

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique



UNIVERSITE DE MSILA
FACULTE DE TECHNOLOGIE
DEPARTEMENT D'ELECTRONIQUE



MEMOIRE D'INGENIEUR

DOMAINE : SCIENCES ET TECHNOLOGIE
FILIERE : GENIE ELECTRONIQUE
OPTION : INSTRUMENTATION

Thème

**COMPRESSION D'IMAGES
COULEURS
PAR LA METHODE JPEG**

Présenté par :

BOUTEMINA Hicham
MAHMOUDI Salim
HAMRIT Hassan

Encadré par :

Mr .OUDIRA Houcine

N°d'ordre : 2012/ /06/14/374

Promotion : JUIN 2012

Remerciements

En premier lieu nous tenons à remercier le bon Dieu Tout Puissant et Miséricordieux de bonne santé, de la volonte et de la patience qu'il nous a accordé tout au long de notre formation .

Nous remercions très sincèrement Mr. H. OUDIRA pour les conseils avisés qu'elle nous a toujours prodigués, pour les connaissances dont elle nous a fait bénéficier, son suivi attentif et sa confiance qui nous a été très précieuse. Mais aussi nous tenons à la remercier pour nous avoir fait l'honneur de nous encadrer.

Nous tenons à exprimer notre gratitude aux membres du jury qui ont bien voulu consacrer de leur temps a l'examen et l'étude de notre travail.

Notre profonde gratitude a tous les enseignants du Département D'Électronique qui nous ont encouragés à donner le meilleur de nous-mêmes en nous assurant une formation aussi meilleur que possible

Un remerciement spécial aux Étudiants de 5ème d'Électronique et a tous ceux qui ont contribué de prés ou de loin a l'aboutissement de ce travail.

Dédicace

Je dédie ce modeste travail...

*À ceux qui sont les plus chers au monde, mon père et ma mère, à qui je
n'arriverai jamais à exprimer ma gratitude, pour leur amour et leur soutien
tout au long de mes études*

À mes frères Djamel, Mouloud, Abed Ennour, Mohammed.

À mes sœurs Meriem, Dalila et Houda .

A tous les nombres de ma famille.

Sans oublier mes fidèles amis ... Noro ,Belkacem et Med cherif.

Et à mes collègues surtout les étudiants de la 5^{eme} année ingénieur

Électronique que je les respecte.

SALIM

Dédicace

Je dédie ce modeste travail...

*À ceux qui sont les plus chers au monde, la mémoire de mon père,
et ma mère à qui je n'arriverai jamais à exprimer ma gratitude,
pour leur amour et leur soutien tout au long de mes études*

À mes frères et mes sœurs et à tous les nombres de ma famille.

*Sans oublier mes fidèles amis et mes collègues surtout les
étudiants de la 5^{eme} année ingénieurat Électronique que je les
respecte.*

Hassan

Dédicace

Louange à dieu

*A mes très chers parents qui ont toujours été là pour moi,
et qui m'ont donné un magnifique modèle de sacrifices et
de persévérance.*

A mes frères.

A toute ma famille.

A tous mes amis.

A tous ceux, qui m'aiment et que j'aime.

Je dédie ce modeste travail

HICHAM

Résumé

L'homme a toujours voulu découvrir la beauté des planètes, vues de l'espace. Il a donc envoyé des satellites capables de photographier celles-ci. Mais l'un des problèmes majeurs est la transmission de ces photos, du satellite à la terre. Cette communication se fait à l'aide des ondes électromagnétiques.

La transmission des informations est d'autant plus facile que le nombre d'informations est faible. Il est donc avantageux d'en réduire le nombre. Une technique employée dans de nombreux domaines est la compression des informations. L'utilisation d'algorithmes de compression d'images permet en effet, une réduction importante de la quantité de données.

Nous allons étudier un algorithme très répandu et utilisé par de nombreuses personnes: le Jpeg.

Mots Clés : Compression d'image, JPEG, Huffman, lecture en zigzag, Matlab, DCT, GUI

Sommaire

| | |
|-----------------------------|---|
| Liste des figures | |
| Liste des tableaux | |
| Introduction Générale | 1 |

Chapitre I : Traitement d'images

| | |
|---|----|
| I.1. Introduction | 2 |
| ❖ Qu'est ce qu'une image?..... | 2 |
| ❖ Pourquoi et comment traiter les images? | 3 |
| I.2. Notions fondamentales en traitement d'images? | 4 |
| I.2.1. Pixel | 4 |
| I.2.2. Voisinage d'un pixel | 4 |
| I.2.3. Image numérique | 5 |
| I.2.4. Résolution d'une image | 6 |
| I.2.5. Poids d'une image | 6 |
| I.2.6. Image en niveaux de gris | 6 |
| I.2.7. Image binaire | 7 |
| I.2.8. Image en couleurs | 8 |
| ❖ Représentation en couleurs réelles..... | 8 |
| ❖ Représentation en couleurs indexées | 9 |
| ❖ Autres modèles de représentation | 10 |
| I.2.9. Bruit dans l'image | 10 |
| I.2.10. Histogramme | 10 |
| I.2.11. Contraste | 11 |
| I.3. Formats des images | 11 |
| I.3.1. L'image bitmap | 11 |
| I.3.1.1. Quelques formats d'image bitmap | 12 |
| ❖ Le format JPEG (Joint Photographique Experts Group) | 12 |
| ❖ Le format JPEG 2000 | 12 |
| ❖ Le format TIFF (Tagged Image File Format)..... | 12 |
| ❖ Le format GIF (Graphics Interchange Format)..... | 13 |
| ❖ Le format PNG (Portable Network Graphics)..... | 13 |
| I.3.2. L'image vectorielle | 13 |
| ❖ Formats de fichiers vectoriels | 13 |
| I.4. Étapes de l'analyse d'images | 14 |
| I.5. Domaines d'applications du traitement d'images | 14 |
| I.6. Conclusion | 15 |

Chapitre II : Généralités sur les techniques de compression d'image

| | |
|--|----|
| II.1. Introduction | 16 |
| II.2. l'image numérique | 16 |
| II.3. Définition de la compression des images | 16 |
| II.4 Echantillonnage | 17 |
| II.5. Mesures des performances de la compression des images | 18 |
| II.5.1.Taux de compression | 18 |
| II.5.2. Erreur quadratique moyenne EQM | 18 |
| II.5.3. Rapport signal sur bruit SNR | 18 |
| II.5.4. Rapport signal sur bruit de crête PSNR | 19 |
| II.6. Classification des méthodes de compression | 19 |
| II.6.1. Les deux principes de base | 19 |
| II.6.1.1. Extraction de l'information pertinente | 19 |
| II.6.1.2 Élimination des redondances | 19 |
| II.6.2. Quelques méthodes de compression | 20 |
| II.6.2.1. Méthode par pixels, groupes des pixels, ou image entière | 20 |
| II.6.2.2 Méthodes adaptatives et non adaptatives | 20 |
| II.6.2.3. Méthodes basées sur les transformées | 20 |
| II.6.2.4 méthodes prédictives | 21 |
| II.6.2.5 méthodes hybrides | 22 |
| II.7. Conclusion | 23 |

Chapitre III : La norme JPEG

| | |
|--|----|
| III.1. Introduction | 24 |
| III.2.Principe générale de la compression des images | 24 |
| III.2.1.Décorrélacion | 24 |
| III.2.2.Quantification | 24 |
| III.2.2.1. Quantification scalaire | 25 |
| III.2.2.2. Quantification vectorielle | 25 |
| III.2.3. Codage | 26 |
| III.3. La norme JPEG | 26 |
| III.3.1. Introduction | 26 |
| III.3.2. Les méthodes de la norme de compression JPEG | 27 |
| III.3.3. La compression non conservative JPEG | 28 |
| III.4. La transformation DCT | 28 |
| III.4.1. Transformation linéaires | 28 |
| III.4.2. Transformation en cosinus discrète bidimensionnelle (DCT) | 29 |
| III.4.3 Propriétés importantes de la DCT en compression d'image | 31 |
| III.4.3.1. Décorrélacion | 31 |
| III.4.3.2. Concentration des coefficients | 31 |
| III.4.3.3. Symétrie, séparabilité, et orthogonalité | 32 |
| III.4.4. Avantages et inconvénients de la DCT | 32 |
| III.5. Compression basée sur la DCT (cas du JPEG) | 33 |
| III.5.1. Première étape : DCT | 33 |

| | |
|--|----|
| III.5.2. Deuxième étape: Quantification | 34 |
| III.5.2.1. Choix de la matrice de quantification: | 35 |
| III.5.3. Troisième étape: codage | 37 |
| III.5.3.1. Balayage en Zigzag | 37 |
| III.5.3.2 Codage d'entropie | 38 |
| ❖ Principe | 38 |
| ❖ Exemples et analyse | 39 |
| III.5.4. La décompression | 41 |
| III.6. Conclusion | 43 |
| Chapitre IV : Simulation et Résultats | |
| IV.1. Introduction | 44 |
| IV.2. Présentation de l'interface graphique | 44 |
| IV.2.1. Cette interface contient | 45 |
| IV.2.2. Démarche à suivre pour la compression des images | 46 |
| IV.3. Les critères de qualité | 48 |
| IV.3.1 Erreur Quadratique Moyenne (EQM) ou Mean Square Error (MSE) | 48 |
| IV.3.3. Le PSNR (Peak SNR) | 48 |
| IV.3.4. Taux de compression (Tc) | 49 |
| IV.4. L'algorithme | 49 |
| IV.4.1. Phase de compression | 51 |
| IV.4.1.1. Acquisition de l'image originale | 51 |
| IV.4.1.2. La DCT (Discret Cosine Transform) | 51 |
| IV.4.1.3. La quantification | 51 |
| IV.4.1.4. Le codage | 52 |
| IV.4.2. La phase de décompression | 52 |
| IV.4.2.1. Le décodage huffman | 52 |
| IV.4.2.2. Le décodage RLE | 52 |
| IV.4.2.3. La dé- quantification | 52 |
| IV.4.2.4. La IDCT | 52 |
| IV.4.2.5. La reconstruction de l'image originale | 52 |
| IV.5. Testes et Résultats de la compression | 53 |
| IV.6. Interprétation | 59 |
| IV.7. Conclusion | 60 |
| Conclusion Générale | 61 |
| Annexe A | |
| Bibliographie | |

Liste des Figures

| | |
|--|----|
| Figure I.1 : Voisinage d'un pixel | 5 |
| Figure I.2 : Valeurs des niveaux de gris et teintes de gris correspondantes | 7 |
| Figure I.3 : Exemple d'une image en niveaux de gris | 7 |
| Figure I.4 : Image noir & blanc « au trait » | 7 |
| Figure I.5 : Image numérisée avec 16 millions de couleurs | 9 |
| Figure I.6 : Histogramme d'image | 11 |
| Figure II.1 : Diagramme de compression d'image pour la transmission ou stockage | 17 |
| Figure II.2 : Codeur typique | 20 |
| Figure II.3 : Histogramme des niveaux de gris de l'image originale (gauche) et de l'erreur de prédiction (droite) | 21 |
| Figure II.4 : Balayage par ligne d'une image | 22 |
| Figure III.1 : Schéma d'un codeur source | 24 |
| Figure III.2: principe de l'algorithme JPEG avec pertes | 27 |
| Figure III.3: principe de l'algorithme JPEG sans pertes | 28 |
| Figure III.4: La compression non conservative "JPEG" | 28 |
| Figure III.5 : Images de la base de l'espace transformé DCT | 31 |
| Figure III.6. Le schéma de codage | 33 |
| Fig. III.7. Matrice du bloc original | 34 |
| Fig.III.8.Matrice de quantification | 36 |
| Fig.III.9. Matrice DCT avant quantification | 36 |
| Fig.III.10. Matrice DCT après la quantification | 37 |
| Fig.III.11.Le chemin de la séquence ZIG-ZAG | 38 |
| Fig.III.12. Schéma général de la décompression | 41 |
| Fig.III.13.Matrice après le décodage | 41 |
| Fig.III.14 Matrice déquantifiée | 41 |
| Figure III.15 matrice après IDCT | 42 |
| Figure IV.1 : Une capture de l'interface..... | 45 |
| Figure IV.2 : Organigramme général de la compression JPEG | 50 |
| Figure IV.3: Schéma bloc de la compression. | 52 |
| Figure IV.4: Schéma bloc de la décompression | 52 |

Liste des tableaux

| | |
|---|----|
| Tableau I.1 : Représentation en vraie couleurs (24 bits) | 8 |
| Tableau I.2 : Exemple d'une palette de couleur | 9 |
| Tableau IV.1 : Résultats de la compression avec FQ=1 | 53 |
| Tableau IV.2 : Résultats de la compression avec FQ=3..... | 54 |
| Tableau IV.3 : Résultats de la compression avec FQ=10 | 55 |
| Tableau IV.4 : Résultats de la compression avec FQ=20 | 56 |
| Tableau IV.5 : Résultats de la compression avec FQ=50 | 57 |
| Tableau IV.6 : Résultats de la compression avec FQ=80 | 58 |
| Tableau IV.7: Résultats de la compression avec FQ= (1 jusqu'à 99) | 59 |

Introduction générale

L'homme s'est toujours servi de représentations graphiques pour transmettre des connaissances ou pour noter des situations dont il voulait garder le souvenir. L'utilisation des images se justifie par le fait que le contenu représenté peut être facilement compréhensible par de larges classes de personnes possédant une culture analogue.

Le traitement d'images a pour vocation l'étude, la conception et la réalisation de systèmes d'exploitation d'images considérées comme véhicules de l'information. Son but ultime consiste à extraire le contenu informationnel (ou information pertinente) des images en vue d'une prise de décision ou d'une action. Cette information peut être à deux niveaux différents :

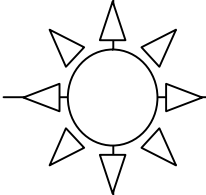
- ❖ Le traitement de « bas niveau » auquel est affectée la tâche d'extraction des primitives pertinentes de l'image dont le but est la réduction de la quantité d'information contenue dans l'image.
- ❖ Le traitement de « haut niveau » destiné à l'interprétation du contenu de l'image dans un but de reconnaissance et de compréhension.

Notre travail s'inscrit dans le cadre d'un projet dont l'objectif consiste à effectuer une compression des images couleurs par le standard « JPEG ».

Dans le premier chapitre, nous allons étudier les différentes caractéristiques de l'image, précisément sa taille qui pose un problème dans sa capacité de stockage et dans sa transmission à travers les différents réseaux surtout en matière de vitesse. Une vue générale sur les différentes techniques utilisées dans la compression d'image sera l'objet du deuxième chapitre, où nous allons étudier les différentes méthodes de compression sans et avec perte.

Le troisième chapitre sera réservé à la description détaillée de notre approche « le standard JPEG » dans le quel s'articule ce travail. Le dernier chapitre sera consacré à la partie simulation où nous allons d'une part appliquer le standard JPEG, afin de limiter la taille, ou le poids, d'une image c'est-à-dire éliminer les informations inintéressantes ou redondantes sans détruire la qualité d'image et d'autre part exposer les résultats.

Chapitre I : Traitement d'images



I.1. Introduction

I.2. Notions fondamentales en traitement d'images?

I.3. Formats des images

I.4. Étapes de l'analyse d'images

I.5. Domaines d'applications du traitement d'images

I.6. Conclusion

I.1. Introduction

Avec la parole, l'image a toujours constituée l'un des moyens le plus privilégié qu'utilise l'homme pour communiquer avec autrui. En effet c'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre. Chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

De ce fait, le terme générique *traitement d'images* est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes en regard de l'application concernée. [1] Avant de donner les notions fondamentales et les outils de traitement d'images, il serait utile de définir la notion d'image et les raisons de son traitement.

❖ Qu'est ce qu'une image ?

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc.

C'est aussi un ensemble structuré d'informations qui, après affichage sur écran, ont une signification pour l'œil humain.

