

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



Université de Ghardaïa

N° d'ordre :
N° de série :

Faculté des Sciences et Technologies
Département d'automatique et électromécanique

Mémoire présenté en vue de l'obtention du diplôme de

MASTER

Domaine : *Sciences et Technologies*

Filière : *Automatique*

Spécialité : *Automatique et Systèmes*

Par : **BOUAMER Bouhafs**

DEHANE Youcef

Thème

**Compression D'Image Par La Méthode TCD
(Transformée En Cosinus Discrète)**

Soutenu publiquement le 25/06/2019

Devant le jury :

Mr. KIFOUCHE Abdessalam	MAA	Université	Président
Mr BENAOUICHA Karim	MAA	Université	Examineur
Mr. ARIF Mohammed	MAA	Université	Examineur
Mr.LAADJAL Boumadiane	MAA	Université	Encadreur

Année universitaire 2018/2019

الإهداء

إلى من حملتني في أحشائها وهزت مهدي بيدها إلى التي لا يطمئن قلبي إلا
وأنا بين أحضانها

إلى من حبي وتقديري لها يفوق كل الحدود .

إليك يا معنى الوجود اهدي كل الورود

إليك يا أماه.....إليك يا أماه

إلى من فتح أبواب الحياة في وجهي وأثار شموع الأمل في دربي إلى من
علمني أحيا في ظل دين

ربي إليك يا أبي .

إلى من قاسموني حب وحنان والداي، سندي في الحياة عائلتي الكريمة

و إلى أصدقائي

إلى كل من لم يخطهم قلبي ولكن طبعا يحملهم قلبي اهدي باكورة هذا العمل

شكر وتقدير

بداية نتقدم بشكرنا إلى الله سبحانه وتعالى الذي منحنا القوة

والصبر في إنجاز هذا العمل المتواضع؛

كما نتقدم بالشكر الجزيل بكل صدق وإخلاص إلى الأستاذ

المشرف:

لعجال بومدين على قبوله الإشراف على هذا العمل

وعلى توجيهاته وإرشاداته القيمة

كما نشكر كل من أمدنا بيد العون والمساعدة سواء من قريب أو

من بعيد

وفي الأخير نشكر كل الأساتذة أعضاء اللجنة على تشريفهم لنا

بقبولهم مناقشة هذه المذكرة

وإلى جميع الأساتذة، بجامعة غرداية

وإلى كل من ساهم من قريب أو من بعيد في إنجاز هذه

المذكرة

Liste de Tableau

N° tableau	Table	Page
Tableau I-1	Table Histogramme	9
Tableau III-1	Exemples la DCT	33
Tableau III-2	Exemples du seuillage TH=4	34
Tableau III-3	Exemples la matrice DCTQ	35
Tableau III.4	exemple huffman code	36
Tableau III.5	mesuré le temps Les étapes de l'algorithme pour changer de seuillage l'Image IRM1	43
Tableau III.6	mesuré le temps Les étapes de l'algorithme pour changer de seuillage l'Image IRM2	46
Tableau III.7	mesuré le temps Les étapes de l'algorithme pour changer de seuillage l'Image IRM3	50
Tableau III.8	Valeurs de CR, PSNR, MSE, et le G(%) pour l'image IRM1	50
Tableau III.9	Valeurs de CR, PSNR, MSE, et le G(%) pour l'image IRM2	51
Tableau III.10	Valeurs de CR, PSNR, MSE, et le G(%) pour l'image IRM3	51

Liste de figure

N° figure	Figure	page
Figure I-1	Représentation d'image numérique.	5
Figure I-2	groupe de pixel formant la lettre A	5
Figure I-3	Représentation de dimension d'une image	6
Figure I-4	Image sans bruit.	7
Figure I-5	Image avec bruit	7
Figure I-6	Contour d'une image	8
Figure I-7	Tracé de l'histogramme	9
Figure I-8	Image binaire (noire et blanc)	10
Figure I-9	Image à niveaux de gris (intensité ou luminance)	10
Figure I-10	Image couleur RGB	11
Figure II.1	Schéma fonctionnel de la compression	16
Figure.II.2	Exemple de l'arbre de Huffman	21
Figure II-3	Schéma de la compression symétrique	24
Figure III.1	Interface graphique et IDE de MATLAB R2011a command Windows et fenêtre editor	30
Figure III.2	Les étapes de compression/décompression	31
Figure III.3	Balayage Zig-zig du standard	35
Figure III.4	exemple huffman	37
Figure III.5	image originale	40
Figure III.6	Image Reconstituée	40
Figure III.7	Image reconstituée et image erreur, pour le bloc [32x32] et le codage Huffman	41
Figure III.8	image originale	44
Figure III.9	Image Reconstituée	44
Figure.III.10	image originale reconstituée et erreur	48

Liste des abréviations

DCT	Discrete Cosine Transforma
IDCT	Discrete Cosine Transforma Inverse
PSNR	Peak Signal to Noise Ratio
Dpi	Dot Per Inch
RGB	Les couleurs : rouge, vert et bleu
H(v)	présente le nombre de fois répété de valeur du pixel dans l'image
Tb	Taux de bits en bits/pixel (bpp)
Q	La matrice de quantification
F*(u, v)	La matrice quantifiée
F(u, v)	La matrice des coefficients à quantifier
Q(u, v)	La table de quantification
RLE	Run lenght Encoding
CR	Le taux de compression
DFT	La transformation de Fourier discrète
TOC	Transformation en ondelettes continue
TOD	Transformation en ondelettes discrètes
MSE	l'erreur quadratique moyenne
PPSNR	Peak to Peak Signal to Noise Ratio
PSNR	Peak Signal to Noise Ratio
ppp	points par pouce
JPEG	Joint Photographique Experts Group

