



République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur Et de La Recherche Scientifique

Faculté des Sciences et Technologies  
Département des Sciences et Technologie

N° d'ordre :  
N° de série :

**Mémoire présenté en vue de l'obtention du diplôme de**

## **MASTER**

**Domaine:** *Sciences et Technologies*

**Filière :** Automatique

**Spécialité :** Automatique

**Par:** MUSTAFA Abdelfattah  
BEN SAHA Abdelkader

## **Thème**

Étude et Simulation de L'Algorithme de Compression d'Images  
JPEG

**Soutenu publiquement le : 19/06/2018**

**Devant le jury :**

<b>Mr. ARIF Mouhamed</b>	MAA	Univ. Ghardaïa	<b>Président</b>
<b>Mr. BENAOUICHA Karim</b>	MAA	Univ. Ghardaïa	<b>Examineur</b>
<b>Mr. KHLIFI Rezki</b>	MAB	Univ. Ghardaïa	<b>Examineur</b>
<b>Mr. BITEUR Kada</b>	MAA	Univ. Ghardaïa	<b>Encadreur</b>

**Année universitaire 2017/2018**

## **Remerciements**

Avant tout, Nous remercier **ALLAH**, le tout puissant,  
de notre avoir donné le courage et la volonté pour  
accomplir ce modeste travail

Nous tenons à remercier nos encadreurs **M.Biteur  
Kada** d'avoir posé un sujet très intéressant et pour leurs  
conseils, leur disponibilité et pour nous avoir fait  
profiter de leurs connaissances et de leurs qualités tant  
professionnelles qu'humaines.

Nous respectueux remerciements à l'ensemble des  
**membres du jury** pour l'intérêt porté sur notre travail.

## DEDICACE

Avant tous, je remercie dieu le tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.

Je dédie ce modeste travail :  
A mes très chers parents, que dieu les garde et les protège pour leurs soutien moral et financier, pour leurs encouragements et les sacrifices qu'ils ont endurés.

A mes frères

A mes sœurs

A mes grandes familles

Aux chers amis

A tous ce que j'aime et qui m'aiment

A tous mes collègues d'études surtout ceux D'automatique

Abdelfattah

## Dédicace

بسم الله الرحمن الرحيم

{رب أوزعني أن اشكر نعمتك التي أنعمت علي وعلى والدي وان اعمل صالحا

ترضاه وأدخلني برحمتك في عبادك الصالحين }

Je dédie ce modeste travail

A ma mère avec toute mon affection.

A mon père avec toute ma reconnaissance.

A mon grand-père et ma grand-mère que dieu les gardes.

A mes frères et mes sœurs.

A tous mes amis

*Abdelkader*

## **Résumé :**

Les images numérisées posent, par leur volume, des problèmes spécifiques de stockage, de traitement et de transmission. IL faut diminuer le volume des fichiers, autrement dit compresser les données.

L'objectif de la compression de données est la réduction du nombre de bits utilisés pour stocker et transmettre les informations comme le texte, le son et l'image. Il existe, en effet, une quantité 'algorithmes de compactage plus ou moins performants. De nouvelles normes et de nouveaux algorithmes apparaissent. Ils permettront de répondre aux besoins des nouvelles applications dites multimédias.

Le travail de ce mémoire de fin d'étude est consacré à l'étude d'une chaîne de compression de la norme JPEG et différentes simulations ont été faites à l'aide de logiciel MATLAB, et une discussion des résultats obtenus.

## **Mots clés :**

Compression d'images, JPEG, MATLAB, Simulation, algorithmes.

## ملخص :

الصور الممسوحة ضوئيا تطرح مشاكل تخزين ومعالجة ونقل محددة حسب حجمها , فمن الضروري تقليل حجمها وبعبارة أخرى ضغط بياناتها , الهدف من ضغط البيانات هو تقليل عدد البيئات المستخدمة لتخزين ونقل المعلومات مثل النص الصوت والصورة , يوجد في الواقع مجموعة من خوارزميات الضغط أكثر أو أقل أداء معاير وخوارزميات جديدة تظهر سوف تلبي احتياجات التطبيقات الجديدة تسمى الوسائط المتعددة.

هذه المذكرة تعمل على دراسة سلسلة الضغط معيار JPEG وتم إجراء عمليات محاكاة مختلفة باستخدام برنامج MATLAB ومناقشة مختلف النتائج التي تم الحصول عليها.

## الكلمات المفتاحية :

ضغط الصورة , JPEG, MATLAB, محاكاة, خوارزميات

## **Abstract**

Scanned images pose, by their volume, specific problems of storage, processing and transmission. It is necessary to reduce the volume of the files, in other words to compress the data.

The objective of the data compression is the reduction of the number of bits. used to store and transmit information such as text, sound and image. There is, indeed, a number of compaction algorithms more or less effective. New standards and new algorithms appear. They will meet the needs new applications called multimedia.

The work of this final thesis is devoted to the study of a compression chain of the JPEG standard and different simulations were made using MATLAB software, and a discussion of the results.

### **Key words :**

image compression ,JPEG, MATLAB, Simulation, algorithms

---

# LISTE DES FIGURES

Figure.I.1: L'image naturelle et l'Image artificielle. ....	3
Figure.I.2 Image binaire ( noire et blanc) .....	4
Figure.I.3: Image à niveaux de gris (intensité ou luminance).....	5
Figure.I.4: quantification des niveaux de gris .....	5
Figure.I.5: Image couleur RGB.....	6
Figure.I.6: La résolution de l'image .....	6
Figure.I.7: groupe de pixel formant la lettre .....	7
Figure.I.8: la définition d'une image.....	8
Figure.I.9: Tracé de l'histogramme .....	10
Figure.I.10: connexité et voisinage .....	12
Figure.I.11: Traitements fondamentaux en traitement d'images .....	12
Figure.I.12: Approche région.....	14
Figure.I.13: Approche contour.....	15
Figure.I.14: Approche classification.....	15
Figure.II.1: Schéma de la compression symétrique .....	18
Figure.II.2: Schéma synoptique de compression .....	19
Figure.III.1: Principe général de la compression/décompression JPEG.....	30
Figure.III.2: Perception de l'œil humain.....	31
Figure.III.3: Répartition RGB – YUV .....	31
Figure.III.4: sous-échantillonnages 4:4:4 .....	32
Figure.III.5: sous-échantillonnages 4:2:2 .....	33
Figure.III.6: sous-échantillonnages 4:2:0 .....	33
Figure.III.7: La transformé DCT et IDCT .....	34
Figure.III.8: La transformation suivant les colonnes et les lignes .....	36
Figure.III.9: les différentes 64 sous-images produites par la DCT.....	36

---

---

Figure.III.10: Codage différentiel du coefficient DC .....	40
Figure.III.11: Lecteur en zig-zig .....	42
Figure.III.12: Fractale du type Julia .....	47
Figure.III.13: Représentation d'une sinusoïde et d'une ondelette .....	49
Figure.III.14: Exemple de fonctions ondelette et leur somme .....	49
Figure.VI.1: Page d'accueil de l'application .....	52
Figure.VI.2: La fenêtre d'édition d'interface .....	52
Figure.VI.3: Page de garde.....	53
Figure.VI.4: Le menu « image».....	54
Figure.VI.5: Le menu « caractéristiques» .....	54
Figure.VI.6: prétraitement.....	56
Figure.VI.7: Compression d'une image Niveaux de gris avec un facteur de qualité (25%) .....	62
Figure.VI.8: Compression d'une image Niveaux de gris avec un facteur de qualité (50%) .....	63
Figure.VI.9: Compression d'une image Niveaux de gris avec un facteur de qualité (75%) .....	63
Figure.VI.10: Compression d'une image Niveaux de gris avec un facteur de qualité (100%) ....	64
Figure.VI.11: Compression d'une image couleur RGB avec un facteur de qualité (25%) .....	65
Figure.VI.12: Compression d'une image couleur RGB avec un facteur de qualité (50%) .....	65
Figure.VI.13: Compression d'une image couleur RGB avec un facteur de qualité (75%) .....	66
Figure.VI.14: Compression d'une image couleur RGB avec un facteur de qualité (100%) .....	66
Figure.VI.15: Compression d'une image Couleur YCbCr avec un facteur de qualité (25%).....	67
Figure.VI.16: Compression d'une image Couleur YCbCr avec un facteur de qualité (50%).....	68
Figure.VI.17: Compression d'une image Couleur YCbCr avec un facteur de qualité (75%).....	68
Figure.VI.18: Compression d'une image Couleur YCbCr avec un facteur de qualité (100%)....	69

---

---

# LISTE DES TABLEAUX

Tableau I.1: Table Histogramme..	9
Tableau III.1 : Catégories d'amplitudes des différences DC.	41
Tableau III.2 : Catégories d'amplitudes des coefficients AC.	43
Tableau III.3: Codes de HUFFMAN des coefficients Diff DC.	44
Tableau III.4: Codes de HUFFMAN des coefficients AC.	45
Tableau III.5: Exemple de compression fractale.....	48
Tableau VI.2: La taille des symboles AC	61
Tableau VI.3: Les résultats de la compression d'une image Niveaux de gris en fonction du facteur de qualité	62
Tableau VI.4: Les résultats de la compression d'une image couleur RGB en fonction du facteur de qualité	64
Tableau VI.5: Les résultats de la compression d'une image couleur YCbCr en fonction du facteur de qualité.....	67

---

---

# LISTE DES ABREVIATION

<b>JPEG</b>	joint photographique experts group
<b>DPCM</b>	Différentiel Pulse Code Modulation
<b>DCT</b>	Discrete Cosine Transform
<b>IDCT</b>	Discrete Cosine Transform Inverse
<b>EQM</b>	Erreur quadratique moyenne
<b>SNR</b>	Signal to Noise Ratio (db)
<b>PSNR</b>	Peak Signal to Noise Ratio
<b>dpi</b>	Dot Per Inch
<b>RGB</b>	Les couleurs : rouge, vert et bleu
<b><math>h(v)</math></b>	présente le nombre de fois répété de valeur du pixel dans l'image
<b>C.C.D.</b>	Charge Coupled Device).
<b><math>T_b</math></b>	Taux de bits en bits/pixel (bpp)
<b>YUV</b>	Luminance, chrominance bleue, chrominance rouge
<b>Q</b>	La matrice de quantification
<b><math>F^*(u, v)</math></b>	La matrice quantifiée
<b><math>F(u, v)</math></b>	La matrice des coefficients à quantifier
<b><math>Q(u, v)</math></b>	La table de quantification
<b><math>F_q</math></b>	Le facteur de qualité de la matrice de quantification
<b>RLE</b>	Run length Encoding
<b>EOB</b>	End Of Bloc

---

---

# TABLE DE MATIERE

INTRODUCTION GENERALE.....	1
Chapitre I : Généralité sur le traitement d'image	
I.1 Introduction.....	3
I.2 Définition de l'image .....	3
I.2.1 D'un point de vue humain .....	3
I.2.2 D'un point de vue mathématique .....	3
I.3 Types d'images .....	4
I.3.1 Image binaire.....	4
I.3.2 Image à niveaux de gris (intensité ou luminance) .....	4
I.3.3 Image couleur (RGB).....	5
I.4 Résolution .....	6
I.5 Caractéristiques de l'image.....	7
I.5.1 Le pixel .....	7
I.5.2 La Définition.....	7
I.5.3 Le poids numérique.....	8
I.5.4 Transparence .....	9
I.5.5 Histogramme de l'image .....	9
I.5.6 La luminance .....	10
I.5.7 La texture .....	11
I.5.8 Contraste .....	11
I.5.9 Homogénéité.....	11
I.5.10 Voisinage d'un objet et connexité .....	11
I.5.1.1 Région .....	12
I.6 Etapes fondamentales en traitement d'images.....	12

---

I.6.1 Acquisition de l'image .....	13
I.6.2 Le rehaussement de l'image.....	13
I.6.3 Les ondelettes .....	14
I.6.4 La morphologie .....	14
I.6.5 La segmentation .....	14
I.6.5.1 Approche région.....	14
I.6.5.2 Approche contour .....	15
I.6.5.3 Approche classification.....	15
I.6.6 La représentation et description .....	16
I.6.7 La reconnaissance .....	16
I.6.8 La restauration d'image .....	16
I.7 Domaines d'application .....	16
I.8 Conclusion .....	17
 Chapitre II : Généralité sur la compression d'images	
II.1 Introduction.....	18
II.2 Définition Compression d'images .....	18
II.3 Compression symétrique et asymétrique .....	18
II.3.1 La Compression symétrique .....	18
II.3.2 La Compression asymétrique .....	19
II.4 Principe général de la Compression des images.....	19
II.5 Classification des méthodes de compression .....	20
II.5.1 Méthodes de compression sans perte d'informations.....	20
II.5.1.1 Méthode de codage RLE (Run-length encoding) .....	20
II.5.1.2 Principe de fonctionnement .....	20
II.5.1.3 Codage Huffman.....	21
II.5.1.4 Codage arithmétique.....	24

---

---

II.5.1.5 Description de la méthode .....	24
II.5.1.6 Compression .....	24
II.5.1.7 Décompression .....	26
II.5.2 Méthodes de compression avec perte d'informations .....	26
II.6 Evaluation de la compression .....	27
II.7 Qualité de reconstruction des images .....	27
II.8 Conclusion .....	28
<b>Chapitre III : La norme JPEG</b>	
III.1 Introduction .....	29
III.2 Fonctionnement général du JPEG .....	29
III.3 Transformation des couleurs .....	30
III.4 Sous- échantillonnage de la chrominance (Cb et Cr).....	32
III.5 Le découpage en bloc de pixels.....	33
III.6 Transformation DCT .....	34
III.7 La Quantification.....	37
III.8 Le codage entropique .....	40
III.8.1 Codage intermédiaire .....	40
III.8.1.1 Codage intermédiaire des éléments DC .....	40
III.8.1.2 Codage intermédiaire des éléments AC .....	42
III.8.2 Codage HUFFMAN .....	44
III.8.2.1 Codage d'HUFFMAN des coefficients Diff DC .....	44
III.8.2.2 Codage HUFFMAN des coefficients AC .....	45
III.9 La décompression.....	46
III.10 Autres méthodes de compression.....	47
III.10.1 La compression fractale .....	47

---

---

III.10.2 La compression par ondelettes .....	48
III.10 Conclusion.....	50
Chapitre VI : Application et implémentation	
VI.1 Introduction .....	51
VI.2 Langage de programmation utilisé.....	51
VI.2.1 Interface Graphique Utilisateur (GUI).....	51
VI.2.2 Présentation générale de GUIDE .....	52
VI.3 Présentation de l' Application .....	53
VI.3.1 Les menus.....	53
VI.3.1.1 Le menu « image».....	53
VI.3.1.2 Le menu « caractéristiques».....	54
VI.4 L'implémentation de l'algorithme JPEG.....	54
VI.4.1 La compression .....	55
VI.4.1.1 Le prétraitement .....	55
VI.4.1.2 La DCT.....	56
VI.4.1.3 Résultats après la DCT.....	57
VI.4.1.4 La quantification .....	58
VI.4.1.5 La compression conservative .....	59
VI.4.1.6 La séquence zigzag .....	59
VI.4.1.7 Le codage RLE .....	59
VI.4.1.8 Codage de Huffman .....	60
VI.5 Simulation .....	62
VI.6 Discussion les résultats .....	69
VI.7 Conclusion.....	69
Conclusion Général .....	70
Bibliographie	

---

**INTRODUCTION**

**GENERALE**

## INTRODUCTION GENERALE

Les évolutions récentes des technologies de l'information et systèmes de communication, ainsi que la diversification des applications et fonctionnalités du multimédia dans plusieurs domaines (télémédecine, télédétection... etc.) impliquent la manipulation de données très volumineuses. Parmi les données les plus volumineuses, on compte les images et vidéos numériques, qui ont de ce fait un besoin particulièrement important d'un traitement adapté à leurs spécificités. Les méthodes de compression de ce type de données peuvent être classées suivant la nécessité de récupération parfaite ou non de l'information originale en deux grandes catégories : La compression sans pertes d'information ou réversible qui a l'avantage de préserver la qualité de l'image originale, mais avec un taux de compression relativement faible et la compression avec pertes qui regroupe des algorithmes caractérisés par leur taux de compression assez élevé tout en gardant le mieux possible une acceptable qualité de l'image originale. Elle est basée généralement sur une phase de transformation qui sert à compacter l'information utile dans un nombre minimum de coefficients non nuls.

Les techniques de compression avec pertes se décomposent généralement en deux ou trois phases : une phase de décorrélation du signal originale (élimination d'une part de l'information redondante), éventuellement une phase de quantification (élimination d'une part d'information non redondante mais jugée non indispensable) et enfin une phase de compression proprement dite (application d'un codage entropique sur le signal décorrélé et quantifié).

Après une longue période de recherche et de développement, avec la participation d'une très grande communauté scientifique, le JPEG (joint photographique experts group) est devenu un standard international en 1992.

La norme JPEG est actuellement très utilisée pour coder les images numériques que l'on trouve dans notre vie quotidienne (internet, appareils photos numériques, ... etc.). Elle est bien adaptée pour les images naturelles et pour un bon taux de compression. Au-delà, des artefacts (effet des blocs) vont apparaître et seront très gênants visuellement.

Notre mémoire s'articule autour de quatre chapitres :

Dans un premier chapitre, nous abordons les concepts de base de traitement d'images.

Le chapitre 2 traite les généralités sur la compression ainsi que les techniques de compression conservatrices les plus répandues.

Le chapitre 3 explique la norme JPEG (Joint Photographic Experts Groupe) recommandée par le CCITT et l'ISO.

Le chapitre 4 est une présentation de notre application de la compression JPEG et une discussion des résultats obtenus.

**CHAPITRE I :**  
**GENERALITE SUR LE**  
**TRAITEMENT**  
**D'IMAGE**

## I.1 Introduction

Le traitement d'images numériques est un ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations. de ce fait, le 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.

Dans ce chapitre, nous aborderons les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différentes techniques connues dans ce domaine. [3]

## I.2 Définition de l'image

### I.2.1 D'un point de vue humain

L'image est une représentation d'une personne ou d'un objet par la peinture, le dessin, la photographie, le film, etc..... c'est aussi un ensemble structuré d'informations qui, après l'affichage sur l'écran, ont une signification pour l'œil humain. [4]

### I.2.2 D'un point de vue mathématique : [1]

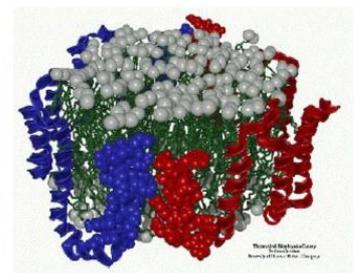
- L'image est une nombre de matrice représentant un signal
  - Plusieurs outils permettent de manipuler ce signal
- **Image naturelle** : plusieurs moyens d'acquisition (caméra, microscope, tomographie, infra- rouge, satellite)
- **Image artificielle** : plusieurs outils de représentation (synthèse d'images, réalité virtuelle, visualisation scientifique, ... etc).



Image naturelle



Image artificielle



**Figure I.1** : L'image naturelle et l'Image artificielle

### I.3 Types d'images

On distingue trois types d'images :

- Binaire : 2 couleurs (arrière-plan et avant-plan).
- Images à niveaux de gris (Monochromes) : variations d'une même teinte.
- Images en couleurs (Polychromes): " vraies " couleurs. [7]

#### I.2.1 Image binaire

Dans une image binaire (Bichromies), les pixels sont représentés par deux états logiques 0 et 1.

Le 0 correspond à un pixel noir et 1 à un pixel blanc, Le niveau de gris est donc codé sur un seul bit (0,1).

