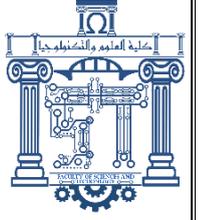




الجمهورية الجزائرية  
الديمقراطية  
الشعبية



République Algérienne Démocratique Et Populaire

وزارة التعليم العالي والبحث  
العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة العربي  
التبسي - تبسة

Université Larbi Tébessi- Tébessa –

Faculté des Sciences et de la Technologie

Département de génie électrique

## MEMOIRE

Présenté pour l'obtention du **diplôme** de **Master Académique**

**Spécialité : Réseaux et Télécommunications**

Par :

DEBAB Ilhem

DAHMANE Meriem

Sujet

# IMPLEMENTATION ET ANALYSE DE LA COMPRESSION JPEG SOUS L'ENVIRONNEMENT MATLAB

Présenté et soutenu publiquement, le 09/06 /2022 , devant le jury composé de :

Pr. MERAOUIMIA Abdallah  
Dr. BENTAHAR Tarek  
Dr. FERROUDJI Karim

Pr  
MCA  
MCB

Président  
Rapporteur  
Examineur

Promotion : 2021/2022

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

### Résumé :

L'image joue un rôle très important dans la technologie de l'information et de la télécommunication (TIC) actuelle. C'est l'élément unitaire des vidéos, et par conséquent l'élément indispensable dans toutes les formes de communications moderne tels que les vidéo conférences et les systèmes de reconnaissances. Toutes ces applications comportent sans doute le traitement d'image.

Malheureusement les supports de transmission, de stockage et les unités de traitement ont des capacités limitées. D'autre part, la qualité de service de nos jours exige une qualité d'image exceptionnelle et de grande taille. Cela conduit à une utilisation incontournable des techniques de compression et de codage afin d'optimiser les ressources numériques.

Notre travail consiste à implémenter l'une des techniques répandues pour ce but, qui est le JPEG. Nous visons principalement à programmer dans l'environnement MATLAB les étapes clés de JPEG y compris la transformation DCT, la mise en Zig-Zag, la Quantification, codage RLE, conversion binaire/ codage entropique et les étapes inverse correspondantes.

A fin d'évaluer la déviation avant et après la compression, le JPEG implémenté est analysé selon deux critères importants : SSIM , MSE, et sous les deux angles :

- Faire varier le point de coupure.
- Faire varier le pas de quantification

### Abstract:

The image plays a very important role in the latest information and telecommunications technologies (ICT). It is the unitary element of videos, and therefore the essential element in all forms of modern communication such as video conferences and recognition systems.

All these applications undoubtedly include image processing. Unfortunately, transmission, storage and processing units' capabilities are limited. On the other hand, today's quality of service demands exceptional and large picture quality.

This causes an unavoidable use of compression and coding techniques in order to optimize digital resources. Our work consists in implementing one of the most widely used techniques adopted for this purpose, which is the JPEG. We mainly aim to program in the MATLAB environment the key steps of JPEG including DCT transformation, Zig-Zag , Quantization, RLE coding, Binary conversion/entropy coding and the corresponding reverse steps.

In order to evaluate the deviation before and after compression, the implemented JPEG is analyzed according to two important criteria: SSIM, MSE, and from both angles:

- Varying the cutoff point.
- Varying the quantization step

### الملخص

تلعب الصورة دورا مهما للغاية اليوم في تكنولوجيا المعلومات والاتصالات (ICT) إنه المكون المهم لمقاطع الفيديو ، وبالتالي فهو المكون الأساسي لجميع أشكال الاتصال الحديثة مثل مؤتمرات الفيديو وأنظمة الاتصالات. كل هذه التطبيقات تتضمن بلا شك معالجة الصور.

لسوء الحظ ، فإن وحدات النقل والتخزين والمعالجة لها قدرات محدودة. من ناحية أخرى ، تتطلب جودة الخدمة الحالية جودة صورته ممتازة واستثنائية. وهذا يؤدي إلى الاستخدام الحتمي لتقنيات الضغط والتشفير لتحسين الموارد الرقمية.

مهمتنا هي تنفيذ إحدى التقنيات الشائعة لهذا الغرض ، وهي JPEG نهدف بشكل أساسي إلى برمجة الخطوات الرئيسية لـ JPEG في بيئة MATLAB ، بما في ذلك DCT ، وتحويل Zig Zag- ، والكمية ، وتشفير RLE ، والتحويل الثنائي والخطوات العكسية المقابلة.

لتقييم الانحراف قبل الضغط وبعده ، يتم تحليل JPEG المنفذ وفقا لمعيارين مهمين SSIM و MSE ومن كلا الزاويتين:

- تغيير نقطة القطع.
- تعديل خطوة التكميم

### **Remerciements :**

Au seuil de ce travail,

Nous remercions notre dieu ALLAH le tout puissant pour toute la volonté et le courage qu'il nous a donné pour l'achèvement de cette thèse, il a été et sera toujours à côté de nous pour réussir terminer n'importe quel travail.

La présentation de ce modeste travail nous offrons l'occasion d'exprimer notre profonde gratitude à Monsieur Dr. BENTAHER Tarek, qui a bien voulu diriger ce travail pendant toute la durée de l'expérimentation et la mise en forme du document final. Ses nombreux conseils ne nous ont jamais fait défaut. Nous sommes heureux de lui exprimer ici notre respectueuse reconnaissance.

Nous remercions également les membres de jury, Pr.MERAOUZIA Abdallah et Dr.FERROUJJI Karim pour l'intérêt et l'attention qu'ils ont accordés à ce modeste travail et pour avoir acceptés de faire partie de ce jury. Nous tenons à vous remercier pour chaque information ou conseil que vous nous avez donné durant notre parcours universitaire.

Nos remerciements sont également adressées à tous personne qui, de loin ou de près, ont contribué à la réalisation de ce travail de recherche, Nous tenons à remercions le chef de département et tous les enseignants.

أول من يشكر ويحمد اناء الليل وأطراف النهار هو العلي القهار الأول والأخر والظاهر والباطن الذي أعرقنا بنعمه التي لا تعد ولا تحصى فله جزيل الحمد والثناء العظيم وصلى الله على نبينا محمد عليه

### أفضل الصلاة وأزكى التسليم

اهدي هذي العمل إلى من علمني أن الدنيا كفاح وسلاحها العلم والمعرفة إلى الذي لم يبخل على بأي شيء إلى من سعى لأجل راحتي ونجاحي إلى أعظم واعز رجل في الكون أبي العزيز...

إلى من ساندتني في صلاتها ودعائها إلى من سهرت الليالي تنير دربي إلى من تشاركني أفراحي وأحزاني إلى نبع العطف والحنان إلى أجمل ابتسامة في حياتي إلى أروع امرأة في الوجود أُمي الغالية...

إلى توأم روحي ورفيقة حياتي إلى أختي وحبيبة قلبي أية..

إلى إخوتي وأعزائي ونور حياتي خير الدين ومعاذ...

إلى الذي لو كان بيننا اليوم لكان اسعد من في الكون جدي حبيبي علي دباب (رحمه الله)

وإلى جدي لخضر لشرح حفظه الله

وإلى جدتي حميدة وجدتي يمينه

وإلى خالاتي مليكة، ليلي، فطيمة، جلييلة وعمتي ليلي

إلى كل بناتهم وأولادهم

إلى أعمامي وأخوالي

إلى الأستاذ المشرف بن طاهر طارق الذي كان له كل الفضل في انجاز هذا العمل

إلى رفيقتي في البحث وفي الحياة وشريكة أفكارني وحظي الجميل في الدنيا مريم دحمان

إلى أختي والأخذة بيدي إلى كل خير صفاء شرقي

إلى زملائي في كل مشواري الدراسي وإلى صديقاتي نرجس، رانية، سهام، بشرى، شمس الهدى وإلى كل من ساندني ودعمني وكان عون لي واخص بالذكر من كانت لي أخت وصديقة حميدة جفال،

وأستاذتي وقدوتي الجميلة بن خذير فيروز والأستاذ رمزي صغير (رحمه الله) والأستاذة حنان هنين

والأستاذ فارس حكيم

والشكر موصل إلى كل معلم وأستاذ أفادنا بعلمه من أولى المراحل الدراسية إلى هذه اللحظة وأخص بالذكر الأستاذة غلوسي لبنة، الأستاذة بن شلوق مليكة، الأستاذة بوراس رتيبة، الأستاذة زياني هناء الأستاذ

كعبي التوفيق، الأستاذة صامت صلاح الدين، الأستاذة رايس حمودة

## الهام

## *Dédicace*

### **Dédicace :**

Je dédie ce travail

A Ma mère qui m'a entouré d'amour, d'affection et qui fait tout pour  
ma réussite, que dieu la garde

A Mon père qui m'a aidé à devenir ce que je suis aujourd'hui, que  
dieu le garde et le protège

A mes chères sœurs, **Takoua, Ines, Riheb**, pour leurs encouragements permanents, et leur  
soutien moral.

A mon cher frère **Abdelkedous**, pour son appui et leur encouragement.

A ma petite chère nièce, la plus belle princesse **Bayane**.

A toutes mes amies, et spécialement a la meilleur binôme au monde **Ilhem**.

A toute ma famille paternelle et maternelle pour leur soutien tout au long de mon  
parcours universitaire.

Sans oublier tout les professeurs que ce soit du primaire, du moyen, du secondaire ou de  
l'enseignement du supérieur et spécialement **Ms.BENTAHAR Tarek**.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien  
infaillible.

Merci d'être toujours là pour moi.

**Meriem**

# Liste des matières

Introduction générale ..... 1

**Chapitre 01 : Image Numérique**

I.1 Introduction:..... 3

I.2 Définition d'une image: ..... 3

I.3 Image numérique: ..... 3

    I.3.1 Echantillonnage (Résolutions spatiale) : ..... 4

    I.3.2 Quantification (Résolution tonale):..... 5

I.4 Caractéristiques d'une image numérique:..... 6

    I.4.1 Dimension (Définition): ..... 6

    I.4.2 Résolution :..... 6

    I.4.3 Histogramme : ..... 6

    I.4.4 Luminance:..... 7

    I.4.5 Contraste: ..... 7

    I.4.6 Profondeur:..... 7

    I.4.7 Le poids de l'image:..... 7

    I.4.8 Contours et textures:..... 7

I.5 Différents types d'images : ..... 7

    I.5.1 Image monochrome (Binaire): ..... 8

    I.5.2 Image en niveaux de gris :..... 8

    I.5.3 Image en couleurs: ..... 9

I.6 Formats standards d image : ..... 11

    I.6.1 Images vectorielles:..... 11

    I.6.2 Image matricielle:..... 12

I.7 Généralités sur les traitements d'image:..... 14

    I.7.1 Définition: ..... 14

    I.7.2 Domaines d'applications: ..... 14

    I.7.3 Principales techniques: ..... 15

    I.7.4 Principales applications:..... 16

I.8 Conclusion:..... 16

**Chapitre 02 : Théorie De L'information**

II.1 Introduction : ..... 17

II.2 Principes de la théorie de l'information : ..... 17

II.2 .1 Définition de l'information :	17
II.2 .2 Définition de la théorie de l'information :	17
II.2 .3 Quantité d'information d'un symbole :	18
II.2 .4 Quantité d'information moyenne (Entropie de Shannon) :	18
II.2 .5 Entropies conditionnelle et conjointe :	19
II.3 Codage de source :	20
II.3.1 Types de sources :	21
II.3.2 Types de codage de source :	21
II.3.3 Classification des codes :	22
II.4 Codage de canal :	28
II.4 .1 Canal de transmission :	28
II.4 .2 Caractéristiques d'un canal:	29
II.4.3 Codage de canal :	29
II.4.4 Principaux Types des canaux :	30
II.5 Taux de codage canal et probabilités d'erreur :	32
II.6 Capacité du canal.....	33
II.7 Conclusion :	33

### ***Chapitre 03 : Technique De Compression***

III.1 Introduction :	34
III.2 But de la compression d'image :	34
III.3 Mesures de performance de la compression d'image :	34
III.3.1 Taux de compression :	34
III.3.2 Taux d'information (Entropie) :	35
III.3.3 Mesures de distorsion (Mesures de fidélité) :	35
III.4 Principe général de la compression d'images :	37
III.5 Classification des méthodes de compression :	38
III.5.1 Méthode avec ou sans perte d'information :	38
III.5.2 Méthodes par pixels, bloc de pixels, ou image entière (scène) :	38
III.5.3 Méthodes Intra et Inter-images :	39
III.5.4 Méthodes spatiales et méthodes par transformation.....	39
III.5.5 Méthodes adaptatives, non adaptatives :	40
III.6 Méthodes réversibles ou sans perte :	40
III.6.1 Méthodes différentielles et prédictives :	41

III.6.2 Méthodes par plages (Run length Encoding) : .....	41
III.6.3 Méthodes par dictionnaire adaptatif (Lempel-Ziv) : .....	43
III.6.4 Codage arithmétique : .....	43
III.7. Méthodes avec perte ou irréversible : .....	45
III.7. 1 Méthodes spatiales : .....	45
III.7. 2 Méthodes par transformation : .....	47
III.8 Normes de compression des images : .....	53
III.8.1 Norme de compression JPEG : .....	53
III.8.2. Norme de compression JPEG 2000: .....	57
III.9 Conclusion : .....	58

### ***Chapitre 04 : Implémentation Et Analyse***

IV.1 Introduction: .....	59
IV.2 Image de test : .....	59
IV.3 Les étapes d'implémentation : .....	61
IV-4 Résultats et Discussion : .....	64
IV.5 Conclusion : .....	74

# Liste des tableaux

## *Liste des tableaux*

Tableau II.1 : Exemple du code de longueur fixe et code de longueur variable.....	20
Tableau II.2 : Code de Huffman.....	24
Tableau II.3 : Exemple du codage de Shannon.....	25
Tableau III.1 : probabilités des symboles .....	44

# Liste des figures

Figure I.1: Élément d'une image (le pixel).....	3
Figure I .2 : Pixels. Images en noir et blanc (a) ou en niveaux de gris (b) .....	4
Figure I.3: Echantillonnage d'une image.....	4
Figure I. 4: Quantification.....	5
Figure I.5 : Résolution spatial: échantillonnage.....	5
Figure I.6 : Résolution tonale: quantification. ....	5
Figure I.7 : Image et histogramme associés.....	6
Figure I.8 : L'image monochrome (binaire). ....	8
Figure I.9 : Pixels et niveaux de gris. ....	9
Figure I .10 : L'image en niveaux de gris.....	9
Figure I.11 : Image en couleurs.....	10
Figure I.12: Image indexée. ....	11
Figure I.13: Image vectorielle.....	11
Figure I.14: Image matricielle.....	13
Figure II.1: Chaîne de traitement de l'information. ....	17
Figure II.2 : Etapes principales de codage sans perte.....	21
Figure II.3: Etapes principales de codage avec perte. ....	22
Figure II.4 : Arbre de Huffman. ....	24
Figure II.5 : Linéarisation en ligne, en colonne et en zig-zag.....	25
Figure II.6 : Arbre d'attribution des codes de Shannon.....	26
Figure II.7: Arbre de décomposition pour la construction de codes sans préfixe.....	26
Figure II.8 : Réduction de l'arbre de la figure .....	27
Figure II.9 : Modèle de canal de transmission .....	29
Figure II.10: Paradigme du codage de Canal. ....	30
Figure III.1: Etapes principales de compression d'images. ....	37
Figure III.2: Un exemple de codage par plage RLE.....	42
Figure III.3: Génération de l'étiquette pour la séquence "acaab". ....	44
Figure III.4: Schéma de principe de la compression / décompression par transformation.....	48
Figure III.5: Transformation en colonnes et en lignes.....	52
Figure III.6 : Décompression progressive.....	53
Figure III.7: Principe de l'algorithme JPEG avec perte. ....	54
Figure III.8: Le parcours d'un bloc en zigzag.....	56
Figure III.9: Exemple d'image compressée avec JPEG. ....	56

Figure III.10: Schéma typique d'un codeur JPEG 2000. ....	57
Figure III.11: Schéma du principe de la compression .....	58
Figure IV.1 : L'image de test et ses trois composantes. ....	60
Figure IV.2 : L'histogramme de l'image de test. ....	60
Figure IV.3: Schéma du principe de la compression/décompression JPEG. ....	61
Figure IV.4: DCT de l'image de test (composante rouge), zone de dix premier pixels. ....	62
Figure IV.5 : Parcours Zig-Zag .....	63
Figure IV.6 : Images décompressées obtenues pour les quatre points choisis. Composante Rouge.....	66
Figure IV.7 : Images décompressées obtenues pour les quatre points choisis. Composante Verte. ....	67
Figure IV.8 : Images décompressées obtenues pour les quatre points choisis. Composante Verte. ....	68
Figure IV.9 : MSE et SSIM en fonction de point de coupure/Composante Rouge. ....	69
Figure IV.10 : MSE et SSIM en fonction de point de coupure/Composante Verte. ....	70
Figure IV.11 : MSE et SSIM en fonction de point de coupure/Composante Bleu. ....	70
Figure IV.12 : MSE en fonction de point de coupure pour les trois composantes. ....	71
Figure IV.13 : Images décompressées obtenues pour les cinq pas choisis.Composante Rouge .....	72
Figure IV.14 : MSE et SSIM en fonction de pas de quantification/Composante Rouge. ....	73

# Liste des Acronymes

### Liste des acronymes :

- **A**
  - ARJ : Archived by Robert K. Jung
  - AMBTC : Absolute Moment Block Truncation Coding
  - ASCII : American Standard Code for Information Interchange
  - AWGN : Additive White Gaussian Noise
  - AC: Alternative Component
- **B**
  - BEC : Canal Binaire à Effacement
  - BMP : Windows Bitmap
  - BSC : Canal Binaire Symétrique
  - BTC : Block Truncation Coding
- **C**
  - CEI : Commission Electronique Internationale
  - CGM : Computer Graphics Metafile
- **D**
  - DCT : Discrète Cosine Transform
  - DPCM : Differential Pulse Code Modulation
  - DWT : Discrete Wavelet Transform
  - DXF : Data eXchange Format
- **E**
  - EPS : Postscript / Encapsulated Postscript
  - EZW : Embedded Zerotrees of Wavelet transforms
- **G**
  - GIF : Graphic Interchange Format
- **H**
  - HSL : Hue Saturation Lightness
  - HVS : Human Visual System
- **I**
  - IRM : Imagerie par résonance magnétique
  - ISO : International Standards Organisation
- **J**
  - JPEG : Joint Photographique Experts Group
- **L**
  - LHA : Lateral Hypothalamic Area
  - LZ : Lempel-Ziv
- **M**
  - MIT : Massachusetts Institute of Technology
  - MPEG : Motion Picture Expert Group
  - MSE : Mean Squared Error
- **P**
  - PCX : PiCture eXchange
  - PSNR : Peak Signal to Noise Ratio

## Liste des Acronymes

- R,G,B : Red, Green, Blue
- RLE : Run length Encodin
- S
  - SBBTC : Single Bitmap BTC
  - SDSM : Source Discrète Sans Mémoire
  - SNR : Signal to Noise Ratio
  - SQ : Scalar Quantization
- T
  - TEP : Tomographie par Emission de Positons
  - TFD : Transformation de Fourier Discrète
  - TH : Transformations de Hadamard
  - THA : Transformations de Haar
  - TIFF : Tag Image File Format
  - TKL : Transformation de Karhunen-Loeve
- V
  - VQ : Vector Quantization
- W
  - WMF : Windows Meta File

# Introduction Générale

### **Introduction générale :**

L'image est une information très importante, on en entend : “ une image vaut plus que mille mots “ .Vu l'importance de l'image, et la grande quantité d'information qu'elle peut contenir, le monde s'y intéresse de plus en plus et tends vers l'universalisation de son utilisation. En effet, l'image touche plusieurs domaines de notre vie : la médecine, la météo, la télécommunication, la cartographie, la géologie, etc.

La technologie numérique a facilité l'intégration des images dans nombreux domaines d'applications. Ce qui rend l'utilisation des images massive dans ces diverses applications. En effet, les supports de stockage et les réseaux de transport atteignent un point de saturation en raison de l'augmentation massive de la quantité de données chargées. Dans ce cas, afin d'économiser les ressources de stockage ou le multiplexage des canaux de communication, la compression semble être un outil important et inévitable pour maintenir la qualité de service fournit.

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

Parmi les techniques de compression nous trouvons principalement le JPEG qui est très répandue. Cette norme a connu un très grand succès dans le domaine de l'informatique et le domaine de la télécommunication et directement le domaine de technologie de l'information et de communication (TIC). En effet l'image aujourd'hui (même la vidéo) doit être vue est convenablement traité sous l'aspect mathématique, informatique et électronique.

Le JPEG comporte plusieurs techniques de codages, de transformations, de conversion, impliquées pour former une seule information compressée. Dans ce travail, nous nous sommes intéressées à implémenter les principales étapes de la compressions JPEG telles que: la transformation DCT, la mise en Zig-Zag, la quantification, codage RLE, conversion binaire/ codage entropique et les étapes inverse correspondantes.

Dans le premier chapitre nous présentons les notions de base de l'image et du traitement de l'information. Ces notions fondamentales et nécessaires sont passées en revue afin d'enchaîner d'une manière logique avec le chapitre concernant la compression.

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

Dans le deuxième chapitre, un rappel de la théorie de l'information est envisagé dans lequel nous résumons les idées sur l'information, la quantité d'information et l'entropie de Shannon.

## *Introduction Général*

Ainsi, nous avons un bref aperçu sur le codage du canal et le codage de source et leurs buts et leurs différents types. Nous -nous focalisons sur les points que nous considérons comme essentiels pour la bonne compréhension de la suite du mémoire.

Dans le troisième chapitre, nous introduisons plusieurs techniques, méthodes et algorithmes de compression d'image sans perte et avec perte (DCT, DWT, LZW, JPEG, JPEG 2000, etc.), ainsi que des concepts généraux tels que les mesures de performance et de distorsion. Nous avons donné quelques concepts et mesures pour la compression d'images.

Ce chapitre est divisé en trois parties principales. La première partie consiste à décrire les images de test utilisées dans ce travail, leurs caractéristiques telles que le type, la dimension, l'histogramme, etc. dans la seconde partie. Nous avons expliqué les principes de chaque technique impliquée dans le traitement JPEG. La troisième partie est consacrée à l'analyse de l'effet (perte) de la compression sous deux angles : sans et avec quantification. Les pertes qui en résultent sont analysées selon deux critères de biais : MSE et SSIM.

# Chapitre 1

## Image Numérique

## I.1 Introduction:

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

Dans ce chapitre, Nous avons présenté les connaissances de base sur les images, leurs caractéristiques, leurs types et leurs formats standards. Nous avons donné quelques concepts de l'image et les principales techniques de traitement d'image, et par la suite nous présenterons quelques applications de traitement d'image.

## I.2 Définition d'une image:

Une image est une représentation plane d'une scène ou d'un objet, généralement située dans un espace tridimensionnel. Il a été développé à partir d'un désir de fournir une entité qui pourrait être observée par l'œil humain. Cela explique son aspect planaire d'une part, et le fait que les informations de base liées à chaque point de l'image soient retranscrites en niveaux de gris ou en couleur d'autre part [1].

## I.3 Image numérique:

Une image numérique est un fichier, un code, acquis, stocké et interprété, reproductible à l'infini. Elle est fonction de l'espace nécessaire pour la stocker, de son système de codage (format et compression) et du matériel d'affichage (moniteur, projecteur) [3].