Afin que l'image puisse être traitée par la machine, sa numérisation est nécessaire. [1]
Nous distinguons trois groupes d'images : [2]

1- Les images *physiques visibles*, qui sont parfaitement matérielles et de nature volatile (soit des images optiques, constituées de photons dans le domaine visible, soit les images électro-optiques), ou de nature permanente (reproductions de toutes sortes : clichés photographique, peintures, sculptures, documents, imprimés, etc.

2- Les images *physiques non visibles*, les images optiques hors du domaine visible ou images de nature immatérielle (spectres de physique, cartes de population, de températures, etc.). Une vue infrarouge est une image non visible mais, après impression d'une pellicule sensible à ce rayonnement, elle devient une image physique visible.

3- Les images *mathématiques*, qui sont des concepts et sont donc invisibles par nature. Elles peuvent être continues (par exemple une fonction $f(x, y)$) ou discrètes : ce sont alors des tableaux ou matrices de nombres.

❖ Pourquoi et comment traiter les images ?

On peut dire que l'on traite des images dès lors que l'on extrait une information de cette image. Il existe de très nombreuses façons de traiter des images. Une première manière de tenter une classification des types de traitement est de se référer au but poursuivi.

Pourquoi cherche-t-on à traiter une image ? [2]

1-On peut chercher à améliorer sa qualité subjective (la rendre plus agréable à l'œil) ou objective, donc mesurable : on veut améliorer son contraste, accroître la perception de certains détails ou contours, faire ressortir plus nettement certaines zones ou certaines formes, diminuer le bruit, etc. Il s'agit donc de techniques d'*amélioration* qui font appel à des procédés variés créant des lissages, rehaussements, etc.

2-On peut chercher à trouver une image « idéale » de l'objet qui a été dégradé par divers processus : on veut diminuer les fluctuations dues à des phénomènes de turbulence. Il s'agit là de techniques de *restauration*.

3-On peut chercher à détecter la présence de certaines formes, certains contours ou certaines textures de modèle connu, sans vouloir préserver les autres informations.

4-On peut chercher à réduire l'énorme quantité d'informations contenues dans une image, à la « compresser » afin de gagner en vitesse de transmission, en encombrement des organes utilisés, en capacité de stockage, etc. tout en ne dégradant que le moins possible les images considérées. Ce domaine est celui du *codage*, de la *compression* des données et de l'*approximation* des images.

5-On peut chercher à faire une analyse de l'image. Cette dernière consiste en l'extraction des informations contenues dans les divers objets de la scène, sans toutefois fournir une interprétation. Il s'agit des techniques d'*extraction d'attributs*, c'est-à-dire de « formes » (contours, textures) et de *segmentations*.

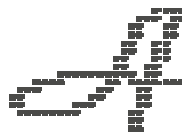
Durant ce chapitre nous verrons quelques notions fondamentales, ainsi que quelques

outils de traitement d'images.

I.2. Notions fondamentales en traitement d'images

I.2.1. Pixel

Abréviation de « Picture Élément », mot anglo-américain de **pix** pour pics, abréviation de « Picture » (image) et de **el** pour « élément ». Élément d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous :



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

Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B). [1]

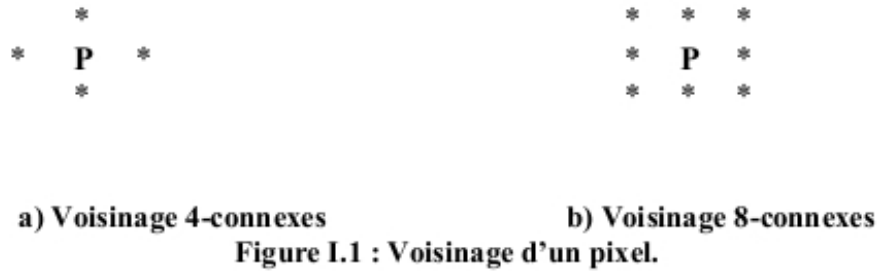
I.2.2. Voisinage d'un pixel

Le voisinage d'un pixel est composé de tous les pixels qui l'entourent immédiatement. Si p est un pixel d'une image D , alors le voisinage de p est le plus petit sous-ensemble de D qui contient p .

Dans une image numérique, on distingue deux types de connexités relatives au voisinage utilisé : la 4-connexité (Figure I.1.a) et la 8-connexité (Figure I.1.b). Le voisinage d'un pixel (i,j) est dit 4-connexe s'il est formé des quatre pixels de coordonnées spatiales

$(i+1,j), (i,j+1), (i-1,j), (i,j-1)$, et il est dit 8-connexe s'il est formé des pixels de coordonnées spatiales

$(i+1,j), (i,j+1), (i-1,j), (i,j-1), (i+1,j-1), (i+1,j+1), (i-1,j+1), (i-1,j-1)$. [3]



I.2.3. Image numérique

Contrairement aux images obtenues à l'aide d'un appareil photo, ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits).

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

La numérisation d'une image est la conversion de celle-ci de son état analogique (distribution continue d'intensités lumineuses dans un plan xOy en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x,y)$ où :

x, y : coordonnées cartésiennes d'un point de l'image.

$f(x, y)$: niveau de gris en ce point

Pour des raisons de commodité de représentation pour l'affichage et l'adressage, les données images sont généralement rangées sous formes de tableau I (matrice) de N lignes et P colonnes. Chaque élément $I(x, y)$ représente un pixel de l'image et à sa

valeur est associé à un niveau de gris codé sur m bits (2^m niveaux de gris ; 0 = noir ; $2^m -$

1 = blanc). La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur. [1]

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,P-1) \\ f(1,0) & f(1,1) & & & f(1,P-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(N-1,0) & \dots & & & f(N-1,P-1) \end{pmatrix}$$

I.2.4. Résolution d'une image

La résolution d'une image est définie par un nombre de pixels par unité de longueur de la structure à numériser (classiquement en dpi (dots per inches) ou ppp (points par pouce)). Ce paramètre est défini lors de la numérisation et dépend principalement des caractéristiques du matériel utilisé lors de processus de numérisation. Plus le nombre de pixels est élevé par unité de longueur de la structure à numériser, plus la quantité

d'information qui décrit cette structure est importante et plus la résolution est élevée. La résolution d'une image numérique définit le degré de détail qui va être représenté sur cette image.

I.2.5. Poids d'une image

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

Exemple : Calcul pour une image 640x480

Nombre de pixels :

$$640 \times 480 = 307200.$$

$$24 \text{ bits} / 8 = 3 \text{ octets}.$$

Le poids de l'image est ainsi égal à :

$$307200 \times 3 = 921600 \text{ octets}.$$

$$921600 / 1024 = 900 \text{ ko}.$$

I.2.6. Image en niveaux de gris

Dans une image en niveaux de gris, la couleur d'un pixel peut prendre des valeurs allant du noir au blanc, en passant par un nombre fini de niveaux intermédiaires. Dans une telle image les intensités du rouge, du vert, et du bleu de chaque pixel sont égales. En général, les images en niveaux de gris renferment 256 teintes de gris (image à 256 couleurs), simplement chacune de ces 256 couleurs est définie dans la gamme des gris. Par convention la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur

255 le blanc (intensité lumineuse maximale). Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux. [1]

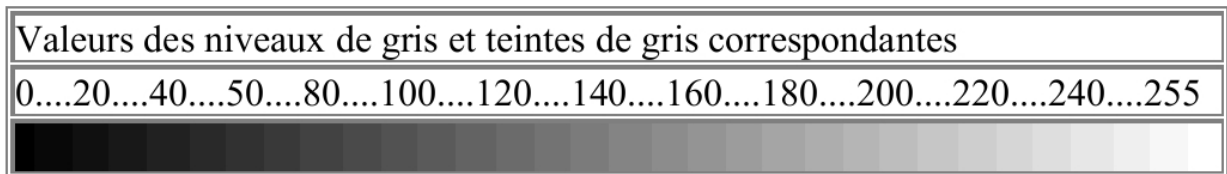


Figure I.2 : Valeurs des niveaux de gris et teintes de gris correspondantes.

Exemple :



Figure I.3 : Exemple d'une image en niveaux de gris.

I.2.7. Image binaire

Ces images sont constituées de points (pixels) qui ne peuvent avoir que deux états : noir ou blanc. On les appelle des images « au trait ». On trouve parmi elles des pages de texte, des signatures, des plans, des dessins, etc. [2]

Exemple:

En savoir plus sur les images

Figure I.4 : Image noir & blanc « au trait ».

I.2.8. Image en couleurs

Une image en couleurs est censée représenter le mieux possible la réalité. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.). [1] [6]

❖ Représentation en couleurs réelles (Images en "vraies couleurs" (ou 24 bits))

Elle consiste à utiliser 24 bits pour chaque point de l'image. Huit bits sont employés pour décrire la composante rouge (R), huit pour le vert (V) et huit pour le bleu (B). Il est ainsi possible de représenter environ 16,7 millions de couleurs différentes simultanément. Cela est cependant théorique, car aucun écran n'est capable d'afficher 16 millions de points. Dans la plus haute résolution (1600 x 1200), l'écran n'affiche que 1 920 000 points. Par ailleurs, l'œil humain n'est pas capable de distinguer autant de couleurs.








| Valeur R | Valeur V | Valeur B | Couleur correspondante | Commentaires |
|----------|----------|----------|---|---|
| 0 | 0 | 0 |  | noir |
| 0 | 0 | 1 |  | un peu moins noir (nuance impossible à détecter à l'oeil par rapport au noir) |
| ... | ... | ... | ... | ... |
| 0 | 0 | 255 |  | bleu |
| ... | ... | ... | ... | ... |
| 0 | 255 | 0 |  | vert |
| ... | ... | ... | ... | ... |
| 255 | 0 | 0 |  | rouge |
| ... | ... | ... | ... | ... |
| 128 | 128 | 128 |  | couleur intermédiaire correspondant à un gris |
| 255 | 255 | 255 |  | blanc |

Tableau I.1 : Représentation en vraie couleurs (24 bits).

L'information couleur de chaque pixel est donc codée par 3 octets, ce qui fait que les images en vraies couleurs sont des images très "lourdes".

Exemple :



Figure I.5 : Image numérisée avec 16 millions de couleurs.

❖ Représentation en couleurs indexées

Afin de diminuer la charge de travail nécessaire pour manipuler des images en 24 bits, on peut utiliser le mode de représentation en couleurs indexées. Le principe consiste à déterminer le nombre de couleurs différentes utilisées dans l'image, puis à créer une table de ces couleurs en attribuant à chacune une valeur numérique correspondant à sa position dans la table.

La table, appelée palette, comporte également la description de chacune des couleurs, sur 24 bits.

Exemple :

| Palette avec les codes RVB | Couleurs correspondantes: |
|----------------------------|---------------------------|
| Couleur 0: 255 255 255 | |
| Couleur 1: 255 255 204 | |
| Couleur 2: 255 255 153 | |
| Couleur 3: 255 255 102 | |
| Couleur 4: 255 255 51 | |
| Couleur 5: 255 255 0 | |
| | |

Tableau I.2 : Exemple d'une palette de couleur.

Ainsi, un pixel à qui sera affectée la couleur numéro 0 sera visualisé en blanc, un pixel à qui sera affectée la couleur numéro 4 sera visualisé en jaune...

❖ **Autres modèles de représentation**

Le modèle R.V.B. représentant toutes les couleurs par l'addition de trois composantes fondamentales, n'est pas le seul possible. Il en existe de nombreux autres. L'un d'eux est particulièrement important. Il consiste à séparer les informations de couleurs (chrominance) et les informations d'intensité lumineuse (luminance). Il s'agit du principe employé pour les enregistrements vidéo. La chrominance est représentée par deux valeurs (selon des modèles divers) et la luminance par une valeur.

I.2.9. Bruit dans l'image

Le bruit dans une image est défini comme étant un phénomène de brusque variation d'un pixel isolé par rapport à ses voisins, il affecte la qualité de l'image et il provient soit du dispositif d'acquisition (scanner, caméra, amplification, quantification...) soit de la scène elle-même (poussières, rayures...).

I.2.10. Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. 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. [1]

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

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

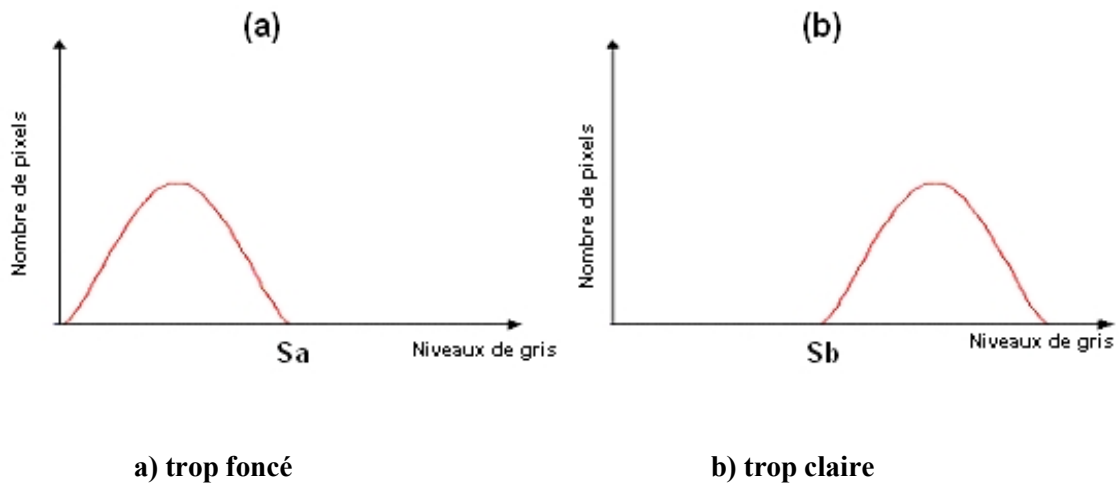


Figure I.6 : Histogramme d'image.

I.2.11. 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. [1]

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 :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \dots\dots(1)$$

I.3. Formats des images

A la différence d'une photographie ou d'un tableau, qui sont à la fois objet de la perception et mémoire de l'information. L'image numérique en tant qu'objet de perception n'existe pas, seule sa représentation est pertinente.

L'image numérique fixe (par opposition à la vidéo numérique ou l'animation en images de synthèse) se répartissent en deux catégories : image bitmap et image vectorielle. [5] [7]

I.3.1. L'image bitmap (matricielles)

L'image bitmap, dite aussi image matricielle, est en effet composée d'une matrice de points, les pixels. Ces points peuvent être en noir & blanc ou en couleur.

La caractéristique fondamentale d'une image bitmap est de représenter le plus exactement

possible la réalité : elle vise au photoréalisme, elle est souvent produite par un périphérique de saisie (scanner, appareil photo numérique ou caméra). Elle sera le plus souvent (plus ou moins) retouchée.

I.3.1.1. Quelques formats d'image bitmap

❖ Le format JPEG (Joint Photographique Experts Group)

Le format JPEG est un format de compression des images graphiques. Ce format accepte les images RGB (RVB) et CYMK (CMJN). IL est sans doute le mode de compression le plus efficace qui soit, avec un bon compromis entre gain d'espace disque, temps de compression/décompression et qualité des images.

Ainsi une image brute de 2Mo n'occupera après conversion en JPEG que 130 à 400 Ko, selon la qualité d'image voulue.

Le format JPEG est un format les plus utilisés dans le monde du World Wide Web (WWW). Il peut être lu directement par les principaux outils de navigation.

❖ Le format JPEG 2000

Le JPEG 2000 constitue une étape importante dans les procédés de codage de l'image numérique en général. Cette norme va probablement s'imposer et remplacer notre vieux JPEG qui nous aura rendu service pendant plus de 10 ans.

Cette évolution sera plus lente qu'on pouvait le penser pour plusieurs raisons. Elle nécessite la modification de nombreux logiciels et de nombreux dispositifs matériels.