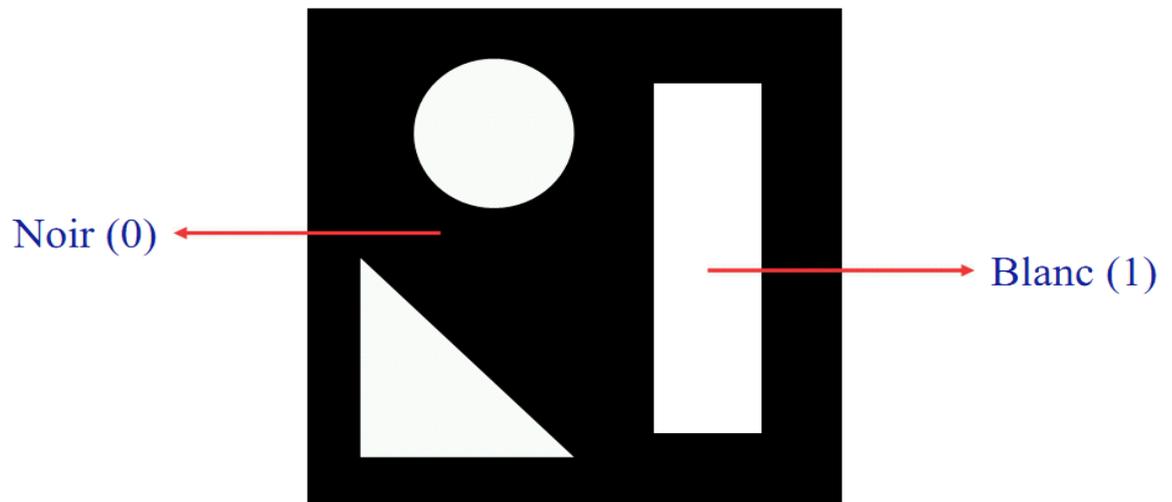
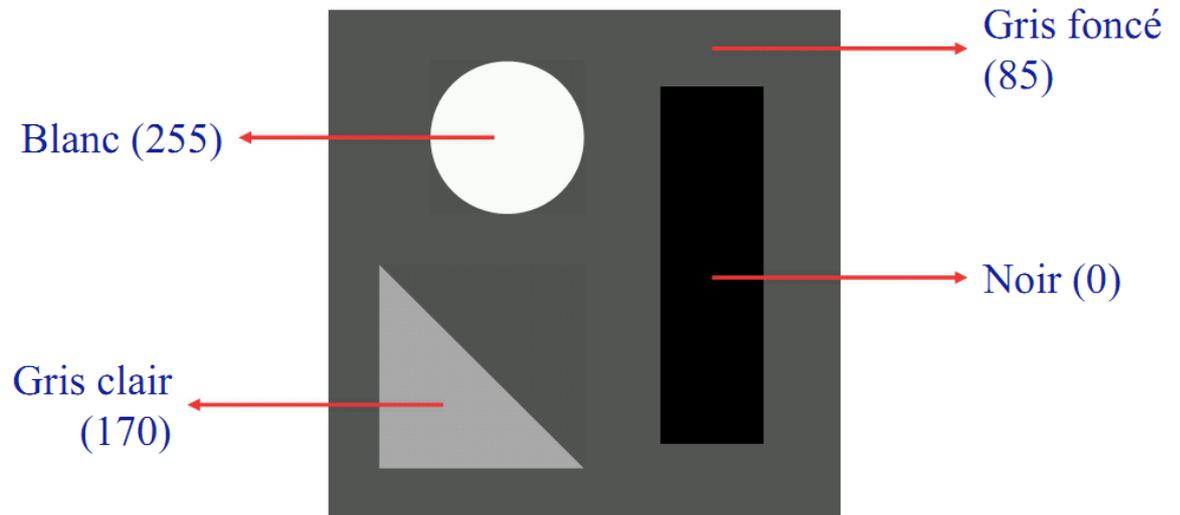


Figure I-2 : Image binaire ( noire et blanc) [5]

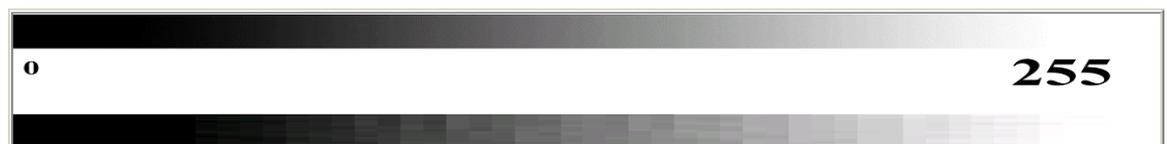
#### I.2.2 Image à niveaux de gris (intensité ou luminance)

Le niveau de gris est la valeur de l'intensité lumineuse en un point, la couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires, donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255, chaque pixel n'est donc plus représenté par un bit, mais par un octet , pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

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 . [3]



**Figure I-3 :** Image à niveaux de gris (intensité ou luminance) [5]



**Figure I-4:** quantification des niveaux de gris

### I.2.3 Image couleur (RGB)

L'image couleur est constituée à trois couleurs rouge, vert et bleu (RVB en anglais RGB), chaque couleur est codée comme une image à niveaux de gris, avec des valeurs allant de 0 à 255, pour  $R=V=B=0$  nous aurons un noir pur, et pour  $R=V=B=255$  nous aurons un blanc pur, la représentation des images couleurs se fait donc soit par une image dont la valeur du pixel est une combinaison linéaire des valeurs des trois composantes couleurs, soit par trois images distinctes représentant chacune une composante couleur, on distingue généralement deux types d'images: [7]

- Images 24 bits ;
- Images à palettes ;

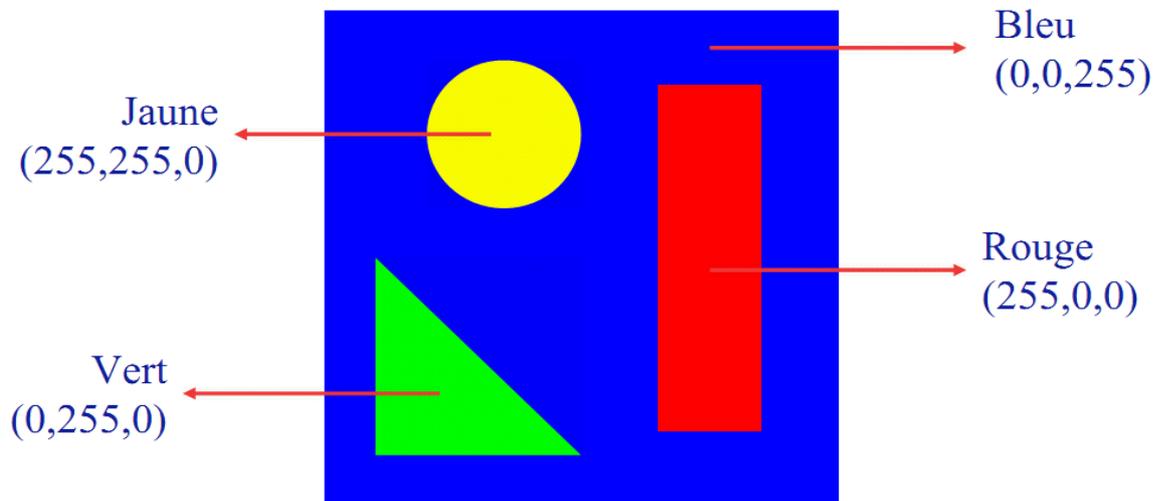


Figure I-5 : Image couleur RGB [5]

## I.4 Résolution

La résolution d'une image est définie par le nombre de pixels par unité de longueur. Usuellement, on compte le nombre de pixels par pouce (1 pouce = 2,54 cm, noté ppp ou dpi) ou par centimètre. (Dot Per Inch ou Point Par Pouce )

Plus le nombre de pixels par unité de longueur est élevé, plus la quantité d'information décrivant l'objet est importante donc la résolution est grande. Son corolaire est une taille de fichier de plus importante.

Ce paramètre est défini souvent lors de l'acquisition de l'image (réglage de l'appareil photo, résolution du logiciel du scanner...) ou ultérieurement dans les logiciels de traitement d'image.

La publication d'image sur Internet correspond souvent à une résolution de 90 ppp et dans la presse écrite de 150 ppp. [6]

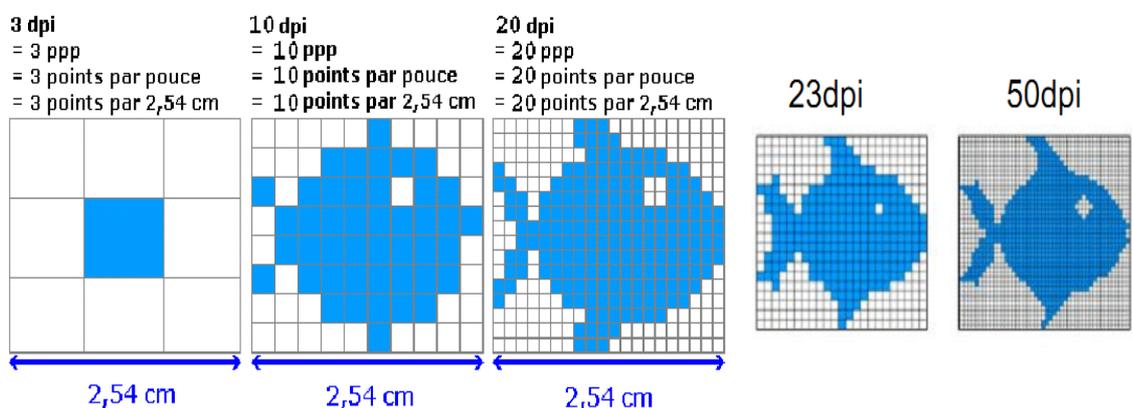


Figure I-6 : La résolution de l'image [2]

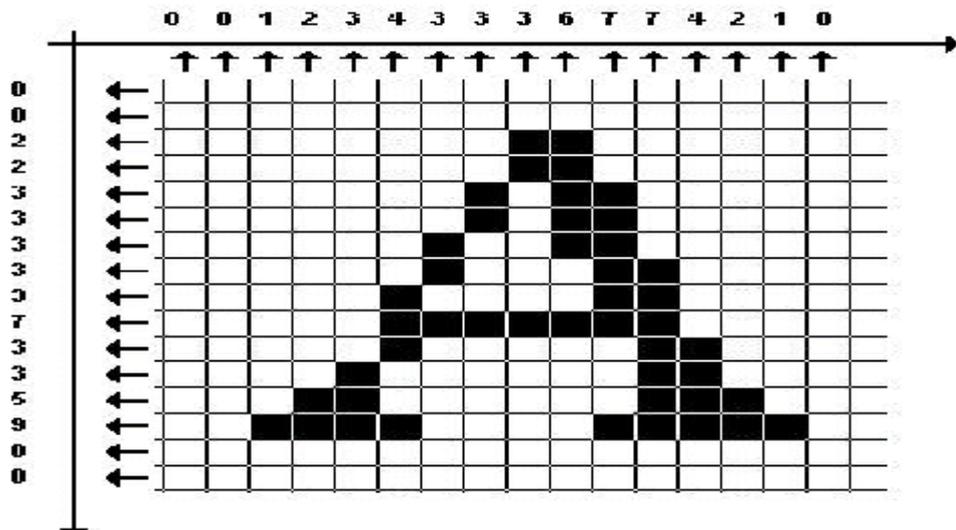
## I.5 Caractéristiques de l'image

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

### I.5.1 Le pixel:

Une image numérique est constituée d'un ensemble de points appelés pixels, abréviation de (PICTure Element) pour former une image. Le pixel représente ainsi le plus petit élément constitutif d'une image numérique, l'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image. [8]

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



**Figure I-7 :** groupe de pixel formant la lettre A,[7]

Pour les images 3D le «pixel» est alors appelé un voxel, et représente un volume élémentaire.

### I.5.2 La Définition:

On appelle définition le nombre de points (pixels) constituant une image: c'est le nombre de colonnes de l'image que multiplie son nombre de lignes. Une image possédant 10 colonnes et 11 lignes aura une définition de 10x11. (lorsqu'on parle de la définition d'un écran, on n'indique pas le résultat.

Pour un périphérique d'acquisition d'image on indique le nombre total des pixels). [2]

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										

**Figure I-8:** la définition d'une image

### I.5.3 Le poids numérique

C'est l'élément capital qui définit la mémoire occupée par le fichier numérique sur le support d'enregistrement.

Il est le produit de la 'Définition multipliée par le nombre Octets (ou Bytes contraction de bit by eight), et s'exprime en OCTETS'.(1 octet = 8 bits = 0 à 255 Niveaux de Densités).[9]

➤ **Exemple :**

Image d'une définition de 2 000 pixels x 3 000 pixels sous 3 Octets (image en couleur RVB en 3 x 8 bits - 24 bits), soit : 2 000 pixels x 3 000 pixels x 3 octet = 18 000 000 Octets ou 18 Mo.

Le poids numérique : **unité**

1 024 octets = 1Ko = 1Kilo-octets

1 048 576 octets = 1Mo= 1Méga-octets

1 048 576 000 octets = 1Go = 1Giga-octets..... Tétra-octet, Péta-octet ...

### I.5.4 Transparence :

La transparence est une caractéristique définissant le niveau d'opacité des éléments de l'image, c'est la possibilité de voir à travers l'image des éléments graphiques situés derrière celle-ci. [8]

### I.5.5 Histogramme de l'image

L'histogramme représente la répartition des pixels en fonction de leur niveau de gris(couleur) , il fournit diverses informations comme les statistiques d'ordre et peut permettre d'isoler des objets.[8]

On veut souvent avoir une information sur la distribution des valeurs des pixels (niveaux) dans une image, pour cela on utilise souvent une table qui donne le nombre de pixels de chaque niveau dans l'image. Cette table, souvent représentée graphiquement, est appelée Histogramme de l'image, est notée  $h(v)$ . [10]

#### ➤ Exemple

Dans cette exemple, on veut représenter l'image de taille  $4 \times 4$  pixels par son Histogramme de l'image et pour cela, on commence par calculer la Table Histogramme de l'image (voir

Tableau (I-1)) , ensuite on trace l'Histogramme de l'image comme montré dans la figure (I-9).

$V$  : présente la valeur du pixel dans l'image.

$h(v)$  : présente le nombre de fois répété de cette valeur du pixel dans l'image .

$v$	1	2	3	4	5	6	7
$h(v)$	4	3	2	2	1	3	1

3	4	5	6
1	1	2	4
3	2	2	7
1	6	1	6

Image

Tableau I-1 : Table Histogramme.

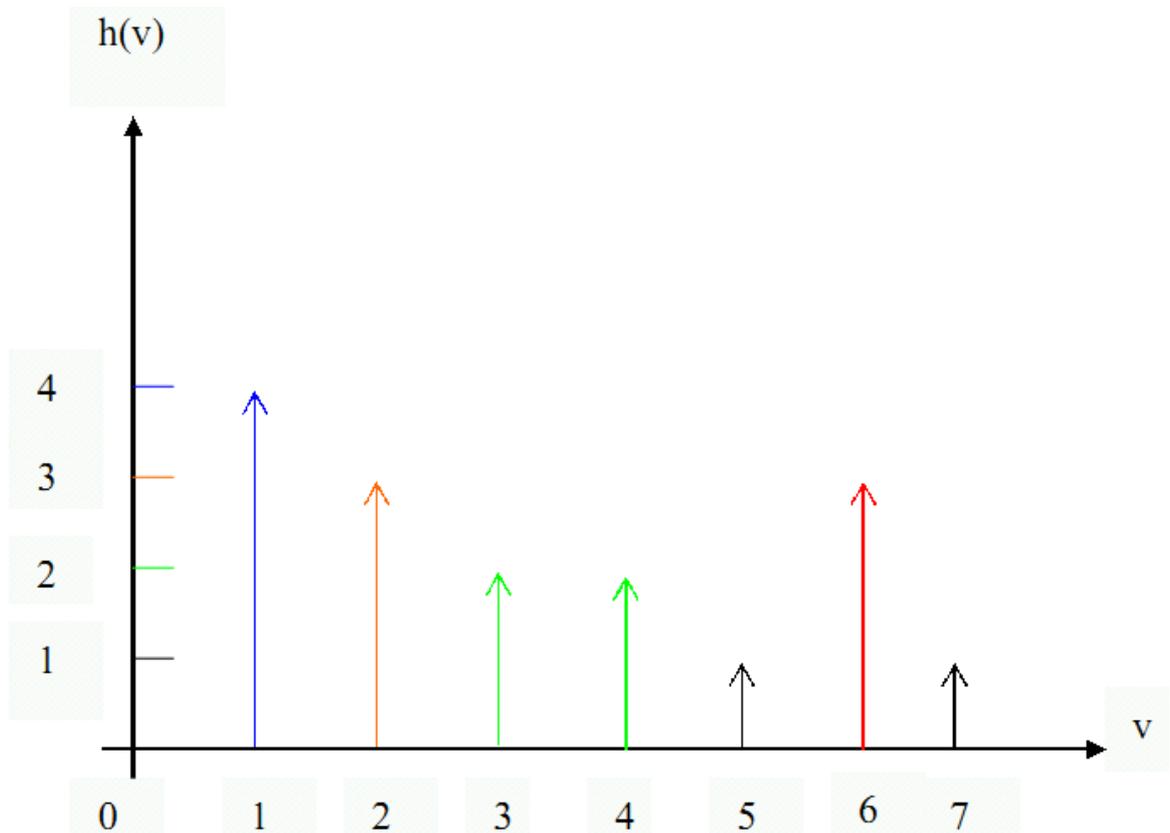


Figure I-9 : Tracé de l'histogramme.

### I.5.6 La luminance : [7]

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes);
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

### I.5.7 La texture

Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont par exemple un motif basique qui se répète. La texture est composée de Texel, l'équivalent des pixels.[7]

### I.5.8 Le contraste :

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

Si  $L_1$  et  $L_2$  sont les degrés de luminosité respectivement de deux zones voisines  $A_1$  et  $A_2$  d'une image, le contraste  $C$  est défini par le rapport : [7]

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

### I.5.9 Homogénéité :

Une région dans une image est dite homogène si elle regroupe un ensemble de pixels qui possèdent des caractéristiques similaires ou uniformes.

L'homogénéité est une information locale et correspond au caractère uniforme d'une région.

Ces caractéristiques peuvent être par exemple la variance du niveau de gris, la couleur, ... [8]

### I.5.10 Voisinage d'un objet et connexité :

On définit le voisinage de tout composant  $A$  de  $I$  par l'union de tous les voisinages des pixels de  $A$ , on peut définir un objet par la relation de voisinage, on dit que deux points  $P$  et  $Q$  sont connectés si il existe un chemin connexe (suite de points  $P=P_0, P_1, \dots, P_{n-1}, P_n=Q$  telle que  $P_i$  est voisin de  $P_{i-1}$ ) entre  $P$  et  $Q$  de points ayant la même valeur. Cette relation est une relation d'équivalence, dont chaque classe définit un objet de l'image.

Le type de voisinage utilisé influe bien sûr sur les objets résultants, des algorithmes très efficaces permettent d'étiqueter les différents objets d'une image.

La notion de connexité dérive de la notion de voisinage, il existe principalement deux ordres de connexité : 4 et 8, ce nombre correspond à la taille du plus petit voisinage non vide d'un pixel. Un pixel a 4 voisins directs si l'on choisit la distance  $d_1$ , il en a 8 avec la distance  $d_{inf}$ . [11]

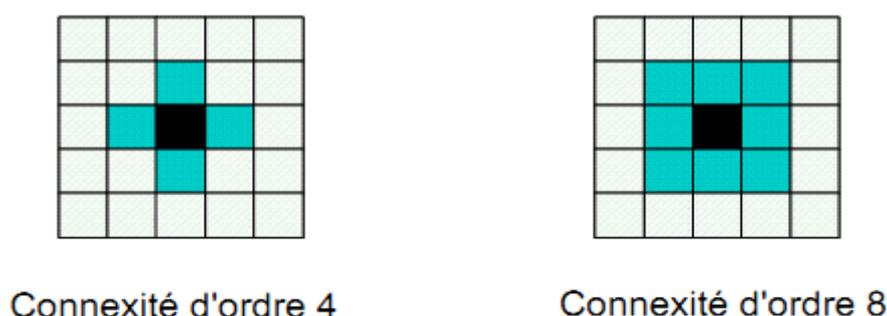


Figure I-10 : connexité et voisinage

### I.5.11 Région :

La région est un ensemble de pixels homogènes et connectés, le pixel n'appartient à la région que s'il contient certaines propriétés (intensité moyenne, centre de gravité,...).

Une région est toujours limitée par un contour.

## I.6 Etapes fondamentales en traitement d'images

Un ensemble de traitements peut être appliquée à l'image numérique, la figure suivante résume l'ensemble de ces traitements.

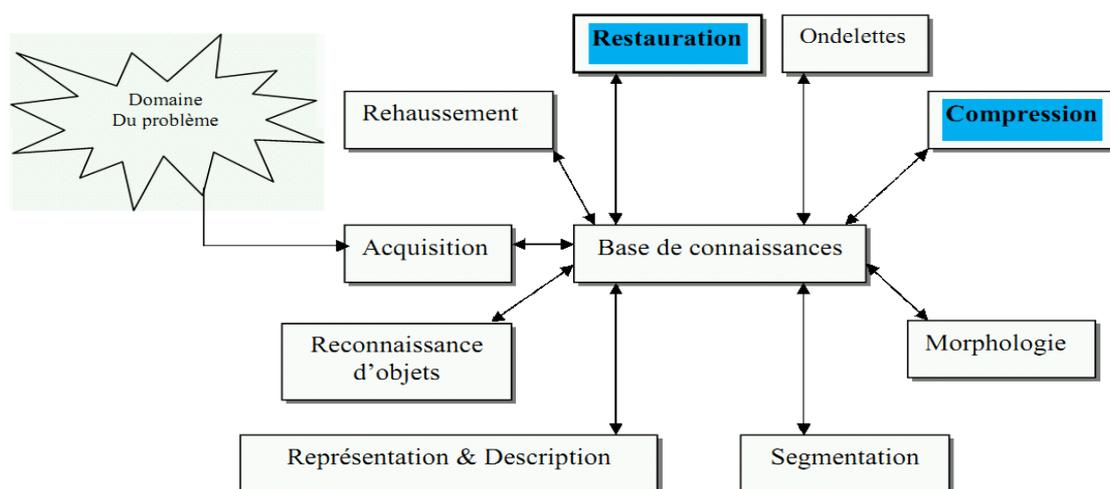


Figure I-11 : Traitements fondamentaux en traitement d'images

### I.6.1 Acquisition de l'image :

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales. [3]

- les caméras numériques,
- et les scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la T.V. ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier [chakib 99]. L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (pixel), mais aussi l'inter corrélation de deux éléments voisins : plus cette intercorrélacion est faible, meilleure est l'image [12].

### I.6.2 Le rehaussement de l'image : [13]

Le mécanisme de formation des images est loin d'être parfait donc présence de différentes formes de bruit, les objectifs du rehaussement d'images est :

#### A. ,, Amélioration du contraste

- Augmenter la plage dynamique
- Isoler et améliorer la perception des caractéristiques

#### B. ,, Filtrage

- Brouiller (analyse multi résolution)
- Accentuer les caractéristiques
- Atténuer le bruit
- Atténuer le bruit impulsionnel
- Filtrer tout en préservant les arêtes

### I.6.3 Les ondelettes :

La transformée en ondelettes permet d'obtenir une représentation locale de l'information contenue dans le signal analysé, en respectant le principe de l'incertitude, alors que la transformée de Fourier fournit une information globale sur le signal, similairement à la transformée de Fourier, la transformée en ondelettes permet de décomposer un signal sur une base de fonctions élémentaires :les ondelettes , à partir d'une fonction de base , appelée ondelette -mère ou ondelette analysante, on construit cette base de fonctions élémentaires par des opérations de dilatation et de translation. [14]

### I.6.4 La morphologie :

La morphologie mathématique a pour but l'étude des objets en fonction de leur forme, de leur taille, des relations topologique avec leur voisinage, de leur texture, et de leurs niveaux de gris.[15]

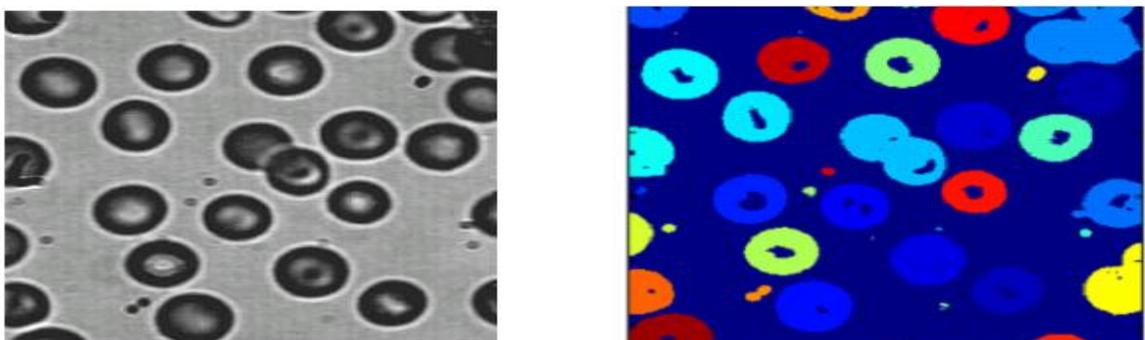
### I.6.5 La segmentation : [15]

La segmentation consiste à découper une image en régions connexes de niveaux de gris, présentant une homogénéité selon un certain critère, une multitude de méthodes de segmentation sont proposées dans la littérature, elles se répartissent en trois grandes familles : la segmentation par approche contour, région et classification.