Quand on parle d'image numérique, on parle de pixels. Pixel est tout simplement le plus petit élément d'une image, et on peut le figurer comme un carré élémentaire. Chaque pixel est caractérisé par sa position qui peut être exprimée par deux coordonnées sur l'axe horizontal X et l'axe vertical Y et sa couleur qui représente sa valeur (Pixel (i, j)) (valeur unique, puisque le pixel est considéré comme indivisible) comme le montre la figure ci-dessous [2].

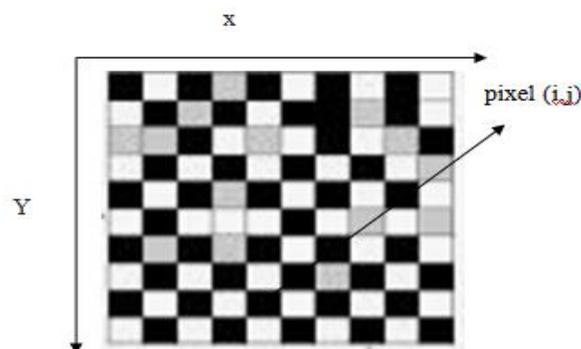
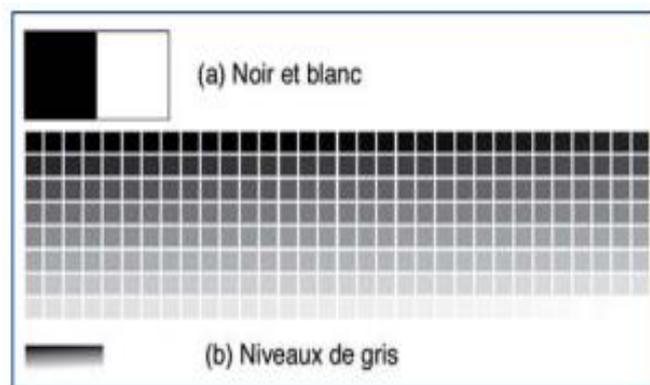


Figure I.1:Élément d'une image (le pixel).

- Chaque pixel représente un point de l'image (dans un espace colorimétrique prédéfini).
- Dans une image en noir et blanc (N&B) par exemple, un pixel sera représenté par 1 bit (0 = noir, 1 = blanc) (figure. I .2.a).
- Une image en niveaux de gris est codée avec n bits et généralement avec 8 bits = 1 octet =  $2^8 = 256$  valeurs du noir au blanc (figure. I .2.b)
- Dans une image en couleurs, codée par exemple dans l'espace RGB (256 teintes de rouge, 256 teintes de vert, 256 teintes de bleu), chaque pixel est représenté par 3 octets =  $3 \times 8$  bits permettant de représenter  $(2^8)^3 = 16,8$  millions de couleurs.

Enfin, si l'on ajoute la transparence, le codage de chaque pixel d'une image en couleur se fait avec 4 octets, c'est-à-dire  $4 \times 8$  bits permettant de représenter  $(2^8)^4 = 4$  295 milliards de couleurs.

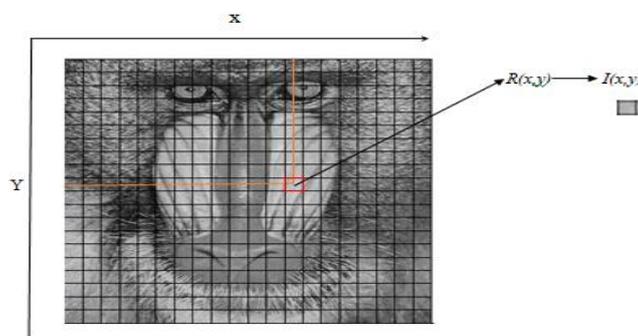


**Figure I .2 :** Pixels. Images en noir et blanc (a) ou en niveaux de gris (b).

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

### I.3.1 L'échantillonnage (Résolutions spatiale) :

Est le procédé de discrétisation spatiale d'une image consistant à associer à chaque zone rectangulaire  $R(x,y)$  d'une image continue une unique valeur  $I(x,y)$ . On parle de *sous échantillonnage* lorsque l'image est déjà discrétisée et qu'on diminue le nombre d'échantillons [2].



**Figure I.3:** Echantillonnage d'une image.

### I.3.2 La quantification (Résolution tonale):

Désigne la limitation du nombre de valeurs différentes que peut prendre  $I(i,j)$ , nombre déterminé en pratique par le nombre de bits sur lequel on code la valeur numérique en question [2].



Figure I.4: Quantification.

- Une image numérique est une image échantillonnée et quantifiée.

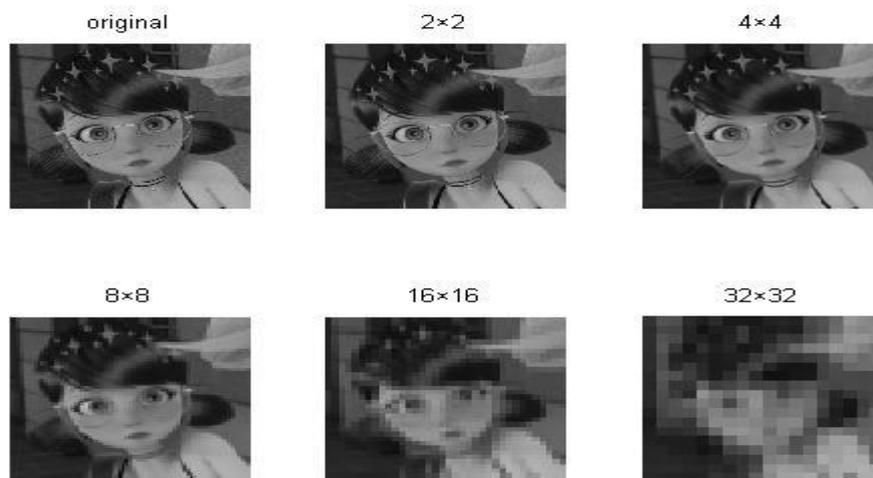


Figure I.5: Echantillonnage (Résolution spatiale).



Figure I.6 : Quantification (Résolution tonale).

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

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

### I.4 Les caractéristiques d'une image numérique:

#### I.4.1 Dimension (Définition):

C'est la taille de l'image, ou le nombre de points (pixels) la composant. Cela correspond au nombre de pixels qui compose l'image en hauteur (axe vertical) et en largeur (axe horizontal): Une image possédant 640 pixels en largeur et 480 en hauteur aura une définition de 640 pixels par 480, notée 640x480 [6].

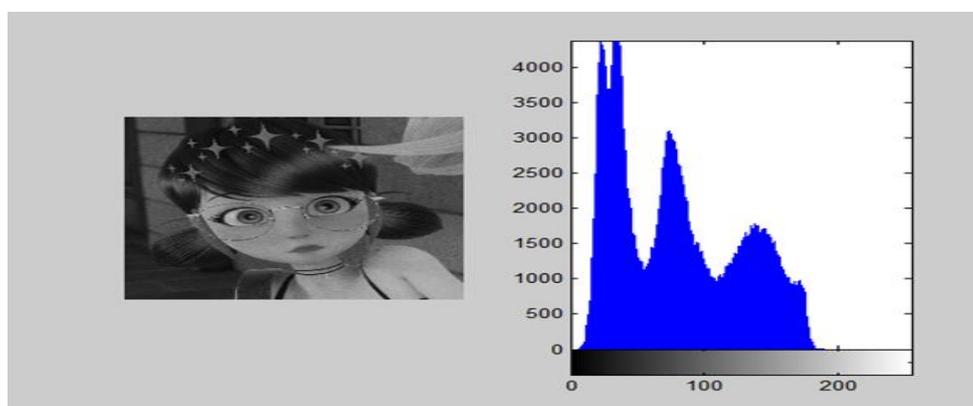
#### 1.4.2 Résolution :

La résolution d'une image est définie par le nombre de pixels par unité de longueur, généralement mesuré en dpi (points par impression) ou ppp (points par pouce). Ce paramètre est défini lors de la numérisation et dépend principalement des caractéristiques matérielles utilisées lors de la numérisation. L'augmentation de la résolution peut entraîner des durées de visualisation et d'impression plus longues, ainsi que des fichiers trop volumineux contenant des images [3].

#### 1.4.3 Histogramme :

Un histogramme est un graphique statistique utilisé pour représenter La répartition de l'intensité des pixels de l'image, c'est-à-dire le nombre de pixels par intensité lumineuse (Figure I.7).

Par convention, les histogrammes représentent les niveaux d'intensité du plus sombre (à gauche) au plus clair (à droite) en abscisse. Pour réduire les erreurs de quantification, pour comparer deux images obtenues sous un éclairage différent, ou pour mesurer une propriété d'une image, l'histogramme correspondant est généralement modifié [3].



**Figure I.7:**Image et histogramme associés.

### I .4.4 Luminance:

C'est le niveau de luminosité du point de l'image. Elle est également définie comme le quotient de l'intensité lumineuse d'une surface divisée par l'aire apparente de cette surface, pour un observateur éloigné, le mot luminosité est remplacé par le mot brillance, qui correspond à la luminosité d'un objet [3].

### I .4.5 Contraste:

C'est l'opposition apparente entre deux zones d'une image, plus précisément entre les zones sombres et claires de cette image. Le contraste est défini en fonction de la luminosité de deux zones d'images [5].

### I .4.6 Profondeur:

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

- par exemple :
  - 24 bits = 16,7 millions de couleurs.
  - 16 bits = 65 536 couleurs.
  - 8 bits = 256 couleurs.

### I .4.7 Le poids de l'image:

Se détermine en fonction de ces deux paramètres: dimensions, profondeur. Le poids de l'image est alors égal à sa dimension multipliée par sa profondeur.

- Par exemple : pour une image 640x480 en vraies couleurs (Truecolors) :
  - Nombre de pixels (dimension) :  $640 \times 480 = 307200$ .
  - Poids de chaque pixel (profondeur) : 24 bits = 3 octets.
  - Le poids de l'image est ainsi égal à :  $307200 \times 3 = 921600$  octets.

### I .4.8 Contours et textures:

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

## I .5 Les différents types d'images :

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

### I.5.1 L'image monochrome (Binaire):

C'est le type d'image le plus simple, où chaque pixel ne peut prendre qu'une valeur de noir ou de blanc (chaque pixel est encodé sur un bit). Il s'agit généralement du type d'image utilisé pour numériser du texte lorsque le texte est composé d'une seule couleur.



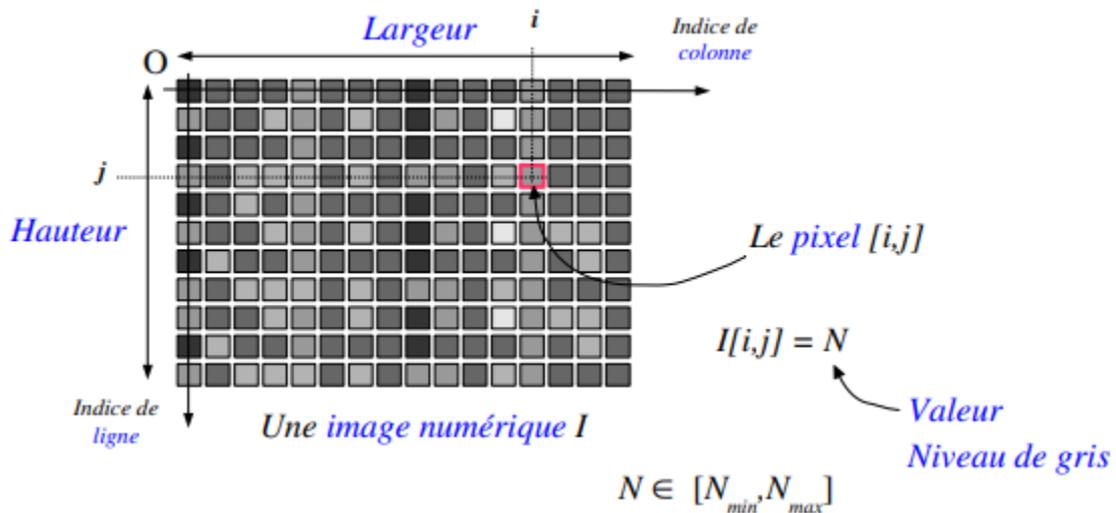
Figure I.8 : L'image monochrome (Binaire).

### I.5.2 L'image en niveaux de gris :

En règle générale, les images en niveaux de gris contiennent 256 nuances de gris. Par convention, une valeur de zéro représente le noir (intensité lumineuse nulle).

Valeur 255 Blanc (intensité lumineuse maximale). Liés au numéro 256 Quantifier l'image. En fait chaque entier représente un niveau Le codage en niveaux de gris est de 8 bits. C'est donc entre 0 et  $2^8 - 1 = 255$ .

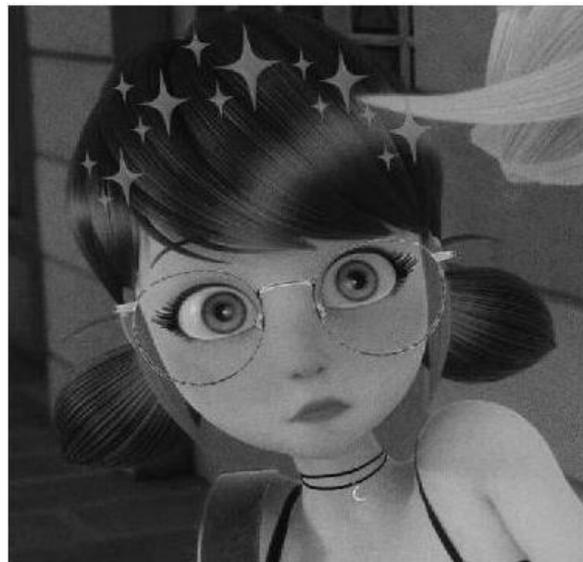
C'est La quantification la plus courante. Vous pouvez encoder l'image en gris Sur 16 bits ( $0 < n < 2^{16} - 1$ ) ou 1 bits : dans ce dernier cas ! classe gray vaut 0 ou 1 : alors une image binaire (noir et blanc) [2].



$$(N_{max} - N_{min}) = \text{nombre de niveaux de gris} \quad (I.1)$$

$$\log_n(N_{max} - N_{min}) = \text{dynamique} \quad (I.2)$$

**Figure I.9 :** Pixels et niveaux de gris.



**Figure I.10 :** L'image en niveaux de gris.

### I.5.3 L'image en couleurs:

Les espaces colorimétriques sont basés sur la composition des couleurs, le mélange de trois composants pour produire une couleur. Un pixel est codé par trois valeurs numériques. La signification de ces valeurs dépend du type d'encodage choisi. Le plus couramment utilisé pour travailler avec des images numériques est l'espace colorimétrique "Rouge-Vert-Bleu" (R,G,B) (RGB en anglais).

La reproduction des couleurs à l'écran utilise cette représentation. C'est une composition additive. Il en existe bien d'autres : Cyan-Magenta-Yellow, Teinte-Saturation-Luminosité (ou HSL en anglais), YUV, YIQ, Lab, XYZ, etc.

Il existe plusieurs modes de représentation de la couleur, le plus utilisé pour les images numériques est le mode RGB. Ce mode est basé sur la synthèse additive des couleurs, c'est à dire que le mélange des trois composantes (R, G, B) donne une couleur. Il existe différents types d'images couleurs en fonction du nombre de bits utilisés pour le stockage de l'information couleur [2].



**Figure I.11 :** Image en couleurs.

### ***1.5.4.1 L'image en "vraies couleurs" (Truecolors) (ou 24 bits) :***

Il s'agit d'une appellation trompeuse car on est dans un monde numérique (discret, fini) qui ne peut pas rendre compte de la réalité (infinie) [4]. Chaque pixels peut prendre une valeur dans le RGB comprise entre 0 et 255 (soit  $256 \times 256 \times 256$  possibilités= plus de 16 millions de possibilités). L'information couleur de chaque pixel est donc codée par 3 octets, par conséquent les images en vraies couleurs sont très "lourdes" en termes de l'espace de stockage.

### ***1.5.4.2 L'image indexée :***

Pour réduire la manipulation des images 24 bits (et/ou pour économiser de l'espace de stockage), vous pouvez utiliser des couleurs indexées pour représenter les motifs. Dans ce cas, nous ajoutons (généralement) une palette de 256 couleurs à l'image. Les 256 couleurs sont choisies parmi 16 millions de couleurs dans la palette RVB.

Chaque pixel ne véhiculera plus le code couleur RVB qui lui est attribué, mais juste un nombre compris entre 0 et 255.

Chacun de ces nombres correspond à une couleur, définie par Son code RVB et est stocké dans une palette (table de correspondance (LUT)) avec les données d'image. Lors de

la visualisation de l'image, il correspond au numéro attribué à la couleur. Chaque pixel (entre 0 et 255) et le code couleur RVB correspondant dans la LUT.[4]



Figure I.12: Image indexée.

## I.6 Les formats standards d'image :

### I.6.1 images vectorielles:

Les images vectorielles (ou images au trait) sont des images numériques. Se compose d'objets géométriques individuels (segments de ligne, polygones, arcs, etc.), chacun défini par diverses propriétés de forme, Position, couleur, etc. Essentiellement, le tracé vectoriel est ré-sinusoïdal pour chaque visualisation, ce qui génère le calcul machine.

L'intérêt est de pouvoir redimensionner les images à volonté. Il n'y a pas d'effet d'échelle. L'inconvénient est d'atteindre. Pour une qualité photoréalisme, il faut avoir de la puissance de calcul importante [2].

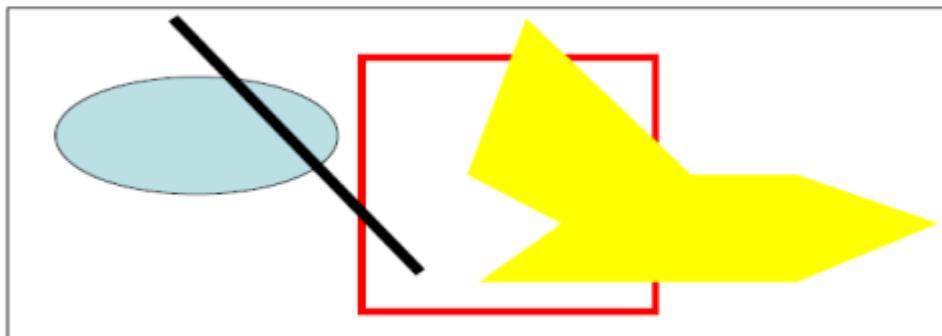


Figure I.13: Image vectorielle.

- **Les avantages des images vectorielles:**
  - Elles sont adaptées au stockage d'images composées de formes géométriques [2]
  - Elles peuvent aisément être redimensionnées [7].
  - Elles prennent moins de place qu'une image bitmap [7].
  - Ce type d'image est également utilisé dans les animations Flash, Utilisé sur Internet

pour créer des bannières publicitaires, des présentations de sites Web et même des sites Web complets [2].

### - Les inconvénients des images vectorielles:

- Elles peuvent difficilement stocker des images complexes comme des photographies [2].
- Les fichiers vectoriels sont plus fragiles que les fichiers raster dont les en-têtes doivent notamment être complets. dégradation minimale des informations Souvent irréparable [7].
- La compatibilité entre les formats est difficile car chaque format de fichier vectoriel a ses propres propriétés [7].

### I.6.1.1 Les principaux formats:

#### ➤ WMF (Windows Meta File):

C'est un format vectoriel utilisé par Windows et il est reconnu par de nombreux programmes fonctionnant sous Windows. C'est un format dans lequel les images matricielles et les images vectorielles peuvent être enregistrées. [6]

#### ➤ EPS (Postscript / Encapsulated Postscript):

C'est un format vectoriel largement utilisé. Par conséquent, c'est le format préféré lors de l'exportation de graphiques vectoriels. Les formats raster peuvent également être convertis au format EPS. Dans ce cas là, il ne s'agit que d'une encapsulation et non d'une conversion en image vectorielle.[5].

#### ➤ CGM (Computer Graphics Metafile):

Les fichiers graphiques CGM restent un moyen privilégié d'échange de dessins vectoriels entre applications, mais il ne traite pas les images matricielles [5].

#### ➤ DXF (Data eXchange Format):

Format des produits Autocad et Autodesk.

#### ➤ PCT (Pict):

Ce format vient du monde Macintosh, où il constitue un standard. Peut contenir aussi bien des données vectorielles que matricielles. Sur PC, l'ouverture d'une image Pict se révèle problématique avec de nombreux logiciels. Les images Pict peuvent être ouvertes avec Quick Time (Picture Viewer) [6].

### I.6.2 image matricielle:

Une image matricielle (ou image bitmap) est composée (comme son nom l'indique) d'une matrice (tableau) de points à plusieurs dimensions, Chaque dimension représente une dimension spatiale (hauteur, largeur, profondeur), temps (durée) ou autre (par exemple, résoudre). Compte tenu de la façon dont les images sont actuellement visualisées, par exemple l'écran de l'ordinateur est principalement dominé par des images raster, et la description vectorielle doit d'abord être convertie dans la description de la matrice avant de l'afficher sous forme d'image [2].

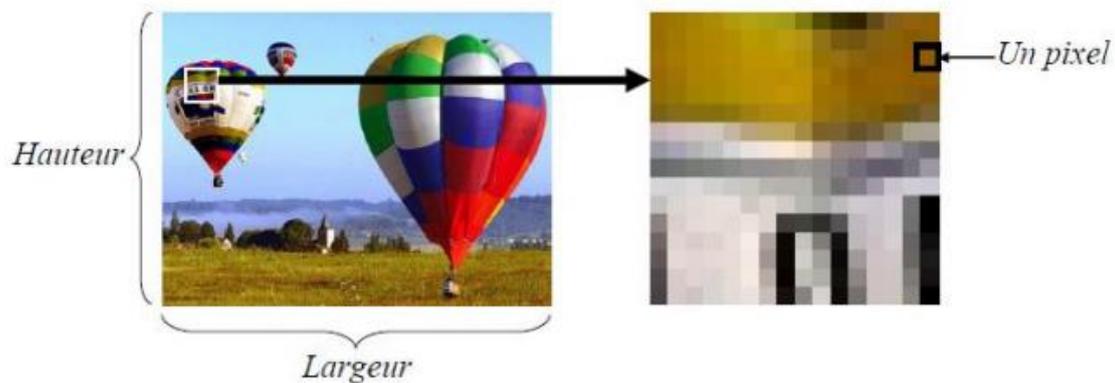


Figure I.14: Image matricielle.

- **Les avantages des images matricielles**
  - Elles peuvent facilement être créées et stockées dans un tableau de pixels représentant l'image.
  - La lecture/écriture d'un pixel facilite la représentation de l'image comme une grille [2].
  - Elles peuvent facilement être affichées sur un écran ou être imprimées [2].
- **Les inconvénients des images matricielles**
  - Les fichiers peuvent être très gros (nécessité de compression) [4].
  - Problème de changement d'échelle (apparition d'effets de marches d'escalier ou de flou avec interpolation).
  - Les dimensions de l'image doivent être prévues pour la résolution de l'interface de sortie (écran, imprimante) [6].

