



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

# MÉMOIRE DE MASTER

Sciences et Technologies  
Electronique  
Réseaux et télécommunications

Réf. : .....

---

Présenté et soutenu par :  
**DEGHNOUCHE Hamoudi**

Le : lundi 8 juillet 2019

## ETUDE DE LA NORME DE COMPRESSION D'IMAGES JPEG

---

### Jury :

M.	BAARIR Zineddine	Pr	Université de Biskra	Président
Mme.	BERKAT Aicha	Dr	Université de Biskra	Examineur
M.	OUAFI Abdelkrim	Pr	Université de Biskra	Rapporteur

Année universitaire : 2018 - 2019

# Bibliographie

- [1] A. ARNEDO, F. ARGOUL, E. BARCY, J. ELEZGARAY and J.-F. MUZY, *Ondelettes, Multifractales et Turbulences*, Diderot Eds - Arts et sciences, 1995.
- [2] A.Rosenfeld . Quadrees and pyramids for pattern recognition and image Processing .
- [3] Bertin E., Parazza F., J-M. Chassery, "Segmentation and Measurement based on 3D Voronoi Diagram: Application to Confocal Microscopy", *Computerized Medical Imaging and Graphics*, Vol. 17, pp. 000-008, 1993.
- [4] Bellaiche, Philippe. Les secrets de l'images vidéo. Edition 2010 ; Groupe Erolles p359.
- [5] Barlaud Michel, LABIT Claude.Floche à mayenne en janvier 2002 P.114 -120
- [6] Benjamin Battin, « compression multi-vues-des flux-auto-stéréoscopiques » thèse doctorat université de Reims Champagne-Ardenne (URCA) , 2010.
- [7] Boucetta.Aldjia, »Etude de l'effet des transformées de décorrélation en compression des images couleurs RGV » thèse université de BATNA,2010.
- [8] Boucetta .A and Melkmi ,K.E « DWT based approach for color compressing généc algorithm », *image and signal processing springer Berlin Heidelberg*, 2012, P.P.476 – 484.
- [9] C. A. Cabrelli, B. Forte, U. M. Molter, and E. R. Vrscay. Iterated fuzzy set Systems : A new approach to the inverse problem for fractals and other sets *Journal of Mathematical Analysis and Applications* ,171(1) :79-100; November 1992.
- [10] Cathel Zitzmann , dtection statistique d'information cachée dans des images naturelles. »thèse doctorat université de technologie de troyes, 2013.
- [11] C.loeffler.A.Lieenberg et G.S.Moschytz, « *pratical irsts 1-d DCT algorithms with 11 multiplications,EEE.pp.988-911,1980.*

- [12] C.Charrier. « Evaluation de la qualité d'image » ;LUSAC-SAINT-LO-EA 2607
- [13] Davoine, 1995; Mercier et al., 2003; Brielmann, 2006].
- [14] Douak,F,Benzid,R and Benoudjit,N « Color image compressing algorithm based on the DCT transform combined to an adaptive block scanning »,AEU-international journal of electronics and communication 2011,65 pp16-26 .
- [15] E.Lutton and J. Lévy\_Véhel. Optimization of fractal functions using genetic Algorithms. In Fractal'93. Londres,1993.
- [16] G. E. oien and S. Lepsoy. Fractal-based image coding with fast decoder convergence, Signal Processing ,40:105-117, October 1994.
- [17] G. Mantica and A.Sloan. Chaotic optimization and the construction of fractals: solution of an inverse problem. Complex Systems,3:37-62,1989.
- [18] H.Samet. Region representation: quadrees from binary arrays. CVGIP, 13:88-93 ,1980  
In Proc. 5<sup>th</sup>. Int. Conf . on Pattern Recognition . pages 802 – 811, University of Maryland.1980 .
- [19] H.Samet. Thequadtree and Related hierarchical data structures. ACM Comput.Surv.16:178-260,1984.
- [20] Klinger. Data structures and pattern recognition. In Proc. Int. Joint Conf.on Pattern Recognition, pages 497-498, 1973
- [21] Messaoudi Abdelhamid, « Contribution à l'amélioration et la mise en œuvre de nouveaux algorithmes pour la compression des signaux » Thèse doctorat, Université Mustapha Ben Boulaid Batna 2 , Département d'Electronique : 2016/2017

- [22] M. BARNESLEY, Fractals Everywhere, Academic Press, Inc., 1988.
- [23] M. F. Barnesley, V. Ervin , D.Hardin, and J. Lancaster. Solution of an inverse problem for fractals and other sets. Proc. Natl. Acad. Sci. USA,83:1975-1977, April 1986.
- [24] P.Beaurepaire. « Compression d'images appliquée aux angiographies cardiaque : aspect algorithme, évaluation de la qualité diagnostique » thèse doctorat université de technologie de compiene, 1997.
- [25] R.C.Gonzalez,R.Eoods. « Digital image processing » ;2<sup>end</sup> Edition Prentice Hall. Ne Jerzey 2002.
- [26] R . Vrscay . Moment and collage methods for the inverse problem of fractal construction with iterated function systems . In Fractal 90 conference, June 1990.
- [27] R. Rinaldo and A. Zakhor. Inverse and approximation problem for two dimensional fractal sets. IEEE Transactions on Image Processing,3(6):802-820, November 1994.
- [28] T.I.P.E. Terre et Espace : Télécommunication : Compression Jpeg  
Avril 1999
- [29] Y. Fisher. Fractal Image Compression with Quadrees, 1995.
- [30] Zeroual Djazia. « Implémentation d'un environnement parallèle pour la compression d'images a l'aide des fractales, mémoire magistère en informatique, université de BATNA,2006
- [31] Y. Mercier, C. Roux, et G. Martineau, "Technologies du Multimédia", ENST Bretagne, dpt ITI, BP 832, F-29280 Brest, France.15 janvier 2003.
- [32] **site internet : <https://avecunphotographe.fr/format-jpeg-avantages-defaults>**







## CONCLUSION GENERALE

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 matrice de quantification paramétrable.

L'avenir de la compression d'images semble maintenant appartenir au nouveau standard du Joint Picture Expert Groupe, JPEG.

En fin, on peut conclure qu'il existe de nombreuses manières de coder, de représenter et de compresser les images. Il existe une multitude de formats pour compenser la diversité des images, différentes par leurs dimensions, leur nombre de couleurs et leurs méthodes de représentation (vectorielle et bitmap). Chaque format est complémentaire des autres. Aujourd'hui, avec la considérable avancée technologique, les algorithmes se permettent de réaliser de nombreux calculs pour compresser et décompresser une image. Les ratios deviennent de plus en plus grands, sans pertes significatives et réelles de la qualité de l'image. Les dernières méthodes de compression (fractale et ondelette) sont particulièrement prometteuses. Elles montrent la nécessité d'associer à la programmation informatique, les mathématiques. L'avenir de la compression ne pourra s'opérer qu'au travers d'algorithmes mathématiques, et les progrès de la recherche mathématique entraîneront sûrement une avancée dans la compression d'images et des fichiers. Afin d'arriver à une meilleure qualité des images des méthodes et des mesures sont introduites pour favoriser des méthodes de compression par rapport aux autres selon le domaine d'utilisation de ces dernières.



## INTRODUCTION GENERALE

La dernière décennie, le monde est devenu très développer grace a la numérisation des données, signal, image et video.

Dans notre vie l'image est considéré comme le support d'information le plus performant tout au niveau de son accessibilité de son universalité, que dans ces différentes applications telle que, l'imagerie médicale comme le scanner et l'IRM, l'imagerie satellitaire les images acquises par les nouveaux satellites sont actuellement en nette augmentation, grâce à la haute technologie des satellites les images sont produites en un délai très court avec une précision indiscutable, la video-conférence, la télévision haute définition, la télécopie, le mobile et les services d'information sur internet, engendrent une circulation de l'information de plus en plus dense.

L'utilisation intensive de l'image a rendu nécessaire l'extension des techniques numériques à des nombreux domaines d'analyse et de traitement de la saisie, du stockage, du transport et de la transmission.

En effet, grâce à l'accroissement massif de la quantité de données chargées, les supports de stockage et les réseaux de transmission arrive à leur saturation. dans ce contexte pour économiser les ressources du stockage ou des canaux de communication, la compression apparait comme l'outil essentiel à la continuité du progrès.

A l'aide de l'opération de compression le stockage et la transmission des données deviennent plus efficace et plus rapide, ce qui nous a permis de résoudre le problème de cout qui peut limiter la faisabilité de stockage et de transmission de tels système avec un volume d'information à transmettre ou à stocker.

Dans notre travail, nous somme intéressé à la compression des images numériques en particulier dans le cadre des normes établies telle que JPEG.

JPEG est l'acronyme de « **J**oin **P**hotographic **E**xpert **G**roupe ». il s'agit d'un comité d'experts qui edite des normes de compression pour l'image fixe .

## INTRODUCTION GENERALE

---

JPEG est le resultat de l'evolution des travaux qui ont débuté dans les années 1978 à 1980 avec les premiers essais en laboratoire de compression d'image .

Le groupe JPEG qui a reuni une trentaine d'experts internationaux a spécifié la norme en 1991. La norme officielle et définitive a été adoptée en 1992.

# LISTE DES ABREVIATIONS

<b>bpp</b>	Bit par pixel
<b>CR</b>	Compression Ratio
<b>DCT</b>	Discrète Cosine Transform
<b>DPCM</b>	Differential Pulse Code Modulation
<b>EQM</b>	Erreur Quadratique Moyenne
<b>FFT</b>	Faste Fourier Transform
<b>JPEG</b>	Joint Photographique Experts Group
<b>JPEG-LS</b>	Joint Photographique Experts Group Losless
<b>LZW</b>	Lempel-Ziv-Welch
<b>MICD</b>	Modulation par Impulsion Codées Différentielles
<b>MPEG</b>	Moving Picture Experts Group
<b>MED</b>	Mediane Edge Detector
<b>MSE</b>	Mean Square Error
<b>PSNR</b>	Peak Signal To Noise Ratio
<b>RLC</b>	Run Length Coding
<b>RLE</b>	Run Length Encoding
<b>SNR</b>	Signal To Noise Ratio
<b>TFD</b>	Transformation de Fourier Discrète
<b>TH</b>	Transformation de Hadamard
<b>THA</b>	Transformation de Haar
<b>TKL</b>	Transformation de Karhunen-Loeve
<b>WT</b>	Wave Transformation

---

# LISTE DES FIGURES

Figure I.1 Schéma fonctionnel de compression/décompression.....	3
Figure I.2 Un exemple de codage par plage RLE.....	6
Figure I.3 Algorithme de <i>Huffman</i> .....	7
Figure I.4 Schéma de compression JPEG sans perte .....	11
Figure I.5 : Principe de la quantification.....	15
Figure I.6 Quantification scalaire uniforme en escalier.....	16
Figure I.7 Le principe du Quantificateur Vectoriel.....	18
Figure II.1 schéma de bloc du codeur JPEG en mode sans perte.....	21
Figure II.2 Le voisinage de la prédiction du prédicteur MED .....	22
Figure II.3 Chaîne de codage de l'algorithme JPEG .....	23
Figure II.4 représentation de la DCT.....	25
Figure II.5 Vecteur de la base DCT .....	26
Figure II.6 Balayage zig-zag .....	31
Figure II.7 l'arbre de codage de Huffman .....	34
Figure II.8 Arbre de Huffman.....	34
Figure II.9 Schéma de principe de la compression par transformation.....	35
Figure III.2.1 compression JPEG (barbara_gray).....	43
Figure III.2.2 comparaison des sous bloc 8x8 de l'image originale et l'image compressée (barbara_gray).....	44
Figure III.2.3 compression JPEG avec un FQ = 10 .....	45
Figure III.2.4 compression JPEG avec un FQ = 30.....	45
Figure III.2.5 compression JPEG avec un FQ = 50 .....	45
Figure III.2.6 comparaison des critères objectifs en fonction FQ .....	46

Figure III.2.7 compression JPEG (pout).....	48
Figure III.2.8 comparaisons des sous bloc 8× 8(pout).....	48
Figure III.9 compression JPEG (Hilbert).....	49
Figure III.10 comparaisons des sous bloc 8× 8 (Hilbert).....	49
Figure III.11 compression JPEG (cameraman).....	50
Figure III.12 comparaisons des sous bloc 8× 8 (cameraman).....	50
Figure III.13 compression JPEG.....	51
Figure III.14 comparaisons des sous bloc 8× 8.....	51
Figure III.15 compression JPEG(eight).....	52
Figure III.16 comparaisons des sous bloc 8× 8 (eight).....	52
Figure III.17 pourcentage d'erreur en fonction de taux de compression.....	53

## LISTE DES TABLEAUX

Tableau II.1 Le tableau des probabilités d'occurrence.....	33
Tableau II.2 exemple d'échelle de notation .....	37
Tableau III.2.1 Les critères objectifs en fonction FQ (image barbara.....)	46
Tableau III.2 PSNR en fonction de taux de compression .....	53

Symbole	Nombre de fois	Probabilités
E	5	5/24







## INTRODUCTION GENERALE

La dernière décennie, le monde est devenu très développé grâce à la numérisation des données signal image et vidéo.

Dans notre vie l'image est considéré comme le support d'information le plus performant tout au niveau de son accessibilité de son universalité, que dans ces différentes applications telle que, l'imagerie médicale comme le scanner et l'IRM, l'imagerie satellitaire les images acquises par les nouveaux satellites sont actuellement en nette augmentation, grâce à la haute technologie des satellites les images sont produites en un délai très court avec une précision indiscutable, la video-conférence, la télévision haute définition, la télécopie, le mobile et les services d'information sur internet, engendrent une circulation de l'information de plus en plus dense.

L'utilisation intensive de l'image a rendu nécessaire l'extension des techniques numériques à des nombreux domaines d'analyse et de traitement de la saisie, du stockage, du transport et de la transmission.

En effet, grâce à l'accroissement massif de la quantité de données chargées, les supports de stockage et les réseaux de transmission arrive à leur saturation. dans ce contexte pour économiser les ressources du stockage ou des canaux de communication, la compression apparait comme l'outil essentiel à la continuité du progrès.

A l'aide de l'opération de compression le stockage et la transmission des données devient plus efficace et plus rapide, ce qui nous a permis de résoudre le problème de coût qui peut limiter la faisabilité de stockage et de transmission de tels système avec un volume d'information à transmettre ou à stocker.

Dans notre travail, nous sommes intéressés à la compression des images numériques en particulier dans le cadre des normes établies telle que JPEG.

JPEG est l'acronyme de « **J**oin **P**hotographic **E**xpert **G**roupe ». il s'agit d'un comité d'experts qui edite des normes de compression pour l'image fixe .

JPEG est le resultat de l'evolution des travaux qui ont débuté dans les années 1978 à 1980 avec les premiers essais en laboratoire de compression d'image .