La mise en œuvre de JPEG 2000 n'est pas aisée car il utilise des algorithmes nouveaux et des connaissances récentes. [8]

❖ Le format TIFF (Tagged Image File Format)

Le format TIFF, conçu à l'origine par la compagnie ALDUS est un format matriciel. Conçu au départ part pour n'accepter que les images en RGB (RVB), ce format permet de coder des images CYMK (CMJN).

Ainsi une image CYMK (CMJN) enregistrée en format TIFF peut être placée dans un logiciel de mise en page et être envoyée à l'impression sans perte de qualité au niveau de l'image, compte tenu de sa résolution d'origine.

Le format TIFF offre l'avantage d'occuper moins d'espace disque, grâce à son propre algorithme de compression appelé LZW. [9]

❖ **Le format GIF (Graphics Interchange Format)**

Ce format a été créé pour faciliter le transport des images. Il est développé par compuserve, un des premiers fournisseurs d'accès Internet, est particulièrement bien adapté (vu sa forme compacte), aux applications de transmission de données. Les fichiers sont au format pixel et peuvent gérer jusqu'à 256 couleurs. [9]

❖ **Le format PNG (Portable Network Graphics)**

Il y a trois types de ce format : en vraies couleurs, en niveaux de gris et en 8 bits basés sur l'utilisation d'une palette. Il utilise une compression sans perte très performante. Il peut être considéré dans tout usage même Internet. [9]

I.3.2. L'image vectorielle

A l'inverse de l'image bitmap, une image vectorielle est le plus souvent créée sans rapport avec la réalité. Elle peut avoir pour objectif de représenter la réalité, mais elle ne vise pas au photoréalisme.

Une image vectorielle n'est pas composée de points. Elle est formée de courbes, d'objets..., dotés ou non de contour et de surface.

Si l'image bitmap est une photo, l'image vectorielle est un dessin. Contrairement à l'image bitmap, l'image vectorielle n'a pas obligatoirement la forme d'un rectangle. Si on représente un cercle en mode vectoriel, le fichier aura la forme d'un cercle.

❖ **Formats de fichiers vectoriels**

A l'inverse des images bitmap qui sont photoréalistes et encombrantes, les images ou illustrations vectorielles sont beaucoup moins encombrantes en ce qui concerne la taille des fichiers

générés, mais elles ne peuvent en aucun cas se prévaloir d'un quelconque aspect photoréalisme.

Les formats de fichiers vectoriels sont moins nombreux que les formats de fichiers bitmap.

I.4. Étapes de l'analyse d'images

La plupart des applications de traitement d'image passent par les étapes suivantes : [10]

- **Acquisition** : Echantillonnage, Quantification.
- **Analyse globale de l'image et transformations ponctuelles** : Histogramme ; statistiques (moyenne, écart-type, etc.) ; transcodage (palette de couleur [LUT]) et classification.
- **Opérations entre images** : Indices, ratio, différence, opérations logiques, masques et seuillage.
- **Amélioration, filtrage et segmentation** : Opérations de convolution (lissage, rehaussement, détection de contours) ; squelettisation ; vectorisation.
- **Interprétation et sémantique** : Cartographie thématique (classification automatique et supervisée) ; cartographie vecteur ; représentation des graphes et de la topologie.

I.5. Domaines d'applications du traitement d'images

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que les télécommunications (T.V., vidéo, publicité,...), la médecine (radiographie, ultrasons,...), la biologie, l'astronomie, la géologie, l'industrie (robotique, sécurité), la météorologie, l'architecture, l'imprimerie, l'armement (application militaire).

De nouvelles applications pratiques sont possibles aujourd'hui et touchent tous les domaines d'activités, tels que : métiers du spectacle, de la radio, créations artistiques, etc. [1]

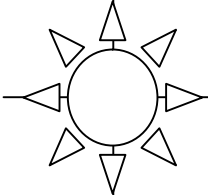
I.6. Conclusion

Dans ce chapitre, nous avons vu et étudié les différentes caractéristiques de l'image, précisément sa taille qui pose un problème dans sa capacité de stockage et dans sa transmission à travers les différents réseaux surtout en matière de vitesse.

Ainsi pour résoudre ce problème l'image doit passer par telles méthodes de compression dite sans ou avec perte, en cherchant d'une part, à réduire au maximum sa taille et d'autre part à grader, par la même occasion et au mieux possible, sa qualité.

Pour cela nous allons étudier dans le chapitre suivant, les différents techniques de compression.

Chapitre II : Généralités sur les techniques de compression d'image



II.1. Introduction

II.2. l'image numérique

II.3. Définition de la compression des images

II.4 Echantillonnage

II.5. Mesures des performances de la compression
des images

II.6. Classification des méthodes de compression

II.7. Conclusion

II.1. Introduction

C'est avec l'apparition des ordinateurs et surtout avec **Internet**, que les images sont devenues prépondérantes. En effet, quoi de plus précis qu'une image ?

Il faudrait plusieurs pages de textes pour décrire une simple photographie, ou un schéma... Cependant, si dans la rapidité d'information l'image détrône facilement le texte, celle-ci devient vite lourde de données, donc excessivement longue à transmettre et à traiter. Voilà pourquoi depuis quelques années, les centres de recherche en informatique dépensent de nombreuses heures sur des algorithmes de compression. Afin de limiter la taille, ou le poids, d'une image, nous devons la compresser, c'est à dire éliminer les informations inintéressantes ou redondantes. Il existe de nos jours plus d'une vingtaine de formats de compression, spécifiquement dans la compression d'image (GIF, JPEG, BMP...), ayant chacun leur propre méthode de codage, en cumulant plusieurs algorithmes, mais tous sont complémentaires.

Presque chaque format de fichier incorpore l'une ou l'autre méthode de compression. L'enjeu de la recherche sur compression d'image est de trouver un moyen de diminuer la taille d'une image, tout en essayant de limiter la dégradation due à la compression.

Nous allons le long de ce chapitre étudier les différentes méthodes de compression sans et avec perte.

II.2. L'image numérique:

Une image numérique, telle qu'on peut la voir sur un écran d'ordinateur, est une mosaïque de pixels (Picture éléments) dont la couleur est choisie dans un ensemble fini: il s'agit d'un objet naturellement discret. Il s'identifie à une matrice à x colonnes et y lignes dont les éléments appartiennent à un ensemble fini E . Typiquement, pour une image en niveau de gris, E est constitué des entiers compris entre 0 et 255 et correspond à l'intensité lumineuse de chaque pixel. Ces 256 valeurs distinctes se codent avec 8 bits ($2^8=256$), d'où le nom d'image 8 bits. Pour les images couleurs, chaque pixel est caractérisé par 3 intensités lumineuses, celles des canaux rouge, vert et bleu, définissant des images $3 \times 8=24$ bits.

Le stockage en mémoire d'une image couleur requiert donc $x \times y \times 24$ bits. Ceci représente rapidement une grande quantité: une image de 4 méga pixels nécessite ainsi $4 \times 2^{20} \times 24$ bits soit 12 mégaoctets, ne permettant le stockage que d'une dizaine de photos sur une carte de 128 mégaoctets. L'objectif de la compression est de réduire la quantité de mémoire nécessaire pour le stockage d'une image ou de manière équivalente de réduire le temps de transmission de celle-ci. Cette compression peut soit conserver l'image intacte, on parle alors de compression sans perte, soit autoriser une dégradation de l'image pour diminuer encore l'empreinte mémoire, on parle ici de compression avec perte.[11]

II.3. Définition de la compression des images

La compression est l'action utilisée pour réduire la taille physique d'un bloc d'information. En compressant des données, on peut placer plus d'informations dans la même espace de stockage,

ou utiliser moins de temps pour le transfert au travers d'un réseau téléinformatique. Parce que généralement les images requièrent une place importante, la compression est devenue une part intégrante des données graphique. Presque tous les formats des fichiers graphiques utilisent l'une ou l'autre méthode de compression. On rencontre souvent la méthode de compression des données comme étant une partie de l'encodage de données au même titre que le cryptage des données (cryptographie) et la transmission de données. Un compresseur, comme son l'indique, compresse les données alors qu'un décompresseur reconstruit les données originelles.[12]

Il est évident qu'un décompresseur ne peut opérer que connaissant l'algorithme de compression utilisée. la plupart des méthodes de compression visent à éliminer la redondance. Plusieurs types de redondance :

* La redondance spatiale entre deux pixels ou blocs voisins dans une image.

*La redondance temporelle entre images successives dans une séquence vidéo.

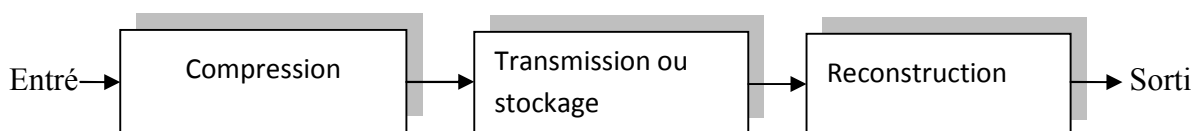


Figure II.1. Diagramme de compression d'image pour la transmission ou stockage

II.4. Echantillonnage

L'objectif de l'opération d'échantillonnage est de convertir une image analogique en une suite finie d'échantillonnage, telle que la reconstitution de l'image originale à partir de ces échantillons soit possible et de qualité acceptable.

La méthode la plus souvent utilisée consiste à diviser uniformément une image en $N \times N$ surfaces élémentaires appelées éléments d'image (pixels). le pas d'échantillonnage désigne la distance entre deux centres voisins consécutifs. [13], [11].

Si l'image échantillonnée correctement, sa reconstitution est exacte. Dans le cas d'un sous échantillonnage, on ne peut pas reconstruire exactement l'image en raison de chevauchement.

Par contre, si le pas d'échantillonnage est diminué, on améliore la qualité de l'image restaurée, mais on augmente la quantité de données.

Selon la théorème d'échantillonnage de Shannon, une image peut être correctement reconstituée a partit de ses échantillons, si le pas d'échantillonnage (T_e).est inférieur ou égale a l'inverse du double de la fréquence de coupure (F_c)

$$T_e \leq \frac{1}{2 \cdot F_c} \quad (\text{II.1})$$

La limite supérieure est le pas de Shannon:

$$T_e \text{ sup} = \frac{1}{2 \cdot F_c} \quad (\text{II.2})$$

Mais cette considération est théorique. En pratique, il convient de choisir un pas d'échantillonnage petit pour au maximum le recouvrement dans l'espace de Fourier et améliorer la qualité de restauration de l'image.

II.5. Mesures des performances de la compression des images

II.5.1. Taux de compression (Tc)

Etant donné que l'objectif d'une compression est de minimiser la quantité d'informations nécessaires à la représentation d'une image: nous définissons alors le paramètre suivant:

$$Tc = \frac{\text{nombre de bits de l'image originale}}{\text{nombre de bits de l'image compressée}} \quad (\text{II.3})$$

Par définition, le taux de compression sera égal à l'unité si nous ne réduisons par le nombre de bits nécessaire à la représentation de l'image original. En revanche si nous imaginons une technique de compression permettant de représenter l'image compressée avec seulement un bit par exemple, alors le taux de compression sera égal au nombre de bits de la représentation de l'image. Par exemple pour une image monochromatique de taille 256x256 codée sur 8 bits par pixel nous aurons un taux de compression 256x 256x8. [14]

II.5.2. L'erreur quadratique moyenne (EQM)

La mesure de distorsion utilisée généralement en compression d'image est l'erreur quadratique moyenne EQM cette grandeur est définie par la moyenne des écarts au carré entre le pixel $I_o(m,n)$ de l'image originale et le pixel $I_r(m,n)$ de l'image reconstruite.[14]

$$EQM = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I_o(i,j) - I_r(i,j)||^2 \quad (\text{II.4})$$

II.5.3. Rapport signal sur bruit (SNR)

Le rapport signal sur le bruit (signal to noise ration) est défini par (SNR):

$$SNR = 10 \cdot \log \frac{\sum_m \sum_n [X(m,n)]^2}{EQM} dB \quad (\text{II.5})$$

II.5.4 Rapport du signal sur bruit de crête (PSNR)

S'écrit:
$$PSNR = 10 \cdot \log \left(\frac{I^2}{EQM} \right) dB \quad (II.6)$$

Où I est la dynamique du signal (la valeur maximum possible pour un pixel). Dans le cas standard d'une image où les composantes d'un pixel sont codées sur 8 bits, $I=255$.

II.6. Classification des méthodes de compression:

II.6.1. Les deux principes de base

Les deux principes sur lesquels s'appuient les méthodes de compression sont l'extraction de l'information pertinente et l'élimination des redondances, au plan spatial comme au plan temporel. Ils sont utilisés conjointement dans un schéma complet de codage. Voici une liste synthétique des principales méthodes utilisées.

II.6.1.1. Extraction de l'information pertinente

- seuillage,
- quantification des coefficients,
- quantification vectorielle,
- techniques de quantification adaptative,
- sous-échantillonnage spatial et temporel.

II.6.1.2 Élimination des redondances

- Codage prédictif
- modulation Delta,
- MICD ("DPCM") ligne par ligne,
- MIC 2-D ("DPCM 2-D"),
- prédiction intertrame.
- Décorrélacion
- Transformée en Cosinus Discret (« Discrete Cosine Transform »),
- codage en sous-bande,
- codage par transformation 3-D.
- Codage entropique
- codage des longueurs de plage,
- codage de Huffman,
- codage arithmétique,

- codage par plans de bits.

II.6.2. Quelques méthodes de compression

II.6.2.1 Méthodes par pixels, groupes de pixels, ou image entière:

Cette classification s'intéresse à la zone de travail de base sur laquelle s'appliquent les algorithmiques: des pixels individuels, des blocs de pixels, ou toute l'image.

Pour citer quelques exemples:

- Codage de blocs de pixels: les standards JPEG travaillent par blocs adjacents de 8x8 pixels
- Codage individuel des pixels: méthodes sans perte par DPCM et codage de Huffman et codage RLE .
- Codage de l'image entière: codage d'image sous-bandes par filtrage ou décomposition sur une base d'ondelettes.[15]

II.6.2.2. Méthodes adaptatives, non-adaptatives

Cette classification indique si la méthode de compression est adaptative ou non. Une méthode non-adaptative applique les mêmes paramètres (bits alloués et quantificateurs) de façon fixe, quel que soit l'ensemble de données à coder. Une méthode adaptative va modifier ses paramètres au fur et à mesure du codage, en s'adaptant aux données d'entrée.[15]

II.6.2.3.Méthodes basée sur les transformées

Un codeur typique d'images à un taux de bits faible comporte trois étages: un étage de transformation, un étage de quantification et un dernier étage de compression de données.

On peut schématiser ce qui précède par la figure

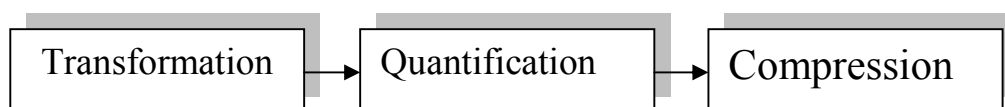


Figure .II.2 Codeur typique

Dans ces méthodes, l'image de dimension $N*N$ est subdivisée en sous images ou blocs de tailles réduites(la quantité de calcul demandée pour effectuer la transformation sur l'image entière est très élevée). Chaque bloc subit une transformation mathématique orthogonale inversible linéaire du domaine spatial vers le domaine fréquentiel, indépendamment des autres blocs (transformée en un ensemble de coefficients plus ou moins indépendants).

Les coefficients obtenus sont alors quantifiés et codés en vue de leur transmission ou de leur stockage. Pour retrouver l'intensité des pixels initiaux, on applique sur ces coefficient la transformation inverse. Parmi les transformations linéaires existantes:

- Transformation de Karhunen-loeve (TKL).