### 1.6.2.1 Les principaux formats :

- **BMP (Windows Bitmap):**

C'est le format actuel utilisé par Windows. Il produit des images de bonne qualité et est reconnu par de nombreuses applications. C'est le format le plus utilisé, par contre, il est extrêmement volumineux lorsqu'il utilise le codage en « truecolors » [7].
- **PCX (PiCtureXchange):**

Le format défini par Paintbrush. Il accepte les modes de couleur, indexés, niveaux de gris et le noir et blanc [5].
- **GIF (Graphic Interchange Format):**

Créé par CompuServe, utilise aussi le codage RGB, mais le format GIF n'utilise pas toutes les 16 millions de couleurs. Il prend les 256 couleurs les plus courantes pour réaliser l'image au format GIF. Cela permet une bonne compression et un affichage rapide de l'image [5].
- **JPG ou JPEG (Joint Photographic Experts Group):**

Créé par un consortium industriel, ce format très utilisé sur Internet, permet d'afficher les images en mode 16 millions de couleurs. Il est conçu pour réduire le plus possible la taille

des fichiers graphiques en acceptant éventuellement de légères pertes de qualité. Il est destiné à la transmission rapide d'information. Ces résultats de compression sont extraordinaires [7].

➤ **TIFF (Tag Image File Format):**

C'est un format d'excellente qualité, mais qui présente des problèmes de compatibilité du fait d'une multiplicité de version. Il existe aussi une version compressée qui fournit des fichiers très compacts sans perte notable de qualité [5].

Ce format est compatible avec d'autres plates formes (macintosh). Il est utilisé par les professionnels [7].

### I.7 Généralités sur les traitements d'image:

#### I.7.1 Définition:

Le traitement d'images désigne l'ensemble des méthodes et techniques de manipulation des images dans le but d'améliorer leur aspect visuel ou d'en extraire des informations jugées pertinentes. Elle se définit comme un ensemble de tâches visant à extraire des images Informations qualitatives et quantitatives [8].

#### I.7.2 Les domaines d'applications:

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

##### *I.7.2.1 Imagerie aérienne et spatiale:*

Cette technique trouve des applications dans les ultrasons, la résonance magnétique nucléaire et l'identification automatique de cellules ou de chromosomes [8].

##### *I.7.2.2 Technologies biomédicales:*

Le traitement implique la recherche d'images satellites, l'analyse des ressources terrestres, la cartographie automatique, l'analyse météorologique [8].

##### *I.7.2.3 La robotique:*

Qui connaît actuellement le plus grand développement et dont les tâches usant de l'imagerie sont principalement l'assemblage (pièce mécanique, composants électroniques,...etc), le contrôle de qualité, ainsi que la robotique mobile [6].

##### *I.7.2.4 la télésurveillance:*

Exemple, radar automatique Les véhicules sont recherchés en temps réel par des caméras fixes identifiant les immatriculations des véhicules dans le flux de véhicules circulant sur la boucle [8].

### ***I.7.2.5 Simulation et contrôle de processus:***

Cette technique est utilisée dans les cours pilotes et les panneaux de contrôle [5].

### ***I.7.2.6 L'astronomie:***

Dans tout traitement d'image, et en particulier en astronomie, on cherche à extraire le maximum d'informations fiables des données expérimentales dont on dispose. Pour cela, on est amené à inverser l'opérateur d'imagerie intervenant lors du processus d'acquisition de ces images. Le problème inverse correspondant, qui s'avère être mal conditionné et donc mal posé doit être régularisé. [9]

Le traitement d'image est également utilisé en chimie, physique nucléaire (identification de trajectoires de particules), armement (guidage de missiles, reconnaissance des formes), métiers du spectacle, télécommunications (TV, radio, vidéo, publicité,...), architecture, imprimerie,...etc.

## **I.7.3 Principal techniques:**

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

### ***I.7.3.1 Acquisition:***

Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation (échantillonnage, quantification). On utilise plus couramment des caméras vidéo, des appareils photos numériques. En médecine, on utilise des imageurs IRM, TEP, scanner X, écho doppler, échographie, scintigraphie..... etc. [10].

### ***I.7.3.2 Filtrage:***

Est une opération qui consiste à réduire le bruit contenu dans une image au moyen d'algorithmes provenant des mathématiques par l'utilisation de méthodes d'interpolation ou de la morphologie mathématique [6].

### ***I.7.3.3 Segmentation:***

Permet d'isoler dans l'image les objets sur lesquels doit porter l'analyse, de séparer, dans une image, les régions d'intérêt du fond. Il existe plusieurs techniques, la plus simple étant le seuillage. La segmentation peut être manuelle ou automatique [11].

### ***I.7.3.4 Convolution:***

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

### I.7.4 Principales applications:

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

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

### I.8 Conclusion:

Le traitement des images pose des problèmes plus complexes que le traitement du texte. En effet, une image est un objet à deux dimensions et doit représenter un espace à trois dimensions, Cela a deux conséquences principales : La quantité de données à traiter est beaucoup plus importante, et la structure de ces données est beaucoup plus complexe. Par conséquent, la manipulation, le stockage et la représentation de ces données rencontrent certaines restrictions. Grâce au traitement des images, ces restrictions sont levées ou contournées. En effet, ce domaine Essayer de détecter la présence d'une forme, d'un contour ou d'une texture d'un modèle connu, c'est le problème de la détection. Un autre aspect du traitement d'image implique l'analyse et la compréhension d'images dans des images. Extraire des informations utiles. Il essaie aussi parfois de compresser l'image pour que Augmenter la vitesse et la capacité de stockage lors du transfert d'informations, tous en dégradant le moins possible les images considérées. C'est le domaine du codage ou compression des images, objet de notre étude, que l'on détaillera dans les chapitres qui suivent.

# Chapitre 2

## **Théorie De L'information**

### II.1 Introduction :

La théorie de l'information se préoccupe des systèmes d'information, des systèmes de communication et de leur efficacité. La notion de système d'information dans le domaine de communication étant large, il en va de même de la théorie de l'information.

Dans ce chapitre, un rappel de la théorie de l'information est envisagé dans lequel nous résumons les idées sur l'information, la quantité d'information et l'entropie de Shannon. Ainsi, nous avons un bref aperçu sur le codage du canal et le codage de source et leurs buts et leurs différents types. Nous nous focalisons sur les points que nous considérons comme essentiels pour la bonne compréhension de la suite du mémoire.

### II.2 Principes de la théorie de l'information :

#### II.2 .1 Définition de l'information :

L'information est un ensemble de données ayant un sens. Elle nous permet d'avoir des renseignements sur une personne, un objet, un événement...etc.

Une information peut être présentée sous différentes formes :

- Un texte (ex. : le nom d'une personne).
- Une image (ex. : la photo d'une personne).
- Un son (ex. : la voix d'une personne).
- Une vidéo (ex. : un vidéo décrivant le comportement d'une personne) [12].

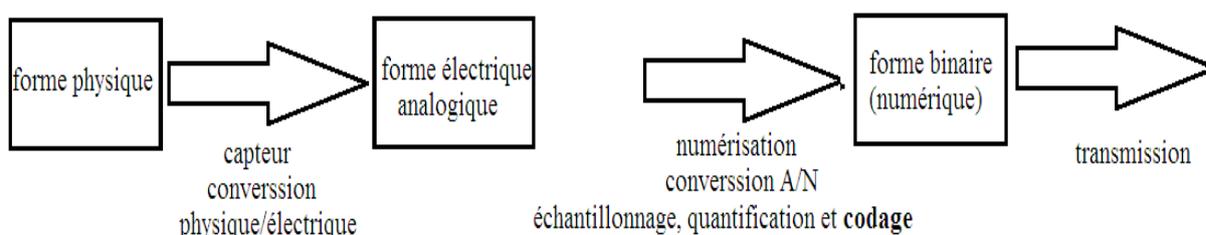


Figure II.1: Chaîne de traitement de l'information.

#### II.2 .2 Définition de la théorie de l'information :

La théorie de l'information est la science qui s'intéresse à l'étude de l'information à transmettre par la détermination des ressources adéquates et nécessaires pour permettre une transmission fiable des informations depuis une source vers un destinataire. Le fondement de cette théorie fut inventé en 1948 par Claude Elwood Shannon, il démontra la relation entre le taux de transmission et la capacité du canal de transmission, de telle sorte que l'on peut atteindre un taux d'erreur aussi proche de zéro que désiré si le taux de transmission est inférieur à la capacité du canal [12].

### II.2 .3 Quantité d'information d'un symbole :

Considérons l'ensemble  $Q$  de symboles que nous avons déjà introduit. Plutôt que d'essayer de définir directement la quantité d'information que nous apporte l'apparition d'un symbole  $q_i$ ,  $i \in [1, M]$  ( $X_n = q_i$ ), essayons de la caractériser. Intuitivement, la quantité d'information  $H_i$  d'un symbole  $q_i$  doit avoir les propriétés suivantes :

- 1- Si on est certain que  $q_i$  apparaît à un instant  $n$  quelconque, alors le fait que  $q_i$  apparaisse à cet instant  $n$  ne nous apporte aucune information ( $H_i = 0$ ).
- 2- Moins on est certain de l'apparition de  $q_i$ , plus grande est la quantité d'information que nous apporte son apparition.
- 3- Si l'apparition de  $q_i$  est indépendante de l'apparition de  $q_j$  ( $i, j \in [1, M]$ ), aux instants  $n$  et  $k$  respectivement, alors l'apparition conjointe de  $q_i$  et  $q_j$  nous apporte une quantité d'information égale à la somme des quantités d'information de  $q_i$  et  $q_j$  (nous appelons une telle source une source discrète sans mémoire (SDSM)).

Ces hypothèses nous amènent à définir  $H_i$  comme [12] :

$$H_i = \log_2 \left( \frac{1}{p_i} \right) = -\log_2 p_i \quad (\text{II.1})$$

En effet,  $H_i$  possède les propriétés suivantes :

$$- H_i = 0 \text{ si } p_i = 1 ; H_i \geq 0 \quad (\text{II.2})$$

$$- H_i > H_j \text{ si } p_i \log_2 p_{ij} < p_j \quad (\text{II.3})$$

$$- H_{ij} = -\log_2 p_{ij} = -\log_2 p_i p_j = -\log_2 p_i - \log_2 p_j = H_i + H_j \quad (\text{II.4})$$

### II.2 .4 Quantité d'information moyenne (Entropie de Shannon) :

La probabilité et l'information sont liées par le concept d'entropie, qui est défini par Boltzmann décrit le mélange statistique en thermodynamique statistique. Dans ce contexte, l'entropie est une mesure qui exprime le degré de désordre ou de désordre d'une caractéristique. Le caractère aléatoire du système, et donc le manque d'informations sur un système donné.

Depuis Avec cette définition et des considérations axiomatiques sur l'information, Shannon a identifié une autre forme d'entropie statistique pour les sources d'information et démontre l'importance de sa formule comme mesure de l'information. Ce concept s'appelle l'entropie de Shannon, il peut être identifié comme l'information transportée par un état donné.

Considérons la source  $\underline{X} = (X_1, X_2, \dots, X_n)$  pour une grande dimension  $n$ , où chaque symbole  $X_i$ , peut prendre un nombre fini des valeurs. Supposons, pour des raisons de simplifications, que cette source stationnaire est « sans mémoire », c'est-à-dire qu'à chaque instant, un symbole est tiré indépendamment des précédents.

Les symboles  $X_1, X_2, \dots, X_n$  sont alors indépendants et identiquement distribués, et en notant  $p(x)$  la probabilité qu'un symbole égale  $x$ , la probabilité  $p(\underline{x})$  d'un message donné  $\underline{x} = (x_1, x_2, \dots, x_n)$  est le produit des probabilités individuelles [13] :

$$p(\underline{x}) = p(x_1) \cdot p(x_2) \dots \cdot p(x_n) \quad (\text{II.5})$$

Shannon décrit l'entropie comme:

$$H = \sum_x p(x) \log_2 \frac{1}{p(x)} \quad (\text{II.6})$$

Est une quantité positive que Shannon appelle *entropie* par analogie avec l'entropie étudiée par Ludwig Boltzmann en mécanique statistique.

**Remarque :** La quantité d'information s'exprime en bit/symbole.

- **Débit d'information :** Le débit d'information d'une source discrète de symboles est définie par :

$$H'(X) = H(X) \cdot f \quad (\text{II.7})$$

L'unité du débit d'information est donc : (bit/symbole)  $\times$  (symbole/sec) = bit/sec [12]

- **Quantité de décision :** La quantité de décision est une mesure du nombre de bits requis pour coder en binaire M différents niveaux de quantification. Il s'exprime comme suit :

$$D = \log_2 M \quad (\text{II.8})$$

- **Redondance :**

La redondance d'une source est la différence entre la quantité d'information maximale de la source et sa quantité d'information moyenne. Elle définie par :

$$R = H(X)_{max} - H(X) \quad (\text{II.9})$$

- **Débit de décisions :**

Le débit de décisions est donné par :

$$D' = D \cdot f \quad (\text{II.10})$$

L'unité du débit de décisions est donc le bit/sec.[12]

### II.2 .5 Entropies conditionnelle et conjointe :

Nous pouvons interpréter ces entropies comme suite :

- $H(X)$  est l'incertitude moyenne relative à l'entrée du canal.
- $H(Y)$  est l'incertitude moyenne relative à la sortie du canal.

#### II 2.5.1 L'entropie conditionnelle $H(X|Y)$ :

Est une mesure de l'incertitude moyenne d'un événement a la sortie si un événement est observée a l'entré.  $H(X|Y)$  est parfois appelée ambigüité de X par rapport à Y.

L'entropie conditionnelle  $H(Y|X)$  est l'incertitude moyenne sur la sortie du canal sachant que X a été transmise.

#### II 2.5.2 L'entropie conjointe $H(X,Y)$ :

L'entropie conjointe mesure l'incertitude d'ensemble relative à l'ensemble du canal de communication. Il existe deux relations intéressantes entre ces diverses quantités [13]:

$$H(X,Y) = H(X/Y) + H(Y) \quad (\text{II.11})$$

$$H(X,Y) = H(X/Y) + H(X) \quad (\text{II.12})$$

### II.3 Le codage de source :

La source d'information délivre le message porteur de l'information. De nature analogique ou numérique, elle peut être un signal audio, une vidéo, une image, un texte ou des données brutes.

De sens abstraite, on peut définir le codage de source comme une transformation de l'information de la source d'une forme initiale présentative à une autre forme adaptée à une éventuelle application, transmission, stockage...etc. généralement il s'agit d'une représentation binaire de différents niveaux de quantification. Le choix du code binaire associé à chaque symbole est basé sur la théorie de l'information. On retrouve ici principalement des codes de longueur fixe et des codes de longueur variable [12].

Le tableau ci-dessous donne un exemple des codes binaires pour un ensemble de dimension 4. Ils intéressent de citer deux types de code de longueur fixe et code de longueur variable.

$X_i$	CODE 1	CODE 2	CODE 3	CODE 3	CODE 4	CODE 5
X1	00	00	0	0	0	1
X2	01	01	1	10	01	01
X3	00	10	00	110	011	001
X4	11	11	11	111	0111	0001

**Tableau II.1 :** Exemple du code de longueur fixe et code de longueur variable.

Le décodage de source effectue l'opération inverse du codage, il permet de reconstituer le message initial à partir des séquences de substitution par l'exécution des algorithmes appropriés de décodage de source (dans certains cas il s'agit d'une décompression) [15].

### II.3.1 Types de sources :

- Sources discrètes sans mémoire : la probabilité d'apparition d'un symbole ne dépend pas des symboles précédents :

$$p = (x_{i_n}/x_{i_{n-1}}, x_{i_{n-2}}, \dots) = p(x_{i_n}) \quad (\text{II.13})$$

- Sources discrètes avec mémoire : modélisation par un processus stochastique : chaînes de Markov :

$$p = (x_{i_n}/x_{i_{n-1}}, x_{i_{n-2}}, \dots) = p(x_{i_n}/x_{i_{n-1}}) \quad (\text{II.14})$$

- Sources discrètes stationnaire = probabilité d'apparition des différents symboles indépendants de temps :

$$p(x_{i_n}) = p(x_{i_{n+k}}) \quad (\text{II.15})$$

#### - Le but de codage de source :

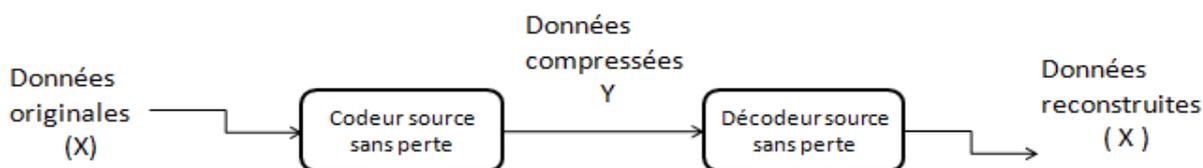
Le codage de source a pour but d'alléger le débit moyen d'information du canal en réduisant la redondance des informations transmises par la source. Par conséquent, la tâche principale du codage de source est de représenter le signal avec le nombre minimum de symboles (binaires) sans dépasser le "niveau de distorsion acceptable" déterminé par l'application [15].

### II.3.2 Types de codage de source :

On peut distinguer deux famille principales de codage de source, appelés codages avec perte (irréversible) et sans perte (réversible), selon que le codage a ou sans perte d'information.

#### II.3.2.1 Codage sans perte :

Ce type de codage supprime uniquement la redondance dans les informations d'origine. Ainsi, le décodeur pourra reconstituer la même version que l'information d'origine [15]

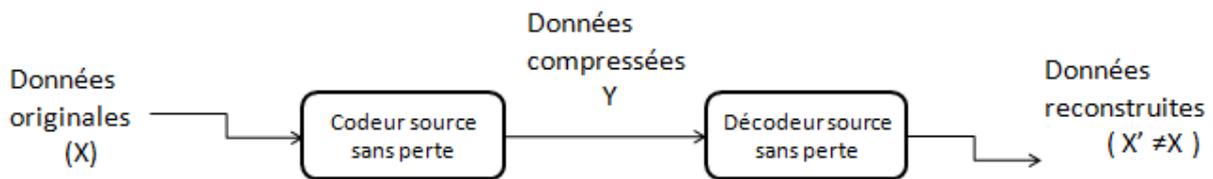


**Figure II.2 :** Etapes principales de codage sans perte.

#### II.3.2.2 Codage avec perte (irréversible) :

Le codage irréversible, en plus d'éliminer la redondance, réduit également les informations moins importantes par rapport aux informations d'origine. Par conséquent, ce

type d'encodage introduit une distorsion appelée Source Distorsion (DS) [3].



**Figure II.3:** Etapes principales de codage avec perte.

### II.3.3 Classification des codes :

#### II.3.3 .1 Code de longueur fixe :

Les codes à longueurs fixes sont les codes dans lesquels chacun des codes binaires associés à chaque symbole dans  $Q$  a une longueur constante. Le code utilisé dans l'exemple ci-dessous est un exemple de code à longueur fixe, de longueur égale à 3 (3 bits/symbole).

D'une manière générale, si on a  $M$  symboles dans l'ensemble  $Q$  alors la longueur minimale  $L_{min}$  d'un code à longueur fixe pour coder les  $M$  symboles dans  $Q$  est :

$$L_{min} = \lceil \log_2 M \rceil \quad (\text{II.16})$$

Ce qui désigne ici le plus petite entier tel que  $\log_2 M \leq L_{min}$

**Exemple :**  $\log_2 6 = 2, 5849625$  donc  $\lceil \log_2 6 \rceil = 3$ .

Nous avons donc la relation :  $L_{min} = L \geq \log_2 M$  (II.17)

Où  $\bar{L}$  désigne la longueur moyenne.

Supposons maintenant que l'on veuille coder des blocs de  $n$  symboles à la fois. C'est-à-dire, si la sortie de la source discrète d'information est une séquence  $\{U_k\}$ ,  $k = 0, 1, 2, \dots$  alors on se propose de coder des blocs de taille  $n$ ,  $[u_i, \dots, U_{i+n}]$  de symboles où  $U_k \in Q$  et  $i$  est un entier quelconque.

On pouvons regarder un bloc de  $n$  symboles comme étant la valeur d'une variable qui prend ses valeurs dans le produit cartésien  $Q \times Q \times \dots \times Q = Q^n$ . Nous pouvons alors distinguer  $M^n$  valeurs possibles pour la variable et donc la longueur nécessaire pour un code à longueurs fixes vérifie les relations suivantes :

$$L_n = \lceil \log_2 M^n \rceil \quad (\text{II.18})$$

$$\bar{L} = \frac{\lceil \log_2 M^n \rceil}{n} \geq \frac{\lceil n \log_2 M \rceil}{n} = \log_2 M \quad (\text{II.19})$$

$$\bar{L} = \frac{\lceil \log_2 M^n \rceil}{n} < \frac{\lceil n \log_2 M \rceil + 1}{n} = \log_2 M + \frac{1}{n} \quad (\text{II.20})$$

Où  $\bar{L}$  désigne la longueur moyenne par symbole. Donc si  $n$  est grand, la longueur moyenne est proche de la longueur nécessaire pour coder chaque symbole séparément.

**Remarque :** La méthode de codage par codes à longueur fixe ne présente aucun problème de synchronisation. Elle est non probabiliste. Mais elle n'offre aucune possibilité de compression de données et par conséquent aucune optimisation à l'égard de la quantité d'information [12].

### II.3.3 .2 Code de longueur variable :

Cette méthode de codage permet la compression de donnée dans le sens où nous pouvons associer les codes les plus courts aux symboles les plus probables et les codes les plus longs aux symboles les moins probables.

Un code à longueur variable associe à chaque symbole  $q$  ( $q \in Q$ ) une série de bits  $C(q)$  de symboles binaires appelés mot de code de longueur  $l(q)$ .

#### Exemple :

Soit l'alphabet  $Q = \{a, b, c\}$ . Un exemple de code à longueur variable est :

$$C(a) = 0$$

$$C(b) = 10$$

$$C(c) = 11$$

- **Propriété de dé-codabilité unique :**

La propriété de dé-codabilité unique est une propriété essentielle qu'un code à longueur variable doit avoir pour qu'il soit déchiffable. Cette propriété consiste en le fait qu'étant donnée une séquence de symboles source et une synchronisation initiale, cette séquence doit pouvoir être reconstruite sans aucune ambiguïté à partir de la série binaire qui lui est associée par le codeur. On peut énoncer la définition suivante :

- **Définition :**

Un code  $C$  pour une source discrète est décodable de manière unique si, pour toute séquence de symboles  $U_1, U_2, \dots, U_m$ , un, la concaténation de ses mots de codes  $C(U_1) C(U_2) \dots C(U_m)$  est différente de la concaténation des mots de codes  $C(U'_1) C(U'_2) \dots C(U'_m)$  de toute autre séquence de symboles  $U'_1, U'_2, \dots, U'_m$  [12]

- **Exemple :**

Le code ci-dessous ne satisfait pas la propriété de dé-codabilité car transmettre  $C(a)C(b)$  peut se confondre avec la transmission de  $C'(c)$  autrement la règle de-décodabilité est que la séquence de longueur  $l$  utilisée pour codé un symbole ne doit pas apparaitre dans le début des longueurs supérieur suivantes :

$$C'(a) = 0$$

$$C'(b) = 1$$

$$C'(c) = 01$$

### II.3.3.2.1 Code de Huffman :

L'imminent mathématicien David Hoffman a proposé en 1952 une méthode statistique qui pourrait attribuer des mots de code binaires à divers symboles (tels que des pixels ou des caractères) à compresser. La longueur de chaque mot de code n'est pas la même pour tous les symboles : les symboles les plus courants (ceux qui apparaissent le plus souvent) sont codés avec des petits mots de code, tandis que les symboles les plus rares reçoivent des codes plus longs [14].

#### Le principe est le suivant:

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

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

#### Exemple:

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

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

Tableau II.2 : Code de Huffman.

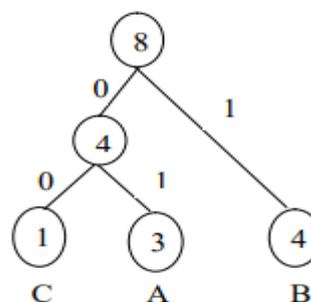


Figure II.4 : Arbre de Huffman.