Le groupe JPEG qui a reuni une trentaine dexperts internationaux a spécifié la norme en 1991. La norme officielle et définitive a été adoptée en 1992.

Ce mémoire est organisé en trois chapitres :

- Le premier chapitre donne un aperçu général sur les généralités de compression d'image fixe.
- Le deuxième chapitre donne une représentation spécifique sur les étapes principales de la méthode de compression JPEG avec leur critères de validation de qualité d'image.
- Le troisième chapitre est consacré sur les résultats de simulation obtenues par le programme Matlab sur des différentes images.

En fin , la conclusion générale présente le bilan de travail réalisé avec une vision sur les perspectives qui peuvent développer cette étude .

## CHAPITRE I

### I.1 INTRODUCTION :

La compression des images est appelée à prendre un rôle encore plus important en raison du développement des réseaux et du multimédia.

L'objectif principal de la compression d'images est de réduire la quantité d'informations nécessaire à sa représentation. Cette réduction est obtenue en minimisant la redondance de l'information présente dans une image. Cette redondance peut être statistique, spatiale ou encore fréquentielle. [13]

### I.2 DEFINITION DE LA COMPRESSION D'IMAGE:

L'art de la compression numérique consiste à supprimer certaines informations de l'image et à simplifier d'autre, tout en faisant en sorte que les modifications apportées échappent le plus possible à la perception humaine.

Un objectif d'autant plus difficile à atteindre des données compressées de taille inférieure à la taille original.

On peut considérer un algorithme de compression / décompression comme une boite à double entrée qui peut être de forme logiciel ou matérielle. Voir figure I.1



Figure I. 1 Ischema fonctionnel de compression/décompression.

## I.3 LES DIFFERENTS TYPES DE COMPRESSION

### I.3.1 Compression physique et logique

#### I.3.1.1 La compression physique

C'est un traitement qui agit directement sur les données, ce traitement est capable de comprimer l'information en agissant sur les données redondantes, afin qu'elles occupent un minimum de place. La lecture du fichier compressé est incompréhensible pour l'utilisateur.

#### I.3.1.2 La compression logique

C'est un traitement différent, c'est plutôt une réorganisation plus compacte des données ; l'information est mieux structurée et les données originales sont substituées par une autre information équivalente. Ce processus de substitution logique consiste à remplacer des symboles par d'autres plus petits, mais qui logiquement ont une même signification. *Exemple* : "type de données abstrait" «TDA» [3].

### I.3.2 Compression avec perte et sans perte

On peut distinguer deux grandes familles d'algorithmes de compression : les méthodes dites *sans perte* ou *réversibles* garantissent la restitution parfaite des images, alors que les méthodes dites *avec perte* ou *irréversibles* modifient plus ou moins la valeur des pixels.

#### I.3.2.1 Les méthodes réversibles ou sans pertes

Comme son nom l'indique, ce type de compression n'occasionne aucune perte de données. Cette compression conservatrice est utilisée dans des applications comme l'archivage des images médicales, l'imagerie satellitaire (le coût des images est élevé et les détails sont importants), les textes, les programmes et tout autre type de donnée

nécessitant une conservation à l'identique des données. Le seul critère d'évaluation des performances est dans ce cas "*Le taux de compression*". Il existe de nombreux types de compression d'image sans perte de données. Voici les plus répandus :

### I.3.2.1.1 Méthodes différentielles et prédictives

La méthode prédictive est l'une des plus anciennes, c'est une méthode décorrélatrice dont le principe est le suivant :

L'idée du codage prédictif (Modulation par Impulsions Codées Différentielles) est de supprimer la redondance entre pixels voisins et de ne coder que la différence entre la valeur du pixel courant et sa valeur prédite à partir des pixels voisins. Ce qui permet l'élimination de la redondance et en ne codant que la nouvelle information apportée par chaque pixel. Dans des systèmes plus complexes et performants, on établit une fonction de prédiction, qui permet d'estimer la valeur d'un pixel en fonction de la valeur des pixels voisins. On code alors l'erreur de prédiction, qui est l'écart entre la vraie valeur du pixel et la valeur prédite [29].

### I.3.2.1.2 Méthodes par plages

- **Le codage par répétition ou "Run Length Coding" (RLC)**

Plusieurs types d'algorithmes sont utilisés pour compresser l'information vont de plus simple au plus complexe, et leur efficacité varie suivant la nature des données à compresser. Le plus simple de ces algorithmes est le *Run Length Encoding (RLE)*, méthode aussi appelée *Run Length Coding (RLC)*.

C'est une technique avec mémoire ("*avec mémoire*" signifie qu'elle code les valeurs d'entrée en prenant en compte les valeurs précédentes). Son principe est de regrouper les valeurs voisines identiques et ne transmettre cette valeur qu'une seule fois, précédée par *le nombre de répétition*. Il est clair que cette approche fonctionne bien s'il y a beaucoup de répétitions dans le signal. Cet algorithme très simple, il peut aboutir à des taux de compression plus élevés. Il existe des variantes dans lesquelles l'image est encodée par pavés de points, selon des lignes, ou bien même en zigzag.

Un exemple de traitement *RLE* est donné à la Figure I.2 [26].

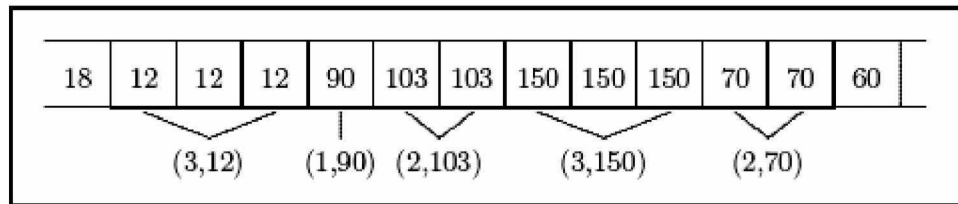


Figure I.2 Un exemple de codage par plage RLE.

### I.3.2.1.3 Codage de *SHANNON-FANO*

*C. Shannon* du laboratoire *Bells* et *R.M. Fano* du *MIT* ont développés à peu près en même temps une méthode de codage basée sur de simples connaissances de la probabilité d'occurrence de chaque symbole dans le message.

La procédure se décrit ainsi :

- **Procédure de codage**

- **etape1** : Classer les  $n$  fréquences non nulles  $\{f_i\}$  par ordre décroissant.
- **etape2** : Deviser l'ensemble des messages en deux sous-ensembles de fréquences aussi proches que possible.
- **etape3** : Attribuer à chaque sous ensemble un bit 0 ou 1 ;
- **etape4** : Rediviser chaque sous ensemble en deux nouveaux sous-ensembles de fréquences équivalentes afin de réitérer l'algorithme jusqu'à ce qu'il n'y ait plus qu'un seul élément dans chaque sous ensemble.

### I.3.2.1.4 Le codage de *HUFFMAN*

En 1952, *David Huffman* inventa une nouvelle méthode de compression appelée compression à Arbre de *Huffman* [22]. Le codage de *Huffman* crée des codes à longueurs variables sur un nombre entier de bits [16]. L'algorithme considère chaque message à coder comme étant une feuille d'un arbre qu'il reste à construire. L'idée est d'attribuer aux deux messages de plus faible probabilité, les mots codes les plus longs.

Ces deux mots codes ne se différencient que par leur dernier bit. Contrairement au codage de *Shannon-Fano* qui part de la racine d'un arbre et évolue par divisions successives, le codage de *Huffman* part des feuilles de l'arbre, et par fusions successives, redescend vers la racine [27].

- **Procédure de codage**

- Les probabilités d'occurrence de chaque message sont placées dans une liste dans un ordre décroissant. Nous dirons que la liste est composée d'enfants.
- Les deux probabilités les plus faibles sont identifiées en fin de liste.
- La somme des deux probabilités est placée à sa place dans la liste triée. Elle constitue un nœud parent. Les deux enfants sont retirés de la liste.
- Le chemin «enfant de plus faible probabilité, parent» est codé par un 1, l'autre par un 0
- La procédure reprend à l'étape 2 jusqu'à ce qu'il ne reste plus qu'une probabilité dans la liste.[30]

Malgré son ancienneté, cette méthode est toujours remis au goût du jour, et offre des performances appréciables. En effet, beaucoup de recherches en algorithmiques ont permis d'améliorer les fonctionnalités de la méthode *Huffman* de base, comme avec les arbres binaires, arbres équilibrés, ...etc. [20]

**Exemple**

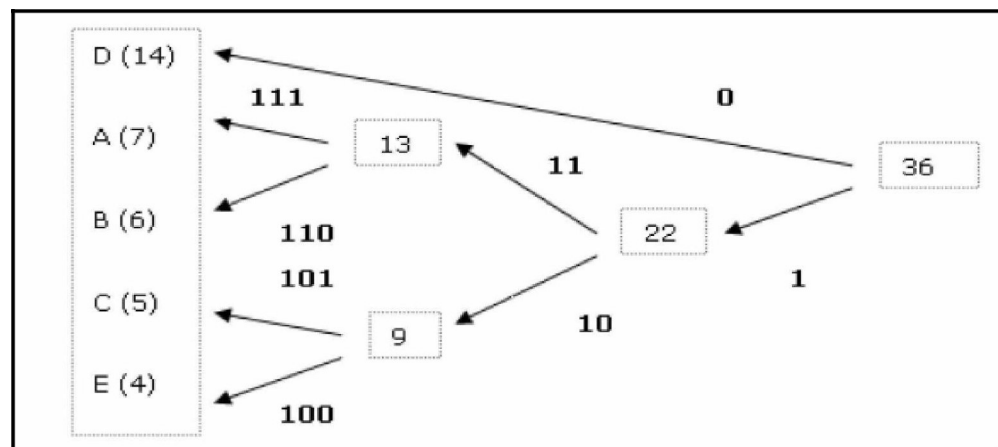


Figure I.3 Algorithme de *Huffman* [30]

Le codage obtenu est donc :

$D$  (occurrence =14) : 0 ;     $A$  (occurrence =7) : 111 ;     $B$  (occurrence =6) : 110 ;  
 $C$  (occurrence =5) : 101 ;     $E$  (occurrence =4) : 100

### I.3.2.1.5 Le codage arithmétique

Le codage arithmétique est un codage récent utilisant un modèle statistique, tout comme le codeur de *Huffman*. Contrairement à ce dernier, il produit un code pour une séquence de symboles tout entière, et non pas un code par symbole. Chaque nouveau symbole lu modifie de façon incrémentale le code de sortie. Ce code de sortie est un nombre à virgule flottante compris entre 0 et 1, dont le nombre de chiffres après la virgule correspond au nombre de symboles. Contrairement à *Huffman*, il n'est pas obligatoire que chaque code ait un nombre entier de bits. Par exemple un symbole de probabilité 0.9 a pour entropie 0.15, mais *Huffman* affectera probablement un code de un bit (ou plus), et la séquence codée aura un nombre de bits plus long qu'en théorie. Le codeur arithmétique est plus performant que le codeur de *Huffman*, mais il est plus complexe à implémenter. [17]

- **Algorithme du codage arithmétique**

Nous décrivons brièvement ci-dessous l'algorithme de codage arithmétique dans le but d'en illustrer le principe, sachant que le décodage opère de manière inverse.

- **Etape 1** : Calculer la probabilité associée à chaque symbole dans la chaîne à coder.
- **Etape 2** : Associer à chaque symbole un sous intervalle proportionnel à sa probabilité, dans l'intervalle  $[0,1[$  (l'ordre de rangement des intervalles sera mémorisé car nécessaire au décodeur).
- **Etape 3** : Initialiser la limite inférieure de l'intervalle de travail à la valeur 0 et la limite supérieure à la valeur 1.
- **Etape 4** : Tant qu'il reste un symbole dans la chaîne à coder :
  - ✓ largeur = limite supérieure - limite inférieure



- ✓ limite inférieure = limite inférieure + largeur x (limite basse du sous intervalle du symbole)
- ✓ limite supérieure = limite inférieure + largeur x (limite haute du sous intervalle du symbole)[30]

➤ **Etape 5** : La limite inférieure code la chaîne de manière unique.

### **I.3.2.1.6 Codage par dictionnaire adaptatif (LZW) (Lempel-Ziv-Welch) ou LZ77**

La société *Unisys* ayant déposée la méthode *LZW*, il en existe une variante non brevetée, le *LZ77*. Elle porte le nom de ses trois inventeurs : *Lempel* et *Ziv* qui l'ont conçue en 1977, et *Welch* qui l'a finalisée en 1984.

Elle utilise un dictionnaire qui n'est pas stocké dans le fichier compressé qu'elle construit dynamiquement, au cours de la compression et de la décompression. Il s'agit cette fois de repérer des motifs composés de séries variables en longueur de bits ou d'octets. Chaque motif est copié dans le dictionnaire et se voit attribué un indice. *Puis chaque motif est remplacé dans le fichier par son indice*, sachant qu'une valeur isolée n'est pas codée. Elle a besoin d'un apprentissage pour être efficace, pour reconnaître des longues chaînes répétées. En effet, l'algorithme ne fonctionne pas sur un nombre fixe de motifs mais *apprend* les motifs du fichier durant la lecture du fichier à compacter. Elle est donc peu efficace sur des petits fichiers. Cette méthode est très rapide. [17]

### **I.3.2.2 Les méthodes de compression avec pertes ou irréversible**

Les méthodes irréversibles permettent des taux de compression assez élevés au prix d'une dégradation de la qualité de l'image. Outre, le taux de compression, une mesure de cette dégradation est nécessaire à l'évaluation des performances de ces méthodes. Plusieurs mesures existent pour l'évaluation des performances de ces méthodes comme le *SNR*, *MSE*, et *PSNR*.

Avec ces méthodes, on peut aussi distinguer :

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

### I.3.2.2.1 Codage par transformée

Dans ces méthodes, l'image de dimension  $N \times N$  est subdivisée en sous images ou blocs de taille réduite (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 coefficients la transformation inverse.

L'objectif de ces transformations est double : il s'agit de :

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

Dans ce cas, on obtiendra une compression effective en codant finement les coefficients des basses fréquences, et grossièrement, voire en les supprimant, les coefficients hautes fréquences. Dans ce but, plusieurs transformations linéaires ont été proposées :

- Transformation de *Karhunen-Loeve (TKL)*.
- Transformation de *Fourrier* discrète (*TFD*).
- Transformation de *Hadamard (TH)*.

- Transformation de *Haar* (*THA*).
  - Transformation en cosinus discrète (*TCD*).
- l'efficacité de ces transformations peut être mesurée par la prise en compte de trois facteurs : l'efficacité de décorrélation, la concentration de l'énergie, et l'existence d'algorithmes rapides pour calculer les transformations.
- Les méthodes de codage par transformation présentent des propriétés d'immunité au bruit de transmission bien supérieures aux méthodes précédentes. Une erreur affecte en effet seulement la valeur d'un coefficient, et sera lissée au décodage lors du calcul de la transformation inverse. L'effet sera ainsi peu visible.