- Transformation de Fourier discrète (TFD).
- Transformation de Hadamard 5TH).
- Transformation de Haar 5THA).
- Transformation en cosinus discrète (DCT).

II.6.2.4. Méthodes prédictives

Les méthodes prédictives ont été introduites très tôt dans la démarche générale de réduction de redondance dans les images. Dans cette approche, on prédit la valeur $x(m,n)$ d'un échantillon $x(m,n)$ en fonction des échantillons voisins. La prédiction peut concerner aussi bien les valeurs des pixels de l'image que les valeurs des échantillons d'une de ses transformées.

La différence:

$$e(m,n) = x(m,n) - \hat{x}(m,n) \quad (\text{II.7})$$

Est appelée erreur de prédiction. C'est ce signal qui est quantifié, codé et transmis à la place du signal original. Les pixels voisins d'une image étant corrélés, on peut, en général, prédire des valeurs $\hat{x}(m,n)$ qui sont très proches des $x(m,n)$. la gamme dynamique de signal erreur de prédiction (figure II.3) est alors beaucoup plus faible que celle de $x(m,n)$. l'erreur de prédiction peut ainsi être quantifiée avec beaucoup moins de niveaux, ce qui réduit considérablement le nombre de bits par pixel[16].

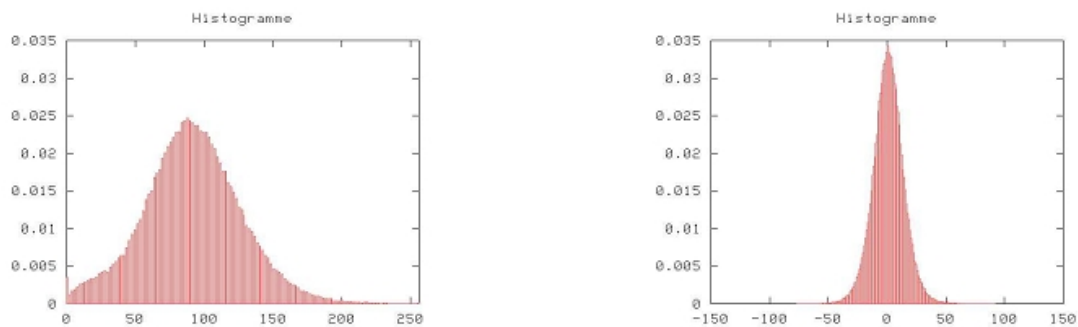


Figure II.3 Histogramme des niveaux de gris de l'image originale (gauche) et de l'erreur de prédiction (droite)

Le nombre de pixel voisins utilisés pour la prédiction définit l'ordre de prédiction. Ces pixels peuvent être choisis de nombreuses manières. En considérant un balayage de l'image (Figure II.4), on aura un voisinage qui servira à la prédiction.

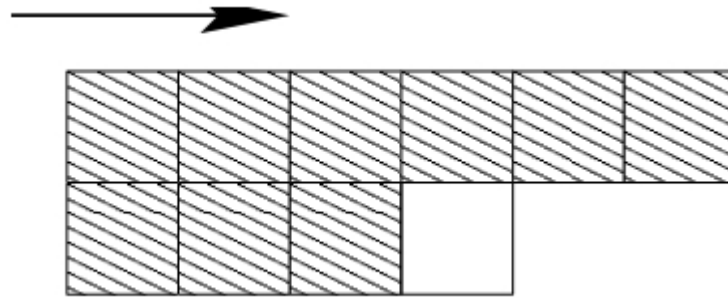


Figure II.4 Balayage par ligne d'une image

❖ Prédiction linéaire

Pour une image isolée, la prédiction linéaire se réalise sur la région R selon la formule

$$x(m, n) = \sum_{(i, j) \in R} a(i, j) x(m - i, n - j) \quad (\text{II.8})$$

En pratique, le nombre de pixels formant la région qui sert de support à la prédiction est assez faible de l'ordre de quelques unités.

Les méthodes prédictives permettent une mise en œuvre facile et conduisent à de bons taux de compression. Elles sont efficaces pour les images dont les évolutions temporelles ou spatiales sont petites.

II.6.2.5. Méthodes hybrides

Le terme hybride fait référence aux techniques qui combinent le codage prédictif et le codage par transformée.

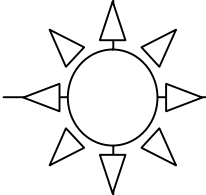
Dans le cas des images fixes, on effectue une transformation à une dimension le long des lignes et ensuite une prédiction le long des colonnes. Pour les images animées, on effectue une combinaison entre une transformation bidimensionnelle dans le domaine spatial et une prédiction le long de la composante temporelle pour exploiter la redondance temporelle du signal d'image. Le codeur hybride regroupe les avantages des deux techniques qui le composent.[17]

II.7. Conclusion

Toutes les méthodes présentées dans ce chapitre produisent un flux de données permettant de reconstruire d'une part une copie exacte des données originales et d'autre part une copie dégradée des données compressées.

De ce qui précède tout au long de ce chapitre et après avoir exposé les différentes méthodes utilisés dans la compression d'image, on va faire une description détaillée dans le chapitre suivant du standard JPEG, qui a regroupé plusieurs méthodes, dans le quel notre travail s'articule.

Chapitre III : La norme JPEG



III.1. Introduction

III.2. Principe générale de la compression des images

III.3. La norme JPEG

III.4. La transformation DCT

III.5. Compression basée sur la DCT (cas du JPEG)

III.6. Conclusion

III.1. Introduction

Les développements technologiques et les exigences des utilisateurs conduisent à des quantités de données toujours plus importantes ; ainsi, de puis les technologies de l'information, le problème de l'exploitation optimale des voies de communication et des capacités de stockage est toujours resté un sujet d'actualité.

De nos jours, l'image joue un rôle très important comme un moyen synthétique de présentation et comme outil d'observation, de reconnaissance et d'analyse. Pour cette raison de multiples études ont été menées sur les algorithmes de compression d'image ; dans le but de mettre les données sous un format tel qu'elles occupent moins de volume. Une fois les données sont compressées, elles ne sont plus directement accessibles, et il est nécessaire de les décompresser pour qu'elles redeviennent intelligibles.

III.2. Principe général de la compression des images:

Le schéma fonctionnel de la compression est présenté dans la figure ci-dessous :

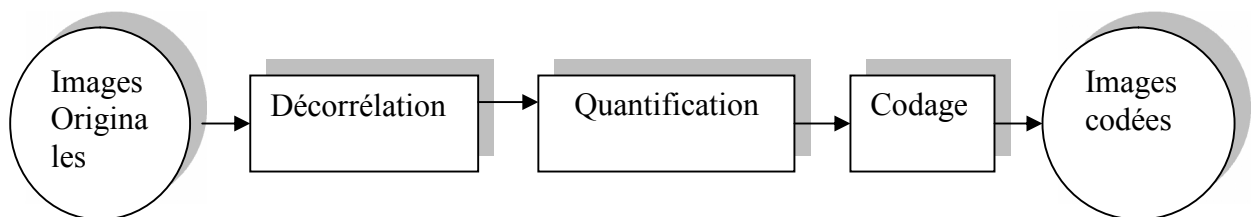


Figure III.1 : Schéma d'un codeur source

III.2.1. Décorrélacion

La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte sur l'image. On essaie donc de tirer une partie de cette corrélation, pour réduire le volume d'information en effectuant une opération de décorrélacion des pixels.

La décorrélacion consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible.

III.2.2. Quantification

La quantification désigne dans sa forme la plus simple, le nombre des bits nécessaire pour stocker une image. C'est une application surjective ce qui veut dire que plusieurs points possèdent la même image (niveau quantifié) et c'est cet aspect qui la rend non réversible. [18]

La quantification peut être appliquée sur un seul élément (quantification scalaire) ou sur un groupe (vecteur) d'éléments (c'est la quantification vectorielle).

Mathématiquement la quantification peut être formulée ainsi :

Soit « E » un espace vectoriel où « d » est la dimension de « E » et R une relation d'équivalence sur E.

Soit $A = \{y_0, y_1, \dots, y_{N-1}\}$ un ensemble de tous les représentants y_i des classes d'équivalence « Si » de $\frac{E}{R}$

La quantification de $X \in E$ est l'application $Q(x)$ définie comme suit :

$$Q(x): \forall i \text{ si } x \in Si \text{ alors } x \rightarrow y_i \text{ avec } y \in Si$$

III.2.2.1. Quantification scalaire

C'est le cas où « d » définie précédemment est égale à « 1 » d'où $Q(x)$ est appelée quantification scalaire où QS et y_i sont appelées niveaux de quantification et 'Si' sont appelées intervalles de quantification. La QS consiste à quantifier les niveaux de gris des pixels les uns après les autres. Le principe de construction du dictionnaire est comme suit :

- Le dictionnaire est construit et optimisé par l'algorithme Lloyd ; on le définit par : La forme et les dimensions de pixels
- La taille du dictionnaire.

Le principe du codage est d'attribuer à chaque amplitude d'entrée la plus proche valeur du dictionnaire scalaire.

L'importance de la quantification scalaire (QS), du point de vue compression, tient à plusieurs raisons parmi lesquelles on peut citer :

- L'intérêt général de la quantification, dont une application particulière est la conversion analogique numérique.
- La simplicité qui permet des traitements en temps réel de l'application difficile.
- La QS est généralement à la base d'autres méthodes.

III.2.2.2. Quantification vectorielle

C'est le cas où « d » défini précédemment est ($d < 1$) d'où $Q(x)$ est appelé quantification vectorielle ou 'QV' et y_i sont appelées représentants et les S_i sont appelées zones ou régions.

Quantifier un phénomène physique consiste à le remplacer par un autre de même nature. La QV consiste à remplacer un bloc complet de pixels par la représentation binaire du numéro du bloc le plus ressemblant établi appelé dictionnaire.

Ce qui sera stockée ou transférée est la liste des étiquettes. Le bloc déclaré le plus ressemblant au bloc original est celui qui, parmi tous les blocs du dictionnaire, conduit à la plus petite erreur.

Le principe de construction du dictionnaire est construit et optimisé par l'algorithme LBG proposée par Linde, Buzo, Gray et il est similaire à celui de la QS, sauf qu'ici on parle d'un bloc et non pas d'un point de l'image. [19]

La quantification vectorielle se décompose en deux applications : codeur, décodeur.

➤ **Codeur**

Le rôle du codeur consiste, pour tout vecteur x_i du signal en entrée à rechercher dans le dictionnaire Y le code vecteur (y_j) ainsi sélectionnée qui sera transmise ou stockée. C'est à ce niveau donc que s'effectue la compression.

➤ **Décodeur**

Il dispose d'une réplique du dictionnaire et consulte celui-ci pour fournir le code vecteur d'indice correspondant à l'adresse reçue. Le décodeur réalise l'opération de décompression.

III.2.3. Codage

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

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

III.3. La norme JPEG

III.3.1. Introduction

A la fin des années 1970 et au début des années 1980, la recherche a débuté sur de nouveaux types de compression d'images promettant de surclasser largement les techniques de compression conventionnelles.

A la fin des années 1980, ce travail a commencé à trouver des applications commerciales pour le traitement d'image sur les systèmes de bureau, principalement sous la forme de cartes d'extension pour les stations de travail UNIX et macintosh. Ces cartes

étaient capables d'effectuer une compression non conservative d'images à des taux aussi élevés que 95% sans dégradation visible de leur qualité.

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 leur stockage sur les supports numériques.

Elle a été réalisée dans la perspective de couvrir les applications les plus diversifiées en tenant compte des contraintes réalistes par rapport aux applications les plus visibles : publication, transmission, banques d'images.

III.3.2. Les méthodes de la norme de compression JPEG:

Les techniques définies par la norme JPEG se divisent en deux classes :

Les méthodes de compression avec pertes qui sont basées sur la TCD suivie d'une quantification et d'un codeur anthropique. La seconde classe, concerne les processus de codage sans pertes, cette classe de codeurs n'est pas basée sur la TCD mais sur le codage MICD suivi d'un codage anthropique. Pour les méthodes avec pertes, quatre codeurs ont été spécifiés : un codage de base où l'image compressée puis décompressée n'est plus identique à l'image originale, ce processus utilise la TCD et un codage de Huffman.

Les trois autres types de codage sont une extension de codage de base. Ils diffèrent de codage de base principalement par le codage anthropique (HUFFMAN + RLE) en utilisant un codage arithmétique ou par restitution progressive de l'image.[20]

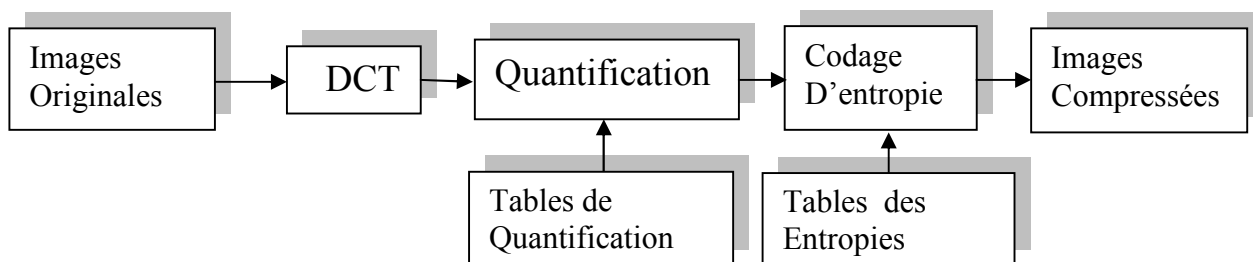


Figure III.2: principe de l'algorithme JPEG avec pertes.

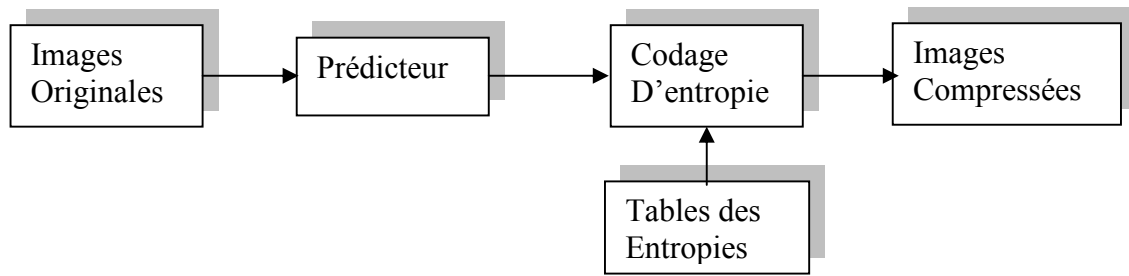


Figure III.3: principe de l'algorithme JPEG sans pertes

III.3.3 La compression non conservative "JPEG"

La compression non conservative est fondamentalement différente de la compression conservative elle accepte une légère perte de données afin de la compression.

La compression non conservative est généralement utilisée sur des données analogiques stockées de façon numérique, avec comme principales applications les fichiers d'images. L'algorithme de compression non conservative JPEG opère en trois étapes successives. Ces trois étapes sont combinées pour un compresseur performant, capable de compresser des images à nuance continue à moins de 10% de leur taille originelle tout en perdant un peu, si perte il y a, de leur fidélité originale.