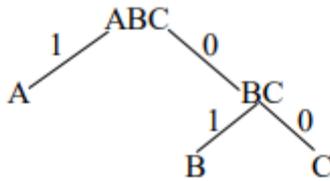


Figure II.6 : Arbre d'attribution des codes de Shannon.

### II.3.6.5 Code sans préfixe:

Une classe importante des codes qui jouissent de la propriété de dé-codabilité unique sont les codes sans préfixe. Un préfixe pour une séquence  $y_1, y_2, \dots, y_n$ , est toute sous-séquence  $y_1, y_2, \dots, y_m$  ( $m < n$ ) de la séquence donnée. Un code est sans préfixe si aucun mot de code n'est un préfixe pour aucun autre mot de code [15].

**Remarque :** Tout code à longueur fixe est un code sans préfixe.

#### - Construction de codes sans préfixe :

On utilise un arbre de décomposition pour la construction d'un code sans préfixe.

Soit :  $Q = \{a, b, c\}$ . Pour construire un code sans préfixe, on ne place jamais les symboles aux nœuds intermédiaires, comme illustré par la figure II.7.

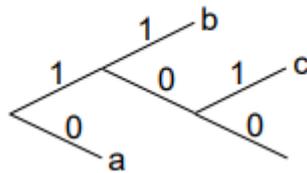
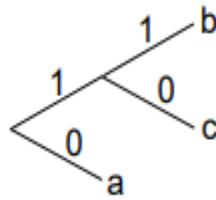


Figure II.7: Arbre de décomposition pour la construction de codes sans préfixe.

Un code sans préfixe est dit plein si tous les nœuds terminaux correspondent à des mots des codes. Sinon, le code n'est pas plein [15].

#### Exemple :

Le code de la figure II.7 n'est pas plein car nous pouvons le réduire à celui qui est montré par la figure II.8.



**Figure II.8 :** Réduction de l'arbre de la figure

Utilisons le code C et supposons que l'on veuille décoder la séquence 1010011 . . . de gauche à droite. 1 n'est pas un mot de code ; le décodeur procède à donc à droite pour décoder 10 comme étant le symbole c. La séquence restante est 10011. On décode encore c, puis a, puis b . . . etc. La séquence décodée sera donc "ccab".

**Remarque :** La synchronisation est d'une importance primordiale pour un code à longueur variable. Le choix du code doit donc être fait minutieusement [15].

- **Inégalité de Kraft pour les codes sans préfixe :**

Un important théorème, dû à Kraft, pour les codes sans préfixe est le suivant :

**Théorème 1 :**

Pour tout code sans préfixe, pour un alphabet Q, ayant les longueurs  $l(q)$ ,  $q \in Q$ , nous avons le résultat suivant [12] :

$$\sum_{q \in Q} 2^{-l(q)} \leq 1 \quad (\text{II.21})$$

Réciproquement, si cette relation est satisfaite alors un code sans préfixe existe avec les longueurs  $l(q)$ . De plus, pour un code plein, nous avons :

$$\sum_{q \in Q} 2^{-l(q)} = 1 \quad (\text{II.22})$$

Et pour un code non-plein

$$\sum_{q \in Q} 2^{-l(q)} < 1 \quad (\text{II.23})$$

**Exemple :** Existe-t-il des codes ayant des longueurs 1, 2, 2 et 1, 1, 2 ?

Revenons au modèle où l'alphabet Q est composé de M symboles  $q_i$  chacun caractérisé par sa probabilité d'apparition  $p_i$ . Supposons que l'on choisisse un ensemble de longueurs  $l(q)$  satisfaisant l'inégalité de Kraft. Nous pouvons voir donc  $l(q)$  comme une variable aléatoire.

Si  $\bar{L}$  désigne la longueur moyenne par symbole, nous aurons :

$$\bar{L} = E[l(q) = l_i] = \sum_{i=1}^M P_i l_i \quad (\text{II.24})$$

Notre objectif est de réduire le plus possible la longueur moyenne  $\bar{L}$

### Théorème 2 :

La longueur moyenne minimale  $\bar{L}_{min}$  pour un code sans préfixe satisfaisant l'inégalité de Kraft satisfait à la relation suivante [12] :

$$H \leq \bar{L}_{min} < H + 1 \quad (\text{II.25})$$

Où H est l'entropie de la source. Si les probabilités  $p_i$  sont des puissances entières de 2 alors la relation devient :

$$\bar{L}_{min} = H \quad (\text{II.26})$$

### Démonstration :

$$H(X) - \bar{L} = \sum_{i=1}^M p_i \log_2 \left( \frac{1}{p_i} \right) - \sum_{i=1}^M p_i l_i \quad (\text{II.27})$$

$$H(X) = \sum_{i=1}^M \log_2 \left( \frac{1}{p_i} \right) + \sum_{i=1}^M p_i (-l_i) \quad (\text{II.28})$$

$$H(X) = \sum_{i=1}^M p_i \log_2 \left( \frac{2^{-l_i}}{p_i} \right) \quad (\text{II.29})$$

$$H(X) - \bar{L} \leq (\log_2 e) \sum_{i=1}^M p_i \left( \frac{2^{-l_i}}{p_i} - 1 \right) = (\log_2 e) (\sum_{i=0}^{N-1} 2^{-l_i} - \sum_{l=0}^{M-1} p_l) \leq 0 \quad (\text{II.30})$$

$$\text{Si } \frac{2^{-l_i}}{p_i} = 1 \text{ ou } l_i = -\log_2 p_i \text{ alors } H(X) - \bar{L} = 0 \quad (\text{II.31})$$

Ce qui veut dire que :

$$H(X) = \bar{L}_{min} \quad (\text{II.32})$$

Pour montrer l'autre inégalité il suffit de remarquer que si l'on choisit  $l_i = \lceil -\log_2 p_i \rceil$  alors nous avons :

$$\bar{L} = \sum_i p_i l_i < \sum_i p_i (\log_2 p_i + 1) \quad (\text{II.33})$$

$$\bar{L} = \sum -p_i \log_2 p_i + \sum_i p_i \quad (\text{II.34})$$

$$\bar{L} = H(X) + 1 \quad (\text{II.35})$$

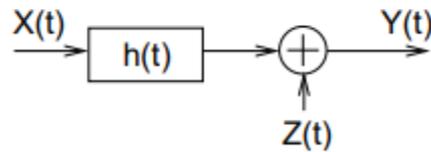
Le résultat du théorème est ainsi établi [12].

## II.4 Codage de canal :

### II.4.1 Canal de transmission :

C'est la liaison câblée ou Hertzienne qui relie l'émetteur au récepteur. Le modèle le plus commun pour modéliser le canal de transmission est le canal linéaire à bruit blanc

gaussien additif (AWGN : Additive White Gaussian Noise). Il est schématisé comme montré par la figure II.8 [12].



**Figure II.9 :** Modèle de canal de transmission

Dans ce schéma  $h(t)$  est la réponse impulsionnelle du filtre caractérisant le canal et  $Z(t)$  est un bruit blanc additif. Nous pouvons écrire donc :

$$Y(t) = X(t) * h(t) + Z(t) \quad (\text{II.36})$$

$$= \int_{-\infty}^{+\infty} X(t - \tau)h(\tau)d\tau + Z(t) \quad (\text{II.37})$$

### II.4.2 Caractéristiques d'un canal:

Un canal se caractérise par :

- **Bande passante:** Limite du taux de modulation (critère de Nyquist).
- **Bruit additif ou impulsionnel :** Limite du prix du signal ou capacité du canal en termes de débit maximal (relation de Shannon-Hartley), limite de capacité du canal en termes de codage (deuxième théorème de Shannon).
- **Atténuation:** l'affaiblissement d'amplitudes [16].

### II.4.3 Codage de canal :

Le codage de canal est une opération de transcoding. Chaque code source est associé à un autre code généralement plus long que celui-ci. En pratique, les codeurs de canal ajoutent une redondance structurée aux informations transmises par le codeur source pour rendre les informations transmises robustes aux erreurs et aux évanouissements introduits par le canal physique.

Le décodeur de canal se fait au niveau du récepteur, permettant ainsi la restitution du message émis par l'application des algorithmes et des méthodes de décodage bien définies pour la détection et la correction des éventuelles erreurs de transmissions [12].

### II.4.5 Le but de codage de canal :

Le but de cette opération est la protection de l'information contre les erreurs de transmission, en introduisant dans le code source une certaine redondance contrôlée. La séquence des codes de canal forme un signal en bande de base qui est transmissible directement ou après modulation.

Notons que le codage de canal sert aussi à adapter le message numérique à transmettre aux caractéristiques du canal de transmission [17].

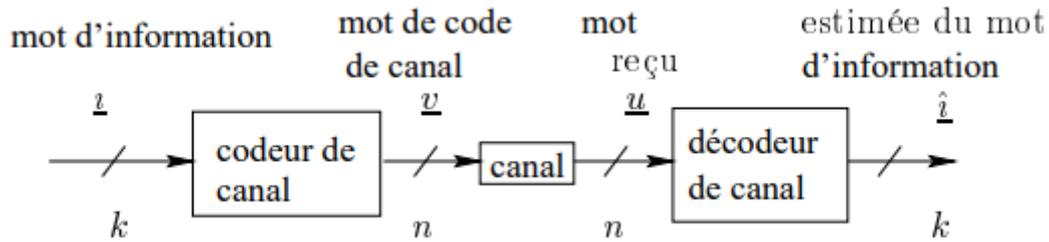


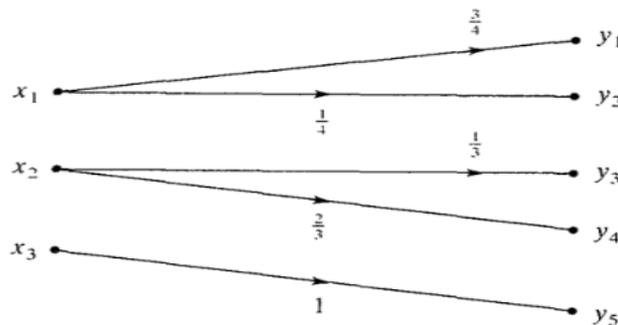
Figure II.10: Paradigme du codage de Canal.

### II.4.6 Principaux Types des canaux :

**II.4.6.1 Canal idéal :** Sans pertes, déterministe ou sans bruit.

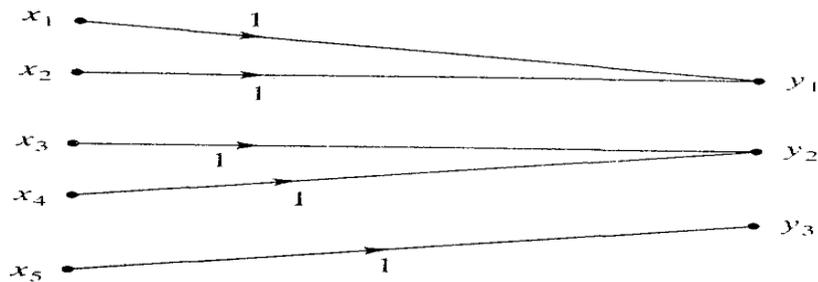
- **Canal sans pertes :** Si la matrice de canaux n'a qu'un seul élément non nul par colonne, on parle de canal sans perte [18].

$$[p(y/x)] = \begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 2/30 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



- **Canal déterministe :** Si la matrice de canal n'a qu'un seul élément non nul par ligne, on l'appelle un canal déterministe. Un canal est dit sans erreur s'il est à la fois sans perte et déterministe.

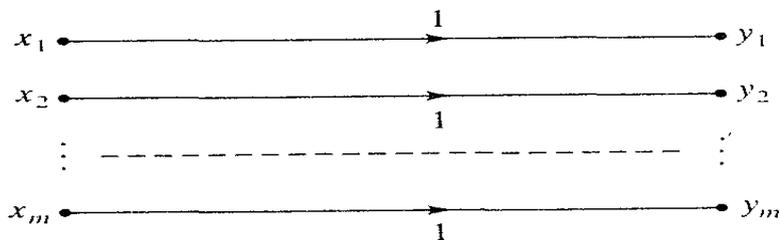
$$[p(y/x)] = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- **Canal sans bruit (sans erreur) :** Dans ce cas, sa matrice a un élément par ligne et colonne, et cet élément ne peut être qu'unique (égal), donc les vecteurs d'entrée et de sortie ont la même dimension.

$$[p(y/x)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pour dimension de 3×3 :

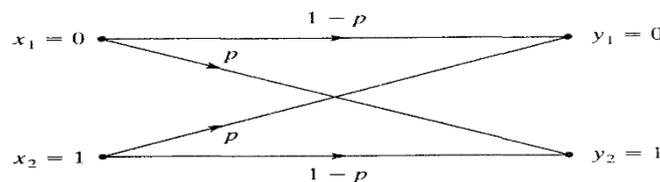


### II.4.6.2 Canal discret stationnaire sans mémoire :

#### ➤ Canal binaire symétrique (BSC) :

Un canal binaire est un canal qui possède deux états ( $x_1 = 0$  et  $x_2 = 1$ ) en entrée et deux à la sortie ( $y_1 = 0$  et  $y_2 = 1$ ). Ce canal est dit symétrique car si un 0 est envoyé, la probabilité de recevoir un 1 est la même que la probabilité de recevoir un 0 après avoir envoyé un 1, et cette probabilité est appelée probabilité d'erreur [18].

$$[p(y/x)] = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$



Ainsi, chaque bit peut être transmis correctement avec une probabilité  $1-P$ , ou corrompu (modifié) avec une probabilité  $P$  d'erreur [18].

#### ➤ Canal binaire à effacement (BEC) :

Le canal est appelé effacement car chaque état de transmission peut être reçu correctement avec une probabilité  $1-P$ , ou effacé sans changement avec une probabilité  $P$  d'erreur [18].

$$[P(Y|X)] = \begin{bmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{bmatrix}$$

### II.4.6.3 Canal continu :

- Canal a bande passante limitée.
- Canal Gaussien (bruit blanc gaussien additif, AWGN) :

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (\text{II.38})$$

- Canal de Rayleigh

$$P(x) = \frac{x^2}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (\text{II.39})$$

## II.5 Taux de codage canal et probabilités d'erreur :

Le taux de codage canal aussi appelé rendement canal et noté  $R_c$  est défini par le rapport suivant :

$$R_c = \frac{k}{n} \log_2 q \quad (\text{II.40})$$

Où :  $\log_2$  désigne la fonction logarithme à base.

La qualité de la transmission se mesure en termes de probabilité d'erreur. On suppose que les mots d'information sont tous équiprobables. La probabilité d'erreur la plus naturelle est la probabilité d'erreur par symbole notée  $P_{es}$  c'est-à-dire la moyenne des  $k$  probabilités qu'un symbole d'information estimé soit différent du Symbole d'information émis [17].

$$P_{es} = \frac{1}{k} \sum_{j=1}^k Pr(\hat{i} \neq i_j) \quad (\text{II.41})$$

Où  $\hat{i}$  et  $i$  sont les mots d'information définis sur la figure II.10 (Figure de Paradigme du codage de Canal).

Cette probabilité est souvent difficile à estimer théoriquement car elle nécessite d'évaluer  $k$  probabilités. Ainsi, le critère de mesure de la qualité de transmission est simplifié en comptant le nombre de fois où le mot de canal  $\hat{u}$  estimé par le décodeur de canal diffère du mot de canal transmis  $u$ .

Définissant ainsi la probabilité d'erreur de chaque mot, notée  $p_{em}$  lorsque le décodeur de canal obtient l'erreur de mot de canal, indiquant qu'au moins un symbole d'information estimé est erroné, et qu'au plus  $k$  symboles estimés sont erronés.

$$\frac{p_{em}}{k} \leq p_{es} \leq p_{em} \quad (\text{II.42})$$

Ce cadre permet à l'ingénieur d'estimer plus simplement la qualité de l'encodage effectué. Les performances du code bloc sur  $C_q$  doivent être comparées aux meilleures disponibles, à l'exception de la longueur du code utilisé. Comme nous le verrons, ce dernier est formulé dans un théorème étonnamment simple, compte tenu de la difficulté d'évaluer la performance exacte de l'ensemble du trajet [18].

### II.6 Capacité du canal :

Le théorème de codage de canal de Shannon stipule que tant que l'efficacité du canal  $R$  est inférieure ou égale à la capacité du canal  $C$ . L'inverse de ce théorème affirme que si l'efficacité du canal est supérieure à la capacité, la probabilité d'erreur diminue d'un terme non nul. La capacité est définie comme la valeur maximale des informations mutuelles entre l'entrée et la sortie du canal.

$$C = \sup_n \max_{p(v)} \frac{1}{n} I(U, V) \quad (\text{II.43})$$

$$I(U, V) = \sum_{u,v} p(u, v) \log_2 \left( \frac{p\left(\frac{u}{v}\right)}{p(u)} \right) \quad (\text{II.44})$$

$I(U, V)$  est appelée information mutuelle entre les deux variables aléatoires  $U$  et  $V$  de dimension  $n$  ayant respectivement pour réalisations  $u$  et  $v$  [18].

### II.7 Conclusion :

Aujourd'hui, nous pouvons considérer que la théorie de l'information existe en tant que domaine propre. Dans ce chapitre nous avons étudié les notions de base de la théorie de l'information, et nous avons donné leurs types, leurs concepts et leurs effets.

Dans la suite nous avons intéressé à la définition d'entropie, de codage de source qu'il sert à fournir une représentation efficace des données (un taux de compression important) tout en préservant l'information essentielle qu'elles portent, de codage de canal qu'il consiste à introduire de la redondance dans le signal émis.

# Chapitre 3

## Techniques De Compression

### III.1 Introduction :

Les méthodes de compression et de codage réduisent le nombre moyen de bits à stocker ou à transmettre par pixel en exploitant la redondance des informations de l'image. À cette fin, un certain nombre d'études scientifiques ont conduit à la naissance de diverses méthodes de compression. Notez que la compression d'image est divisée en deux axes principaux : la compression sans perte et la compression avec perte. La première compression, utilisant uniquement le principe de réduction information et ne conduit pas à la perte, le second type, quand il s'agit d'elle, définit une représentation approximative de l'information. Ce chapitre se limite aux cas particuliers des images fixes et liste les méthodes disponibles pour comprendre les principes et tracer les grandes lignes prévisibles du domaine.

### III.2 Le but de la compression d'image :

La compression consiste à réduire la taille physique (bits) d'un bloc d'informations. Les caractéristiques de l'image doivent être représentées (en éliminant la redondance) Permet de réduire les coûts de stockage et de transférer les données rapidement.

Trois types de redondance sont exploités dans la compression d'images [19] :

- **Redondance spatiale** entre pixels ou blocs adjacents dans une image.
- **Redondance temporelle** entre images consécutives dans une séquence vidéo.
- **Redondance spectrale** entre plans de couleurs ou bandes spectrales.

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

#### III.3.1 Taux de compression :

Ce critère est très important puisqu'il dépend directement de la technique de compression utilisée [20] :

- Il est représenté comme une formule, équation (III.1).
- Dans l'équation (III.1),  $I_0$  est la taille de l'image originale en octet et  $I_c$  la taille de l'image compressée.
- Le taux de compression peut être aussi quantifié par le nombre moyen de bits par pixel ( $bpp$ ), équation (III.2). L'élément  $Bits_{I_c}$  est le nombre total de bits de l'image compressée et  $Pixels_{I_0}$  est le nombre total de bits de l'image compressée et  $Pixels_{I_0}$  est le nombre

total de pixels de l'image originale [20].

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

$$T_C = \frac{\text{Bits}_{I_C}}{\text{Pixels}_{I_0}} \quad (\text{III.2})$$

### III.3.2 Taux d'information (Entropie) :

L'entropie est une quantité qui caractérise la quantité d'informations que contient une image. Par exemple, une image dans laquelle tous les pixels ont la même valeur contient très peu d'informations car Il est très redondant, donc il a une faible entropie. Par contre, toutes les images Les pixels ont des valeurs aléatoires qui contiennent beaucoup d'informations et leur entropie est élevée [22]. L'entropie (en bits) est calculée par la formule suivante [21]:

$$H = - \sum_{i=1}^N P_i \log_2 P_i \quad (\text{III.3})$$

Où :

- $P_i$  est la probabilité de présence du niveau de gris  $i$ .
- $N$  étant le nombre du niveau de quantification ( $N = 256$  pour une image à niveau de gris codée sur 8 bits).

Plus l'entropie est faible, moins on a besoin de bits pour représenter les niveaux de gris des pixels de l'image. Par définition l'entropie d'ordre zéro  $H_0$  est donnée par [5]:

$$H_0 = - \sum_{i=1}^{2^R-1} P_i \log_2 P_i \quad (\text{III.4})$$

Où :

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

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

Deux techniques sont utilisées pour évaluer la distorsion : subjective et objective [20].

**Des méthodes subjectives**, nécessitent des tests psychovisuels de l'œil humain. Les tests sont effectués dans différentes populations.

**Des méthodes objectives**, impliquant la définition de grandeurs, permettant d'évaluer numériquement la qualité des images reconstruites.

La distorsion (D) est l'erreur introduite par l'opération de compression car l'image reconstruite peut ne pas être exactement la même que l'originale. La mesure de distorsion couramment utilisée dans la compression d'image est une erreur quadratique moyen MSE (erreur quadratique moyenne). Cette grandeur est définie par la moyenne des écarts au carré. La relation entre le pixel  $I(m, n)$  de l'image d'origine et le pixel  $\hat{I}(m, n)$  de l'image reconstruite est la suivante [5]:

$$MSE = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N (I(m, n) - \hat{I}(m, n))^2 \quad (\text{III.5})$$

Où :

- $(M \times N)$  est la taille de l'image.
- $I(m, n)$  et  $\hat{I}(m, n)$  sont respectivement les amplitudes de pixel sur les images originale et dégradée. Il est vraisemblable que l'œil tienne beaucoup plus compte des erreurs à grandes amplitudes, ce qui favorise la mesure quadratique.

Le rapport signal sur bruit (Signal to Noise Ratio) est défini par [22] :

$$\text{SNR} = 10 \cdot \log_{10} \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [x(m, n)]^2}{MSE} \quad (\text{III.6})$$

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

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{(2^R - 1)^2}{MSE} \right) \text{dB}(\text{décibels}) \quad (\text{III.7})$$

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

$$\text{PSNR} = 10 \cdot \log_{10} \frac{(255)^2}{MSE} \quad (\text{III.8})$$

Les mesures de distorsion sont très utiles pour déterminer la performance d'une méthode par rapport à d'autres. Ces mesures de distorsion sont objectives et simples à

calculer. Certaines méthodes de compression recherchent le meilleur compromis entre la performance et la distorsion [21].

L'inconvénient de cette méthode est qu'elle ne rend pas compte de la perte de qualité visuelle engendrée par la compression.

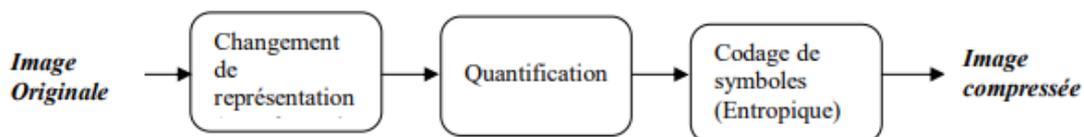
De plus, la MSE par exemple est une mesure globale sur toute l'image, qui gomme les variations locales. Par exemple, dans une image médicale, si des détails anatomiques importants sont dégradés par la compression et si la majeure partie du reste de l'image est fidèlement restituée, alors la MSE est relativement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

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

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

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

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



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

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

- **Transformation ou dé-corrélation :** la dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte sur l'image. La dé-corrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés pour réduire le volume d'information, c'est une opération réversible [5].
- **Quantification :** vise à réduire le nombre des valeurs de la grandeur (l'info), elle fait

partie uniquement des algorithmes irréversibles (consiste à réduire les redondances psychovisuelles, elle peut être scalaire ou vectorielle) [21].