## ❖ Le schéma de compression des méthodes de transformations d'images

Le principe d'un système de codage par transformation est le suivant :

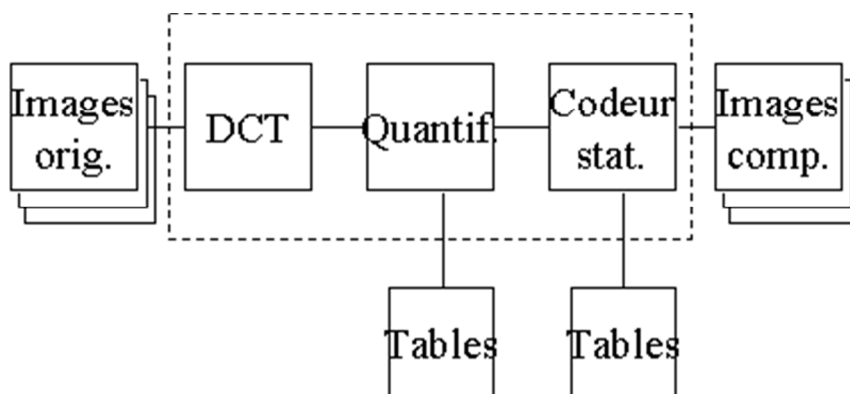


Figure I.4 Schéma de compression JPEG sans perte

## I.3.2.2.1.1 Transformée de fourrier

Elle permet simplement de passer du domaine spatial au domaine fréquentiel. Cette transformée rend donc visible les composantes en fréquence de l'image

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega \cdot t} \cdot dt \quad (1-1)$$

Il est intéressant de noter que nous pouvons revenir au domaine spatial via la transformée de fourrier inverse. Une application brute de cette formule étant extrêmement longue. Une autre façon d'effectuer ce calcul permet de limiter considérablement la durée de cette transformation. C'est ce que l'on appelle la *FFT (Fast Fourier Transform)*. [18]

## I.3.2.2.1.2 Transformation en cosinus discrète DCT

La *DCT (Discret Cosinus Transform)* est une transformée fort semblable à la *FFT*, travaillant sur un signal discret. Elle prend un ensemble de points d'un domaine spatial et les transforme en une représentation équivalente dans le domaine fréquentiel. La *DCT* transforme un signal d'amplitude (chaque valeur du signal représente l'amplitude d'un phénomène) discret bidimensionnel en une information bidimensionnelle de "fréquences"

Les équations qui suivent, donnent respectivement la transformée en cosinus discrète directe et inverse.

$$DCT(i,j) = \frac{1}{2\sqrt{2N}} C(i)C(j) \sum_{X=0}^{N-1} \sum_{Y=0}^{N-1} pixel(X,Y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (1-2)$$

$$pixel(X,Y) = \frac{1}{2\sqrt{2N}} \sum_{X=0}^{N-1} \sum_{Y=0}^{N-1} C(i)C(j) DCT(i,j) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

Où  $C(i) = 1/\sqrt{2}$  si  $i = 0$  ;  $C(i) = 1$  si  $i \neq 0$

*Pixel* ( $X, Y$ ) désigne la valeur du pixel de coordonnées ( $X, Y$ ) et *DCT* ( $i, j$ ) le coefficient repéré par la ligne  $i$  et la colonne  $j$  dans la matrice *DCT*.

La *TCD* est effectuée sur une matrice carrée  $N \times N$  de valeurs de pixels en donnant une matrice carrée  $N \times N$  de coefficients de fréquence. Le temps de calcul requis pour chaque élément dans la *TCD* dépend de la taille de la matrice.

Vu la difficulté d'appliquer la *TCD* sur la matrice entière, celle-ci est décomposée en blocs de taille  $8 \times 8$  pixels (compression *JPEG*). Concrètement, et en terme simple, cette transformation va essayer de faire correspondre des blocs de  $8 \times 8$  de l'image en une somme de fonction basique qui sont données dans la matrice  $8 \times 8$  de la *DCT* (*DCT matrix*).

- **Différence entre la *FFT* et la *DCT***

- la *DCT* est actuellement une version simplifier de la *FFT*.
- seule la partie réelle de la *FFT* est conservé
- beaucoup plus simple en terme de coup de programmation
- la *DCT* est efficace dans la compression de multimédia (*JPEG*)
- *DCT* Beaucoup plus utilisée.

### I.3.2.2.2 La compression par ondelettes

Les ondelettes c'est d'abord une théorie mathématique récente d'analyse du signal, développée dans les années 80. On peut considérer qu'il s'agit d'une extension de l'analyse de *Fourier* [19]. La technologie de compression à base d'ondelettes offre une plus grande finesse au niveau de l'analyse du signal, et permet de mieux s'adapter aux propriétés locales de l'image. La transformation par Ondelettes est une technique de compression d'image fixe très performante.

Quand on enregistre une image en utilisant les ondelettes, on divise sa résolution par deux et on code l'information perdue par des coefficients d'ondelettes. On commence donc à coder les détails les plus fins (les hautes fréquences). On recommence l'opération autant de fois que nécessaire jusqu'à ce que l'image se réduise à 1 pixel. A chaque étape l'image est «lissée» et les détails perdus sont codés en coefficients d'ondelettes.

L'intérêt de la transformation par ondelettes par rapport aux autres méthodes de compression est que celle-ci ne considère pas l'image dans son ensemble pour la coder mais, la travaille par couche, cherchant à enregistrer les détails les plus importants (ceux qui se démarquent le plus du reste du signal) à chaque résolution. [2]

### I.3.2.2.3 Quantification

En général, la quantification apparaît en deuxième lieu dans un processus de compression, pour réduire la quantité d'information, de manière souvent *irréversible*. [31]

Ainsi, une quantification est utilisée pour simplifier la représentation, tout en préservant l'information la plus pertinente. On remplace ainsi les valeurs initiales par un ensemble fini d'éléments qui donneront des résultats acceptables lors de la phase de décompression. Ces éléments peuvent être scalaires lorsque, on remplace l'information en chaque pixel par des nombres entiers; ou des vecteurs lorsque l'on considère des groupes de pixels représentant des configurations typiques.

Dans les deux cas, le principe est décrit par la Figure I.5. Un élément quelconque (scalaire ou vecteur) est remplacé par l'élément du dictionnaire le plus approprié.

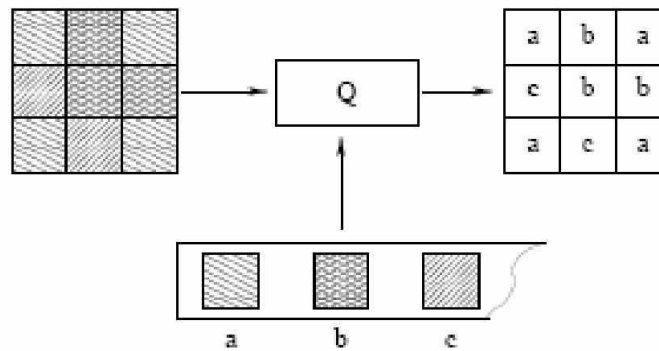


Figure I.5 : Principe de la quantification

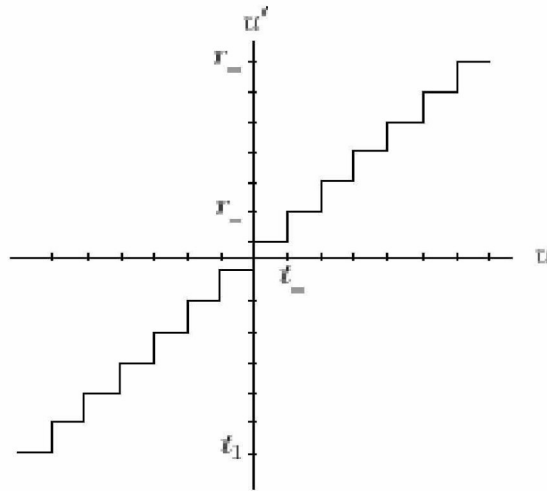
### I.3.2.2.3.1 Quantification scalaire

La quantification scalaire [15], consiste à réduire le nombre de valeurs que peut prendre une grandeur scalaire. C.à.d. remplacer un nombre très grand de symboles par un nombre restreint de codes. C'est une opération irréversible très largement employée en compression (il est à noter que les méthodes de codage utilisant un quantificateur ne sont jamais réversibles parce que l'étape de quantification introduit inévitablement une distorsion).

Un quantificateur scalaire est un opérateur qui associe à une variable continue  $u$  une variable discrète  $u'$  pouvant prendre un nombre plus faible, et fini de valeurs ; il est généralement définie comme une fonction en escalier (Figure I.6), l'intervalle entre chaque valeur étant appelé intervalle de décision.

Pour un nombre de niveaux de quantification fixé  $L$ , on peut choisir les régions de décision

$\{t_k, k = 1, \dots, L+1\}$  ainsi que les seuils de décision  $\{\tau_1, \dots, \tau_L\}$  de façon à minimiser la distorsion entre l'entrée et la sortie . Si  $u$  se trouve dans la région  $(t_k, t_{k+1})$   $u'$  aura pour valeur  $\tau_k$  ?



**Figure I.6 Quantification scalaire uniforme en escalier**

La quantification scalaire porte sur la grandeur physique associée à l'information de niveau de l'image. L'objectif de la quantification est de diviser la gamme dynamique ou dynamique de la grandeur physique en un nombre fini d'intervalles, et d'attribuer à toutes les valeurs du même intervalle une seule valeur, dite valeur quantifiée. Trois questions se posent alors :

- trouver la gamme dynamique.
- choisir le nombre d'intervalles.
- répartir ces intervalles.

Il existe plusieurs formes de quantifications scalaires dont la plus simple est la quantification scalaire uniforme. C'est la forme la plus couramment utilisée en compression d'images.



## I.3.2.2.3.2 Quantification vectorielle

L'approche de la *quantification vectorielle* ( $QV$ ) est une généralisation naturelle de la quantification scalaire ( $QS$ ) pour des vecteurs. Par conséquent elle se place dans la deuxième phase de la chaîne de codage.

La  $QV$ , est un sujet principal de plusieurs recherches et publication dans le domaine de la compression d'images voir [9], [11], [15], et [29]. Nous décrivons par la suite schématiquement le principe de la quantification vectorielle, codage et décodage (voir Figure I.7).

- **Principe**

L'image à coder est découpée en blocs qui ne se chevauchent pas mais qui couvrent toute l'image. Chaque bloc de taille  $k$  est comparé aux imagerie d'un ensemble de blocs, appelé *dictionnaire*  $W = \{w_1, w_2, \dots, w_N\} \subset R^k$ . Ces blocs prédéfinis sont nommés *mots de code* ou vecteurs de reproduction. La comparaison consiste à calculer une mesure de distance entre le bloc (vecteur) à coder et les mots de code.

- **Le codage** s'effectue en ignorant le bloc original et en gardant seulement l'*indice* (l'adresse) du mot de code le plus proche. La distance appliquée est en général la distance euclidienne, ce qui est équivalent à la minimisation de l'erreur quadratique moyenne.
- **Le décodeur** reprend tout simplement les mots de code correspondants Aux indices reçus (transmis ou stockés), et reconstruit ainsi l'image.

Le taux de compression dépend du nombre de mots de code ainsi que de leur taille. Avec une taille du dictionnaire égale à  $N=2^n$ , le taux de compression s'exprime par :

$$T_c = mk/n \quad (1-3)$$

Où  $k$  : est le nombre des pixels dans un bloc ;

$m$  : désigne le nombre de bits par pixel dans l'image originale.

IL est évident que la  $QV$  comme la  $QS$  est une approche irréversible. La qualité du codage est fonction de la taille du dictionnaire, de la taille et du contenu des mots de code.

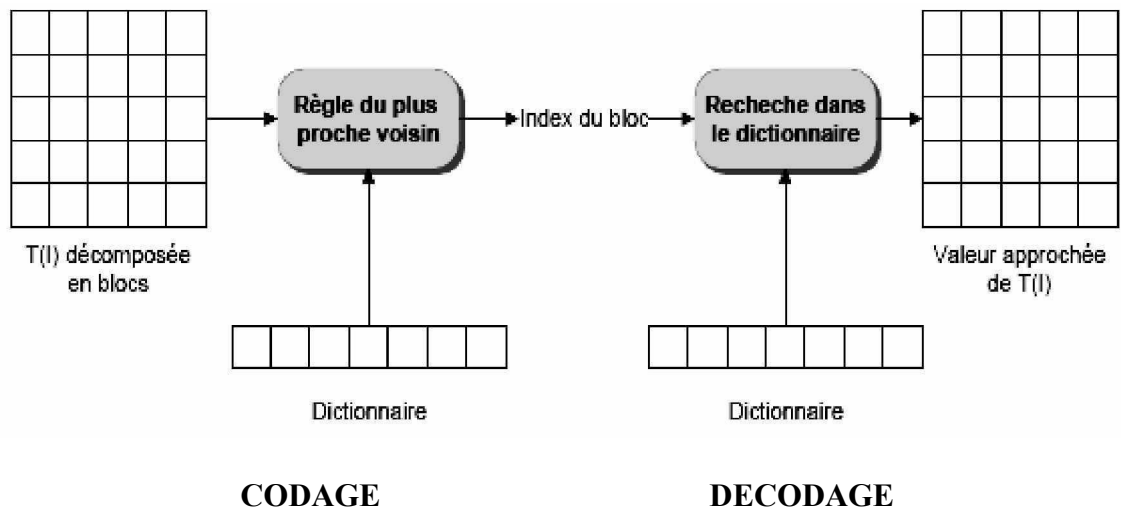


Figure I.7 Le principe du Quantificateur Vectoriel.

## I.4 COCLUSION

Dans ce chapitre, nous avons fait le point sur la compression d'image et présenté les principales méthodes et les différents types de la compression d'image.

L'utilisation d'algorithmes de compression d'images permet en effet, une réduction importante de la quantité de données. A cet effet dans le chapitre suivant nous allons étudier un algorithme très répandu et utilisé par de nombreuses personnes : le Jpeg. Après un bref historique du Jpeg, nous allons présenter son principe et son utilisation. Nous aborderons ensuite, un aspect plus mathématique, concernant les théorèmes fondamentaux utilisés pour la compression.

Enfin, nous expliquerons la méthode de décompression ainsi que les critères d'évaluation de la compression d'image.

## CHAPITRE II

### II.1 INTRODUCTION

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, ou cumulant plusieurs algorithmes, mais tous sont complémentaires. Nous allons étudier un algorithme très répandu et utilisé par de nombreuses personnes : le Jpeg.

Des méthodes de compression d'images sont standardisées et obéissent donc à un processus clairement défini et connu. Les deux principaux standards sont à leur actuelle JPEG pour les images fixes et MPEG pour les images animées. nous n'examinerons ici, que le cas de JPEG mais avec assez de détails.