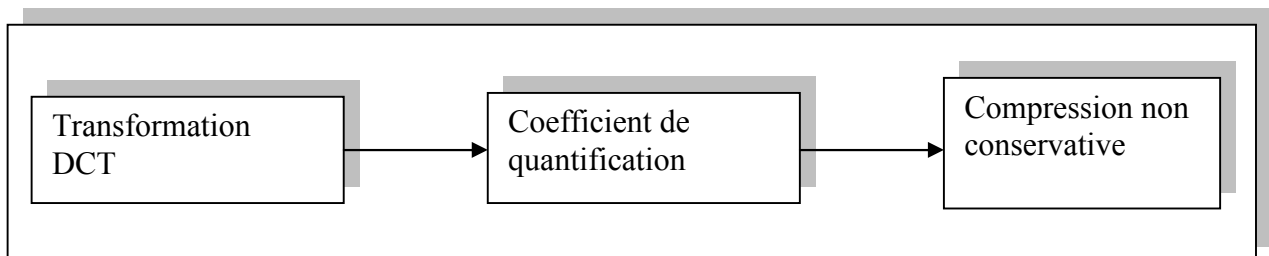


Figure III.4: La compression non conservative "JPEG"

III.4. La Transformation "DCT"

III.4.1. Transformations linéaires

Les transformations linéaires agissent directement sur le signal et le transforment afin de révéler des caractéristiques propres à une compression efficace. Les transformations linéaires présentant ces qualités sont nombreuses mais la DCT, qui appartient à la famille des transformées de Fourier, présente un compromis nettement supérieur à ses concurrentes, ce qui lui a valu d'être retenue dans un grand nombre de normes dont la norme JPEG.

Prenons l'exemple d'une matrice de pixels. Cette matrice est immédiatement divisée en blocs de 8x8 pixels. Il est alors possible de construire à l'aide de ces blocs un espace vectoriel muni d'opérations internes et externes. On peut donc considérer chaque bloc comme un vecteur, B dont les pixels n_0, n_1, \dots, n_{64} sont les composantes. La transformation linéaire se fait donc sur ces blocs.

La transformation linéaire s'écrit :

$$f(B) = \begin{pmatrix} M \\ n(u) \\ M \end{pmatrix} = A \begin{pmatrix} M \\ n(i) \\ M \end{pmatrix} \quad (\text{III.1})$$

Où A est la matrice associée à F .

Après transformation les blocs perdent leur nature initiale (ils ne représentent plus alors une image pour l'œil). Une transformation linéaire n'est donc que provisoire et suppose que son inverse soit effectué avant restitution. Le problème soulevé est l'intérêt d'une telle transformation pour la compression.

La DCT génère, quelles que soient les valeurs des $n(i)$, des valeurs de $n(u)$ très différentes les unes des autres. Certaines sont devenues très grandes, d'autres très petites. Une annulation arbitraire de ces dernières n'entraîne ainsi qu'une distorsion réduite. La compression est alors le fruit d'un abandon d'un certain nombre de composantes. Il appartient au codeur en fonction de la distorsion maximale autorisée, ou bien en fonction du taux de compression demandé, de décider du nombre de composantes à abandonner.

III.4.2. Transformation en cosinus discrète bidimensionnelle (DCT)

Le principe de la DCT (Discrete cosine transform) est de transformer (pour chaque valeur Y, I, Q du signal fonction de x et y) la matrice $f(x,y)$ en matrice des "fréquences" $F(u,v)$ permettant de révéler les caractéristiques propres à une compression efficace. Elle prend donc un ensemble de points d'un domaine spatial et les transforme en une représentation équivalente dans le domaine fréquentiel. Après compression, la transformation inverse permet de revenir à l'image en (x, y) . La DCT est une transformation linéaire permettant de disproportionner certains coefficients transformés de telle sorte que leur abandon n'entraîne pas de distorsion significative après reconstitution.

La transformation est réalisée, non pas sur l'image entière, mais généralement sur des fractions de l'image, des blocs de 8 x 8 pixels (matrice 8 x 8 en x, y). D'une part parce que le signal de l'image doit être présenté sur une matrice carrée et d'autre part parce que cela

s'avérerait trop complexe et trop long. En augmentant la taille des blocs nous obtiendrions une meilleure compression mais au détriment du temps de traitement. En fait, pour pouvoir utiliser la DCT, ces blocs sont dupliqués symétriquement par rapport à l'axe des x et à l'axe des y, pour produire une matrice 16 x16 en x, y. Plus généralement, si N désigne le nombre d'"échantillons" x et y (N=8 dans le cas général), la DCT fait correspondre à chaque valeur de Y(x, y) une valeur de F(u, v) donnée par la formule (III.2)

$$F(u, v) = \frac{2}{N} C(u) \cdot C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} Y(x, y) \cdot \cos\left(\frac{\pi}{N} u \left(x + \frac{1}{2}\right)\right) \cdot \cos\left(\frac{\pi}{N} v \left(y + \frac{1}{2}\right)\right) \quad (\text{III.2})$$

$$\text{Avec : } 0 \leq u \leq N-1 \quad \text{et} \quad 0 \leq v \leq N-1$$

A la matrice initiale N x N en x, y des valeurs de Y, la transformation fait correspondre une matrice N x N en u, v des "fréquences" du signal. La transformation inverse permettant de revenir à l'image est donnée par la formule (III.3) :

$$Y(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) \cdot C(v) F(u, v) \cdot \cos\left(\frac{\pi}{N} u \left(x + \frac{1}{2}\right)\right) \cdot \cos\left(\frac{\pi}{N} v \left(y + \frac{1}{2}\right)\right) \quad (\text{III.3})$$

Dans ces formules, on a : $c(0) = (2)^{-1/2}$

$c(w) = 1$ pour $w = 1, 2, \dots, N-1$

La matrice F transformée par DCT présente la propriété de regrouper les valeurs les plus élevées dans le coin supérieur gauche de la matrice (les valeurs devenant d'autant plus faibles que l'on s'approche du coin inférieur droite, donc que l'on se situe dans les "hautes fréquences"). Ainsi le maximum d'informations sur l'image se trouve concentré sur la partie supérieure gauche de la matrice et la compression de l'image sera facilitée.

La DCT est donc effectuée sur chaque matrice 8x8 de valeurs de pixels, et elle donne une matrice 8x8 de coefficients de fréquence : l'élément (0,0) représente la valeur moyenne du bloc, les autres indiquent la puissance spectrale pour chaque fréquence spatiale. La DCT est conservatrice si on ne prend pas en compte les erreurs d'arrondis de qualité introduit.

Après DCT et transformation inverse, lorsque $N=8$ (comme c'est le cas dans la plupart des applications pratiques), on obtient 64 transformées inverses (correspondant aux valeurs de (u, v)), des 64 points élémentaires (correspondant aux valeurs de (i, j)). L'image est alors décomposée en une "combinaison linéaire" de ces images de base :



Figure III.5 : Images de la base de l'espace transformé DCT

On remarquera que entre les blocs calculés et affichés sur un écran d'ordinateur et ces mêmes blocs représentés ci dessus sur du papier, un certain nombre d'opérations d'imprimerie ont apporté des dégradations importantes. L'effet final est alors difficile à prévoir, certains niveaux de gris pouvant se confondre.

III.4.3. Propriétés importantes de la DCT en compression d'image

III.4.3.1. Décorrélacion

La DCT permet de réduire ou d'éliminer la redondance inter pixel, les pixels sont décorrélés afin de réduire la variance des pixels voisins dans l'image ce qui permet une compression efficace en codant chaque pixel indépendamment.

III.4.3.2. Concentration des coefficients

La DCT est très efficace pour des images fortement corrélées du fait qu'elle permet de concentrer les coefficients qui représentent les basses fréquences dans une seule partition de la matrice image, cela permet la séparation des fréquences basses des fréquences hautes, et si c'est une image faiblement corrélée, les coefficients sont concentrés dans plusieurs différentes partitions de la matrice image.

III.4.3.3 Symétrie, séparabilité, et orthogonalité :

La DCT peut être séparée comme suit :

$$F(u, v) = \frac{2}{N} C(u) \cdot C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} Y(x, y) \cdot \cos\left(\frac{\pi}{N} u \left(x + \frac{1}{2}\right)\right) \cdot \cos\left(\frac{\pi}{N} v \left(y + \frac{1}{2}\right)\right) \quad (\text{III.4})$$

$$\text{Avec :} \quad 0 \leq u \leq N-1 \quad \text{et} \quad 0 \leq v \leq N-1$$

De l'équation précédente on peut sortir la propriété de symétrie qui va nous conduire à une expression simplifiée du calcul de la DCT:

$$Y = F y F^t \quad (\text{III.5})$$

Où : « y » est la matrice NxN et « Y » sa transformée DCT

« F » est la matrice NxN, ces éléments f (i, j) se calculent par l'équation suivante :

$$F(i, j) = \alpha(j) \sum_{j=0}^{N-1} \cos\left[\frac{\pi(2j+1)i}{2N}\right] = \alpha(j) \sum_{j=0}^{N-1} \cos\left[\frac{\pi}{N} i \left(j + \frac{1}{2}\right)\right] \quad (\text{III.6})$$

A ce stade, on peut conclure avec la propriété d'orthogonalité : va nous permettre d'avoir l'IDCT de la forme suivante : $F^{-1} = F^T$

$$y = F^{-1} Y F^{-1} = F^{-1} F y F^T F^{-1} = y$$

On remarque qu'on obtient la matrice image originale au final. Ces propriétés réduisent considérablement les calculs dans les algorithmes de compression d'image du fait que ces éléments seront déjà calculés à l'avance (calcul de la matrice C).

III.4.4. Avantages et inconvénients de la DCT

L'avantage essentiel de la DCT est le taux de compression important pour une qualité donnée. La valeur de ce taux est difficile à évaluer compte tenu du fait qu'il dépend de la nature des signaux et de leurs qualités initiales (quantification, colorimétrie). Notons quand même qu'il est possible d'obtenir des taux voisins de 15 sans que l'on puisse observer des différences à l'œil nu avec des images de type photographique ou sur des écrans de très bonne qualité.

L'inconvénient majeur de la DCT est le découpage des signaux en blocs. Leur silhouette apparaît à partir d'un certain taux de compression qui dépend de la nature du signal. Cet effet généralement invisible apparaît à l'œil nu, dans le cas des images photographiques, dès que le taux de compression atteint quelques dizaines. Aujourd'hui de nombreux testeurs se penchent sur des techniques visant à accélérer la DCT, par traitement mathématique à virgule fixe ; chaque cycle gagné sur le temps pris pour calculer la DCT vaut son pesant d'or.

III.5. Compression basée sur la DCT (cas du JPEG)

Nous donnerons d'abord une idée relativement précise de l'algorithme léger JPEG. Ce nom (joint photographique expert group) provient du groupe d'experts internationaux, elle a été établi en 1991, c'est la norme que nous utilisons actuellement [15].

En fait, la DCT (discret cosin Transform) qui est le cœur de la méthode, a été proposée en 1974 par le professeur Rao de l'université du Texas en 1974.

La DCT peut-être vue comme une version discrète de Cosinus de Fourier. Par ses propriétés, elle est plus proche de la transformée optimale "KLT" (Karhuen- Loève Transform) calculée en utilisant une transformée de Fourier rapide "FFT" (Fast Fourier transform). A la différence de la FFT, la DCT produit des coefficients à valeur réelle et fournit une meilleure approximation du signal avec moins de coefficients. La norme JPEG de 91 décrit le format des données compressées et le schéma de codage et celui de la Figure III.4.

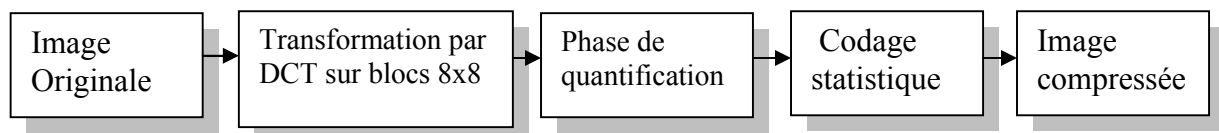


Figure III.6. Le schéma de codage

Nous allons prendre un exemple et suivre les étapes du schéma de principe.

III.5.1. Première étape : DCT

Une fois la DCT calculée sur un bloc nous obtenons une matrice carrée des valeurs pour chacune des fréquences. Les valeurs de la matrice DCT ont été arrondies à l'entier le plus proche. La composante (0, 0) est le coefficient continu de la matrice ; c'est un nombre proportionnel à la somme de toutes les valeurs du signal. Les autres valeurs de la matrice représentent les « écarts » par rapport à ce nombre. Les composantes trouvées en

ligne et en colonne zéro transportent une information plus utile que les composantes hautes fréquences. Lorsque nous nous éloignons des composantes continues de l'image, nous pouvons constater que les coefficients ont tendance à avoir de faibles valeurs mais aussi qu'ils deviennent moins importants pour la description de l'image. Cela traduit le fait que l'information effective d'une image soit toujours concentrée dans les basses fréquences. L'entrée de la fonction DCT est constituée de valeurs de pixels sur huit bits mais les valeurs qui en sortent sont dans l'intervalle $[-1024 ; 1024]$ et occupent onze bits. Cette étape non conservatrice du processus : la quantification.

Voici un exemple, où on a une étape conservatrice qui prépare les données au prêt d'application de la DCT à une matrice de 8x8 pixels à 256 niveaux de gris.

Matrice des pixels d'entrée :

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

Fig. III.7. Matrice du bloc original

❖ Quel est l'intérêt de cette transformation?

On observe que les coefficients de forte valeur absolue sont situés en haut et à gauche. L'importance des coefficients pour la reconstitution de l'image diminue quand on se déplace en diagonale du haut à gauche vers le bas à droite.

Nous allons passer à l'étape suivante, celle de la quantification.

III.5.2. Deuxième étape: Quantification

La quantification représente la phase non conservatrice du processus de compression JPEG. Elle permet, moyennant une diminution de la précision de l'image, de réduire le nombre de bits nécessaires au stockage. Pour cela, elle réduit chaque valeur de la matrice DCT en la divisant par un nombre (quantum), fixé par une table (matrice 8 x 8) de quantification :

Valeur quantifiée $(i, j) = \text{valeur DCT}(i, j) / \text{quantum}(i, j)$, arrondie à l'entier le plus proche. Ultérieurement, 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. La valeur du quantum peut être d'autant plus élevée que l'élément correspondant de la matrice DCT contribue peu de la qualité de l'image, donc qu'il se trouve éloigné du coin supérieur gauche ($i=j=0$). C'est pourquoi les matrices de quantification comportent généralement des valeurs constantes selon des diagonales ascendantes $(i, j, i-1, j+1)$, mais croissantes d'une diagonale de la méthode suivante : cet accroissement constitue le pas du quantum, ou le "facteur de qualité", car la perte de précision sera d'autant plus grande que ce pas sera élevé.

Les nombreux tests réalisés ont conduit à retenir en pratique des facteurs de qualité compris entre 1 (l'image reste excellente) et 25 (dégradation maximale acceptable). Il est important d'observer que, même avec des pas de quantum faible, la quantification conduit à une valeur nulle nombre d'éléments situés dans le coin inférieur droit de la matrice : ces éléments ne seront pas restitués. Par la déquantification, d'où le caractère non conservateur de cette opération. Un grand nombre de schémas pourraient être choisis pour définir les valeurs de la matrice de quantification. Ces schémas peuvent être testés de deux manières différentes, soit en mesurant l'erreur mathématique entre l'image d'entrée et celle de sortie après décompression, soit en jugeant l'effet de la décompression sur l'œil humain. Bien que la norme JPEG n'impose aucune contrainte sur la matrice de quantification, l'ISO a développé un ensemble standard de valeurs de quantification fournies aux programmeurs de code JPEG. Par exemple en choisissant des tailles de pas de quantification très élevées, le résultat est un taux de compression important pour une qualité d'image médiocre. En choisissant des tailles de pas petites la qualité de l'image reste excellente et les taux de compression n'auront rien d'extraordinaire.

III.5.2.1. Choix de la matrice de quantification:

C'est à cette étape que nous allons perdre de l'information et nous allons la perdre d'une manière « astucieuse » parce que le pas de quantification dont dépend la précision de l'image restituée va dépendre de la position de la valeur dans la matrice.