- **Le codage entropique** : le codage entropique effectue un codage sans perte sur les valeurs quantifiées. Cette dernière étape est nécessaire dans les méthodes sans perte, mais elle est souvent présente aussi dans les algorithmes irréversibles, puisque les valeurs transformées et quantifiées contiennent davantage de redondances [21].

### III.5 Classification des méthodes de compression :

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

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

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

- **Compression sans perte** : lorsqu'il n'y a pas de perte de données sur les informations d'origine, la compression est dite sans perte et les informations compressées sont autant qu'avant [21].
- **Compression avec perte** : La compression avec perte ne convient que pour les données perceptives, généralement des données sonores ou visuelles, qui peuvent être modifiées, parfois importantes, mais ne peuvent pas être perçues par l'homme [21]. La perte d'informations est irréversible et il est impossible de retrouver les données d'origine après une telle compression.  
Pour cette raison, la compression avec perte est parfois appelée compression irréversible ou non conservatrice.

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

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

- **Niveau pixel** : Dans cette méthode de visualisation d'une image sous la forme d'un réseau de pixels, l'encodeur reçoit les pixels un par un, les encode et les transmet au décodeur. Cependant, les dépendances existant entre pixels adjacents doivent être exploitées pour améliorer les performances de cette méthode [5]. Nous pouvons donner

quelques exemples, le codage de Huffman, la quantification scalaire et les méthodes sans perte de DPCM (Differential Pulse Code Modulation).

- **Niveau bloc** : Cette stratégie est largement appliquée principalement au travers d'algorithmes irréversibles, et elle consiste à regrouper les pixels en blocs, généralement non superposés et qui recouvrent tous l'image [5]. Cette fois, l'unité de codage est un bloc.

Par conséquent, toutes les opérations sont effectuées au niveau du bloc au lieu du niveau du pixel. Le passage à ce niveau supérieur est important car l'interprétation des blocs offre un éventail de possibilités plus large que les pixels individuels, ce qui est obtenu en exploitant les corrélations intra-blocs. Les standards JPEG et MPEG travaillent par blocs adjacents de 8x8 pixels et une quantification scalaire [21].

- **Niveau scène**: Le contenu de l'image est le plus haut niveau d'information pouvant être utilisé dans un système de compression. Les méthodes de codage de région, la transformée en cosinus discrète DCT, le codage d'image en sous-bande par filtrage ou décomposition en ondelettes sont des exemples de méthodes appliquées au niveau de la scène [21].

### III.5.3 Méthodes Intra et Inter-images :

La troisième classification s'applique aux séquences d'images (séries chronologiques ou séquences de tranches 3D). La méthode intratrame compresse chaque image indépendamment. Les méthodes inter-images exploitent la redondance entre les images consécutives. La norme MPEG encode des séquences d'images en détectant le mouvement d'une image à l'autre [5]

### III.5.4 Méthodes spatiales et méthodes par transformation :

La quatrième catégorie traite des domaines où les opérations de base de la compression sont effectuées. Une image peut être représentée de deux manières strictement équivalentes :

- Dans le domaine spatial, les images sont représentées en pixels. C'est la zone visuellement accessible à l'observateur.
- Dans le domaine fréquentiel, l'image est représentée sous forme de coefficients fréquents.

La transformation d'un domaine à un autre se fait par les transformées mathématiques les plus connues, telles que la transformée de Fourier ou la transformée en cosinus discrète (DCT)

Dans le domaine spatial, les informations contenues dans l'image sont réparties sur toute la matrice de l'image. Mais dans le domaine fréquentiel, l'information est généralement plus concentrée. Cette méthode est largement utilisée, par exemple dans les normes JPEG et MPEG [22].

### III.5.5 Méthodes adaptatives, non adaptatives :

Cette cinquième catégorie indique si le procédé de compression est adaptatif.

- **Méthode non adaptative** : applique les mêmes paramètres de manière fixe, quel que soit le jeu de données à encoder.
- **Méthode adaptative** : Au fur et à mesure de l'encodage, ses paramètres sont modifiés pour s'adapter aux données d'entrée.

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

### III.6 Méthodes réversibles ou sans perte :

Les méthodes sans perte peuvent être appliquées dans le domaine spatial, mais sont plus difficiles dans le domaine fréquentiel. Ce type de compression n'entraîne aucune perte de données. Il est utilisé dans des applications telles que l'imagerie médicale, l'imagerie satellite (les images sont chères et les détails sont importants), le texte, les programmes et toute autre archive de types de données qui doivent être conservées intactes.

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

- **L'efficacité du codage**, qui peut être mesurée en bits par pixel, est limitée par l'entropie de la source. Plus l'entropie de la source est grande, plus elle est difficile à compresser.
- **Le temps d'encodage**, qui est lié à la complexité du processus d'encodage ou de décodage. Si vous augmentez la puissance de calcul du composant de traitement, vous pouvez la diminuer. Pour certaines applications, ce temps est limité, ce qui nécessite le choix de la technologie de codage.
- **La complexité de l'encodeur**, qui peut être mesurée à l'aide de la quantité de ressources utilisées par la mémoire et du nombre d'opérations arithmétiques.

Cette famille d'algorithmes est essentielle dans nos ordinateurs, et tous les programmes d'archivage sont basés sur cette technique de compression sans perte d'informations. Il

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

### III.6.1 Méthodes différentielles et prédictives :

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

Dans les systèmes plus complexes et efficaces, une fonction de prédiction est construite qui estime la valeur d'un pixel en fonction des valeurs des pixels voisins. Ensuite, nous encodons l'erreur de prédiction, qui est la différence entre la vraie valeur du pixel et la valeur prédite. La façon de coder l'erreur de prédiction est généralement basée sur la quantification scalaire, comme nous le verrons plus tard. Le codage prédictif ainsi décrit correspond à la Differential Pulse Code Modulation (DPCM) [22] [24].

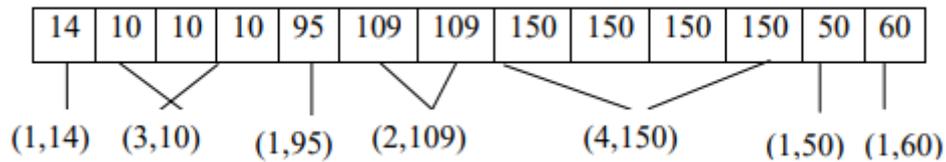
L'efficacité du procédé de prédiction est étroitement liée à l'adéquation de la fonction de prédiction pour les pixels à coder, de sorte que la valeur d'erreur de prédiction est faible. Dans les systèmes d'angiographie cardiaque numérique, Philips MedicalSystems utilise depuis le début des années 1990 un système basé sur la méthode DPCM pour coder les images stockées sur disque dur.

Une méthode différentielle ou prédictive bien connue est la modulation delta (DM). C'est la plus simple de ces méthodes. Le premier pixel est transmis et chaque pixel est prédit comme étant égal à son prédécesseur. Encode l'erreur de prédiction sur 1 bit en ne conservant que les informations sur ce symbole erroné. Chaque pixel est décodé en l'augmentant ou en le diminuant par rapport à sa taille de pas fixe précédente, selon que l'erreur de prédiction était positive ou négative[21].

### III.6.2 Méthodes par plages (Run length Encoding) :

Le principe de la compression RLE est facile à mettre en œuvre. Elle est basée sur le fait que dans une image, plusieurs répétitions d'un même pixel ou séquence de mêmes pixels, toutes juxtaposées. Ainsi au lieu d'encoder chaque pixel de l'image, RLE regroupe les mêmes valeurs adjacentes et ne transmet la valeur qu'une seule fois, précédée du nombre de répétitions. Évidemment, cette méthode fonctionne bien s'il y a beaucoup de répétitions dans l'image [22].

Cet algorithme très simple donne des taux de compression plus élevés. Un exemple de traitement RLE est donné à la figure III.2.



**Figure III.2:** Un exemple de codage par plage RLE.

Si l'encodage des octets à répéter est relativement simple, et se répète sur un nombre de bits suffisant, cette approche peut être très pénalisante dans certains cas : si deux octets consécutifs sont toujours différents, la taille des données compressées sera gonflée par rapport aux données d'origine.

Pour éviter cet inconvénient, les versions RLE de pointe utilisent un code différenciateur pour indiquer le début de la séquence d'octets à répéter plus le nombre de répétitions, les octets orphelins sont toujours codés dans leur forme originale. Il est important de noter qu'il existe plusieurs variantes de cet encodage, certains types encodent des images en détectant des séquences redondantes basées sur des lignes, des colonnes ou encore des zigzags [16].

Il est important de noter qu'il existe plusieurs variantes de cet encodage, certains types encodent des images en détectant des séquences redondantes basées sur des lignes, des colonnes, ou même en Zig-Zag [22].

### III.6.3 Méthodes par dictionnaire adaptatif (Lempel-Ziv) :

C'est une technique d'encodage utilisant un dictionnaire. Nous recherchons les chaînes en double dans un fichier en stockant les chaînes en double dans un dictionnaire. L'encodage consiste alors à remplacer les chaînes mémorisées par des adresses (ou indices) construits dans le dictionnaire [23].

Le premier algorithme de ce type a été développé par Lempel et Ziv en 1977 et a abouti au programme LZ77. Cela lit le flux de symboles et recherche la chaîne équivalente dans la fenêtre 4K avant le flux d'entrée. Les équivalents sont remplacés par du code. La compression des formats ZIP, ARJ et LHA est basée sur cet algorithme [22].

Lempel et Ziv ont développé en 1978 une nouvelle version de leur algorithme, LZ78, où le dictionnaire a été construit à partir de tous les symboles précédemment rencontrés plutôt qu'à travers une fenêtre, il est spécialisé dans la compression d'images et de fichiers binaires. Puis en 1984, Terry Welch modifia le format LZ78 pour pouvoir l'utiliser dans les contrôleurs de disques durs, le format LZW est né.

Le dictionnaire dans ce cas est initialement construit et contient l'ensemble des codes ASCII. Il est élaboré au fur et à mesure. Ce qui permet de changer la taille du dictionnaire au cours du codage [15]. Le codage LZW est une technique de compression réversible qui peut être appliquée à tout type de fichier de données, que ce soit : texte, image, fichier informatique, etc. Elle a été adoptée pour la mise en œuvre du format de compression d'images 'GIF' [22].

### III.6.4 Codage arithmétique :

Contrairement aux algorithmes de Huffman et de Shannon-Fano, ce dernier associe des symboles à des motifs binaires dont les grandeurs dépendent de leur distribution. Le codeur arithmétique traite le fichier dans son ensemble, en l'associant à un seul nombre décimal rationnel [24]. Ce code de sortie est un nombre à virgule flottante entre 0 et 1 avec le nombre de chiffres après la virgule correspondant au nombre de symboles. Les encodeurs arithmétiques sont plus efficaces que les encodeurs Huffman, mais sont plus complexes à mettre en œuvre [25].

**Algorithme du codage arithmétique :** Nous décrivons brièvement ci-dessous l'algorithme de codage arithmétique pour illustrer le principe, sachant que le décodage opère dans le sens inverse [22].

- Calculer la probabilité associée à chaque symbole de la chaîne à coder.
- Associer à chaque symbole un sous-intervalle proportionnel à sa probabilité, dans l'intervalle [0,1[ (l'ordre des intervalles sera retenu car requis par le décodeur).
- Initialiser la borne inférieure de l'intervalle de travail à la valeur 0 et la borne supérieure à la valeur 1.
- Tant qu'il reste un symbole dans la chaîne à coder :
  - $f$  largeur = limite supérieure - limite inférieure
  - Limite inférieure = limite inférieure + largeur  $\times$  (limite basse du sous intervalle du symbole)

- Limite supérieure = limite inférieure + largeur x (limite haute du sous intervalle du symbole).

A la fin, toute valeur dans l'intervalle de fin représente de manière unique la séquence d'entrée. Généralement, la limite inférieure de l'intervalle est sélectionnée comme représentative de la série, et la valeur moyenne peut également être sélectionnée. Il est à noter qu'à la fin du processus de codage, le codeur arithmétique génère un fichier composé d'un en-tête suivi de la représentation binaire du code choisi. De plus, l'en-tête peut contenir soit la table des probabilités, soit la table des fréquences de tous les symboles [26].

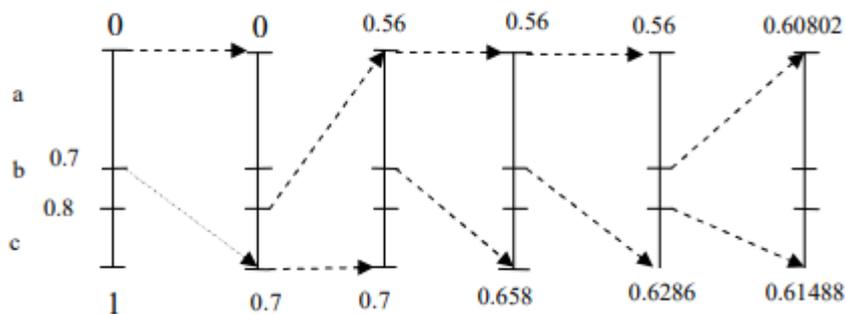
### Exemple :

Supposons que l'on veuille coder une partie "acaab" d'une longue séquence avec une probabilité d'apparition indiquée dans le Tableau III.1 [20].

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

**Tableau III.1 :** Probabilités des symboles.

La figure III.3 donne une représentation du processus de codage arithmétique de la chaîne "acaab".



**Figure III.3 :** Génération de l'étiquette pour la séquence "acaab".

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

### III.7. Méthodes avec perte ou irréversible :

Contrairement aux méthodes de compression sans perte, ce type implique une perte de données lors de l'encodage/décodage. Le résultat qui peut être obtenu est une version modifiée de l'image originale. Le but de cette compression est de supprimer le plus d'informations possible sans dégrader la qualité d'image perçue par le système visuel humain [22]. Avec ces méthodes, on peut aussi distinguer :

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

#### III.7.1 Les méthodes spatiales :

- **Sous-échantillonnage** : Le sous-échantillonnage consiste à ne conserver qu'une partie des données [19].
- **Le codage par troncature des blocs (BTC)** : BTC est une technique simple de compression des images au niveau de gris développé par Delp et Mitchell [27].

**L'algorithme de codage :**

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

**Étape2 :** Pour chaque bloc, calculer la moyenne ( $\bar{X}$ ) et l'écart type ( $\sigma$ ), tels que [27]:

$$\bar{X} = \frac{1}{k} \sum_{i=1}^k x_i \quad (\text{III.10})$$

$$\sigma^2 = \frac{1}{k} \sum_{i=1}^k x_i^2 - \bar{X}^2 \quad (\text{III.11})$$

Où :

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

**Étape 3 :** Construire le modèle de  $q$  niveaux (table binaire) de taille  $n \times n$ , si la valeur du pixel est supérieure à la moyenne ( $\bar{X}$ ) alors la valeur du pixel est égale à 1 sinon elle est égale à 0.

**Étape 4 :** Calculer les niveaux de reconstruction (a et b).

$$a = \bar{X} - \sigma \sqrt{\frac{q}{k-q}} \quad (\text{III.12})$$

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

Où :

- $\sigma$  est l'écart type.
- $k$  est le nombre total de pixel dans le bloc.
- $q$  est le nombre de pixels plus grands que  $\bar{X}$ .

**Etape 5 :** Stocker  $a$ ,  $b$  et la table binaire associés à chaque bloc.

### L'algorithme de décodage :

Pour reconstruire l'image ou en créer une approximation, les éléments égaux à 0 sont remplacés par la valeur de  $a$ , et les éléments égaux à 1 sont remplacés par la valeur de  $b$ . Afin d'améliorer la qualité, la méthode BTC comporte plusieurs modifications, nous pouvons donner un exemple :

- La méthode AMBTC (Absolute Moment BTC) qui utilise le critère de la conservation des moments absolus des échantillons. Elle s'appuie sur l'idée du codage par troncature de blocs, mais elle est plus simple dans sa mise en œuvre pratique.
- Une autre variante de BTC est basée sur la prédiction qui donne une bonne qualité d'image en bas débit binaire avec un coût minimal dedans la procédure de codage et décodage [28].
- Un BTC de trois niveaux est également employée pour améliorer la qualité des images reconstruites.
- Ces variantes de la technologie BTC peuvent être appliquées séparément à chaque plan de couleur de l'image. Puisqu'il existe un degré élevé de corrélation entre les plans de couleur, cela peut être exploité pour réduire le débit binaire. La méthode (SBBTC) développée par Wu et Coll (d'après [27]) exploite cette corrélation inter-couleurs.

### III.7. 2 Les méthodes par transformation :

#### III.7. 2.1. Principe général :

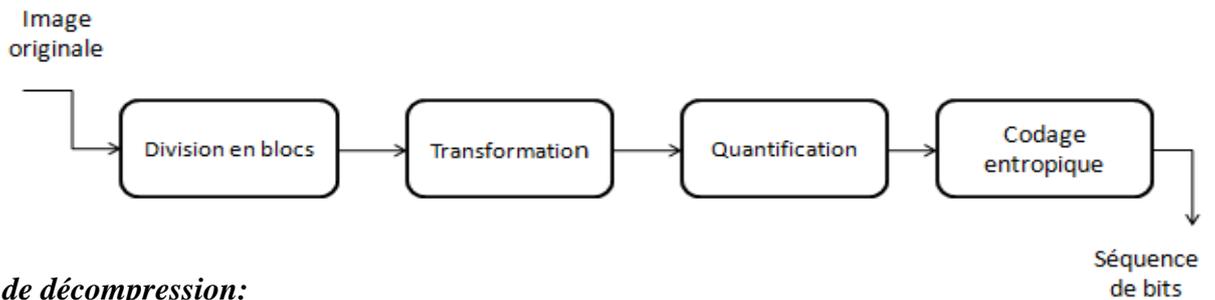
Les méthodes de transformation sont l'une des techniques de compression les plus utilisées, et elles n'agissent pas directement sur l'image numérique dans sa représentation canonique, mais sur son domaine de transformation. Ils peuvent atteindre des taux de compression élevés tout en conservant une bonne qualité d'image. Ces méthodes invoquent à leur tour plusieurs principes de compression. Ils sont utilisés par les normes internationales d'encodage d'images fixes et de vidéos (JPEG et MPEG) [19] [22].

En général, les schémas de codage par transformation subdivisent une image  $N \times N$  en sous-images plus petites avant de transformer chacune de ces sous-images. Les objectifs de ces transformations sont doubles :

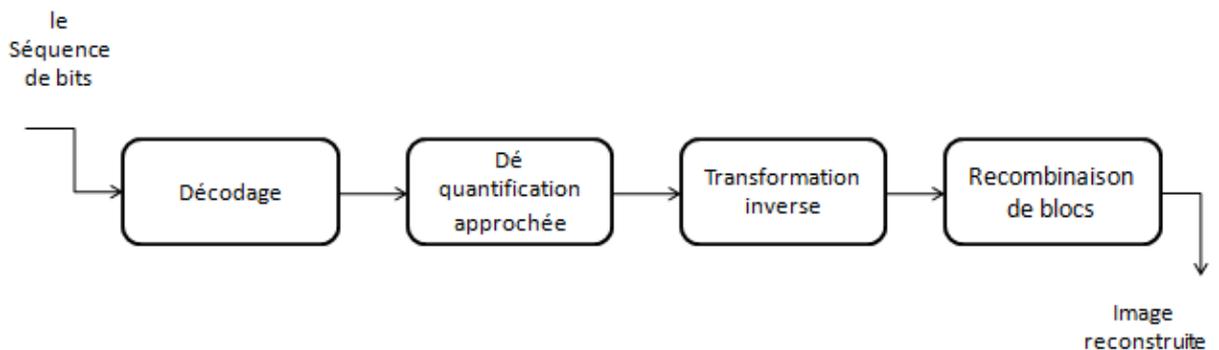
- Il s'agit de dé-corréler les données, c'est-à-dire d'obtenir des coefficients de transformée moins corrélés que les pixels de l'image d'origine.
- Concentrer l'énergie sur un nombre réduit de coefficients, les coefficients ayant une valeur plus importante aux basses fréquences qu'aux hautes fréquences.

Dans ce cas, on obtiendra une compression effective en codant finement les coefficients des basses fréquences, et grossièrement, voire en supprimant, les coefficients des hautes fréquences. Les méthodes par transformation suivent le schéma de fonctionnement présenté dans la Figure III.4.

*Etape de compression :*



*Etape de décompression:*



**Figure III.4 :** Schéma de principe de la compression / décompression par transformation.

### 1- Division en blocs:

La première étape consiste à diviser l'image en blocs sur lesquels vont s'appliquer indépendamment. La principale raison de ce découpage est de limiter le nombre de pixels à traiter à la fois pour diminuer les temps de calcul. Il peut résulter de cette division un effet visuel appelé effet de "blocs" ou "artifacts" : à des taux élevés, les frontières des blocs deviennent visibles, car ils ont été comprimés indépendamment. La taille des blocs est variable selon les méthodes. Elle est de 8x8 pour JPEG [22].

### 2- Transformation :

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

Parmi les principales transformations linéaires les plus connues en compression d'images, on peut citer la transformation de Karhunen-Loeve (TKL), la transformation de

fourrier discrète (TFD), la transformation en cosinus discrète (DCT), la transformation par ondelette discrète (DWT) et d'autres méthodes telles que les transformations de Hadamard (TH) et Haar (THA) [22]. Dans les sections suivantes on présente les transformations d'espace les plus courantes TFD, DCT et DWT.

### 3- Quantification et codage

La troisième étape est la quantification des coefficients transformés, afin de se ramener à un nombre limité de valeurs différentes. La quantification est souvent précédée d'une pondération psychovisuelle des coefficients, afin de préserver ceux auxquels l'œil est le plus sensible. On distingue en général deux types de quantification : la quantification scalaire et la quantification vectorielle [29].

- **La quantification scalaire:** La quantification scalaire SQ-(ScalarQuantization) consiste à remplacer un nombre très grand de symboles par un nombre restreint de codes [19]. C'est une opération irréversible très largement employée en compression. Par exemple, le fait d'arrondir des nombres flottants aux entiers les plus proches est une quantification scalaire.  
D'une manière générale, on peut la définir comme étant l'association de chaque valeur réelle  $x$ , à une autre valeur  $q$  qui appartient à un ensemble fini de valeurs. La valeur  $q$  peut être exprimée en fonction de la troncature utilisée : soit par l'arrondi supérieur, l'arrondi inférieur, ou l'arrondi le plus proche [16].
- **Quantification Vectorielle :** Le principe général du quantificateur vectoriel VQ-(VectorQuantization) consiste à quantifier une suite d'échantillons (vecteur) d'une image au lieu de quantifier chaque valeur, comme c'est le cas pour la quantification scalaire.