#### I.6.5.1 Approche région :

Une région est un ensemble de pixels ayant des caractéristiques communes (intensité, texture ... etc.), qui les différencient des pixels des régions voisines.

L'approche région a pour objectif de décomposer l'image en un ensemble de régions connexes les plus homogènes possible, et les plus différentes pour celles qui se côtoient.



(A) Image originale,

(B) Image Segmentée par région.

**Figure I-12** : Approche région

### I.6.5.2 Approche contour :

Un contour est un ensemble de pixels formant une frontière entre deux ou plusieurs régions voisines, où la limite entre deux pixels dont le niveau de gris représente une différence significative.

L'approche contour consiste à identifier les discontinuités qui séparent les différentes régions de l'image, cette approche cherche les di similarités.

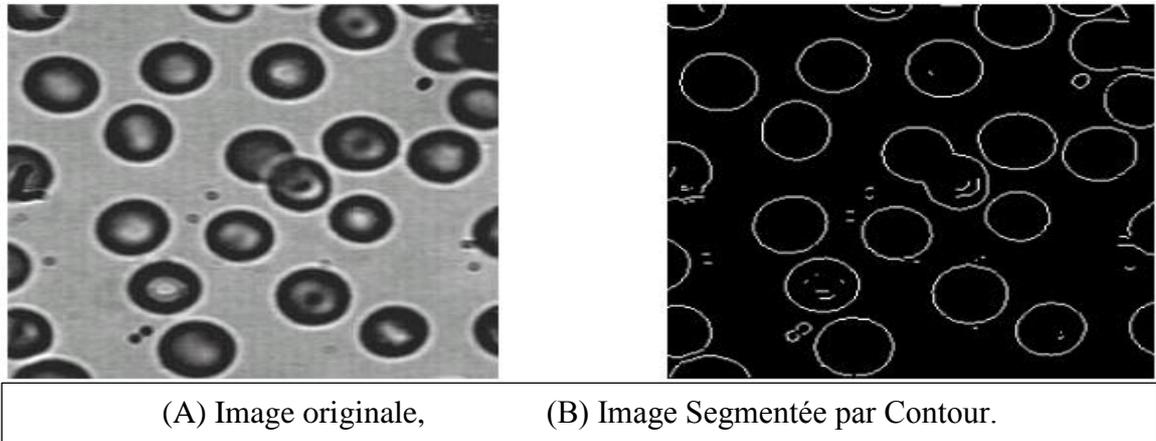


Figure I-13: Approche contour

### I.6.5.3 Approche classification :

L'approche classification cherche à regrouper les pixels qui présentent une similarité et une uniformité selon un critère prédéfini, on parle de partitionnement ou de clustering.

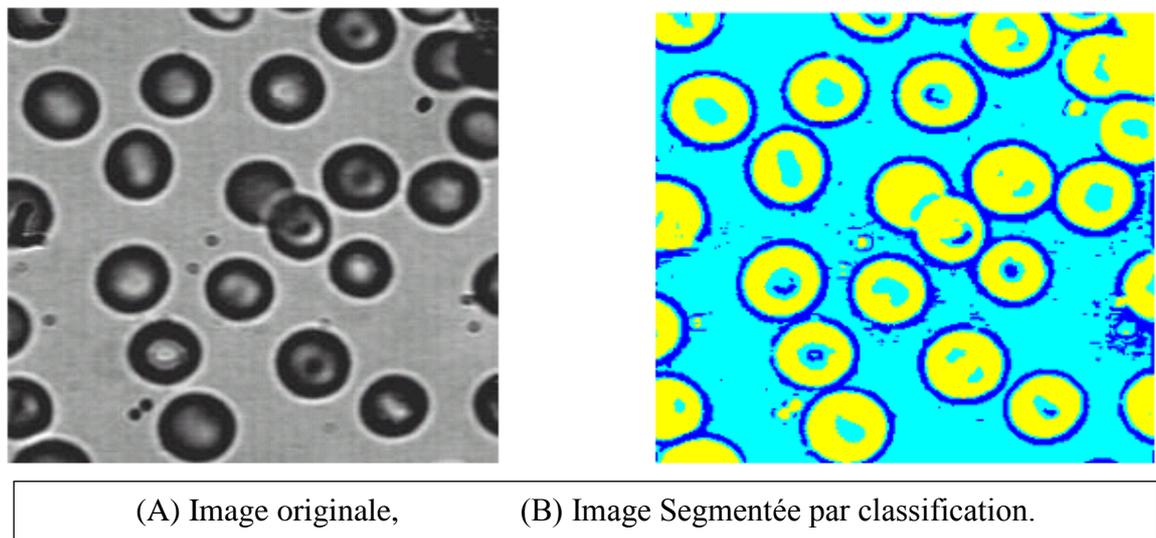


Figure I-14 : Approche classification

### **I.6.6 La représentation et description :**

La représentation est l'étape qui vient juste après la segmentation. Le résultat de la segmentation est un ensemble de pixels relatifs à une région, ces données doivent être converties en une forme traitable par un ordinateur, on peut soit représenter la région ou sa frontière, la description est l'extraction d'attribut permettant de distinguer une classe d'objets d'une autre classe. [8]

### **I.6.7 La reconnaissance :**

La reconnaissance d'objets est une branche de la vision artificielle et un des piliers de la vision industrielle, elle consiste à identifier des formes pré-décrites dans une image numérique, et par extension dans un flux vidéo numérique.

Il ne faut pas confondre reconnaissance d'objets (en anglais : « object recognition » ou « shape recognition ») et reconnaissance de formes (« pattern recognition » en anglais).

La première s'attache à reconnaître des formes géométriques dans une image, alors que la seconde cherche à identifier des motifs dans des données statistiques, la confusion vient du fait qu'on utilise souvent la reconnaissance de formes comme technique appliquée à la reconnaissance d'objets. [12]

### **I.6.8 La restauration d'image**

Les images subissent des dégradations, ces dégradations sont dues, d'une part au appareilles d'acquisition et d'autre part aux conditions de prise de vue... Son but est d'améliorer la qualité d'une image, atténuer, supprimer les dégradations.

La restauration se fait par des technique comme noir et blanc, négative, rotation, etc.

- **Noir et blanc** : L'image noir et blanc, est une image dont les couleurs ont été remplacés par le gris sauf le noir et le blanc de l'image.
- **Négative** : est une image dont les couleurs ont été inversées par rapport à l'originale par exemple le noir devient blancet inversement. [8]

### **I.7 Domaines d'application**

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,...), biologie, astronomie, 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,... [7]

### **I.8 Conclusion :**

Dans ce chapitre, nous avons cité les différents traitements que peut subir une image, dans la suite on va entamer l'objectif primordial de notre mémoire ce qui concerne les méthodes de la compression d'images et leurs principes, ainsi la classification des méthodes de compression.

**CHAPITRE II :  
GENERALITE SUR LA  
COMPRESSION D'IMAGES**

## II.1 Introduction :

La compression des images est une nécessité plus vitale dans le domaine des multimédia, car les images numériques comme nous allons le découvrir, forment un énorme ensemble de données, même lorsqu'il s'agit d'une simple image fixe, d'où leur stockage et leur transmission qui deviennent très pénibles. Pour cela, nous observons la naissance de toute une branche de la science, celle qui traite les images fixe et mobile.[10]

## II.2 Définition Compression d'images :

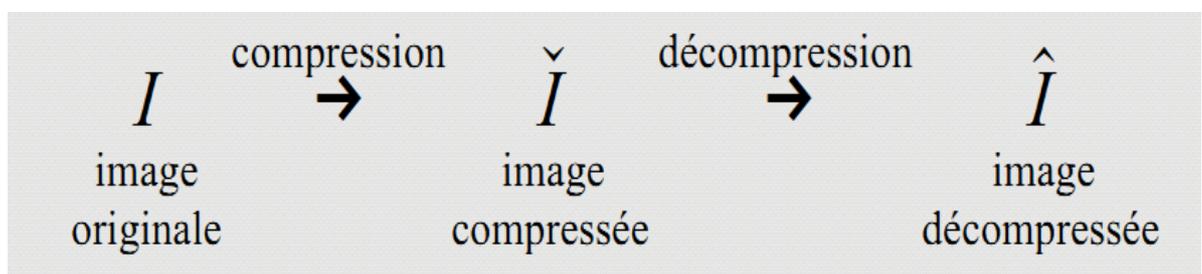
La compression d'image est l'ensemble des méthodes et des algorithmes qui permet de réduire la taille de l'image tout en gardant les informations utiles portées par l'image, l'opération inverse s'appelle la décompression. Le résultat de ces deux transformations enchainées doit produire le même résultat que les données de départ C'est-à-dire qu'une compression suivie d'une décompression ne doit pas changer les données initiales. Cependant pour la compression d'images, il est acceptable de perdre un peu d'information pour obtenir un taux de compression plus élevé. L'image résultante n'est donc pas identique, bien que très proche visuellement. On dit alors que c'est une compression avec perte d'information.[14]

## II.3 Compression symétrique et asymétrique :

### II.3.1 La Compression symétrique :

Si la quantité de travail pour compresser et décompresser est la même, alors la compression est symétrique car la même méthode est utilisée pour chacune ces opérations.[10]

Schéma de la compression symétrique



**Figure II-1** : Schéma de la compression symétrique [17]

### II.3.2 La Compression asymétrique :

Si l'une des deux opérations demande plus de travail, on dit que la compression est asymétrique. Généralement la compression est plus lente que la décompression. [14]

## II.4 Principe général de la Compression des images : [10]

La compression d'une image se fait en général selon le schéma synoptique suivant:

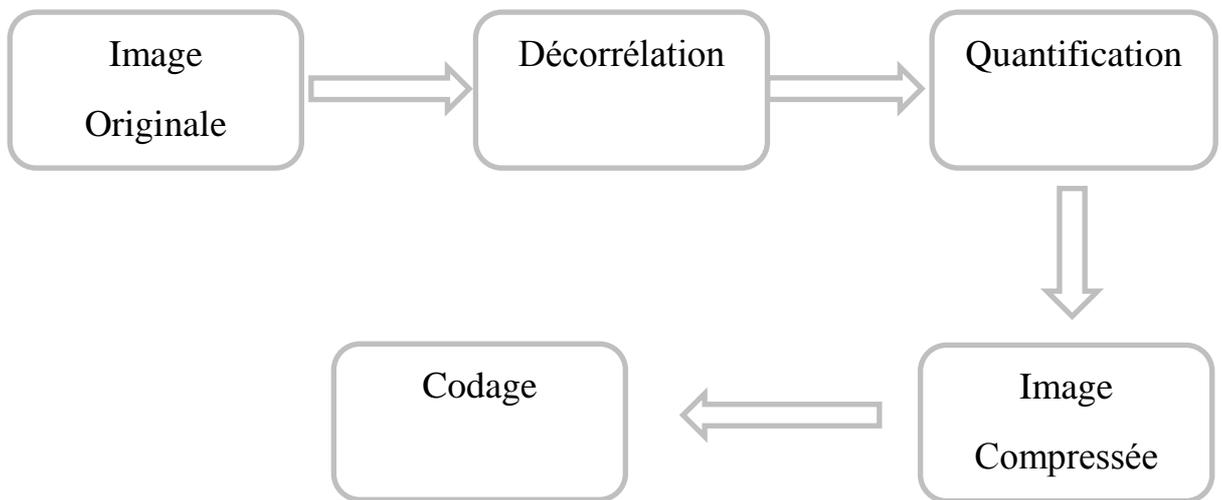


Figure II-2 : Schéma synoptique de compression.

#### ➤ 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 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 .

#### ➤ Quantification

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

## ➤ Codage

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

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

Plusieurs types de codage seront détaillés ci-après.

## **II.5 Classification des méthodes de compression :**

### **II.5.1 Méthodes de compression sans perte d'informations :**

Est dite aussi compactage ; elle est appelée ainsi, su après la décompression de fichier compressé il n'y a aucune perte d'informations par rapport à l'image originale. Cette méthode est basée sur les répétitions. L'information à compresser est vue comme la sortie d'une source de symboles qui produit des textes finis selon certaines règles. Le but est de réduire la taille moyenne des textes obtenus après la compression tout en ayant la possibilité de retrouver exactement le message d'origine comme RLE. [14]

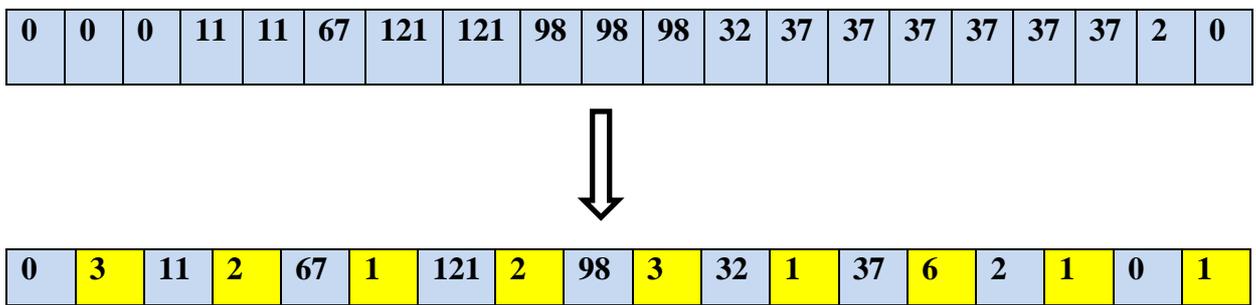
#### **II.5.1.1 Méthode de codage RLE (Run-length encoding) :**

Cette méthode de compression sans perte code les données en y recherchant des successions d'octets identiques. Le RLE est utilisé pour compresser des images codées sur un faible nombre de bits car, dans ce type d'image, il est fréquent de trouver des pixels contigus identiques. La méthode RLE est utilisée pour la compression des images au format PCX, dans une version du format BMP ainsi qu'au format JPEG. [17]

#### **II.5.1.2 Principe de fonctionnement :**

L'algorithme du RLE parcourt les données et recherche les octets identiques qui se suivent. Lorsqu'il trouve un même octet répété successivement au moins  $x$  fois, il crée une paire de données composée du nombre de répétitions suivi de l'octet en question.

La compression RLE est efficace pour compresser des données où les répétitions d'éléments consécutifs sont nombreuses ce qui n'est pas toujours le cas. C'est pourquoi, lorsque le nombre de répétition est trop petit (c'est à dire s'il est inférieur à  $x$ ), l'algorithme ajoute un caractère spécial dans le couple de données (nombre de répétition de l'octet, octet).

➤ **Exemple :****II.5.1.3 Codage Huffman :**

Le codage de Huffman est une méthode de compression statistique de données qui permet de réduire la longueur du codage d'un alphabet. Le code de Huffman (1952) est un code de longueur variable optimal, c'est-à-dire tel que la longueur moyenne d'un texte codé soit minimale. On observe ainsi des réductions de taille de l'ordre de 20 à 90%. [20]

Son principe est d'attribuer un code court à un symbole fréquent et un code plus long à un symbole plus rare. En se plaçant dans le cas d'un alphabet de sortie binaire, l'attribution du nombre de bits pour chaque symbole se fait par le biais d'un arbre binaire, construit en fonction de la fréquence d'apparition des symboles.

A chaque nœud de l'arbre est associé la fréquence d'apparition de l'ensemble des symboles présents dans son sous arbre. L'arbre est construit de manière ascendante, à partir de ses feuilles. Chaque feuille correspond à un symbole, et à sa fréquence d'apparition. Itérativement, jusqu'à l'obtention de la racine, un nœud père est généré pour le couple des deux nœuds ayant la fréquence d'apparition la plus faible.

Le code de chacun des symboles est alors le chemin partant de la racine jusqu'à la feuille lui correspondant (bit 0 pour aller au fils gauche, bit 1 pour aller au fils droit). Ce code est défini de telle sorte qu'aucun des symboles de l'alphabet ASO ne soit le préfixe d'un autre : chaque symbole étant associé à une feuille de l'arbre le chemin qui permet d'y accéder ne peut en aucun cas être le sous-chemin d'un autre. Ainsi, le codage est non ambigu.

La création de cet arbre peut être effectuée à l'aide d'un algorithme de complexité  $O(N \log_2 N)$ , avec  $N$  la taille de l'alphabet. [21]

➤ **Exemple de codage Huffman : [20]**

Supposons que notre fichier soit extrêmement simple, et constitué d'un mot unique : (anticonstitutionnellement)

Il y a 25 caractères dans ce fichier; chaque caractère étant codé par un octet de 8 bits (codage ASCII) cela signifie donc 25 octets, ou encore 200 bits.

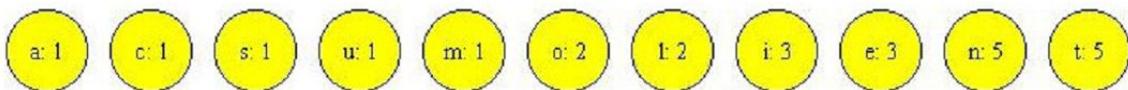
Table des fréquences:

Clé	a	c	s	u	m	o	l	i	e	n	t
Fréquence	1	1	1	1	1	2	2	3	3	5	5

Tous les autres octets dont la fréquence est nulle ne sont pas représentés.

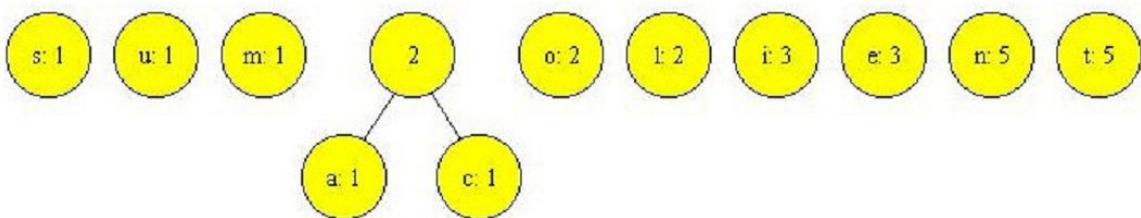
Créer un nœud terminal pour chaque entrée du tableau:

▪ **Étape 0:**

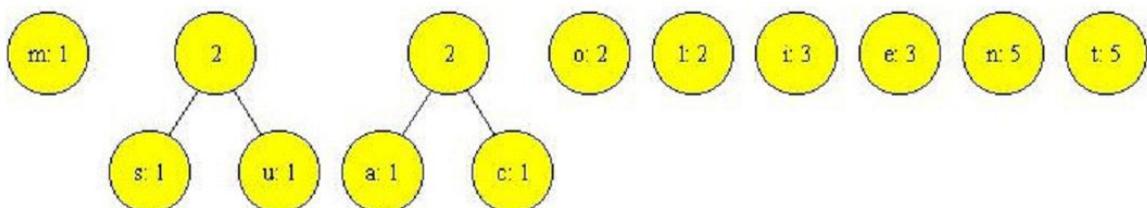


Ce qui nous fait pour l'instant 11 arbres contenant un seul nœud chacun . On commence l'itération: à chaque fois on supprime les deux arbres de gauche et on les remplace par un "arbre somme". Le nouvel arbre est inséré dans la liste en respectant l'ordre croissant, et on recommence jusqu'à ce qu'il ne reste plus qu'un seul arbre :

▪ **Étape 1:**

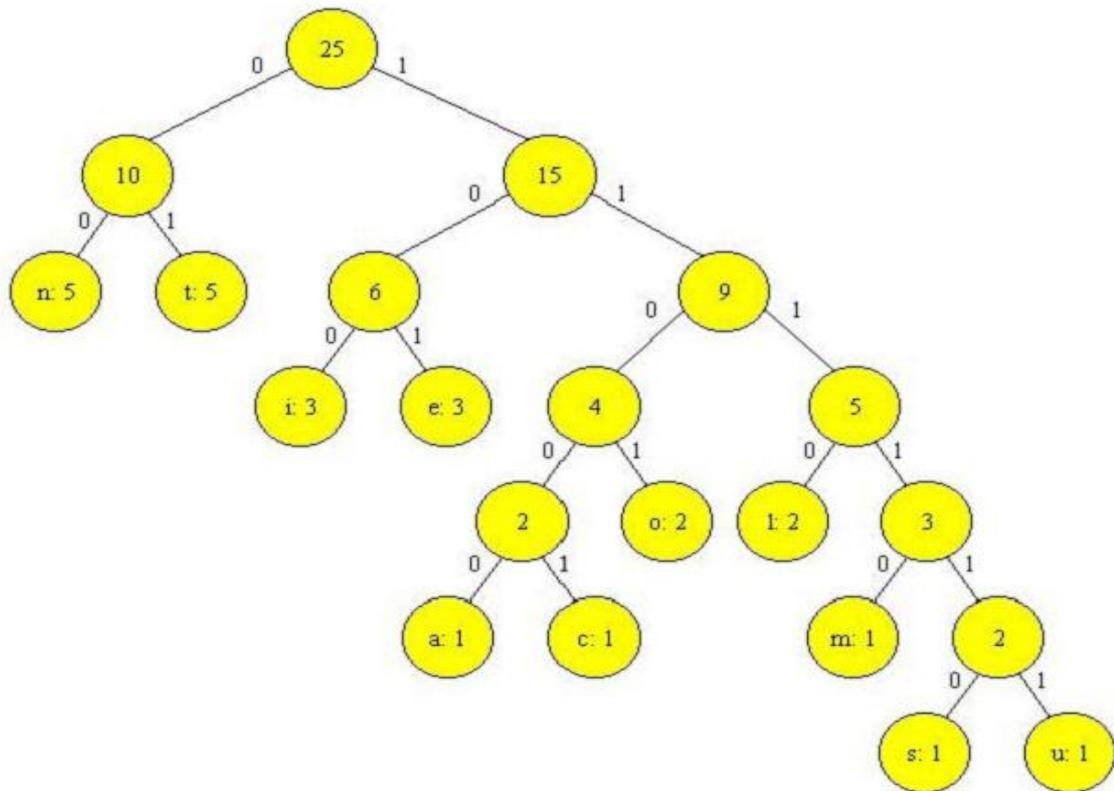


▪ **Étape 2:**



etc. ...