Nous allons prendre un pas relativement petit pour les valeurs importantes (en haut à gauche) et prendre un pas de plus en plus grand au fur et à mesure qu'on descend

vers le bas et la droite de la matrice. L'ensemble des pas qui vont être utilisés constitue ce que l'on appelle une matrice de quantification. Certaines ont été construites en fonction de critères psycho-visuels, pour ce soir nous allons en fabriquer une avec une petite formule :

$$Q(i,j) = 1 + (1 + i + j) \times Fq \quad (\text{III.7})$$

pour les matheux, on indiquera une formule un peu plus complexe, permettant d'obtenir un grand nombre de matrice différente[20]

$$Q(i,j) = 1 + (1 + \mu (i^n + j^n)) \times Fq \quad (\text{III.8})$$

Par souci de simplification, on a pris ici $\mu = n = 1$ et $Fq = 2$. On peut bien entendu compliquer.

Nous prendrons $Fq = 2$. Il s'agit d'un facteur de qualité, c'est celui que vous modifiez quand vous choisissez la qualité de la restitution dans un logiciel comme Photo shop.

Voici la matrice de quantification que nous obtenons :

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

Fig.III.8. Matrice de quantification

L'application de cette table à la matrice DCT aboutit à une matrice DCT quantifiée.

Matrice DCT avant quantification :

| | | | | | | | |
|-----|-----|-----|-----|-----|----|----|----|
| 92 | 3 | -9 | -7 | 3 | -1 | 0 | 2 |
| -39 | -58 | 12 | 17 | -2 | 2 | 4 | 2 |
| -84 | 62 | 1 | -18 | 3 | 4 | -5 | 5 |
| -52 | -36 | -10 | 14 | -10 | 4 | -2 | 0 |
| -86 | -40 | 49 | -7 | 17 | -6 | -2 | 5 |
| -62 | 65 | -12 | -2 | 3 | -3 | -2 | 0 |
| -17 | 14 | -36 | 17 | -11 | 3 | 3 | -1 |
| -54 | 32 | -9 | -9 | 22 | 0 | 1 | 3 |

Fig.III.9. Matrice DCT avant quantification

Matrice DCT après quantification :

| | | | | | | | |
|-----|----|----|---|----|---|---|---|
| 31 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| -52 | -8 | 1 | 2 | 0 | 0 | 0 | 0 |
| -12 | 7 | 0 | 1 | 0 | 0 | 0 | 0 |
| -6 | -3 | -1 | 1 | -1 | 0 | 0 | 0 |
| -8 | -3 | 3 | 0 | 1 | 0 | 0 | 0 |
| -5 | 4 | -1 | 0 | 0 | 0 | 0 | 0 |
| -1 | 1 | -2 | 1 | 0 | 0 | 0 | 0 |
| -3 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |

Fig.III.10. Matrice DCT après la quantification

III.5.3. Troisième étape: codage

L'étape finale est le codage des images quantifiées. La première modifie le coefficient continu en position (0,0) d'une valeur absolue en une valeur relative. Puisque des blocs adjacents d'une image montrent un haut degré de corrélation, le codage des éléments continus par une différence avec les éléments continus précédent produit généralement un nombre très petit. En suite, les coefficients de l'image sont arrangés en une "séquence zig-zag". Puis ils sont codés en utilisant deux mécanismes différents. Le premier est un codage des longueurs de séquence de valeurs nulles. Le second est ce que JPEG appelle le "codage d'entropie", Il implique la sortie des codes de coefficient en utilisant soit un codage de huffman soit un codage arithmétique en fonction du choix du programmeur.

III.5.3.1. Balayage en Zigzag

Une des raisons de l'efficacité de l'algorithme JPEG est qu'un grand nombre de coefficients de l'image DCT sont tronqués à des valeurs zéro pendant l'étape de quantification. Il y a tant de valeur qui sont mises à zéro que le comité JPEG a choisi de traiter les valeur nulles différemment des autres valeur de coefficient au lieu de se baser sur un codage de huffman ou arithmétique pour compresser les valeurs zéros, on utilise l'algorithme de codage des longueurs de séquences (RLE). Un code simple est développé, donnant un compte des consécutives de zéro dans l'image. Puisque plus de la moitié des coefficients sont réduits à zéro dans l'image, on a l'opportunité d'une compression de premier plan. Une manière d'augmenter la longueur des séquences est de réordonner les coefficients en une Séquence zig-zag. Au lieu de compresser les coefficients par ordre de ligne, comme le ferait probablement un programmeur, l'algorithme JPEG se déplace parmi les blocs suivant une diagonale, choisissant en premier les valeurs hautes, puis continuant par les valeurs les plus faibles.

Le chemin réel de la séquence ZIG-ZAG est montré dans la table suivante:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
| 1.0 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 |
| 2.0 | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | 2.6 | 2.7 |
| 3.0 | 3.1 | 3.2 | 3.3 | 3.4 | 3.5 | 3.6 | 3.7 |
| 4.0 | 4.1 | 4.2 | 4.3 | 4.4 | 4.5 | 4.6 | 4.7 |
| 5.0 | 5.1 | 5.2 | 5.3 | 5.4 | 5.5 | 5.6 | 5.7 |
| 6.0 | 6.1 | 6.2 | 6.3 | 6.4 | 6.5 | 6.6 | 6.7 |
| 7.0 | 7.1 | 7.2 | 7.3 | 7.4 | 7.5 | 7.6 | 7.7 |

Fig.III.11.Le chemin de la séquence ZIG-ZAG

III.5.3.2 Codage d'entropie

❖ Principe:

L'idée est la suivante : nous avons l'habitude de coder toutes les valeurs que nous avons à stocker avec le même nombre de bits, généralement un ou plusieurs octets. Nous utilisons un octet par caractère quand nous travaillons avec du texte, un ou deux octets par pixel et par couche quand nous travaillons avec des images. Est-ce la meilleure solution ?

La théorie nous apprend que non. Il est en effet plus efficace d'utiliser des codes plus courts pour des valeurs fréquentes et de réserver des codes plus longs pour les valeurs moins fréquentes. L'intérêt de cette méthode dépendra bien entendu de la distribution des valeurs à coder telle qu'on peut

L'étudier en construisant l'histogramme des valeurs à coder. Pour une image, Photoshop sait faire ça très bien.

En fait l'examen des histogrammes de quelques images comme celles que nous utilisons journalièrement est assez encourageant : il y a toujours dans les images naturelles des variations de fréquence significatives.

On va donc s'intéresser aux VLC (Variable Length Code), on dit aussi : codage d'entropie.

Nous voyons bien l'intérêt de la méthode mais la mise en œuvre est un peu plus délicate.

❖ Exemples et analyse

Prenons un petit exemple. Supposons que nous ayons à traiter un message qui ne contient que 4 caractères que nous nommerons A, B, C et D. Et que les fréquences de ces caractères dans notre message soient les suivantes :

A : 60 % B : 30 % C : 5 % D : 5 %

On peut imaginer le code suivant :

A : 0 B : 00 C : 01 D : 10

Il nous permettra assurément de gagner de la place mais si nous recevons le message suivant

: 00001, Comment l'interpréter ?

4 A ? 2 B ? 2 A et 1 B ? 1 B et 2 A ? Ça ne marche pas ! Nous venons de découvrir

Le problème de la synchronisation.

Comment le résoudre ?

Utiliser des caractères séparateurs ? C'est contradictoire avec l'objectif de compression.

La solution c'est le **VLC préfixé**.

Un code est préfixé s'il n'est le début d'aucun autre. Le VLC que nous avons tenté d'utiliser tout à l'heure n'était pas préfixé. Revenons à notre exemple et essayons de corriger. A : 0 B : 10 C : 110 D : 111 Ça marche ! Est-ce la solution optimale ? Comme on l'a obtenue par tâtonnement, on n'en sait rien mais rassurez-vous la théorie permet d'affirmer que **oui**.

Nous allons maintenant prendre un exemple un peu plus complexe et présenter une méthode permettant de déterminer un VLC efficace (même s'il n'est pas toujours optimal).

Supposons que nous voulions transmettre le message suivant :

LE PRESIDENT EST ENTRE DANS LA SALLE

Il compte 36 caractères et occupe dans un logiciel comme Xpress ou WordPerfect 36 octets. Nous allons compter les occurrences des différents caractères de l'alphabet utilisés et les classer par fréquences décroissantes, nous obtenons la liste suivante :

| | | |
|------------|-----|-----|
| E : 7 | T:3 | I:1 |
| Espace : 6 | A:3 | |
| L : 4 | R:2 | |
| S : 4 | D:2 | |
| N : 3 | P:1 | |

Nous allons maintenant regrouper les caractères en deux groupes dont les fréquences d'apparition sont aussi proches que possibles puis diviser chacun de ces groupes de la même façon jusqu'à parvenir à chacune des fréquences de départ (il est équivalent de travailler avec le nombre d'occurrences ou les fréquences). Voici le développement du calcul et le codage affecté :

| | | | |
|------------|------|-------|--------|
| E : 7 | 00 | D : 2 | 1110 |
| Espace : 6 | 010 | P : 1 | 11110 |
| L : 4 | 011 | I : 1 | 111110 |
| S : 4 | 100 | | |
| N : 3 | 1010 | | |
| T : 3 | 1011 | | |
| A : 3 | 1100 | | |
| R : 2 | 1101 | | |

On numérote les branches de l'arborescence en binaire et on parcourt chaque branche

Pour déterminer les codes qui figurent dans la colonne de droite. On peut vérifier que l'on obtient bien un VLC préfixé. Mesurons son efficacité. Nous sommes partis d'un message de 36 caractères soit $36 \times 8 = 288$ bits, si l'on avait codé chaque caractère sur 4 bits (c'était suffisant), on utilisait 144 bits. En utilisant le codage que nous venons de réaliser nous utiliserons :

$$(2 \times 7) + (3 \times 6) + (3 \times 4) + (3 \times 4) + (4 \times 3) + (4 \times 3) + (4 \times 3) + (4 \times 2) + (4 \times 2) + (5 \times 1) + (5 \times 1) \\ = 14 + 18 + 12 + 12 + 12 + 12 + 12 + 8 + 8 + 5 + 5 = 118 \text{ bits}$$

Le taux de compression par rapport à la représentation sur 4 bits est $T_c = 144/118 = 1,22$.

III.5.4. La décompression

Pour la décomposition, il faut d'abord indiquer la matrice de quantification utilisée. Le schéma général de la décompression est donné par la Figure suivante :

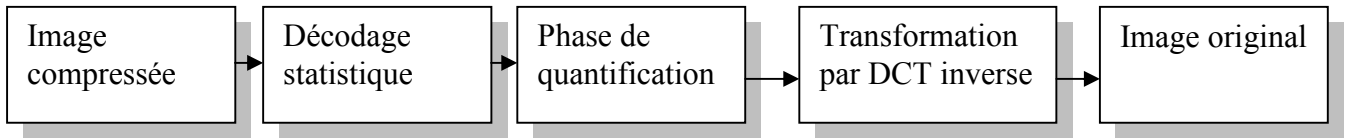


Fig.III.12. Schéma général de la décompression

Nous allons comme auparavant suivre les étapes du schéma. Après le décodage statistique nous retrouvons la matrice de la figure III.12.

| | | | | | | | |
|----|----|----|----|----|----|---|---|
| 24 | -7 | 2 | -3 | 0 | -2 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| -4 | -1 | 1 | 0 | 0 | -1 | 0 | 0 |
| 0 | 0 | -1 | 1 | -1 | 0 | 0 | 0 |
| 0 | 0 | 3 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | -2 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Fig.III.13. Matrice après le décodage

Nous allons la déquantifier en multipliant chaque terme par le coefficient de la matrice de quantification correspondant. On obtient la matrice de la figure III.13

| | | | | | | | |
|-----|-----|-----|-----|---|---|---|---|
| 90 | 0 | -7 | 0 | 0 | 0 | 0 | 0 |
| -35 | -56 | 9 | 11 | 0 | 0 | 0 | 0 |
| -84 | 54 | 0 | -13 | 0 | 0 | 0 | 0 |
| -45 | -33 | 0 | 0 | 0 | 0 | 0 | 0 |
| -77 | -39 | 45 | 0 | 0 | 0 | 0 | 0 |
| -52 | 60 | 0 | 0 | 0 | 0 | 0 | 0 |
| -15 | 0 | -19 | 0 | 0 | 0 | 0 | 0 |
| -51 | 19 | 0 | 0 | 0 | 0 | 0 | 0 |

Fig.III.14 Matrice déquantifiée

Nous allons maintenant appliquer la DCT inverse qui est donnée par:

$$n(i, j) = \frac{2}{M} \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} C(u)C(v) \cos \left[\frac{\pi u}{2M} (2i + 1) \right] \cos \left[\frac{\pi v}{2M} (2j + 1) \right] f(u, v) \quad (\text{III.9})$$

Avec: $(i, j) \in [0, M - 1] * [0, M - 1]$ et $C(0) = \frac{1}{\sqrt{2}}$ et $\forall \alpha C(\alpha) = 1$

On ajoute 128 à chaque terme, ce qui permet d'obtenir le bloc décompressé de la Figure II.14

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 146 | 141 | 152 | 138 | 139 | 155 | 161 | 155 |
| 148 | 152 | 140 | 147 | 140 | 148 | 167 | 179 |
| 169 | 155 | 136 | 167 | 163 | 162 | 152 | 172 |
| 158 | 145 | 156 | 160 | 152 | 155 | 136 | 160 |
| 122 | 148 | 156 | 148 | 145 | 136 | 147 | 162 |
| 136 | 167 | 140 | 155 | 155 | 140 | 136 | 162 |
| 138 | 156 | 123 | 167 | 162 | 144 | 130 | 147 |
| 158 | 155 | 136 | 155 | 152 | 147 | 147 | 126 |

Figure III.15 matrice après IDCT

III.6. Conclusion

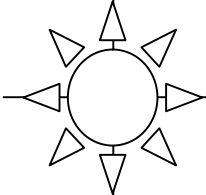
Les méthodes déjà utilisées couramment sont efficaces et sophistiquées (Huffman, LZW, JPEG) et utilisent des théories assez complexes, les méthodes émergentes sont prometteuses mais nous sommes loin d'avoir épuisé toutes les pistes de recherche. Les méthodes du futur pourront sans doute s'adapter à la nature des données à compresser et utiliseront l'intelligence artificielle. Pour en terminer on citera les principales caractéristiques envisagées pour la norme JPEG-2000:

- concernera les images à 2 niveaux, les images en niveaux de gris et les images en couleurs.
- pourra traiter de très grandes images.
- utilisera un algorithme unique pour toutes les images.
- sera compatible avec des transmissions imparfaites et efficaces pour les images de synthèse.
- sera efficace avec des images qui comporte des zones de nature différente.
- permettra une transmission progressive pour une reconstitution à résolution croissante.
- sera compatible avec des canaux à faible largeur de bande et des périphériques disposant de peu de mémoire.

Le dernier chapitre sera réservé à la partie résultat où nous allons appliquer le standard JPEG sur plusieurs images avec différents facteurs de qualité.

En fin, et pour une meilleure explication de la norme JPEG, vous trouvez dans l'annexe A un exemple manuel qui montre les étapes détaillés de la compression.

Chapitre IV : Simulation et Résultats



IV.1. Introduction

IV.2. Présentation de l'interface graphique

IV.3. les critères de qualité

IV.4. L'algorithme

IV.5. Testes et Résultats de la compression

IV.6. Interprétation

IV.7. Conclusion

IV.1. Introduction

Ce chapitre présente la partie simulation et résultats des codes introduits aux chapitres précédents. Rappelons que notre travail consiste à développer un algorithme de compression d'images utilisant une transformée de cosinus discrète (DCT) et une quantification afin d'atteindre de bons taux de compression pour les images sans trop altérer la qualité.

Nous évaluerons ici les résultats obtenus pour différents paramètres de précision (facteur de qualité FQ) et nous verrons l'influence de ce paramètre sur la qualité de compression obtenue.