Table de matières

Dédicaces	
Remerciement	
Liste des tableaux	
Liste de figure et photo	
Liste des abréviations	
Table de matières	
Introduction général	
CHAPITRE I : Généralité sur le Traitement d'image	
I.1 Introduction	4
I.2 Définition d'une image	4
I.3 Image numérique	4
I.4 Caractéristiques d'une image numérique	5
I.5 Types d'images	11
I.6 Conclusion	13
CHAPITRE II : Les différentes méthodes de compression	
II.1.Introduction	15
II.2 Définition de la compression.....	15
II.3 Compression d'images	16
II.4 Méthodes de compression.....	18
II.5 La transformation.....	22
II.5.1 La transformation de Fourier discrète DFT.....	22
II.5.2 Transformée en ondelettes.....	22
II.6 Compression symétrique et asymétrique.....	24
II.7 La décompression	24
II.8 Transformée en DCT	24
II.9 Compression et décompression par la DCT	25
II.10.Les avantages et les inconvénients de DCT	27
II.11 Conclusion	27
CHAPITRE III : Application et Implémentation	
III.1 Introduction.....	29
III.2 Définition de Matlab	29

Table de matières

III.3 L'interface de programme d'application MATLAB	29
III.4 L'implémentation de l'algorithme	30
III.5 Critères de performances	38
III.6 les images de type IRM	39
III.7 Les Tableaux : Valeurs de CR, PSNR, MSE, et le G(%) en fonction des différents paramètres avec le codage huffman pour l'image IRM1 IRM2 et IRM3	50
III.8 Conclusion	52
Conclusion général	
Bibliographie	
Résumé	

Introduction

Général

Introduction général

Introduction général

Dans le monde numérique et du partage d'information, la vitesse et la capacité de transmission est primordiale. La compression de données est donc de plus en plus abordée.

Elle vise à réduire le nombre de bits comportant l'information à transmettre ou à stocker. L'objectif du développement des techniques de compression/décompression est d'économiser les ressources de stockage ou l'optimisation de l'usage des canaux de communication. Les algorithmes utilisés par les techniques de compression/décompression de données (image, vidéo, ...) pour réduire la bande passante requise sont basés sur diverses techniques telles que le traitement du signal et l'image et le codage d'information qui produisent des signaux codés qui sont malheureusement sensibles aux erreurs de transmission qui peuvent influencer la qualité de l'information décodée.

Le premier a été utilisé par l'inventeur américain Samuel Morse en 1836 et a été nommé "Morse". Code

Nous avons besoin de la compression de données pour plusieurs raisons, y compris la fourniture d'espace de stockage occupé par les fichiers de grande taille qui nous permet de stocker plus de données sur l'espace de stockage lui-même, et pour le monde de l'émission en réseau est essentiel et très important, puisque nous avons aujourd'hui affaire avec de gros fichiers sous forme de fichiers son, l'image et d'autres. Si nous l'envoyons dans son format d'origine sans aucun processus de compression de données, vous aurez besoin de beaucoup de temps dans les processus de transmission et de réception, ainsi que d'un goulot d'étranglement dans le trafic réseau. Si les mêmes données sont compressées et transférées, elles se déplaceront beaucoup plus rapidement. Nous avons donc réduit le temps de transmission au même Les données.

Comprimer les données (images) revient tout simplement à en éliminer toute information superflue. On parle alors de réduction de la redondance. La redondance se rapporte à la qualité ou à l'état d'être en surnombre, par rapport à la logique. Ce qui peut avoir la connotation négative de superflu, mais aussi un sens positif quand cette redondance est voulue afin de prévenir un dysfonctionnement. Nous nous intéresserons dans le cadre de ce mémoire à l'une des méthodes la de compression irréversibles qui est la compression d'image selon la technique de Transformée en cosinus discrète (TCD).

La transformée en cosinus discrète ou TCD (de l'anglais : DCT ou *Discrete Cosine Transform*) est une transformation proche de la transformée de Fourier discrète (DFT). Le

Introduction général

noyau de projection est un cosinus et crée donc des coefficients réels, contrairement à la DFT, dont le noyau est une exponentielle complexe et qui crée donc des coefficients complexes. On peut cependant exprimer la DCT en fonction de la DFT, qui est alors appliquée sur le signal symétrisé.

Dans ce mémoire il y'a trois chapitres, selon l'organisation suivante :

Le premier chapitre est consacré à la présentation des différents outils permettant le traitement des images numérique.

Le deuxième chapitre a pour objet d'expliquer les différentes techniques de compression d'images numérique, nous présentons les techniques de codages sans perte ainsi les méthodes de compression avec pertes.

Le troisième chapitre consiste à présenter la chaine algorithmique de compression d'image par la transformée en cosinus discret (DCT), et comporte les résultats de simulations effectuées et la discussion des résultats obtenus. Ce travail est achevé par une conclusion générale.

Chapitre I

Généralité sur le Traitement d'Image

I.1 Introduction

Le traitement d'image peut être défini comme l'ensemble des méthodes et techniques opérant sur l'image afin d'extraire les informations les plus pertinentes ou tout simplement pour fournir une image plus sensible à l'œil humain.

Dans ce chapitre nous présentons quelques notions de base du domaine de traitement d'image numérique tels que: la définition d'image, les types d'image, caractéristiques d'image, système de traitement d'image, analyse élémentaire, filtrage, La convolution, segmentation et en fin quelques exemples concrets de traitement d'images.