L'arbre final:



Maintenant, le code associé à chaque Clé n'est autre que le chemin d'accès au nœud terminal correspondant, depuis la racine, en notant 0 la branche de gauche et 1 la branche de droite.

**Finalemment :**

Clé	Code binaire
n	00
t	01
i	100
e	101
a	11000
c	11001
o	1101
l	1110
m	11110
s	111110
u	111111

Et voici maintenant, transcrit avec notre nouveau code, le mot de départ:

11000000110011001110100111110011000111111011001101000010111101110101111101  
010001 ce qui nous fait 81 bits, au lieu de 200 au départ ! Cela correspond à un taux de  
compression de 60 %.

#### **II.5.1.4 Codage arithmétique :[18]**

Cette partie va présenter une compression sans perte appelé codage arithmétique. Elle va d'abord définir l'intérêt de l'algorithme avant de présenter la compression et la décompression effectuées par ce codage.

#### **II.5.1.5 Description de la méthode :**

Le codage arithmétique est un codage statistique, c'est-à-dire que plus un caractère est représenté, moins il faudra de bits pour le coder.

Il s'agit d'un cousin du codage de Huffman qui cependant reste toujours plus efficace que ce dernier (sauf dans le cas particulier où tous les poids des feuilles/nœuds/racines de l'arbre de Huffman sont des puissances de 2). Il est aussi plus simple à implémenter.

L'avantage que possède le codage arithmétique sur le codage de Huffman est que ce dernier va coder un caractère sur un nombre entier de bits (il ne peut coder sur 1.5 bits) là où le codage arithmétique le peut. Par exemple, si un caractère est représenté à 90%, la taille optimale du code du caractère serait de 0.15 bit, alors que Huffman coderait sûrement ce symbole sur 1 bit, soit 6 fois trop.

Ce codage n'est que très peu utilisé en pratique mais elle reste présente, notamment dans le format JPEG2000.

#### **II.5.1.6 Compression :**

Pour présenter la compression, nous allons utiliser un exemple et nous décrivons chaque étape de compression. Codons le mot "ESIPE" à l'aide du codage arithmétique.

La première étape consiste à décompter chaque lettre du mot. Nous avons donc 2 'E', 1 'S', 1 'I' et 1 'P'. Nous en générons alors une probabilité de présence dans le mot soit 40% de chance de trouver un E et 20% de chance pour les autres lettres. Dernière actions à effectuer

pour cette première partie, nous affectons à chaque lettre un intervalle entre 0 et 1 de la manière suivante :

- La lettre 'E' à une probabilité de 40% (soit 0.4). Son intervalle est donc [0,0.4[
- La lettre 'P' a une probabilité de 20% (soit 0.2). Son intervalle est donc [0.4,0.6[
- Etc...

On obtient dès lors le tableau suivant :

Lettre	Probabilité	Intervalle
E	4/10	[0,0.4[
S	2/10	[0.4,0.6[
I	2/10	[0.6,0.8[
P	2/10	[0.8,1.0[

Le codage va maintenant consister à remplacer le mot ESIPÉ par un nombre flottant lui correspondant. Pour cela, le mot va se voir affecter un intervalle compris entre 0 et 1 où chaque nombre compris entre les deux intervalles permettra de retrouver le mot ESIPÉ.

L'algorithme appliqué est le suivant : le mot commence avec un intervalle de [0,1[. Puis pour chaque lettre croisée, nous appliquons la formule suivante :

- La borne inférieure (BI) du mot est modifiée avec le résultat du calcul " $BI + (BS - BI) * \text{Borne\_Inférieure\_Lettre}$ "
- La borne supérieure (BS) du mot est modifiée avec le résultat du calcul " $BI + (BS - BI) * \text{Borne\_Supérieure\_Lettre}$ "

Le tableau suivant montre les étapes du calcul:

Lettre	Borne Inférieure	Borne Supérieure
	0.0	1.0
E	0.0	0.4
S	0.16	0.24
I	0.208	0.224
P	0.2208	0.224
E	0.2208	0.22208

Dès lors, tous nombre flottant entre 0.2208 et 0.22208 est le format compressé du mot "ESIPE"

### II.5.1.7 Décompression

De la même manière, nous allons présenter la décompression par l'exemple en décompressant notre format compressé.

Prenons le nombre 0.2208 qui code le mot "ESIPE". Le principe de la décompression est très simple. Celle-ci suit les deux étapes suivantes qui se répète jusqu'à l'obtention du mot :

- La prochaine lettre du mot est celle dont l'intervalle contient le nombre du mot actuel (Ex : 0.2208 est dans l'intervalle de E donc la première lettre est E).
- On modifie le nombre représentant le mot à l'aide du calcul « (nombre du mot – borne inférieure de la lettre) / probabilité de la lettre (Ex : nombre du mot =  $(0.2208 - 0.0) / 0.4 = 0.552$ )

Le tableau suivant montre les différentes étapes de la décompression :

Mot	Lettre	Nouveau code
	E	0.552
E	S	0,76
ES	I	0,8
ESI	P	0,0
ESIP	E	

Ainsi, nous avons récupéré notre premier mot: ESIPE

### II.5.2 Méthodes de compression avec perte d'informations :

La compression avec pertes (en anglais lossy compression), par opposition à la compression sans pertes (lossless compression), se permet d'éliminer quelques informations pour avoir le meilleur taux de compression possible, tout en gardant un résultat qui soit le plus proche possible des données originales. C'est le cas par exemple de certaines compressions d'images ou de sons, telles que le MP3 ou le format Ogg Vorbis.

Etant donné que ce type de compression supprime des informations contenues dans les données à compresser, on parle généralement de méthodes de compression irréversibles.

Les fichiers exécutables ne peuvent par exemple pas être compressés selon ce type de méthode car ils ont notamment besoin de conserver leur intégrité pour fonctionner. En effet, il n'est pas concevable de reconstruire approximativement un programme en omettant des bits en en ajoutant.

En revanche, les données multimédias (audio, vidéo) peuvent tolérer un certain niveau de dégradation sans que les capteurs sensoriels (oeil, tympan, etc.) ne discernent une dégradation significative. [19]

## II.6 Evaluation de la compression :[10]

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

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

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

$$2) \text{ Taux de compression} = (1 - 1/\text{Quotient}) \times 100 \%, \text{ exprimé en pourcentage.} \quad \text{II-2}$$

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

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

**Remarque :** Le taux de bits est aussi appelé débit de compression.

## II.7 Qualité de reconstruction des images :

Pour les méthodes de compression sans perte d'informations, les images reconstruites sont exactement les images originales. Le problème de la qualité se pose pour les méthodes décompression avec perte d'informations. En effet, l'image reconstruite n'est pas exactement l'image originale, ainsi il nous faut une mesure de la qualité de l'image reconstruite. Parmi ces fonctions, on trouve :

$$EQM \frac{1}{N \times M} = \sum_{n=1}^N \sum_{m=1}^M (\mathbf{X}_{origine}(n, m) - \mathbf{X}_{reconstruite}(n, m))^2 \quad \text{II-4}$$

- 1) EQM (Erreur quadratique moyenne) : Elle permet de mesurer la dégradation entre une image  $X_{origine}$  et l'image  $X_{reconstruite}$  après compression de coordonnées  $(n,m)$  et de dimension  $(N,M)$ .

Avec :

$X_{origine}(n,m)$  et  $X_{reconstruite}(n,m)$  : représentent les valeurs des pixels d'image d'origine et reconstruite .

- 2) SNR (Signal to Noise Ratio) : Il permet de mesurer le signal original sur bruit d'une image.  $SNR=10$

$$SNR = 10 \times \log_{10} \left( \frac{\frac{1}{N \times M} \times \sum_{n=1}^N \sum_{m=1}^M (X_{origine}(n,m))^2}{EQM} \right) db \quad II-5$$

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

$$PSNR = 10 \times \log_{10} \left( \frac{(X_{max})^2}{EQM} \right) \quad II-6$$

Avec :

$X_{max}$  : désigne la valeur du pixel maximale possible dans l'image qui prend une valeur 255 pour une image à niveau de gris codé sur 8 bits. Une valeur de PSNR infini correspond à une image non dégradée, et cette valeur décroît en fonction de la dégradation.

## II.8 Conclusion

Les méthodes de compression des images n'ont cessé de s'améliorer, afin d'optimiser le stockage et la transmission travers les réseaux, Donc les experts travaillent sur le développement pour augmenter le taux de compression avec une qualité d'image très raisonnable.

Dans le chapitre suivant nous allons expliquer en détail la compression de la norme JPEG

# **CHAPITRE III :**

## **LA NORME JPEG**

### III.1 Introduction :

Durant les dernières années précédentes, le groupe joint photographique experts group, plus connu sous l'acronyme JPEG [10], a fait de grands efforts dans le but d'établir un standard international pour la compression des images numériques. Le but de JPEG était de concevoir une norme internationale qui satisfait à tous les besoins des différentes applications (médical, militaire, astronomique, ou journalistique) de la compression des images fixes (à niveau de gris et couleur), entre autres le stockage des images ainsi que l'échange (transmission) entre divers organismes, qui doit être plus facile et plus économique. Parmi plusieurs propositions, le groupe a fait une rigoureuse sélection et c'était le choix de la DCT  $8 \times 8$  qui a donné de bons résultats. Dans ce chapitre, nous décrirons l'algorithme de base de la norme JPEG (Baseline).

### III.2 Fonctionnement général du JPEG :

Le format JPEG, commence par traité séparément chaque composante de couleur.[10] Le JPEG utilise un autre format colorimétrique que le RVB : le format YUV. Y est la luminance (approximativement, la clarté du pixel), U la chrominance bleue (approximativement la déférence entre canal bleu et la luminance), et V la chrominance rouge.

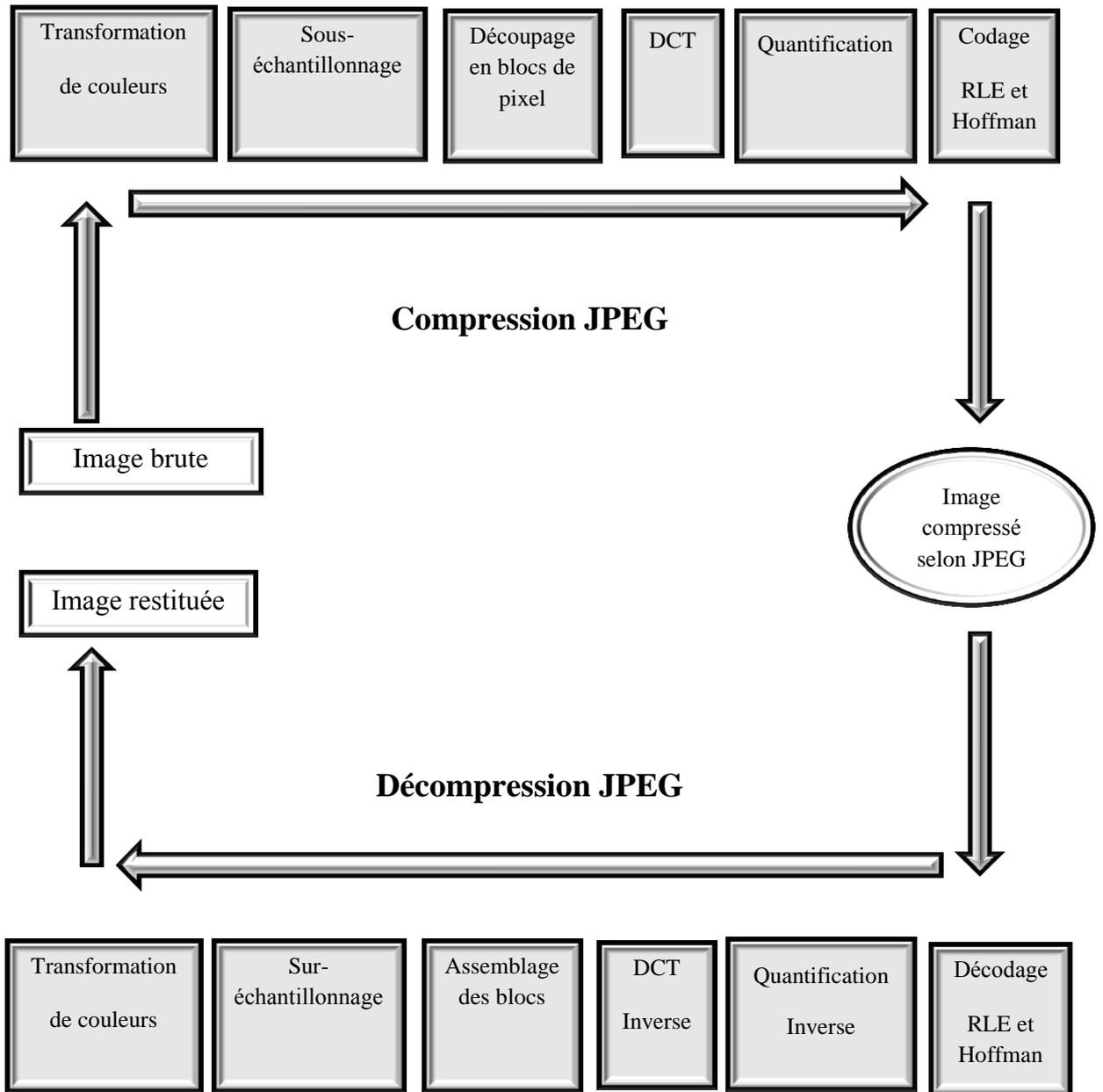
L'œil est moins sensible aux variations de chrominance : on peut se permettre de plus grandes approximations.

Pour rendre les calculs plus rapides, la compression JPEG découpe ensuite l'image en carreaux de  $8 \times 8$  pixels, auxquels elle applique la DCT.

On applique ensuite une quantification des valeurs, plus ou moins forte selon le niveau de compression et la fréquence associée. Pour cela, on divise chaque coefficient par une valeur donnée, et on arrondit le résultat. C'est à ce niveau-là que se situent les pertes.

Enfin, on linéarise l'image en lisant chaque carreau dans l'ordre croissant des fréquences on réalise une compression RLE, et une compression Huffman avant d'écrire dans le fichier. En général, les hautes fréquences sont peu représentées : à part les premières valeurs, après quantification, il y a beaucoup de coefficients nuls. La compression RLE est alors utile pour éliminer ces nombreux 0 consécutifs. En fonction de la quantification, les valeurs trouvées peuvent être assez peu variées : la compression Huffman peut alors efficacement réduire le poids du fichier.

La figure suivante explique les étapes de codage JPEG :



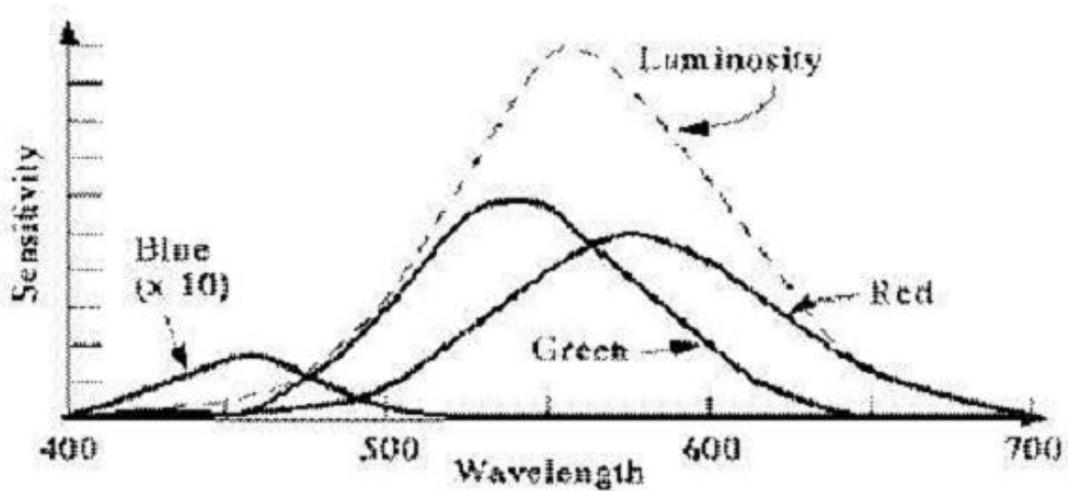
**Figure III-1:** Principe général de la compression/décompression JPEG.

### III.3 Transformation des couleurs :

C'est la transformation de l'image dans un espace de couleurs optimales.

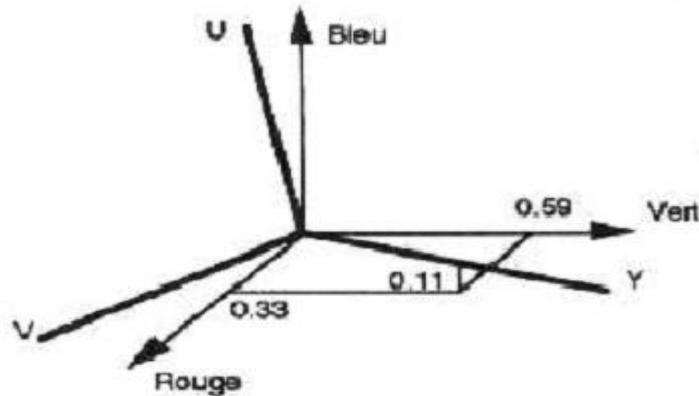
Luminance: Désigne le signal qui détermine les valeurs de contraste d'une image, du noir le plus profond jusqu'au blanc le plus pure [22]. (Son unité est la candela par mètre carré).

Chrominance: Désigne la partie du signal qui détermine les valeurs de couleur de l'image.



**Figure III-2:** Perception de l'œil humain

Dans ce graphe, on constate que l'œil est beaucoup plus sensible à la luminosité qu'à la chrominance, étant donné que l'image brute peut-être codée sur un mode de couleurs différent du YCbCr ou YUV (RVB le plus souvent), il faut donc transformer les couleurs, passer d'un mode de couleurs au YCbCr ou YUV.



**Figure III-3 :** Répartition RGB – YUV

Cela se fait à l'aide de la formule suivante :

**YUV :**

$$Y = 0.299 R + 0.587 V + 0.114 B$$

$$U = -0.16874 R - 0.33126 V + 0.5 B$$

$$V = 0.5 R - 0.41869 V - 0.08131 B$$

**YCbCr :**

$$\begin{pmatrix} XY[n] \\ XCb[n] \\ XCr[n] \end{pmatrix} \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} XR[n] \\ XG[n] \\ XB[n] \end{pmatrix} \quad \text{III-1}$$

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

Cette conversion permet une compression meilleure sans réduire considérablement la qualité de l'image.

#### III.4 Sous- échantillonnage de la chrominance (Cb et Cr) :

L'objectif ici va être de réduire l'information occupée par la chrominance. Si l'œil humain est très sensible à la luminance, elle distingue par contre très mal la chrominance. Ainsi, une perte d'information sur ce point passera relativement inaperçu.

On distingue 3 types majeurs de sous-échantillonnages.[12]

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

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

**Figure III-4 :** sous-échantillonnages 4:4:4

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

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

**Figure III-5 :** sous-échantillonnages 4:2:2

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

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

**Figure III-6 :** sous-échantillonnages 4:2:0

### III.5 Le découpage en bloc de pixels [12]

À partir d'ici, la compression va travailler sur des blocs de pixels seulement. Ces blocs sont des matrices de pixels 8\*8 soient 64 pixels.

### III.6 Transformation DCT :

C'est le fait d'appliquer une transformée en cosinus discret à un bloc de pixels pour retirer la redondance des données de l'image.

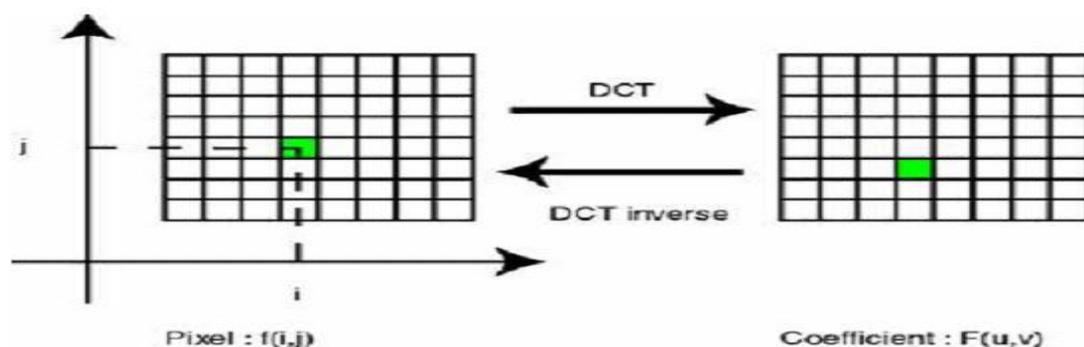
Jusqu'à présent, on a juste divisé l'image en sous-blocs de 8x8 pixels, transformé le mode de couleur en YUV ou YCbCr et appliqué un sous-échantillonnage sur les composantes de la chrominance.