Le Joint Photographic Expert Groupe (JPEG) correspond à la succession de trois opérations principales : application d'une transformation DCT ; quantification ; codage. Le décodeur effectue exactement en sens inverse ces opérations. Des circuits intégrés existent qui effectuent automatiquement les opérations de codage et décodage.

Pour comprendre le processus de compression JPEG, nous étudions un exemple d'une image quelconque est tout d'abord divisée en sous-images à des blocs  $8 \times 8$  de pixels.

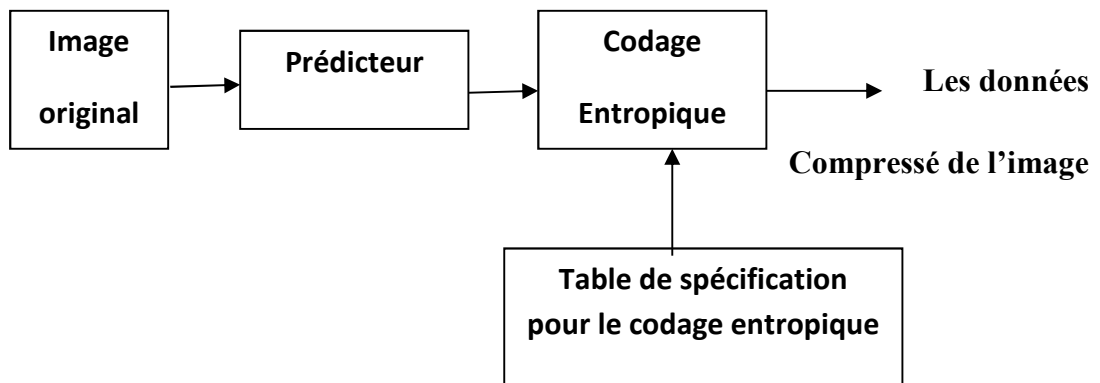
Les techniques de compression se classent en deux catégories principales spécifiées en deux modes de compressions, un mode de compression sans perte basé sur la prédiction du pixel courant à partir de son voisinage (lossless JPEG) et un mode de compression avec perte basé sur la DCT et permettant de choisir la qualité d'image reconstruite via l'utilisation d'un paramètre dédié.

## II.2 LA COMPRESSION SANS PERTE JPEG-LS

Le mode de fonctionnement sans perte JPEG – LS ( lossless en anglais) ou non destructif est une simple méthode basée sur le codage prédictif de la méthode DPCM (Differential Pulse Code Modulation)en utilisant les valeurs des pixels voisins. La figure II.1 montre un tel codeur .le prédicteur est de type simple DPCM où chaque pixel de chaque composante couleur est codé d'une manière différentielle .la prédiction d'un pixel d'entrée x est effectuée à partir des trois pixels voisins de la même composante de couleur.

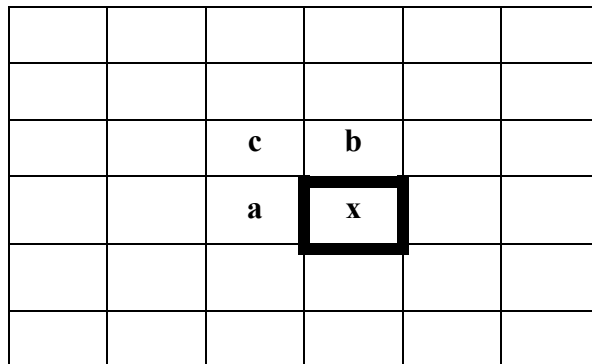
Ce prédicteur ; connu sous le nom Mediane Edge Detector (MED)

Voir figure II.1 [10]



**Figure II.1 schéma de bloc du codeur JPEG en mode sans perte.**

La prédiction est ensuite soustraite de la valeur réelle du pixel à la position x et la différence est codée en entropie soit par la technique *Huffman*, soit par le codage arithmétique. Les spécifications du tableau d'entropie figure II.2 déterminent les caractéristiques de la méthode choisie pour le codage entropique.



**Figure II.2 Le voisinage de la prédiction du prédicteur MED .**

L'idée de technique prédictive est d'éliminer la redondance mutuelle entre les pixels voisins ( a,b,c) et de coder seulement l'information non redondante qui représente l'erreur de prédiction c'est à dire la différence entre la valeur du pixel(x) et sa valeur prédite à partir de ses voisins.

### II.3 LA COMPRESSION AVEC PERTES DU JPEG

Le mode de fonctionnement avec perte JPEG (lossy en anglais) permet des taux de compression nettement supérieurs, mais impose de négliger arbitrairement certaines informations de l'image, en tenant compte de sa nature et de notre perception visuelle.[4]

Pour des raisons à la fois historiques (rapidité de calcul), de précision et enfin de corrélation entre pixels, l'algorithme de compression s'effectue sur des imagerie de bloc, de taille 8×8. C'est à l'application de scinder chacune des composantes de l'image en de tels blocs et de régler le cas des dimensions d'images qui ne sont pas des multiples entiers de 8. Tous les blocs, ou encore vecteurs, sont traité successivement et identiquement au cours des trois étapes successives : une transformation linéaire (DCT), une quantification scalaire, un codage entropique. Le décodeur s'effectue les opérations inverses. [5]

L'algorithme JPEG avec pertes fonctionne conformément au schéma de la figure II.3. L'image originale subit une transformation colorimétrique suivie d'un sous-échantillonnage des composantes de chrominance. L'image résultante est découpée en blocs de  $8 \times 8$ . Une transformation DCT bidimensionnelle est appliquée sur chaque bloc, suivi par l'étape de quantification. Enfin, un codage entropique sans perte est appliqué sur l'ensemble des coefficients quantifiés. [21]

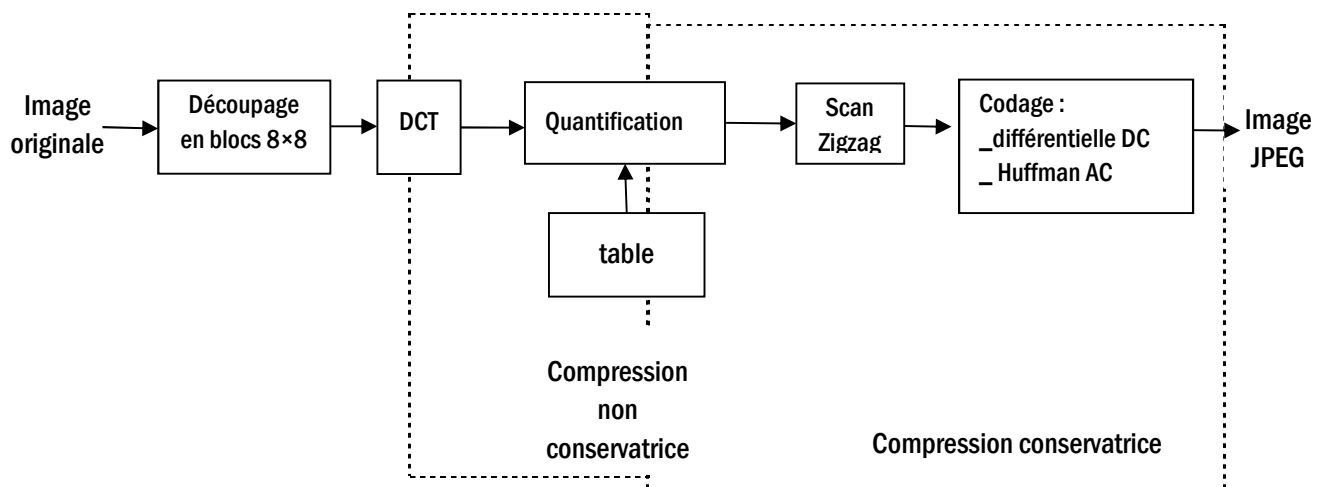


Figure II.3 Chaîne de codage de l'algorithme JPEG [15]

Nous allons détailler chaque partie de la compression JPEG, et en étudiant les fondements.

### II.3.1 description des fonctions clés du standard de compression jpeg avec perte

#### II.3.1.1 transformation DCT : transformée en cosinus discrète bidimensionnelle

La clé du processus de compression est la **DCT** (**D**iscret **C**osine **T**ransform). La DCT est une transformée fort semblable à la FFT : la transformée de Fourier rapide (Fast Fourier Transform), travaillant sur un signal discret unidimensionnel. Elle prend un ensemble de points d'un domaine spatial et les transforme en une représentation

équivalente dans le domaine fréquentiel. Dans le cas présent, nous allons opérer la DCT sur un signal en trois dimensions. En effet, le signal est une image graphique, les axes **X** et **Y** étant les deux dimensions de l'écran, et 'axe des **Z** reprenant l'amplitude du signal, la valeur du pixel en un point particulier de l'écran. La DCT transforme un signal d'amplitude (chaque valeur du signal représente l'amplitude d'un phénomène, ici la couleur) discret bidimensionnel en une information bidimensionnelle de « fréquences » .t [28]

- **DCT**

$$F(u, v) = \frac{2}{N} c(u) \cdot c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{Img}(x, y) \cdot \cos \left[ \frac{\pi}{N} u \left( x + \frac{1}{2} \right) \right] \cdot \cos \left[ \frac{\pi}{N} v \left( y + \frac{1}{2} \right) \right] \quad (2.1)$$

- **La transformation inverse l'IDCT :**

$$\text{Img}(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u) \cdot c(v) \cdot F(u, v) \cdot \cos \left[ \frac{\pi}{N} u \left( x + \frac{1}{2} \right) \right] \cdot \cos \left[ \frac{\pi}{N} v \left( y + \frac{1}{2} \right) \right] \quad (2.2)$$

$$\text{Où } \begin{cases} C(0) = (2)^{-1/2} \\ C(w) = 1 \text{ pour } w = 1, 2, \dots, N - 1 \end{cases} \quad \text{ù}$$

Le calcul de la DCT ne peut pas se faire sur une image entière d'une part parce que cela générerait trop de calculs et d'autre part parce que le signal de l'image doit absolument être représenté par une matrice carrée. Dès lors, le groupe JPEG impose la décomposition de l'image en blocs de 8 pixels sur 8 pixels. la méthode de compression sera donc appliquée indépendamment sur chacun des blocs. Les plus petits blocs en bordure devront être traités par une autre méthode.



La DCT est donc effectuée sur chaque matrice  $8 \times 8$  de valeurs de pixels, et elle donne une matrice  $8 \times 8$  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 conservative si l'on ne tient pas compte des erreurs d'arrondis qu'elle introduit.

Lorsqu'on travaille avec le signal  $\text{Img}(x,y)$ , les axes X et Y représentent les dimensions horizontales et verticales de l'image. Lorsqu'on travaille avec la transformée de cosinus discrète du signal DCT  $(i,j)$ , les axes représentent les fréquences du signal en deux dimensions .[28]

Si on écrit les coefficients  $F(i,j)$  dans une matrice avec j l'indice de ligne et i l'indice de colonne, on a :  $F = {}^t P.A.P$  .

La matrice P possède des propriétés intéressantes. C'est une matrice orthogonale : les vecteurs sont orthogonaux deux à deux et orthonormés. (d'où l'utilité de  $c(i)$  pour ramener la norme du premier vecteur à 1).

Donc  ${}^t P = P^{-1}$  si on considère que la matrice A représenté un endomorphisme de  $\mathbb{R}^8$ , l'expression  $F = {}^t P.A.P$  correspond à un changement de base avec P la matrice de passage.

La DCT inverse en découle simplement :  $A = P F {}^t P$ . (2.3)

On a alors trouvé les 64 images de bases (vecteurs de base) qui permettent de décomposer l'image et plus le coefficient dans la matrice F se trouve dans le coin en bas à droite, plus il correspond à des hautes fréquences. [5]

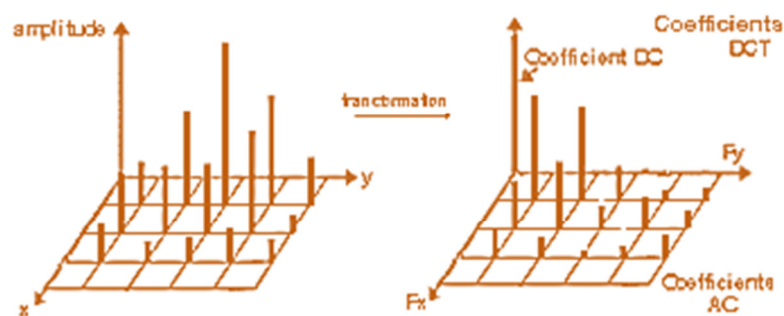
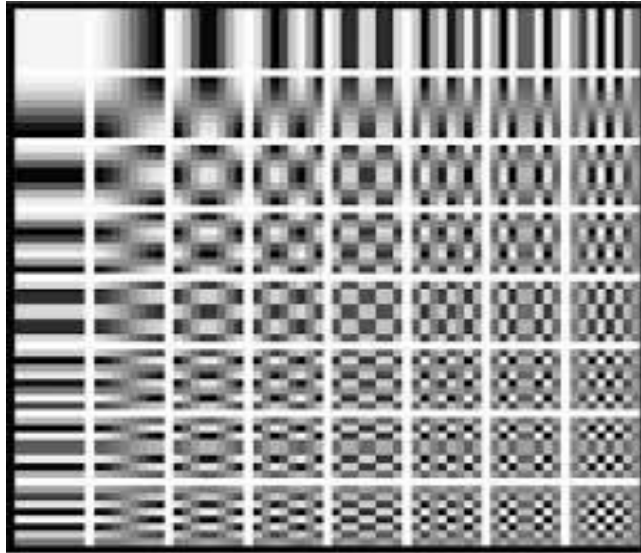


Figure II.4 représentation de la DCT.

L'image est donc décomposée en une «combinaison linéaire» de ces images de base.



**Figure II.5 Vecteur de la base DCT [5]**

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 reconstruction. De plus, la DCT génère des coefficients réels, les plus petits de ceux-ci étant localisés dans une zone fréquentielle où l'œil a une acuité faible.

