette associée.

I.6. Les formats standards d'image

On distingue généralement deux grands types de formats :

I.6.1 Les images matricielles: Il s'agit d'images pixellisées, c'est-à-dire un ensemble de points (pixels) contenus dans un tableau, chacun de ces points possédant une ou plusieurs valeurs décrivant sa couleur [7].

I.6.1.1 Les principaux formats :

- BMP (Windows Bitmap) [7]: c'est le format actuel utilisé par Windows. Il produit des images de bonne qualité et est reconnu par de nombreuses applications. C'est le format le plus utilisé, par contre, il est extrêmement volumineux lorsqu'il utilise le codage en « true colors ».
- PCX (PiCture eXchange) [5]: le format défini par Paintbrush. Il accepte les modes de couleur, indexés, niveaux de gris et le noir et blanc.
- GIF (Graphic Interchange Format) [5] [7]: créé par CompuServe, utilise aussi le codage RGB, mais le format GIF n'utilise pas toutes les 16 millions de couleurs. Il prend les 256 couleurs les plus courantes pour réaliser l'image au format GIF. Cela permet une bonne compression et un affichage rapide de l'image.
- JPG ou JPEG (Joint Photographique Experts Group) [7] : créé par un consortium industriel, ce format très utilisé sur Internet, permet d'afficher les images en mode 16 millions de couleurs.

Il est conçu pour réduire le plus possible la taille des fichiers graphiques en acceptant éventuellement de légères pertes de qualité. Il est destiné à la transmission rapide d'information. Ces résultats de compression sont extraordinaires.

- TIFF (Tag Image File Format) [7]: c'est un format d'excellente qualité, mais qui présente des problèmes de compatibilité du fait d'une multiplicité de version. Il existe aussi une version compressée qui fournit des fichiers très compacts sans perte notable de qualité. Ce format est compatible avec d'autres plates formes (macintosh). Il est utilisé par les professionnels.

I.6.1.2. Avantages et inconvénients

On peut résumer les avantages et les inconvénients en:

- **Avantages**
 - ✓ Le mode de codage des images bitmap (24 bits, codage RGB) les rend adaptées au fonctionnement des principaux périphériques, notamment les contrôleurs d'écran "true colors"[7].
 - ✓ Simplicité de stockage en mémoire, puisqu'il suffit de coder la succession des valeurs de la matrice.
 - ✓ Grande facilité de traitement par des algorithmes élémentaires au niveau du pixel [7].
 - ✓ Adaptable aux images complexes [5].
- **Inconvénients**
 - ✓ La création d'une image à la main est difficile et il est conseillé de passer par un périphérique de numérisation : scanner, appareil photo-numérique... mais les retouches sont délicates [5].

- ✓ Les images bitmap sont "lourdes": les fichiers, lorsque l'on traite des images en haute définition, ont des tailles qui varient entre 10 et 30 Mo par image.... Elles sont donc encombrantes, difficiles à faire passer sur le réseau, etc [4] [5].
- ✓ Elles supportent mal les opérations de redimensionnement, réduction ou agrandissement [6].

1.6.2. Les images vectorielles : Les données sont représentées par des formes géométriques simples qui sont décrites d'un point de vue mathématique. Par exemple, un cercle est décrit par une information du type (cercle, position du centre, rayon) [6] [7]. Ces images sont essentiellement utilisées pour réaliser des schémas ou des plans.

Mais pour afficher à l'écran ou imprimer, ces images vectorielles sont en fait automatiquement traduites en images matricielles, car c'est le seul format directement affichable par ordinateur [5].

1.6.2.1. Les principaux formats :

- WMF (Windows Meta File) [6]: c'est le format vectoriel utilisé par Windows, il est reconnu par beaucoup de programmes fonctionnant sous windows. C'est un format qui permet de sauvegarder tant les images matricielles que les images vectorielles.
- EPS (Postscript / Encapsulated Postscript): il s'agit d'un format vectoriel très utilisé. C'est donc le format à privilégier lors des exports de vos dessins vectoriels. Les formats matriciels peuvent aussi être convertis au format EPS. Dans ce cas là, il ne s'agit que d'une encapsulation et non d'une conversion en image vectorielle.
- CGM (Computer Graphics Metafile): les fichiers graphiques CGM restent un moyen privilégié d'échange de dessins vectoriels entre applications, mais il ne traite pas les images matricielles [5].
- DXF (Data eXchange Format): format des produits Autocad et Autodesk.
- PCT (Pict): ce format vient du monde Macintosh, où il constitue un standard. Peut contenir aussi bien des données vectorielles que matricielles. Sur PC, l'ouverture d'une image Pict se révèle problématique avec de nombreux logiciels. Les images Pict peuvent être ouvertes avec Quick Time (Picture Viewer).

1.6.2.2. Avantages et inconvénients :

• Avantages

- ✓ Les fichiers sont légers [7].
- ✓ Les retouches sont aisées puisque les différents éléments de l'image sont indépendants et sélectionnables [7].
- ✓ Adaptable à chaque périphérique (car ajustable, coordonnées relatives, adaptable aux drivers intelligents, ...) [5].
- ✓ Adapté aux objets mathématiques et leur extension (voiture, avion,...) [5].

• Inconvénients

- ✓ Inutilisables pour des images complexes, des photographies [5].

- ✓ Non reconnues par les navigateurs Internet et par certains logiciels multimédia.
- ✓ Chaque format de fichier vectoriel possédant ses propres attributs, la compatibilité entre les formats est difficile [7].
- ✓ Un fichier vectoriel est plus fragile qu'un fichier matriciel dont l'en-tête surtout doit être intacte. La moindre dégradation de l'information est souvent irréparable [7].

I.7. Généralités sur les traitements d'image

I.7.1. Définition

On désigne par traitement d'images l'ensemble des méthodes et techniques opérant sur l'image, dont le but est d'améliorer son aspect visuel ou d'en extraire des informations jugées pertinentes. Il se définit comme un ensemble de tâches destinées à extraire de l'image des informations qualitatives et quantitatives [8].

I.7.2 Les domaines d'applications

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que :

- **Imagerie aérienne et spatiale:** dans laquelle les traitements concernent l'étude des images satellites, l'analyse des ressources terrestres, la cartographie automatique, les analyses météorologiques [5] [8].
- **Technologies biomédicales:** on trouve des utilisations de cette technique dans l'échographie, la résonance magnétique nucléaire, ainsi que dans le domaine de la reconnaissance automatique des cellules ou de chromosomes [8].
- **La robotique:** qui connaît actuellement le plus grand développement et dont les tâches usant de l'imagerie sont principalement l'assemblage (pièce mécanique, composants électroniques,...), le contrôle de qualité, ainsi que la robotique mobile [5] [6].
- **la télésurveillance :** (exemple, radar automatique : recherche en temps réel d'un véhicule par reconnaissance de son immatriculation parmi un flot de véhicules circulant sur le boulevard périphérique par caméra fixe [8]).
- **Simulation et contrôle de processus :** on trouve des utilisations de cette technique dans les cours de pilotage et le contrôle des panneaux [5].

Aussi :

L'astronomie, la chimie, la physique nucléaire (identification de trajectoires de particules), l'armement (guidage de missiles, reconnaissance des formes), métiers du spectacle, les télécommunications (TV, radio, vidéo, publicité,...), l'architecture, l'imprimerie,...

I.7.3. Principales techniques

La panoplie des techniques de traitement d'images est étendue et variée, On peut citer les techniques suivantes :

- **Acquisition** [8] [9] : pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation (échantillonnage, quantification). On utilise plus couramment des caméras vidéo, des appareils photos numériques. En médecine, on

utilise des imageurs IRM, TEP, scanner X, écho doppler, échographie, scintigraphie etc.

➤ **Filtrage** [6] [10] : est une opération qui consiste à réduire le bruit contenu dans une image au moyen d'algorithmes provenant des mathématiques par l'utilisation de méthodes d'interpolation ou de la morphologie mathématique. Il y a plusieurs types de filtrage nous distinguons :

❖ Filtre passe-bas (lissage) : ce filtre n'affecte pas les composantes de basse fréquence dans les données d'une image, mais doit atténuer les composantes de haute fréquence. L'opération de lissage est souvent utilisée pour atténuer le bruit et les irrégularités de l'image.

❖ Filtre passe-haut (accentuation) : le renforcement des contours et leur extraction s'obtiennent dans le domaine fréquentiel par l'application d'un filtre passe-haut. Le filtre digital passe-haut a des caractéristiques inverses du filtre passe-bas. Ce filtre n'affecte pas les composantes de haute fréquence d'un signal, mais doit atténuer les composantes de basse fréquence.

❖ Filtre passe-bande (différentiation) : cette opération est une dérivée du filtre passe-bas. Elle consiste à éliminer la redondance d'information entre l'image originale et l'image obtenue par filtrage passe-bas. Seule la différence entre l'image source et l'image traitée est conservée.

❖ Filtre directionnel : dans certains cas, on cherche à faire apparaître des détails de l'image dans une direction bien déterminée. Pour cela, on utilise des filtres qui opèrent suivant des directions (horizontales, verticales et diagonales).

➤ **Segmentation** : permet d'isoler dans l'image les objets sur lesquels doit porter l'analyse, de séparer, dans une image, les régions d'intérêt du fond. Il existe plusieurs techniques, la plus simple étant le seuillage. La segmentation peut être manuelle ou automatique [10].

➤ **Convolution** : est le remplacement de la valeur d'un pixel par une combinaison linéaire de ses voisins. Elle consiste à faire balayer une fenêtre (masque) sur l'ensemble des points de l'image [8].

I.7.4. Principales applications

Le traitement numérique d'image est un domaine particulièrement vaste, les applications sont nombreuses et variées : gestion de bases de données images, analyse et compréhension d'images, reconnaissance de l'écriture, compression d'images fixes ou animées en vue de leur stockage ou leur transmission (et c'est l'objet de notre étude que l'on détaillera dans les chapitres qui suivent), etc.

La compression d'images [1] [5] : consiste à trouver une représentation numérique de taille inférieure à celle de l'image initiale sans trop de détérioration. La compression d'images peut s'employer avec des contraintes et des attentes très différentes, selon l'utilisation à laquelle les images sont destinées. On peut vouloir réduire le nombre de bits d'une image avec une contrainte sur la capacité de stockage, la vitesse de transmission, etc.

I.8. Conclusion

La manipulation des images pose des problèmes beaucoup plus complexes que celle du texte. En effet, l'image est un objet à deux dimensions, censé représenter un espace à trois dimensions, ce qui a deux conséquences majeures :

- Le volume de données à traiter est beaucoup plus important.

- La structure de ces données est nettement plus complexe.

Il en résulte que la manipulation, le stockage et la représentation de ces données se heurtent à certaines limitations.

Grâce au traitement d'image, ces contraintes sont levées ou contournées. En effet, ce domaine cherche à détecter la présence de certaines formes, certains contours ou certaines textures d'un modèle connu, c'est le problème de la détection.

Un autre aspect de traitement d'image concerne l'analyse et la compréhension de l'image dans le but d'en extraire des informations utiles. Il cherche aussi parfois à comprimer l'image, afin de gagner en vitesse lors de la transmission de l'information, et en capacité de stockage, tous en dégradant le moins possible les images considérées.

C'est le domaine du codage ou compression des images, objet de notre étude, que l'on détaillera dans les chapitres qui suivent.

Chapitre II

Transformation d'espaces de couleurs et étude énergétique

II.1. Introduction

Dans ce chapitre, nous allons dans un premier temps, lister les différents espaces de couleurs existants et les relations existantes entre-eux. Par la suite, on présentera une étude énergétique de chaque espace afin de trouver le meilleur espace de couleurs adapté à la compression des images.

II.2. Définition de l'espace de couleurs

Un **espace colorimétrique** ou *espace de couleurs* associe des nombres aux couleurs visibles. Compte tenu des limites de la vision humaine, ces nombres se présentent généralement sous la forme de triplets [12]. Chaque couleur de lumière peut donc être caractérisée par un point dans un espace à trois dimensions.

Il y a plusieurs espaces de couleurs qui sont utilisés dans différents domaines. L'espace CIE XYZ sert en photométrie, CIE LUV en visualisation scientifique, CIELAB dans les textiles, RGB pour les moniteurs, CMY pour l'impression, YIQ pour la télévision, HSV, HSI et HLS sont utilisés pour la sélection de couleurs, Munsell en psychologie et Ostwald en peinture [13]. Il existe d'autres espaces encore, mais ceux-ci sont les principaux. Par conséquent, certains offrent une palette de couleurs très réduite (l'espace RAL par exemple utilisé en peinture ne dispose que d'environ 1900 couleurs), alors que les plus complets présentent plusieurs millions de couleurs [14]. Dans la suite on exposera, d'une manière aussi claire que possible, la distribution énergétique des principaux espaces de couleurs.

II.3. Les espace CIE

II.3.1. L'espace RGB

a) Définition :

L'espace RGB (Red, Green, Blue, pour Rouge Vert Bleu, en français RVB), mis au point en 1931 par la Commission Internationale de l'Eclairage (CIE) consiste à représenter l'espace des couleurs à partir de trois rayonnements monochromatiques de couleurs (rouge, vert et bleu) [11][12].

Cet espace de couleurs correspond à la façon dont les couleurs sont généralement codées informatiquement, ou plus exactement à la manière dont les tubes cathodiques des écrans d'ordinateurs représentent les couleurs [12].

Le RGB est un espace de couleur additif (Figure II.1) -c-à-d, ses trois couleurs primaires (rouge, vert, et bleu) se combinent d'une manière additive pour produire toutes les couleurs désirées [14]. Chaque couleur est représentée par un groupe de trois valeurs : une pour le rouge, une pour le vert et une pour le bleu. Chacune de ces valeurs varie dans l'intervalle [0 – 255]. Si ces trois valeurs sont égales à 0, la couleur correspondante est le noir, et le blanc si les trois valeurs sont égales à 255. Lorsque toutes les composantes ont une valeur égale, on obtient une nuance de gris neutre. Si vous vous représentez ce modèle comme des projecteurs, il est facile de comprendre que des couleurs faibles (valeurs basses) signifie couleurs plus sombres, et couleurs fortes (valeurs élevées) signifie couleurs plus brillantes [14].

RGB peut représenter plus de 16 millions de couleurs (Ce qu'on appelle souvent couleurs vraies (True colors)). Ceci concerne les images dont la profondeur est de 24 bits par pixel (8 bits pour chaque composante : 256 intensités). Toutefois, cette valeur n'est que théorique car elle dépend fortement du matériel d'affichage utilisé.

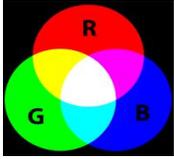


Figure II.1 : Composition additive des couleurs.»

Le codage RGB repose sur trois composantes proposant la même gamme de valeurs, on le représente généralement graphiquement par un cube dont chacun des axes correspond à une couleur primaire [12] (Figure II.2).

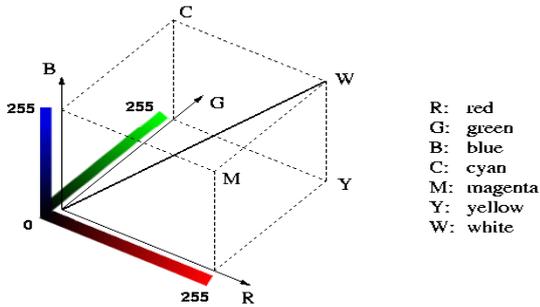


Figure II. 2 : Présentation des couleurs dans l'espace RGB.

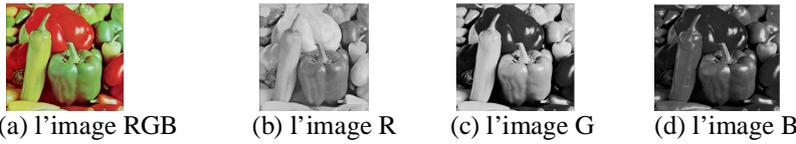


Figure II.3 : Une image RGB, avec ses composantes séparées R, G et B.

II.3.2.L'espace CIE XYZ

a) Définition :

Pour certaines valeurs de λ (longueur d'onde) la valeur R du RGB est négative et donc certaines couleurs ne sont pas représentées. L'espace RGB ne représente donc pas tout le spectre visible, et c'est pour remédier à cela que l'espace CIE XYZ a été développé [18]. Ce dernier caractérise une couleur en fonction de deux composantes X et Z perpendiculaires représentant la chromacité (tonalité et saturation), et d'une composante Y perpendiculaire aux autres axes qui représente la luminance (clarté), l'ensemble définissant une couleur [18].



Figure II.4 : Une image RGB, avec ses composantes X , Y et Z .

a) Transformations

- *Transformation RGB* ► *XYZ* [20]

Le passage de l'espace RGB vers l'espace XYZ se fait grâce à une matrice 3x3 inversible.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} 0.618 & 0.177 & 0.205 \\ 0.299 & 0.587 & 0.114 \\ 0 & 0.056 & 0.944 \end{bmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (\text{II.1})$$

- *Transformation XYZ ► RGB* [20]

Le passage de l'espace XYZ vers l'espace RGB est donné par la relation suivante :

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{bmatrix} 1.876 & -0.533 & -0.343 \\ -0.967 & 1.998 & -0.031 \\ 0.057 & -0.118 & 1.061 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (\text{II.2})$$

II.4. Les espaces luminance-chrominance

Cette famille des espaces couleurs dissocie l'information chromatique des couleurs de leur intensité lumineuse. En effet ce type d'espaces décrits un stimulus couleur selon une composante de luminance, et deux autres composantes allouées à la quantification de l'aspect chromatique du stimulus [11]. Les composantes de ces espaces sont calculées à partir des composantes trichromatiques RGB. On distingue différents types d'espaces luminance-chrominance:

II.4. 1. L'espace YUV

a) Définition :

Le modèle YUV définit un espace colorimétrique en trois composantes. Le paramètre Y représente la luminance (c'est-à-dire l'information en noir et blanc), tandis que U et V permettent de représenter la chrominance, c'est-à-dire l'information sur la couleur (Figure II.5). Ce modèle a été mis au point afin de permettre de transmettre des informations colorées aux téléviseurs couleurs, tout en s'assurant que les téléviseurs noir et blanc existant continuent d'afficher une image en tons de gris [11].

YUV est utilisé dans les systèmes de diffusion télévisuelle PAL (Phase Alternation Line), SECAM (Séquentiel Couleur avec Mémoire), et NTSC [15].

Le signal YUV est créé depuis une source RGB (rouge, vert et bleu). Les valeurs de R , G et B sont additionnées selon leur poids relatifs pour obtenir le signal Y . Ce dernier représente la luminance de la source. Le signal U est obtenu en soustrayant le Y du signal bleu d'origine ; de façon similaire le V est obtenu en soustrayant Y du signal rouge.

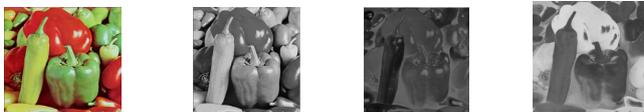


Figure II.5: Une image RGB, avec ses composantes Y , U , et V .

b) Transformations :

Le passage de l'espace RGB à l'espace YUV est défini par l'équation suivante [15] :

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{II.3})$$

Les équations de passage des composantes YUV aux composantes RGB s'obtiennent facilement par l'équation ci-dessous [15] :

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.140 \\ 1 & -0.395 & -0.581 \\ 1 & 2.032 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad (\text{II.4})$$

Où $(R, G, B, Y) \in [0 ; 1]^4$, $U \in [-0,436 ; 0,436]$ et $V \in [-0,615 ; 0,615]$

II.4. 2. L'espace YCbCr

a) Définition :

L'espace YCbCr défini par le IRCC (International Radio Consultative Committee), est souvent Utilisé dans la compression des images. Cet espace se compose de Y qui est la composante de luminance (appelée aussi Luma), ainsi que C_b et C_r qui sont les composantes de chrominance (bleue et rouge) voir la figure II.6 [15] [20].

Il est largement appliqué dans la vidéo et la compression d'images telles que MPEG et JPEG.

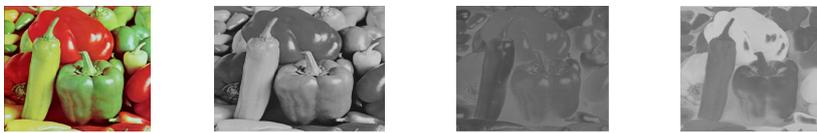


Figure II.6: Une image RGB, avec ses composantes Y , C_b , et C_r .

b) Transformations :

- Transformation $RGB \rightarrow YCbCr$ [16]

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.1687 & -0.33126 & -0.50 \\ 0.5000 & -0.41869 & 0.08131 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{II.5})$$

- Transformation $YCbCr \rightarrow RGB$ [16]

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34413 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} \quad (\text{II.6})$$

II.4. 3. L'espace YIQ

a) Définition :

L'espace YIQ est adopté en premier lieu par le NTSC (National Television System Committee) pour assurer une diffusion efficace pour les téléviseurs couleurs en garantissant en même temps la compatibilité avec les téléviseurs noir et blanc. La composante Y capture l'intensité lumineuse ; I et Q correspondent à la chrominance [13].



Figure II.7 : Une image RGB, avec ses composantes Y , I , et Q .

b) Transformations :

- Transformation RGB ► YIQ [20]

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{II.7})$$

- Transformation YIQ ► RGB [20]

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.621 \\ 1 & -0.272 & -0.647 \\ 1 & -1.106 & 1.703 \end{bmatrix} \times \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad (\text{II.8})$$

II.4. 4. L'espace YDbDr

a) Définition :

YDbDr est l'espace colorimétrique utilisé dans la norme de télévision couleur SECAM. Y est la luminance, Db et Dr sont les différences de couleurs bleue et rouge (Figure II.8) [13]. Il est très proche de l'espace YUV. Le passage entre DbDr et UV se fait grâce aux relations suivantes :

$$Db = + 3.059U, Dr = - 2.169V.$$

Une variante de la norme PAL, appelée PAL-N utilise elle aussi l'espace colorimétrique YDbDr.