Maintenant on va appliquer la DCT (Discrete Cosine Transform, ou bien (transformée en cosinus discrète en français)..[22]

La DCT est une variante de la transformée de Fourier. Elle a pour but de regrouper l'énergie en passant de la notion de pixels et couleurs à la notion de fréquences et amplitudes. En effet, elle prend un ensemble de points d'un domaine spatial et les transforme en une représentation équivalente dans le domaine fréquentiel. Dans notre cas, on étudie une image couleur, c'est à dire qu'il sera traité trois fonctions (de manière indépendantes) à 3 dimensions : X et Y, indiquant le pixel, et Z avec la valeur du pixel en ce point. Ces trois fonctions correspondent chacune à un des canaux RVB. Après l'application de cette transformée sur un bloc, on va avoir l'information essentielle stockée dans les basses fréquences (luminance) et l'énergie sera regroupée en haut à gauche de la matrice.

Pour ce qui est des hautes fréquences (chrominance), on les retrouve en bas à droite de la matrice, plus on approche du bas droite de la matrice plus leurs coefficients tendent vers 0.

Cette transformée va être appliquée à chaque composante pour chaque bloc. En effet, pour une image codée sur 24bits, on va appliquer la DCT 3 fois sur chaque bloc (1 pour la composante Y, 1 pour Cb ou U et 1 pour V ou Cr).



**Figure III-7** : La transformé DCT et IDCT

La transformée DCT s'exprime mathématiquement par

$$DCT(i, j) = \left(\frac{1}{\sqrt{2N}}\right) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) \cdot C(v) \cdot F(u, v) \cdot \cos\left(\frac{(2i+1) \cdot u \cdot \pi}{2N}\right) \cdot \cos\left(\frac{(2j+1) \cdot v \cdot \pi}{2N}\right) \quad \text{III-2}$$

N : la largeur d'un bloc ,ici N=8

i,j :les indices d'un coefficient de la DCT dans un bloc.

u,v : les indices d'un pixel de l'image dans un bloc.

DCT(i,j) : la valeur d'un coefficient dans un bloc.

F(u,v) : représente la valeur de la DCT au point de coordonnées (u, v)

Et la transformée DCT inverse (Ce qui donne dans notre cas (bloc de 8x8 pixels)

S'exprime par :

$$F(u, v) = C(u) \cdot C(v) \sum_{i=0}^7 \sum_{j=0}^7 f(i, j) \cdot \cos\left[\frac{\pi}{8} \left(i + \frac{1}{2}\right) u\right] \cos\left[\frac{\pi}{8} \left(j + \frac{1}{2}\right) v\right] \quad \text{III-3}$$

f(i,j) : La valeur du pixel aux coordonnées (i,j).

f(u,v) : La valeur de la DCT aux coordonnées (u,v).

Avec :

$$\left\{ \begin{array}{l} C(u) = C(v) = \frac{1}{\sqrt{2}} \quad \text{pour } u = v = 0 \\ C(u) = C(v) = 1 \quad \text{autres } u \neq 0, v \neq 0 \end{array} \right.$$

La transformation DCT est en fait une décomposition d'une image en plusieurs sous-images de bases (fonctions de bases). Dans notre cas, on va utiliser la transformation DCT dite 2 dimensions. En effet, la transformation se compose d'une transformation suivant les colonnes et une transformation suivant les lignes :

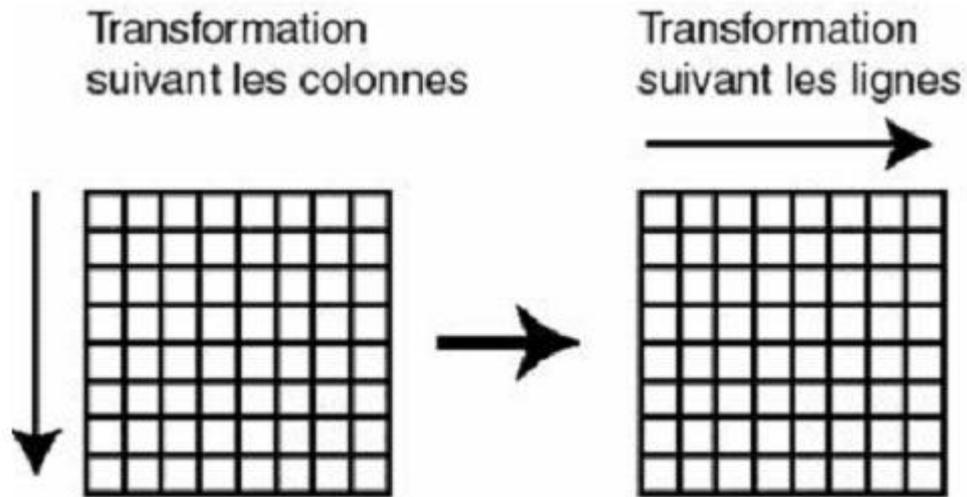


Figure III-8 : La transformation suivant les colonnes et les lignes

Le schéma suivant montre les différentes 64 sous-images produites par la DCT :

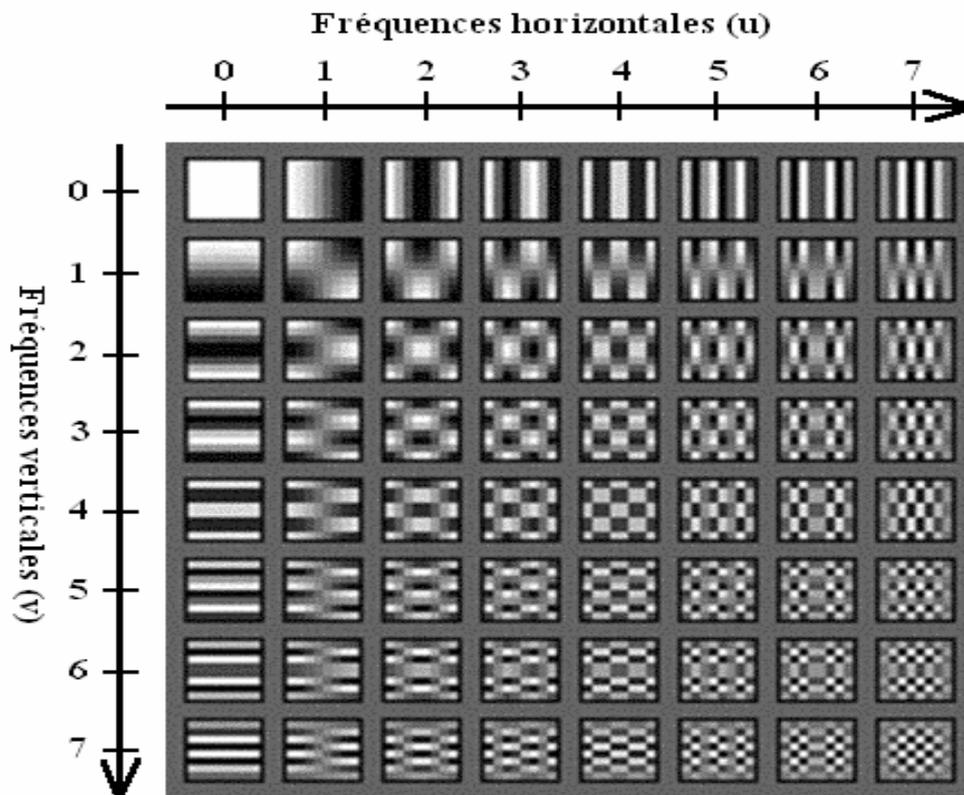


Figure (III-9) : les différentes 64 sous-images produites par la DCT

Pour illustrer la transformée DCT nous avons pris un exemple d'un bloc de  $(8 \times 8)$  pixels appelé S :

S=

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	15	157	157
162	162	161	161	163	158	158	158

En effectuant la transformée DCT à la matrice S on obtient la matrice des fréquences suivante

1260	-1	-12	-5	2	-2	-3	1
-23	-17	-6	-3	-3	0	0	-1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	1	1	0	0	0
-1	-1	1	2	0	-1	1	1
2	0	2	0	-1	1	1	-1
-1	0	0	-1	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

### III.7 La Quantification :

La quantification a pour objectif d'atténuer les hautes fréquences d'une image qui ont été mis en évidence par la DCT. L'œil humain distingue mal les zones de contrastes (les hautes fréquences) et la quantification va permettre de diminuer l'importance de ces informations superflues.[12]

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

La quantification est donc réalisée par une matrice de quantification  $Q$  de la taille du bloc de  $8 \times 8$  pixels. Elle s'applique sur un bloc de la DCT de telle sorte que [20], [22] :

$$F^*(u, v) = \text{entier le plus proche de } \frac{F(u, v)}{Q(u, v)} \quad \text{III-4}$$

Avec :

$F^*(u, v)$  est la matrice quantifiée,  $F(u, v)$  est la matrice des coefficients à quantifier et  $Q(u, v)$  est la table de quantification.

Et pour la quantification inverse :

$$F(u, v) = F^*(u, v) \times Q(u, v) \quad \text{III-5}$$

Avec  $F^*(u, v)$  représente la matrice quantifiée .

Dans les traitements d'images actuels, la table de quantification est définie par [22] :

$$Q(i, j) = 1 + (1 + i + j)Fq \quad \text{III-6}$$

Où

$Fq$  : Représente le facteur de qualité de la matrice de quantification. Plus  $Fq$  est grand, plus  $Q$  rendra les coefficients de la matrice à compresser se rapprochant du coin supérieur gauche à des valeurs nulles, et donc plus le signal bidimensionnel original sera dégradé. plus  $Fq$  est grand, plus la compression sera forte mais moins bonne sera la qualité de l'image reconstituée :  $Fq$  détermine le compromis à faire entre le gain en compression et la qualité de l'image reconstituée par rapport à l'image d'origine.

#### ➤ Exemple de quantification

Matrice de quantification pour  $Fq = 5$ ,  $q(i; j) = 1 + (1+i + j) * Fq$ .



Cette opération a éliminé les hautes fréquences non primordiales dans l'image d'où l'apparition de nombreux zéros en bas à droite.

### III.8 Le codage entropique [23]:

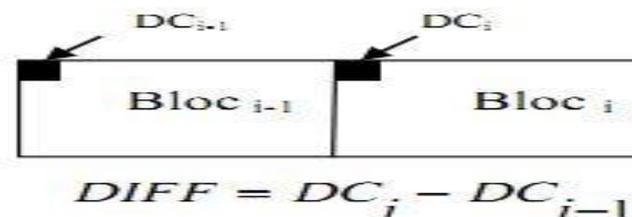
Enfin, la dernière étape est le codage des coefficients quantifié, c'est une opération sans perte. Un codage entropique consiste à coder les symboles qui ont une probabilité d'apparition forte avec de mots de longueur plus faible que les symboles qui ont une probabilité d'apparition faible.

#### III.8.1 Codage intermédiaire :

##### III.8.1.1 Codage intermédiaire des éléments DC:

Le coefficient DC et les 63 coefficients AC sont codé séparément comme représenté sur les figures (III-10) et (III-11) .Les coefficient DC sont codé en DPCM (Différentiel Pulse Code Modulation) en utilisant la valeur de coefficient DC de bloc précédent (équation(III.7) comme représenté sur la figure (III-10).

$$\text{DIFF} = \text{DC}_i - \text{DC}_{i-1} \quad \text{III-7}$$



**Figure III-10 :** Codage différentiel des coefficients DC

Le but de ce traitement est d'exploiter la corrélation entre les valeurs DC des blocs adjacents et les codes plus efficacement, car ils contiennent typiquement la plus grande partie d'énergie d'image.

Cette différence est codée comme suivant :

On assigne au coefficient différentiel Diff DC (i) deux symboles de longueur variable.

#### SIZE / AMPLITUDE

**SIZE :** définit la catégorie d'amplitude ou appartient le coefficient Diff DC (i) (voir tableau (III-1)) et qui est codée de 4 bits qu'on notera (SSSS) en binaire.

**AMPLITUDE:** représente la valeur de l'amplitude du coefficient différentiel Diff DC (i) en binaire.

Catégorie	SSSS	Amplitude
0	0000	0
1	0001	-1...0,0...1
2	0010	_3...-2,2...3
3	0011	-7...-4,4...7
4	0100	-15...-8,8...15
5	0101	-31...-16,16...31
6	0110	-63...-32,32...63
7	0111	-127...-64,64...127
8	1000	-255...-128,128...255
9	1001	-511...-256,256...511
10	1010	-1023...-512,512...1023
11	1011	-2047...-1024,1024...2047

**Tableau III-1** :Catégories d'amplitudes des différences DC.

Pour les nombres négatifs on va écrire le complément à 1 de la valeur absolue en la Diff DC (i) de l'amplitude.

➤ **Exemple**

Pour deux blocs successifs on a :

$$D(i) = 500.$$

$$DC(i-1) = 355.$$

$$\text{Diff DC}(i) = 500 - 355 = 145.$$

D'après le tableau (II-1), 145 appartient à la catégorie 8 qui a une valeur en binaire 1000

( SSSS= 1000) et son codage complet sera 100010010001 :

Composé comme suit

SSSS	Amplitude
1000	10010001

### III.8.1.2 Codage intermédiaire des éléments AC :

Les 63 coefficients AC commençant à partir du coefficient AC (0,1) sont codé par plage après un balayage de zigzag comme représenté sur la figure (III.11).

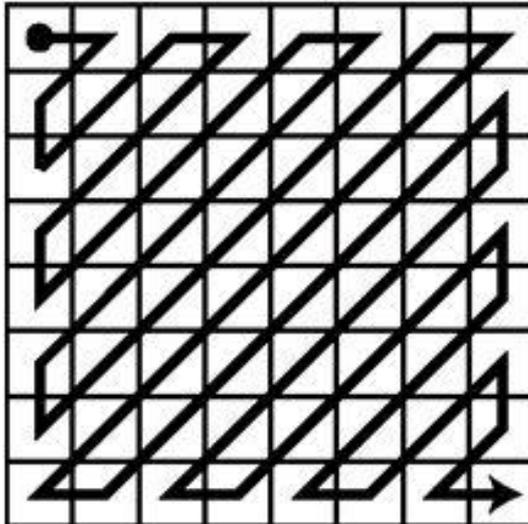


Figure III-11 : Lecteur en zig-zig

Le but de l'adoption d'un modèle de balayage de zigzag est de faciliter le codage par entropie en rencontrant d'abord les coefficients les plus susceptibles d'être non-nuls (les coefficients de basses fréquences). C'est dû au fait que, pour la plupart des scènes naturelles, l'énergie d'image se concentre principalement dans quelques coefficients de basses fréquences.

On va utiliser un codage qui est basé sur la longueur de la séquence des zéro consécutifs le codage RLE (Run length Encoding). Le codage de chaque coefficient AC, sera constitué de trois parties :

#### 1 Longueur de la séquence des zéro (Run Length)

C'est le nombre de zéro consécutifs qui précède un coefficient AC non nul (Après un arrangement zigzag de ces coefficients). Ce nombre va être codé sur 4 bits (RRRR), il est compris entre 0 et 15.

#### 2 size

C'est la gamme dynamique de l'amplitude du coefficient AC non nul qui suit la séquence des zéro, définit la catégorie d'amplitude ou appartient le coefficient AC (tableau (II-2) ) et qui est codée en 4 bits qu'on notera (SSSS) en binaire.

Catégorie	SSSS	Amplitude
1	0001	-1...0,0...1
2	0010	_3...-2,2...3
3	0011	-7...-4,4...7
4	0100	-15...-8,8...15
5	0101	-31...-16,16...31
6	0110	-63...-32,32...63
7	0111	-127...-64,64...127
8	1000	-255...-128,128...255
9	1001	-511...-256,256...511
10	1010	-1023...-512,512...1023

Tableau III-2 :Catégories d'amplitudes des coefficients AC.

### 3 Amplitude:

Représente la valeur de l'amplitude du coefficient AC non nul en binaire.

#### ▪ Remarque 1

Lorsqu'une séquence de zéro est rencontrée à partir d'un point donné jusqu'à la fin du bloc, on écrira le code 00000000 (RRRR = 0000, SSSS = 0000).

#### ▪ Remarque 2

Pour les coefficients AC négatifs, on représentera leur amplitude par le complément à 1 de la valeur absolue du coefficient.

#### ➤ Exemple 1

Soit la séquence suivante issue d'un arrangement zigzag :

.....6,0,0,0,0,0,0,15,. Son codage sera : 011101001111, composé comme suit :

RRRR	SSSS	Amplitude
0111	0100	1111
Nombre de zéros 7	Appartiens à la Catégorie4	15

### III.8.2 Codage HUFFMAN :

Le codage d'HUFFMAN nécessite plusieurs tables de codage. Celle-ci sont soit fournies au décodeur en les transmettant avec les données comprimées ou construites par lui-même.

#### III.8.2.1 Codage d'HUFFMAN des coefficients Diff DC:

A chaque coefficient Diff DC, le codeur va lui attribuer un code de Huffman selon sa catégorie (indiquée par les 4 bits SSSS) dans la table de codage d' Huffman, la deuxième information (l'amplitude) n'est pas modifiée. Les coefficients Diff DC seront codés de la manière suivante :

Code HUFF (SSSS) + Amplitude

L'attribution du code HUFFMAN est donnée par le tableau (III-3).

Catégorie	SSSS	Code HUFFMAN	Longueur du code
0	0000	010	3
1	0001	011	3
2	0010	100	3
3	0011	00	2
4	0100	101	3
5	0101	1110	4
6	0110	11110	5
7	0111	111110	6
8	1000	1111110	7
9	1001	11111110	8
10	1010	111111110	9
11	1011	1111111110	10

**Tableau III-3 :** Codes de HUFFMAN des coefficients Diff DC.

#### ➤ Exemple

Soit la séquence binaire suivante issue d'un codage intermédiaire d'un coefficient Diff DC :

100010010001, composé comme suit :

SSSS	Amplitude
1000	10010001

Son codage HUFFMAN d'après la table (III-3) pour la catégorie 8 (SSSS =1000) sera le suivant : **111111010010001**, composé de la manière suivante :

Code HUFFMAN de SSSS	Amplitude
1111110	10010001

### III.8.2.2 Codage HUFFMAN des coefficients AC :

Pour les éléments AC du codage intermédiaire, seules les deux premières informations (codées sur 8 bits RRRRSSSS) vont être codées, la troisième information (amplitude du coefficient AC) ne sera pas modifiée. Les coefficients AC seront codés de la manière suivante **Code HUFF (RRRRSSSS) + Amplitude**.

Les codes de HUFFMAN des coefficients AC sont donnés dans le tableau (III-4) :

Runlength /Catégorie (size)	Code de Hoffman	(Longueur de code)
0 /0	1010 (=EOB)	4
0 /1	00	2
0 /2	01	2
0 /3	100	3
0 /4	1011	4
0 /5	11010	5
0 /6	111000	6
0 /7	1111000	7
0 /8	111110110	10
0 /9	111111110000010	16
0 /A	111111110000011	16
1 /1	1100	4
1 /2	111001	6
1 /3	1111001	7
1 /4	111110110	9
1 /5	11111110110	11
1 /6	111111110000100	16
1 /7	111111110000101	16
1 /8	111111110000110	16
1 /9	111111110000111	16
1 /A	111111110001000	16
2 /1	11011	5
2 /2	11111000	8
.	.	.
.	.	.
.	.	.
3 /5	1111111110010001	16
3 /6	1111111110010010	16
.	.	.
.	.	.
.	.	.
F /7	1111111111111011	16
F /8	1111111111111100	16
F /9	1111111111111101	16
F /A	1111111111111110	16

Tableau III-4 : Codes de HUFFMAN des coefficients AC.

➤ **Exemple :**

Soit la séquence binaire issue d'un codage intermédiaire d'un coefficient AC :

0011010101100, composé comme suit :

<b>RRRR</b>	<b>SSSS</b>	<b>Amplitude</b>
<b>0011</b>	<b>0101</b>	<b>01100</b>

Son codage HUFFMAN d'après le tableau (III-4) pour (Run length /Catégorie (size)=3/5) sera : 111111111001000101100 , composé de la manière suivante :

<b>Code HUFFMAN de (RRRRSSSS)</b>	<b>Amplitude</b>
<b>1111111110010001</b>	<b>01100</b>

### III.9 La décompression :

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

Voici dans notre exemple le résultat de la décompression :

$$F = \begin{bmatrix} 144 & 146 & 149 & 152 & 154 & 156 & 156 & 156 \\ 148 & 150 & 152 & 154 & 156 & 156 & 156 & 156 \\ 155 & 156 & 157 & 158 & 158 & 157 & 156 & 155 \\ 160 & 161 & 161 & 162 & 161 & 159 & 157 & 155 \\ 163 & 163 & 164 & 164 & 162 & 160 & 158 & 156 \\ 163 & 163 & 164 & 164 & 162 & 160 & 158 & 157 \\ 160 & 161 & 162 & 162 & 162 & 161 & 159 & 158 \\ 158 & 159 & 161 & 161 & 162 & 161 & 159 & 158 \end{bmatrix}$$

Ainsi que la matrice d'erreur :

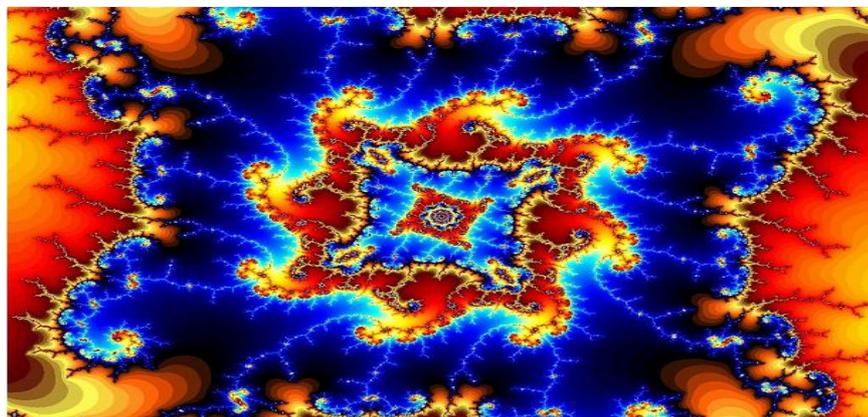
$$e = \begin{bmatrix} -5 & -2 & 0 & 1 & 1 & -1 & -1 & -1 \\ -4 & 1 & 1 & 2 & 3 & 0 & 0 & 0 \\ -5 & -1 & 3 & 5 & 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & -2 & -1 & 0 & 2 & 4 \\ -1 & 0 & 1 & -2 & -1 & 0 & 2 & 4 \\ -2 & -2 & -3 & -3 & -2 & -3 & -1 & 0 \\ 2 & 1 & -1 & 1 & 0 & -4 & -2 & -1 \\ 4 & 3 & 0 & 0 & 1 & -3 & -1 & 0 \end{bmatrix}$$

**Remarque :**

Les erreurs sont au maximum de 5 et en moyenne 1,6 sur environ 150 ce qui nous donne une erreur moyenne d'environ 1 %, et tout cela pour un passage de 64 à 10 valeurs (avec le caractère de fin) ; à cela, il faut rajouter la matrice de quantification, mais comme généralement on compresse de gros fichiers, elle n'influence que peu.