Une fois la DCT calculée sur un bloc, nous obtenons une matrice carrée des valeurs pour chacune des fréquences. La figure I montre un exemple de compression JPEG sur un bloc de  $8 \times 8$  pixels à 256 niveaux de gris. Les valeurs de la matrice DCT ont été arrondies à l'entier le plus proche. La composante (0,0) est le coefficient continu (1210). Il représente une valeur moyenne de la grandeur d'ensemble de la matrice d'entrée. Ce n'est pas exactement la moyenne au sens statistique du terme, l'ordre de grandeur n'étant pas le même, mais c'est un nombre proportionnel à la somme de toutes les valeurs du signal. Les autres valeurs de la DCT représentent des écarts par rapport à la moyenne. Les valeurs de la matrice de l'indices (0, j) (respectivement (i,0)) sont les composantes continues le long de l'axe Y (resp. X) pour

la fréquence  $J$  (resp.  $i$ ) le long de l'axe  $X$  (resp.  $Y$ ). on remarque une tendance générale des valeurs de la matrice à s'approcher de 0 lorsqu'on s'éloigne du coin supérieur gauche, c'est-à-dire lorsqu'on monte dans les plus hautes fréquences. Cela traduit le fait que l'information effective de l'image est concentrée dans les basses fréquences. C'est le cas de la majorité des images. [11]

### Matrice de pixels d'entrée

140	144	147	140	140	155	170	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

### Matrice DCT

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

### II.4 LA QUANTIFICATION

Le but de la deuxième étape de la méthode JPEG, l'étape de quantification, est de diminuer la précision du stockage des entiers de la matrice DCT pour diminuer le nombre de bits occupés par chaque entier. C'est la seule partie non-conservative de la méthode (excepté les arrondis effectués). Puisque les informations de basses fréquences sont plus pertinentes que les informations de hautes fréquences, la diminution de précision doit être plus forte dans les hautes fréquences. La perte de précision va donc être de plus en plus grande lorsqu'on s'éloigne de la position (0,0). Pour cela on utilise une *matrice de quantification* contenant des entiers par lesquels seront divisées les valeurs de la matrice DCT. Ces entiers seront de plus en plus grands lorsqu'on s'éloigne de la position (0,0). Elle filtre les hautes fréquences. [28]

La valeur d'un élément de la matrice DCT quantifiée sera égale à l'arrondi, à l'entier le plus proche, du quotient de la valeur correspondante de la matrice DCT par la valeur correspondante de la matrice de quantification. Lors de la décompression, il suffira de multiplier la valeur de la matrice DCT quantifiée par l'élément correspondant de la matrice de quantification pour obtenir une approximation de la valeur de la DCT. La matrice obtenue sera appelée *matrice DCT déquantifiée*.

Bien que la spécification JPEG n'impose aucune contrainte sur la matrice de quantification, l'organisme de standardisation ISO a développé un ensemble standard de valeurs de quantifications utilisables par les programmeurs de code JPEG. Les matrices de quantifications intéressantes sont celles permettant de "choisir" la perte de qualité acceptable. Ce choix a été rendu possible grâce aux tests intensifs des matrices. Habituellement, on prend la notification  $Q = (q_{i,j})$  avec  $q_{i,j} = \frac{1}{1+K(1+I+J)}$  avec  $i$  l'indice de ligne,  $j$  l'indice de colonne et  $K$  le facteur de qualité (choisi entre 25).

## CHAPITRE II STANDARDS DE COMPRESSION D'IMAGE-JPEG

On remarque que beaucoup de composantes de hautes fréquences de la matrice quantifiée ont été tronquées à zéro, éliminant leurs effets sur l'image. Par contre, les composantes pertinentes ont été peu modifiées.

Voici un exemple complet de compression et de décompression jpeg :

<u>Matrice de pixels d'entrée</u>								<u>Matrice DCT quantifiée</u>							
140	144	147	140	140	155	170	175	403	-4	2	-1	2	-1	-1	-1
144	152	140	147	140	148	167	170	4	-5	3	-1	-1	1	1	0
152	155	136	167	163	162	152	172	-1	-3	0	0	-1	0	-1	0
168	145	156	160	152	155	136	160	-1	0	1	-1	0	0	0	0
162	148	156	148	140	136	147	162	0	1	1	0	-1	1	1	1
147	167	140	155	155	140	136	162	0	0	-1	0	0	0	0	0
136	156	123	167	162	144	140	147	1	0	0	0	0	0	0	0
148	155	136	155	152	147	147	136	0	0	0	0	0	0	0	0
<u>Matrice DCT</u>								<u>Matrice DCT déquantifiée (décompression)</u>							
1210	-18	15	-9	23	-9	-14	-19	1200	-20	14	-9	22	-13	-15	-17
21	-34	26	-9	-11	11	14	7	20	-35	27	-11	-13	15	17	0
-10	-24	-2	6	-18	3	-20	-1	-7	-27	0	0	-15	0	-19	0
-8	-5	14	-15	-8	-3	-3	8	-9	0	13	-15	0	0	0	0
-3	10	8	1	-11	18	18	15	0	13	15	0	-19	21	23	25
4	-2	-18	8	8	-4	1	-7	0	0	-17	0	0	0	0	0
0	1	-3	4	-1	-7	-1	-2	15	0	0	0	0	0	0	0
0	-8	-2	2	1	4	-6	0	0	0	0	0	0	0	0	0
<u>Matrice de quantification</u>								<u>Matrice de pixels de sortie (décompression)</u>							
3	5	7	9	11	13	15	17	142	143	154	141	133	153	170	170
5	7	9	11	13	15	17	19	139	152	129	151	144	154	163	181
7	9	11	13	15	17	19	21	150	156	139	166	162	163	154	172
9	11	13	15	17	19	21	23	163	145	160	153	151	153	145	154
11	13	15	17	19	21	23	25	168	150	156	145	140	139	141	159
13	15	17	19	21	23	25	27	148	164	133	164	158	140	136	163
15	17	19	21	23	25	27	29	130	159	123	164	165	140	134	145
17	19	21	23	25	27	29	31	148	156	140	148	159	146	153	141

## II.5 CODAGE ENTROPIQUE

Dans cette étape, on doit maintenant parcourir chaque bloc 8×8 afin de coder les 64 coefficients qui le constitue. D'après les tables de quantification précédentes, il est facile de voir que plus l'on va aller dans les hautes fréquences (coin inférieur droit du bloc), plus on a de chance de trouver des valeurs nulles. Afin de minimiser l'entropie globale après le parcours (et ainsi atteindre des taux de compression plus importants), on va devoir constituer des suites de valeurs nulles les plus longues possibles. [6]

Pour cela, la meilleure solution consiste à utiliser un parcours Zigzag. Celui-ci parcourt des différentes fréquences en partant du coefficient DC (à l'emplacement (0,0)) jusqu'aux fréquences les plus hautes orthogonalement à la diagonale, maximisant ainsi le nombre de valeurs nulles consécutives en fin de parcours [6] une fois la procédure de parcours terminée, on dispose d'une suite de 64 coefficients (1 coefficient DC et 63 coefficients AC).

Ces coefficients sont codés séparément, en effet, la composante DC du bloc 8×8 étant fortement corrélée avec la composante DC du bloc précédent, ces coefficients vont être codés différemment des coefficients AC, en utilisant un codage différentiel de type DPCM. C'est la différence  $E_k$  qui sera codée ; et non pas le coefficient lui-même ; selon l'équation : (3)

$$E_k = DC_k - DC_{k-1} \quad (2.4)$$

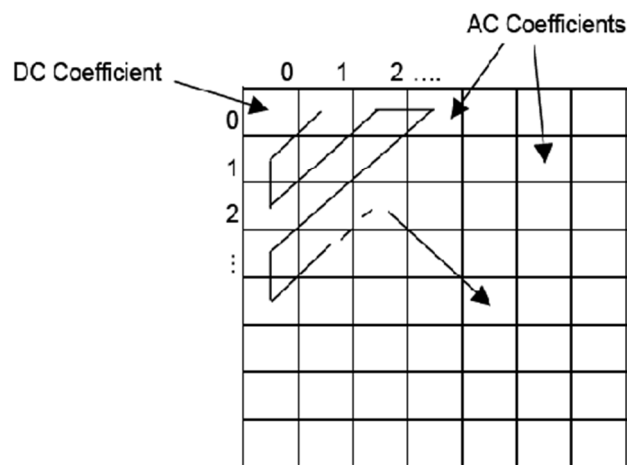
La figure II.6 Représente le codage des coefficients DC. Par contre, les 63 coefficients AC commençant à partir du coefficient AC (0,1) sont codés par une combinaison de RLE et *Huffman* ou codage *arithmétique* après un balayage de *Zigzag*. L'idée est que la séquence des coefficients AC contient juste quelques nombres non nuls, avec une séquence de zéros entre eux. L'algorithme JPEG utilise des tables de correspondances entre les différentes valeurs de coefficient et de la taille de séquence de zéros pour coder efficacement un bloc. Seuls les coefficients non nuls

ne sont codés, mais d'une façon qui prend en compte le nombre de zéros qui les précèdent. Pour chacun de ces coefficients on obtient un code composite fait de deux codes : un code de longueur fixe et un code de longueur variable.

Le code de longueur fixe est composé de 8 bits, sa représentation binaire est **ZZZZCCCC**.

Les quatre bits de poids fort **ZZZZ** servent à représenter la longueur de la plage de zéros qui précède le symbole courant, les quatre bits de poids faible **CCCC** servent à représenter la catégorie dont fait partie la valeur absolue du coefficient. La catégorie **C** est l'intervalle délimité par des puissances de deux dont fait partie le coefficient. Pour une valeur absolue de coefficient appartenant à l'intervalle  $[2^{C-1}, 2^C - 1]$  la catégorie est **C**. Le code de longueur variable sert à préciser le signe de coefficient et sa position dans l'intervalle. Ce code a donc une longueur **C**. [24]

Les codes de longueur fixe sont codés par un code de *Huffman*, afin de gagner encore en compression. Pour coder une plage de zéros dont la longueur est supérieure ou égale à seize, on utilise un code **ZZZZCCCC = 11110000** et un ou plusieurs autres codes composites.



$$E_x = DC_K - DC_{K-1}$$

**Figure II.6 Balayage zig-zag [4]**

Ce résultat est ensuite compressé selon un algorithme RLE basé sur la valeur 0 (le codage RLE intervient uniquement sur cette dernière), puis un codage entropique de type *Huffman* ou *arithmétique*.

### II.5.1 Le codage RLE

RLE est l'une des méthodes les plus anciennes, les plus simples et la plus utilisée. Tout son secret consiste à identifier et supprimer des redondances d'information en les codant sous une forme plus compacte. RLE s'emploie à réduire la taille physique d'une répétition de chaîne de caractère. Cette chaîne répétée est appelée un passage (run) et est typiquement codée avec 2 bytes. Le premier byte représente le nombre de caractères dans le passage et est appelé le compteur de passage (run count). Il peut prendre une valeur comprise entre 0h et 128h ou 256h.

Le second byte est la valeur du caractère dans le passage qui peut prendre la valeur 0h à FFh. Ce dernier byte est appelé la valeur du passage (run value).

Le problème est bien évidemment qu'un fichier ne contenant aucune répétition aura une taille deux fois plus importante que l'original. En pratique, on encode les répétitions sur trois caractères : le premier est un caractère spécial indiquant la présence d'une répétition ; le second indique le nombre d'occurrences et le troisième la valeur à répéter. Ainsi la chaîne « aiiiiibcddddde » sera encodée avec, pour caractère spéciale le signe # : « a#6ibc#5de », ce qui représente une compression de 33,33%. Le codage RLE est notamment employé dans les formats d'image PCX ou BMP, ou bien avant un autre algorithme de compression (notamment HUFFMAN dans le cas de JPEG).

### II.5.2 Algorithme de Huffman

Comme pour le codage de Shannon-Fano, les probabilités d'apparition des symboles sont placées dans un tableau trié par ordre décroissant de probabilités. L'algorithme de Huffman est implémenté suivant une structure d'arbre.

Le principe de cet algorithme consiste à regrouper les deux symboles de probabilités la plus faible pour en faire un nouveau symbole dont la probabilité est la somme des probabilités de ces deux symboles. On itère cette opération et à chaque étape le nombre de symboles diminue. On construit de cette manière un arbre dont les feuilles sont les symboles à coder et les embranchements les codages intermédiaires.



### Algorithme de Huffman:

Considérons à nouveau l'exemple précédent :

Il s'agit de coder la phrase : "Le codage est indispensable". Le tableau des probabilités d'occurrence est le suivant :

SYMBOL	NOMBRE DE FOIS	Probabilités
E	5	5/24
S	3	3/24
A	2	2/24
D	2	2/24
I	2	2/24
N	2	2/24
L	2	2/24
B	1	1/24
G	1	1/24
P	1	1/24
T	1	1/24
O	1	1/24
C	1	1/24

**Tableau II.1 Le tableau des probabilités d'occurrence**

On applique successivement les étapes de l'algorithme selon le principe de l'algorithme précédemment indiqué.

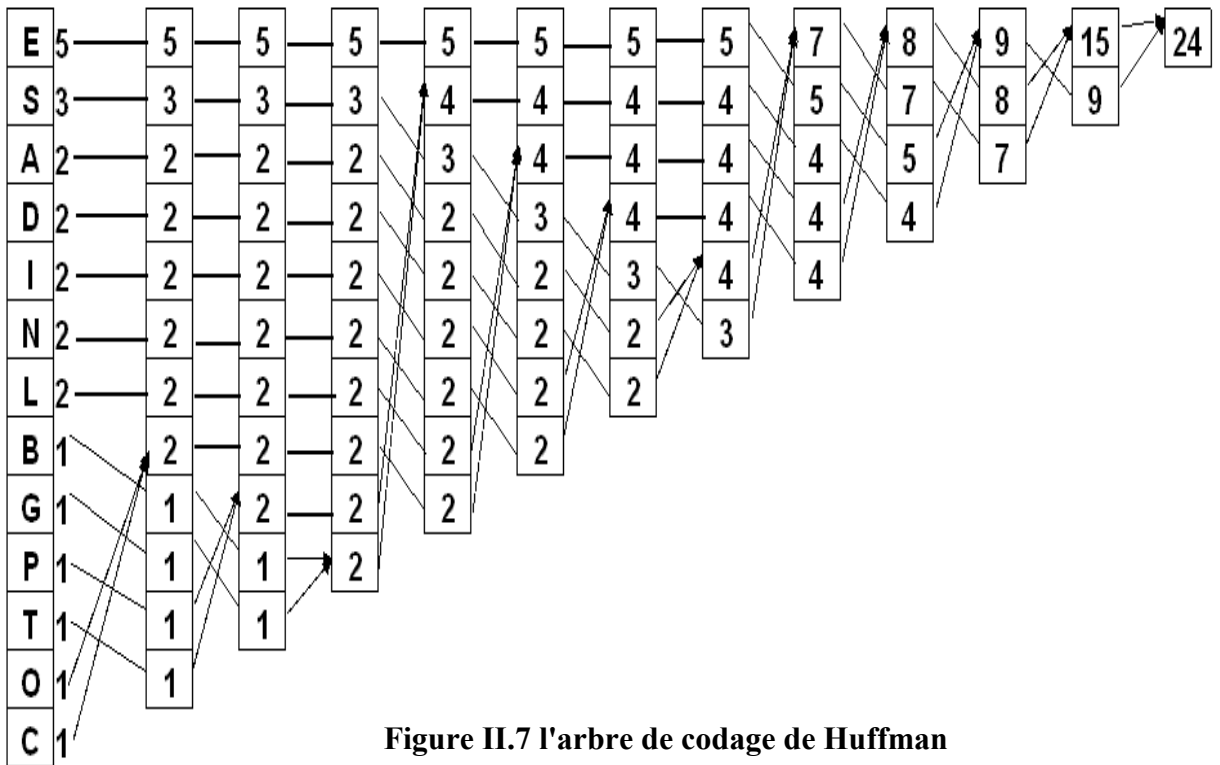


Figure II.7 l'arbre de codage de Huffman

L'arbre de Huffman associé est présenté ci-après :