I.2 Définition d'une image :

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

Elle peut être décrite sous la forme d'une fonction $I(x, y)$ analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation [1].

I.3 Image numérique :

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs[1].

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x,y)$, comme le montre la figure où :

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

$F(x, y)$: niveau d'intensité.

La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur.

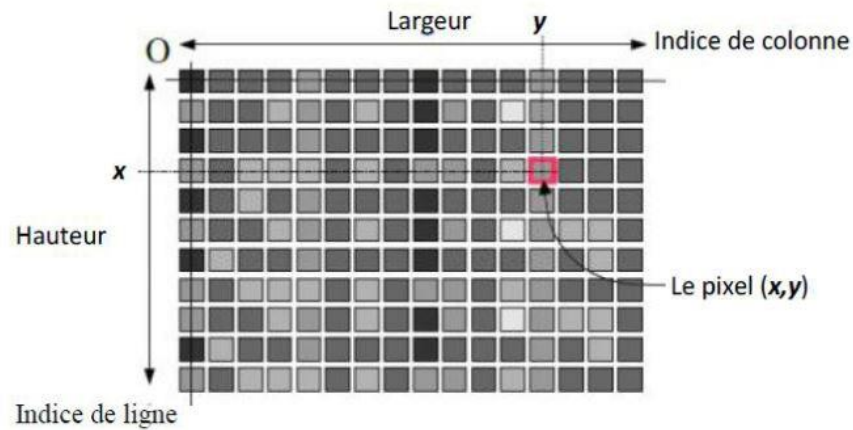


Figure I-1 : Représentation d'image numérique.

I.4 Caractéristiques d'une image numérique :

Comme nous l'avons vu, l'image est un ensemble structuré d'informations parmi ses caractéristiques nous pouvons citer les paramètres suivants:

I.4.1 Pixel :

Contraction de l'expression anglaise " Picture Elements ": éléments d'image, le pixel est le plus petit point de l'image, c'est une valeur numérique représentative des intensités lumineuses.

Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels sur l'image. La lettre A par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous [2].

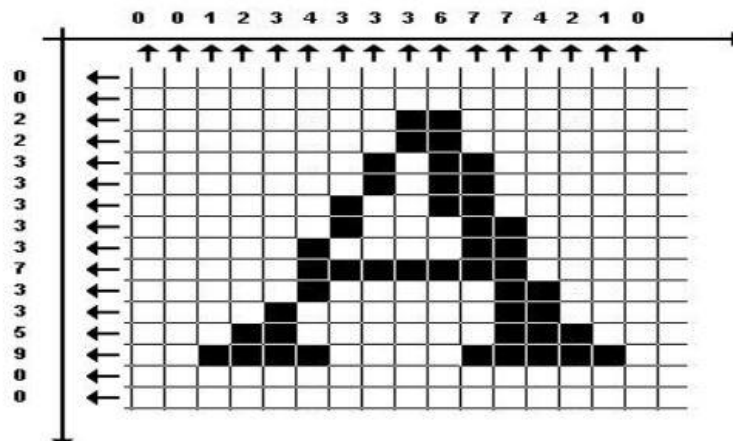


Figure I-2 : groupe de pixel formant la lettre A.

I.4.2 La résolution

La résolution d'une image correspond au niveau de détail qui va être représenté sur cette image. C'est le nombre de pixels par unité de longueur dans l'image à numériser. Elle est en dpi (dots per inch) ou en ppp (points par pouce).

Plus le nombre de pixels est élevé par unité de longueur de l'image à numériser, plus la quantité d'inflation qui décrit l'image est importante et plus la résolution est élevée [3].

I.4.3 Dimension :

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image [3].

➤ Exemple :

Pour une image de 640x480 en couleur :

- ✓ Nombre de pixel = $640 \times 480 = 307200$
- ✓ Poids de chaque pixel = 3 octets
- ✓ Le poids de l'image = $307200 \times 3 = 921600$ octets = 900 Ko

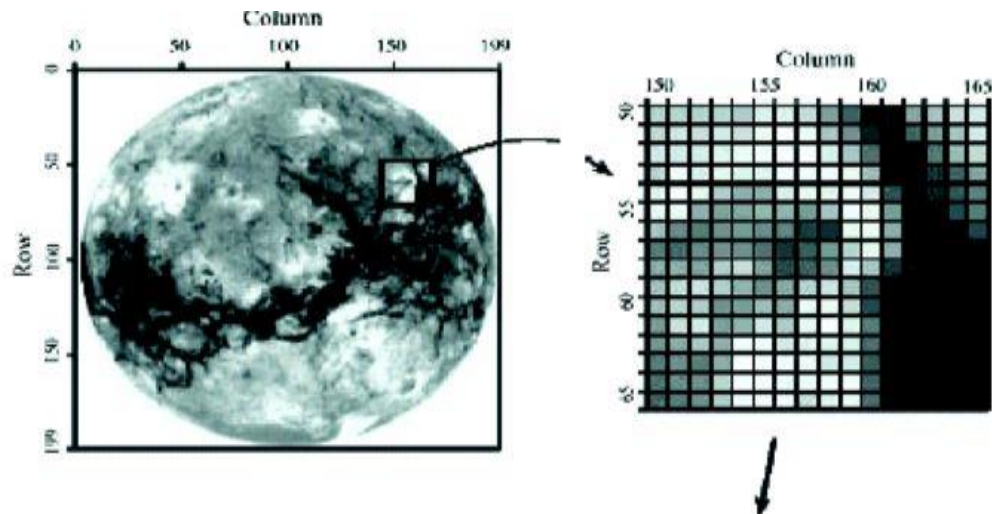


Figure I-3 Représentation de dimension d'une image

I.4.4 La texture :

Une texture est une région dans une image numérique qui a des caractéristiques Homogènes. Ces caractéristiques sont par exemple un motif basique qui se répète .la texture est composée de Texel, l'équivalent des pixels [4].