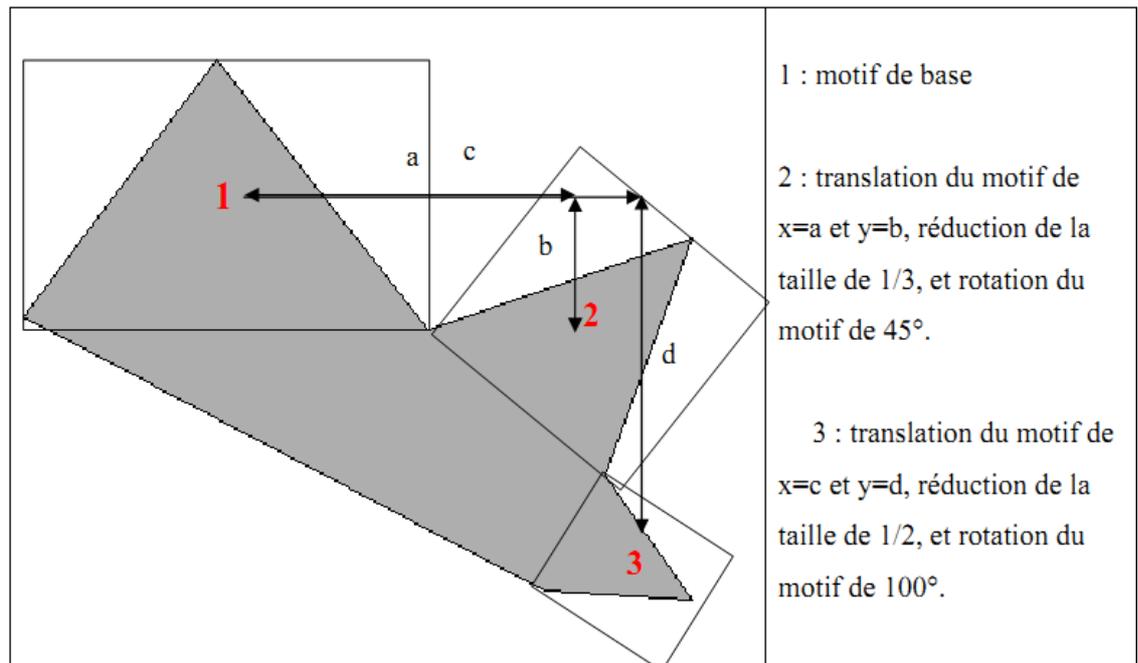
**III.10 Autres méthodes de compression :****III.10.1 La compression fractale : [24]**

Dans les années 1950, un mathématicien nommé benoît Mandelbrot, découvrit ce qu'on appelle aujourd'hui, les fractales. La représentation graphique des solutions délivrées par un algorithme spécifique donne une image constituée de répétitions infinies d'une même forme, d'un même motif. ( Figure III-12)



**Figure III-12 :** Fractale du type Julia

Puisqu'une image fractale est une répétition d'un motif plus ou moins rétréci, plus ou moins transformé (translation, rotation, homothéties...), Barnsley eût l'idée de rechercher dans n'importe quelle image, des formes similaires. Ainsi, au lieu de coder toute l'image, la compression fractale n'encode plus que le ou les motifs, ainsi que les transformations à réaliser.



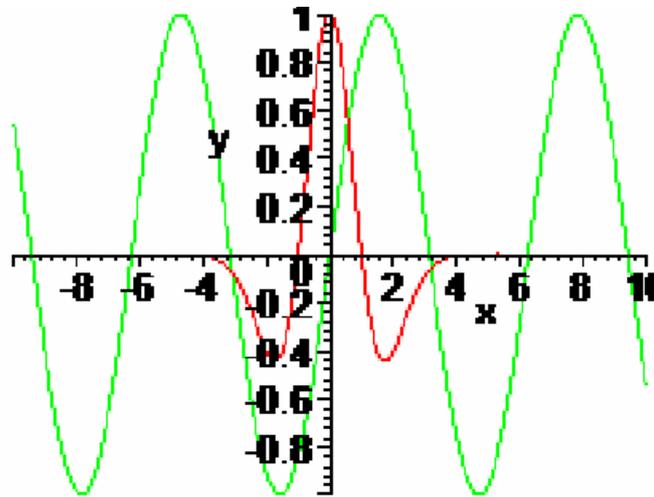
**Tableau III-5** : Exemple de compression fractale

En 1988, on annonçait des taux de compression allant jusqu'à  $1/10\ 000$ . Mais ce taux de compression fabuleux a été réalisé en 100 heures de travail, avec un ordinateur biprocesseur (donc deux processeurs), et toute une équipe de chercheurs. Aujourd'hui, et malgré l'avancée considérable des microprocesseurs, la compression fractale peut compter sur des taux allant de  $1/4$  à  $1/100$ , pour des temps raisonnables.

### III.10.2 La compression par ondelettes : [24]

le JPEG 2000, utilisant une nouvelle et prometteuse méthode de compression, appelée compression par ondelettes.

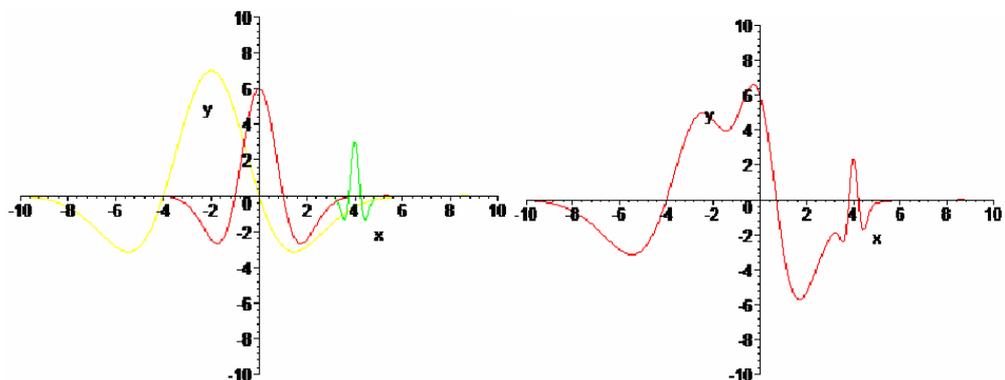
La théorie des ondelettes a été inventée par le mathématicien hongrois Alfred Haar dans les années 1910. Une ondelette est une transformation de fonction, comme Laplace ou Fourier, qui oscille principalement dans un intervalle restreint.



**Figure III-13 :** Représentation d'une sinusoïde et d'une ondelette

Nous pouvons voir sur la (Figure III-13) que, contrairement à la fonction Cosinus (en vert), l'ondelette « Chapeau Mexicain » possède une amplitude variable, et est quasiment nulle en dehors de l'intervalle  $[-4,4]$ . Cette variation très locale de la fonction permet néanmoins de savoir précisément ce qui se passe en n'importe quel endroit du signal original (non transformé).

Contrairement au format JPEG, qui décompose une image en blocs de  $8 \times 8$  pixels, la compression JPEG 2000 transforme chaque ligne horizontale en un signal, qui sera ensuite transformé en somme d'ondelettes. En effet, la variation de couleur et d'intensité de chaque pixel d'une ligne peut être assimilée à la variation de deux signaux. Chaque signal sera ensuite directement transformé en une série d'ondelettes, répétées en différents endroits, et à différentes échelles, pour que la somme décrive le plus exactement le signal original (Figure III-14). L'algorithme éliminera les variations les plus infimes pour compresser encore d'avantage l'image.



**Figure III-14 :** Exemple de fonctions ondelette et leur somme

Aujourd'hui, le format JPEG 2000 attend d'être officiellement validé par l'Organisation Internationale de Normalisation (ISO), pour se voir enfin utilisé dans tous les logiciels d'imagerie. Une variante de ce format a aussi vu le jour, le MJ2 (Motion JPEG 2000). Celui-ci utilise aussi la compression par ondelettes, mais pour des images en mouvement c'est-à-dire pour la vidéo numérique.

### **III.11 Conclusion :**

Dans ce chapitre, nous avons étudié en détail la compression des images fixes selon la norme JPEG (Baseline), qui s'effectue en plusieurs étapes qu'on peut résumer ici en deux étapes essentielles et cela après transformation couleur et sous-échantillonnage de l'image bien sûr:

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

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

2) La 2<sup>ème</sup> étape Codage de Huffman : Qui s'effectue sans pertes. Un codage qui se base sur des caractéristiques statistiques; il s'agit du codage de Huffman. Dans cette partie, les informations codées seront reconstituées lors du décodage telles qu'elles sont, c'est-à-dire sans aucune perte. Dans le chapitre suivant on va appliquer la norme JPEG sous la programmation MATLAB

**CHAPITRE VI :  
APPLICATION ET  
IMPLEMENTATION**

## IV.1 Introduction :

Dans ce chapitre nous allons présenter notre application de compression d'image JPEG dans MATLAB et différentes simulations obtenus

## IV.2 Langage de programmation utilisé:

### Pourquoi a-t-on choisi Matlab ?

MATLAB permet le travail interactif soit en mode commande, soit en mode programmation, tout en ayant toujours la possibilité de faire des visualisations graphiques. Il possède les particularités suivantes :

- Puissance de calcul ;
- la continuité parmi les valeurs entières, réelles et complexes ;
- l'étendue de gamme des nombres et leurs précisions ;
- la compréhension de la bibliothèque mathématique ;
- l'inclusion des fonctions d'interface graphique et des utilitaires dans l'outil graphique ;
- La possibilité de liaison avec les autres langages classiques de programmations ;

### IV.2.1 Interface Graphique Utilisateur (GUI) :

Pour l'interface graphique, des représentations scientifiques et même artistiques des objets peuvent être créées sur l'écran en utilisant les expressions mathématiques ou bien directement en utilisant un outil graphique. En effet, pour la conception de notre simulateur nous avons choisi la boîte à outil GUIDE sous MATLAB.

Une interface graphique utilisateur est un ensemble d'éléments graphiques permettant d'interagir entre des (parties de)programmes et l'utilisateur en affichant des informations (textes, graphiques, images, ...) et en déclenchant des actions(callbacks) à la suite d'évènements : clic de souris, déplacement de souris, entrée au clavier, ...). Un exemple d'interface utilisateur est la fenêtre liée à une figure et les objets qu'elle peut contenir (menus, boutons, axes, ...).

La version 6 et 7 de MATLAB possède un outil nommé GUIDE (Graphical User Interface Développement Environnement)facilitant la génération des scripts MATLAB nécessaires à la création d'une interface utilisateur.

## IV.2.2 Présentation générale de GUIDE :

La page d'accueil de l'outil est la suivante :

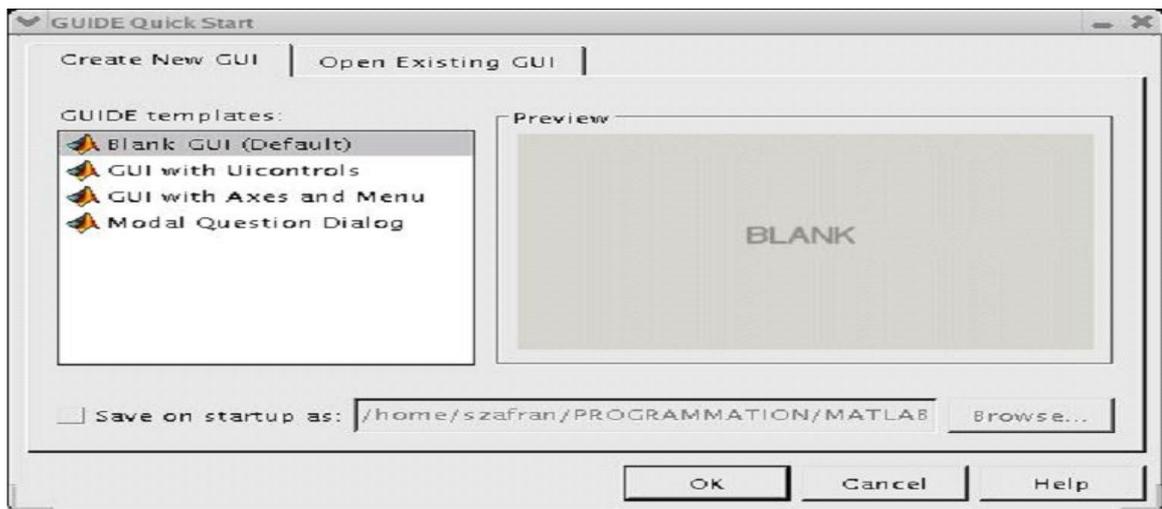


Figure VI-1 : Page d'accueil de l'application

Cliquez sur OK pour créer une nouvelle interface. La fenêtre d'édition d'interface s'ouvre :

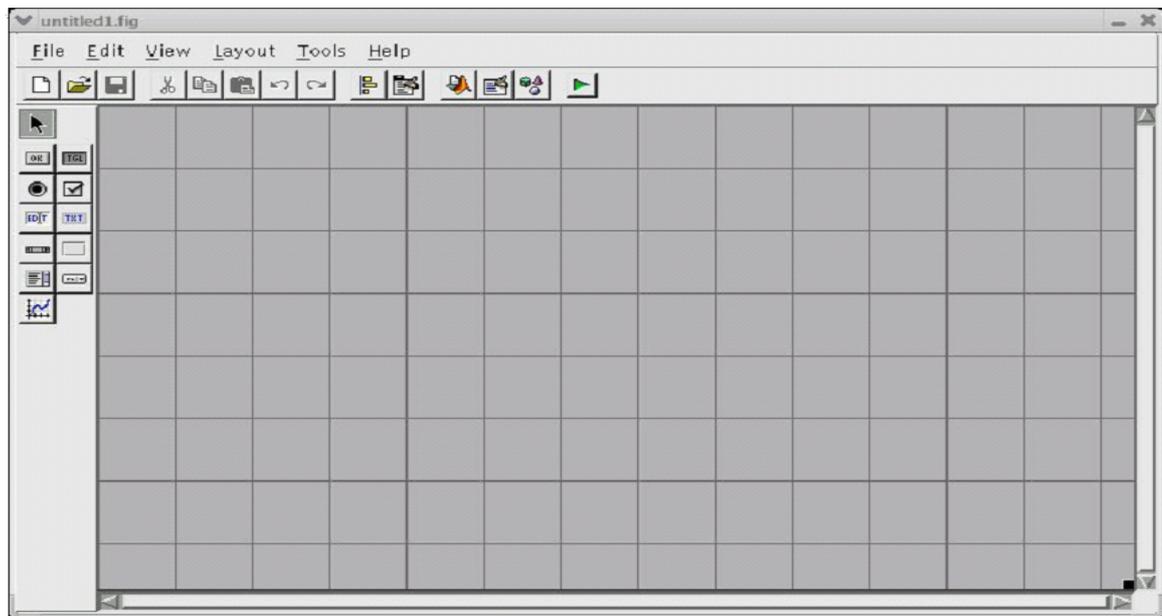


Figure VI-2 : La fenêtre d'édition d'interface

Cette fenêtre permet l'ajout/édition/suppression des différents éléments de l'interface. Au début, seul l'objet correspondant à la figure est présent.

### IV.3 Présentation de l'Application :

Notre application a été développée avec MATLAB 7 en utilisant Interface Graphique Utilisateur(GUI) sous WINDOWS XP, qui était nommée **Application de Compression JPEG**



Figure VI-3 : Interface d'entrée de l'application

#### IV.3.1 Les menus:

##### IV.3.1.1 Le menu « image»:

Ce menu contient les fonctions suivantes:

- **Charger** : pour charger l'image et ces paramètres à compresser dans la fenêtre de travail.
- **Effacer tout**: efface toute les informations et les images u logiciel
- **Enregistrer** : pour enregistrer
- **Imprimer** : si l'utilisateur possède une imprimante il peut imprimer la fenêtre de travail.
- **Quitter** : pour sortir de notre application.



Figure VI-4 :Le menu « image»

#### IV.3.1.2 Le menu « caractéristiques»:

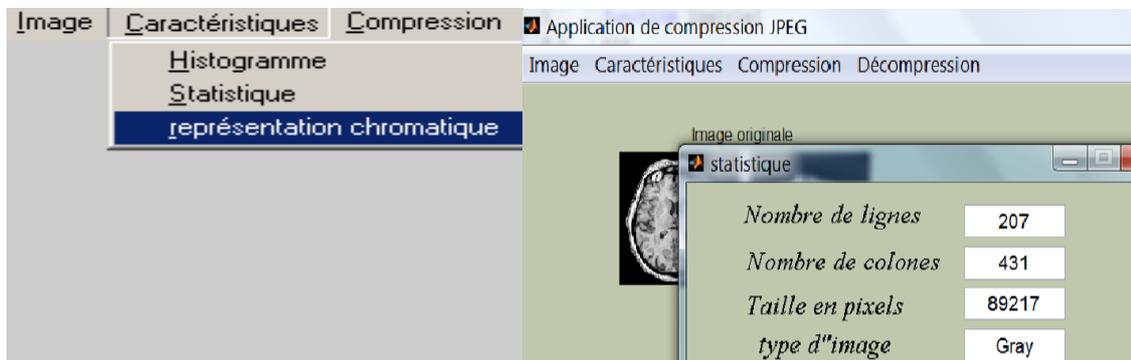


Figure VI-5 :Le menu « caractéristiques»

Dans ce menu on peut visualiser l'histogramme de l'image ainsi que quelque information concernant l'image par le bouton «statistique » :

### VI.4 L'implémentation de l'algorithme JPEG:

L'algorithme de compression JPEG va suivre les étapes suivantes:

- **Pour la compression:**

- Le prétraitement
- La DCT
- La quantification
- La lecture zigzag
- Le RLE

- Le codage entropique de Huffman
- **Pour la décompression:**
  - Chargement du fichier compressé
  - Décodage de Huffman
  - Inverse du RLE
  - Lecture inverse du zigzag
  - Restitution de l'image décompressée.

### **VI.4.1 La compression:**

#### **VI.4.1.1 Le prétraitement:**

Dans cette étape nous avons essayé les trois méthodes de prétraitement citées auparavant dans le chapitre 3.

Donc l'utilisateur peut choisir dans le cas d'une image en couleur vraie entre :

- Saisir directement la représentation RGB de l'image et lui appliquée une décomposition en blocs de 8x8 pour entamer l'étape de la DCT.
- Ou bien, passé à la représentation YCrCb et ensuite poursuivre le processus JPEG
- Ou passer à la représentation YCrCb de l'image et prendre la moyenne des pixels deux a deux (2pixels en hauteur et 1 en largeur)
- Ou encore passé à la représentation YCrCb et prendre la moyenne des pixels quatre a quatre (2pixels en hauteur et deux en largeur).

Bien sûr en parle de prendre la moyenne des pixels dans les matrices de chrominance car l'œil est moins sensible au changement de couleur qu'au changement d'intensité cela veut dire qu'on n'a pas touché à la matrice Y.

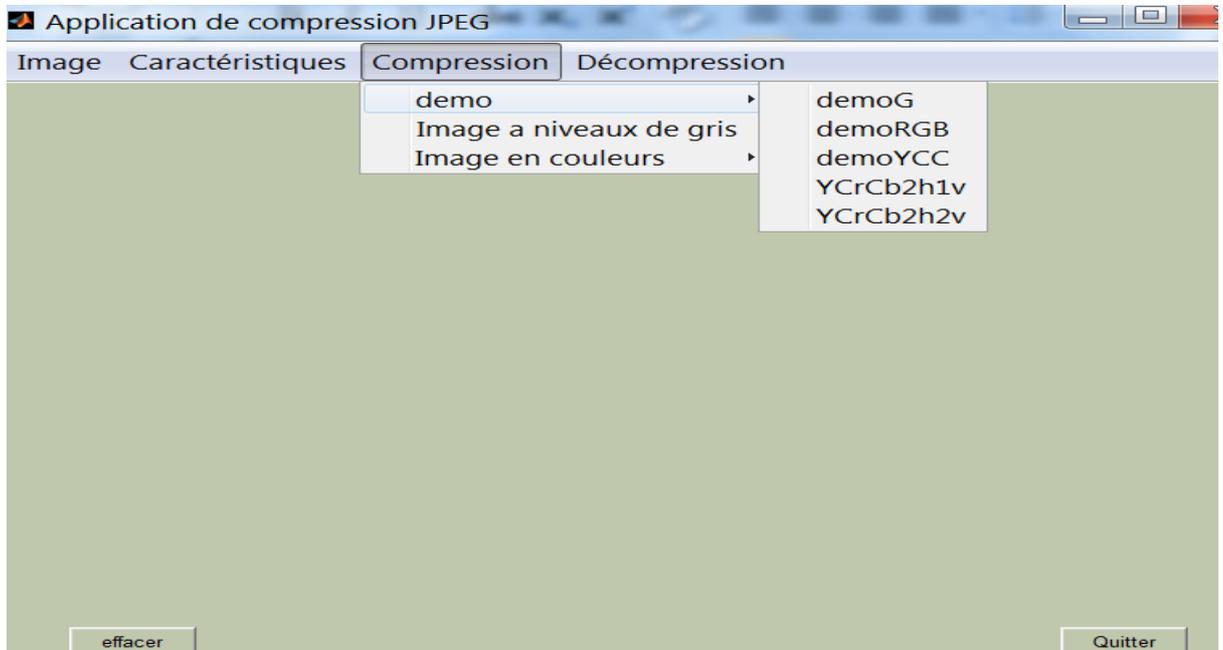


Figure VI-6:prétraitement

#### VI.4.1.2 La DCT:

Cette transformée est la clef de voûte de la compression JPEG. En effet cette transformée permet de convertir une courbe analogique ou digitale en sa représentation sous forme de fréquences. Il est important de noter que cette transformation est réversible (on parle de DCT inverse) et qu'elle est conservative (il n'y a pas de dégradation des données).