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

La transformée de Fourier discrète (TFD) est un outil mathématique de traitement du signal numérique. Elle est l'équivalent discret de la transformée de Fourier continue utilisée pour le traitement du signal analogique. Sa définition mathématique pour un signal  $S$  de  $N$  échantillons est la suivante [21] :

$$S(k) = \sum_{n=0}^{N-1} s(n) \cdot e^{-2i\pi \frac{kn}{N}} \quad (\text{III.14})$$

$k$  : le temps,  $n$  : fréquence

La transformée inverse est donnée par :

$$S(k) = \frac{1}{N} \sum_{n=0}^{N-1} S(n) \cdot e^{2i\pi \frac{nk}{N}} \quad (\text{III.15})$$

Cette transformation mathématique transforme un ensemble de données d'un domaine spatial au domaine fréquentielle. Elle est très utilisée en traitement du signal et de l'image, et spécialement en compression [28].

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

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

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

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

Les équations qui suivent, donnent respectivement les transformées en cosinus discrètes directe et inverse [29].

$$DCT(i, j) = \frac{1}{\sqrt{2N}} c(i)c(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{pixel}(x, y) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right) \quad (\text{III.16})$$

$$\text{Pixel}(i, j) = \frac{1}{\sqrt{2N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} c(i)c(j) DCT(x, y) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right) \quad (\text{III.17})$$

Avec

- $c(i) = 1/\sqrt{2}$  si  $i=0$ .
- $c(i) = 1$  si  $i > 0$

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

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

En pratique, l'image couleur est généralement codée en RGB, et traitée comme trois images monochromes ou plans. La DCT est appliquée successivement aux données du plan de R, G et B. Pour illustrer la transformation DCT on prend l'exemple suivant :

$$F = \begin{pmatrix} 226 & 226 & 223 & 223 & 226 & 226 & 228 & 227 \\ 226 & 226 & 223 & 223 & 226 & 226 & 228 & 227 \\ 226 & 226 & 223 & 223 & 226 & 226 & 228 & 227 \\ 226 & 226 & 223 & 223 & 226 & 226 & 228 & 227 \\ 226 & 226 & 223 & 223 & 226 & 226 & 228 & 227 \\ 227 & 227 & 227 & 22 & 226 & 228 & 226 & 230 \\ 228 & 228 & 225 & 224 & 225 & 29 & 229 & 229 \\ 223 & 223 & 226 & 221 & 227 & 225 & 226 & 228 \end{pmatrix} \Rightarrow \begin{pmatrix} 1806.7 & -1.1 & 8.2 & 4.2 & 1.7 & 3.6 & -1.2 & 2.5 \\ -1.5 & 0.4 & 0 & 1.9 & 0.8 & 1.2 & -3.4 & -0.9 \\ -0.7 & -0.8 & -0.7 & -0.9 & -0.1 & -0.2 & 0.9 & 1.2 \\ 2.7 & 1.0 & 1.4 & 0.2 & -0.5 & -0.5 & 1 & -0.3 \\ -3.2 & -0.9 & -1.5 & -0.2 & 0.8 & 0.2 & -1 & 1.6 \\ 2.4 & 0.5 & 1.1 & 0.8 & -0.6 & 0.7 & -0.4 & -0.7 \\ -1.1 & -0.1 & -0.6 & -0.3 & 0.3 & -1.4 & 1.7 & -1.5 \\ 0.2 & -0.1 & 0.2 & 1 & -0.1 & 1.1 & -1.6 & -0.9 \end{pmatrix}$$

Matrice d'origine
Matrice transformée DCT

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

Les ondelettes c'est d'abord une théorie mathématique récente d'analyse du signal, développée dans les années 80. On peut considérer qu'il s'agit d'une extension de l'analyse de Fourier. L'intérêt de cette théorie est au départ l'analyse des signaux [30].

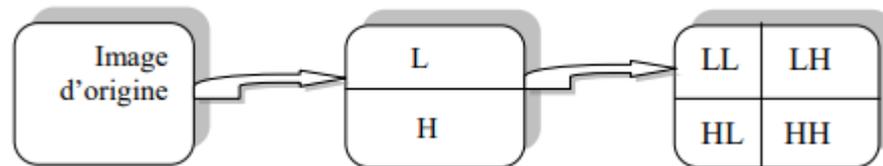
La technologie de compression à base d'ondelettes offre une plus grande finesse au niveau de l'analyse du signal, et permet de mieux s'adapter aux propriétés locales de l'image. La transformation par ondelette est une technique de compression d'image fixe très performante [22]. Son intérêt par rapport aux autres transformées de compression est que celle-ci ne considère pas l'image dans son ensemble pour la coder mais, la travaille par couche, cherchant à enregistrer les détails les plus importants à chaque résolution [19].

La transformée en ondelette repose sur le fait que toute fonction f intégrable, ce qui se note couramment  $\int_{-\infty}^{+\infty} f(x)dx$ , peut être vue sous la forme d'une somme de fonction  $\psi_{a,b}(x)$ , Les fonctions ondelettes, toutes issues d'une seule et unique fonction, l'onde mère de manière à ce que, pour tout b réel et a strictement positif, et pour tout x réel [19]:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \quad \text{(III.18)}$$

- **Algorithme de compression par ondelettes (Wavelets):**

Passons maintenant à l'algorithme pyramidal utilisé. La décomposition en coefficients d'ondelettes n'utilise pas une fonction de moyenne, mais s'appuie sur deux filtres. Un filtre passe bas (L) et un filtre passe haut (H). La combinaison de ces filtres permet d'obtenir quatre sous images HH, HL, LH et LL. Ces filtres sont nommés filtres miroirs en quadrature (figure III.5).



**Figure III.5 :** Transformation en colonnes et en lignes.

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

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

- **Décompression ondelettes :**

La transformation inverse par ondelettes reconstruit une image approximée. La reconstruction de l'image à partir des sous-bandes restitue l'image en mode progressif. L'affichage de l'image s'effectue selon le mode suivant :

La résolution de l'image augmente au fur et à mesure de la lecture du fichier compressé comme illustré sur la figure III.6.



Figure III.6 : Décompression progressive.

### III.8 Les normes de compression des images :

#### III.8.1 La norme de compression JPEG :

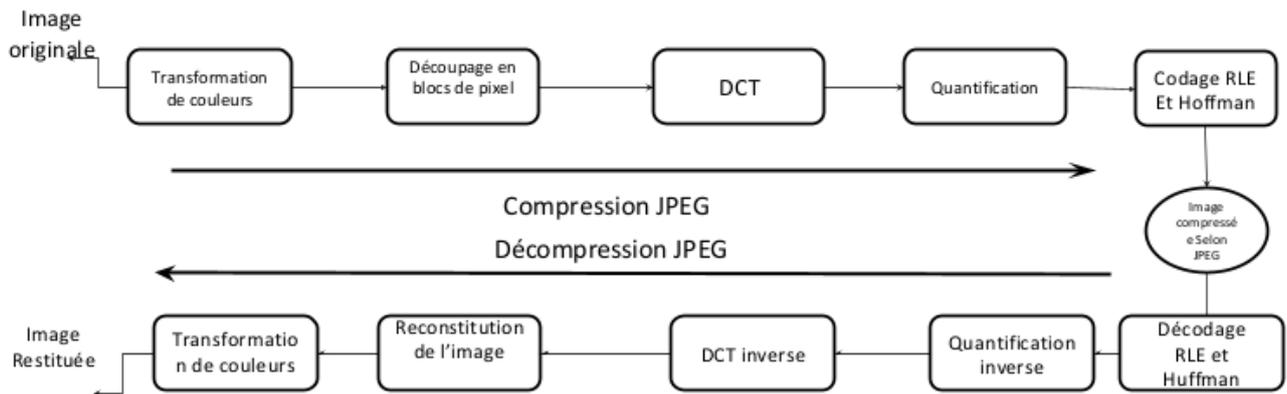
La norme JPEG (Joint Photographic Experts Group) est conçue par le groupe ISO (International Standards Organisation) et le groupe CEI (Commission Electronique Internationale). Elle est destinée à la compression des images fixes en couleurs et à niveaux de gris en vue de leurs stockages sur les supports numériques [22].

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

Les techniques définies par la norme JPEG se divisent en deux classes : les méthodes de compression avec pertes qui sont basées sur la DCT suivie d'une quantification et d'un codeur entropique. La seconde classe, concerne les processus de codage sans pertes, cette classe de codeur n'est pas basée sur la DCT mais sur le codage DPCM suivi d'un codage entropique.

##### - Le principe de compression de la norme JPEG non réversible:

La figure III.7 : explique clairement les différentes étapes de l'algorithme de compression JPEG non réversible.



**Figure III.7 :** Principe de l'algorithme JPEG avec perte.

Cette compression se décompose en six étapes :

- 1- **Découpage en blocs :** le format JPEG, comme le font généralement les algorithmes de compression avec perte, commence par découper l'image en blocs ou carreaux généralement carrés de 64 (8 x 8) ou 256 (16 x 16) pixels.
- 2- **Transformation de l'espace de couleurs :** habituellement, les images couleurs sont stockées sous le format RGB [22].
- 3- **Sous échantillonnage :** la façon la plus simple d'exploiter la faible sensibilité de l'œil à la chrominance est simplement de sous-échantillonner les signaux de chrominance. Généralement on utilise un sous-échantillonnage de type 2h1v ou 2h2v. Dans le premier cas (le plus utilisé) on a un sous-échantillonnage 2:1 horizontalement et 1:1 verticalement, dans le deuxième cas on a un sous-échantillonnage 2:1 horizontalement et verticalement. Ces sous-échantillonnages sont utilisés pour les chrominances. Pour la luminance on n'utilise jamais de sous échantillonnage.
- 4- **Transformée DCT :** on applique la DCT aux blocs de 8x8 pixels. Le calcul d'une DCT est complexe. C'est l'étape qui coûte le plus de temps et de ressources dans la compression et la décompression JPEG, mais c'est peut-être la plus importante car elle permet de séparer les basses fréquences et les hautes fréquences présentes dans l'image [5].
- 5- **Coupage et quantification des coefficients issus de la DCT :** la quantification est la phase non conservatrice du processus de compression JPEG. Elle amène, moyennement à une diminution de la précision de l'image, mais aussi à réduire le nombre de bits nécessaires au stockage [16].

La quantification consiste à diviser cette matrice par une autre, appelée matrice de quantification, et qui contient 8x8 coefficients choisis par le codeur. Habituellement, la matrice de quantification  $Q$  est obtenue à partir de formules plus ou moins simples, permettant de choisir la perte de qualité. La formule suivante est un exemple qui donne de bons résultats [20] :

$$Q = q(i, j) \text{ avec } q(i, j) = 1 + K(1 + m(i + j)) \quad (\text{III.19})$$

Où :

- $i$  l'indice de ligne
- $j$  l'indice de colonne
- $m$  une constante (souvent égale à 1)
- $K$  le facteur de qualité.

La quantification est décrite par l'équation suivante :

$$Q_{\text{quantifié}}(i, j) = \frac{\text{valeurDCT}(i, j)}{q(i, j)} \quad (\text{III.20})$$

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

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

6- **Compression RLE et Huffman** : le codage s'effectue en zigzag comme indiqué à la figure III.8. Après la phase de quantification un grand nombre de valeurs de la matrice résultante se trouvent réduites à 0. Le fait qu'il y ait beaucoup de valeurs mises à 0 va nous permettre de compresser encore mieux grâce à la compression RLE.

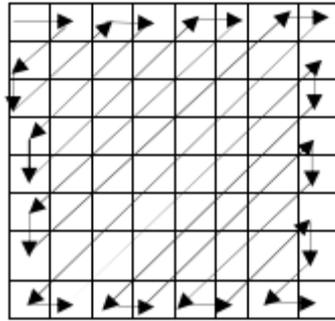


Figure III.8 : Le parcours d'un bloc en zigzag.

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

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

- **Algorithme JPEG réversible:**

La seconde classe, concerne les processus de codage sans pertes, cette classe de codeurs n'est pas basée sur la DCT mais sur le codage DPCM suivi d'un codage entropique (Huffman ou arithmétique). le traitement se fait sur l'image entière, et non pas sur des blocs. Cet algorithme sert surtout pour la transmission d'images médicales pour éviter de confondre des artefacts avec de réels signes pathologiques. La compression est alors beaucoup moins efficace (facteur 2 seulement) [5].



(a) image originale



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



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

Figure III.9 : Exemple d'image compressée avec JPEG.

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

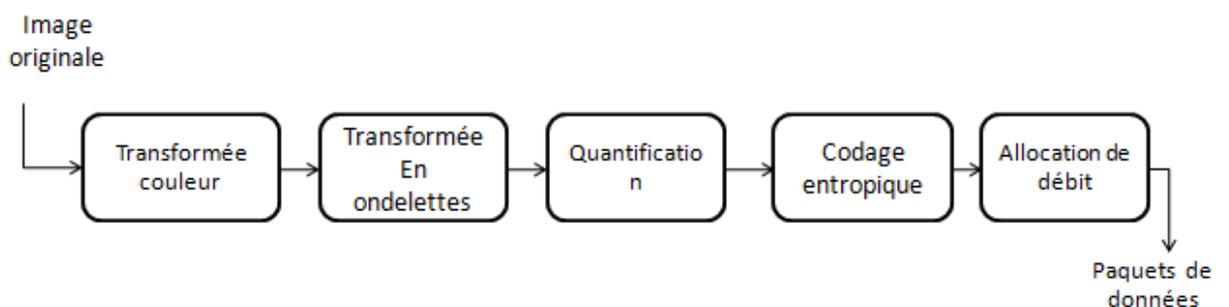
### III.8.2.La norme de compression JPEG 2000:

C'est une norme de compression d'images produite par le groupe de travail Joint Photographic Experts Group. JPEG 2000 est capable de travailler avec ou sans perte, utilisant une transformation en ondelettes. En compression irréversible, JPEG 2000 est plus performante que la méthode de compression JPEG ISO/CEI 10918-1 (JPEG baseline). On obtient donc des fichiers d'un poids inférieur pour une qualité d'image égale. De plus, les contours nets et contrastés sont mieux rendus [26].

Ce nouveau standard a pour objectif d'offrir de nouvelles fonctionnalités permettant de répondre à une demande croissante, à savoir [22] :

- Obtenir des performances de compression supérieures à son prédécesseur JPEG, notamment pour des débits très faibles.
- Permettre d'organiser le fichier compressé de plusieurs manières, notamment en fonction de la résolution désirée ou de la qualité de reconstruction.
- D'autres fonctionnalités les plus marquantes de cette norme sont, d'une part, la possibilité de compresser sans perte ou avec perte en utilisant le même algorithme.
- Permet de ne décoder uniquement qu'une certaine zone d'une image lorsque celle-ci est très grande. Elle permet encore la plupart des manipulations géométriques élémentaires (flip, permutation paysage/portrait...) directement dans le domaine compressé et sans aucune perte en qualité.
- Permet une affectation non uniforme du budget de qualité dans une image.
- En JPEG 2000 le même algorithme de codage s'applique à une grande variété de types d'images (couleurs, niveaux de gris, multi-composantes...) [22].

- **Principe général :**



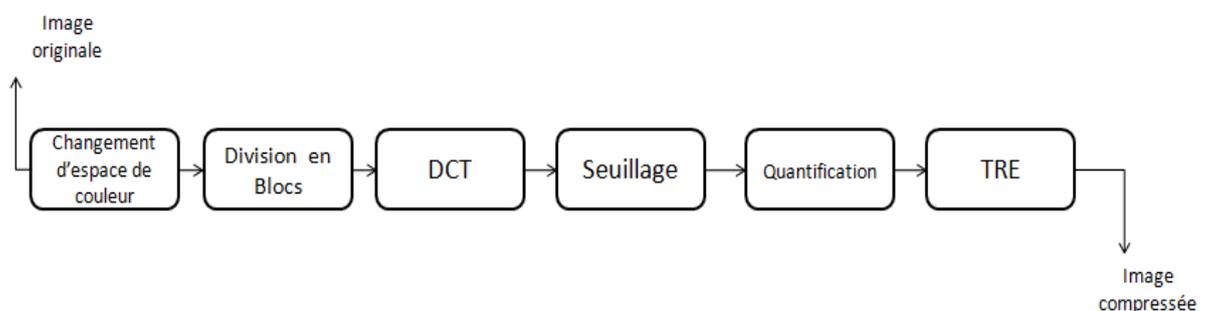
**Figure III.10 :** Schéma typique d'un codeur JPEG 2000

Les nouveautés par rapport à JPEG du point de vue compression sont l'utilisation d'une transformée en ondelettes, qui offre une scalabilité naturelle, mais surtout d'un algorithme de codage entropique très sophistiqué. Celui-ci est fortement basé sur l'algorithme EBCOT [26] de David Taubman. Il consiste en un regroupement et une modélisation des coefficients ondelettes qui fournissent à un codeur arithmétique adaptatif un train binaire possédant les propriétés statistiques adéquates.

Il s'ensuit une étape d'allocation de débit qui permet de respecter le débit cible, et dont le travail est facilité par le partitionnement du train binaire formé par EBCOT. La dernière étape est la mise en forme syntaxique du CodeStream JPEG 2000, avec la formation des paquets, puis la syntaxe haut niveau, particulièrement abondante dans JPEG 2000.

Dans la norme JPEG 2000 un codestream est l'ensemble des données formées par les données images compressées regroupées dans des paquets ainsi que la syntaxe de haut niveau : en-têtes de tuiles, en-tête principale. Les métadonnées du format de fichier JP2 ne font pas partie du codestream. JP2 encapsule le codestream JPEG 2000 dans un format de fichier [31].

Le schéma général de cette compression est présenté dans la figure III.16



**Figure III.11** : Schéma du principe de la compression.

### III.9 Conclusion :

Dans ce chapitre, nous avons présenté plusieurs techniques de compression d'images fixes sans et avec perte (LZW, JPEG, JPEG 2000, ...) ainsi que des notions générales telles que les mesures de performance et de distorsion. Nous avons donné quelques concepts et mesures pour la compression d'images. Dans le chapitre suivant nous proposons une implémentation modifiée de la compression JPEG.

# Chapitre 4

## **Implémentation et analyse**

### **IV.1 Introduction:**

Comme il est mentionné dans les chapitres précédents, le but principal de la compression est de réduire la taille d'information d'image sans altérer le contenu utile. Cela optimise les ressources informatiques et électroniques dédiées au traitement, au stockage (sauvegarde) et à la transmission. Dans ce chapitre, les étapes clés incorporées dans la compression JPEG sont implémentées sous le MATLAB.

Le présent chapitre est organisé en trois sections principales. La première section est pour décrire l'image de test utilisée lors de ce travail, son type, sa dimension, son histogramme et autres caractéristiques. Dans la deuxième section, les différents blocs de traitement de la compression/décompression JPEG sont illustrés et expliqués. Pour donner une aperçue claire, le résultat de chaque bloc est montré sous forme d'une portion d'image ou une série (séquence) d'information. Au fur à mesure, nous avons expliqué le principe de chaque technique impliquée dans le traitement JPEG. La troisième section est consacrée à analyser l'effet (la perte) de compression étudiée sous deux angles : Sans quantification c-à-d seulement en fonction de point de coupure, et Avec quantification c-à-d en fonction de pas de quantification. La perte produite est analysée selon les deux critères de déviation : MSE et SSIM.

### **IV.2 L'image de test :**

L'image de test utilisé dans notre travail est de format RGB (Red, Green and Blue) (Rouge Vert et Bleu). La dimension de chaque matrice (chaque composante) est 564 x 564 pixels. La figure IV.1 montre cette image de test, (a) l'image entière, (b), (c) et (d) respectivement ses trois images composantes : rouge, verte et bleu. La figure IV.2 montre l'histogramme correspondant, la distribution est non-uniforme ce qui donne un appui de crédibilité à notre analyse.

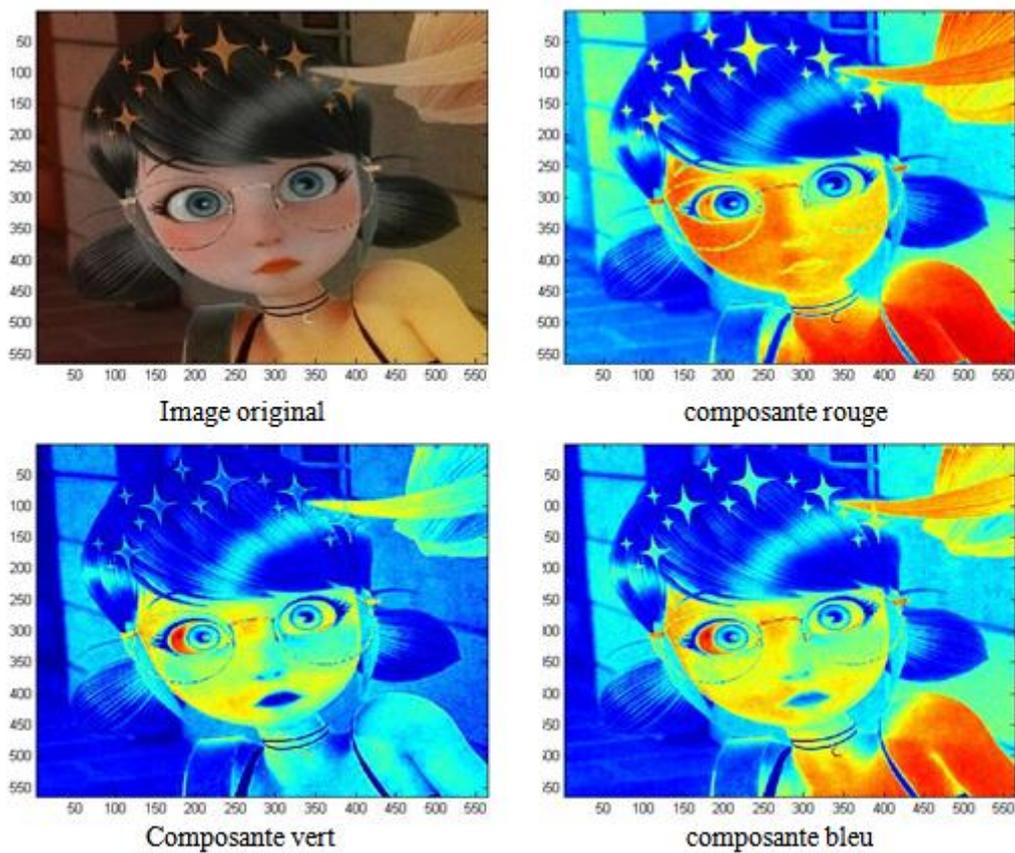


Figure IV.1 : L'image de test et ses trois composantes.