Figure II.8 : Une image RGB, avec ses composantes Y , Db , et Dr .

b) Transformations :

- Transformation RGB ► YDbDr [13]

Le passage entre RGB et YDbDr se fait grâce aux relations suivantes :

$$\begin{bmatrix} Y \\ D_b \\ D_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.450 & -0.883 & 1.333 \\ -1.333 & 1.116 & 0.217 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{II.9})$$

- Transformation YDbDr ► RGB [13]

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.00009230316148 & -0.52591260661865 \\ 1 & -0.12913280890509 & 0.26789938207599 \\ 1 & 0.66467909978955 & -0.00007920543533 \end{bmatrix} \times \begin{bmatrix} Y \\ D_b \\ D_r \end{bmatrix} \quad (\text{II.10})$$

Où $R, G, B, Y \in [0,1]$ et $Db, Dr \in [-1.33, 1.33]$

II.4. 5. L'espace YPbPr

a) Définition :

L'espace YPbPr est utilisé dans l'électronique visuelle, il est la version analogue de l'espace de couleurs de YCbCr; les deux sont numériquement équivalents, mais YPbPr est conçu pour l'usage dans les systèmes analogues, tandis que YCbCr est prévu pour la vidéo numérique [13].

Y est la luminance (luma), Pb est la différence entre le bleu et la luminance, et Pr est la différence entre le rouge et la luminance.

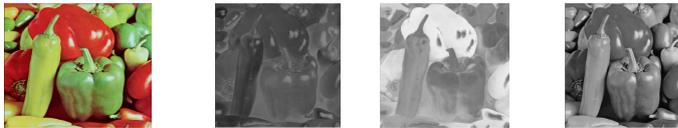


Figure II.9 : Une image RGB, avec ses composantes Y , Pb , et Pr .

b) Transformations :

- Transformation RGB ► YPbPr [13]

$$\begin{aligned} Y &= (r+g+b)/3 \\ Pb &= b - Y \\ Pr &= r - Y \end{aligned} \quad (\text{II.11})$$

- Transformation YPbPr ► RGB [13]

$$\begin{aligned} R &= Y+Pr \\ B &= Y+Pb \\ G &= Y-Pr-Pb \end{aligned} \quad (\text{II.12})$$

Où R , G , B sont dans l'intervalle $[0,1]$, Y est dans l'intervalle $[0,1]$, Pb et Pr sont dans l'intervalle $[-0.5, 0.5]$.

II.4. 6. L'espace O1O2O3

a) Définition :

L'espace O1O2O3 est un espace de couleurs qui décompose la couleur en trois composantes, $O1$ c'est l'intensité ou la luminance, tandis que $O2$ et $O3$, ensemble, représentent la chrominance à chaque pixel (Figure II.10) [17].



Figure II.10: une image RGB, avec ses composantes $O1$, $O2$ et $O3$.

b) Transformations :

- Transformation RGB ► O1O2O3 [17]

$$O_1 = \left\lfloor \frac{R+G+B}{3} + 0.5 \right\rfloor$$

$$O_2 = \left\lfloor \frac{R-B}{2} + 0.5 \right\rfloor$$

$$O_3 = B - 2G + R$$

- Transformation O1O2O3 ► RGB [17]

$$B = O_1 - O_2 + \left\lfloor \frac{O_3}{2} + 0.5 \right\rfloor - \left\lfloor \frac{O_3}{3} + 0.5 \right\rfloor$$

$$G = O_1 - \left\lfloor \frac{O_3}{3} + 0.5 \right\rfloor$$

$$R = O_1 + O_2 + O_3 - \left\lfloor \frac{O_3}{2} + 0.5 \right\rfloor - \left\lfloor \frac{O_3}{3} + 0.5 \right\rfloor$$

II.5. Les espaces répondant à la perception humaine

Cette famille de systèmes couleurs décrit la couleur selon la perception humaine de la couleur. Elle regroupe deux catégories de systèmes [11]:

- Les systèmes uniformes au sens de la perception visuelle.
- Les systèmes de type Luminance, Teinte et Saturation.

II.5.1. Les systèmes uniformes

Ces systèmes peuvent décrire fidèlement, au sens de la perception visuelle, les écarts couleurs proches. Ils possèdent en effet une métrique qui reflète au mieux les distances couleurs selon la vision humaine des couleurs. Parmi ces systèmes, on peut citer :

II.5.1.1. L'espace Lab :

a) Définitions :

CIE Lab (plus précisément $L^*a^*b^*$) est un modèle de représentation de couleurs développé en 1976 par la Commission internationale de l'éclairage (CIE) [20]. Comme tous les systèmes issus du système CIE XYZ, il caractérise une couleur à l'aide d'un paramètre d'intensité correspondant à la luminance et de deux paramètres de chrominance qui décrivent la couleur (figure II.15). Cet espace a aussi été construit pour palier au problème de non cohérence du système RGB avec la perception logarithmique humaine de la couleur, en effet dans l'espace Lab deux couleurs proches pour l'oeil le seront aussi en terme de distance euclidienne [14].

Il possède le grand avantage d'être uniforme. Il est très utile dans le cas de mélanges de pigments, par exemple, pour l'industrie graphique ou du textile [11].

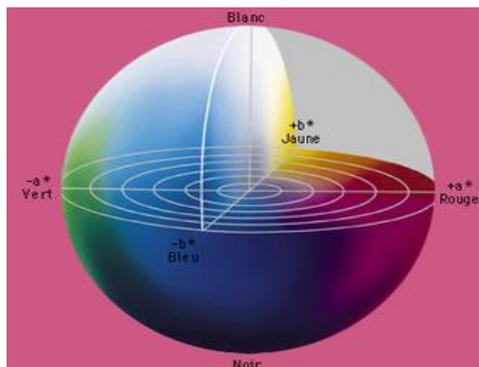


Figure II.11 : L'espace Lab.

Une des difficultés majeures de ce système est qu'il utilise un système mixte de repérage des points de couleurs. La saturation est mesurée de manière cartésienne, alors que la teinte et la luminosité sont mesurées de manière angulaire.

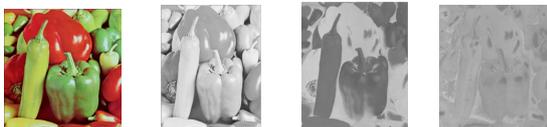


Figure II.12 : Une image RGB, avec ses composantes L, a et b.

b) Transformations

Ici, les moyens de passer d'un espace à un autre ne sont plus de simples matrices 3x3 mais des algorithmes. De plus, on est obligé d'effectuer la transformation RGB ► XYZ avant de passer en CIE-Lab. En effet le système Lab n'est pas directement relié au système RGB.

- *Transformation XYZ-Lab* [11] [20]:

$$L=116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}}-16 \quad \text{pour } \frac{Y}{Y_n} > 0.008856$$

$$L=903.3-\frac{Y}{Y_n} \quad \text{pour } \frac{Y}{Y_n} \leq 0.008856$$

$$a=500\left(f\left(\frac{X}{X_n}\right)-f\left(\frac{Y}{Y_n}\right)\right)$$

$$b=200\left(f\left(\frac{Y}{Y_n}\right)-f\left(\frac{Z}{Z_n}\right)\right)$$

$$\text{Où : pour } t > 0.008856 \quad f(t) = \sqrt[3]{t}$$

$$\text{pour } t \leq 0.008856 \quad f(t) = 7.787t + \frac{16}{116}$$

X_n , Y_n et Z_n correspondent au blanc décrit dans l'espace XYZ. On les obtient pour RGB = (255, 255, 255).

- *Transformation Lab-XYZ*:

$$Y=Y_n \cdot \frac{L}{903.3} \quad \text{pour } L \leq 8$$

$$Y=Y_n \left(\frac{L+16}{116}\right)^3 \quad \text{pour } L > 8$$

$$X=Z_n \cdot f\left(\frac{a}{500} + \frac{L+16}{116}\right)$$

$$Z=Z_n \cdot f\left(\frac{L+16}{116} - \frac{b}{200}\right)$$

$$\text{Où : pour } t > 0.207 \quad f(t) = t^3$$

$$\text{Pour } t \leq 0.207 \quad f(t) = \frac{116t-16}{903.3}$$

II.5.2. Les systèmes de type Luminance, Teinte et Saturation

Il paraît difficile selon la perception humaine des couleurs d'évaluer une couleur selon des coordonnées RGB abstraites. Il convient d'utiliser à la place des notions subjectives telles que la teinte, la luminance et la saturation [11]. En effet, ces notions sont faciles à interpréter et sont liées à nos habitudes quotidiennes pour différencier les couleurs entre elles. Plusieurs systèmes quantifiant ces notions sont proposés dans la littérature. Ces systèmes fournissent une description des images numériques couleurs qui sont facile à assimiler. D'ailleurs les ordinateurs adoptent ce type de système pour communiquer avec l'homme lorsqu'il s'agit de décrire une couleur [11].

Les systèmes de type Luminance-Teinte-Saturation « LTS » sont basés sur un partitionnement uniforme de l'espace couleur selon un pas de discrétisation arbitraire, et qui n'est d'ailleurs pas relié à des grandeurs physiologiques de la vision humaine [11]. Cependant, l'objectif de ces systèmes est de classer les couleurs selon un principe basé sur la perception psycho-visuelle. Ils sont surtout considérés comme des représentations descriptives de la couleur.

La famille des systèmes LTS est mentionnée en littérature sous différentes nominations, citons à titre d'exemple :

- HSV pour Hue, Saturation and Value,
- ISH pour Intensity, Saturation and Hue,
- ITS pour Intensité, Teinte et Saturation,

- LCH pour Luminance, Chroma and Hue,
- HSL pour Hue, Saturation and Luminance,
- HSB pour Hue, Saturation and Brightness.

II.5.2.1 L'espace HSL

a) Définition :

L'espace HSL (Hue, Saturation, Luminance, ou en français TSL), s'appuyant sur les travaux du peintre Albert H. Munsell (qui créa l'Atlas de Munsell), est un modèle de représentation dit "naturel", c'est-à-dire proche de la perception physiologique de la couleur par l'oeil humain, il consiste à décomposer la couleur en trois variables [20] :

- La teinte (**Hue**) permet de déterminer la couleur souhaitée (rouge, vert, jaune, ...).
- La saturation, également appelée intensité, mesure la pureté des couleurs. Elle permet de distinguer les couleurs "vives" des couleurs "pastels" ou "délavées".
- La luminance, également appelée brillance, indiquant la quantité de lumière de la couleur, c'est-à-dire son aspect clair ou sombre.

Certes le mode RGB est très adapté pour le codage numérique de la couleur ainsi que sa manipulation par les écrans d'ordinateurs, mais il est trop limité quant à la sélection de couleurs d'une manière plus adéquate ainsi qu'en termes de retouche d'image [12]. Ainsi le modèle HSL a-t-il été mis au point afin de pallier cette lacune du modèle RGB. HSL permet de sélectionner plus facilement et plus efficacement la couleur recherchée et donne des possibilités de retouches plus originales. Il est utilisé par de nombreux programmes de traitement d'image [11] [14].

Voici une représentation graphique du modèle HSL (Figure II.13), dans lequel la teinte est représentée par un cercle chromatique et la luminance et la saturation par deux axes :

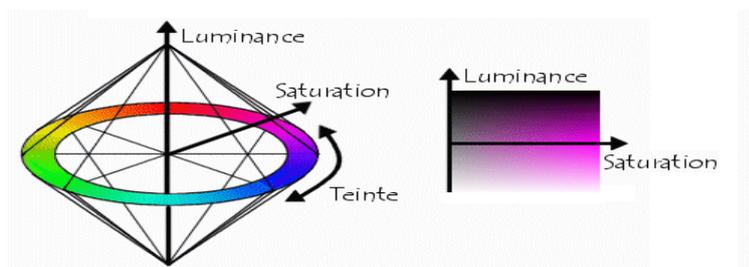


Figure II.13 : Présentation des couleurs dans l'espace HSL.



Figure II.14 : Une image avec ses composantes H, S, et L.

b) Transformations [11]:

- *Passage de RGB à HSL*

$$H = \begin{cases} 0, \text{if } \max = \min \\ (60^\circ \times \frac{G-B}{\max-\min} + 360^\circ) \bmod 360^\circ, \text{if } \max = R \\ 60^\circ \times \frac{B-R}{\max-\min} + 120^\circ, \text{if } \max = G \\ 60^\circ \times \frac{B-G}{\max-\min} + 240^\circ, \text{if } \max = B \end{cases}$$

$$S = \begin{cases} 0, \text{if } \max = \min \\ \frac{\max - \min}{2L}, \text{if } L \leq \frac{1}{2} \\ \frac{\max - \min}{2 - 2L}, \text{if } L > \frac{1}{2} \end{cases}$$

$$L = \frac{1}{2}(\max + \min)$$

Où $S, L \in [0,1]$, et $H \in [0, 360]$

- *Passage de HSL à RGB*

$$q = \begin{cases} L \times (1 + S), & \text{if } L < \frac{1}{2} \\ L + S - (L \times S), & \text{if } L \geq \frac{1}{2} \end{cases}, \quad p = 2 \times L - q, \quad h_k = \frac{H}{360} \quad (\text{H normalisé pour être dans la l'intervalle } [0,1])$$

$$T_R = h_k + \frac{1}{3}, \quad T_G = h_k, \quad T_B = h_k - \frac{1}{3}$$

$$\text{if } T_C < 0 \quad T_C = T_{C+1.0} \quad \text{pour chaque } C \in \{R, G, B\}$$

$$\text{if } T_C > 1 \quad T_C = T_{C-1.0} \quad \text{pour chaque } C \in \{R, G, B\}$$

Calculer pour chaque composant $C \in \{R, G, B\}$:

$$C = \begin{cases} p + ((q - p) \times 6 \times T_c), \text{If } T < \frac{1}{6} \\ q, \text{If } \frac{1}{6} \leq T < \frac{1}{2} \\ p + ((q - p) \times 6 \times (\frac{2}{3} - T_c)), \text{If } \frac{1}{2} \leq T < \frac{2}{3} \\ p, \text{Autrement} \end{cases}$$

II.5.2.2 L'espace HSV

a) Définition :

L'espace HSV est un modèle de représentation dit "naturel", c'est-à-dire proche de la perception physiologique de la couleur par l'oeil humain. Il consiste à décomposer la couleur selon des critères physiologiques [11] [20] :

- La teinte (en anglais Hue), correspondant à la perception de la couleur.
- La saturation, décrivant la pureté de la couleur, c'est-à-dire son caractère est vif ou terne.

- La luminance (en anglais value), indiquant la quantité de lumière de la couleur, c'est-à-dire son aspect clair ou sombre.

Il existe deux modèles de représentation de cet espace : hexagonal et double-hexagonal comme représenté dans la figure ci-dessous.

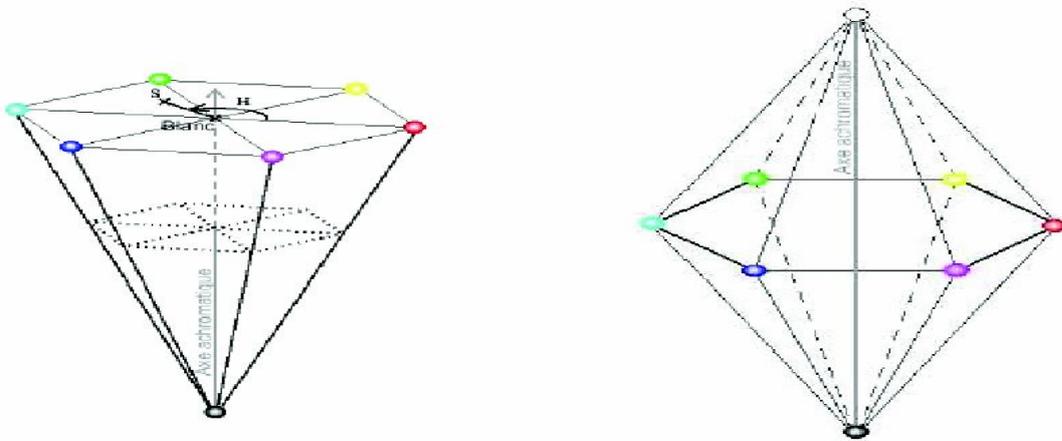


Figure II.15 : Présentation des couleurs dans l'espace HSV.

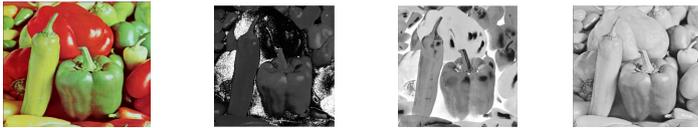


Figure II.16 : Une image RGB, avec ses composantes H, S, et V.

c) Transformations :

- Passage de RGB à HSV [20]

$$H = \begin{cases} 0, & \text{if } \max = \min \\ (60^\circ \times \frac{G - B}{\max - \min} + 360^\circ) \bmod 360^\circ, & \text{if } \max = R \\ 60^\circ \times \frac{B - R}{\max - \min} + 120^\circ, & \text{if } \max = G \\ 60^\circ \times \frac{B - G}{\max - \min} + 240^\circ, & \text{if } \max = B \end{cases}$$

$$S = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max}, & \text{Autrement} \end{cases}$$

$$V = \max$$

Avec : $H \in [0 - 360^\circ]$, S et $V \in [0 - 1]$, $R, G, B \in [0 - 1]$

- Passage de HSV à RGB [20]

$$H_i = \left\lfloor \frac{H}{60} \right\rfloor \bmod 6 \quad f = \frac{H}{60} - \left\lfloor \frac{H}{60} \right\rfloor \quad p = V \times (1 - S), \quad q = V \times (1 - f \times S), \quad t = V \times (1 - (1 - f) \times S)$$

Calculer le vecteur de couleurs (r, g, b) :

$$(r, g, b) = \begin{cases} (v, t, p), & \text{if } H_i = 0 \\ (q, v, p), & \text{if } H_i = 1 \\ (p, v, t), & \text{if } H_i = 2 \\ (p, q, v), & \text{if } H_i = 3 \\ (t, p, v), & \text{if } H_i = 4 \\ (u, p, q), & \text{if } H_i = 5 \end{cases}$$

II.6. Les autres espaces de représentation

En plus des familles standards de représentation de la couleur nous que venons de voir, il existe d'autres représentations qui sont le plus souvent dédiées à un type d'application. C'est le cas, par exemple, de l'espace CMY qui a été spécialement créé pour les besoins d'impression couleur.

II.6.1. L'espace de Couleur CMY

a) Définitions :

L'espace Cyan, Magenta, Yellow (CMY) est un espace de couleurs de base soustractive [14]. Cette dernière est dite soustractive car l'addition des couleurs de base donne du noir. C'est pour cette raison que cet espace est surtout utilisé en imprimerie couleur. En effet avec cet espace on peut obtenir du noir avec les autres couleurs, ce qui n'était pas le cas avec l'espace RGB.

L'espace CMY est donc le complémentaire de l'espace RGB (voir Figure II.18).

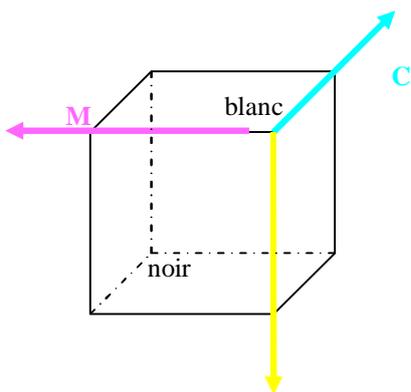


Figure II.17 : Cube CMY.



Figure II.18 : Une image avec ses composantes C, M et Y.

b) Transformations

Les relations liant l'espace RBG et l'espace CMY sont les suivantes :

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{II.13})$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad (\text{II.14})$$

II.8. Etude énergétique :

Pour mesurer l'énergie dans chaque composante d'espace de couleurs notée $T1$, $T2$ et $T3$, premièrement on calcule l'énergie totale (ET) de l'espace de couleurs T1T2T3 par l'expression suivante :

$$ET = \sum_{k=1}^3 \sum_{i=1}^M \sum_{j=1}^N x_k(i, j)^2 \quad (\text{II.15})$$

où (M, N) représente la dimension de l'image.

Ensuite l'énergie relative, dans chaque composante ($T1$, $T2$ ou $T3$), est calculée par la formule ci-dessous :

$$RET_i = 100 \times \frac{ET_i}{ET} \quad (\text{II.16})$$

Avec $i \in \{1, 2, 3\}$ et $ET_i = \sum_{k=1}^M \sum_{j=1}^N T_i(k, j)^2$

Il est impératif de signaler que l'objectif de cette étude est de trouver l'espace de couleurs le plus adapté à la compression des images, c-à-d celui qui donne un taux de compression le plus grand (possible) avec une représentation visuelle aussi fidèle et faisable de l'image originale.

Les tableaux ci dessous montrent les résultats collectés après changement d'espaces de couleurs appliqué sur dix différentes images couleurs RGB.

Image	RE _R (%)	E _G (%)	RE _B (%)
Peppers	49.11	38.11	12.81
Lena	58.37	21.09	20.53
Airplane	31.90	32.66	35.43
Baboon	38.22	32.87	28.90
Elephant	42.17	38.91	18.91
House	33.35	30.90	35.74
Jerusalem	30.12	34.76	35.11
Monolake	18.31	34.73	46.94
Pelican	32.44	32.66	34.89
Athens	36.91	31.04	32.04
Moyenne	37.09	32.77	30.13

Tableau II.1 : L'énergie dans l'espace RGB.

Image	RE _Y (%)	RE _{Cb} (%)	RE _{Cr} (%)
Peppers	85.51	5.65	8.83
Lena	88.94	1.49	9.55
Airplane	99.20	0.52	0.27
Baboon	91.85	3.16	4.98
Elephant	96.64	2.78	0.56
House	96.30	1.10	2.58
Jerusalem	97.06	1.53	1.39
Monolake	89.27	5.39	5.33
Pelican	98.91	0.71	0.37
Athens	99.15	0.44	0.40
Moyenne	94.83	2.77	3.42

Tableau II.2 : L'énergie dans l'espace YCbCr.