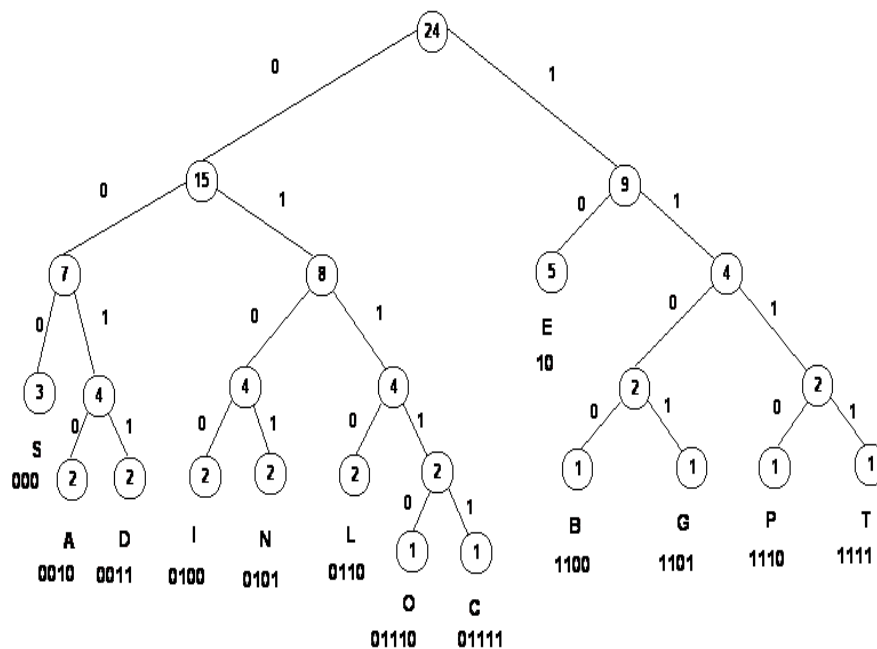


Figure II.8 Arbre de Huffman

### II.6 DECOMPRESSION D'UNE IMAGE FIXE DE FORMAT JPEG

Alors avant de citer les critères de l'évaluation de la compression d'image, il faut d'abord parler de la décompression d'image fixe.

#### II.6.1 Schéma synoptique de compression

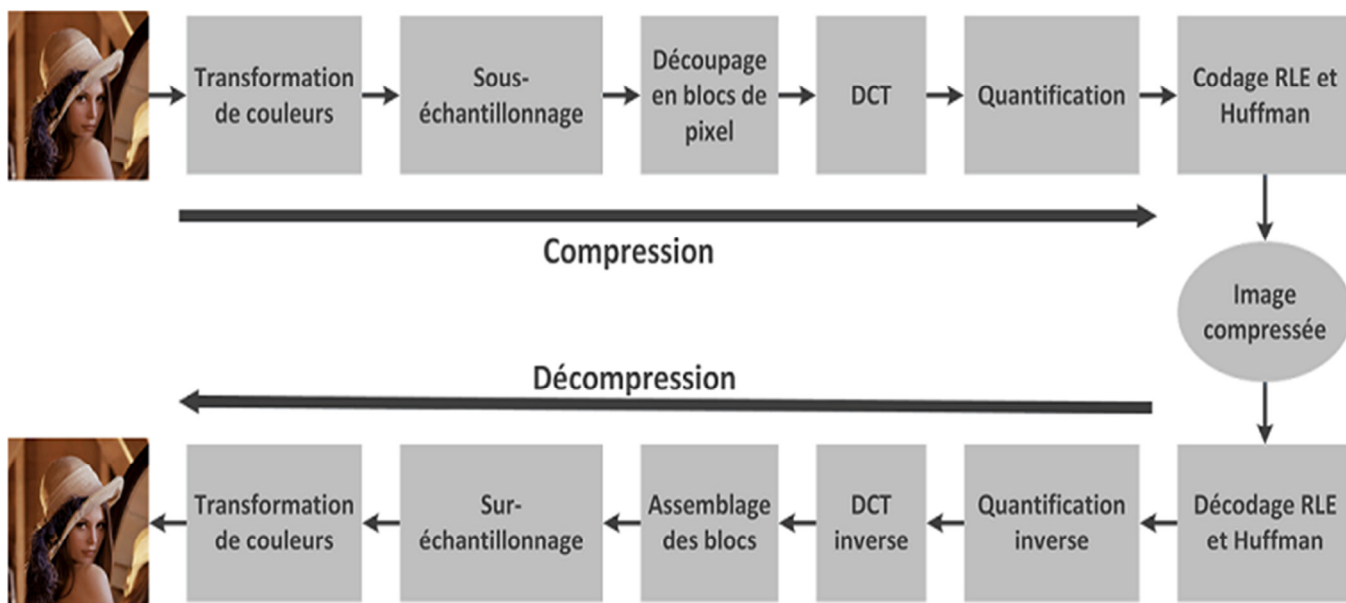


Figure II.9 Schéma de principe de la compression par transformation

La décompression d'image consiste à effectuer les mêmes étapes de la compression mais d'une façon inverse. Pour cette raison, nous allons nous restreindre à ses étapes principales. Un logiciel qui va décompresser une image au format JPEG va suivre les étapes principales suivantes :

- ouverture du fichier concerné
- Remise en forme de la matrice quantifiée, en suivant le chemin inverse de la méthode de Huffman.
- Produit terme à terme des coefficients de la matrice DCT quantifiée, par les coefficients de la matrice de quantification.

- Régénération de la matrice de pixels en appliquant la DCT inverse.

La DCT inverse se fait rapidement en utilisant les notations matricielles. La décompression est un processus plus rapide que la compression.

### II.7 CRITERES D'EVALUATION DE LA COMPRESSION D'IMAGE

Lorsqu'on utilise des méthodes de compression non *réversibles*, trois critères sont généralement retenus pour l'évaluation de la compression : *la distorsion*, *le taux de compression*, et *la charge de calcul* induite par la méthode. La pertinence des méthodes de compression ne dépend pas seulement du taux de compression obtenu, mais aussi de la perte de la qualité entre l'image originale et l'image restituée.

D'une façon générale, il existe deux types de critères permettant d'évaluer cette qualité : les critères subjectifs et les critères objectifs.

#### II.7.1 Critère subjectif :

Le critère subjectif est tout simplement le critère visuel. L'œil est un outil essentiel pour apprécier la qualité, la netteté, les contours et le contenu d'une image.

L'évaluation de la qualité visuelle d'une image n'est pas évidente.

Dans ce type de critère ; basé sur l'évaluation de la qualité par des observateurs humains ; l'image est classée dans des catégories de qualité : excellente, bonne, acceptable, mauvaise et inacceptable.

Le critère consiste à faire attribuer une note de qualité ; entre 5 (qualité excellente) et 1 (qualité inacceptable) ; par un ensemble d'observateur. Cette note est obtenue en calculant la moyenne des résultats d'une série de tests standards où les observateurs donnent leurs avis sous la forme de point pour évaluer la qualité de l'image. Ces tests standards exigent que les observateurs examinent les images dans les mêmes conditions, telles que la taille de l'image, la durée d'exposition et l'environnement lumineux dans lequel se déroule l'expérience. [14]

Echelle à cinq notes		
Qualité		Dégradation
Excellente	5	Imperceptible
Bonne	4	Perceptible mais non gênant
Assez bonne	3	Légèrement gênant
Médiocre	2	Gênant
Mauvaise	1	Très gênant

**Tableau II.2 exemple d'échelle de notation [12]**

Toutefois ces tests sont lourds à mettre en œuvre, chers et surtout très longs et ne constituent donc pas une solution pratique pour les différents opérateurs. [12]

Cependant, cette évaluation ne peut être que subjective puisque il n'existe aucune mesure correcte pouvant traduire fidèlement la perception de l'œil. Sans pour autant négliger ce critère, il est préférable d'introduire des critères plus objectifs. [25]

### II.7.2 Critère objectif

Les critères objectifs sont basés sur des critères mathématiques pour évaluer la qualité des images.

Généralement, les mesures utilisées en compression sont l'erreur moyenne quadratique (MSE) et le rapport signal à bruit SNR (Signal to Noise Ratio).

#### II.7.2.1 Erreur quadratique moyenne

La distorsion est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale. L'erreur quadratique moyenne (MSE) est la plus simple mesure de qualité de l'image reconstruite. Elle est définie par :

$$MSE = \frac{1}{N \times M} \times \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I(i,j) - \hat{I}(i,j))^2 \tag{2.5}$$

Où  $I$  : Image originale et  $\hat{I}$  : Image reconstruite.

## II.7.2.2 Rapport signal à bruit en pic

La mesure la plus couramment utilisée par la communauté internationale est la mesure du rapport signal à bruit SNR. Elle est donnée par :

$$\text{SNR} = 10 \times \log_2 \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} I(i,j)^2}{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I(i,j) - \hat{I}(i,j))^2} \quad (2.6)$$

Il existe différentes variantes à cette mesure. La plus utilisée est le PSNR (Peak Signal to Noise Ratio), évaluant le rapport signal à bruit en pic et se mesure en décibel (dB). Pour les images en niveau de gris le PSNR est calculé par :

$$\text{PSNR} = 10 \times \log_2 \frac{255^2}{MSE} \quad (2.7)$$

Pour les images en couleurs le PSNR est défini par (10)(11)

$$\text{PSNR} = 10 \times \log_2 \frac{255^2 \times 3}{MSE(R) + MSE(G) + MSE(B)} \quad (2.8)$$

Où MSE (.) représente les erreurs quadratiques moyennes dans les trois plans R, G et B.

## II.7.2.3 Taux de compression

Le taux de compression CR (Compression Ratio) caractérise un facteur incontournable dans le domaine spécifique de la compression. Il donne une mesure de la réduction de la quantité d'information à stocker. Il existe différentes variantes de cette mesure. [24][25][14]

$$\text{CR} = \frac{\text{la taille de l'image originale en bits}}{\text{la taille de l'image compressé en bits}} \quad (2.9)$$

Une autre variante est largement utilisée qui est le *bpp* (*bit per pixel*). Pour mesurer le pouvoir de compactage d'une méthode de compression, on utilise la notion de *débit binaire*. Ce qui donne une mesure en *bit par pixel*. [28]

$$bpp = \frac{\text{nombre de bits nécessaires pour coder l'image}}{\text{nombre de pixel de l'image}} \quad (2.10)$$

Les équations (2.11) et (2.12) permettent de calculer le *bpp* pour une image niveau de gris et couleur respectivement :

$$bpp = \frac{8 \text{ bits}}{CR} \quad (2.11)$$

$$bpp = \frac{3 \times 8 \text{ bits}}{CR} \quad (2.12)$$

### II.7.2.4 Temps de calcul

La complexité d'un algorithme ou d'une méthode de compression est difficile à évaluer. Elle recouvre divers aspects tels que sa rapidité, l'existence de composants intégrés, la charge de calcul etc. la complexité fait également intervenir les contraintes spécifiques à l'application, les performances des processeurs et le coût économique.

La complexité d'un algorithme peut se décrire par le temps nécessaire pour son exécution. Les machines ayant des architectures différentes et par conséquent des vitesses d'exécution différentes, on préfère exprimer la charge de calcul par une quantité absolue qui est le nombre d'opérations mathématiques. Ceci permet d'établir une différence entre les algorithmes rapides et les algorithmes lents, entre les algorithmes fonctionnant en temps réel et ceux ne pouvant pas le faire.

En effet, il serait dommage, dans une application de transmission, que le temps gagné par une réduction de la taille des données à transmettre soit inférieur au temps passé à la compression/décompression. Cela sera cependant moins crucial dans les applications visant l'archivage des données. Il est important de noter que la charge de calcul n'est pas forcément identique entre le codeur et le décodeur. [21]

### II.8 AVANTAGES ET DEFAUTS DU JPEG

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

#### II.1 Les avantages du Jpeg

Les plus gros avantages du format Jpeg sont la simplicité, l'universalité et la légèreté.

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

#### II.2 Les défauts

Le format Jpeg a les défauts de ses qualités :

- **La compression des données** : la légèreté due à la compression amène une perte de qualité (la compression du Jpeg est destructrice et irréversible).
- **Pas de marge d'erreur** : plus embêtant, une photo au format Jpeg doit être parfaitement exposée au moment de la prise de vue car la qualité des fichiers Jpeg ne permet pas de rattraper une sous-exposition ou une surexposition.
- **Développement définitif** : à la différence du Raw, les photos au format Jpeg sont en quelque sorte « développées » par l'appareil et les réglages tels que la Balance couleur ou les styles d'images appliqués définitivement. [32]



### II.8 CONCLUSION

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.

### CHAPITRE III

#### III.1 INTRODUCTION

Le but de ce projet est de découvrir le fonctionnement d'un système complet de compression d'images, par réalisation des différentes étapes par simulation sur Matlab.

Le système permettra la compression et la décompression des images, de manière à pouvoir évaluer la qualité de la compression par des critères mathématiques tels que la puissance du rapport signal sur bruit, le taux de compression ou bien l'erreur quadratique moyenne (EQM).

Ce chapitre est consacré aux expérimentations numérique et pour objectif de valider la pertinence de la méthode (JPEG).

#### III.2 RESULTATS OBTENUS

Les résultats de simulation ont été obtenus avec un programme réalisé sur Matlab où j'ai utilisé la boîte à outils de traitement d'image (image processing toolbox) pour l'opération de lecture, conversion et affichage des graphiques.

##### III.2.1 Image de teste utilisé

Six images de test naturelles aux niveaux de gris au format BMP de taille de 512×512, 8 bits / pixel, en sont utilisées pour montrer l'efficacité de la méthode de compression et décompression JPEG.

##### III.2.2 Interprétation des résultats

Après avoir réaliser la compression et la decompresion des différentes images niveau de gris ,nous avons obtenu des résultats ( au dessous) qui sont associer à des critères objectifs qui conditionne la qualité et la pertinence de la compression d'une image .

Quatre critère sont choisis parmi lesquels : PSNR, Taux de Copmression, BPP, EQM.