On a constaté lors de l'implémentation de l'algorithme JPEG que les formules classiques de la DCT et de la IDCT (DCT inverse) données par :

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

La transformée inverse de la DCT a pour équation :

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

Avec :

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

Sont complexes à implémenter et très lentes lors de l'exécution du programme ce qui rend l'algorithme inefficace, par exemple :

Pour une image a niveaux de gris de dimension 256x256 la DCT et la IDCT pour elles seules consommaient une dizaine de minute a peu près.

Une manière plus efficace de réaliser la DCT est d'utiliser les opérations de matrices. Pour effectuer cette opération il faut d'abord créer une matrice NxN, c'est la matrice de transformée de cosinus C. Une fois cette matrice construite il nous faudra aussi construire la matrice transposée pour que la formule devienne:

$$\text{DCT} = \text{C} * \text{Pixels} * \text{C}^t$$

$$C = \begin{matrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4175 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.02778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{matrix}$$

$C^t$  :est la matrice transposé de C

### VI.4.1.3 Résultats après la DCT:

Le point le plus important de la DCT est le résultat produit par celle-ci. En effet pour la matrice résultante le coefficient situé en (0,0) appelé "coefficient continu" représente une moyenne de la grandeur d'ensemble de la matrice d'entrée. Ce coefficient continu est plus grand d'un ordre de grandeur à toute autre valeur de la matrice de la DCT. Plus on s'éloigne du coefficient continu moins les valeurs apportent une contribution à l'image.

Donc cela signifie qu'en effectuant la DCT sur les données d'entrées, nous avons concentré la représentation de l'image dans les coefficients en haut à gauche de la matrice de sortie.

#### VI.4.1.4 La quantification:

Pour cette étape qui a est l'origine des pertes dans l'algorithme JPEG on va utiliser la relation cité dans le chapitre précédent est qui est définie par :

$$\text{Valeur quantifiée} = \frac{DCT(i, j)}{Quantum(i, j)} \text{ arrondi à l'entier le plus proche.}$$

Avec :  $Quantum(i, j) = 1 + (1 + a(i^n + j^n))F_q$

On a pris:  $a=n=1$

Donc le quantum devient :  $Quantum(i, j) = 1 + (1 + i + j)F_q$

Avec  $F_q$ : facteur de qualité

Les tables de quantification par défaut pour les matrices de la chrominance et la luminance pour un facteur de qualité moyen sont données par:

$$T_{CH} = \begin{matrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 36 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{matrix}$$

$$T_L = \begin{matrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{matrix}$$

Remarquez bien que les valeurs des éléments de ces deux matrices augmentent en allant du coin gauche supérieur vers le coin droit inférieur.

#### **VI.4.1.5 La compression conservative :**

Une fois tout ceci effectué il ne nous reste plus qu'à coder les données ainsi produites. Nous allons voir que le reste du codage va s'effectuer en trois étapes:

- la séquence zigzag
- le codage RLE
- le codage e Huffman

#### **VI.4.1.6 La séquence zigzag:**

Comme nous l'avons vu précédemment grâce à la quantification un grand nombre de valeurs de la matrice se trouvent réduite à 0. Le fait qu'il y ait beaucoup de valeurs mises à 0 va nous permettre de compresser encore mieux grâce à la compression RLE que nous verrons plus tard.

Une manière d'augmenter la séquence de 0 (et donc de gagner en compression) est de modifier l'ordre de codage de la matrice. En effet si on utilise une séquence zigzag on augmente considérablement le nombre de 0 qui se suivent.

#### **VI.4.1.7 Le codage RLE :**

Comme le coefficient (1,1) est proportionnel à la valeur moyenne du bloc de pixels, il ne varie pas beaucoup d'un bloc à l'autre, on en code la différence par rapport au précédent (codage DPCM : Différentiel Pulse Code Modulation). Cela permet de stocker le coefficient DC dans une structure plus petite.

Le reste des coefficients subissent en un premier temps un codage de longueur de séquence (Run-Lengh-Encoding).

Le principe du codage RLE adopté par la norme JPEG consiste à parcourir la séquence en remplaçant chaque coefficient non nul par le couple (lengh, coef AC), où lengh représente le nombre de zéro qui précèdent cet AC.

Par conséquent, le RLE transforme la séquence des AC en un vecteur de couples (lengh, coefficient AC). La norme JPEG prévoit l'utilisation de deux symboles spéciaux, il s'agit du symbole ZRL (Zéro Lengh Encoding) et le symbole EOB (End Of Block).

Le symbole ZRL est introduit dans la séquence résultat lorsqu'en parcourant le vecteur des AC, une longueur égale à 15 zéros est rencontrée, quant au symbole EOB, il est inséré au moment où on détecte une suite de zéros jusqu'à la fin du vecteur.

#### VI.4.1.8 Codage de Huffman:

Nous allons à présent détailler le principe du codage de Huffman :

Chaque couple d'informations (longueur, coefficient de QDCT) issu du codeur RLE est transformé en un triplet (longueur, taille, coefficient de QDCT) où taille désigne le nombre de bits sur lesquels est représenté le coefficient QDCT. Un symbole correspond alors au couple (longueur, taille) pour les coefficients AC et à (taille) pour le coefficient DC.

Durant le processus du codage, ce symbole est utilisé comme une adresse dans la table de Huffman pour récupérer le mot de code correspondant aux coefficients DC et AC ces derniers étant différents, deux types de tables de Huffman sont utilisés, l'une pour les coefficients AC et l'autre pour les coefficients DC.

Il est important de préciser ici que, le nombre d'entrées dans la table des DC est égal à 12 du fait que la taille DC varie entre 0 et 11

(1Bit pour la valeur de la différence DC lorsque celle-ci est nulle et 11 bits, lorsqu'elle atteint Les valeurs  $-2047$  ou  $2047$ ).

Les 160 entrées dans des AC correspondent aux valeurs que peut prendre le couple (length, taille). Sachant que la taille varie entre 1 et 10 (car les valeurs AC varie entre  $-1023$  et  $1023$ ) de plus que le nombre des zéros varie entre 0 et 15. Les deux autres entrées correspondent aux deux symboles spéciaux : ZRL et EOB.

La séquence des couples ( mot de code, coefficient QDCT) issue du codage de Huffman présente l'image compressée qui sera transmise.

Il convient ici de mentionner une particularité très importante des algorithmes statistiques de base :

Puisque les codes séparables ne sont pas connus d'office, le décompacteur doit posséder la table des fréquences pour pouvoir les calculer et retrouver ainsi les octets du fichier d'origine. Cette table est en général sauvegardée en début de fichier compacté, et s'appelle (en-tête de l'algorithme statistique).

Taille	Valeurs de différences
0	0
1	-1,1
2	-3...-2,2...3
3	-7...-4,4...7
4	-15...-8,8...15
5	-31...-16,16...31
6	-63...-32,32...63
7	-127...-64,64...127
8	-255...-128,128...255
9	-511...-256,256...511
10	-1023...-512,512...1023
11	-2047...-1024,1024...2047

Table IV.1 : « La taille des symboles DC »

Catégorie	Amplitude
1	-1...0,0...1
2	_3...-2,2...3
3	-7...-4,4...7
4	-15...-8,8...15
5	-31...-16,16...31
6	-63...-32,32...63
7	-127...-64,64...127
8	-255...-128,128...255
9	-511...-256,256...511
10	-1023...-512,512...1023

Table IV.2 : « La taille des symboles AC »

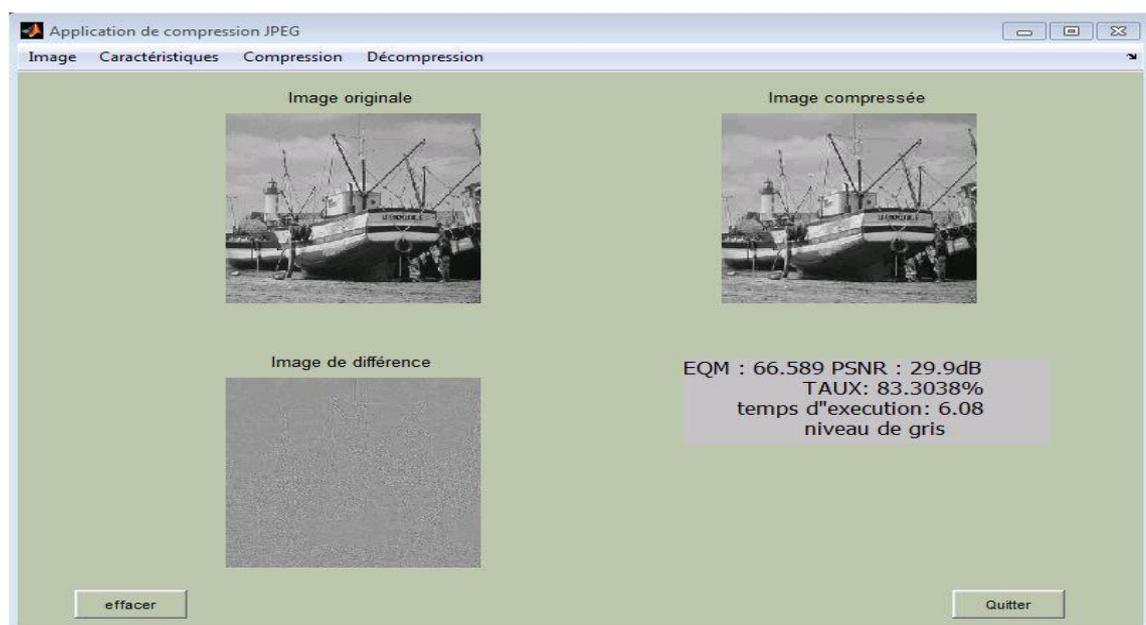
## VI.5 Simulation :

Pour notre simulation on utilisé trois type des images, image niveaux de gris, image couleur et image YCrCb et on a appliqué la compression par la norme JPEG après on a calculé EQM ,PSNR et le taux de compression pour différentes valeurs de facteur de qualité (voir les résultats ci-après)

### ➤ Image à niveau de gris :

Facteur de qualité	25%	50%	75%	100%
EQM	<b>66.589</b>	<b>32.945</b>	<b>14.858</b>	<b>0.00045776</b>
PSNR	<b>29.9</b>	<b>32.95</b>	<b>36.41</b>	<b>81.52</b>
Taux de compression(%)	<b>83.3038</b>	<b>80.2414</b>	<b>75.676</b>	<b>27.6062</b>
Temps d'exécution	<b>6.08</b>	<b>7.78</b>	<b>10</b>	<b>25.4</b>

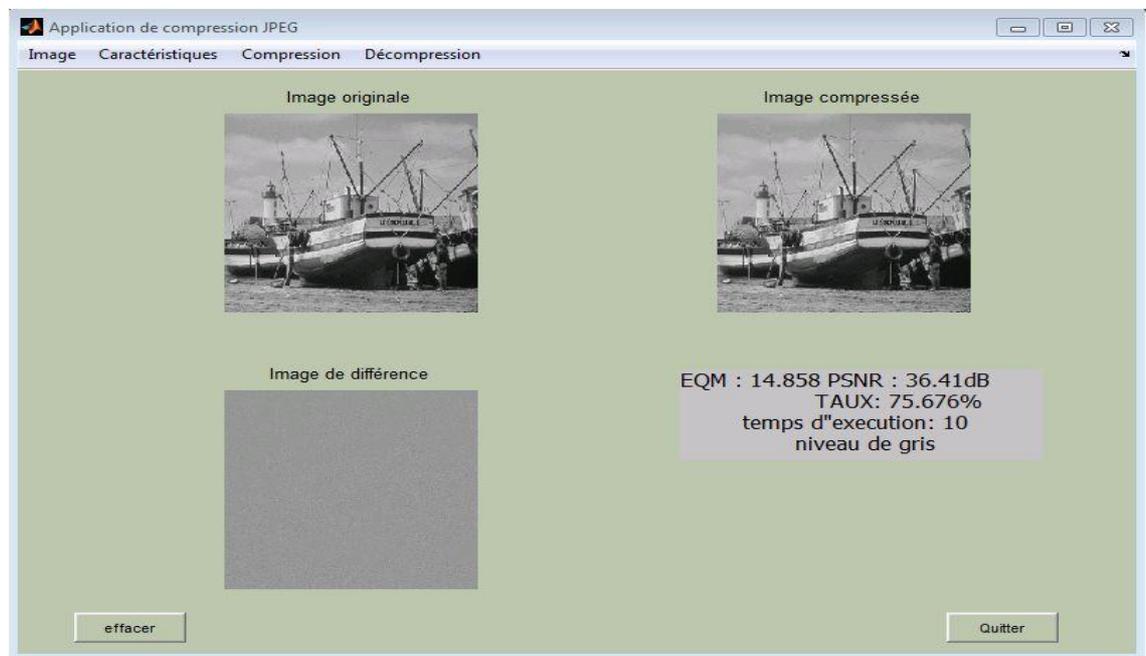
**Tab VI-3** : Les résultats de la compression d'une image Niveaux de gris en fonction du facteur de qualité



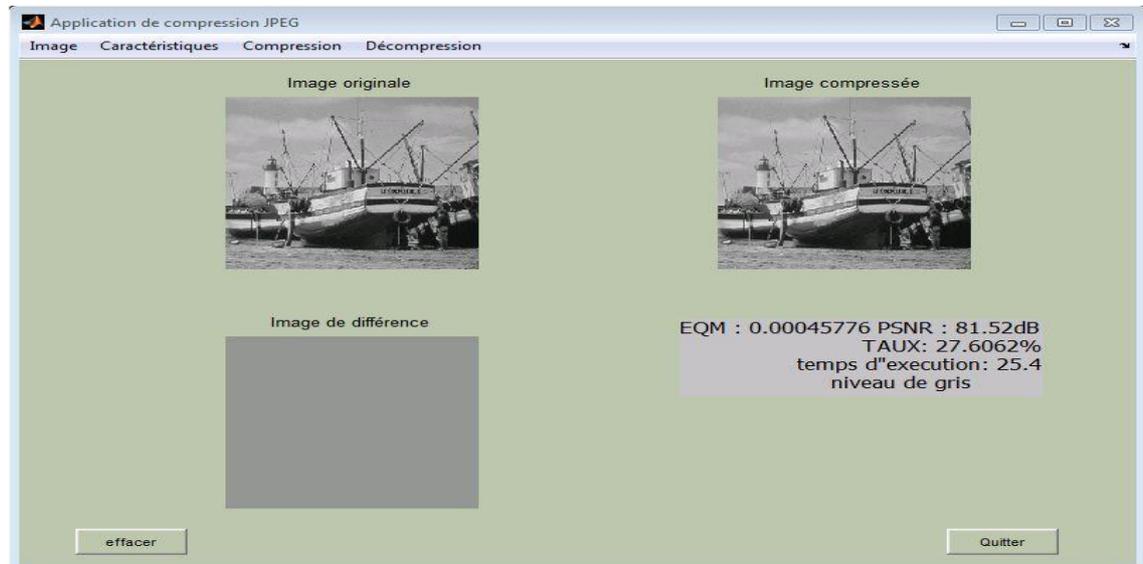
**Figure VI-7** : Compression d'une image Niveaux de gris avec un facteur de qualité (25%)



**Figure VI-8 :** Compression d'une image Niveaux de gris avec un facteur de qualité (50%)



**Figure VI-9 :** Compression d'une image Niveaux de gris avec un facteur de qualité (75%)

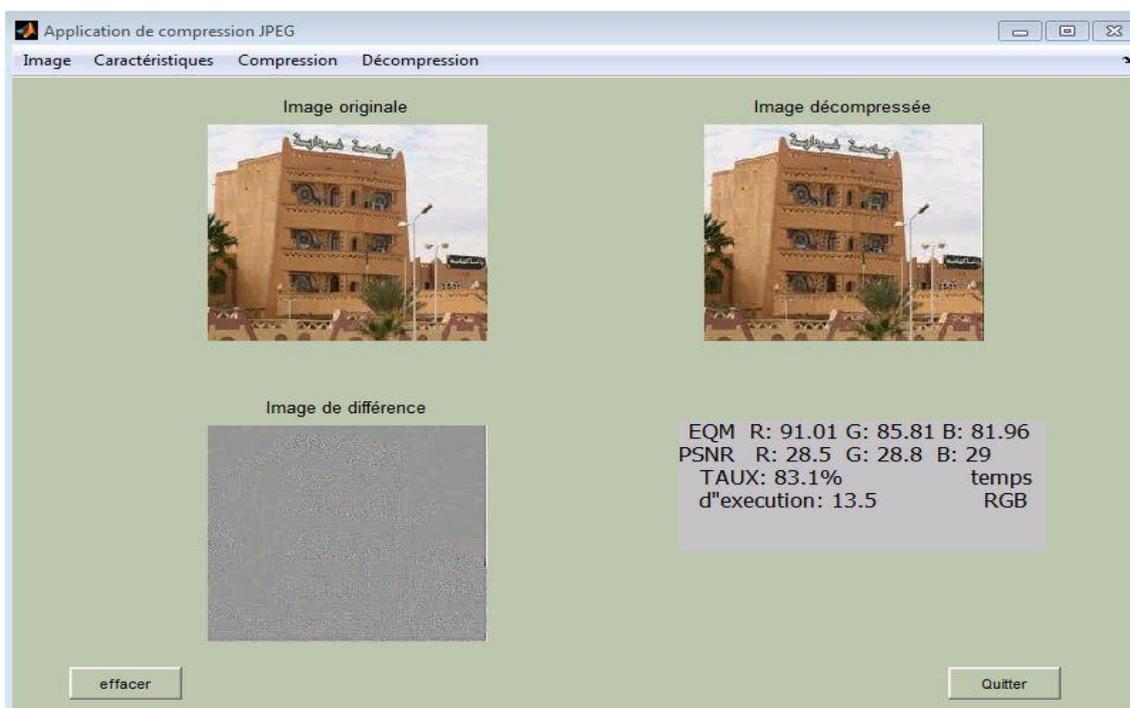


**Figure VI-10 :** Compression d'une image Niveaux de gris avec un facteur de qualité (100%)

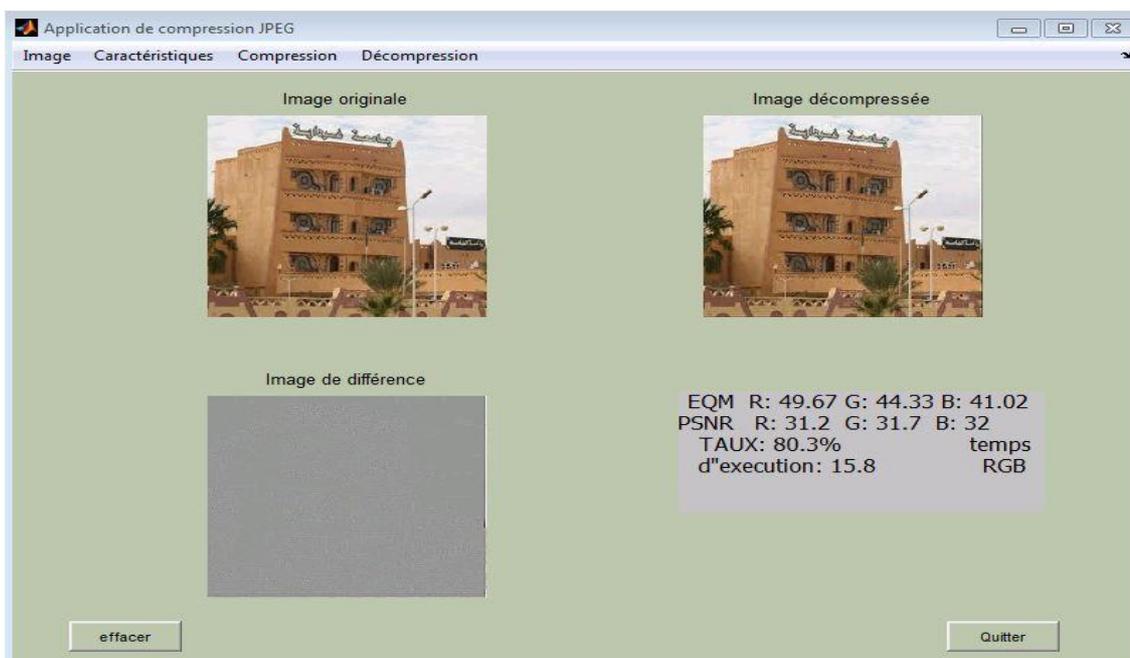
➤ **Image couleur RGB :**

Facteur de qualité	25%	50%	75%	100%
EQM(R)	91.01	49.67	41.71	39.24
EQM(G)	85.81	44.33	36.37	34.15
EQM(B)	81.96	41.02	33.02	48.27
PSNR(R)	28.5	31.2	31.9	32.2
PSNR(G)	28.8	31.7	32.5	34.15
PSNR(B)	29	32	32.9	30.31
Taux de compression(%)	83.1	80.3	77.2	52
Temps d'exécution	13.5	15.8	17	35.8