I.4.5 Bruit :

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [5].



Figure I-4 : Image sans bruit.



Figure I-5 : Image avec bruit.

I.4.6 La luminance :

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

Une bonne luminance se caractérise par :

- ✓ Des images lumineuses (brillantes) ;
- ✓ Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir ; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- ✓ L'absence de parasites [6].

I.4.7 Contours :

Les contours représentent la frontière 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 [4].



Figure I-6 : Contour d'une image.

I.4.8 Histogramme :

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image.

Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée [7].

Exemple :

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


Généralité sur le Traitement d'image

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

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

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

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



Image

Tableau I-1 : Table Histogramme.

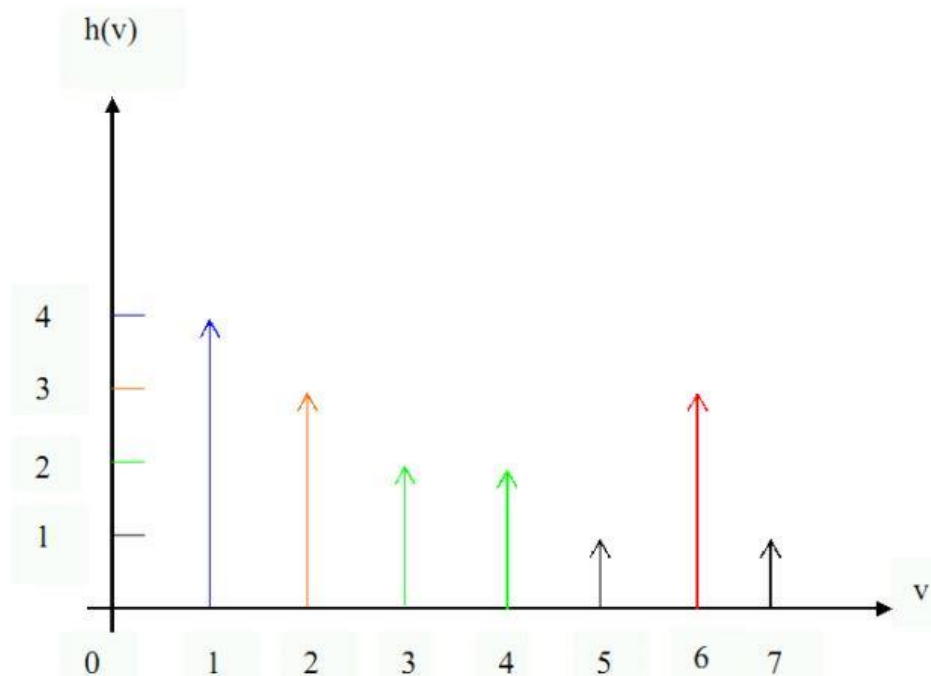


Figure I-7 : l'histogramme de l'image.

I.4.9 Images binaires (en noir et blanc)

Les images binaires sont les plus simples. (dont les pixels ne peuvent avoir que les valeurs 0 et 1) .Le 0 correspond à un pixel noir et 1 à un pixel blanc. Le niveau de gris est donc codé sur un seul bit.

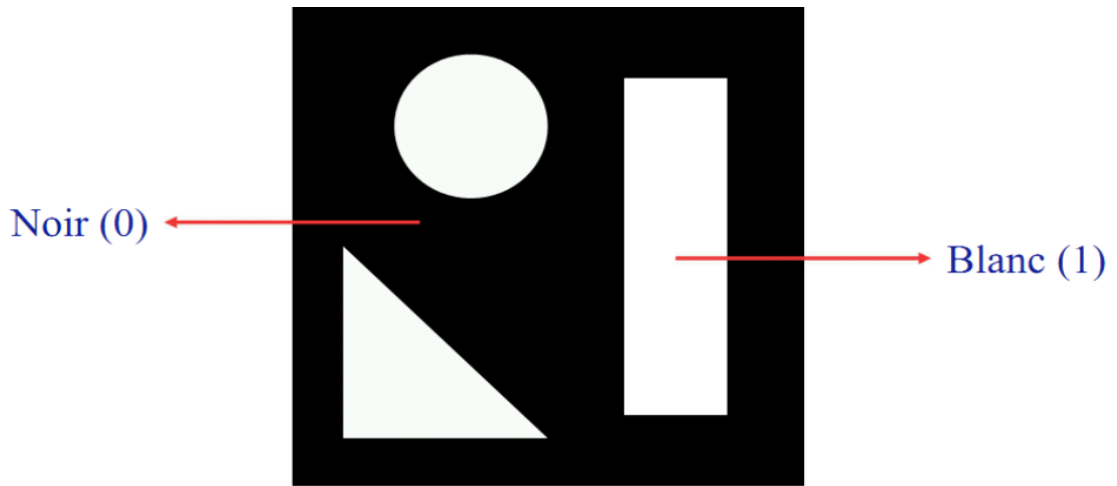


Figure I-8 : Image binaire (noire et blanc)

I.4.10 Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la "couleur" de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux [6].