Image originale (barbara\_gray) (1)

Image Originale



DCT



Image décompressée



**Figure III.2.1 compression JPEG (barbara\_gray)**

Taille originale : 262144

taux\_compression = 1.1316

taille\_img\_compressor = 231648

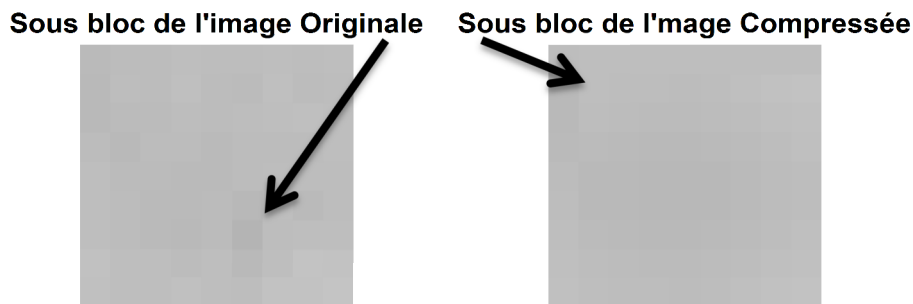
EQM = 33.3746/PSNR = 32.8966/

bpp = 0.8837

On remarque que la taille de l'image a été compressé avec un taux de compression de 1.13 ce qui est acceptable on aurait pu avoir plus si on aurait pris une image plus corrélée que celle-ci .

On remarque aussi que la DCT dans chaque sous blocs fait une concentration des coefficients basses fréquences (en blanc).

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  $EQM=33.3746$  cela veut dire qu'il y a une erreur d'une différence moyenne de 33.37 environs entre les éléments des deux matrices images mais elle n'est pas notable seulement à l'échelle d'un sous bloc  $8 \times 8$ , où on peut remarquer cette différence sur la figure III.2



**Figure III.2.2 comparaison des sous bloc  $8 \times 8$  de l'image originale et l'image compressée (barbara\_gray)**

Certains pixel ne sont pas les mêmes dans les deux sous blocs et non pas le même niveau de gris, cela est due au fait que l'étape de la quantification est une étape avec perte, où on a arrondi à l'entier près après avoir divisé sur la table Q.

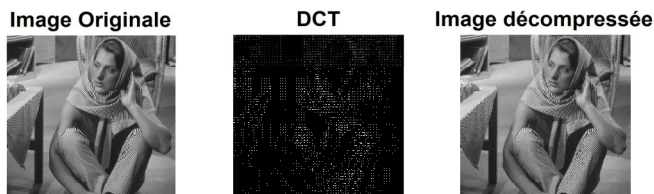
Le rapport crête signal sur bruit 32.90 dB est acceptable, ce qui montre qu'on a fait une compression efficace en matière de taille/ qualité. Le nombre de bit par pixel a été réduit à 0.88 cela nous a fait perdre du poids au fichier image.

Si on veut aller plus loin, on ajoute la fonction suivante qui va générer d'autre table de quantification  $8 \times 8$  en fonction de facteur de qualité (FQ).

$$Quantification[i][j] = 1+(i+j+1) \times FQ$$

Nous obtiendrons les résultats suivants pour différents facteurs de qualité :

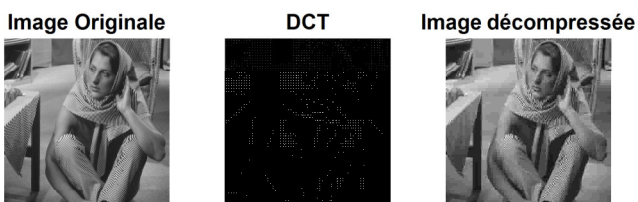
- QF = 10



taille\_originale = 262144  
taille\_img\_compressor= 117965  
taux\_compression = 2.2222  
EQM = 90.3645  
PSNR = 28.5708  
bpp = 0.4500

Figure III.2.3 compression JPEG avec un FQ = 10

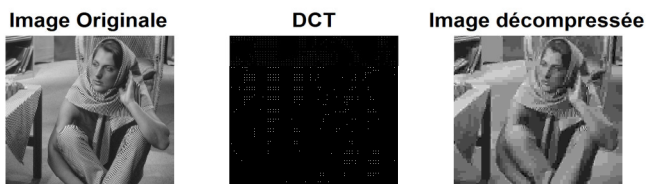
- QF = 30



taille\_originale = 262144  
taille\_img\_compressor= 48051  
taux\_compression = 5.4555  
EQM = 254.5273  
PSNR = 24.0735  
bpp=0.1833

Figure III.2.4 compression JPEG avec un FQ = 30

- QF = 50



taille\_originale = 262144  
taille\_img\_compressor= 36519  
taux\_compression = 7.1783  
EQM = 343.3031  
PSNR = 22.7740  
bpp = 0.1393

Figure III.2.5 compression JPEG avec un FQ = 50

FQ	Les critères objectifs			
	CR	EQM	PSNR	BPP
FQ=10	2,2222	90,3645	28,5708	2
FQ=30	5,4555	254,5273	24,035	0,18
FQ=50	7,1783	343,3031	22,774	0,1393

Tableau III.2.1 Les critères objectifs en fonction FQ (image barbara\_gray)

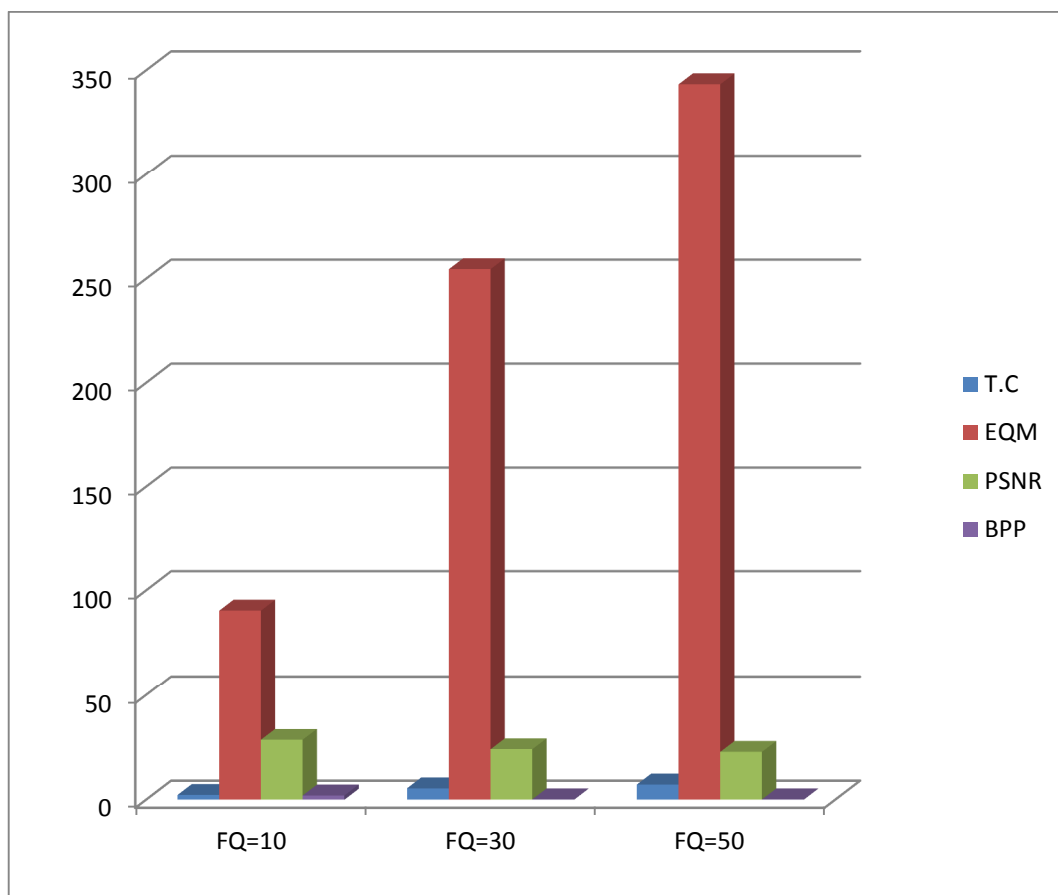


Figure III.2.6 comparaison des critères objectifs en fonction FQ (image barbara\_gray)

On remarque que plus qu' on choisit un facteur de qualité grand, plus le taux de compression est meilleur du fait qu'on augmente le seuil de FQ, mais l'erreur quadratique moyen augmente aussi ce qui va influencer l'efficacité de la compression constaté visiblement et aussi au vu de rapport PSNR médiocre du fait qu'on a perdu beaucoup de données pendant l'étape de quantification et cela est devenue très visible, ce qui rend l'image exploitable à certaines limites.

## III.2.3 Autres exemples de compression JPEG :



Image originale (pout)(2)

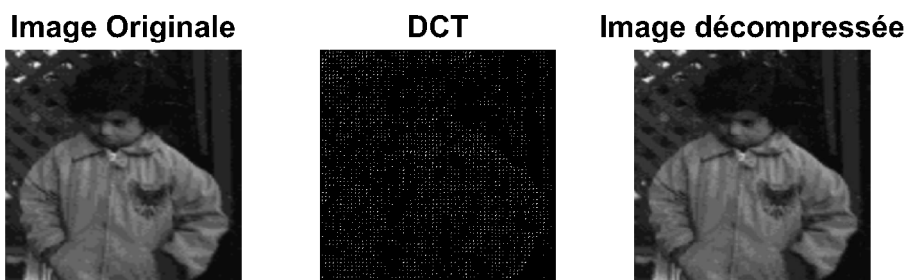


Figure III.2.7 compression JPEG (pout)

Taille originale : 262144

taux\_compression : 2.8323

taille\_img\_compressor : 92555

EQM = 2.1129 /PSNR = 44.8821 /bpp = 0.3531



Figure III.2.8 comparaisons des sous bloc 8x8(pout)





Image originale(Hilbert) (3)

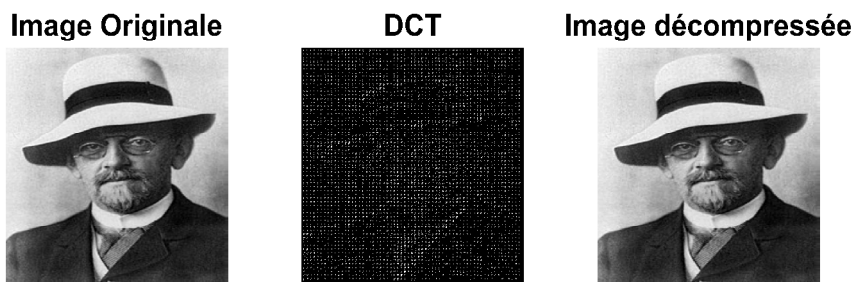


Figure III.9 compression JPEG (Hilbert)

Taille originale : 262144

taux\_compression : 1.6875

taille\_img\_compressor : 155345

EQM = 5.8832 /PSNR = 40.4346 /bpp = 0.5926

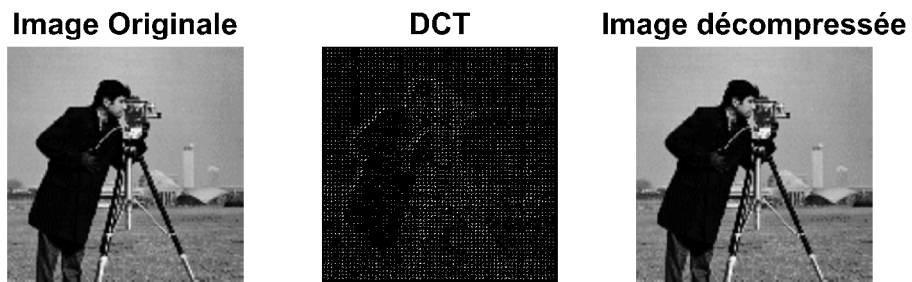
Sous bloc de l'image Originale      Sous bloc de l'image Compressée



Figure III.10 comparaisons des sous bloc 8× 8 (Hilbert)



**Image originale(cameraman)(4)**



**Figure III.11 compression JPEG (cameraman)**

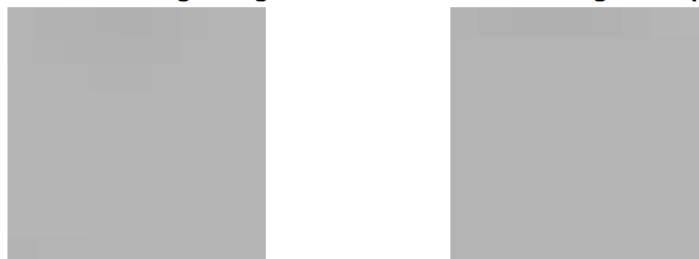
Taille originale : 262144

taux\_compression : 2.1945

taille\_img\_compressor : 119454

EQM = 3.3553/PSNR = 42.8735 /bpp = 0.4557

**Sous bloc de l'image Originale      Sous bloc de l'image Compressée**



**Figure III.12 comparaisons des sous bloc 8× 8  
(cameraman)**

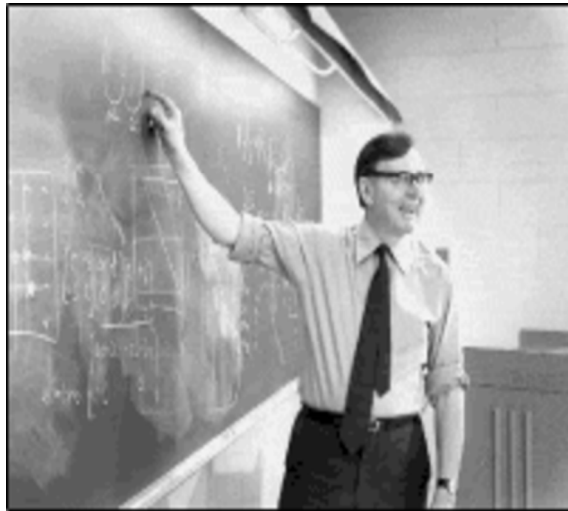


Image originale (Wilkinson)(5)

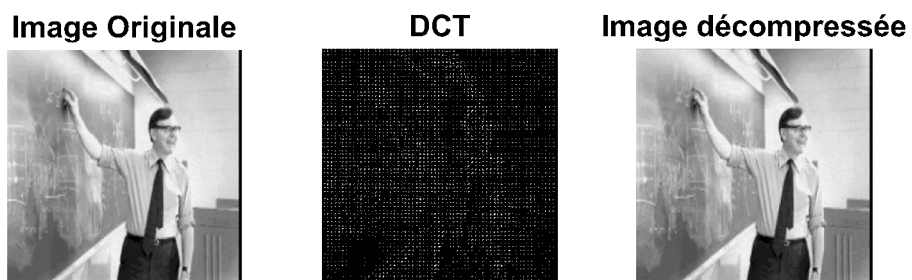


Figure III.13 compression JPEG

Taille originale : 262144

taux\_compression : 2.1582

taille\_img\_compressor : 121462

EQM = 3.9132/PSNR = 42.2055 /bpp = 0.4633

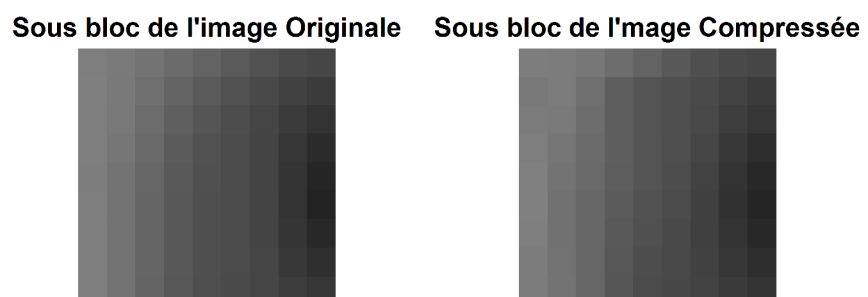


Figure III.14 comparaisons des sous bloc 8× 8



Image originale(eight)(6)