Lorsqu'on utilise des techniques de compression avec perte, l'utilisation de mesure de qualité est indispensable pour l'évaluation des performances. Le problème majeur dans l'évaluation des techniques de compressions avec pertes réside dans la difficulté à décrire la nature et l'importance des dégradations sur l'image reconstruite.

La notion de critère qualité en adéquation avec les applications en aval, est donc particulièrement intéressante pour évaluer l'impact d'un procédé de compression sur l'exploitation de l'image finale. Pour ce faire, nous allons définir certains critères tels que le PSNR, ainsi que le MSE ou le SNR, pour permettre l'évaluation de la qualité de compression de l'algorithme élaborée.

Afin de faciliter l'utilisation des algorithmes élaborés pour la compression des images et de manipuler les différents paramètres nous avons regroupé ces algorithmes dans un GUI (Graphita User Interface) sous MATLAB.

Cette interface graphique permettra de sélectionner une image couleur à l'aide d'un explorateur et d'appliquer la chaîne de compression qu'a été implémentée afin de voir l'amélioration apportée par l'algorithme.

Cette interface permettra de calculer le PSNR, EQM, Bpp, le gain et le taux de compression et d'afficher l'image compressée et l'image originale afin de faire une comparaison visuelle des résultats. Nous allons présenter dans ce qui suit cette interface plus en détail et décrire son utilisation.

IV.2. Présentation de l'interface graphique

Le GUI (Graphical User Interface) de MATLAB permet de créer des interfaces où le programmeur choisit plusieurs types d'objets (boutons, edit box, liston.....) appelés Handler. Ensuite, il réalise la programmation pour obtenir l'interaction qu'il souhaite entre ces différents objets.

L'interface réalisée permet la compression des images couleurs à l'aide d'une phase de compression et une autre pour la décompression.

Une vue de l'interface est donnée dans la figure suivante

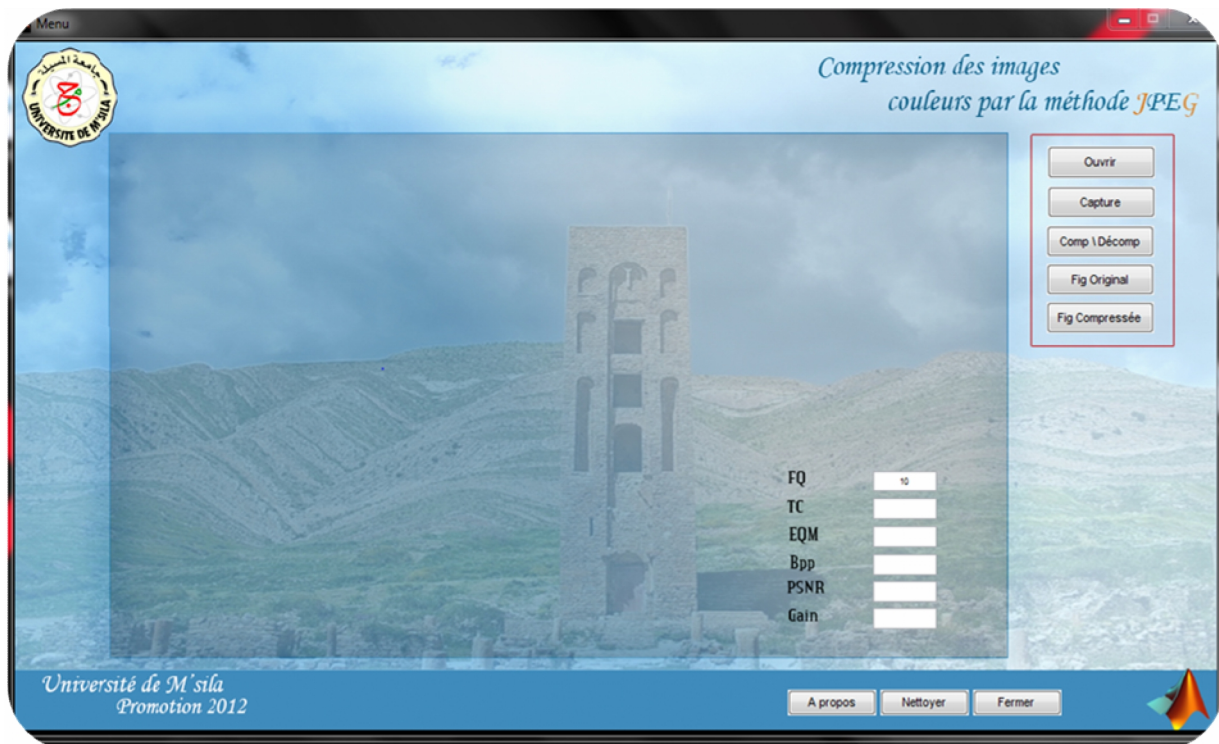


Figure IV.1 : Une capture de l'interface.

IV.2.1 Cette interface contient

➤ Quatre 'Pushbutton'

- 1- ouvrir: pour le chargement des images
- 2- capture : pour le chargement des images, en utilisant dans ce cas la webcam.
- 3- comp/décomp: permette d'exécuter la compression et la décompression
- 4- Fig Originale : nous permet d'afficher l'image avant compression
- 5- Fig compressée: nous permet d'afficher l'image après compression/décompression

➤ Un 'Edit boxes'

- pour définir facteur de qualité (FQ)

➤ Quatre 'Output'

Nous permet d'afficher

- 1- EQM : l'erreur quadratique moyenne
- 2- TC: taux de compression
- 3- SNR : le rapport signal sur bruit
- 4- PSNR : le rapport signal sur bruit crête

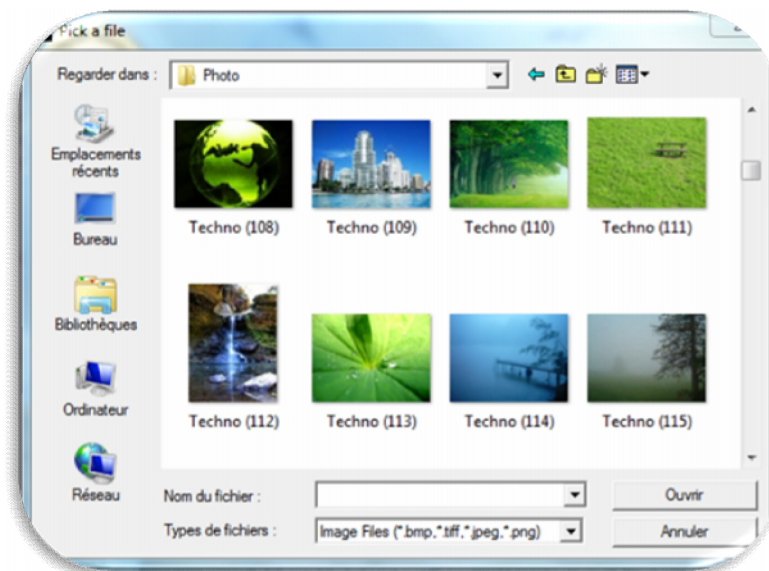
IV.2.2 Démarche à suivre pour la compression des images

1. Pour sélectionner une image :

- Soit vous cliquez sur le bouton « ouvrir ».....



Une fenêtre s'ouvre, permettant de parcourir le disque et de choisir n'importe quelle image.



Ou bien vous cliquez sur le bouton « capture » pour prendre une photo à l'aide de webcam



2. Ensuite à l'aide du clavier numérique nous saisissons le facteur de qualité (FQ) désiré.



3. Une fois que tous ces paramètres sont définis il ne reste plus qu'à compresser l'image par un clic sur le Botton 'Compresser/Décompresser'; la compression par DCT s'exécute.



L'ajoute de 'Fig Originale' et 'Fig compressée' dans notre interface graphique (GUI de MATLAB) nous permet de récupérer les deux images, l'une avant compression (originale) et l'autre après compression/décompression (décompressée) et de les visualiser dans une fenêtre sur notre écran.



Les cinq 'output' (PSNR, EQM, GAIN, Tc et Bpp) nous permettent d'afficher le PSNR, EQM, Bpp, Gain et le TC (taux de compression) ; de l'image compressée. La figure suivante nous donne un exemple des résultats obtenus avec notre algorithme.

| | |
|------|---------|
| FQ | 10 |
| TC | 4.4457 |
| EQM | 23.2734 |
| Bpp | 0.22591 |
| PSNR | 34.2637 |
| Gain | 77.4894 |

IV.3. les critères de qualité

Afin de permettre l'évaluation de la qualité de compression de l'algorithme élaboré, nous allons définir les critères de qualité suivants :

IV.3.1 Erreur Quadratique Moyenne (EQM) ou Mean Square Error (MSE)

Elle représente l'erreur quadratique moyenne entre l'image compressée et l'image originale. Cette mesure va permettre d'évaluer l'influence des dégradations de la compression sur l'image compressée. Elle est définie comme suit :

$$MSE = \frac{1}{n_x n_y n_\lambda} \sum_{x,y,\lambda} (I(x,y,\lambda) - \tilde{I}(x,y,\lambda))^2$$

IV.3.2. Rapport Signal sur Bruit (RSB) ou Signal to Noise Ratio (SNR)

C'est le rapport entre la puissance de l'erreur avec la puissance du signal original (PS).

$$SNR_{(db)} = 10 \log_{10} \frac{PS}{MSE}$$

IV.3.3. Le PSNR (Peak SNR)

C'est la mesure de la distorsion entre l'image originale et compressée. Il s'agit de quantifier la performance des codeurs en mesurant la qualité de reconstruction de l'image compressée par rapport

à l'image originale. Les valeurs typiques du PSNR pour des images de bonne qualité varient entre 30 et 40 dB.

$$PSNR_{(db)} = 10 \log_{10} \frac{PeakSignal^2}{MSE}$$

IV.3.4. Taux de compression (Tc)

Pour déterminer le degré de décompression obtenu, on définit également : le taux de compression (Tc) :

$$Tc = \frac{\text{Nombre de bits de l'image originale}}{\text{Nombre de bits de l'image comprimée}}$$

IV.4. L'algorithme

Dans cette partie, et après avoir quelques notions sur les méthodes de compression avec perte et sans perte, nous avons choisi la compression JPEG en utilisant la DCT qui nous a permis d'implémenter un programme de compression et de décompression des images fixes couleurs.

Nous allons donner une description sur les différentes étapes de conception de programme et puis évaluer les résultats visuellement (l'image originale / l'image décompressée / la DCT).

On résume les étapes de compression et de la décompression dans l'organigramme de la figure suivante :

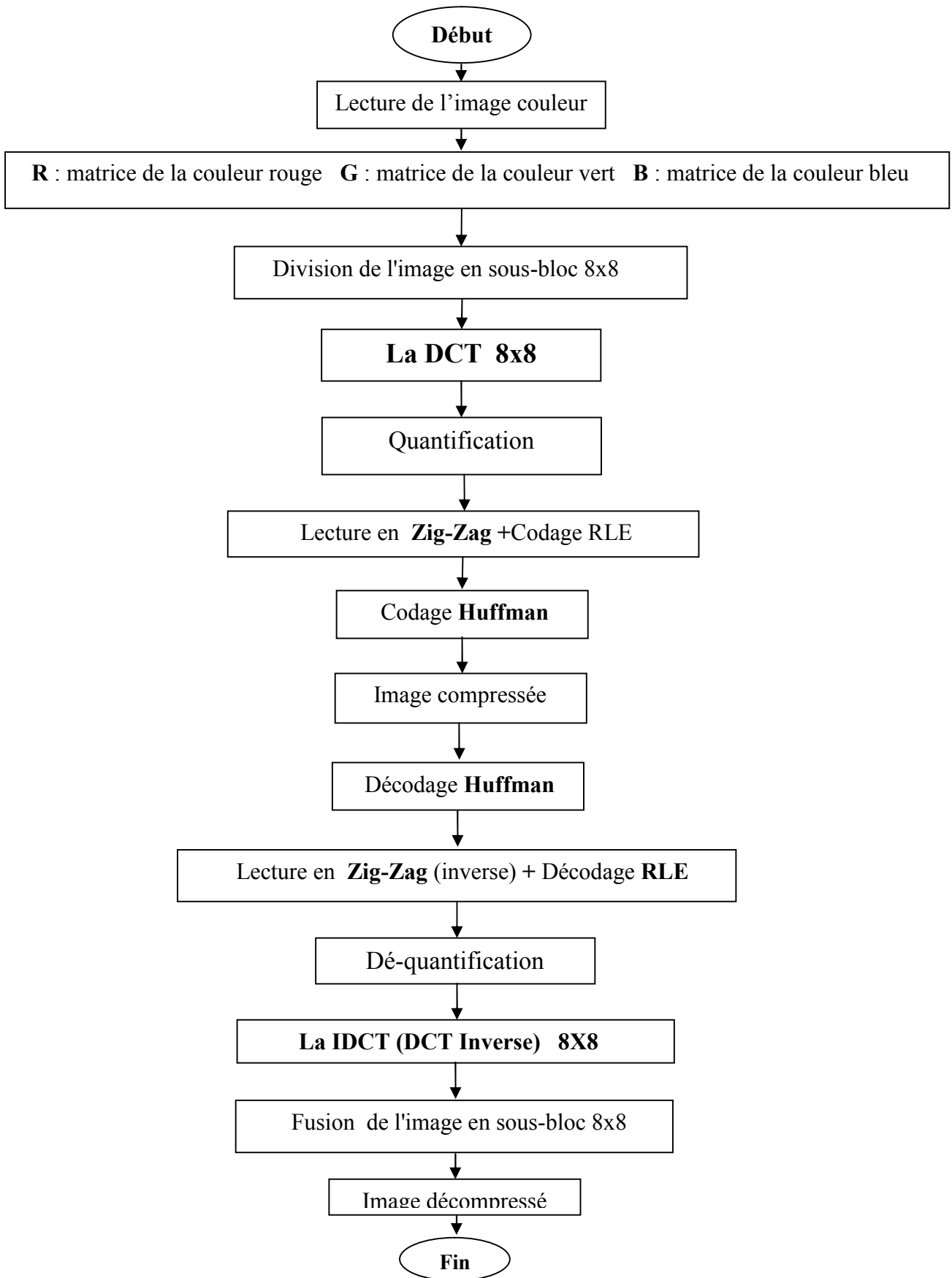


Figure IV.2 : Organigramme général de la compression JPEG .

IV.4.1. Phase de compression

Cette phase est divisée en plusieurs étapes :

IV.4.1.1. Acquisition de l'image originale

Par l'utilisation des instructions Matlab nécessaires pour lire une image, on stocke cette image sous une forme matricielle.

IV.4.1.2. La DCT (Discret Cosine Transform)

C'est une variante de la transformé de Fourier discrète, qui permet de garder seulement les cosinus et d'éliminer les Sinus, c'est-à-dire, obtenir une représentation fréquentiel purement réel.

Cette transformé est très largement utilisée dans la compression audio et la compression image comme JPEG et MPEG.

Dans notre travail, l'utilisation de la méthode DCT nous donne beaucoup des avantages

A/ Dé-corrélation :

La DCT permet de réduire ou d'éliminer la redondance inter pixel, les pixels sont Dé-corrélés afin de réduire la variance des pixels voisins dans l'image ce qui permet une compression efficace on codant chaque pixel indépendamment.

B/ Concentration des coefficients :

La DCT est très efficace pour des images fortement corrélée du fait qu'elle permet de compacté les coefficients qui représente les basses fréquences dans une seule partition de la matrice image, cela permet la séparation des fréquences basses des fréquences hautes, et si c'est une image faiblement corrélée, les coefficients sont concentrés dans plusieurs différentes partitions de la matrice image.