Image	RE _Y (%)	RE _U (%)	RE _V (%)
Peppers	83.54	4.14	12.31
Lena	85.89	1.09	13.01
Airplane	99.22	0.36	0.40
Baboon	90.62	2.30	7.06
Elephant	97.09	2.13	0.77
House	95.54	0.78	3.66
Jerusalem	96.79	1.09	2.10
Monolake	88.32	3.79	7.87
Pelican	98.94	0.51	0.54
Athens	99.15	0.34	0.50
Moyenne	93.51	1.653	4.822

Tableau II.3 : L'énergie dans l'espace YUV.

Image	E _Y (%)	RE _I (%)	RE _Q (%)
Peppers	83.53	12.24	4.21
Lena	85.89	11.86	2.23
Airplane	99.22	0.43	0.33
Baboon	90.62	8.34	1.03
Elephant	97.08	1.92	0.98
House	95.54	3.93	0.51
Jerusalem	96.79	3.06	0.14
Monolake	88.31	11.40	0.28
Pelican	98.94	1.00	0.06
Athens	99.15	0.70	0.13
Moyenne	93.52	5.49	1.00

Tableau II.4 : L'énergie dans l'espace YIQ.

Image	RE _Y (%)	RE _{Db} (%)	RE _{Dr} (%)
Peppers	46.37	21.51	32.11
Lena	54.60	6.51	38.87
Airplane	94.92	3.24	1.83
Baboon	62.33	14.83	22.82
Elephant	80.47	16.52	3.007
House	79.53	6.11	14.34
Jerusalem	82.76	8.78	8.45
Monolake	54.91	22.08	23.00
Pelican	93.08	4.50	2.41
Athens	94.70	3.04	2.25
Moyenne	74.67	10.71	14.90

Tableau II.5 : L'énergie dans l'espace YDbDr.

Image	RE _Y (%)	RE _{Pb} (%)	RE _{Pr} (%)
Peppers	72.51	11.58	15.89
Lena	82.21	4.00	13.77
Airplane	98.66	0.81	0.51
Baboon	83.54	6.69	9.76
Elephant	91.96	5.65	2.38
House	93.02	2.40	4.57
Jerusalem	93.24	3.17	3.57
Monolake	78.39	9.88	11.72
Pelican	97.46	1.39	1.14
Athens	98.26	0.90	0.82
Moyenne	88.92	4.64	6.41

Tableau II.6 : L'énergie dans l'espace YPbPr.

Image	RE _{O1} (%)	RE _{O2} (%)	RE _{O3} (%)
Peppers	47.50	7.83	44.66
Lena	62.97	6.01	31.00
Airplane	95.22	0.51	4.26
Baboon	77.17	7.19	15.63
Elephant	85.86	3.51	10.61
House	88.19	3.07	8.73
Jerusalem	94.57	3.36	2.06
Monolake	84.19	11.41	4.38
Pelican	98.31	1.27	0.41
Athens	97.43	0.81	1.75
Moyenne	83.41	4.49	12.34

Tableau II.7: L'énergie dans l'espace O1O2O3.

Image	RE _X (%)	RE _Y (%)	RE _Z (%)
Peppers	42.21	41.41	16.36
Lena	45.12	32.40	22.46
Airplane	32.45	32.46	35.08
Baboon	35.06	34.25	30.67
Elephant	38.73	39.91	21.35
House	32.68	31.64	35.66
Jerusalem	31.44	33.20	35.34
Monolake	24.30	29.50	46.19
Pelican	32.66	32.64	34.68
Athens	34.88	32.96	32.15
Moyenne	34.95	34.03	30.95

Tableau II.8 : L'énergie dans l'espace XYZ.

L'image	RE _H (%)	RE _S (%)	RE _L (%)
Peppers	20.51	47.43	32.04
Lena	44.31	26.90	28.77
Airplane	34.68	3.81	61.50
Baboon	28.05	28.74	43.20
Elephant	10.46	34.20	55.32
House	42.76	15.15	42.08
Jerusalem	49.01	16.41	34.57
Monolake	35.08	32.27	32.64
Pelican	22.89	14.59	62.51
Athens	46.72	6.20	47.07
Moyenne	33.45	22.57	43.97

Tableau II.9 : L'énergie dans l'espace HSL.

Image	RE _H (%)	RE _S (%)	RE _V (%)
Peppers	14.01	42.69	43.30
Lena	36.81	22.59	40.59
Airplane	32.20	5.48	62.31
Baboon	22.25	23.21	54.53
Elephant	7.76	32.00	60.22
House	36.67	12.98	50.34
Jerusalem	36.92	26.96	36.10
Monolake	25.58	31.72	42.68
Pelican	22.21	7.86	69.91
Athens	39.89	13.29	46.80
Moyenne	27.43	21.88	50.68

Tableau II.11 : L'énergie dans l'espace HSV.

Image	RE _C (%)	RE _M (%)	RE _Y (%)
Peppers	49.16	38.12	12.71
Lena	58.52	21.03	20.44
Airplane	31.89	32.66	35.43
Baboon	38.22	32.87	28.90
Elephant	42.22	38.94	18.82
House	33.33	30.89	35.77
Jerusalem	30.10	34.76	35.12
Monolake	18.22	34.72	47.05
Pelican	32.43	32.66	34.90
Athens	36.93	31.01	32.05
Moyenne	37.10	32.76	30.11

Tableau II.13 : L'énergie dans l'espace CMY.

Image	RE_L(%)	RE_a(%)	RE_b(%)
Peppers	66.78	18.19	15.02
Lena	76.67	19.95	3.37
Airplane	96.80	2.20	0.99
Baboon	86.52	5.69	7.77
Elephant	91.13	1.45	7.41
House	92.50	5.78	1.70
Jerusalem	96.19	1.12	2.67
Monolake	92.81	2.19	4.98
Pelican	97.51	1.02	1.45
Athens	95.70	2.15	2.13
Moyenne	89.26	5.97	4.74

Tableau II.10 : L'énergie dans l'espace Lab

Après cette étude énergétique de chaque espace de couleurs, on peut résumer les différentes observations dans les points suivants :

- Degré élevé de corrélation entre les trois plans de l'espace RGB.
- Dans les espaces de types luminance/chrominance, l'énergie est concentrée dans la première composante (Y). C-à-d, elle contient les informations principales de l'image. Par exemple, dans l'espace YUV, l'énergie moyenne dans la composante Y est de 93.51%, mais dans les autres composantes (U , V) elles sont respectivement 1.65%, 4.82%. La même chose est constatée pour les autres espaces (YIQ, YCbCr, YDbDr, YPbPr, O1O2O3...).
- La majorité des informations de l'image couleur est dans la première composante (Luminance).
- Dans les espaces HSV et HSL, en général, l'information est concentrée dans les deux composantes teintes et luminance (premier et troisième plans).
- Dans l'espace XYZ il y'a une forte corrélation entre les composantes X , Y et Z .
- Dans l'espace Lab, les informations principales de l'image se trouvent dans la composante qui représente la luminance (L).
- Dans l'espace CMY les résultats sont similaires à ceux de RGB, c-à-d la corrélation entre C , M et Y est très élevée.

II.9. Conclusion

Dans ce chapitre, nous avons présenté les principaux espaces de couleurs. Une analyse détaillée en terme énergétique de chaque espace a été effectuée dans le but de trouver le meilleur espace de couleurs pour la compression (objectif de notre travail). Notre conclusion peut se résumer à ce que les espaces de type luminance/chrominance sont les plus adaptés à la compression car l'oeil humain perçoit mieux les changements de luminosité que les changements de couleur. Donc, Il en résulte que nous pouvons nous contenter de transmettre l'information de couleur avec moins de détails que l'information de luminance.

Par conséquent, dans le chapitre 4, on confirmera le choix adéquat (en se basant sur la distribution énergétique) de l'espace de couleurs le mieux adapté à la compression des images couleurs RGB.

Chapitre III

Revue des méthodes de compression

III.1. Introduction

Les méthodes de compression et de codage réduisent le nombre moyen de bits par pixel à stocker ou à transmettre, en exploitant la redondance informationnelle de l'image. A cet effet, plusieurs recherches scientifiques ont contribué à la naissance de diverses méthodes de compression.

Notons que la compression d'images est divisée en deux axes principaux : compression sans perte et celle avec perte. Le premier type de compression, utilise uniquement le principe de la réduction de l'information et n'engendre pas de perte, le deuxième type, quand à lui, définit une représentation approximative de l'information.

Ce chapitre se limite aux cas particuliers des images fixes, et à faire l'inventaire des méthodes disponibles, en vue de comprendre le principe et de tracer les grandes lignes prévisibles dans ce domaine.

III.2 Le but de la compression d'image

La compression consiste à réduire la taille physique d'un bloc d'information (le nombre de bits) nécessaire pour représenter les caractéristiques d'une image (en éliminant les redondances) pour permettre la réduction du coût de stockage et la transmission rapide des données.

Trois sortes de redondances sont exploitées dans la compression d'images [21] :

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

III.3. Mesures de performance de la compression d'image

III.3.1 Taux de compression :

Ce critère est très important puisqu'il dépend directement de la technique de compression utilisée [16] :

Il est représenté soit comme une formule, équation (III.1), soit comme un facteur, équation (III.2). Dans les équations (III.1) et (III.2), I_0 est la taille de l'image originale en octet et I_c la taille de l'image compressée.

Le taux de compression peut être aussi quantifié par le nombre moyen de bits par pixel (bpp), équation (III.3). L'élément $Bits_{Ic}$ est le nombre total de bits de l'image compressée et $Pixels_{I0}$ est le nombre total de pixels de l'image originale [16].

$$T_c = \frac{I_0}{I_c} \quad (III.1)$$

$$T_c = \left(\frac{I_0}{I_c} \right) : (1) \quad (III.2)$$

$$T_c = \frac{Bits_{Ic}}{Pixels_{I0}} \quad (III.3)$$

III.3.2 Taux d'information (Entropie)

L'entropie est une grandeur qui caractérise la quantité d'information que contient une image. Par exemple, une image dont tous les pixels ont la même valeur contient très peu d'informations car elle est extrêmement redondante, donc son entropie est faible. En revanche une image dont tous les pixels ont une valeur aléatoire contient beaucoup d'informations, son entropie est forte [21] [5].

L'entropie (en bits) est calculée par la formule suivante [39]:

$$H = - \sum_{i=1}^N P_i \log_2 P_i \quad (III.4)$$

où P_i est la probabilité de présence du niveau de gris i , et N étant le nombre du niveau de quantification ($N = 256$ pour une image à niveau de gris codée sur 8 bits). Plus l'entropie est faible, moins on a besoin de bits pour représenter les niveaux de gris des pixels de l'image.

Par définition l'entropie d'ordre zéro H_0 est donnée par [5]:

$$H_0 = - \sum_{i=1}^{2^R-1} P_i \log_2 P_i \quad (\text{III.5})$$

où R est le nombre de bits par pixels.

L'entropie H_0 d'une image originale fournit le débit minimal qu'il est possible d'atteindre par compression, pixel par pixel sans dégrader l'image, est par la même, un taux de compression sans perte maximale [5].

III.3.2. Mesures de distorsion (Mesures de fidélité)

Deux techniques sont utilisées pour évaluer la distorsion : subjective et objective [16].

- Les méthodes subjectives, nécessitent des tests psychovisuels de l'oeil humain. Les tests sont réalisés à plusieurs échelles avec des groupes de personnes.
- Les méthodes objectives, s'agissant de définir des quantités permettant d'évaluer numériquement la qualité de l'image reconstruite.

La distorsion (D) est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale [5]. La mesure de distorsion utilisée généralement en compression d'image, est l'erreur quadratique moyenne MSE (Mean Square Error). Cette grandeur est définie par la moyenne des écarts au carré entre le pixel $I(m, n)$ de l'image originale et le pixel $\hat{I}(m, n)$ de l'image reconstruite comme suit :

$$MSE = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I(m, n) - \hat{I}(m, n)]^2 \quad (\text{III.6})$$

Le rapport signal sur bruit (Signal to Noise Ratio) est défini par [21] :

$$SNR = 10 \cdot \log_{10} \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [x(m, n)]^2}{MSE} \quad (\text{III.7})$$

Le rapport signal crête sur bruit pour une image dont le maximum est $(2^R - 1)$ dénoté $PSNR$ (Peak Signal to Noise Ratio) entre la source initiale et celle distordue, où R est le nombre de bits de l'image originale, il est déterminé par la formule [21] :

$$PSNR = 10 \cdot \log_{10} \left(\frac{(2^R - 1)^2}{MSE} \right) \text{dB (Décibels)} \quad (\text{III.8})$$

En compression d'images le $PSNR$ d'une image de taille $(8 \times (512)^2)$ bits ($(512)^2$ indique une image de taille 512 par 512 pixels, chaque pixel est codé sur 8 bits) est défini plus souvent par [5].

$$PSNR = 10 \cdot \log_{10} \frac{(255)^2}{MSE} \quad (\text{III.9})$$

Le $PSNR$ d'une image couleur RGB est défini par l'équation [17] :

$$PSNR = 10 \cdot \log_{10} \frac{255^2 \times 3}{MSE(R) + MSE(G) + MSE(B)} \quad (\text{III.10})$$

Les mesures de distorsion sont très utiles pour déterminer la performance d'une méthode par rapport à d'autres. Ces mesures de distorsion sont objectives et simples à calculer. Certaines méthodes de compression recherchent le meilleur compromis entre la performance et la distorsion [21].

L'inconvénient de cette méthode est qu'elle ne rend pas compte de la perte de qualité visuelle engendrée par la compression. De plus, la MSE par exemple est une mesure *globale* sur toute

l'image, qui gomme les variations locales. Par exemple, dans une image médicale, si des détails anatomiques importants sont dégradés par la compression et si la majeure partie du reste de l'image est fidèlement restituée, alors la MSE est relativement faible mais, pour l'expert médical, cette image a une qualité diagnostique médiocre [21].

De très nombreuses recherches visent à trouver des mesures objectives de distorsion qui prédisent suffisamment bien la qualité perceptuelle [21]. Ces travaux ont apporté une connaissance sur les réponses du système visuel humain (dénomé Human Visual System: HVS) à certaines formes de dégradation.

III.4. Principe général de la compression d'images

La problématique de la compression peut être considérée de différents points de vue. Dans cette section nous recherchons à établir un modèle général qui guidera l'étude des méthodes de compression.

Le schéma général souvent utilisé pour décrire le fonctionnement des algorithmes de compression est celui présenté dans la figure III.1 :

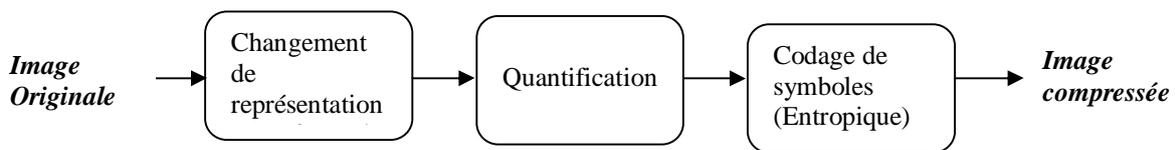


Figure III.1 : Etapes principales de compression d'images.

Dans ce modèle, le codeur, qui reçoit en entrée l'image, réduit les redondances et produit en sortie un code binaire. On distingue trois blocs :

- **Transformation ou 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. La décorrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés pour réduire le volume d'information, c'est une opération réversible [5].
- **Quantification** : réalise une réduction du nombre de valeurs représentées, qui fait partie uniquement des algorithmes irréversibles (consiste à réduire les redondances psychovisuelles, elle peut être scalaire ou vectorielle) [21].
- **Le codage entropique** : le codage entropique effectue un codage sans perte sur les valeurs quantifiées [21]. Cette dernière étape est nécessaire dans les méthodes sans perte, mais elle est souvent présente aussi dans les algorithmes irréversibles, puisque les valeurs transformées et quantifiées contiennent davantage de redondances.

III.5. Classification des méthodes de compression

On peut regrouper les méthodes de compression en cinq classes à l'aide de différents critères [19]. Puis nous détaillons dans les sections suivantes les méthodes les plus courantes, en incluant la description, l'algorithme et la performance de chacune.

III.5.1. Méthode avec ou sans perte d'information :

Cette première classification s'intéresse à la présence ou non d'une distorsion ou perte d'information introduite par la compression.

- *Compression sans perte* : la compression est dite sans perte lorsqu'il n'y a aucune perte de données sur l'information d'origine, Il y a autant d'information après la compression qu'avant [21].
- *Compression avec perte* : la compression avec perte ne s'applique qu'aux données perceptuelles, en général sonores ou visuelles, qui peuvent subir une modification, parfois importante, sans que cela ne soit perceptible par un humain [21]. La perte d'information est irréversible, il est impossible de retrouver les données d'origine après une telle compression.

La compression avec perte est pour cela parfois appelée compression irréversible ou non conservatrice

III.5.2. Méthodes par pixels, bloc de pixels, ou image entière (scène) :

Cette deuxième classification s'intéresse à la zone de travail de base sur laquelle s'appliquent les algorithmes de compression : le niveau pixel, blocs de pixels, ou toute l'image [1].

- *Niveau pixel* : dans cette approche considérant l'image comme un tableau de pixels, le codeur reçoit un par un les pixels, le code et le transmet au décodeur. Cependant, il est impératif d'exploiter les dépendances qui existent entre les pixels adjacents afin d'augmenter la performance de cette approche [5]. On peut citer quelques exemples, codage de Huffman; quantification scalaire et méthode sans perte par DPCM (Differential Pulse Code Modulation).
- *Niveau bloc* : cette stratégie est très appliquée principalement par les algorithmes irréversibles, elle consiste à regrouper les pixels en blocs, qui, en général, ne se recouvrent pas, et l'ensemble de ces blocs couvre l'image [5]. Cette fois l'unité de codage est le bloc. Par conséquent, toutes les opérations s'effectuent au niveau de blocs au lieu de pixels. Le passage à ce niveau supérieur est très important, puisqu'il y a des possibilités plus larges pour interpréter un bloc que des pixels individuels, ceci est en exploitant les corrélations intra-bloc.

Les standards JPEG et MPEG travaillent par blocs adjacents de 8x8 pixels et une quantification scalaire [21].

- *Niveau scène* : Le contenu de l'image est l'information du plus haut niveau que l'on peut exploiter dans un système de compression.

Des approches de codage par région, transformée cosinus discret DCT, codage d'images sous bandes par filtrage ou décomposition en ondelettes sont des exemples des méthodes appliquées au niveau scène [21].

III.5.3. Méthodes Intra et Inter-images :

Cette troisième classification s'applique aux séquences d'images (séries temporelles ou séries de coupes 3D). Les méthodes intra-images (intra-frame en anglais) effectuent la compression de chaque image indépendamment. Les méthodes inter-images (inter-frame) exploitent la redondance entre les images successives. Le standard MPEG code des séquences d'images en détectant le mouvement d'une image à l'autre [5].

III.5.4. Méthodes spatiales et méthodes par transformation :

La quatrième classification s'intéresse au domaine dans lequel s'effectuent les opérations de base de la compression. Une image peut être représentée de deux façons strictement équivalentes :

- ❖ Dans le domaine spatial, dans lequel l'image est représentée sous forme de pixels. C'est le domaine accessible visuellement à l'observateur.
- ❖ Dans le domaine fréquentiel, dans lequel l'image est représentée sous forme de coefficients de fréquences.

Le passage d'un domaine à l'autre se fait par des transformations mathématiques les plus connues, telles que la transformation de Fourier ou la transformation en cosinus discrète (DCT, Discrète cosine transform) [23].

Dans le domaine spatial, l'information contenue dans l'image est distribuée sur toute la matrice image. Mais dans le domaine de fréquences, l'information est généralement plus concentrée. Cette approche est largement utilisée, par exemple dans les standards JPEG [26] et MPEG.

III.5.5. Méthodes adaptatives, non adaptatives :

Cette cinquième classification indique si la méthode de compression est adaptative ou non.

- *Méthode non-adaptative* : applique les mêmes paramètres de façon fixe, quel que soit l'ensemble de données à coder.

- *Méthode adaptative* : modifie ses paramètres au fur et à mesure du codage, en s'adaptant aux données d'entrée.

Il peut également y avoir une compression semi adaptative (mélange des deux méthodes) [5].

III.6. Méthodes réversibles ou sans perte

Les méthodes sans perte peuvent s'appliquer dans le domaine spatial, mais plus difficilement dans le domaine fréquentiel. Ce type de compression n'occasionne aucune perte de données. Il est utilisé dans des applications comme l'archivage des images médicales, l'imagerie satellitaire (le coût des images est élevé et les détails sont importants), les textes, les programmes et tout autre type de données nécessitant une conservation intégrale.

Dans ce cas, les compromis liés à ce mode de compression sont [21] :

- Efficacité du codage, ceci peut être mesuré en bit par pixel, elle est limitée par l'entropie de la source. Plus l'entropie de la source est grande plus il est difficile à compresser.
- Temps de codage, celui-ci est lié à la complexité du processus de codage ou de décodage. Il peut être réduit si on augmente la capacité de calcul du composant de traitement. Pour certaines applications ce temps est contraint, ce qui impose le choix de la technique de codage.
- Complexité du codeur, elle peut être mesurée à l'aide de la quantité de ressources utilisées en termes de mémoire et du nombre d'opérations arithmétiques.

Cette famille d'algorithmes est essentielle dans nos ordinateurs, la totalité des programmes d'archivage sont bâtis sur cette technique de compression sans perte d'information.

Il existe de nombreux types d'algorithmes de compression d'images sans perte. Voici les plus répandus.