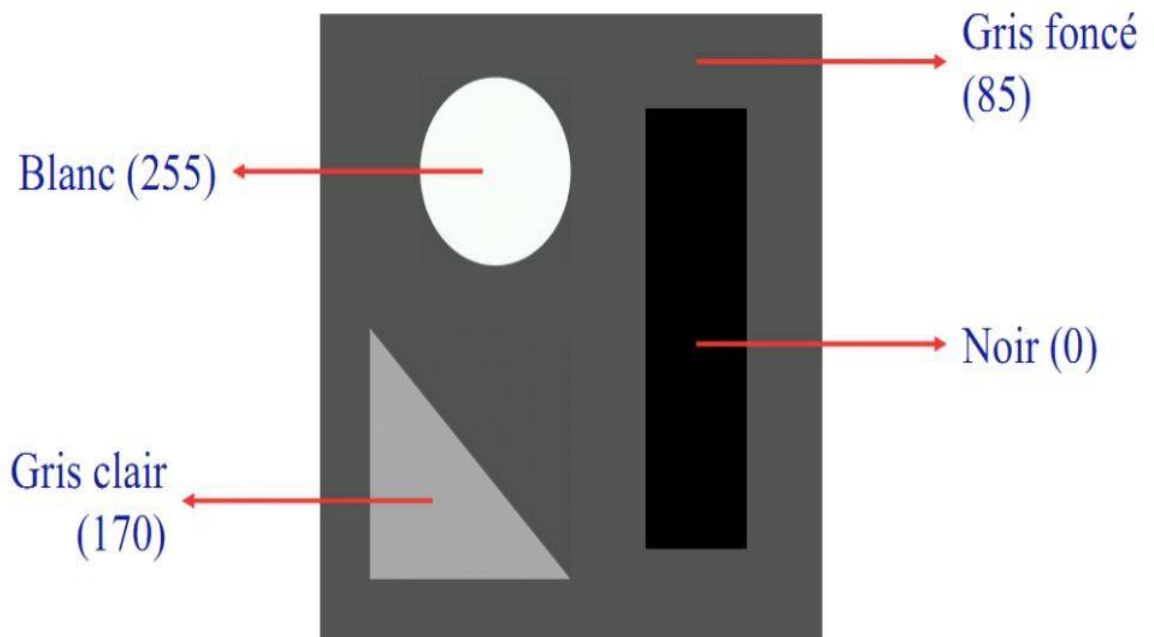


Figure I-9 : Image à niveaux de gris (intensité ou luminance)

I.4.11 Images en couleurs :

La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.) [5][6][8].

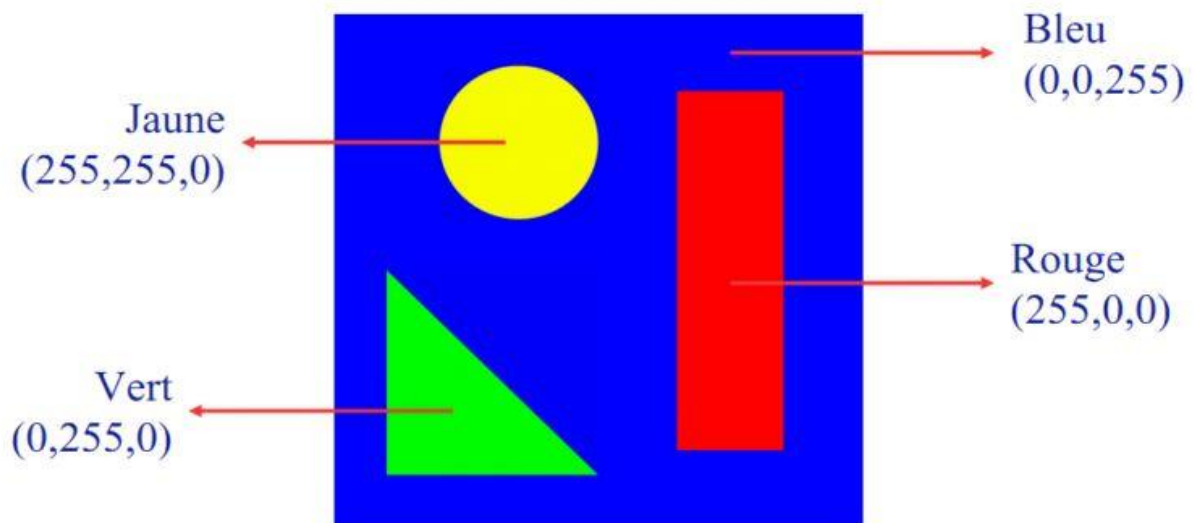


Figure I-10 : Image couleur RGB

I.5 Types d'images :

I.5.1 L'image vectorielle :

L'image vectorielle est une représentation conceptuelle de forme calculée par des formules mathématiques, (exemple, un cercle n'est pas déterminé par des pixels mais par une formule mathématique qui détermine sa forme, sa taille et son emplacement).

L'image Vectorielle utilise également la technique du Pixel, mais cette fois, leur position et leur couleur ne sont pas figées puisqu'elles sont calculées dynamiquement.