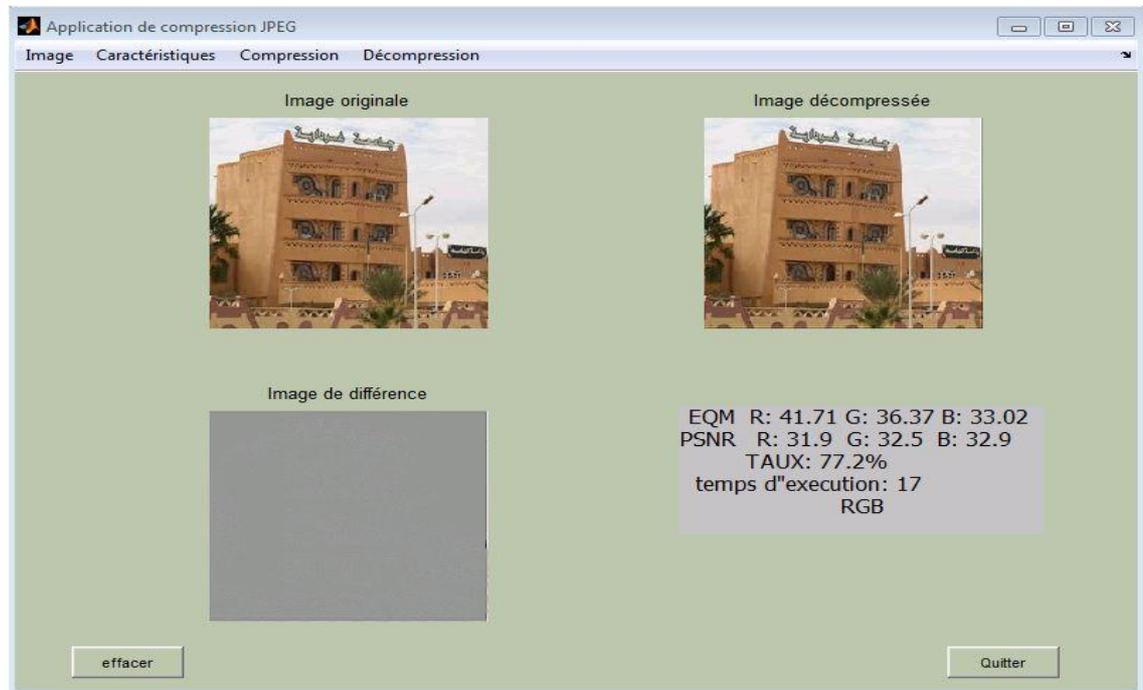
**Tab VI-4 :** Les résultats de la compression d'une image Couleur RGB en fonction du facteur de qualité



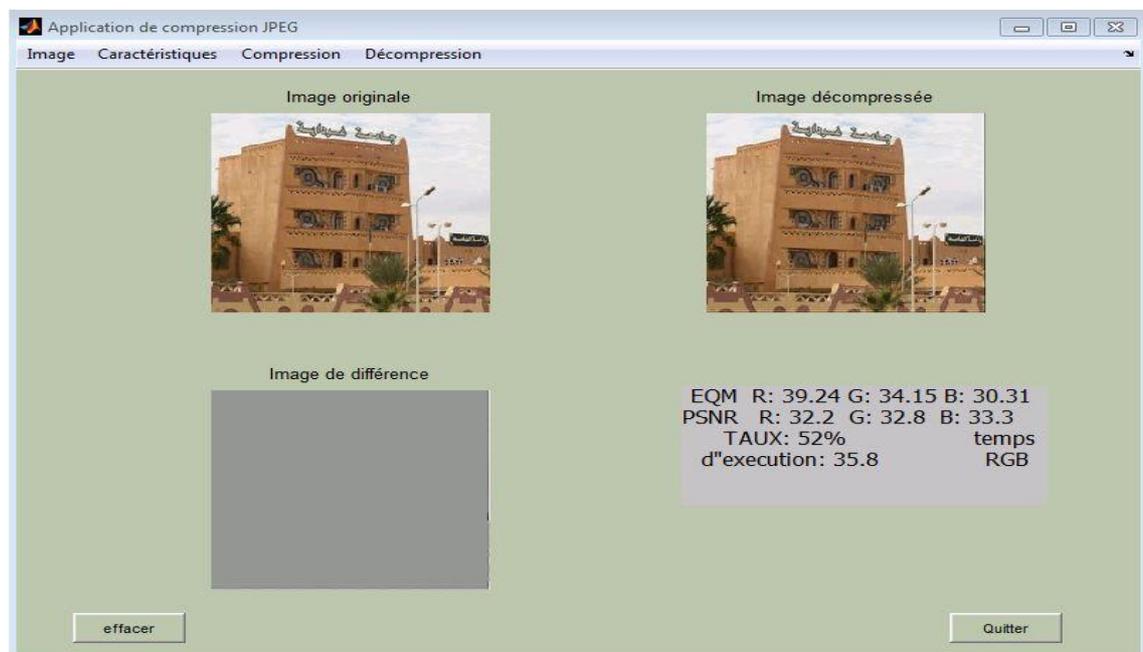
**Figure VI-11:** Compression d'une image couleur RGB avec un facteur de qualité (25%)



**Figure VI-12 :** Compression d'une image couleur RGB avec un facteur de qualité (50%)



**Figure VI-13 :** Compression d'une image Couleur RGB avec un facteur de qualité (75%)

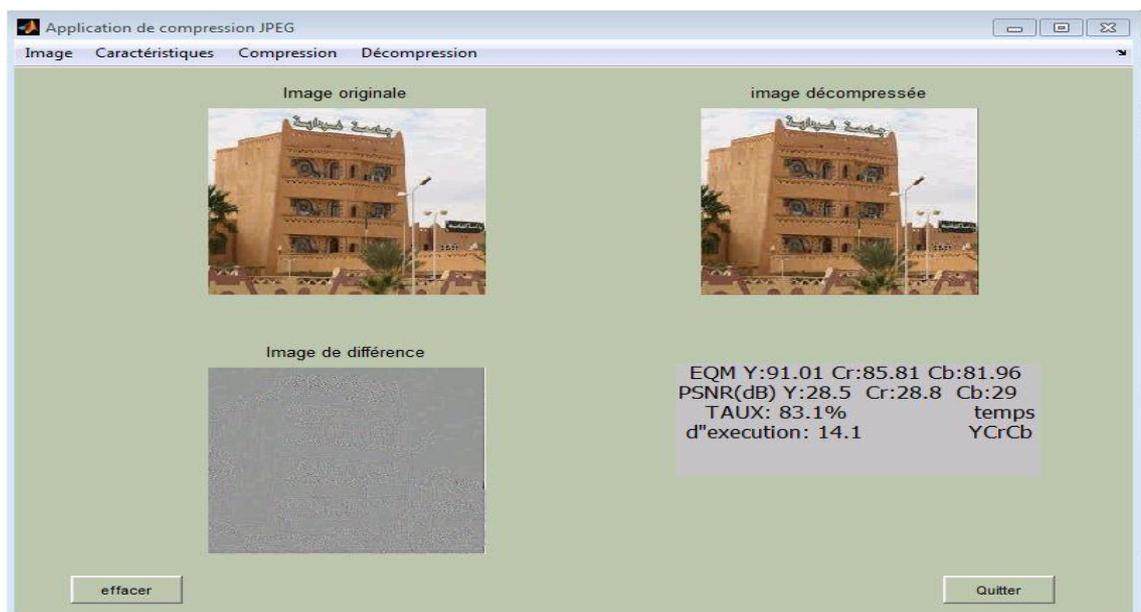


**Figure VI-14 :** Compression d'une image Couleur RGB avec un facteur de qualité (100%)

➤ **Image couleur YCrCb :**

➤ Facteur de qualité	25%	50%	75%	100%
EQM(Y)	91.01	49.67	41.71	39.24
EQM(Cr)	85.81	44.33	36.37	34.15
EQM(Cb)	81.96	41.02	33.02	48.27
PSNR(Y)	28.5	31.2	31.9	32.2
PSNR(Cr)	28.8	31.7	32.5	34.15
PSNR(Cb)	29	32	32.9	30.31
Taux de compression(%)	83.1	80.3	77.2	52
Temps d'exécution	13.5	15.8	17	35.9

**Tab VI-5 :** Les résultats de la compression d'une image en couleur YCbCr en fonction du facteur de qualité



**Figure VI-15 :** Compression d'une image Couleur YCbCr avec un facteur de qualité (25%)

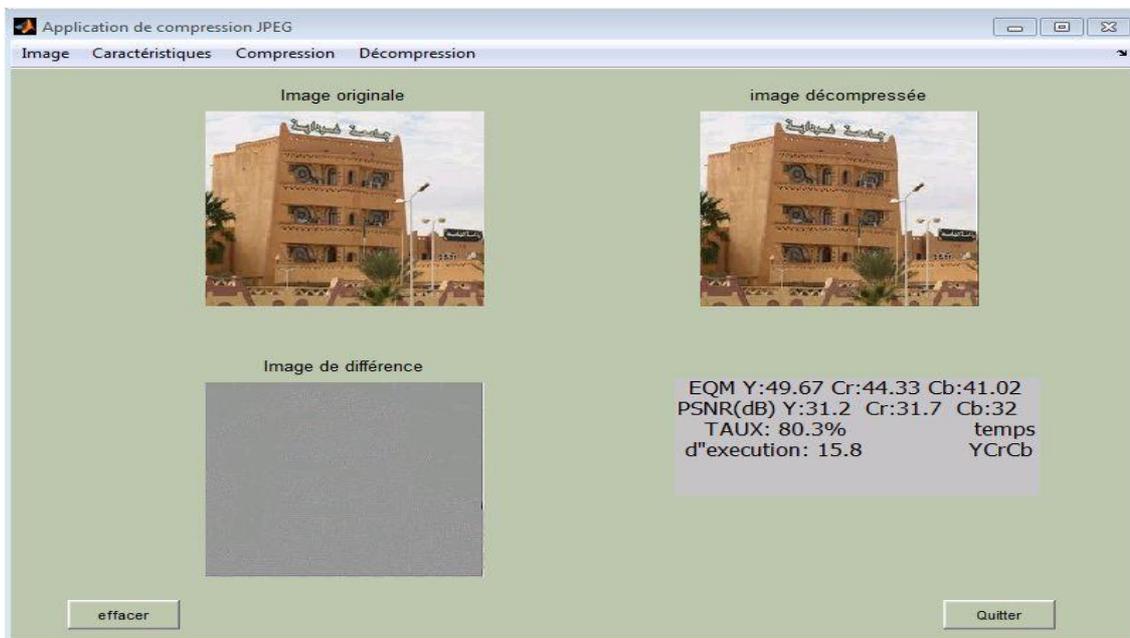


Figure VI-16 : Compression d'une image Couleur YCbCr avec un facteur de qualité (50%)

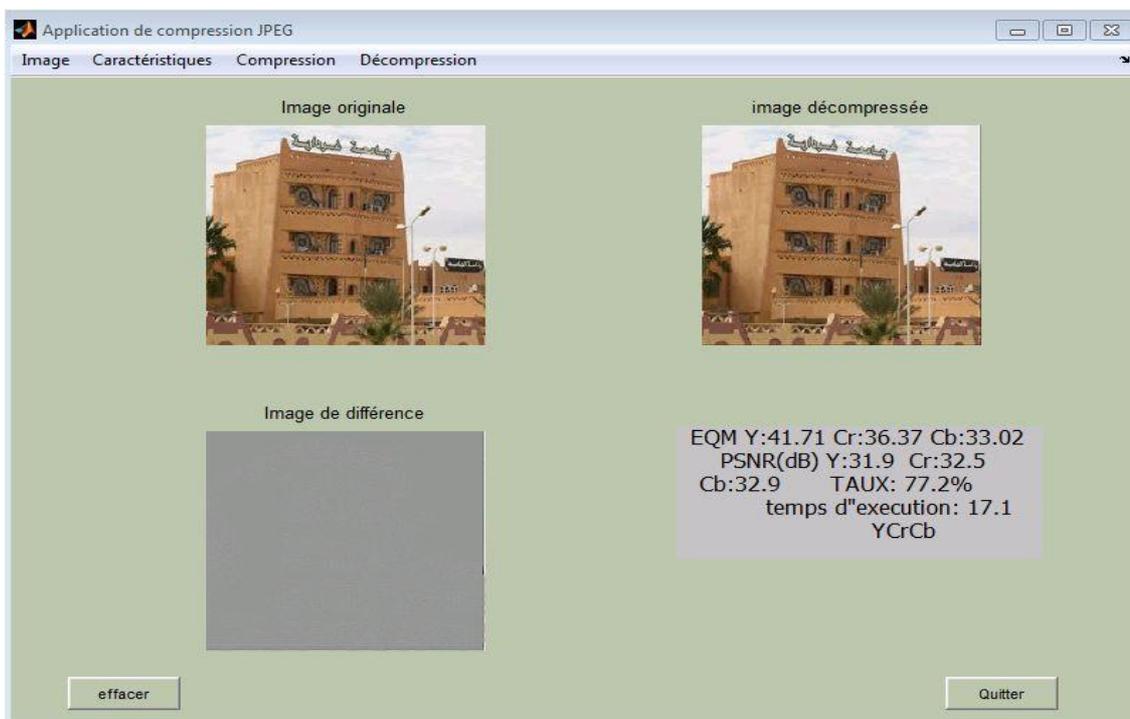
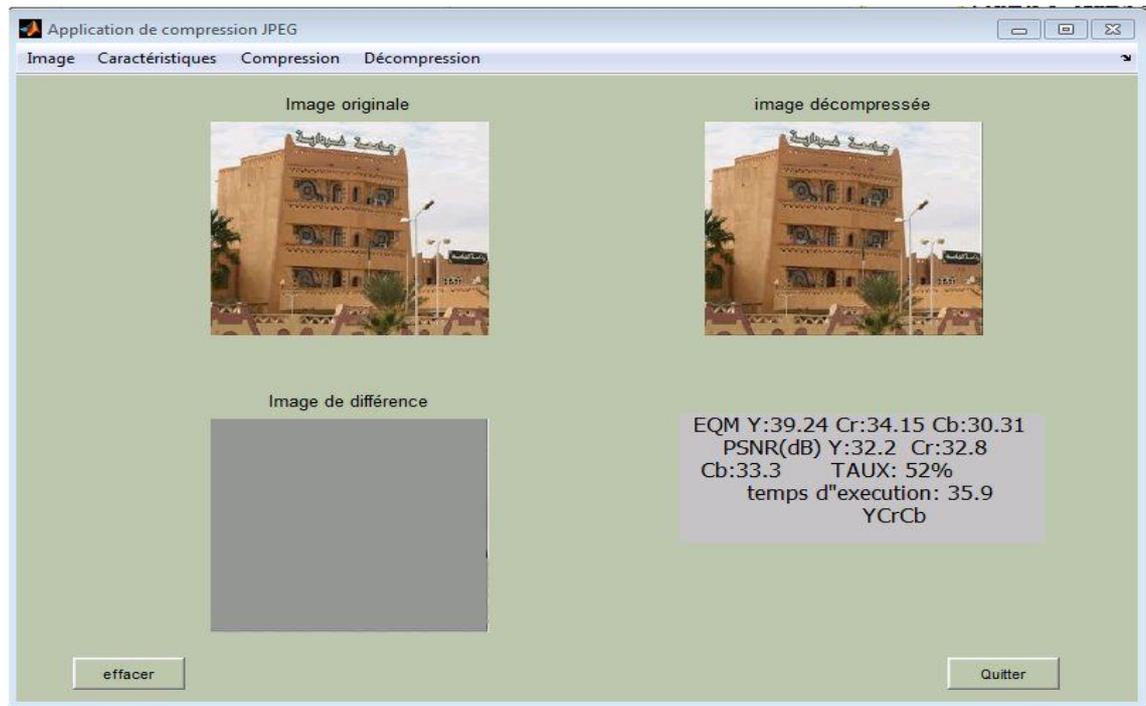


Figure VI-17 : Compression d'une image Couleur YCbCr avec un facteur de qualité (75%)



**Figure VI-18 :** Compression d'une image Couleur YCbCr avec un facteur de qualité (100%)

## VI.6 Discussion les résultats

La compression d'image avec perte utilisant les codecs JPEG entraîne une dégradation de la qualité de l'image à cause de la quantification subie. En général, plus le taux de compression est élevé, plus l'image décodée est distordue. Les tableaux (VI. 3, 4 et 5 montrent les résultats de compression d'une image en termes de taux de compression, de PSNR et de nombre de bits par pixels en utilisant les standards JPEG.

On peut constater que plus le facteur de qualité est élevé plus le PSNR ainsi élevé et l'erreur quadratique moyenne et le taux de compression est diminuent.

## VI.7 Conclusion

Dans ce chapitre de simulation, nous nous sommes intéressés à l'étude, l'implémentation et l'évaluation de l'algorithme JPEG. Nous variant le facteur de qualité entre 25% et 100%, et comparé entre l'image originale et l'image compressé en fonction de taux de compression et l'erreur quadratique moyenne (EQM) et le PSNR.

## CONCLUSION GENERALE

La compression des données et particulièrement des images est toujours d'actualité compte des exigences techniques et technologiques liées aux multimédias d'une manière générale et aussi aux moyens de communications comme les réseaux informatiques. De nombreux algorithmes existent actuellement, certains sont devenus des normes et des standards. Les techniques de compression avec pertes assurent généralement des taux de compression relativement élevés mais avec un certain degré de dégradation. Les objectifs des nouvelles techniques de compression est donc de réduire plus et d'altérer les moins possible. Parmi les techniques de compression avec perte nous avons vu les méthodes basées sur les transformations comme la DCT et la DWT.

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.

Dans notre simulation nous avons choisi la méthode non conservatrice est très utilisée 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. Malgré tous les avantages fournis par la norme JPEG ; L'avenir va être pour les méthodes émergentes telles que les Fractales et Ondelettes.

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 indubitablement une avancée dans la compression d'images et de fichiers.

- [01] Cours TS Traitement et analyse d'images -ANIMAG (Institute national des sciences appliquées de lyon)
- [02] Traitement numérique de l'image Raphaël Isdant - 2009
- [03] Résolution de problèmes par ecosystems (Application au traitement d'images), présenté par: M<sup>me</sup>.Boukhoulouf Djemaa (17/11/2005)
- [04] Mémoire de fin d'études pour l'obtention du diplôme de Master en informatique  
Thème (Développement d'un Système de Dénombrement de Cellules Sanguines)  
Réalisé par : Mme Bouklab Radja Université Abou Bekr Belkaid
- [05] L'image numérique (Ecole d'Ingénieurs du Pas-de-Calais)
- [06] Transmettre et stocker de l'information, Auteur ( [Christophe.alleau@ac-poitiers.fr](mailto:Christophe.alleau@ac-poitiers.fr))([http://media.education.gouv.fr/file/special\\_8\\_men/99/0/physique\\_chimie\\_S\\_195990.pdf](http://media.education.gouv.fr/file/special_8_men/99/0/physique_chimie_S_195990.pdf))
- [07] Mémoire Master Professionnel, Segmentation d'image par Cooperation région-contours, présenté par: Serra BENFRIHA et Asma HAMEL
- [08] Mémoire de Licence en Informatique, Development d'une application de traitement d'images, présenté par: Sabri akram et Benali Mustafa
- [09] STAGE TECHNIQUE INTERNATIONAL D'ARCHIVES, THEORIES ET PRATIQUES ARCHIVISTIQUES  
AVRIL 2010, CARACTERISTIQUES TECHNIQUES DES IMAGES NUMERIQUES
- [10] Mémoire de MAGISTER, Compression des images numériques par la technique des ondelettes Présenté par: SAHIR MOURAD (19 / 06 / 2011)
- [11] WWW.Wikipédia.COM
- [12] Cours 5 Rehaussement des images –Christophe BLANC
- [13] Mémoire de Master Académique, Segmentation pour la modélisation et la compression d'image (Simulation sous Scilab) , présenté par: Benkadari kheira (17/09/2014)

- [14] Mémoire Master Professionnel, Traitement d'image et morphologie mathématique ,  
Présenté par: Sebaa Brahim / Belmesmar Djamilia (01/06/2016)
- [15] TI – Traitement d'Images Semaine 12: Compression d'images Olivier Losson, Master  
Informatique: <http://www.fil.univ-lille1.fr> ,Spécialité IVI : <http://master-ivi.univ-lille1.fr>
- [16][http://www.ulb.ac.be/cours/acohen/travaux\\_2006\\_infodoc/CompressionNumerique/SansPerteRLE.htm](http://www.ulb.ac.be/cours/acohen/travaux_2006_infodoc/CompressionNumerique/SansPerteRLE.htm)
- [17] [http://igm.univ-mlv.fr/~dr/XPOSE2013/La\\_compression\\_de\\_donnees/arithm.html](http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/arithm.html)
- [18] <http://www.commentcamarche.com/contents/1195-compression-de-donnees>
- [19][https://www.google.dz/url?sa=t&source=web&rct=j&url=http://projet.eu.org/pedago/sin/term/5huffman.pdf&ved=2ahUKEwjOkqLOsN\\_ZAhWDORQKHbYLB6wQFjAAegQICBAB&usg=AOvVaw3TYKYnJi5RIWu0s1dG0uIb](https://www.google.dz/url?sa=t&source=web&rct=j&url=http://projet.eu.org/pedago/sin/term/5huffman.pdf&ved=2ahUKEwjOkqLOsN_ZAhWDORQKHbYLB6wQFjAAegQICBAB&usg=AOvVaw3TYKYnJi5RIWu0s1dG0uIb)
- [20] Une introduction à la compression d'images médicales volumiques, Jonathan Taquet  
— Claude Labit, INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET  
EN AUTOMATIQUE (INRIA), N° 7324 Juin 2010
- [21] <http://www.imagenumerique.50webs.com/>
- [22] Mémoire pour l'obtention du diplôme de MAGISTER "Development et  
implementation des algorithmes de compression d'images base sur des transformées  
entières " par M<sup>elle</sup>: NABILA BRAHIMI, Université FARHAT ABBAS-SETIF,  
13/01/2011
- [23] Article "COMPRESSION D'IMAGE "M. K. Kholadi and M. C. Batouche Equipe  
PRAI, Institut d'Informatique, University of Constantine, Route de Ain El-Bey,  
Constantine 25000, Algeria. Tel & Fax : 213 4 92 28 35 / 213 4 92 51 20
- [24] Les formats de compression d'image Réalisé par (Mr. Brault et Mme. Dougherty )  
Institut Universitaire de Technologie de Tours , Département Génie Électrique et  
Informatique Industrielle