III.6.1. Méthodes différentielles et prédictives

Ces méthodes exploitent la redondance entre un pixel et ses voisins, qui en général se ressemblent beaucoup. Par exemple, on code le premier pixel, on calcule la différence avec le second pixel et on code cette différence. Cette dernière nécessite moins de bits que les pixels eux-mêmes car cette différence est souvent faible. On code ensuite la différence entre le deuxième pixel et le troisième, etc...

Dans des systèmes plus complexes et performants, on établit une fonction de prédiction qui permet d'estimer la valeur d'un pixel en fonction des valeurs des pixels voisins. On code alors l'erreur de prédiction, qui est l'écart entre la vraie valeur du pixel et la valeur prédite.

La façon de coder les erreurs de prédiction est souvent basée sur la quantification scalaire, que nous verrons plus loin. Le codage prédictif ainsi décrit correspond à la modulation par impulsion et codage différentielle (MICD, ou DPCM en anglais) [23] [24].

L'efficacité d'une méthode prédictive est fortement liée à l'adéquation de la fonction de prédiction aux pixels à coder afin que les valeurs d'erreurs de prédiction soient faibles.

Dans les systèmes numériques d'angiographie cardiaque, la société *Philips Medical Systems* utilise depuis le début des années 90 un système basé sur la méthode DPCM pour le codage des images stockées sur disque dur [21].

Une méthode bien connue de type différentielle ou prédictive est la modulation delta (DM) [21]. C'est la plus simple de ces méthodes. On transmet le premier pixel, et on prédit chaque pixel comme étant égal à son prédécesseur. L'erreur de prédiction est codée sur 1 bit en ne conservant que l'information sur le signe de cette erreur. On décode chaque pixel en l'augmentant ou le diminuant d'un pas fixe par rapport à son prédécesseur, selon que l'erreur de prédiction était positive ou négative.

III.6.2 Méthodes par plages

- Le Codage par plage (Run length Encoding) :

Le principe de compression RLE est assez simple à mettre en oeuvre. Il repose sur le fait que dans une image, il existe de nombreuses répétitions d'un même pixel, ou d'une même séquence de pixels, tous juxtaposés [22]. Ainsi, au lieu de coder chaque pixel d'une image, le RLE regroupe les

valeurs voisines identiques et ne transmet une valeur qu'une seule fois, précédée par le nombre de répétitions.

Il est clair que cette approche fonctionne bien s'il y a beaucoup de répétitions dans l'image. Cet algorithme très simple, peut aboutir à des taux de compression plus élevés. Un exemple de traitement RLE est donné à la figure III.3.

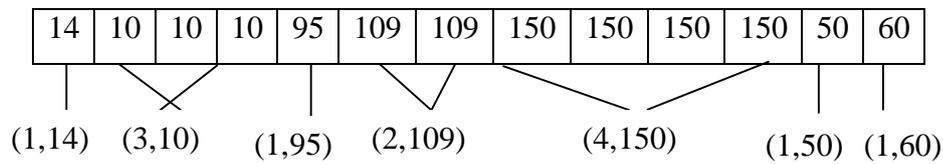


Figure III.3: Un exemple de codage par plage RLE.

S'il est relativement simple de coder l'octet à répéter, précédé du nombre de répétitions dans un nombre suffisant de bits, cette méthode peut se révéler très pénalisante dans certains cas : à la limite si deux octets consécutifs sont toujours différents, la taille des données compressées sera distendue par rapport à celle des données initiales.

Pour éviter cet inconvénient, les versions les plus avancées du RLE utilisent un code discriminant pour indiquer le début d'une séquence octet à répéter plus le nombre de répétitions, les octets isolés restant codés sous leur forme initiale [16].

Il est à noter qu'il existe plusieurs variantes de ce codage, certains types encodent l'image en décelant des séquences redondantes selon les lignes, les colonnes, ou même en zig-zag (figure III.4).

III.6. 3. Codage de shannon-fano :

C. Shannon du laboratoire Belis et R. M. Fano du MIT ont développé à peu près en même temps une méthode de codage basée sur la simple connaissance de la probabilité d'occurrence de chaque symbole dans le message [24].

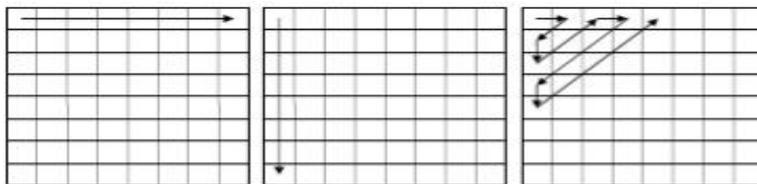


Figure III.4. Linéarisation en ligne, en colonne et en zig-zag.

Le procédé de Shannon-Fano consiste à construire une arborescence partant de la racine, et procédant par divisions successives. Le classement des fréquences se fait par ordre décroissant, ce qui suppose une première lecture du fichier et la sauvegarde de l'en-tête.

Le principe est le suivant:

1. Classer les n fréquences non nulles par ordre décroissant.
2. Deviser l'ensemble des fréquences en deux sous ensembles de fréquences aussi proches que possible.
3. Attribuer à chaque sous ensemble un bit 0 ou 1.
- 4 : Rediviser chaque sous ensemble en deux nouveaux sous ensembles de fréquences équivalentes afin de réitérer l'algorithme jusqu'à ce qu'il n'y ait plus qu'un seul élément dans chaque sous ensemble.
5. Attribuer aux symboles les codes binaires correspondant aux bits de description de l'arborescence.

Exemple :

La chaîne que nous allons traiter est : AABACBA de 7 octets.

Le tableau suivant classe les caractères par ordre décroissant de fréquences et contient aussi le code attribué suite au codage de Shannon.

Le taux de compression, sans tenir compte de l'en-tête est de 82%.
 La figure III.5 montre l'arborescence qui a permis la distribution des codes.

Symbole	Fréquence	Code attribué
A	4	1
B	2	01
C	1	00

Tableau III.1 : Exemple du codage de Shannon.

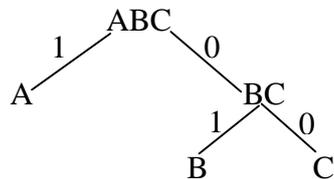


Figure III.5 : Arbre d'attribution des codes de Shannon.

II.6.4. Codage de Huffman :

L'imminent mathématicien, David Huffman, a proposé en 1952 une méthode statistique qui permet d'attribuer un mot de code binaire aux différents symboles à compresser (pixels ou caractères par exemple) [24] [40]. La longueur de chaque mot de code n'est pas identique pour tous les symboles: les symboles les plus fréquents (qui apparaissent le plus souvent) sont codés avec de petits mots de code, tandis que les symboles les plus rares reçoivent de plus longs codes binaires.

Le principe est le suivant:

1. Les probabilités d'occurrence de chaque message sont placées dans une liste dans un ordre décroissant. Nous dirons que la liste est composée d'enfants.
2. Les deux probabilités les plus faibles sont identifiées en fin de liste.
3. La somme des deux probabilités est placée à sa place dans la liste triée. Elle constitue un noeud parent. Les deux enfants sont retirés de la liste.
4. Le chemin «enfant de plus faible probabilité, parent» est codé par un 1, l'autre par un 0.
5. La procédure reprend à l'étape 2 jusqu'à ce qu'il ne reste plus qu'une probabilité dans la liste.

Le codeur de Huffman est très couramment employé en compression d'image. Il constitue très souvent l'étape finale produisant le flot binaire dans les méthodes par transformations.

Exemple : Soit un message à coder "“ABBBBAAC”". La fréquence d'apparition ainsi que le code de Huffman correspondant sont donnés dans le tableau III.2 et représentés par la figure III.6.

Symbole	Fréquence	Code
C	1	00
A	3	01
B	4	1

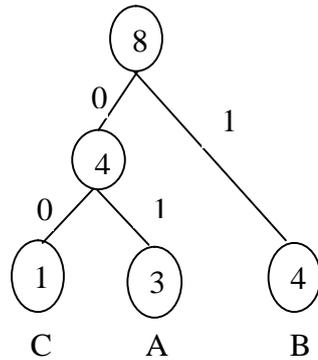


Tableau III.2 : Code de Huffman.

Figure III.6 : Arbre de Huffman.

II.6.5.Méthodes par dictionnaire adaptatif (Lempel-Ziv) :

C'est une technique de codage qui utilise un dictionnaire. On cherche dans le fichier les chaînes qui se répètent en les mémorisant dans un dictionnaire. Ensuite, le codage consiste à remplacer les chaînes mémorisées par leurs adresses (ou indices) construite dans le dictionnaire [16] [24].

Le premier algorithme de ce type a été mis au point par Lempel et Ziv en 1977 [23], et a donné lieu au programme LZ77. Celui-ci lit un flot de symboles et cherche des chaînes équivalentes dans une fenêtre de 4Ko précédant le flot d'entrée. Les équivalences sont remplacées par des codes.

Les formats ZIP, ARJ et LHA basent leur compression sur cet algorithme [22].

Lempel et Ziv ont développé une nouvelle version de leur algorithme en 1978, LZ78, où le dictionnaire est construit à partir de tous les symboles précédemment rencontrés et non par une fenêtre, il est spécialisé dans la compression d'images et de fichiers binaires [21][22].

Puis en 1984, Terry Welch modifia le format LZ78 pour pouvoir l'utiliser dans les contrôleurs de disques durs, le format LZW est né [22]. Le dictionnaire dans ce cas est initialement construit et contient l'ensemble des codes ASCII. Il est élaboré au fur et à mesure. Ce qui permet de changer la taille du dictionnaire au cours du codage [15].

Le codage LZW est une technique de compression réversible qui peut être appliquée à tout type de fichier de données, que ce soit : texte, image, fichier informatique, etc. Elle a été adoptée pour la mise en oeuvre du format de compression d'images 'GIF' [24].

II.6.6.Codage arithmétique :

Contrairement aux algorithmes de Huffman et de Shannon-Fano qui associent à des symboles des motifs binaires dont les tailles dépendent de leurs distributions. Le codeur arithmétique traite le fichier dans son ensemble, en lui associant un unique nombre décimal rationnel [24] [39]. Ce code de sortie est un nombre à virgule flottante compris entre 0 et 1, dont le nombre de chiffres après la virgule correspond au nombre de symboles. Le codeur arithmétique est plus performant que le codeur de Huffman, mais il est plus complexe à implémenter [21] [44].

Algorithme du codage arithmétique : nous décrivons brièvement ci-dessous l'algorithme de codage arithmétique dans le but d'en illustrer le principe, sachant que le décodage opère de manière inverse [24].

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

- largeur = limite supérieure - limite inférieure
 - limite inférieure = limite inférieure + largeur x (limite basse du sous intervalle du symbole)
 - limite supérieure = limite inférieure + largeur x (limite haute du sous intervalle du symbole)
- A la fin, n'importe quelle valeur de l'intervalle final représente d'une manière unique la séquence d'entrée

Généralement, on choisit comme représentant de la séquence la limite inférieure de l'intervalle, ou bien on peut choisir la valeur moyenne.

Il est à noter qu'à la fin du processus de codage, le codeur arithmétique génère un fichier composé d'une en-tête suivie de la représentation binaire du code choisi. De plus, l'en-tête peut contenir soit la table des probabilités, soit la table des fréquences de tous les symboles [27].

Exemple :

Supposons que l'on veuille coder une partie "acaab" d'une longue séquence avec une probabilité d'apparition indiquée dans le Tableau III.3 [16].

Symbole	Probabilité	Intervalle
A	0.7	[0, 0.7]
B	0.1	[0.7, 0.8]
C	0.2	[0.8, 1.0]

Tableau III.3 : Probabilités des symboles.

La figure III.7 donne une représentation du processus de codage arithmétique de la chaîne "acaab".

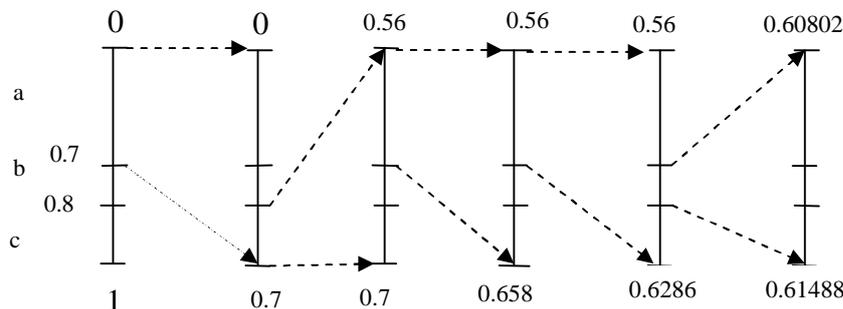


Figure III.7 : Génération de l'étiquette pour la séquence "acaab".

Un nombre contenu dans l'intervalle final comme 0.60803 code sans ambiguïté le message "acaab".

III.7. Méthodes avec perte ou irréversible

Contrairement aux méthodes de compression sans perte d'informations, ce type comporte une perte de données pendant le processus de codage/décodage. Le résultat qu'on peut en obtenir est une version altérée de l'image originale [24]. Le but de ce type de compression est d'éliminer le plus d'information possible sans atténuer la qualité de l'image perçue par le système visuel humain.

Avec ces méthodes, on peut aussi distinguer :

- *Les méthodes spatiales* (ou directes) qui agissent directement sur les échantillons d'une image dans le domaine spatial.
- *Les méthodes par transformation* qui reposent sur une transformée (en général linéaire) de l'image originale.

III.7. 1 Les méthodes spatiales

1. Sous-échantillonnage

Le sous-échantillonnage consiste à ne conserver qu'une partie des données [21] [44]. Par exemple, si on ne stocke qu'un pixel de l'image sur deux, on obtient un T_C de 4:1. L'image reconstruite s'obtient par interpolation, par exemple en remplaçant chaque pixel manquant par la moyenne de deux pixels adjacents. Cette méthode extrêmement simple est à employer avec précaution car la distorsion n'est pas contrôlée.

2. Le codage par troncature des blocs (BTC) :

BTC est une technique simple de compression des images au niveau de gris développé par Delp et Mitchell [31].

➤ L'algorithme de codage [31]:

Etape 1: Segmenter l'image originale en blocs de $n \times n$.

Etape2 : Pour chaque bloc, calculer la moyenne (\bar{X}) et l'écart type (σ), tels que :

$$\bar{X} = \frac{1}{k} \sum_{i=1}^k x_i \quad (\text{III.11})$$

$$\sigma^2 = \frac{1}{k} \sum_{i=1}^k x_i^2 - \bar{X}^2 \quad (\text{III.12})$$

où k est le nombre de pixels dans le bloc de taille $n \times n$ ($k=n^2$) et $x_1, x_2 \dots, x_k$ sont les valeurs d'intensité des pixels dans un bloc de l'image originale.

Etape3 : Construire le modèle de q niveaux (table binaire) de taille $n \times n$, si la valeur du pixel est supérieure à la moyenne (\bar{X}) alors la valeur du pixel est égale à 1 sinon elle est égale à 0.

Etape3 : Calculer les niveaux de reconstruction (a et b).

$$a = \bar{X} - \sigma \sqrt{\frac{q}{k-q}} \quad (\text{III.13})$$

$$b = \bar{X} + \sqrt{\frac{k-q}{q}} \quad (\text{III.14})$$

Où σ est l'écart type, k est le nombre total de pixel dans le bloc, et q est le nombre de pixels plus grands que \bar{X} .

Etape4 : Stocker a, b et la table binaire associés à chaque bloc.

➤ L'algorithme de décodage :

Pour reconstruire l'image, ou créer son approximation, les éléments qui sont égaux à 0 sont remplacés par la valeur de a , tandis que les éléments qui sont égaux à 1 sont remplacés par la valeur de b .

Il y a plusieurs modifications de la méthode de BTC pour améliorer la qualité, on peut citer par exemple :

La méthode AMBTC (Absolute Moment BTC) qui utilise le critère de la conservation des moments absolus des échantillons [17]. Elle s'appuie sur l'idée du codage par troncature de blocs, mais elle est plus simple dans sa mise en œuvre pratique.

Une autre variante de BTC est basée sur la prédiction qui donne une bonne qualité d'image en bas débit binaire avec un coût minimal dedans la procédure de codage et décodage [17].

Un BTC de trois niveaux est également employée pour améliorer la qualité des images reconstruites.

Ces variations des techniques de BTC peuvent être appliquées séparément sur chacun des plans de couleurs d'une image. Comme il y a un degré élevé de corrélation entre les plans de couleurs, ceci

peut être exploité pour réduire le débit binaire. La méthode (SBBTC), développée par Wu et Coll (d'après [31]), exploite cette corrélation inter-couleurs.

III.7. 2 Les méthodes par transformation

III.7. 2.1. Principe général :

Les méthodes par transformation figurent parmi les techniques de compression les plus employées, elles n'agissent pas directement sur l'image numérique dans sa représentation canonique, mais sur le domaine de sa transformée. Elles permettent d'obtenir des taux de compression élevés tout en conservant une bonne qualité d'image. Ce sont des méthodes qui font appel successivement à plusieurs principes de compression. Elles sont utilisées par des standards internationaux pour le codage des images fixes et de la vidéo (JPEG et MPEG) [21] [26].

En général, les schémas de codage par transformation subdivisent l'image de taille $N \times N$ en sous-images de taille plus petite avant de faire subir à chacune de ces sous-images une transformation.

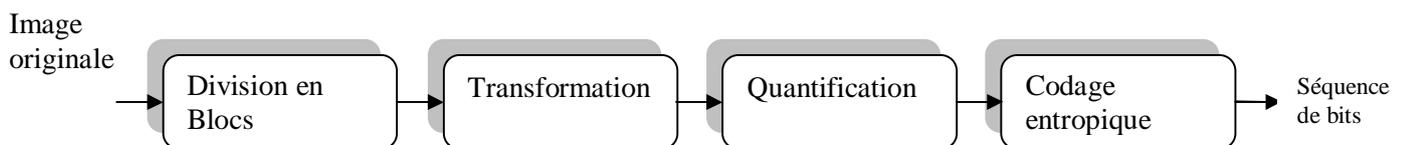
L'objectif de ces transformations est double :

- Il s'agit de décorréler les données, c'est-à-dire d'obtenir des coefficients transformés moins corrélés que les pixels de l'image d'origine ;
- Concentrer l'énergie sur un nombre réduit de coefficients, les coefficients ayant une valeur plus importante aux basses fréquences qu'aux hautes fréquences.

Dans ce cas, on obtiendra une compression effective en codant finement les coefficients des basses fréquences, et grossièrement, voire en supprimant, les coefficients des hautes fréquences.

Les méthodes par transformation suivent le schéma de fonctionnement présenté dans la Figure III.8.

Etape de compression:



Etape de décompression:

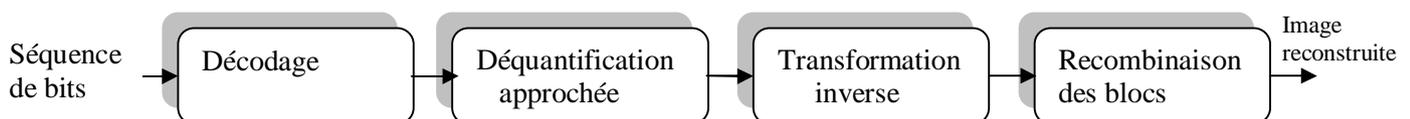


Figure III.8 : Schéma de principe de la compression / décompression par transformation.

1. Division en blocs

La première étape consiste à diviser l'image en blocs sur lesquels vont s'appliquer indépendamment les étapes suivantes. La principale raison de ce découpage est de limiter le nombre de pixels à traiter à la fois pour diminuer les temps de calcul. Il peut résulter de cette division un effet visuel appelé effet de "blocs" ou "artifacts" : à des taux élevés, les frontières des blocs deviennent visibles, car ils ont été comprimés indépendamment [21]. La taille des blocs est variable selon les méthodes. Elle est de 8×8 pour JPEG.

2. Transformation

La deuxième étape consiste à appliquer une transformation mathématique à chaque bloc. Le but de cette transformation est de décorréler les pixels, ce qui a pour effet en général de redistribuer l'énergie de l'image dans un nombre restreint de coefficients transformés. De cette façon, un grand nombre de coefficients transformés ont de très faibles valeurs, et peuvent être supprimés ou se voir allouer un nombre très faible de bits lors de l'étape de quantification [21].

Parmi les principales transformations linéaires les plus connues en compression d'images, on peut citer la transformation de Karhunen-Loeve (TKL), la transformation de Fourier discrète (TFD), la transformation en cosinus discrète (DCT), la transformation par ondelette discrète (DWT) et d'autres méthodes telles que les transformations de Hadamard (TH) et Haar (THA) [24].

Dans les sections suivantes on présente les transformations d'espace les plus courantes TFD, DCT et DWT.

3. Quantification et codage

La troisième étape est la quantification des coefficients transformés, afin de se ramener à un nombre limité de valeurs différentes. La quantification est souvent précédée d'une pondération psychovisuelle des coefficients, afin de préserver ceux auxquels l'oeil est le plus sensible [18].

On distingue en général deux types de quantification : la quantification scalaire et la quantification vectorielle [24] [41].

➤ La quantification scalaire

La quantification scalaire SQ -(Scalar Quantization) consiste à remplacer un nombre très grand de symboles par un nombre restreint de codes [21]. C'est une opération irréversible très largement employée en compression. Par exemple, le fait d'arrondir des nombres flottants aux entiers les plus proches est une quantification scalaire.

D'une manière générale, on peut la définir comme étant l'association de chaque valeur réelle x , à une autre valeur q qui appartient à un ensemble fini de valeurs. La valeur q peut être exprimée en fonction de la troncature utilisée : soit par l'arrondi supérieur, l'arrondi inférieur, ou l'arrondi le plus proche [16].

➤ Quantification Vectorielle

Le principe général du quantificateur vectoriel VQ -(Vector Quantization) consiste à quantifier une suite d'échantillons (vecteur) d'une image au lieu de quantifier chaque valeur, comme c'est le cas pour la quantification scalaire.