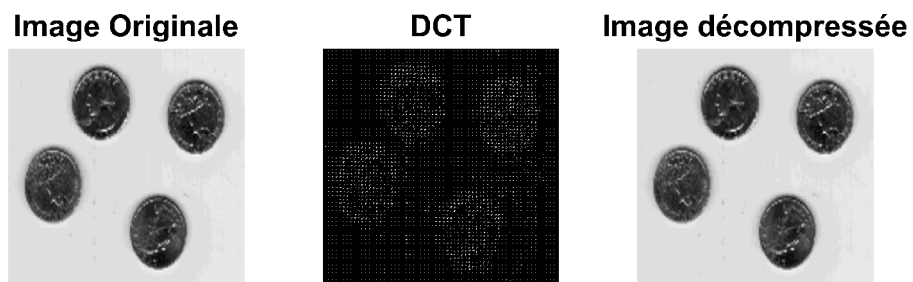


Figure III.15 compression JPEG(eight)

Taille originale : 262144

taux\_compression : 3.0976

taille\_img\_compressor : 84628

EQM = 2.1408/PSNR = 44.8250 /bpp = 0.3228



Figure III.16 comparaisons des sous bloc 8x 8 (eight)

## III.2.4 Relation entre le PSNR et le taux de compression

TAUX DE COMPRESSION (RC)	PSNR
0	0
1,13	32,9
1,69	40,43
2,16	42,2
2,19	42,5
2,83	44,9
3	44,82

Tableau III.2 PSNR en fonction de taux de compression

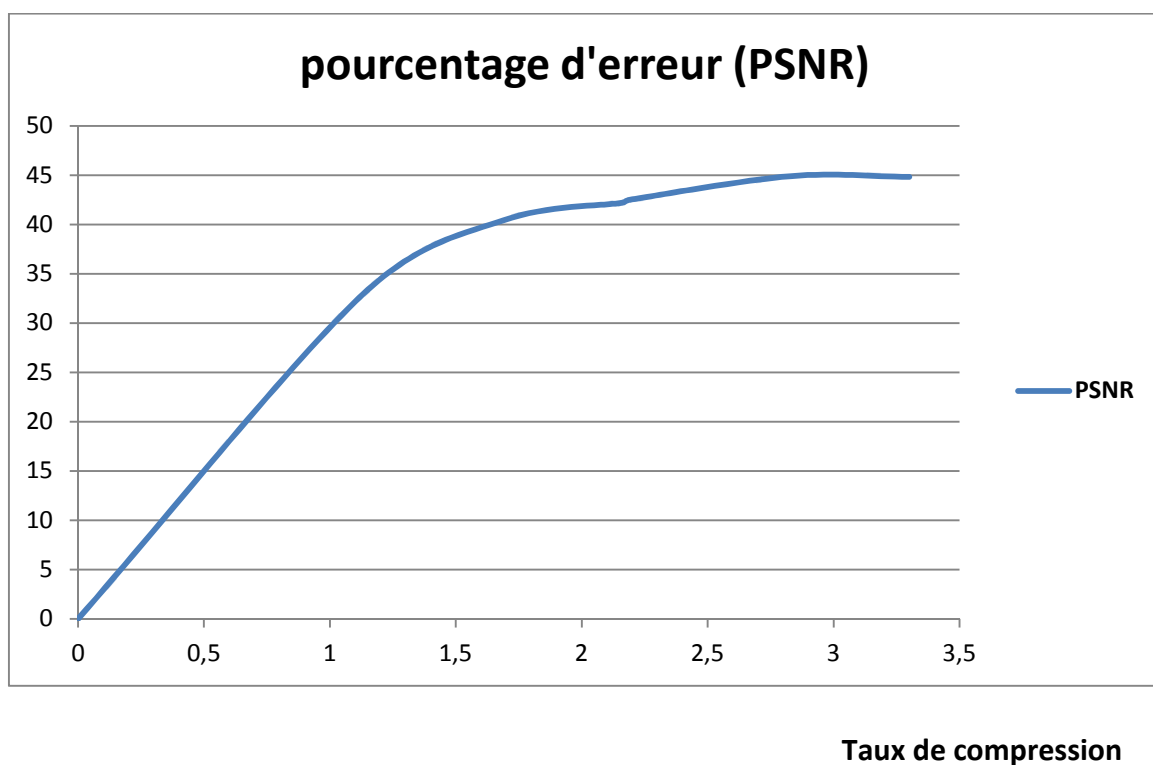


Figure III.17 pourcentage d'erreur en fonction de taux de compression

Le PSNR permet d'évaluer la ressemblance entre l'image originale et celle compressée, plus il est fort et plus les images sont proches. A l'inverse, un faible PSNR indique qu'une partie de l'information de l'image est perdue.

Nous remarquons dans la figure III.2 que l'Erreur augmente avec le RC( taux de compression ). en effet, plus on compresses l'image de départ ,plus qu'on perd une partie d'information , c'est pourquoi l'evolution de l'erreur est complémentaire à celle de la compression.

### III.3 CONCLUSION

Le procédé de compression JPEG permet en général de compresser l'image sans perte significative de qualité.

La compression JPEG pour qu'elle soit efficace doit être effectuée de façon optimale afin de ne pas perdre trop la qualité de l'image et en parallèle réduire le poids de l'image, c'est-à-dire trouver un compromis rapport taille/qualité qu'est le standard JPEG.

La compression JPEG en mode codage DCT séquentielle est une compression avec perte non adapté aux besoins de précision, mais serait le format le mieux adapté pour les mémoire (grâce à JPEG que l'appareil photo numérique a fait leurs apparitions) ou pour le web (rapidité de transmission), cela dit, un format nouveau a été développer pour les application qui demande une fidélité maximum de l'image traité à l'image réel, comme l'imagerie médical, c'est le format JPEG qui a les mêmes avantage du JPEG mais basé sur une compression sans perte on utilisant une nouvelle transformé appelé la transformé ondelette.

### CONCLUSION GENERALE

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 matrice de quantification paramétrable.

L'avenir de la compression d'images semble maintenant appartenir au nouveau standard du Joint Picture Expert Groupe, JPEG.

En fin, on peut conclure qu'il existe de nombreuse manière de coder, de représenter et de compresser les images.il existe une multitude de formats pour compenser la diversité des images, différentes par leurs dimension, leur nombre de couleur et leurs méthode de représentation (vectorielle et bitmap). Chaque format est complémentaire des autres.

Aujourd'hui, avec la considérable avancée technologique, les algorithmes se permettent de réaliser de nombreux calculs pour compresser et décompresser une image. Les ratios devenant de plus en plus grand, sans pertes significative et réelle de la qualité de l'image. Les dernières méthodes de compression (fractale et ondelette) sont particulièrement prometteuses.

Elles montrent la nécessité d'associer à la programmation informatique, les mathématiques. L'avenir de la compression ne pourra s'opérer qu'au travers d'algorithmes mathématiques, et les progrès de la recherche mathématique entraîneront sûrement une avancée dans la compression d'images et des fichiers. Afin d'arriver à une meilleure qualité des images des méthodes et des mesures sont introduite pour favoriser des méthodes de compression par rapport aux autres selon le domaine d'utilisation de ces dernières.





---

## Perspective

L'avance de recherche permet de proposer des nouvelles méthodes de compression avec perte plus efficace que la méthode de JPEG, D'où une nouvelle génération de norme JPEG. Ce domaine est très vaste et aussi reste toujours ouvert pour l'amélioration.

Les images animées ont fourni un second défi scientifique pour la compression dont l'apparition d'autres méthodes de compression comme MJPEG<sup>1</sup> à MJPE21, C'est la seconde révolution .

## *Dédicaces*

*Je dédie aussi ce travail à :*

*La mémoire de mon père qu'Allah lui accorde ses miséricordes et son vaste paradis,*

*Ma très chère mère qu'Allah la protège,*

*Ma femme et mes enfants Mohamed EL Mahdi et Maria source de bonheur d'espoir et de  
motivation,*

*Mes frère et sœurs,*

*Mes beaux-parents, beaux-frères et belles-sœurs,*

*Tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès,*

*Mes enseignants,*

*Mes collègues,*

## *Remerciements*

*Avant tout, je remercie le bon dieu, le tout puissant de m'avoir donné la force nécessaire pour mener à terme ce travail.*

*Je tiens à exprimer mes vifs remerciements à mon encadreur Mr OUAFI ABDELKRIM pour son encadrement de qualité et pour les considérables conseils qu'il a fournis afin de m'aider.*

*Je tiens d'abord à remercier Monsieur **BAARIR ZINEDDINE** d'avoir accepté la présidence de mon jury de mémoire, ainsi que le membre de jurée: madame **BARKT AICHA** qui a accepté de m'honorer en acceptant d'examiner, de juger et d'évaluer mon mémoire de fin d'études pour l'obtention du master réseaux et télécommunication*

*Je tiens également à remercier tous mes enseignants*

*En particulier, Mes sincères remerciements s'adressent à mes amis **TEBERMACINE BADRE EDDINE, KHEL RIDHA, RAHMAANI NACER ET BENZAID MONCEF** pour leur aide et soutien et leurs encouragements.*

*Je remercie toute ma famille et mes amis, ainsi que tous ceux qui pourraient se sentir concernés par des remerciements.*



## Résumé

La compression numérique fait beaucoup parler d'elle de nos jours, pour comprendre étape par étape comment s'effectue la réduction du débit d'image numérique fixe.

C'est dans ce cadre de standard de compression JPEG notre mémoire va aborder la description des principales phases de la compression JPEG avec perte tel que la transformation en cosinus discret (DCT), la quantification, le codage entropique, après on discutera le résultat obtenues par teste sous MATLAB sur des différents types d'images.

**Mots clés :** compression d'images monochromes, DCT, quantification scalaire, taux de compression, PSNR, bpp.

## Abstract

Digital compression is getting a lot of talk nowadays, to understand step by step how to reduce the fixed digital image rate.

It is in the theme of JPEG compression standard our memory will address the description of the main principal phases of lossy JPEG compression such as discrete cosine transformation (DCT), quantization, entropy coding, after we discuss the result obtained by test under MATLAB on different types of images.

**Key words:** monochrome image compression, DCT, scalar quantization, compression ratio, PSNR, bpp.

## ملخص

يحصل الضغط الرقمي على الكثير من الحديث في الوقت الحاضر، لفهم كيفية تقليل معدل الصورة الرقمية الثابت خطوة بخطوة. في هذا الإطار من معيار ضغط JPEG ، سوف نتطرق في مذكرتنا لوصف المراحل الرئيسية لضغط JPEG المفقود مثل تحويل جيب التمام المنفصل (DCT) ، القياس الكمي ، ترميز الإنتروبيا ، بعد مناقشة النتيجة التي تم الحصول عليها عن طريق اختبار تحت MATLAB على أنواع مختلفة من الصور.

**الكلمات الأساسية:** ضغط الصورة أحادية اللون، DCT ، القياس الكمي ، نسبة الضغط ، PSNR ، .bpp.

# TABLE DES MATIERES

## Table DES FIGURES

## Liste des tableau

<b>INTRODUCTION GENERALE</b> .....	<b>1</b>
------------------------------------	----------

---

## CHAPITRE I

### GENERALITE SUR LA COMPRESSION D'IMAGE

---

I.1 INTRODUCTION .....	3
I.2 DEFINITION DE LA COMPRESSION D'IMAGE.....	3
I.3 LES DIFFERENTS TYPES DE COMPRESSION.....	4
I.3.1 Compression physique et logique.....	4
I.3.1.1 La compression physique.....	4
I.3.1.2 La compression logique .....	4
I.3.2 Compression avec perte et sans .....	4
I.3.2.1 Méthodes réversible ou sans pertes.....	4
I.2.1.1 Méthodes différentielles ou prédictives.....	5
I.3.2.2 Méthodes par plages .....	5
I.3.2.3 Le codage SHANNON-FANO.....	6
I.3.2.4 Le codage HUFFMAN.....	6
I.3.2.5 Le codage arithmétique.....	8
I.3.2.2 Les méthodes de compression avec pertes ou irréversible .....	9
I.3.2.2.1 Codage par transformée.....	10
I.3.2.2.1.1 Transformée de fourrier.....	12
I.3.2.2.1.2 Transformation en cosinus discrète CT.....	12
I.3.2.2.2 La compression par ondelettes.....	13

I.3.2.2.3 Quantification.....	15
I.3.2.2.3.1 Quantification scalaire.....	16
I.3.2.2.3.2 Quantification vectorielle.....	17
I.4 COCLUSION .....	20

---

## C H A P I T R E II

### S T A N A R D D E C O M P R E S S I O N D ' I M A G E J P E G

---

II.1 INTRODUCTION.....	20
II.2 LA COMPRESSION SANS PERTE JPEG-LS.....	21
II.3 LA COMPRESSION AVEC PERTES DU JPEG .....	22
II.3.1 Description du standard de compression jpeg avec perte.....	23
II.3.1.1 Transformation DCT : .....	23
II.4 LA QUANTIFICATION.....	28
II.5 CODAGE ENTROPIQUE .....	30
II.5.1 Le codage RLE .....	32
II.5.2 Algorithme de Huffman.....	32
II.6 DECOMPRESSION D'UNE IMAGE FIXE DE FORMAT JPEG .....	35
II.6.1 Schéma synoptique .....	35
II.7 CRITERES D'EVALUATION DE LA COMPRESSION D'IMAGE .....	36
II.7.1 critère subjectif .....	36
II.7.2 critère objectif .....	37
II.7.2.1 Erreur quadratique moyenne.....	37
II.7.2.2 Rapport signal à bruit en pic.....	38
II.7.2.3 Taux de compression .....	38
II.7.2.4 Temps de calcul.....	39



II.8 AVANTAGES ET DEFAUTS DU JPEG.....	40
II.1 Les avantages du Jpeg.....	40
II.2 Les défauts.....	40
II.9 CONCLUSION .....	41

**CHAPITRE III**  
**RESULTATS ET SIMULATION**

---

III.1 INTRODUCTION.....	42
III.2 RESULTATS OBTENUS.....	42
III.2.1 Image de teste utilisé .....	42
III.2.2 Interprétation des résultats.....	42
III.2.3 Autres exemples de compression JPEG : .....	48
III.2.4 Relation entre le PSNR et le taux de compression .....	53
III.3 CONCLUSION.....	54
III. CONCLUSION GENERALE.....	55