I.5.2 L'image Bitmap (matricielle) :

Les images affichées sur l'écran d'un PC sont des images matricielles, encore appelées bitmap (image bit), chaque image est en réalité une matrice de pixels. [9][10]

a. Les formats d'images Matricielle [7] :

- Le format BMP origine Microsoft pour Windows 3.x de plus en plus répandu Contraction de Bitmap. Constitue un format de fichier image, pixel par pixel et sans compression
- Le format GIF (Graphique Inter change Format) : Créé par CompuServe, utilise aussi le codage RVB 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 un bon algorithme de compression LZW et un affichage rapide de l'image.
- Le format JPG ou JPEG (Joint Photographique Experts Group): Créé par un consortium industriel, ce format très utilise sur Internet, permet d'afficher les images en mode 16millions de couleurs et donc c'est le mode sans perte. Le codage d'un pixel (point de l'affichage) 3x1, byte (RVB) soit : $(0 < R < 255)$, $(0 < V < 255)$, $(0 < B < 255)$. Le format JPG peut aussi être utilisé pour compacter les images, il utilise un algorithme de compression qui réduit énormément la taille des images.
- Le format PCX (Picture Exchange Image Bitmap Zsoft) : A été créé à l'origine par Zsoft pour un programme de dessin (paient bruche). Permet de traiter des images 8 bits et non adapté aux images 16, 24, 32 bit, il utilise un algorithme de compression : RLC
- Le format TIFF (Tagged Image File Format) origine Aldus et Microsoft, pour les images scannées, la compression du format TIFF est effectué sans perte de données il utilise plusieurs algorithmes de compression : RLC, LZW
- Le format TGA (True vision Targa) origine True vision, algorithme de compression RLC.
- Le format FAX utilisé pour la transmission de documents (télécopie), codage binaire, application d'un codage RLC puis d'un codage de Huffman

I.6 Conclusion

Dans ce chapitre, nous avons essayé de faire un récapitulatif sur les notions élémentaires d'image numérique (vectorielle et matricielle) comme le pixel, dimension, résolution, bruit, histogramme, contours et textures, luminance, contraste, niveau de gris et la couleur et ses différents types

Chapitre II

Les Différentes Méthodes De Compression

II.1.Introduction :

Avec le développement de l'outil informatique, on effectue des échanges de volumes importants d'information. Or la gestion d'une telle masse pose des problèmes de stockage et de transfert. Pour cela, des études ont été menées afin de mettre en évidence des algorithmes de compression et décompression de données. Leur but est de changer le format des informations de telle sorte qu'elles occupent moins de volume. Une fois compressées, les données ne sont plus accessibles en tant que données cohérentes ; pour les récupérer, il suffit de les décompresser.

Dans ce chapitre, nous donnons quelques notions essentielles sur la compression d'image.

II.2 Définition de la compression :

En général, la compression des données consiste à prendre un flux de symboles et à les transformer en codes [5]. La compression est effectuée par un programme qui utilise une formule ou un algorithme pour déterminer comment réduire la taille des données pour gagner l'espace de stockage. La compression de données est également appelée codage source ou réduction de débit est le processus de modification, d'encodage ou de conversion de la structure des bits de données de manière à consommer moins d'espace sur le disque. Il permet de réduire la taille de stockage d'une ou plusieurs instances ou éléments de données. L'objectif de la compression des données est de représenter une source d'information aussi précisément que possible en utilisant le plus petit espace de stockage.

II.3 Compression d'images :

II.3.1 Principe de compression :

Le schéma fonctionnel de la compression est présenté dans la figure ci-dessous : [11]

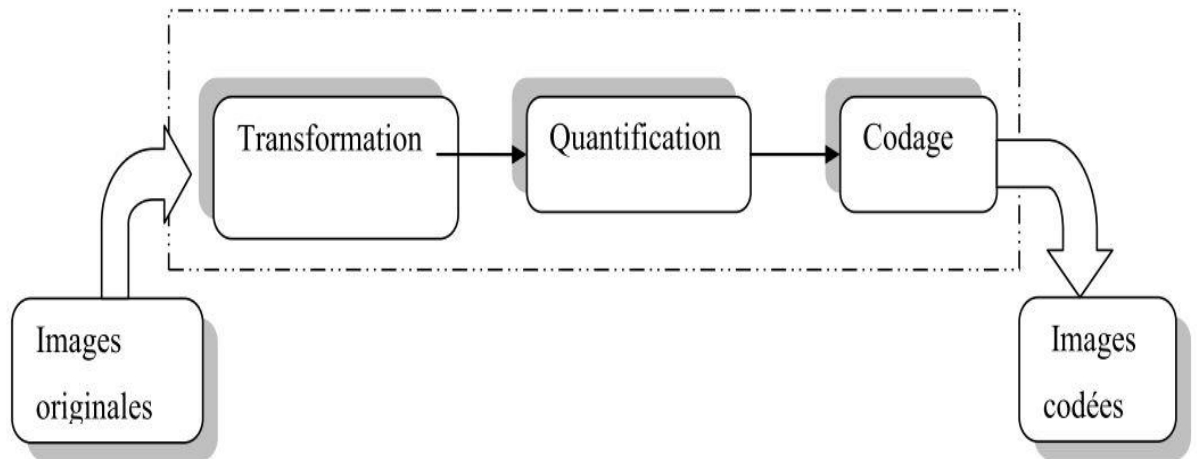


Figure II.1 : Schéma fonctionnel de la compression

A. Transformation :

La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte sur l'image. On essaie donc de tirer parti de cette corrélation, pour réduire le volume d'information en effectuant une opération de transformation des pixels. La transformation consiste à transformer les pixels initiaux en un ensemble de coefficients plus corrélés, c'est une opération réversible.[11]

B. Quantification :

La quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. Elle représente une étape clé de la compression. Elle approxime chaque valeur d'un signal par un multiple entier d'une quantité q , appelée quantum élémentaire ou pas de quantification, elle peut être scalaire ou vectorielle.[11]