Pour réaliser ceci, on commence par créer un catalogue de vecteurs de référence appelé codebook ou dictionnaire à partir d'un ensemble d'images tests par un algorithme d'apprentissage [24].

De nombreux algorithmes de génération du dictionnaire ont été proposés : LBG [5] ; K-Means [5] ; KOHONEN, [23] et l'algorithme d'apprentissage à compétition (AC) [25].

La VQ produit de meilleurs résultats que la SQ , néanmoins la VQ nécessite un codage complexe et de grandes capacités de mémoire [16].

Un codage entropique est effectué sur les coefficients quantifiés, pour aboutir au flot binaire de sortie. En général, c'est le degré de quantification qui détermine le niveau de compression obtenu. Une quantification grossière donne une compression importante mais introduit une forte dégradation.

III.7. 2.2.Transformée de fourrier (DFT)

La transformée de Fourier discrète (TFD) est un outil mathématique de traitement du signal numérique. Elle est l'équivalent discret de la transformée de Fourier continue utilisée pour le traitement du signal analogique.

Sa définition mathématique pour un signal S de N échantillons est la suivante [21] :

$$S(k) = \sum_{n=0}^{N-1} s(n).e^{-2i\pi \frac{nk}{N}} \quad (\text{III.11})$$

La transformée inverse est donnée par :

$$S(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k).e^{2i\pi \frac{nk}{N}} \quad (\text{III.12})$$

III.7. 2.3.Transformation en Cosinus Discrète (DCT)

C'est une transformation mathématique qui transforme un ensemble de données d'un domaine spatial en un spectre de fréquences, elle est très utilisée en traitement du signal et de l'image, et spécialement en compression [28]. La DCT possède en effet une excellente propriété de

regroupement de l'énergie (l'information est essentiellement portée par les coefficients basses fréquences).

La DCT est effectuée sur une matrice carrée $N*N$ de valeurs de pixels et donne une matrice carrée $N*N$ de coefficients de fréquence. Le temps de calcul requis pour chaque élément dans la DCT dépend de la taille de la matrice. Vu la difficulté d'appliquer la DCT sur la matrice entière, celle-ci est décomposée en blocs de taille $8*8$ pixels (compression JPEG).

A la sortie de la DCT, la valeur de la position (0,0) est appelée *DC* (Direct Component). Cette valeur représente une moyenne de la grandeur d'ensemble de la matrice d'entrée. Le coefficient DC est plus grand que toutes les autres valeurs de la matrice DCT appelés *AC* (Alternative Component).

Par convention, les 64 valeurs transformées (de chaque bloc) sont positionnées d'une certaine manière, ainsi la valeur moyenne de tous ces coefficients est placée en haut à gauche de ce bloc [5]. Plus on s'éloigne plus leur grandeurs tendent à diminuer. Ce qui signifie que la DCT concentre la représentation de l'image en haut à gauche de la matrice de sortie, les coefficients en bas et à droite de cette matrice contiennent moins d'information utile.

Les équations qui suivent, donnent respectivement les transformées en cosinus discrètes directe et inverse [16].

$$DCT(i, j) = \frac{1}{\sqrt{2N}} c(i)c(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} Pixel(x, y) \cos\left(\frac{(2x+1)i\Pi}{2N}\right) \cos\left(\frac{(2y+1)j\Pi}{2N}\right) \quad (III.13)$$

$$Pixel(i, j) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} c(i)c(j) DCT(i, j) \cos\left(\frac{(2x+1)i\Pi}{2N}\right) \cos\left(\frac{(2y+1)j\Pi}{2N}\right) \quad (III.14)$$

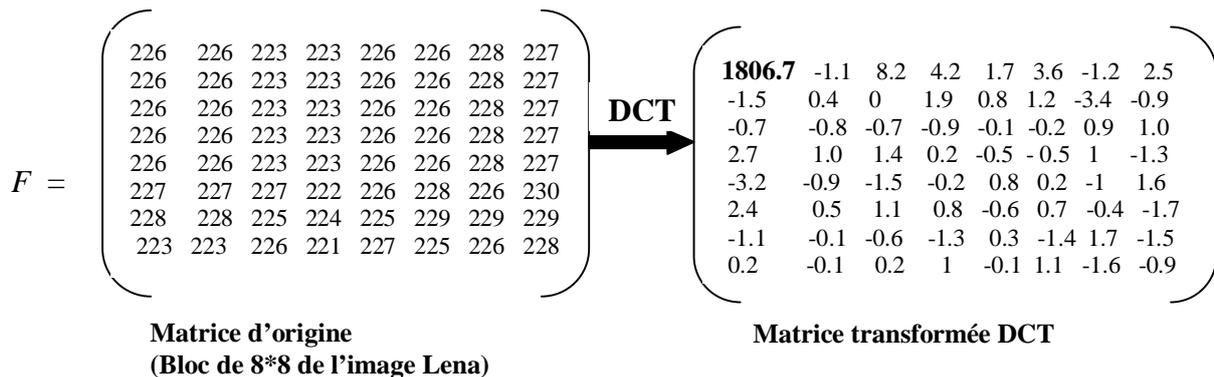
Avec : $c(i)=1/\sqrt{2}$ si $i=0$; $c(i)=1$ si $i>0$

$Pixel(X, Y)$ désigne la valeur du pixel de coordonnées (x, y)

$DCT(i, j)$ le coefficient repéré par la ligne i et la colonne j dans la matrice DCT.

En pratique, l'image couleur est généralement codée en RGB, et traitée comme trois images monochromes ou plans. La DCT est appliquée successivement aux données du plan de R, G et B.

Pour illustrer la transformation DCT on prend l'exemple suivant :



III.7. 2.3. Codage en sous-bandes (SBC) :

Le codage en sous-bandes a été introduit pour la parole par Crochiere et al. en 1976[21], puis appliqué à l'image sous la forme que nous connaissons aujourd'hui par Woods et O'Neil en 1986 [5]. Il consiste à décomposer le signal ou l'image en différentes bandes de fréquences spatiales et à coder chacune d'entre elles de manière indépendante ou non [30]. Dans les schémas classiques, la bande spectrale du signal original est d'abord divisée en deux parties à l'aide de deux filtres numériques (passe-bas et passe-haut). La procédure est ensuite répétée dans chacune des sous-bandes fréquentielles. Le résultat constitue une structure arborescente, symétrique ou non,

composée d'un nombre entier de sous-bandes. Le signal est reconstruit par recombinaison des sous-bandes à l'aide de filtres d'interpolation [21].

La Figure III.9 montre la décomposition d'une image en quatre sous-bandes, la première correspondant aux basses fréquences (approximation de l'image), la deuxième aux hautes fréquences colonnes (détails horizontaux), la troisième aux hautes fréquences lignes (détails verticaux), la quatrième aux hautes fréquences lignes et colonnes (détails diagonaux).

Après décomposition en sous-bandes et sous-échantillonnage, les sous-images résultantes sont codées avec des stratégies adaptées à leur contenu énergétique. Tout comme les méthodes par transformation, on tend à privilégier les basses fréquences qui sont riches en énergie, et à coder plus grossièrement les hautes fréquences en prenant en compte la sensibilité de l'oeil humain. Les méthodes de codage couramment employées sur les sous-bandes sont la DPCM pour la sous-bande basse fréquence, et la quantification vectorielle pour les autres sous-bandes [21] [30].

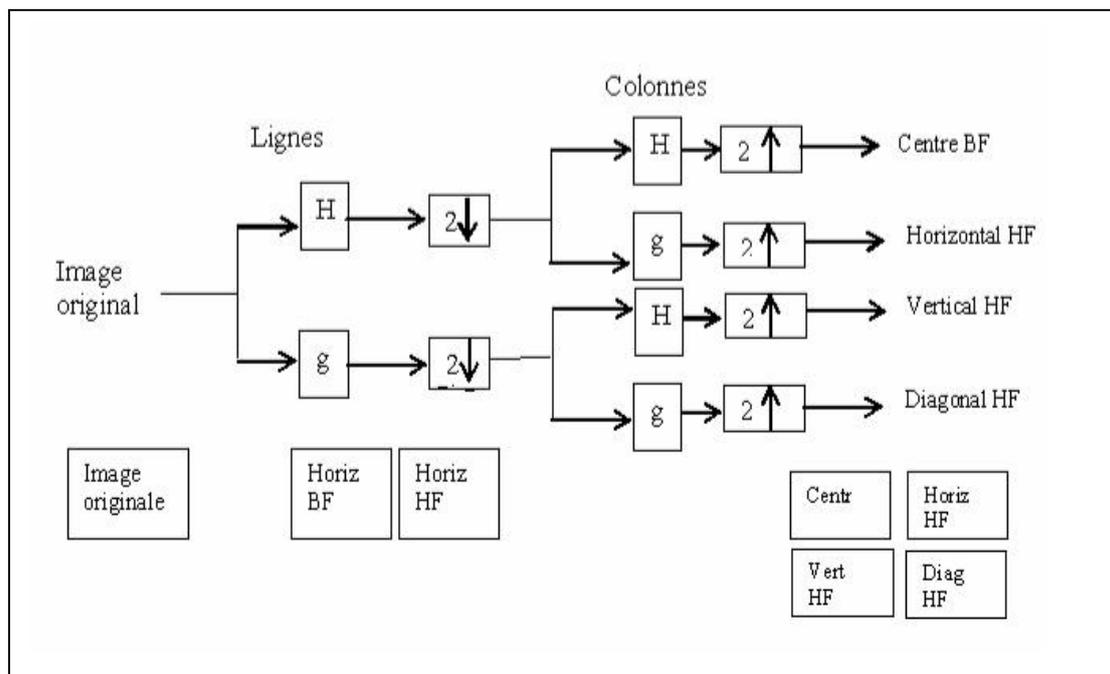


Figure III.9 : Décomposition d'une image en sous-bandes.

Les deux éléments clés d'un schéma de codage en sous-bandes sont :

- Le choix des bancs de filtres pour la décomposition sous-bandes
- Le choix de(s) la technique(s) de codage appliquée(s) aux sous-bandes.

Ces deux éléments ont suscité un grand intérêt pour les techniques multirésolution qui se sont traduites par les travaux sur l'utilisation des ondelettes en analyse du signal (Rioul et Vetterli) dans les années 80, et de nombreuses applications au codage de la parole, de la musique et de l'image fixe et animée. C'est un domaine encore très actif aujourd'hui [21].

III.7. 2.4. La transformation par ondelette discrète (DWT) :

Les ondelettes c'est d'abord une théorie mathématique récente d'analyse du signal, développée dans les années 80. On peut considérer qu'il s'agit d'une extension de l'analyse de Fourier [21] [28]. L'intérêt de cette théorie est au départ l'analyse des signaux.

La technologie de compression à base d'ondelettes offre une plus grande finesse au niveau de l'analyse du signal, et permet de mieux s'adapter aux propriétés locales de l'image.

La transformation par ondelette est une technique de compression d'image fixe très performante [26].

L'intérêt de la transformation par ondelettes par rapport aux autres transformées de compression est que celle-ci ne considère pas l'image dans son ensemble pour la coder mais, la travaille par couche, cherchant à enregistrer les détails les plus importants à chaque résolution [21].

➤ **L'aspect mathématique :**

La transformée en ondelette repose sur le fait que toute fonction f intégrable, ce qui se note couramment $\int_{-\infty}^{+\infty} f(x)dx$, peut être vue sous la forme d'une somme de fonction $\Psi_{a,b}(x)$, les fonctions ondelettes, toutes issues d'une seule et unique fonction, l'onde mère de manière à ce que, pour tout b réel et a strictement positif, et pour tout x réel, $\Psi_{a,b}(x) = \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right)$ voir [22] [24].

➤ **Algorithme de compression par ondelettes ou Wavelets :**

Passons maintenant à l'algorithme pyramidal utilisé. La décomposition en coefficients d'ondelettes n'utilise pas une fonction de moyenne, mais s'appuie sur deux filtres. Un filtre passe bas (L) et un filtre passe haut (H). La combinaison de ces filtres permet d'obtenir quatre sous images HH, HL, LH et LL. Ces filtres sont nommés filtres miroirs en quadrature (figure III.10).

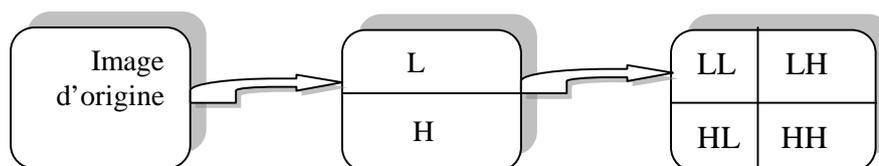


Figure III.10 : Transformation en colonnes et en lignes.

Chacune des quatre images obtenues par la transformation représente des informations bien distinctes.

- Les étapes de compression par ondelettes sont usuellement :
 1. Transformation par ondelettes.
 2. Quantification : les valeurs des coefficients de détails inférieurs à un certain niveau sont éliminés, en fonction de l'efficacité recherchée. C'est cette étape qui introduit des pertes.
 3. Codage des valeurs restantes : les données restantes sont transmises à un encodeur entropie, c'est à dire à un algorithme de compression de données (LZW, HUFFMAN, RLE,...).

- **Décompression ondelettes :**

La transformation inverse par ondelettes reconstruit une image approximée. La reconstruction de l'image à partir des sous-bandes restitue l'image en mode progressif.

L'affichage de l'image s'effectue selon le mode suivant :

La résolution de l'image augmente au fur et à mesure de la lecture du fichier compressé comme illustré sur la figure III.11.

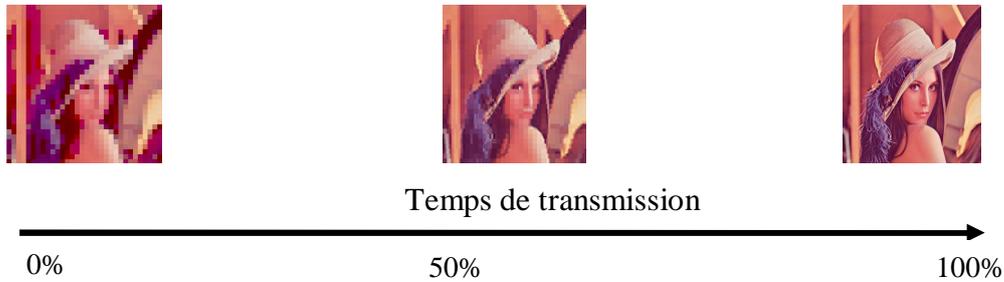


Figure III.11 : Décompression progressive .

Les principaux algorithmes basés sur la DWT :

Il y a plusieurs algorithmes basés sur la transformée en ondelette pour la compression des images. Ceux-ci incluent EZW, SPIHT, SFQ, CREW, EPWIC, EBCOT, SR, codage d'image en utilisant des paquets d'ondelettes et la VQ [28]. Nous discuterons brièvement quelques uns de ces algorithmes.

- Embedded Zerotree Wavelet coding (EZW) :

La méthode de codage progressif connue sous le nom de "Embedded Zerotree Wavelet coding" (EZW), proposée par Shapiro [24][23], est une méthode simple et très efficace. Elle a démontré sa puissance dans les deux formes de compression (avec et sans perte d'informations) depuis son élaboration en 1993.

En général, dans une représentation d'image par coefficients d'ondelettes, l'image obtenue est organisée de façon à représenter les principaux traits de l'image dans les bandes de basses fréquences, puis les détails dans les bandes de hautes fréquences. Le principe de l'EZW s'appuie sur cette représentation, pour coder les coefficients d'une manière progressive, ainsi, on commence par les basses fréquences L , ensuite on code les détails (hautes fréquences). L'avantage de cet algorithme, est que l'on a en tout temps un niveau de compression et que l'on peut arrêter en tout moment le codage [24].

Plusieurs variantes de ce type de codage ont été proposées par différents chercheurs dans le domaine, ce qui fait sa force. On peut citer par exemple le SPIHT (Set Partitioning In Hierarchical Tree) réalisé par A. SAID et W. PEARLMAN [24][28] qui est la variante la plus populaire de l'EZW.

- Set Partitioning in Hierarchical Trees (SPIHT) :

Cette méthode proposée par A. Said et Pearlman [24] diffère de ses antécédentes par la manière avec laquelle les coefficients sont classés ainsi que pour le traitement des coefficients significatifs. Un autre point important est que cette méthode, génère un flot binaire intéressant (au niveau de la compression) sans passer par un codage entropique contrairement au codage EZW dont les résultats n'apparaissent qu'une fois associé au codage arithmétique [24][28].

Le principal avantage de cet algorithme est qu'il est plus rapide en exécution que EZW, et peut donner de meilleurs résultats [24].

III.8 Les normes de compression des images

III.8.1. La norme de compression JPEG :

La norme JPEG (Joint Photographic Experts Group) est conçue par le groupe ISO (International Standards Organisation) et le groupe CEI (Commission Electronique Internationale). Elle est destinée à la compression des images fixes en couleurs et à niveaux de gris en vue de leurs stockages sur les supports numériques [26].

Ce standard a vu le jour en 1992. Il est devenu le format le plus populaire pour le stockage en photographies numériques, car il offrait alors une meilleure compression dépassant tous les formats connus à l'époque.

Les techniques définies par la norme JPEG se divisent en deux classes : les méthodes de compression avec pertes qui sont basées sur la DCT suivie d'une quantification et d'un codeur entropique.

La seconde classe, concerne les processus de codage sans pertes, cette classe de codeur n'est pas basée sur la DCT mais sur le codage DPCM suivi d'un codage entropique.

- **Le principe de compression de la norme JPEG non réversible**

La figure III.12 explique clairement les différentes étapes de l'algorithme de compression JPEG non réversible.

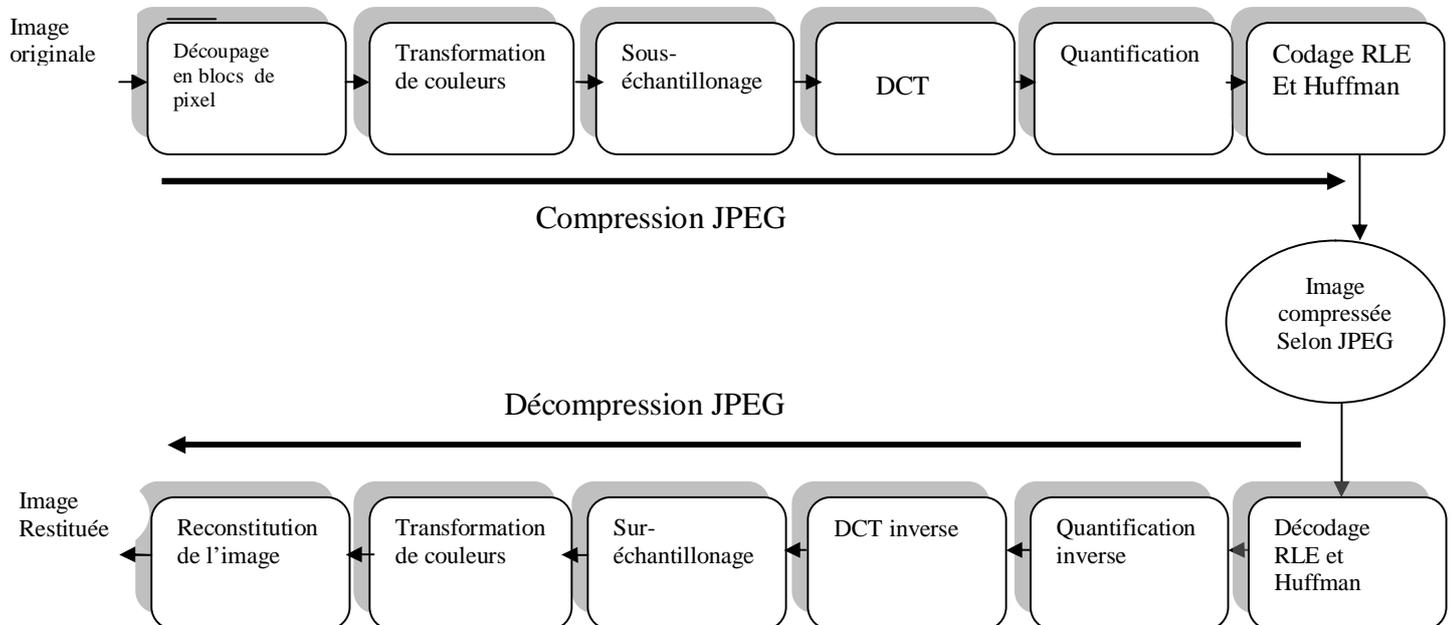


Figure III.12 : Principe de l'algorithme JPEG avec perte.

Cette compression se décompose en six étapes :

1. Découpage en blocs : le format JPEG, comme le font généralement les algorithmes de compression avec perte, commence par découper l'image en blocs ou carreaux généralement carrés de 64 (8 x 8) ou 256 (16 x 16) pixels.
2. Transformation de l'espace de couleurs : habituellement, les images couleurs sont stockées sous le format RGB.

Or le système RGB n'est pas le mieux adapté pour appliquer le codage JPEG. En effet, l'oeil humain est plus sensible à la luminance qu'à la chrominance d'une image. Ainsi, une transformation est réalisée pour passer à un espace couleur, plus adapté à la compression. C'est l'espace YCbCr qui a été choisi [26].

3. Sous échantillonnage : la façon la plus simple d'exploiter la faible sensibilité de l'œil à la chrominance est simplement de sous-échantillonner les signaux de chrominance. Généralement on utilise un sous-échantillonnage de type $2h1v$ ou $2h2v$. Dans le premier cas (le plus utilisé) on a un sous-échantillonnage 2:1 horizontalement et 1:1 verticalement, dans le deuxième cas on a un sous-échantillonnage 2:1 horizontalement et verticalement. Ces sous-échantillonnages sont utilisés pour les chrominances. Pour la luminance on n'utilise jamais de sous échantillonnage.