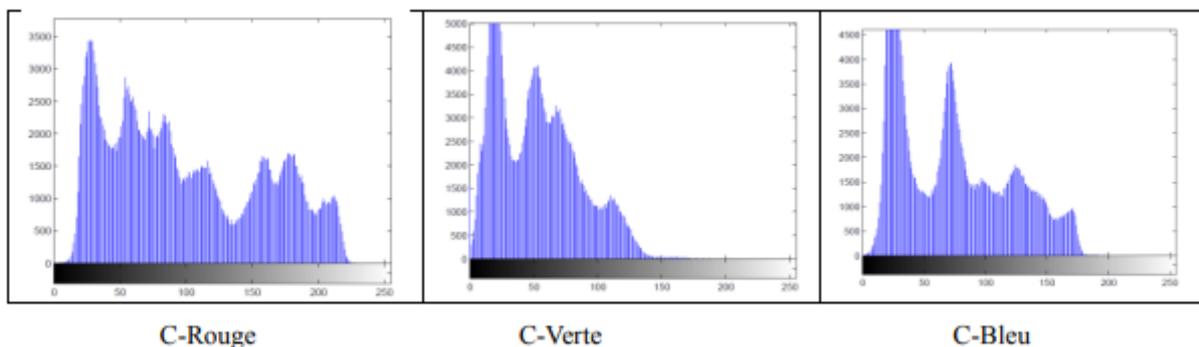
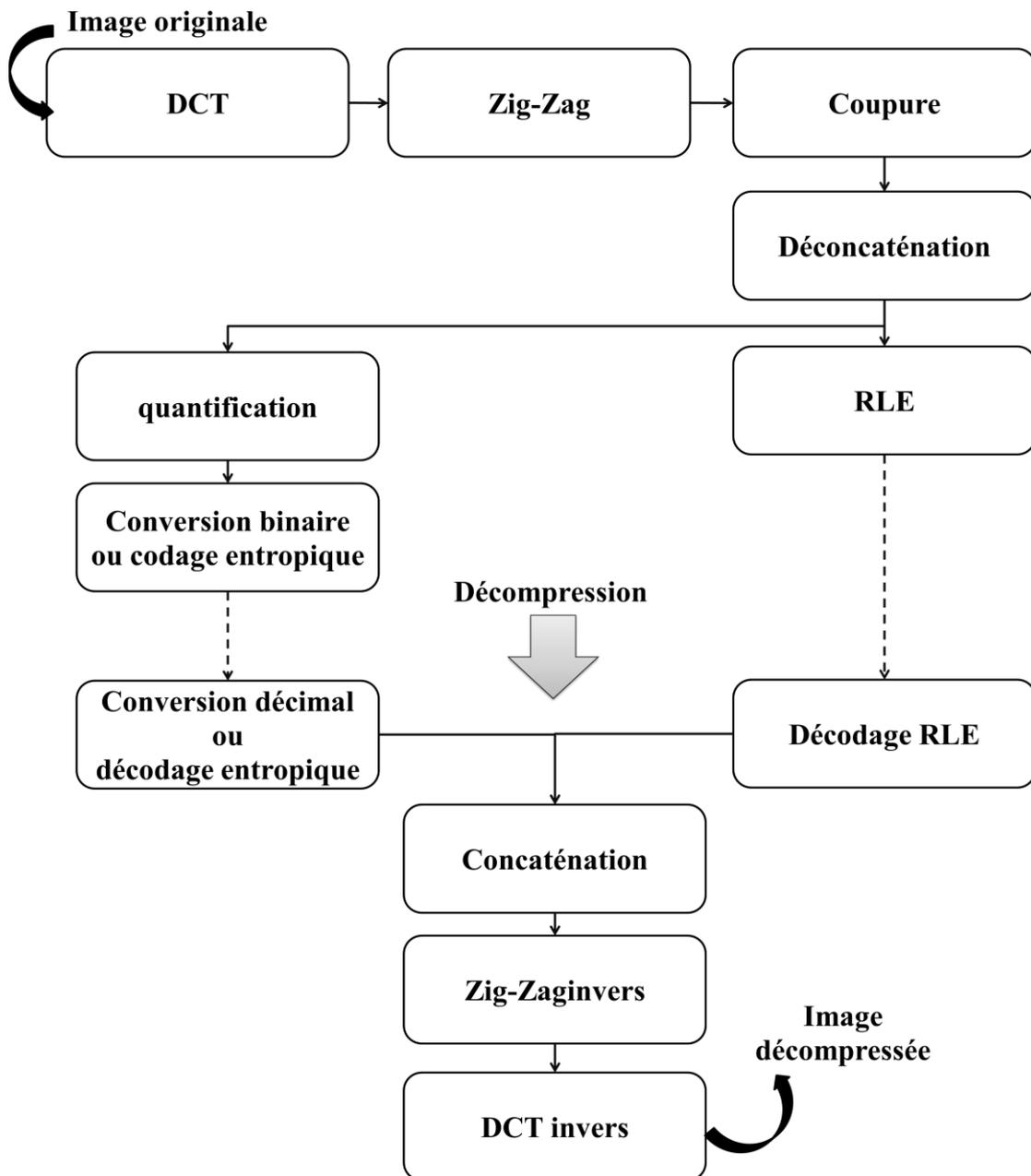


Figure IV.2 : L'histogramme de l'image de test.

### IV.3 Les étapes d'implémentation :

Dans cette section nous allons présenter les étapes principales d'implémentation de la compression JPEG. Les étapes de la compression et également la décompression sont illustrées en schéma bloc de la figure IV.3. Chaque composante est traitée comme image originale indépendamment et elle doit suivre les mêmes étapes. Donc, comme l'image RGB est tri-matricielle, la forme compressée est tri-séquentielle.

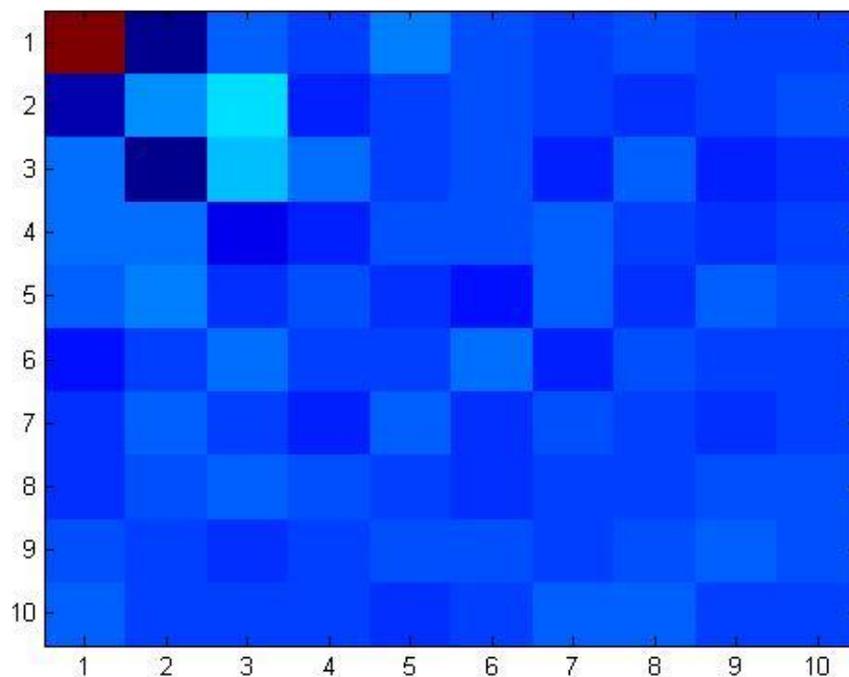


**Figure IV.3:** Schéma du principe de la compression/décompression JPEG implémenté.

### ➤ DCT :

DCT transforme les informations d'image du domaine spatial en la même représentation dans un domaine similaire au fréquentiel. Pourquoi ce changement de domaine est-il si intéressant ? Précisément parce que les images classiques admettent qu'il existe une grande continuité entre les valeurs des pixels alors qu'ils existent des informations utiles non-adjacentes en présentation spatiale, mais peuvent être visibles par la transformation. Étant donné que les hautes fréquences sont réservées aux changements rapides d'intensité des pixels, ces dernières seront au coin de bas, et les basses seront au coin de haut. On arrive donc à représenter toute l'information de l'image sur très peu de coefficients, correspondant à des fréquences plutôt basses. Composants continus importants pour l'œil (moyenne des images traitées).

La DCT en MATLAB se fait par l'instruction " dct2" relative à DCT en deux dimensions (ou encore bidimensionnelle). Pour des raisons de visibilité, la figure 4-4 présente une zone de dix premiers pixels (coin Haut-Droit) où les variations des coefficients DCT sont importantes.



**Figure IV.4:** DCT de l'image de test (composante rouge), zone de dix premier pixels.

### ➤ Zig-Zag :

Le codage de la matrice DCT quantifiée va se faire en parcourant les éléments dans l'ordre imposé par une séquence particulière appelée Zig-Zag (voir figure IV.5). Dans l'exemple de cette figure, la matrice donne la suite suivante : 150, 80, 92, 26, 75, 20, 4, 18, 19, 3, 1, 2, 13, 3, 1, 0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, etc. La séquence a la propriété de parcourir les éléments en commençant par les basses fréquences et en traitant des fréquences de plus en plus hautes. Étant donné que la matrice DCT quantifiée contient de nombreuses composantes hautes fréquences de zéros, l'ordre de la séquence en dents de scie produit une longue séquence de 0 en continu.

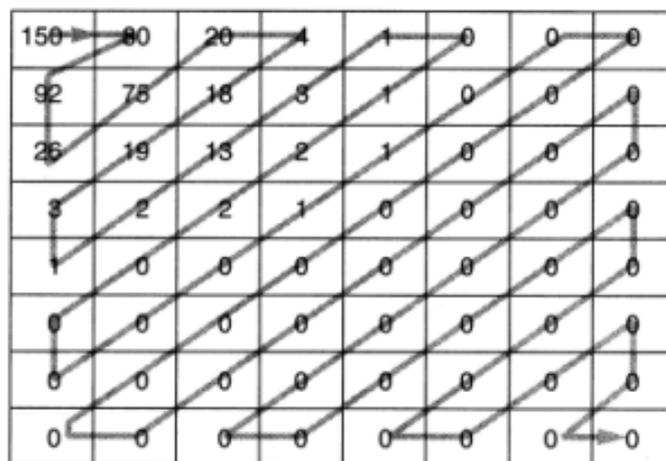


Figure IV.5 : Parcours Zig-Zag

Deux mécanismes sont mis en œuvre pour compresser la matrice DCT quantifiée. D'une part, les séquences de valeurs nulles ne sont simplement codées en donnant le nombre de zéros consécutifs ; c'est le principe de RLE. En revanche, les valeurs non nulles seront encodées à l'aide de méthodes statistiques de type entropique tel que Huffman, Arithmétique ou facilement une simple conversion binaire de longueur fixe.

### ➤ Coupure :

Le point où les coefficients DCT sont négligeables et sont à rendre zéros est appelé point de coupure. C'est l'un des paramètres ou plus précisément variable utilisée pour évaluer les pertes. Comme  $564 \times 564 = 318096$  point de Zig-Zag, la gamme de la variation de point de coupure dans notre travail est [1 à 318097]. Le un supplémentaire veut dire aucune coupure n'est appliquée c-à-d tous les coefficients sont maintenus et aucune négligence n'est prise en

compte, par conséquent le MSE et SSIM obtenus dans le processus sans quantification sont respectivement 0 et 1, autrement dit aucune perte de compression n'est produite.

### ➤ **RLE :**

Tous les coefficients rendus nuls sont présentés en un seul nombre entier. C'est le codage RLE. Bien que le vrai RLE est de coder aussi les coefficients non-nuls convertis en binaire, cette version simplifiée minimise considérablement le temps de calcul.

### ➤ **Quantification :**

L'étape de quantification consiste à réduire la précision de stockage des entiers dans la matrice DCT et par conséquent, pour réduire le nombre de bits qui seront utilisés pour chaque entier connus sous le nom de « Symbole » d'alphabet à coder. C'est la seule partie non conservatrice de la méthode, c'est pour cette raison et à cause de cette quantification que le JPEG est avec perte. Les coefficients DCT rendus nuls sont aussi à cause de la quantification. Étant donné que les informations à basse fréquence sont plus pertinentes que les informations à haute fréquence, la baisse de précision doit être issue de la perte des hautes fréquences. Donc, la perte de la précision augmentera à mesure que vous vous éloignez de la position initiale (0,0).

### ➤ **Conversion binaire ou codage entropique:**

Les coefficients DCT quantifiés non-nuls (jugés considérables) sont en suite convertis en binaire en utilisant un codage de longueur fixe ou un codage entropique de longueur variable tels que Huffman ou Shannon-Fano. Le dernier cas (Codage longueur variable) est plus optimal en termes de la quantité d'information ou tout simplement en termes de nombre de bits utilisés dans le codage. Mais l'inconvénient est qu'il consomme du temps lors de calcul de codage/décodage. La séquence résultante est ajoutée au RLE précédent pour former une seule entité Codée & Compressée.

## IV-4 Résultats et Discussion :

Les résultats sont obtenus de deux implémentations différentes. Dans la première, aucune quantification n'est appliquée. Pourtant ce processus n'existe pas en réalité ; tous les formats JPEG comportent l'étape de quantification, mais nous l'avons adopté dans l'élaboration de ce travail pour évaluer l'ampleur de la perte due au point de coupure. Dans la

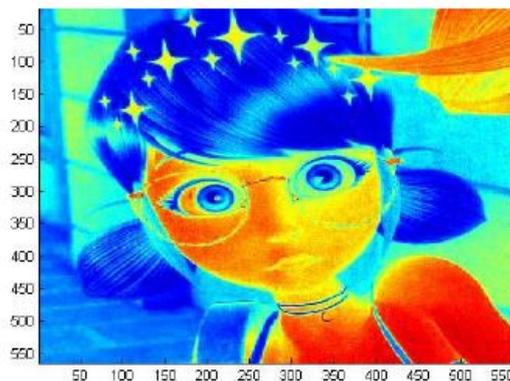
seconde implémentation, le point de coupure est fixé sur  $3 \times 10^4$  et le pas de quantification s'est fait varier dans l'intervalle  $[0 \text{ à } 10^4]$ .

### ➤ *Sans quantification*

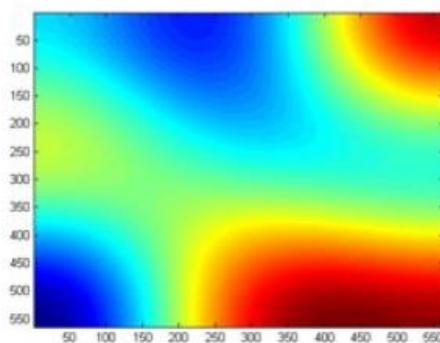
Dans cette section nous avons ignoré la quantification dans le but d'analyser l'effet de la coupure. Dans le cas où la quantification n'est pas utilisée, le codage binaire ou entropique ne sera pas appliqué. Lors la compression, l'image d'entrée est transformée par DCT, les coefficients obtenus sont mis en Zig-Zag et finalement le point de coupure est choisi pour rendre les coefficients après ce point nuls. Nous obtenons alors une séquence composée de deux parties : partie contenant des coefficients DCT réels et une partie de longue suite des zéros.

Lors la décompression, la séquence est remise en matrice (Zig-Zag inverse), puis la DCT inverse est appliquée. La perte ici est seulement due à hautes fréquences supprimées. Pour quantifier la perte, nous mesurons le SSIM et le MSE entre l'image décompressée et l'image d'entrée.

Les figures IV.6, IV.7 et IV.8 ci-dessous montrent les résultats obtenus pour quatre point de coupure : 10, 1000, 10000 et 318097, et respectivement pour les trois composantes : Rouge, Verte et Bleu. La qualité est visuellement différente, plus le point de coupure est loin, plus la qualité est meilleur et la perte est petit (petite valeur de MSE et grande valeur de SSIM).



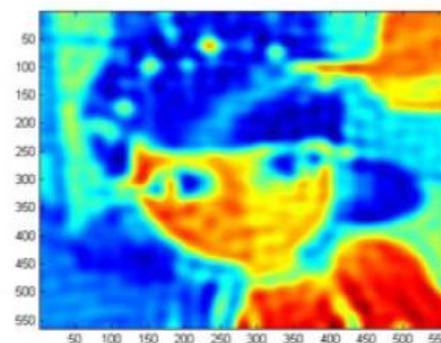
Composante rouge



Point de coupure =10

Valeur de SSIM=  $5.951 \cdot 10^{-5}$

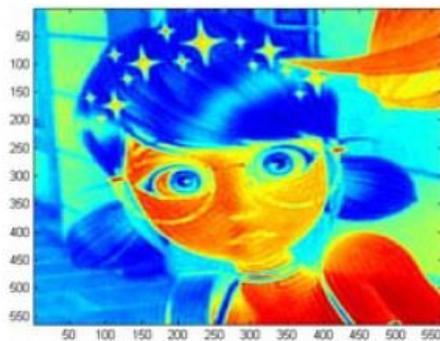
Valeur de MSE=  $1.61 \cdot 10^3$



Point de coupure =1000

Valeur de SSIM= 0.32

Valeur de MSE= 0.135



Point de coupure =10000

Valeur de SSIM= 0.030

Valeur de MSE= 55.545

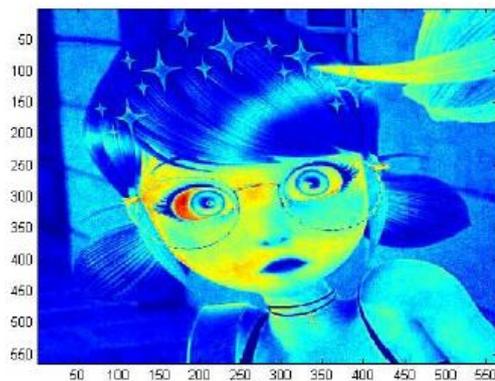


Point de coupure =318097

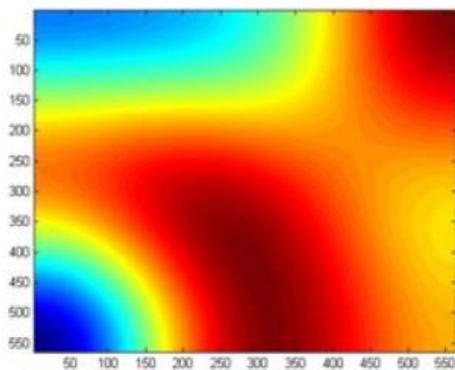
Valeur de SSIM= 1

Valeur de MSE= 0

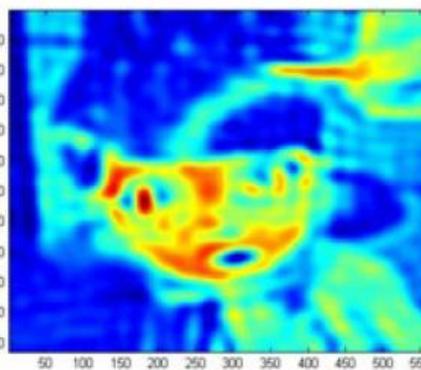
Figure IV.6 : Images décompressées obtenues pour les quatre points choisis.  
Composante Rouge



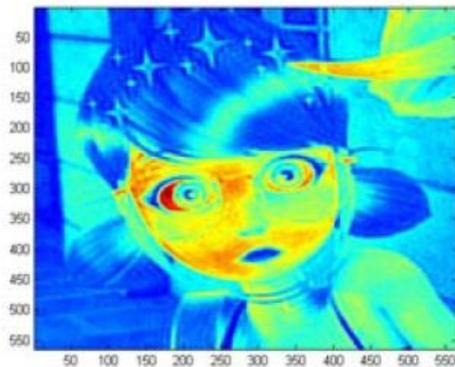
Composante verte



Point de coupure =10  
Valeur de SSIM=  $2.713 \cdot 10^{-5}$   
Valeur de MSE= 676.413



Point de coupure =1000  
Valeur de SSIM= 0.32  
Valeur de MSE= 0.245

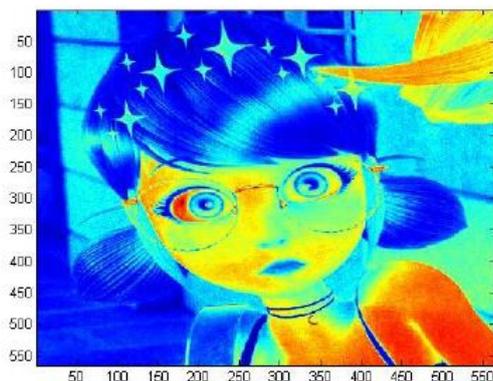


Point de coupure =10000  
Valeur de SSIM= 0.030  
Valeur de MSE= 52.356

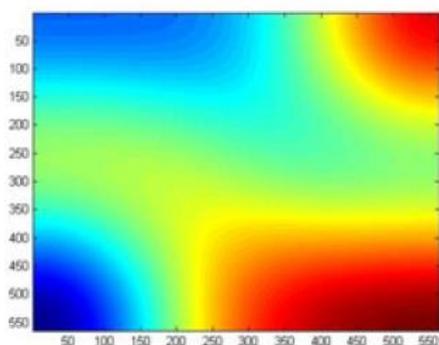


Point de coupure =318097  
Valeur de SSIM= 1  
Valeur de MSE= 0

Figure IV.7 : Images décompressées obtenues pour les quatre points choisis.  
Composante Verte.



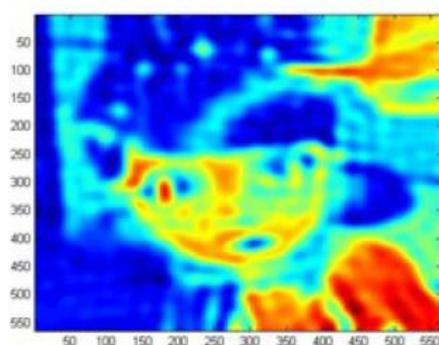
Composante bleu



Point de coupure =10

Valeur de SSIM=  $3.605 \cdot 10^{-5}$

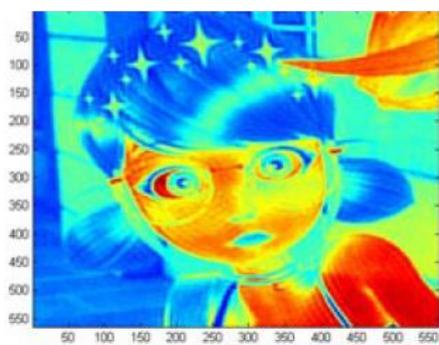
Valeur de MSE= 941.411



Point de coupure =1000

Valeur de SSIM= 0.04

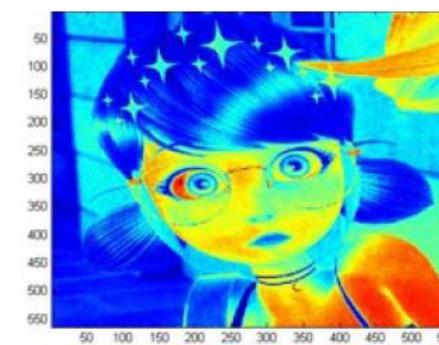
Valeur de MSE= 0.3



Point de coupure =10000

Valeur de SSIM= 0.030

Valeur de MSE= 54.034



Point de coupure =318097

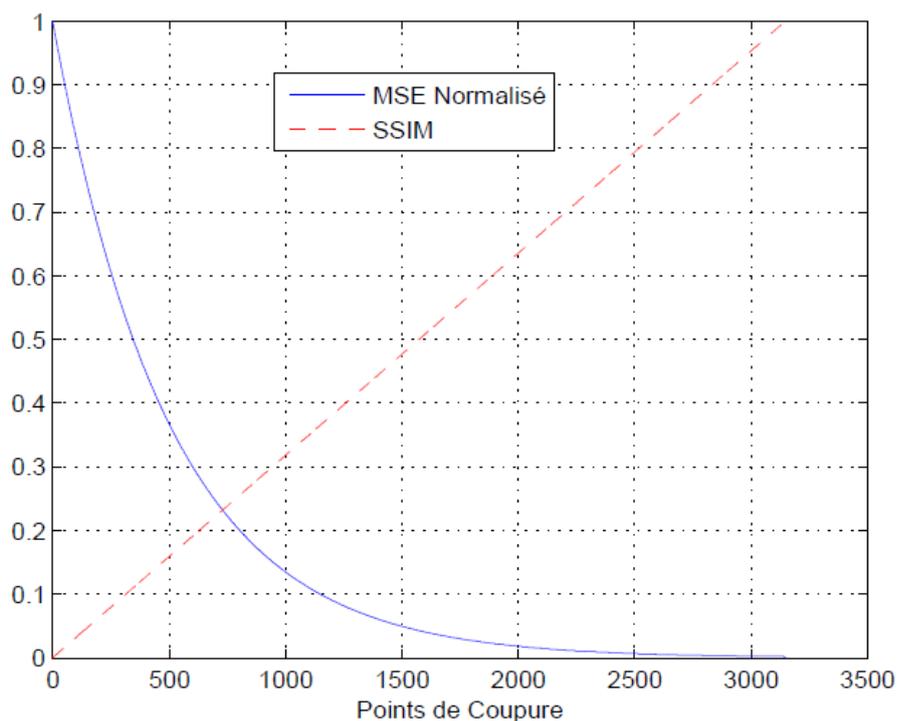
Valeur de SSIM= 1

Valeur de MSE= 0

Figure IV.8 : Images décompressées obtenues pour les quatre points choisis.