C. Codage :

Les coefficients quantifiés forment un train binaire le plus réduit possible. Cette opération ne provoque aucune distorsion ou perte à l'information contenue dans l'image.[11]

II.3.2 Performance de la compression :

A. Taux de compression :

Etant donné que l'objectif de la compression est de minimiser la quantité de l'information indispensable pour la représentation d'une image, on définit le paramètre taux de compression CR (Compression Ratio) :

$$CR = \frac{\text{Nombre de bits avant la compression}}{\text{Nombre de bits après la compression}} \quad (\text{II.1})$$

A partir de ce quotient on définit aussi le taux de compression en pourcent par le rapport :

$$Tc = (1 - (1/RC)) * 100 \quad (\text{II.2})$$

On peut aussi définir le taux de bits par le rapport suivant :

$$\text{Taux de bits} = \frac{\text{Nombre d'octets après la compression}}{\text{Nombre de bits avant la compression}} \times 8 \text{ (bpp)} \quad (\text{II.3})$$

Ainsi en augmentant le taux de compression (en diminuant le taux de bits), on gagne en plus l'espace mémoire mais on dégrade la qualité de l'image après sa restitution ce qui risque de perdre totalement l'information. Par ce fait on introduit la notion de la «distorsion».[11]

b. Mesures de distorsion :

La distorsion est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale.

La mesure de distorsion utilisée généralement en compression d'image est l'erreur quadratique moyenne MSE (Mean Square Error). Cette grandeur est définie par la moyenne des écarts aux carrés entre le pixel (i, j) de l'image originale est celui de l'image reconstruite suivant l'équation suivante : [11]

$$MSE = \frac{\sum_{i=1}^N \sum_{j=1}^M [I(i,j) - Ir(i,j)]^2}{N \times M} \quad (\text{II.4})$$

Avec :

N x M : Taille de l'image.

I (i, j) : Pixel de l'image originale.

$I_r(i, j)$: Pixel de l'image résultante.

On référence cette erreur par rapport à la dynamique de l'image en décibels. On obtient un rapport signal sur bruit crête pour une image (dans notre cas $R=8$), appelé PSNR (Peak Signal to Noise Ratio) qui se traduit par l'équation suivante :

$$PSNR = 10 \log_{10} \frac{[2^R - 1]^2}{MSE} \quad (II.5)$$

Ou : $2^R - 1$ représente la dynamique de l'image.

Lorsque la reconstruction est parfaite, la MSE est nulle et le PSNR tend vers l'infini. Ces mesures de distorsion sont objectives et simples à calculer. Cependant elles ne rendent pas compte de la qualité visuelle engendrée par la compression. [11]

C. Entropie :

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

L'entropie d'ordre zéro d'une Image en niveau de gris dont les pixels sont codés sur R bits peut s'exprimer par :

$$H_0 = - \sum_{K=0}^{2^R-1} P(K) \log[P(k)] \quad \text{bpp} \quad (II.6)$$

$p(k)$: probabilité d'apparition de la valeur k .

L'entropie H d'une image originale fournit le débit minimal qu'il est possible d'atteindre par compression pixel par pixel sans dégrader l'image, et par là même un taux de compression sans perte maximal. [11]

II.4 Méthodes de compression :

II.4.1 Compression sans pertes :

Appelée aussi compression non destructrice, la qualité de l'image après décompression est la même que celle de l'image originale, le taux de compression de ce type est limité. Ce

type de compression on le trouve beaucoup dans le domaine où la précision est majeure comme l'image médicale (IRM par ex.) ou la télédétection (imagerie satellite par ex.).

a. Méthode par plages (Run-length encoding) :

Le codage RLE est très répandu dans les algorithmes de compression d'images. Ce codage est beaucoup moins efficace que l'autre, il est facile à implémenter, rapide à exécuter et représente une bonne alternative. Le codage RLE travaille sur la réduction de la taille physique des chaînes de caractères répétitives. Ces répétitions nomment en anglais un «run»

Le premier octet représente le nombre de répétition; en pratique cette valeur est comprise le plus souvent entre 1 et 128 ou 256. L'octet suivant représente l'octet à itérer [11].

L'efficacité du RLE dépend du nombre de transitions des niveaux de gris de l'image à coder. Plus le nombre de niveaux de gris est faible, plus le codage est meilleur. Le taux de compression atteint dépend de la complexité de l'image.

a.1. Principe de fonctionnement :

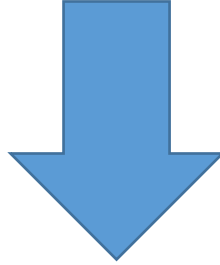
L'algorithme du RLE parcourt les données et recherche les octets identiques qui se suivent. Lorsqu'il trouve un même octet répété successivement au moins x fois, il crée une paire de données composée du nombre de répétitions suivi de l'octet en question.