4. Transformée DCT : on applique la DCT aux blocs de 8x8 pixels.

Le calcul d'une DCT est complexe. C'est l'étape qui coûte le plus de temps et de ressources dans la compression et la décompression JPEG, mais c'est peut-être la plus importante car elle permet de séparer les basses fréquences et les hautes fréquences présentes dans l'image [5].

5. Seuillage et quantification des coefficients issus de la DCT : la quantification est la phase non conservatrice du processus de compression JPEG. Elle amène, moyennement à une diminution de la précision de l'image, mais aussi à réduire le nombre de bits nécessaires au stockage [16].

La quantification consiste à diviser cette matrice par une autre, appelée matrice de quantification, et qui contient 8x8 coefficients choisis par le codeur.

Habituellement, la matrice de quantification Q est obtenue à partir de formules plus ou moins simples, permettant de choisir la perte de qualité. La formule suivante est un exemple qui donne de bons résultats [16] :

$$Q = q(i, j) \text{ avec } q(i, j) = 1 + K(1 + m(i + j)) \quad (\text{III.14})$$

Avec i l'indice de ligne, j l'indice de colonne, m une constante (souvent égale à 1) et K le facteur de qualité (choisi entre 1 et 25).

La quantification est décrite par l'équation suivante :

$$\text{Valeur quantifiée}(i, j) = \frac{\text{valeur DCT}(i, j)}{Q(i, j)} \quad (\text{III.15})$$

Cette valeur est arrondie à l'entier le plus proche. Lors de la restitution de l'image (décompression), il suffira de réaliser l'opération inverse (déquantification) en multipliant chaque valeur de la matrice quantifiée par le quantum correspondant, pour retrouver une matrice DCT déquantifiée, à partir de laquelle sera établie la matrice des pixels de sortie.

Le but ici est d'atténuer les hautes fréquences, c'est-à-dire celles auxquelles l'œil humain est très peu sensible. Ces fréquences ont des amplitudes faibles, et elles sont encore plus atténuées par la quantification (les coefficients sont même ramenés à 0).

5. Compression RLE et Huffman : le codage s'effectue en zigzag comme indiqué à la figure III.13.

Après la phase de quantification un grand nombre de valeurs de la matrice résultante se trouvent réduites à 0. Le fait qu'il y ait beaucoup de valeurs mises à 0 va nous permettre de compresser encore mieux grâce à la compression RLE.

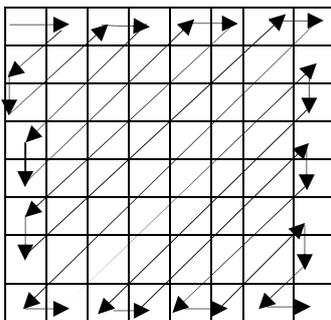


Figure III.13 : Le parcours d'un bloc en zigzag.

Un codage entropique des coefficients de la dernière matrice termine la chaîne de traitement. JPEG propose deux méthodes : le codage de Huffman, et le codage arithmétique.

La décompression JPEG comporte les mêmes étapes que la compression mais dans le sens inverse en commençant par la méthode de décodage statistique, puis la matrice obtenue est multipliée par la matrice de quantification que l'on reconstitue grâce au facteur de qualité et enfin on applique la DCT inverse (IDCT) pour retrouver une image plus ou moins dégradée par rapport à l'image initiale.

Pour plus d'informations sur le standard de compression JPEG vous pourrez consulter la référence [26]

- **Algorithme JPEG réversible :** La seconde classe, concerne les processus de codage sans pertes, cette classe de codeurs n'est pas basée sur la DCT mais sur le codage DPCM suivi d'un codage entropique (huffman ou arithmétique). le traitement se fait sur l'image entière, et non pas sur des blocs [5].

Cet algorithme sert surtout pour la transmission d'images médicales pour éviter de confondre des artefacts avec de réels signes pathologiques. La compression est alors beaucoup moins efficace (facteur 2 seulement).



(a) image originale



(b) image JPEG
Qualité=32% et $T_c = 40.33:1$



(c) image JPEG
Qualité=5% et $T_c = 106.53:1$

Figure III.14 : Exemple d'image compressée avec JPEG.

La dégradation de l'image avec la compression JPEG lorsque le taux augmente.

III.8.2. La norme de compression JPEG 2000:

C'est une norme de compression d'images produite par le groupe de travail *Joint Photographic Experts Group*. JPEG 2000 est capable de travailler avec ou sans perte, utilisant une transformation en ondelettes.

En compression irréversible, JPEG 2000 est plus performante que la méthode de compression JPEG ISO/CEI 10918-1 (JPEG baseline). On obtient donc des fichiers d'un poids inférieur pour une qualité d'image égale. De plus, les contours nets et contrastés sont mieux rendus [26].

Ce nouveau standard a pour objectif d'offrir de nouvelles fonctionnalités permettant de répondre à une demande croissante, à savoir [5] [26] :

- Obtenir des performances de compression supérieures à son prédécesseur JPEG, notamment pour des débits très faibles.
- Permettre d'organiser le fichier compressé de plusieurs manières, notamment en fonction de la résolution désirée ou de la qualité de reconstruction.
- D'autres fonctionnalités les plus marquantes de cette norme sont, d'une part, la possibilité de compresser sans perte ou avec perte en utilisant le même algorithme.
- Permet de ne décoder uniquement qu'une certaine zone d'une image lorsque celle-ci est très grande. Elle permet encore la plupart des manipulations géométriques élémentaires (flip, permutation paysage/portrait...) directement dans le domaine compressé et sans aucune perte en qualité.
- Permet une affectation non uniforme du budget de qualité dans une image.
- En JPEG 2000 le même algorithme de codage s'applique à une grande variété de types d'images (couleurs, niveaux de gris, multi-composantes...).

- **Principe général**

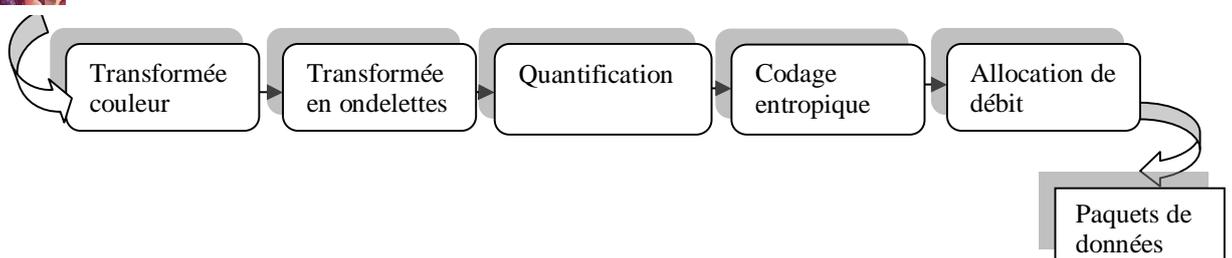


Figure III.15 : Schéma typique d'un codeur JPEG 2000.

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 [26] 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* JPEG 2000, avec la formation des paquets, puis la syntaxe haut niveau, particulièrement abondante dans JPEG 2000.

Dans la norme JPEG 2000 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 tuiles, en-tête principale. Les métadonnées du format de fichier JP2 ne font pas partie du *codestream*. JP2 encapsule le *codestream* JPEG 2000 dans un format de fichier [33].

III.9 Particularité des images couleurs

Dans ce qui suit, nous proposons un algorithme de compression se basant sur une idée similaire à celle de la norme JPEG, mais dont les différences majeures résident en :

- Un changement d'espace de couleurs (RGB-YIQ) au lieu de (RGB-YCbCr).
- L'utilisation d'un autre quantificateur [16].
- L'utilisation de l'encodeur entropique TRE [32].

Le schéma général de cette compression est présenté dans la figure III.16, et bien détaillé dans le chapitre suivant.

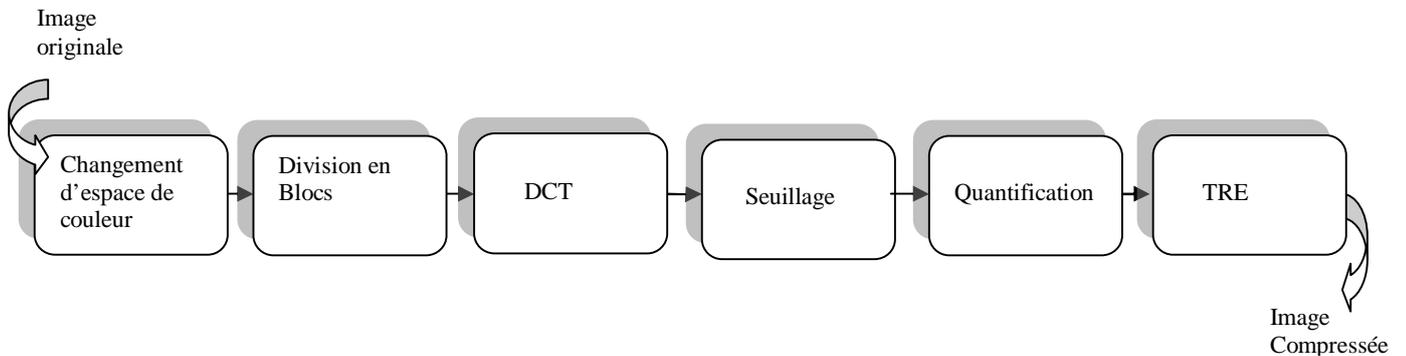


Figure III.16 : Schéma du principe de la compression.

III.10 Conclusion

Dans ce chapitre, nous avons présenté plusieurs techniques de compression d'images fixes sans et avec perte (EZW, SPIHT, JPEG, JPEG 2000, ...). Tout d'abord, nous avons donné quelques concepts et mesures pour la compression d'images. Des recherches très actives se poursuivent évidemment dans le domaine des images animées et du son.

Dans le chapitre suivant, l'algorithme proposé sera présenté en détails.

Chapitre IV

Méthodologie proposée et résultats de simulation

IV.1 Introduction

Le codage JPEG est aujourd'hui largement utilisé dans les secteurs de l'informatique et de la communication (appareils photo numériques, scanners, imprimantes, télécopieurs...).

Dans ce contexte, et afin d'optimiser l'espace de stockage dans de tels appareils, nous allons proposer dans ce chapitre une méthode de compression d'images similaire à JPEG mais au lieu de changer l'espace RGB vers YCbCr dans l'étape transformation de l'espace de couleurs, nous choisissons plusieurs espaces de couleurs par exemple (YCbCr, YIQ, YUV, YDbDr, HSV....), ceci est afin d'effectuer une étude comparative pour trouver l'espace de couleurs le plus adapté à la compression des images (objectif de notre travail).

Autres différences par rapport à JPEG sont : l'utilisation d'un nouveau quantificateur ainsi que l'usage d'un autre codage entropique appelé TRE (Two-Role Encoder). Les résultats obtenus ainsi que leurs discussions seront présentés. Pour valider ce travail, nous allons exposer nos tests accomplis sur différents types d'images connues tout en présentant une étude comparative avec les méthodes CBTC-BF [17] et DCT-Adaptive Scanning [16].

IV.2 Présentation de notre méthode

Dans cette section, nous présentons notre approche de compression et décompression d'images. La figure IV.1 résume la structure de cet algorithme, les images couleurs comprennent trois plans qui subissent chacun le même procédé, avec un changement d'espace de couleurs RGB vers un autre espace, tels que YUV, YIQ, YCbCr, YDbDr, YPbPr,.... Ceci est dans l'optique de trouver l'espace de couleurs le plus adapté à la compression des images.

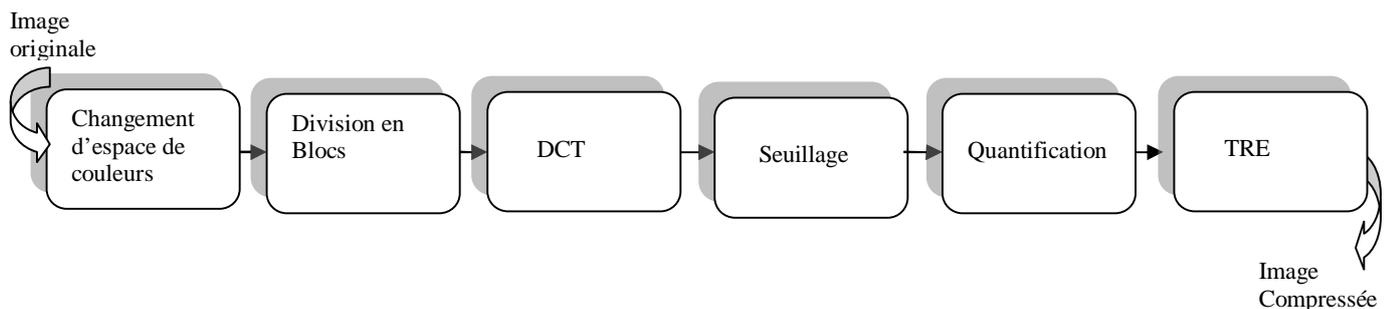


Figure IV.1 : Schéma du principe de la compression.

La décompression de l'image effectue les traitements dans l'ordre inverse (Figure IV.2).

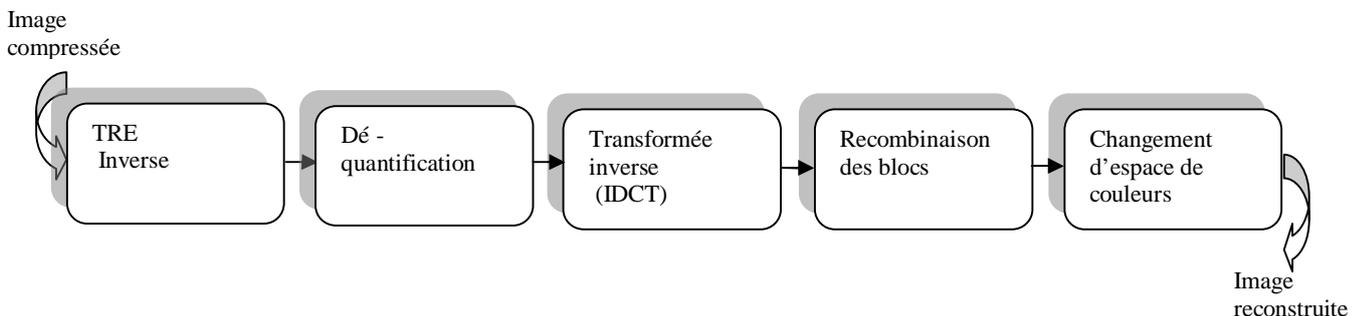


Figure IV.2 : Schéma de décompression.

(a) La phase de compression

- **Changement d'espace de couleurs** : habituellement, les images couleurs sont stockées sous le format RGB [33]. Or, le système RGB n'est pas le mieux adapté pour appliquer le processus de compression car il y a une forte corrélation entre les plans RGB. Mais, avec un changement d'espace de couleurs RGB vers un autre moins corrélé (par exemple YUV), cette transformation permet une compression plus efficace car l'information est

principalement condensée dans le plan Y, on peut donc comprimer plus efficacement U et V.

Dans cette étape, nous choisissons plusieurs espaces de couleurs, par exemple YCbCr, YIQ, YUV, YDbDr, YPbPr, O1O2O3, NTSC, HSV, CMY, XYZ, Lab dans le but de trouver l'espace de couleurs le plus adéquat à la compression.

- **Division en blocs** : consiste à diviser l'image en blocs sur lesquels sont appliquées indépendamment les étapes suivantes. La principale raison de ce découpage est de limiter le nombre de pixels à traiter à la fois pour diminuer le temps de calcul [16]. Il peut résulter de cette division un effet visuel appelé effet de blocs à des taux de compression élevés, les frontières des blocs deviennent visibles car ils ont été comprimés indépendamment. La taille des blocs est variable selon les méthodes. Dans notre cas, nous avons employé des blocs de taille 8x8, 16x16 et 32x32.
- **L'application de la DCT** : nous calculons la DCT [34] de chaque bloc, ce qui permet de transformer les pixels d'un bloc 8x8 (16x16 et 32x32) d'une image en un autre bloc de 8x8 (16x16 et 32x32) contenant les composantes fréquentielles correspondantes.

Le but de cette transformation est de décorréler les pixels, ce qui a pour effet en général de redistribuer l'énergie de l'image dans un nombre restreint de coefficients transformés [16]. De cette façon, un grand nombre de coefficients transformés ont de très faibles valeurs, et peuvent être supprimés ou se voir alloués un nombre très faible de bits lors de l'étape de quantification.

- **Seuillage** : Mettre à zéros tous les coefficients inférieurs ou égaux à TH. Dans cette étape nous utilisons l'algorithme bisection [32] pour contrôler à l'avance le PSNR (UPSNR) avec une précision de convergence ϵ . Celle-ci est détaillée un peu plus loin.
- **Quantification** : les coefficients non nuls (NZDCTC) sont en suite quantifiés par un quantificateur linéaire de Q bits, l'objectif de cette quantification est de réduire le nombre de bits nécessaires à la représentation des coefficients de façon que cette réduction n'apporte pas de dégradation visuelle notable à l'image.

Notons que les hautes fréquences, qui sont généralement de faibles amplitudes et auxquelles l'œil n'est que peu sensible, sont généralement éliminées par ce processus.

La quantification des coefficients NZDCTC est effectuée selon la formule suivante [16] :

$$QNZDCTC = \left\lfloor 1 + \frac{NZDCTC - DCT_{Min}}{DCT_{Max} - DCT_{Min}} \times (2^Q - 2) \right\rfloor \quad (IV.1)$$

Où : $\lfloor \cdot \rfloor$ représente la valeur entière inférieure la plus proche.

DCT_{Max} : valeur maximale de NZDCTC.

DCT_{Min} : valeur minimale de NZDCTC.

Q : nombre de bits de quantification.

Dans cette étape chaque NZDCTC est quantifié pour devenir un élément de QNZDCTC avec la plus basse résolution possible (un nombre de Q bits est alloué pour représenter chaque QNZDCTC). Il est facile de remarquer que le QNZDCTC minimum est codé par 1 et le maximum est codé par $2^Q - 1$.

- **Two-Role Encoder (TRE)** : Le codage s'effectue en zigzag en adoptant le codeur sans perte TRE [32]. Il se fait de la manière suivante :

Les coefficients quantifiés non nuls (QNZDCTC) sont codés chacun par un nombre entier non négatif de largeur égale (Q+1) bits. Selon la formule (IV.1), il est tout à fait facile de remarquer que le QNZDCTC minimum prend la valeur 1, cependant le QNZDCTC maximum est de valeur $2^Q - 1$.

Nous ajoutons 2^Q à chaque QNZDCTC différent de 1; et $2^Q - 1$ à chaque QNZDCTC égal à 1. Donc chaque QNZDCTC est codé par un code TRE de (Q+1) bits dans l'intervalle

$[2^Q - 2^{Q+1} - 1]$.

Il est à noter que l'étape de seuillage produit beaucoup de séries de zéros. Chacune est remplacée par un seul code TRE de $(Q+1)$ bits (entier codé sur $(Q+1)$ bits) et prend une valeur dans l'intervalle $[1 - 2^Q - 1]$.

(b) la phase de décompression

Le processus de décompression est tout à fait facile (figure IV.2).

On applique en premier lieu le décodage TRE qui est très facile, pour chaque code TRE, nous avons :

- Si $TRE > 2^Q$ alors TRE est un QNZDCTC. Donc, la soustraction de la quantité 2^Q mène à la vraie valeur de QNZDCTC.
- Si $TRE = 2^Q$, alors soustraire $2^Q - 1$. On obtient, ainsi, un QNZDCTC égale à 1.
- Si $TRE \leq 2^Q - 1$, alors le décodeur produit une succession de zéros égale à la valeur contenue dans ce code TRE.

A ce stade, l'image reconstruite sera obtenue en effectuant la déquantification (formule IV.2), la transformée en cosinus discret inverse (IDCT), recombinaison des blocs et enfin le changement inverse de l'espace de couleurs.

$$NZDCTC = \left\lfloor \frac{(QNZDCTC - 1) \times (DCT_{Max} \times DCT_{Min})}{2^Q - 2} \right\rfloor + DCT_{Min} \tag{IV.2}$$

Nous avons inclus dans l'algorithme un contrôle de qualité. Donc l'utilisateur peut exiger d'avance la qualité à obtenir. Ceci, est garanti par la méthode de recherche communément appelée la bisection [32].

L'algorithme bisection :

1. Initialisation :

- Fixer en avance une valeur désirée dite UPSNR (User PSNR) à atteindre.
- Sélectionner l'intervalle de recherche $[TH_{Min}, TH_{Max}]$, où l'intervalle peut être initialisé par $[0, DCT_{Max}]$. Notons que DCT_{Max} est le maximum en valeur absolue des coefficients DCT.
- Donner la précision de convergence ϵ .

2. Calcul de $TH = \frac{TH_{Max} + TH_{Min}}{2}$

3. Seuillage : Mettre à zéros tous les coefficients inférieurs ou égaux à TH.

4. Quantification/Déquantification : quantifier les coefficients non nuls (NZDCTC) par un quantificateur linéaire de taille Q bits (formule IV.1). La déquantification se fait dans l'ordre inverse en utilisant l'équation IV.2.

5. IDCT : application inverse de la DCT.

6. Calcul du PSNR en fonction de TH : calculer le PSNR de l'image seuillée pour la valeur courante de TH.

7. Mise à jour du coefficient de TH :

si $(PSNR < UPSNR)$ alors $TH_{Min} = TH$,
sinon $TH_{Max} = TH$.

8. Condition de terminaison

Si $|PSNR - UPSNR| > \epsilon$ aller à l'étape 2, sinon STOP.

IV.3 Résultats de la simulation

Dans cette partie, nous présentons les résultats expérimentaux obtenus sur plusieurs images de différents types. Ces résultats sont évalués en utilisant les deux mesures (PSNR et bpp). Pour évaluer les performances de notre méthode, l'algorithme a été appliqué sur 30 images couleurs (RGB) de 24 bits et de différentes tailles par exemple : Airplane, Peppers, F15 et Lena de 512×512 ; Barche, Girl, Couple, House et Zelda de 256×256 ; Terraux de 768x512 et Bandon de 610x403. Elles sont présentées sur la figure IV.3



Airplane



Peppers



Lena



F15



jerusalem



Girl



Couple



House



Zelda



Balloons



Bandon



Terraux

Figure IV.3 : Les différentes images originales.