Composante Bleu.

Le MSE et SSIM calculés sont pour l'intervalle [1 à 318097]. Cependant, les courbes des figures IV.9, IV.10 et 4-11 montrent le MSE normalisé et SSIM obtenus pour l'intervalle [1 à environ 3000] car les résultats sont stables après ce point. Nous avons normalisé le MSE pour des raisons de mettre les deux courbes sur même figure, de ce fait, la mauvaise qualité (Perte maximale) correspond à  $MSE=1$  et  $SSIM=0$ . Et la bonne qualité (perte minimale/aucune perte) correspond à  $MSE=0$  et  $SSIM=1$ .



**Figure IV.9 :** MSE et SSIM en fonction de point de coupure/Composante Rouge.

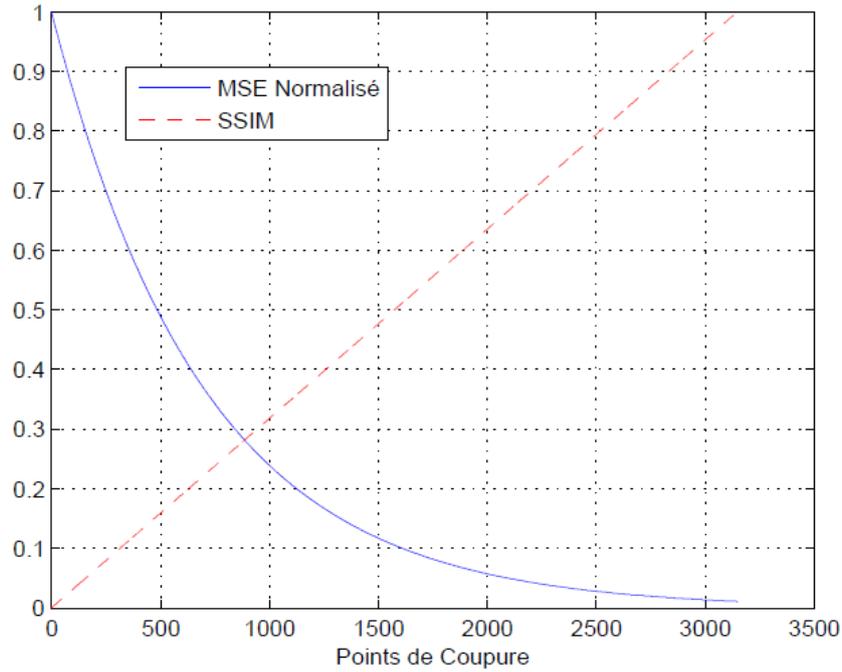


Figure IV.10 : MSE et SSIM en fonction de point de coupure/Composante Verte.

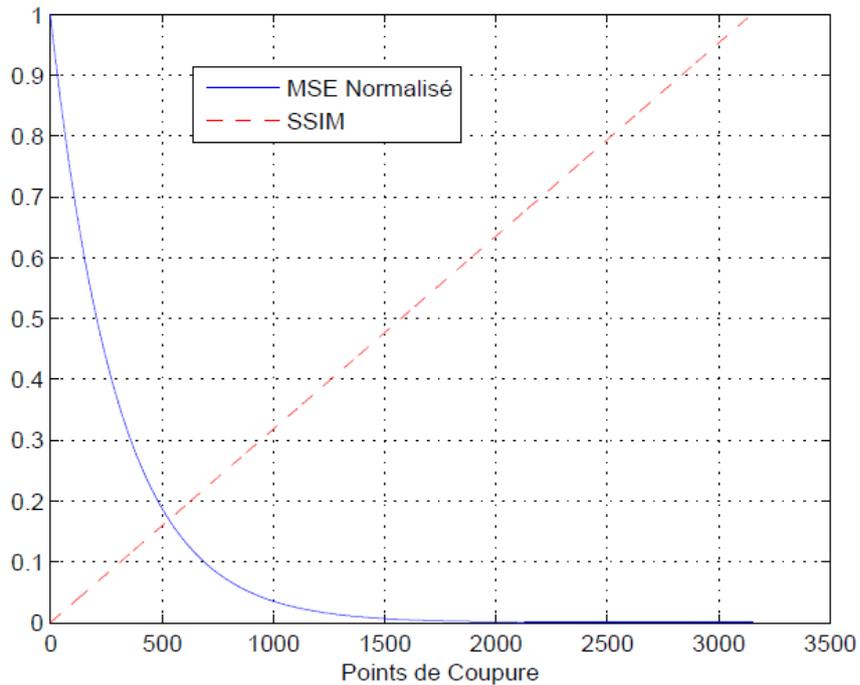
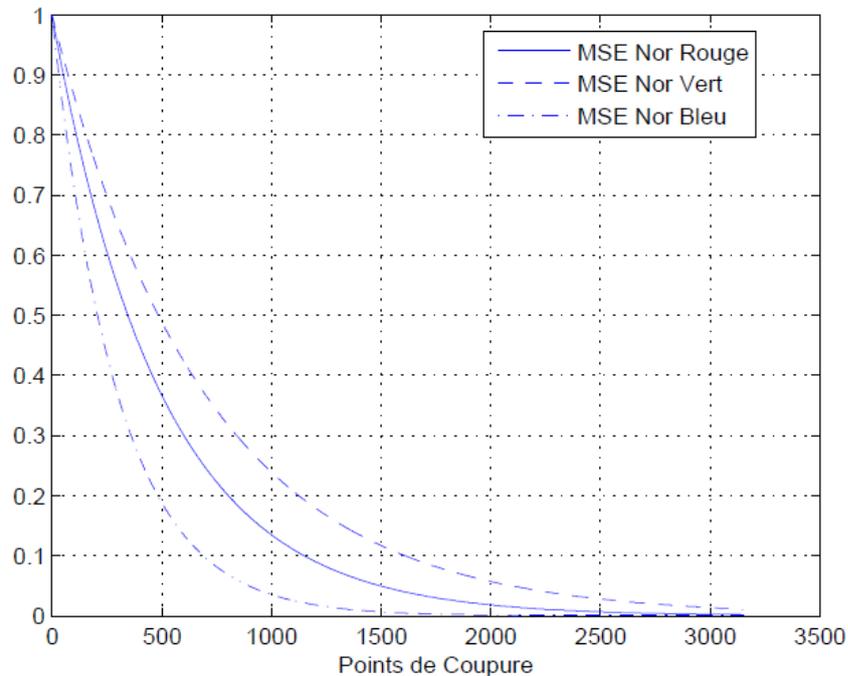


Figure IV.11 : MSE et SSIM en fonction de point de coupure/Composante Bleu.

Comme comparaison entre les trois composantes, la perte est presque la même pour les mêmes points. Les courbes de la figure IV .12 montre ce propos. Néanmoins, l'effet de point de coupure sur une composante est considérable. La perte est à la moitié avant le point 1500 si l'on considère le SSIM seulement. Après ce point la perte commence à être petite jusqu'à le

point optimal entre 3000 et 3200. Après le point optimal, la compression peut se réaliser (318096-3000=315096 hautes fréquences supprimées) sans alter la qualité (MSE normalisé proche à 0 et SSIM proche à 1).



**Figure IV.12** : MSE en fonction de point de coupure pour les trois composantes.

### ➤ Avec quantification:

Dans cette section, nous allons implémenter le JPEG typique c.-à-d. avec quantification. Le point de coupure adopté pour notre image de test est celui mentionné ci-dessus ; soit 3000. Nous ne devons pas s'éloigner trop de ce point pour que la compression soit moins coûteuse en termes de quantité d'information et de nombre de bits attribués. L'intervalle des pas de quantification est  $[0 \text{ à } 10^4]$ . Pourtant la valeur maximale est mathématiquement faisable et veut dire un nombre infini des états à coder (pas de quantification), mais la valeur minimale n'est pas réalisable pratiquement. Nous l'avons remplacé par un nombre aussi petit que possible, soit dans notre cas 0.001 pour assimiler le zéro.

La figure 4-13 ci-dessous montrent les résultats obtenus pour cinq pas de quantification : 1, 10, 100, 1000 et 10000, et pour la composante Rouge. Nous n'avons pas illustré les deux autres composantes puisqu'elles ont presque la même qualité visuelle pour chaque pas choisi.

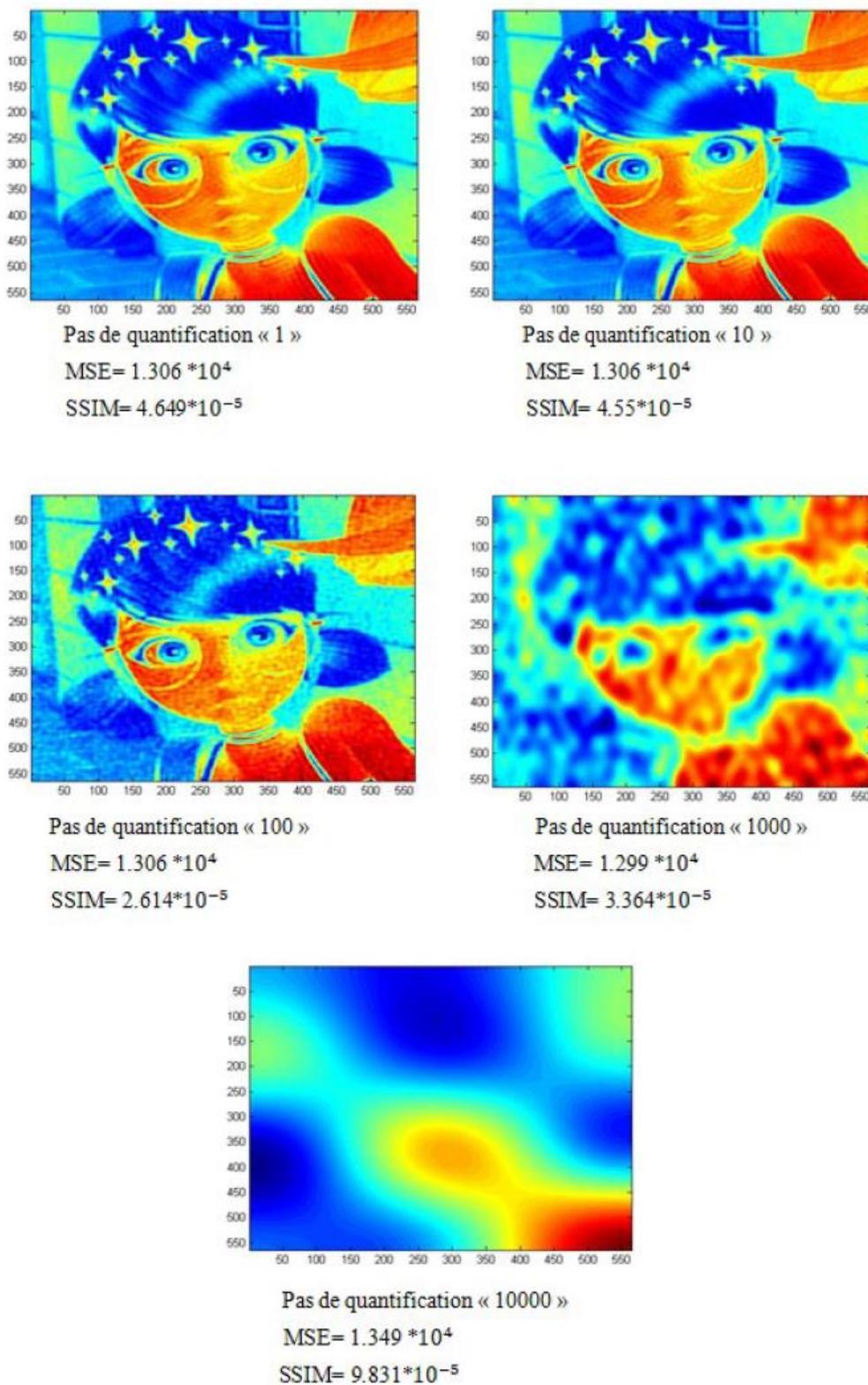
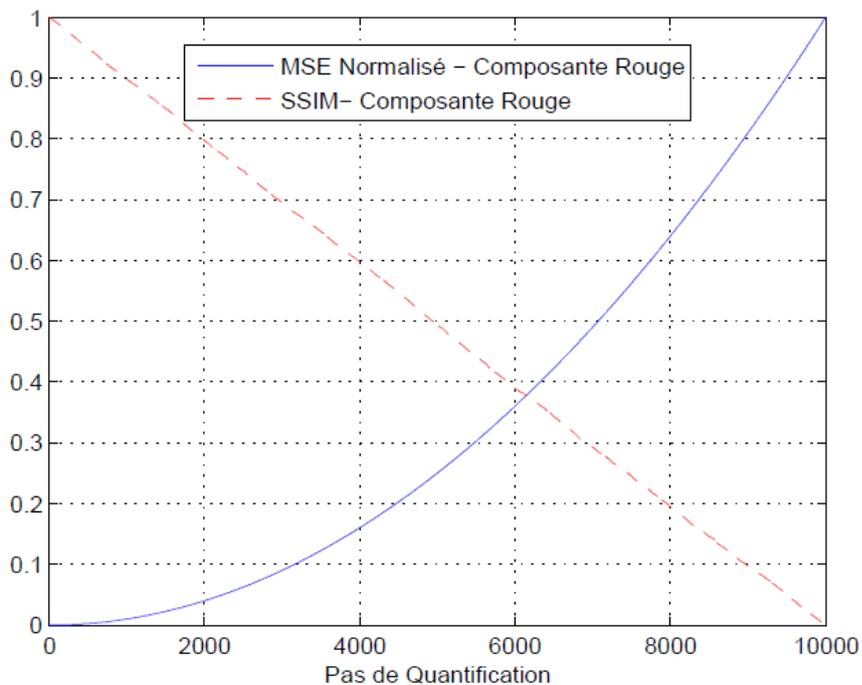


Figure IV .13 : Images décompressées obtenues pour les cinq pas choisis. Composante Rouge

A l'instar d'analyse en fonction de point de coupure, la qualité est visuellement différente d'un pas à un autre. Mais inversement proportionnel, plus le pas de quantification est grand, la qualité est mauvaise (grande valeur de MSE et petite valeur de SSIM). Les courbes de la figure 4-14 montrent les variations de MSE et SSIM en fonction de tous les pas d'intervalle étudié. Cela confirme la relation proportionnelle inverse de la Qualité/Pas de quantification.

Avant finaliser notre manuscrit nous voulons juste ajouter une remarque concernant l'allure des toutes les courbes MSE et SSIM. Vous vous certainement posez la même question : pourquoi l'allure de MSE est courbée alors que le SSIM est presque droite ? La réponse est que MSE suit une variation quadratique et le SSIM est un rapport de même ordre (voir les formules du chapitre précédent).



**Figure IV.14** : MSE et SSIM en fonction de pas de quantification/Composante Rouge.

### **IV.5 Conclusion :**

Dans ce chapitre, nous avons présenté notre implémentation du JPEG sous MATLAB. L'implémentation a été faite sous deux volets, la première est sans quantification et elle vise à analyser les pertes seulement en fonction de point de coupure. La seconde est avec quantification et elle a pour le but d'analyser l'effet de pas de quantification. Afin d'optimiser la compression, dans la seconde implémentation, le point de coupure a été sélectionné comme le point optimal de la première implémentation. Les résultats de l'implémentation typique (avec quantification) montrent que la qualité de compression est meilleure si le pas de quantification est petit. Cependant, le problème pratique réside au niveau des limitations matérielles où le grand nombre des états à coder contraignent les capacités de stockage et de calcul. Effectivement grand nombre d'états implique grand nombre des bits.

Conclusion Générale

1e

### **Conclusion générale :**

Le traitement d'image avec ses axes de travail tels que le codage, la compression et le cryptage présente un domaine d'étude et de recherche motivant. Dans nos jours le nombre de publications scientifiques sous forme de articles et des livres publiés ou des mémoires de fin d'étude comptent des milliers voir même des millions. Dans notre manuscrit nous avons élaborés une technique de compression d'image largement utilisée. Avant tout, nous avons entamé notre manuscrit par une présentation théorique de quelques entités importantes dans deux volets principaux : le traitement d'image et la théorie de l'information. Ensuite dans le dernier volet, nous avons présenté notre implémentation pratique de la technique de la compression JPEG sous l'environnement MATLAB.

Le manuscrit présenté inclut tout ce qui est important pour le JPEG. Premièrement, nous avons décrit l'entité de l'image en montrant les généralités essentielles connues dans le domaine de traitement d'image. Nous avons présenté : l'image numérique et ses caractéristiques, dimension, textures, contour, profondeur, histogramme..etc. ainsi que les différents types d'image et différents formats.

Nous avons aussi donné les notions de base sur la théorie de l'information. Nous avons présenté la notion de la quantité d'information, entropie de Shannon, entropie conjointe et conditionnelle ...etc. Les concepts de le codage de canal et le codage de source, ainsi leur type, leur effets, sont aussi envisageables.

Nous avons présenté plusieurs méthodes importantes de codage, de transformations et des techniques de compressions d'image avec et sans perte. Parmi les techniques étudiées nous avons choisi celles utilisées ou convenables à utiliser dans la compression JPEG. Nous en citons le codage RLE, LZW, codage arithmétique, FFT, DCT et DWT qui faisaient l'objet principale de ce chapitre.

Finalement nous avons implémenté le JPEG modifié de telle sorte qu'elle soit compréhensible sans changer l'aspect global et le principe de base. Les programmes réalisés visent principalement les blocs essentiels de la chaîne de cette technique. L'image est tout d'abord dégroupée en trois composantes principales (Rouge, Vert et Bleu). Ensuite, le DCT, Zig-Zag, la dé-concaténation, RLE, la quantification avec les fonctions inverses correspondantes sont appliqués. L'analyse a été faite en basant sur la comparaison de l'image avant et après la compression sous les critères d'évaluation de la perte (déviations) : SSIM et MSE.

Référence **B**ibliographique

### Références:

- [1] BOLON, Philippe, CHASSERY, Jean-Marc, COCQUEREZ, Jean-Pierre, *et al.* *Analyse d'images: filtrage et segmentation*. Masson, 1995.
- [2] BERGOUNIOUX, Maitine. *Introduction au traitement mathématique des images-méthodes déterministes*. Springer, 2015.
- [3] AMEISEN, David. Qu 'est-ce qu 'une image numérique. In : *Conference Paper*. 2013. p. 169-172.
- [4] SONDES, Tebini, HASSENE, Seddik, ZOUHAIR, Mbarki, *et al.* RGB image de-noising using new low-pass filter with variable Gaussian core real time optimized by neural networks. In : *2013 International Conference on Electrical Engineering and Software Applications*. IEEE, 2013. p. 1-6.
- [5] ZEROUAL, Djazia. *Implémentation d'un environnement parallèle pour la compression d'images a l'aide des fractales*. 2006. Thèse de doctorat. Batna..
- [6] YAKHLEF, Rabiaa, KOUAHI, Manel, et BRAHIMI, Tahar Encadreur. *Étude de l'effet des transformations réversibles sur la performance de compression sans perte des images couleur*. 2020. Thèse de doctorat. Université de Jijel.
- [7] YAKHLEF, Rabiaa, KOUAHI, Manel, et BRAHIMI, Tahar Encadreur. *Étude de l'effet des transformations réversibles sur la performance de compression sans perte des images couleur*. 2020. Thèse de doctorat. Université de Jijel.
- [8] G. Almouzni, "Traitement numérique des images ", Cours Ecole Internationale des Sciences du Traitement de l'Information (EISTI), France, 2008.
- [9] BOUYOUCEF, Karim. *Sur les aspects multirésolution en reconstruction d'image*. 1993. Thèse de doctorat. Toulouse.
- [10] J. Serra, "Notions de Base de traitement d'image", Cours Morphologie, Ecole des Mines de Paris, 2000.
- [11] M. Bergounioux, "Méthodes mathématiques pour le Traitement d'image", Cours Traitement et reconnaissance d'image, Master 2, Université Pierre et Marie CURIE, Paris 10 mai 2009.
- [12] MOHAMED, A. Z. N. I. Communications Numériques 1 (UEF 21).
- [13] SLAOUI, Abdallah. Les relations intrinsèques des théories des ressources quantiques et leurs rôles dans la métrologie quantique.
- [14] MEKOUAR, Monsef. *Compression d'images médicales par ondelettes et régions d'intérêt*. 2001. Thèse de doctorat. École de technologie supérieure.

## Bibliographié

- [15] SHANNON, Claude E. Communication theory of secrecy systems. *The Bell system technical journal*, 1949, vol. 28, no 4, p. 656-715.
- [16] RIOUL, Olivier. *Théorie de l'information et du codage*. Paris : Lavoisier, 2007.
- [17] BERGOT, François-Xavier. *Dualité entre codage de source et codage de canal*. 2000. Thèse de doctorat. Télécom ParisTech.
- [18] BENTAHAR Tarek, " cour codage et compression " université Laarbi tbessi -Tebessa-
- [19] BERETTA, Patricia. *Compression d'images appliquée aux angiographies cardiaques: aspects algorithmiques, évaluation de la qualité diagnostique*. 1997. Thèse de doctorat. Lyon, INSA.
- [20] DOUAK, Fouzi. *Reconstruction des images compressées en utilisant les réseaux de neurones artificiels et la DCT*. 2008. Thèse de doctorat. Université de Batna 2.
- [21] CZIHO, Andras. *Quantification vectorielle et compression d'image. Application à l'imagerie médicale*. 1999. Thèse de doctorat. Rennes 1.
- [22] ACHARYA, Tinku et TSAI, Ping-Sing. *JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures*. John Wiley & Sons, 2005.
- [23] MEKOUAR, Monsef. *Compression d'images médicales par ondelettes et régions d'intérêt*. 2001. Thèse de doctorat. École de technologie supérieure.
- [24] CZIHO, Andras. *Quantification vectorielle et compression d'image. Application à l'imagerie médicale*. 1999. Thèse de doctorat. Rennes 1.
- [25] NELSON, Mark. *La compression de données*. 1993.
- [26] BOUHLEL, Atef MASMOUDI Mohamed Salim. *Un nouvel algorithme de compression exploitant le codage arithmétique en lui introduisant de nouveaux paramètres de codage. Institut Supérieur de Biotechnologie de Sfax (ISBS)-TUNIS*, 2007.
- [27] MITCHELL, H. B., ZILVERBERG, N., et AVRAHAM, M. A comparison of different block truncation coding algorithms for image compression. *Signal processing: Image communication*, 1994, vol. 6, no 1, p. 77-82.
- [28] DHARA, Bibhas Chandra et CHANDA, Bhabatosh. Color image compression based on block truncation coding using pattern fitting principle. *Pattern Recognition*, 2007, vol. 40, no 9, p. 2408-2417.
- [29] LANDRY, Christiane. *Correction interactive de couleur par association*. 1999. Thèse de maîtrise. Université de Montréal.
- [30] S. Saha, "Image Compression – from DCT to Wavelets: A review", ACM Cross words students magazine, Vol.6, No.3, spring 2000.