IV.4.1.3. La quantification :

Le quantificateur permet de faire une pondération des coefficients de la transformé en se basant sur une table de quantification qui a été établit sur les caractéristiques de la vision humaine, les coefficients de cette table imposeront un seuil de qualité qui décidera des coefficients de la transformé qu'on veut garder et ceux qu'on veut supprimer jugées non important visuellement, d'une autre manière, cette étape permet de choisir le taux de dégradation visuel qu'on souhaite faire à l'image (Facteur de qualité), et le but de tous cela c'est d'éliminer la redondance psycho-visuel. Cette étape se trouve seulement dans les compressions avec perte, et pour cela on a ajouté la fonction suivante qui va générer d'autres tables de quantification en fonction de FQ:

$$\text{Quantificateur}(i, j) = 1 + [(1 + i + j)*FQ]$$

IV.4.1.4. Le codage

***codage RLE** : élimination des répétitions.

***codage huffman** : codage les valeurs par de codes de tailles variables, ce qui permet de réduire le nombre de bit par pixel.

On résume la phase de la compression dans le schéma bloc de la figure suivante :

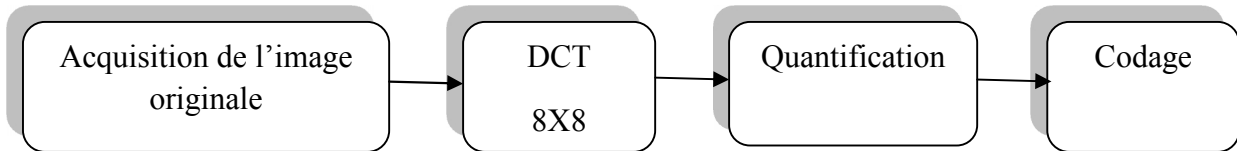


Figure IV.3: Schéma bloc de la compression.

IV.4.2. La phase de décompression

IV.4.2.1. Le décodage huffman

L'identification des éléments à l'aide de la table de huffman, chaque séquence sera codée sur 8bits (remplacée par son code équivalent).

IV.4.2.2. Le décodage RLE

La récupération des coefficients de DCT quantifiées.

IV.4.2.3. La dé-quantification

Extraire les coefficients de la DCT avec les moindres pertes possibles.

IV.4.2.4. La IDCT

La récupération des éléments de la matrice représentant l'image décomposée.

IV.4.2.5. La reconstruction de l'image originale

Cette étape consiste à reconstruire l'image originale.

On résume la phase de la compression dans le schéma bloc de la figure suivante :

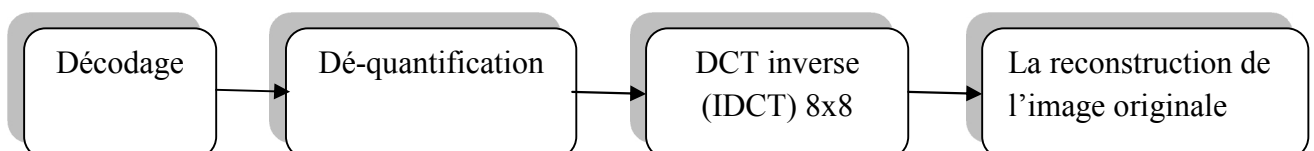


Figure IV.4: Schéma bloc de la décompression.

IV.5. Testes et Résultats de la compression

L' algorithme mentionnée a été mise en oeuvre sous Matlab R2008b et les performances ont été évaluées pour différents facteur de qualité : taux de compression(T_c), l'erreur quadratique moyenne (EQM),le PSNR,la taille de l'image décompressée et sa qualité .

Nous avons obtenu les résultats suivants pour différentes valeurs du facteur de qualité (FQ).

➤ Pour FQ=1



Image original 512x512



Image décompressée

| Taille de l'image original (bit) | Taille de l'image Compressée(bit) | Taux de compression | PSNR (db) | EQM |
|----------------------------------|-----------------------------------|---------------------|-----------|-----|
| 786432 | 675469 | 1.16 | 43.77 | 2.7 |

Tableau IV.1 : Résultats de la compression avec FQ=1

➤ Pour FQ=3

Image original 512x512



Image décompressée

| Taille de l'image original (bit) | Taille de l'image Compressée(bit) | Taux de compression | PSNR (db) | EQM |
|----------------------------------|-----------------------------------|---------------------|-----------|------|
| 786432 | 340588 | 2.30 | 39.26 | 7.70 |

Tableau IV.2 : Résultats de la compression avec FQ=3

➤ Pour FQ=10

Image original 512x512



Image décompressée

| Taille de l'image original (bit) | Taille de l'image Compressée(bit) | Taux de compression | PSNR (db) | EQM |
|----------------------------------|-----------------------------------|---------------------|-----------|-------|
| 786432 | 252159 | 3.11 | 34.46 | 23.27 |

Tableau IV.3 : Résultats de la compression avec FQ=10

➤ Pour FQ=20



Image original 512x512



Image décompressée

| Taille de l'image original (bit) | Taille de l'image Compressée(bit) | Taux de compression | PSNR (db) | EQM |
|----------------------------------|-----------------------------------|---------------------|-----------|-------|
| 786432 | 133916 | 5.87 | 31.69 | 43.99 |

Tableau IV.4 : Résultats de la compression avec FQ=20

➤ Pour FQ=50



Image original 512x512



Image décompressée

| Taille de l'image original (bit) | Taille de l'image Compressée(bit) | Taux de compression | PSNR (db) | EQM |
|----------------------------------|-----------------------------------|---------------------|-----------|--------|
| 786432 | 106364 | 7.39 | 27.78 | 108.20 |

Tableau IV.5 : Résultats de la compression avec FQ=50

➤ Pour FQ=80



Image original 512x512



Image décompressée

| Taille de l'image original (bit) | Taille de l'image Compressée(bit) | Taux de compression | PSNR (db) | EQM |
|----------------------------------|-----------------------------------|---------------------|-----------|--------|
| 786432 | 99672 | 7.89 | 25.99 | 163.35 |

Tableau IV.6 : Résultats de la compression avec FQ=80

On résume toute les résultats de compression pour les déférentes valeurs de FQ dans le tableau suivant :

| FQ | Taille de l'image Compressée(bit) | T_C | PSNR (db) | EQM | bpp | G_C (%) |
|-----------|--|----------------------|----------------------|------------|------------|------------------------------|
| 1 | 675469 | 1.16 | 43.77 | 2.72 | 0.858 | 14.10 |
| 3 | 340588 | 2.30 | 39.26 | 7.70 | 0.433 | 56.69 |
| 5 | 252159 | 3.11 | 37.16 | 12.48 | 0.320 | 67.93 |
| 10 | 177041 | 4.44 | 34.46 | 23.27 | 0.225 | 77.48 |
| 20 | 133916 | 5.87 | 31.69 | 43.99 | 0.170 | 82.97 |
| 50 | 106364 | 7.39 | 27.78 | 108.20 | 0.135 | 86.47 |
| 80 | 99672 | 7.89 | 25.99 | 163.35 | 0.126 | 87.32 |
| 99 | 97655 | 8.05 | 24.08 | 253.76 | 0.124 | 87.58 |

Tableau IV.7: Résultats de la compression avec FQ= (1 jusqu'à 99)

Avec :

Bpp : C'est le débit qui permet d'estimer le nombre de bits moyens utilisés pour coder un élément. il est donné par la relation suivante :

$$Bpp=(T_C)^{-1}$$

G_C : C'est le gain de compression, il est donné par la relation suivante :

$$G_c = (1-Bpp)*100$$

IV.5. Interprétation

- D'après les résultats visuels de la compression, on remarque que dans le cas de (FQ=10) l'image a été compressée avec taux de compression **T_C = 4.44** ce qui est acceptable.

L'image n'a pas perdu de sa fidélité apparente, cela étant seulement visuellement apparent, parce que l'erreur quadratique moyenne n'est pas nulle et égale à **EQM = 23.27** cela veut dire qu'il y'a une erreur d'une différence moyenne environs **23.3** entre les éléments des deux matrices (matrice originale /matrice décompressée).

Le rapport crête signal sur bruit **PSNR=34.46 dB** est acceptable, ce qui montre qu'on a fait une compression efficace en matière de taille/qualité.

Le nombre de bit par pixel (bpp) a été réduit à **0.225** cela nous a fait perdre du poids au fichier image.

- D'après le tableau (IV.5), on remarque que plus on choisit un facteur de qualité grand, plus le taux de compression est meilleur du fait qu'on augmente le seuil de la table Q mais EQM augmente aussi ce qui va influencer sur l'efficacité de la compression constatée visuellement et aussi au vue du rapport PSNR médiocre du fait qu'on a perdus beaucoup de données pendant l'étape de quantification et cela est devenue très visible, ce qui rend l'image exploitable à certaines limites.

IV.6. Conclusion

En conclusion, nous avons montré comment la compression JPEG est atteinte et qu'elle est efficace dans son objectif de réduction la taille des fichiers tout en conservant la qualité. Nous avons démontré que l'algorithme de compression utilise une transformée de cosinus discrète pour modifier la base de la matrice image à partir de la base spatiale à la base fréquentiel, et au même temps de séparer les basses fréquences, qui contiennent le plus d'informations sur l'image dans le coin supérieur gauche de chaque bloc 8 X 8. Puis, la quantification a été réalisée et, enfin, par l'utilisation du codage efficace de (Huffman) la taille du fichier de l'image est devenue beaucoup plus petite.

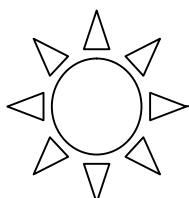
Le principal avantage de ce format est **le taux de compression réglable** qui permet à l'utilisateur de trouver un compromis entre le taux de compression et la qualité de l'image.

Conclusion générale

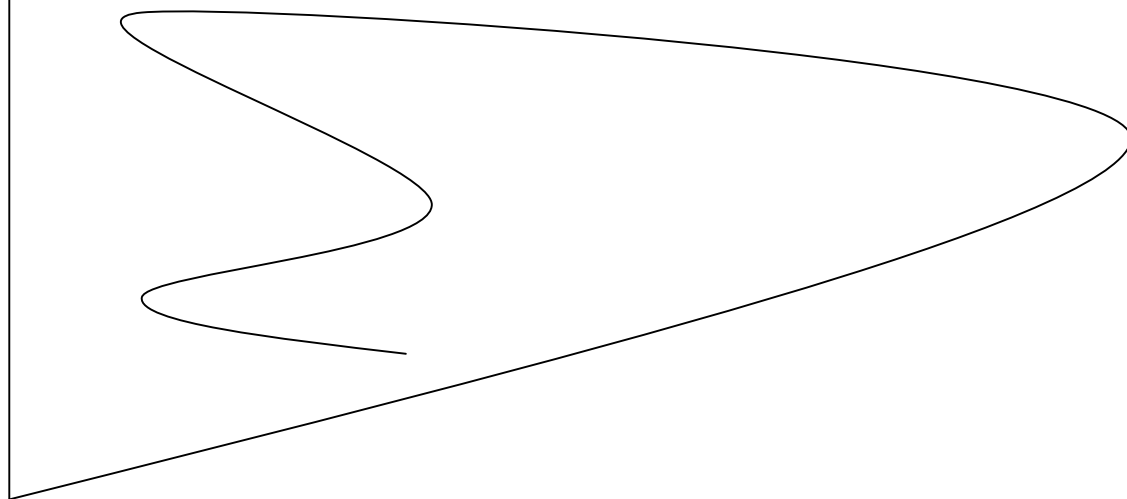
Comme de nombreuses méthodes de compression, le Jpeg est basé sur des principes mathématiques très compréhensibles. Mais la difficulté intervient lorsque l'on entre en détail dans les démonstrations de l'algorithme. Théorèmes et principes fondamentaux doivent alors être démontrés, pour expliquer les fondements de la méthode.

L'utilisation de telles méthodes de compression des informations est très répandue et utilisées dans de nombreux domaines : informatique, téléphonie, hi-fi, vidéo,...

A l'heure actuelle la méthode de compression JPEG est parmi les plus utilisées parce qu'elle atteint des taux de compression très élevés sans que les modifications de l'image ne puissent être décelées par l'œil humain. De plus, beaucoup d'implémentations permettent de choisir la qualité de l'image comprimée grâce à l'utilisation de matrices de quantification paramétrables. La réputation et donc le nombre d'utilisateurs d'un algorithme de compression dépendent de différents facteurs : le rapport taille/qualité, la vitesse de compression et de décompression. Il est donc intéressant de comparer les différentes méthodes, pour pouvoir choisir la plus adaptées à nos besoins et à son utilisation.



Annexe



Annexe A

Codage par plage

| | | | | | | | | | |
|---|-----|---|----|---|---|---|---|----|----|
| 0 | 386 | 0 | -1 | 0 | 9 | 0 | 3 | 17 | -1 |
|---|-----|---|----|---|---|---|---|----|----|

Le codage d'entropie

| Le nombre | La probabilité d'apparition | Le codage |
|------------|-----------------------------|------------|
| 0 | 4 | 00 |
| -1 | 2 | 010 |
| 3 | 1 | 100 |
| 9 | 1 | 101 |
| 17 | 1 | 110 |
| 386 | 1 | 111 |

La série codée:

| | | | | | | | | | |
|----|-----|----|-----|----|-----|----|-----|-----|-----|
| 0 | 386 | 0 | -1 | 0 | 9 | 0 | 3 | 17 | -1 |
| 00 | 111 | 00 | 010 | 00 | 101 | 00 | 100 | 110 | 010 |

Bibliographie

- [1] KADDOUR Chakib, « Généralités sur le traitement d'images », 1999
- [2] André MARION, « Introduction aux techniques de traitement d'images »,
Edition : EYROLLES, Paris, 1987.
- [3] R. HAROUN- S. A. BOUHRICHE, « Séparation des chromosomes se touchant ou se chevauchant à partir de Meta phases », diplôme d'ingénieur d'état en Electronique,
ENP- Promotion 2001-2002
- [4] <http://www.commentcamarche.net/video>
- [5] Gérard MICHEL-DUTHEL, « Images numériques & Formats graphiques», Edition : Campus presse, Paris, 2000
- [6] C. BOUDRY, « Qu'est-ce qu'une image numérique »,
URL: <http://www.ecr.jussien.fr/urfist/image-numérique.htm>
- [7] Gaël JABOULAY, « Format de fichiers d'images »,
URL : http://tecfa.unige.ch/tecfa/teachg/staf13/fiches-mm/format_fichier.htm
URL: <http://iquebec.ifrance.com/kadchakib/chap1/chap1.htm>
- [8] Sylvain Renard, «Première approche de JPEG 2000 », 2003
URL: <http://ruses.com/pdf/jp2000.pdf>
- [9] Brault et Dougherty, «les formats de compression d'image», 2004
Institut Universitaire de Technologie de Tours
- [10] Jean FRUITET, « Outils et méthodes pour le traitement des images par ordinateur », URL: http://etudiant.univ-mev.fr/~fruitet/Trait_image/an_image.html
URFIST de Paris, 2100-2002
- [11] Saint-Moulin, «compression numérique & taux de transferts " les images " »,2002.

- [12] S.Benabdelkader, «compression des images satellites monochromatiques via la transformation en ondelettes », Thèse de magister, Université de Constantine ,1996
- [13] R.J.Clark , «Transform coding of images», Academic press INC, London 1985.
- [14] Hemers, «La compression des images numériques », Paris,1995.
- [15] A.Boukhelif , «Techniques de compression d'image »Edition , 2004.
- [16] I.Ben belouar .M.Ousfane, «La compression d'image par quadrettes »
- [17] M.Grégoire,Christian.R et Gilbert .M , «Technologies du multimédia », ENST Bretagne,2003.
- [18] J.Fages «JEPEG2000-Principes, implémentation et évaluation », Mémoire d'ingénieur ,Université de Houari Boumediene, 2002.
- [19] I.Block , «Traitement des image /Tome 01 », Département TSI_ Télécom, Paris ,2004.
- [20] J. Demaeyer, M. Bebronne et S. Forthomme , «Les Ondelettes », Université Libre de Bruxelles, Belgique.
- [21].<http://Généralités sur la compression .htm>.