Le tableau IV.1 compare l'effet du changement de l'espace de couleurs et la taille de bloc sur les valeurs concernant le bpp et le PSNR.

Image	Espace de couleurs	PSNR	Taille de bloc=16x16	Taille de bloc = 32x32	Taille de bloc=8x8
			bpp	bpp	bpp
Peppers (512x512)	RGB	30.15	1.16	1.45	1.34
	YUV		1.08	1.32	1.30
	YIQ		1.12	1.38	1.33
	YCBCR		1.10	1.34	1.33
	YPbPr		1.08	1.32	1.29
	YDbDr		1.34	1.72	1.47
	O1O2O3		1.53	2.00	1.60
	HSV		6.96	9.97	5.46
	XYZ		1.50	1.94	1.64
	LAB		2.26	3.07	2.20
Lena (512x512)	RGB	31.93	1.49	1.76	1.65
	YUV		1.05	0.69	1.32
	YIQ		1.04	1.16	1.31
	YCBCR		1.09	1.23	1.37
	YPbPr		1.07	1.20	1.32
	YDbDr		1.35	1.56	1.54
	O1O2O3		1.34	1.56	1.53
	HSV		8.98	12.74	7.53
	XYZ		1.80	2.12	2.02
	LAB		1.77	2.08	2.00
Airplane (512x512)	RGB	30.36	1.18	1.35	1.38
	YUV		0.82	0.85	1.03
	YIQ		0.84	0.91	1.04
	YCBCR		0.84	0.90	1.16
	YPbPr		0.83	0.90	1.13
	YDbDr		0.94	1.05	1.04
	O1O2O3		0.99	1.11	1.18
	HSV		2.64	3.70	2.30
	XYZ		1.80	2.12	2.02
	LAB		1.36	2.08	2.00
House (256x256)	RGB	31.79	1.50	1.83	1.58
	YUV		1.18	1.39	1.38
	YIQ		1.14	1.35	1.34
	YCBCR		1.20	1.43	1.39
	YPbPr		1.46	1.78	1.56
	YDbDr		1.14	1.35	1.34
	O1O2O3		1.37	1.68	1.50
	HSV		6.45	9.08	4.67
	XYZ		1.90	2.35	1.93
	LAB		1.71	2.09	1.80
Girl (256x256)	RGB	35.13	1.22	1.77	1.38
	YUV		0.69	0.87	0.98
	YIQ		0.75	0.96	1.06
	YCBCR		0.70	0.90	0.97
	YPbPr		0.81	1.04	1.03
	YDbDr		0.70	0.89	0.99
	O1O2O3		0.81	1.04	1.06
	HSV		2.87	5.02	2.17

	XYZ		1.41	2.06	1.60
	LAB		1.04	1.43	1.26
Zelda (256x256)	RGB	31.31	1.70	2.12	1.75
	YUV		1.18	1.42	1.35
	YIQ		1.19	1.48	1.40
	YCBCR		1.23	1.48	1.42
	YPbPr		1.49	1.85	1.60
	YDbDr		1.19	1.42	1.35
	O1O2O3		1.58	2.00	1.68
	HSV		11.64	14.84	8.76
	XYZ		2.12	2.65	2.12
	LAB		1.88	2.34	1.92
Couple (256x256)	RGB	32.44	1.26	1.65	1.37
	YUV		1.24	1.61	1.38
	YIQ		1.31	1.69	1.46
	YCBCR		1.34	1.75	1.43
	YPbPr		1.62	2.20	1.62
	YDbDr		1.24	1.61	1.38
	O1O2O3		1.54	2.08	1.58
	HSV		10.95	15.05	8.08
	XYZ		2.43	3.30	2.32
	LAB		3.77	2.95	2.18
Lighthouse (512x512)	RGB	35.49	2.07	1.49	1.41
	YUV		2.03	1.47	1.39
	YIQ		2.30	1.66	1.76
	YCBCR		2.10	1.52	1.44
	YPbPr		2.67	2.03	1.69
	YDbDr		2.03	1.47	1.39
	O1O2O3		3.06	2.35	1.93
	HSV		6.59	5.47	3.28
	XYZ		4.33	3.40	3.00
	LAB		0.95	0.58	0.95
F15 (732x500)	RGB	34.46	0.84	0.47	0.91
	YUV		0.77	0.41	0.88
	YIQ		0.93	0.56	0.97
	YCBCR		0.87	0.48	0.92
	YPbPr		0.94	0.55	0.94
	YDbDr		0.77	0.41	0.88
	O1O2O3		0.88	0.58	0.86
	HSV		1.04	0.66	1.03
	XYZ		1.15	0.82	1.15
	LAB		2.35	2.71	2.29
Athens (512x512)	RGB	31.20	1.03	1.08	1.30
	YUV		1.02	1.09	1.23
	YIQ		1.07	1.13	1.34
	YCBCR		1.03	1.08	1.28
	YPbPr		1.12	1.22	1.36
	YDbDr		1.02	1.09	1.23
	O1O2O3		5.54	7.19	4.27
	HSV		2.62	3.04	2.59
	XYZ		1.40	1.55	1.61
	LAB		2.42	1.76	1.54

Blueeye (512x512)	RGB	32.84	1.62	1.06	1.24
	YUV		1.65	1.09	1.25
	YIQ		1.70	1.13	1.32
	YCBCR		1.69	1.14	1.29
	YPbPr		2.18	1.53	1.49
	YDbDr		2.57	1.88	1.66
	O1O2O3		7.36	6.36	3.47
	HSV		2.93	2.16	2.01
	XYZ		12.10	9.52	6.14
	LAB		1.62	1.04	1.22
Beeflower (512x512)	RGB	31.64	1.58	1.01	1.18
	YUV		1.58	1.01	1.18
	YIQ		1.64	1.06	1.23
	YCBCR		1.67	1.08	1.21
	YPbPr		1.58	1.01	1.18
	YDbDr		1.92	1.30	1.33
	O1O2O3		4.64	3.78	1.38
	HSV		1.93	1.30	1.40
	XYZ		2.07	1.41	1.44
	LAB		2.87	3.41	2.73
Butterfly (512x512)	RGB	30.76	1.42	1.61	1.56
	YUV		1.42	1.62	1.57
	YIQ		1.48	1.65	1.66
	YCBCR		1.78	2.08	1.89
	YPbPr		1.42	1.62	1.57
	YDbDr		1.97	2.29	2.01
	O1O2O3		4.10	5.45	3.47
	HSV		3.39	4.01	3.53
	XYZ		3.34	3.95	3.16
	LAB		1.84	2.32	1.85
Jelly beans (256x256)	RGB	34.12	1.60	1.96	1.68
	YUV		1.65	2.01	1.77
	YIQ		1.58	1.95	1.72
	YCBCR		1.95	2.44	1.93
	YPbPr		1.65	2.01	1.77
	YDbDr		2.04	2.55	2.07
	O1O2O3		4.42	6.92	3.30
	HSV		2.33	2.92	2.30
	XYZ		2.42	3.15	2.42
	LAB		2.05	2.32	2.19
Elephant (512x512)	RGB	30.80	1.31	1.42	1.60
	YUV		1.38	1.50	1.71
	YIQ		1.36	1.48	1.60
	YCBCR		1.93	2.13	2.13
	YPbPr		1.30	1.40	1.57
	YDbDr		2.37	2.70	2.56
	O1O2O3		5.76	7.98	4.92
	HSV		2.56	2.84	3.33
	XYZ		4.93	6.17	4.73
	LAB		3.26	3.93	3.15
House	RGB		2.38	2.77	2.49
	YUV		2.48	2.91	2.59

(512x512)	YIQ	31.80	2.55	3.01	2.60
	YCBCR		3.14	3.79	3.05
	YPbPr		2.38	2.81	2.49
	YDbDr		3.11	3.77	3.02
	O1O2O3		5.30	7.04	4.44
	HSV		4.08	4.79	4.33
	XYZ		3.75	4.51	3.84
	LAB		1.63	1.63	1.80
Jerusalem (512x512)	RGB	33.25	1.58	1.58	1.73
	YUV		1.81	1.81	2.04
	YIQ		1.66	1.66	1.83
	YCBCR		2.41	2.41	2.42
	YPbPr		1.58	1.58	1.73
	YDbDr		2.32	2.32	2.40
	O1O2O3		7.52	7.52	6.10
	HSV		3.98	3.98	5.23
	XYZ		6.91	6.91	5.58
	LAB		3.29	3.29	2.96
Monolake (512x512)	RGB	31.12	3.25	3.25	2.92
	YUV		3.37	3.37	3.19
	YIQ		3.46	3.46	3.09
	YCBCR		4.48	4.48	3.80
	YPbPr		3.25	3.25	2.95
	YDbDr		4.01	4.01	3.58
	O1O2O3		8.07	8.07	6.50
	HSV		5.85	5.85	5.38
	XYZ		4.44	5.11	4.34
	LAB		3.67	4.09	4.00
Lochness (814x559)	RGB	34.20	3.68	4.10	4.00
	YUV		4.04	4.51	5.10
	YIQ		3.92	4.34	4.14
	YCBCR		5.65	6.43	5.77
	YPbPr		3.68	4.19	4.00
	YDbDr		6.25	7.22	6.13
	O1O2O3		6.74	6.86	4.75
	HSV		7.21	8.17	5.12
	XYZ		1.33	1.45	1.52
LAB	0.93	0.97	1.21		
Desert (512x512)	RGB	32.75	0.86	0.87	1.16
	YUV		0.98	1.03	1.27
	YIQ		0.98	1.00	1.22
	YCBCR		1.26	1.36	1.46
	YPbPr		0.86	0.87	1.16
	YDbDr		5.37	7.39	4.12
	O1O2O3		1.61	1.83	1.75
	HSV		2.25	2.79	2.14
	XYZ		0.54	0.48	0.89
	LAB		0.67	6.62	1.02
Bandon (512x512)	RGB		0.62	0.58	0.97
	YUV		0.79	0.74	1.20
	YIQ		0.69	0.65	1.03
	YCBCR		0.79	0.80	1.13

	YPbPr	30.03	0.76	0.76	1.09
	YDbDr		8.50	12.52	6.08
	O1O2O3		0.67	0.64	1.02
	HSV		1.51	1.74	1.74
	XYZ		4.76	6.23	4.31
	LAB		2.79	3.44	2.89
Pelican (512x512)	RGB	33.20	2.65	3.27	2.74
	YUV		3.02	3.63	4.65
	YIQ		3.10	3.80	3.16
	YCBCR		2.71	3.38	2.79
	YPbPr		3.42	4.28	3.46
	YDbDr		7.80	10.93	6.13
	O1O2O3		5.76	7.24	2.05
	HSV		5.40	6.69	2.16
	XYZ		3.20	4.01	3.06
	LAB		1.35	1.60	1.54
Terraux (768x512)	RGB	35.12	1.33	1.57	1.51
	YUV		1.44	1.70	1.77
	YIQ		1.48	1.76	1.66
	YCBCR		1.33	1.57	1.51
	YPbPr		1.39	1.65	1.58
	YDbDr		9.57	12.85	7.15
	O1O2O3		3.45	4.26	1.85
	HSV		1.96	2.28	2.13
	XYZ		2.66	3.00	2.74
	LAB		1.43	1.56	1.68
Opunphare (909x614)	RGB	31.50	1.43	1.53	1.67
	YUV		1.44	1.56	1.67
	YIQ		1.69	1.86	1.82
	YCBCR		1.43	1.53	1.72
	YPbPr		1.99	2.21	1.91
	YDbDr		2.66	3.21	2.20
	O1O2O3		3.06	3.40	1.98
	HSV		2.34	2.56	1.93
	XYZ		0.72	0.75	1.05
	LAB		0.42	0.37	0.77
Kid (487x703)	RGB	30.00	0.54	0.49	0.94
	YUV		0.43	0.38	0.78
	YIQ		0.47	0.43	0.82
	YCBCR		0.42	0.37	0.76
	YPbPr		0.44	0.40	0.79
	YDbDr		0.85	0.89	1.12
	O1O2O3		0.79	0.84	1.12
	HSV		0.63	0.63	0.96
	XYZ		1.88	2.24	2.14
	LAB		2.13	2.47	2.82
Brandy rose (518x744)	RGB	32.74	2.17	2.46	2.7
	YUV		2.41	2.77	3.08
	YIQ		2.59	3.10	2.97
	YCBCR		2.13	2.47	2.82
	YPbPr		4.01	4.74	3.97
	YDbDr		5.91	8.38	4.66

	O1O2O3		2.54	2.96	4.10
	HSV		5.10	6.27	4.88
	XYZ		1.99	2.39	2.04
	LAB		1.94	2.37	2.00
Opera (695x586)	RGB	34.25	2.20	2.65	2.40
	YUV		2.00	2.43	2.04
	YIQ		2.14	2.62	2.16
	YCBCR		1.94	2.37	2.03
	YPbPr		2.02	2.48	2.07
	YDbDr		6.68	9.53	4.91
	O1O2O3		5.55	6.52	2.13
	HSV		0.81	0.95	1.05
	XYZ		0.71	0.81	1.00
	LAB		0.71	0.81	0.98
Balloons (256x256)	RGB	30.68	0.72	0.82	1.01
	YUV		0.76	0.86	1.02
	YIQ		0.89	1.05	1.09
	YCBCR		0.71	0.81	0.99
	YPbPr		0.89	1.02	1.14
	YDbDr		5.69	8.07	4.30
	O1O2O3		1.03	1.25	1.22
	HSV		1.29	1.54	1.52
	XYZ		0.64	0.58	1.01
	LAB		0.65	0.60	1.01
Splash (512x512)	RGB	32.76	0.65	0.62	1.02
	YUV		0.66	0.62	1.04
	YIQ		0.64	0.60	1.00
	YCBCR		0.74	0.76	1.08
	YPbPr		0.65	0.62	1.02
	YDbDr		0.78	0.80	1.09
	O1O2O3		6.06	8.04	2.30
	HSV		0.79	0.80	1.89
	XYZ		1.04	1.13	1.73
	LAB		1.76	2.14	1.92
Girl (256x256)	RGB	33.26	1.75	2.14	1.92
	YUV		1.34	1.56	1.53
	YIQ		1.34	1.58	1.54
	YCBCR		1.39	1.63	1.62
	YPbPr		1.38	1.61	1.57
	YDbDr		1.61	1.90	1.73
	O1O2O3		1.64	1.99	1.77
	HSV		4.33	6.48	3.37
	XYZ		2.07	2.49	2.27
	LAB		2.21	2.70	2.39

Tableau IV.1 : Résultats obtenus du PSNR et du bpp pour les différentes images.

Le tableau IV.2 résume les résultats en valeurs moyennes.

Espace de couleurs	PSNR moyenne	Taille de bloc=16x16	Taille de bloc = 32x32	Taille de bloc=8x8
		bpp moyenne	bpp moyenne	bpp moyenne
RGB	32.37	2.29	2.57	2.2
YUV		1.46	1.76	1.62
YIQ		1.45	1.55	1.60
YCBCR		1.56	1.67	1.81
YPbPr		1.53	1.63	1.65
YDbDr		1.90	2.08	1.93
O1O2O3		1.97	2.17	1.97
HSV		6.12	7.88	4.5
XYZ		2.80	3.08	2030
LAB		3.03	3.31	2.48

Tableau IV.2 : Les résultats moyens.

D'après ces tests on peut donner les remarques suivantes :

- La meilleure performance peut être obtenue avec l'espace de couleurs YIQ comparé aux autres espaces de couleurs.
- Les tailles de bloc [16x16] et [32x32] sont les plus adaptées à cet algorithme.

La figure IV.3 illustre les bpp et les PSNR obtenus pour les trois tailles de découpage 8x8, 16x16 et 32x32, dans l'espace de couleurs YIQ.



PSNR=30.76 bpp=1.00



PSNR= 31.12 bpp= 1.07



PSNR= 34.78 bpp= 0.98



PSNR= 35.84 bpp= 1.10

(a)

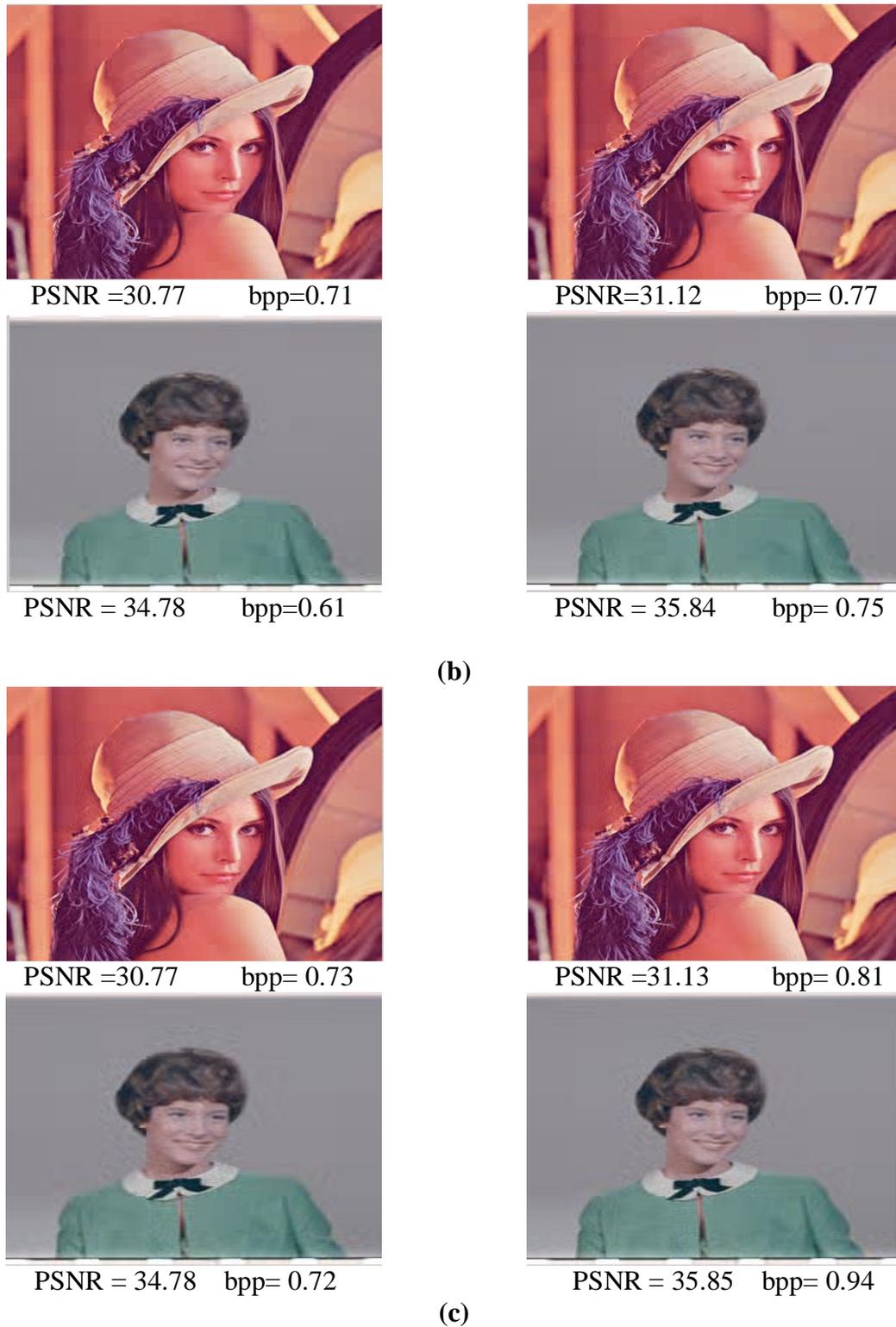


Figure IV.4 : Images reconstruites (Lena, Girl) pour différentes tailles de bloc : (a) 8x8, (b) 16x16, (c) 32x32.

La figure IV.4 illustre les bpp et les *PSNR* obtenus dans les deux espaces RGB et YIQ, avec la taille de bloc 16x16 (Q=8).

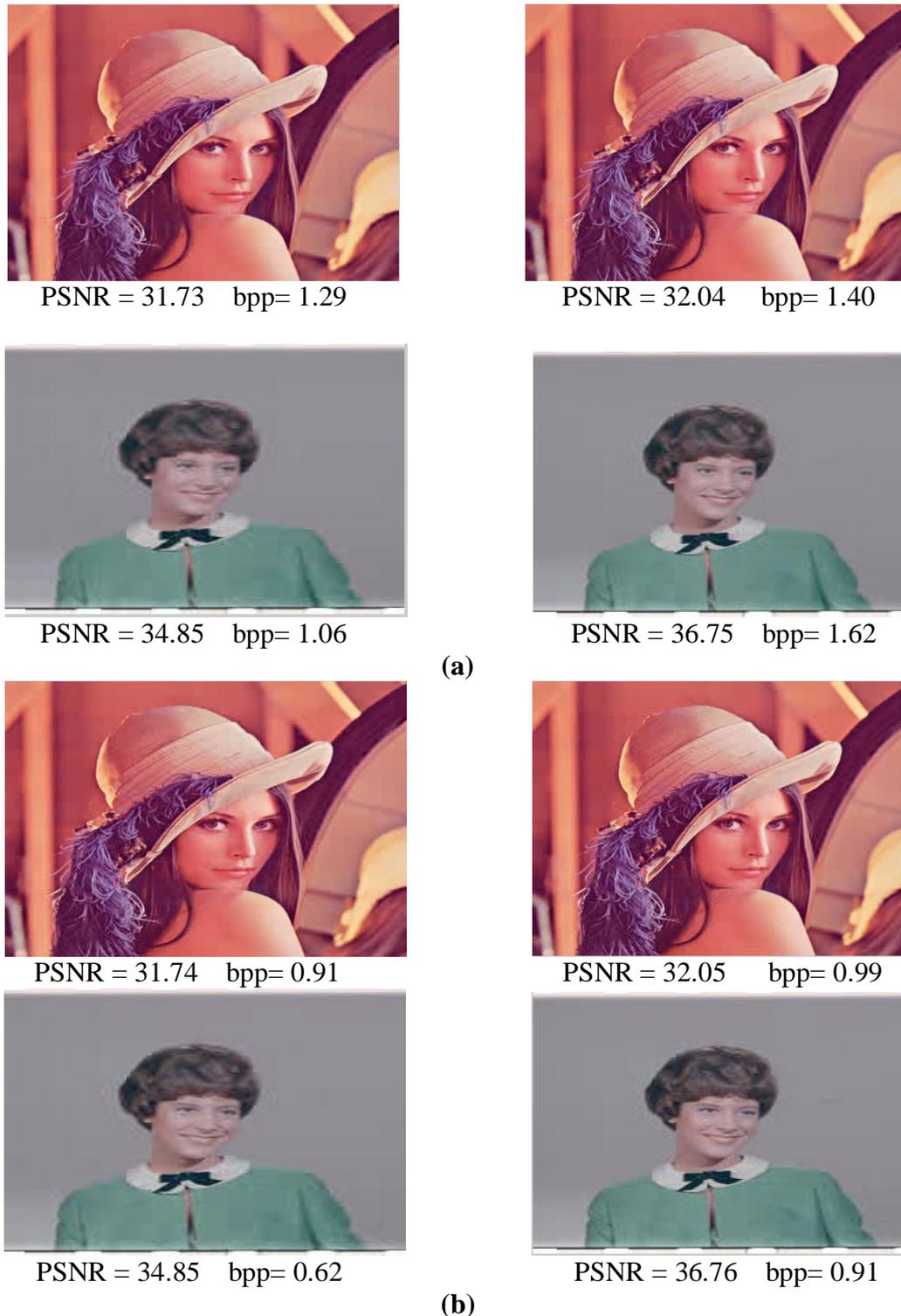


Figure IV.5 : Images reconstruites (Lena, Girl) dans les deux espaces : (a) RGB, (b) YIQ.

IV.4 Etude comparative

Dans cette section, nous comparons les performances de la méthode proposée avec d'autres méthodes de compression d'images : CBTC-BF présentée dans l'article [17] et la méthode DCT-Adaptive Scanning présentée dans la thèse [16].

La comparaison avec la méthode DCT-Adaptive Scanning est très intéressante puisque cette méthode utilise, comme la méthode proposée, la transformation DCT.

Ces comparaisons sont illustrées dans le tableau IV.3 et le tableau IV.4 respectivement.

Images	La méthode proposée		CBTC-PF	
	PSNR	bpp	PSNR	bpp
<i>taille de l'image 512 x512</i>				
Airplane	30.5886	0.8268	30.36	1.04
Peppers	30.9429	1.3296	30.15	1.50
Lena	31.9450	0.9614	31.93	1.17
<i>taille de l'image 256x256</i>				
Girl	35.1352	0.6557	35.13	0.60
Couple	32.6820	1.2225	32.44	1.00
House	31.8647	1.0718	31.79	1.20
Zelda	31.4251	1.1289	31.31	1.12
Moyenne	32.0834	1.0281	31.87	1.09

Tableau IV.3: Comparaison de nos résultats à ceux de la CBTC-PF.

Les résultats reportés dans le tableau ci-dessus montre clairement que ceux de notre approche sont particulièrement comparables à ceux de CBTC-PF.

Le tableau IV.4 donne les résultats des *PSNR* et *bpp* obtenus par la méthode DCT-Adaptive Scanning présentée dans [16] face à ceux de la méthode proposée.

Images	La méthode DCT-Adaptive scanning				La méthode proposée					
	RGB		YCbCr		RGB		YCbCr		YIQ	
	PSNR	bpp	PSNR	bpp	PSNR	bpp	PSNR	bpp	PSNR	bpp
Airplane	30.6064	1.1870	30.5854	0.7381	30.5837	1.1170	30.5903	0.8084	30.5742	0.8248
Peppers	30.1580	1.4020	30.9369	1.4685	30.9374	1.3409	30.9289	1.3191	30.9307	1.3246
Lena	31.8981	1.5496	31.9513	1.0054	31.9563	1.3737	31.9479	1.0279	31.9450	0.9614
Girl	34.9971	1.4784	35.1323	0.6914	35.1364	1.1247	35.1404	0.7175	35.1352	0.6557
Couple	32.4386	2.3673	32.3664	1.3588	32.3680	1.8208	32.3597	1.1859	32.3601	1.1258
House	31.7871	1.5559	31.8760	1.0879	31.8701	1.3956	31.8790	1.1461	31.8647	1.0718
Zelda	31.3051	1.8684	31.1993	1.1250	31.3088	1.5519	31.3123	1.1024	31.3081	1.0964
Moyenne	31.8843	1.6298	32.0068	1.0679	32.0230	1.3892	32.0226	1.0439	32.0169	1.0086

Tableau IV.4 : Comparaison des résultats avec la méthode DCT-Adaptive Scanning.

De même que le tableau IV.3, les résultats de notre approche sont légèrement meilleurs que ceux de DCT-Adaptive Scanning [16].



Peppers
 $PSNR = 30.9289$ db
 $bpp = 1.3191$



Airplane
 $PSNR = 30.5903$ db
 $bpp = 0.8084$



Lena
 $PSNR = 31.9479$ db
 $bpp = 1.0279$

(a)



Peppers
 $PSNR = 30.9429$ db
 $bpp = 1.3296$



Airplane
 $PSNR = 30.5886$ db
 $bpp = 0.8268$



Lena
 $PSNR = 31.9450$ db
 $bpp = 0.9614$

(b)

Figure IV.6 : Résultats d'application de notre méthode sur les images " Peppers ", "Airplane" et "Lena" dans (a) l'espace YCbCr, (b) l'espace YIQ.

D'après le tableau IV.4 et la figure IV.6, on observe que la méthode proposée permet d'obtenir de meilleurs résultats que ceux de la méthode DCT-Adaptive Scanning dans les trois espaces de couleurs (RGB, YCbCr et YIQ), mais les résultats sont mieux dans l'espace YIQ. La comparaison des $PSNR$ obtenus ainsi que les qualités visuelles des images compressées démontrent l'efficacité de la méthode proposée.

Les différentes images reconstruites par la méthode proposée sont illustrées par la figure suivante :



Airplane
 PSNR = 30.75
 bpp= 0.85



Peppers
 PSNR = 31.19
 bpp=1.45



F15
 PSNR =34.46
 bpp=0.42



lena
 PSNR = 32.14
 bpp= 1.01



Girl
 PSNR =35.19
 bpp= 0.66



Boat
 PSNR =30.77
 bpp=1.94



Couple
 PSNR =33.45
 bpp= 1.48



House
 PSNR =32.54
 bpp= 1.25



Zelda
 PSNR =31.64
 bpp= 1.20



Balloons
 PSNR =32.46
 bpp= 0.92



Terraux
 PSNR = 33.65
 bpp= 1.02

Figure IV.7: Les images reconstruites dans l'espace YIQ avec la taille de bloc 16x16.

IV.5 Conclusion

Dans ce chapitre, nous avons proposé une technique de compression d'image, en changeant l'espace de couleurs RGB vers un autre espace moins corrélé, suivi respectivement par l'application de la transformée DCT, une étape de quantification et seuillage (partie destructrice du procédé) et enfin par un codage sans perte en utilisant l'encodeur TRE.

Les résultats que nous avons obtenus et que nous avons présentés dans ce chapitre sont assez satisfaisants du point de vue amélioration de performances du codeur (*PSNR*, *bpp*) par rapport à CBTC-PF [17] et à DCT-Adaptive Scanning [16].

Conclusion générale

Conclusion

Le travail présenté dans ce mémoire a été consacré à l'étude de l'effet des transformées de décorrélation en compression des images couleurs RGB. Ceci, est dans le but de trouver l'espace de couleurs le plus adapté à la compression. Afin d'atteindre cet objectif, nous avons proposé une technique de compression similaire à la norme JPEG.

Du fait qu'il y a une forte corrélation entre les trois plans constituants (R, G, B), nous effectuons, dans une première étape, un changement de l'espace de couleurs RGB vers un autre espace de couleurs moins corrélé. Evidemment, cette transformation permet une compression plus efficace car l'information est principalement condensée dans un plan d'une manière dominante par rapport aux autres.

Il est à noter, que nous avons testés plusieurs espaces de couleurs à savoir : YCbCr, YIQ, YUV, YDbDr, YPbPr, O1O2O3, NTSC, HSV, CMY, XYZ, Lab,...

Dans la seconde étape, la transformée DCT est appliquée, suivie d'une étape de quantification (nouveau quantificateur par rapport à JPEG [16]) et de seuillage (parties destructrices du procédé). Enfin, une étape de compression sans perte est appliquée en utilisant le codage TRE.

Une étude comparative avec une méthode récente, utilisée dans la littérature dans le domaine de la compression (CBTC-PF) ainsi qu'avec la méthode DCT-Adaptive Scanning, a été faite pour valider notre approche.

Les résultats que nous avons obtenus, et que nous avons présentés dans ce travail, sont assez satisfaisants du point de vue amélioration de performances du codeur (*PSNR*, *bpp*) par rapport aux méthodes précédemment évoquées.

L'application de l'approche proposée sur différentes images, a montré que :

- ✓ La meilleure performance peut être obtenue avec l'espace de couleurs **YIQ** comparé aux autres espaces de couleurs.
- ✓ Les tailles de blocs [16x16] et [32x32] sont les plus adaptés à cette approche.

En perspectives, nous allons travailler sur l'amélioration de la technique proposée en utilisant les ondelettes.

Nous envisageons également de considérer d'autres types de représentation des images couleurs, telle que les images halftoning.

Bibliographie

- [1] A. Abdelaziz, "Compression d'images par fractale", Thèse de magister en informatique, Université de Batna, 2002.
- [2] A. Manzanera, " Les images numériques", cours traitement et reconnaissance d'image, Master IAD, Ecole Nationale Supérieure de Techniques Avancées/Unité d'Électronique et d'Informatique, Université Pierre et Marie CURIE, Paris 2005.
- [3] A. Manzanera, "TERI : Traitement et reconnaissance d'image", Cours Traitement et reconnaissance d'image, Master, Université Pierre et Marie CURIE, Paris 2006.
- [4] J. Luc, "Les images numériques, Généralités ", Cours TERI, Diplôme d'Université Assistant à l'Usage des Technologies de l'Information et de la Communication (DU AUTIC) DEUST, Université de Bretagne Occidentale, 2002/2003.
- [5] D. Zeroual, "Implémentation d'un environnement parallèle pour la compression d'images a l'aide des fractales", Thèse de magister en informatique, Université de Batna, 2006.
- [6] J. Fruitet, "Outils et méthodes pour le traitement des images par ordinateur", Cours Université de Marne-la-vallée, France 2000.
- [7] L. Bouanzi, "Les formats de numération des images fixes", DESS en informatique Documentaire Rapport de stage, Ecole Nationale Supérieure des Sciences de l'Information et des Bibliothèques (ENSSIB), Université Claude Bernard Lyon1, 1999.
- [8] G. Almouzni, "Traitement numérique des images ", Cours Ecole Internationale des Sciences du Traitement de l'Information (EISTI), France, 2008.
- [9] J. Serra, "Notions de Base de traitement d'image", Cours Morphologie, Ecole des Mines de Paris, 2000.
- [10] M. Bergounioux, "Méthodes mathématiques pour le Traitement d'image", Cours Traitement et reconnaissance d'image, Master 2, Université Pierre et Marie CURIE, Paris 10 mai 2009.
- [11] R. Heus, "Approches virtuelles dédiées à la technologie des puces à tissus «Tissue MicroArrays » TMA : Application à l'étude de la transformation tumorale du tissu colorectal", Thèse de doctorat, Université Joseph Fourier, 28 Septembre 2009.
- [12] C. Bencheriet, A. Boualleg, H. Tebbikh, "Segmentation de la Couleur de Peau par Seuillage Selon Différents Espaces de Couleur", JIG'2007 - 3^{èmes} Journées Internationales sur l'Informatique Graphique, Université 8 Mai 45 de Guelma, 2007.

- [13] J. C. Russ, "The image processing –Handbook-", Third Edition. CRC Press, CRC Press LLC, 1998.
- [14] V. Risson, "Application de la Morphologie Mathématique à l'Analyse des Conditions d'Éclairage des Images Couleur", Thèse de doctorat, Ecole des Mines de Paris, 17 Décembre 2001.
- [15] B. Rémi, N. François, "La compression JPEG", Rapport de Projet, Ecole nationale supérieure d'ingénieurs de Caen et centre de recherche (ENSICAEN), Université de Caen Basse-Normandie, Année 2005-2006.
- [16] F. Douak, "Reconstruction des images compressées en utilisant les réseaux de neurones artificiels et la DCT", Thèse de magister en micro ondes pour télécommunications, Université de Batna, 2008.
- [17] C. D. Bibhas,, C. Bhabatosh, "Color image compression based on block truncation coding using pattern fitting principle", ELSEVIER, Vol.9, pp. 2408-2417, 2007.
- [18] C. Landry, "Correction interactive de couleur par association", Mémoire présenté à la faculté des études supérieures en vue de l'obtention du grade de Maîtrès sciences en informatique. Université de Montréal, Février 1999.
- [19] M. Beladgham, F. Derraz, M. Khélif, D. Chouguer, "Application de l'algorithme EZW pour la compression d'image médicale ", Symposium International : Qualité et Maintenance au Service de l'Entreprise, Université Abou Bekr Belkaid –Tlemcen, Algérie 2004.
- [20] A. Anwander, "Segmentation d'images couleurs par opération gradient vectoriel multiéchelle et contour actif : application à la quantification des phases minéralogiques du clinker de ciment", Thèse de doctorat, Institut national sciences appliquées de Lyon, Décembre 2001.
- [21] P. Beaurepaire, " Compression d'images appliquée aux angiographies cardiaques : aspects algorithmiques, évaluation de la qualité diagnostique ", Thèse de doctorat, Université Lyon, 21 novembre 1997.
- [22] J.J. Brault, D. Dougherty, "Les formats de compression d'image", Rapport de projet, Institut Universitaire de Technologie de Tours, Département Génie Électrique et Informatique Industrielle, 2004.
- [23] L. Belhumeur, "La compression d'images fixe infrarouges contenant des cibles tenues", Mémoire Présenté pour l'obtention du grade de Maîtrès Sciences (M. Sc.), Université Laval, juillet 97.

- [24] M. Mekouar, " Compression d'images médicales par ondelettes et régions d'intérêt", Mémoire pour l'obtention de la maîtrise en technologie des systèmes, Université du Québec, Montréal, 12 juin 2001.
- [25] O. Delgrange, "la compression informatique", Compte-rendus des Dixièmes Journées de Mathématique et de Sciences, Université de Mons-Hainaut, 1996.
- [26] T. Acharya,P-S. Tsai, "JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures", John Wiley & Sons, Inc, Canada, 2005.
- [27] A. Masmoudi, M. Salim Bouhleb, "Un nouvel algorithme de compression exploitant le codage arithmétique en lui introduisant de nouveaux paramètres de codage", Unité de Recherche: Sciences et Technologies de l'Image et des Telecommunications, Institut Supérieure de Biotechnologie de Sfax (ISBS)-TUNIS, 2007.
- [28] S. Saha, "Image Compression – from DCT to Wavelets: A review", ACM Cross words students magazine, Vol.6, No.3, spring 2000.
- [29] O. Kanoun, S.Mezghani et M. Salim Bouhleb, "Compression adaptative des images médicales", 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunication (SETIT 2005), Tunisia, March 27-31, 2005.
- [30] E. Christophe, "Imagerie Satellite et Compression", Cours Analyse Multi-Résolution, Université du Condorcet, 24 mai 2005.
- [31] H. B. Mitchell, N. Zilverberg, and M. Avraham, "A comparison of different block truncation coding algorithms for image compression," Signal Processing: Image Communications, vol. 6, no.1, pp. 77-82, March 1994.
- [32] R. Benzid, F. Marir, and N.-E. Bouguechal, "Electrocardiogram Compression Method Based on the Adaptive Wavelet Coefficients Quantization Combined to a Modified Two-Role Encoder", IEEE signal processing letters, vol. 14,pp.373-376, no. 6, June 2007.
- [33] M. Saadia, "Etude de la norme JPEG 2000 (en utilisant certaines types d'ondelettes dans la compression d'images monochromes)", Thèse de magister en électronique, Université de Biskra, 2007.
- [34] D. Chikouche, R. Benzid, M. Bentoumi, "Application of the DCT and Arithmetic Coding to Medical Image Compression", 3rd International Conference on Information and Communication Technologies: From Theory to Applications, ICTTA 2008.

- [35] Y.Q.Shi, H. Sun, "Image and video compression for multimedia engineering ", CRC Press 1999.
- [36] A. Skodras, C. Christopoulos et T. Ebrahimi, "The JPEG2000 Still Image Compression Standard", IEEE Signal Processing Magazine, vol. 18, pp. 36-58, September 2001.
- [37] O. Hammami, "Etudes d'optimisation algorithmiques de JPEG2000 (EIRE) ", Ministère de l'industrie dans le cadre du Réseau National de la Recherche en Télécommunications (RNRT), février 2004.
- [38] F. Charot, "Architectures Parallèles Spécialisées pour le Traitement D'image", INRIA, N°1978, IRISA Campus, Universitaire de Beaulieu, France, Avril 1993.
- [39] A. Cziho, "Quantification et Compression d'image. Application à l'imagerie Médicale", Thèse de doctorat, Université de Rennes 1, France, Mai 1999.
- [40] T. Totozafiny, "Compression d'images couleur pour application à la télésurveillance routière par transmission vidéo à très bas débit", Thèse de Doctorat, Université de Pau, juillet 2007.
- [41] S. Renard, "Compresser les images", INTERGRAPHIC 2000 Séminaires, Palais des Congrès, Paris, Janvier 2000.
- [42] D. Santacruz, R. Grosbois et T. Ebrahimi, "JPEG 2000 - la nouvelle norme pour le codage d'images", Cours Ecole Polytechnique, Fédérale de Lausanne, 27 mars 2001.
- [43] A. Gersho et R.M. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers.1993.
- [44] M. Nelson, " La compression de données : texte, images, sons", Editions Dunod, 1993.
- [45] C. Stalder, "Les algorithmes de compression sans perte", Cours Ecole spécialisée de suisse, 7 janvier 2005.
- [46] A. OUAFI, "Compression d'images fixes biomédicales par les transformées en ondelettes associées aux algorithmes de quantification vectorielle et de codage entropique", Thèse de magister en Electronique, Université de Biskra, Mai 2001.
- [47] R. Andrew Muyschondt, "Wavelet transform in image compression ", Thesis master of science in electrical engineering, Faculty of Texas Tech University, December, 1995.

- [48] X. Kai, Y. Jie, Z. Yue Min, L. Xiao Liang, " HVS-based medical image compression", European Journal of Radiology, vol. 7, pp. 139–145, 2005.
- [49] S. Ameer, O. Basir, "Image compression using plane fitting with inter-block prediction", Image and Vision Computing, vol.6, pp. 385–390, 2009.
- [50] <http://sipi.usc.edu/database/>

Résumé

L'objectif principal de ce travail, est l'étude de l'effet des transformées de décorrélation en compression d'images couleurs RGB.

Dans ce contexte, nous avons proposé une méthode de compression basée sur la DCT, et similaire à JPEG mais au lieu de changer l'espace de couleurs RGB vers YCbCr dans l'étape de transformation de l'espace de couleurs, nous testons plusieurs espaces de couleurs par exemple (YCbCr, YIQ, YUV, YDbDr, HSV....), ceci est afin d'effectuer une étude comparative pour trouver l'espace de couleurs le plus adapté à la compression. Des différences majeures par rapport à JPEG sont portées. Elles résident dans l'utilisation d'un quantificateur linéaire proposé par (Douak, Benzid et Benoudjit : DCT-Adaptive Scanning) ainsi que l'usage d'un autre codage.

entropique appelé TRE (Two-Role Encoder).

Les résultats que nous avons obtenus, sont assez satisfaisants du point de vue amélioration de performances du codeur (*PSNR*, *bpp*) par rapport à la méthode CBTC-BF et à DCT-Adaptive Scanning.

L'application de l'approche proposée sur différentes images, a montré que : la meilleure performance peut être obtenue avec l'espace de couleurs YIQ comparé aux autres espaces de couleurs et les tailles de bloc [16x16] et [32x32] sont les plus adaptées à cet algorithme.

De plus, cette technique présente l'avantage de contrôler la qualité souhaitée avant même d'effectuer la compression.

Mots clés :

Compression d'image couleurs, DCT, Espace de couleurs, Two Role Encoder (TRE).

Abstract

The main goal of this work is the study of the effect of the decorrelating transforms in RGB color image compression.

In this context, a color image compression method based on the DCT transform is proposed. It uses a similar idea as JPEG, except the difference in the color space transformation step that converts the RGB color image to another color space such as YUV, YIQ, YCbCr, YDbDr, and HSV. This is done in order to find the most suitable one to the RGB image compression. Also, the proposed method uses a linear quantizer (Douak Benzid and Benoudjit: DCT adaptive-scanning) and finally an entropy coder called TRE (Two-Role Encoder).

The results of our approach are particularly powerful compared to the CBTC-BF method and DCT-Adaptive Scanning method.

Applying the proposed algorithm on various images, showed that better performance can be achieved with RGB-YIQ transform compared to other color spaces and block sizes [16x16] and [32x32] are the most adapted to this compression scheme.

Moreover, this technique presents the advantage of quality controlling, in advance, before even carrying out compression.

Key words: Color image compression, DCT, Color Spaces, Two Role Encoder (TRE).