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

Les différentes méthodes de compression

➤ Exemple :

0	0	0	0	63	63	144	14	14	14	14	14	12	12	12	11	11	11	11	11	11	11
---	---	---	---	----	----	-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



0	4	63	2	144	1	14	5	12	3	11	7
---	---	----	---	-----	---	----	---	----	---	----	---

a.1.1 Le codage de HUFFMAN :

Le codage de Huffman permet de créer des codes à longueur variable sur un nombre entier de bits. La procédure pour construire l'arbre est simple et élégante. Les symboles sont disposés individuellement sous forme d'une chaîne de nœuds et de feuilles et sont connectés par un arbre binaire. Chaque nœud a un poids qui est la fréquence ou la probabilité d'apparition des symboles. L'arbre est ensuite créé suivant les étapes suivantes (voir exemple figure II.2) :

- ✓ Les deux nœuds libres de poids les plus faibles sont identifiés.
- ✓ Un nœud parent de ces deux nœuds est créé ; il obtient un poids égal à la somme de celui de ces deux fils.
- ✓ Le nœud parent est ajouté à la liste des nœuds libres, et les deux fils sont enlevés.
- ✓ Un des deux nœuds fils est désigné comme le chemin pris à partir du nœud parent pour décoder un bit 0, l'autre nœud étant alors pris pour décoder un bit 1.
- ✓ Les étapes précédentes sont répétées tant qu'il reste plus d'un seul nœud libre. Ce nœud devient alors la racine de l'arbre.

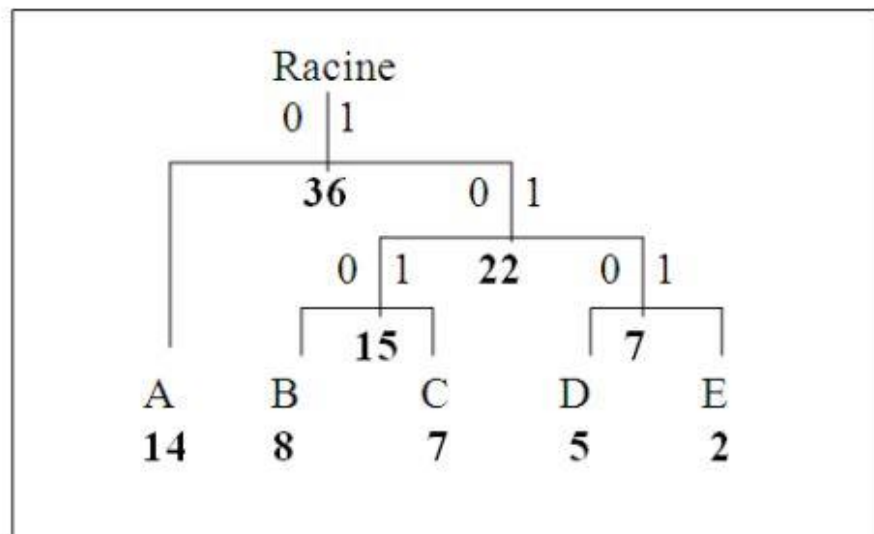


Figure.II.2. Exemple de l'arbre de Huffman

a.1.2 Codage arithmétique :

Le codage arithmétique est un codage récent utilisant un modèle statistique, tout comme le codeur de Huffman. Contrairement à ce dernier, il produit un code pour une séquence de symboles tout entier, et non pas un code par symbole.

Chaque nouveau symbole lu modifie de façon incrémentale le code de sortie. Ce code de sortie est un nombre à virgule flottante compris entre 0 et 1, dont le nombre de chiffres après la virgule correspond au nombre de symboles. Contrairement à Huffman, il n'est pas obligatoire que chaque code ait un nombre entier de bits [11].

II.4.2 Compression avec pertes :

Tout comme les méthodes sans pertes, les méthodes avec pertes peuvent avoir pour Domaine de travail le plan spatial ou le plan des fréquences.

a. Quantification scalaire :

La quantification scalaire consiste à remplacer un nombre très grand de symboles par un nombre de codes. C'est une opération irréversible très largement employée en compression. Par exemple le fait d'arrondir des nombres flottants à l'entier le plus proche est une quantification scalaire. [11]

b. Quantification vectorielle :

La quantification vectorielle découpe la matrice de symboles (pixels ou coefficients de la plane des fréquences) en petits blocs. Chaque bloc est codé par un indice dans un dictionnaire de blocs types. Le codage consiste à repérer dans le dictionnaire des mots codes disponible celui qui ressemble le plus au bloc d'entrée. Le critère consiste en général à minimiser l'erreur quadratique. Le décodage consiste simplement à remplacer l'indice par le mot code correspondant dans le dictionnaire. La création du dictionnaire de bloc types est une étape préalable et cruciale. [11]

II.5 La transformation :

II.5.1 La transformation de Fourier discrète DFT :

La transformation de Fourier discrète s'introduit quand il s'agit de calculer la transformée de Fourier d'une fonction à l'aide d'un ordinateur numérique. En effet, un tel opérateur ne peut traiter que des nombres et de plus en quantité par la taille de la mémoire.

Soit un signal discret $x(n)$, la DFT et la DFTI sont données par les formules suivantes [11] :

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{nk}{N}} \quad (\text{II.7})$$

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{j2\pi \frac{nk}{N}} \quad (\text{II.8})$$

II.5.2 Transformée en ondelettes :

Les ondelettes sont des familles de fonctions déduites d'une même fonction par des opérations de translation dans le temps 'b' et par dilatation en fréquence d'un facteur d'échelle 'a'. Elles ont trouvé, d'une part la puissance de leur théorie, des applications dans de nombreux domaines aussi variés que les mathématiques (analyse, probabilités, fractales), le traitement du signal et d'images (compression, astronomie, sismique), la physique (mécanique quantique, turbulence). En effet, cet outil permet la représentation de fonctions de x , dans $L^2(\mathbb{R})$ une base bien localisée en temps et en fréquence, offrant les

avantages de l'analyse de Fourier et s'affranchissant des inconvénients du manque de localisation de cette dernière. Les ondelettes sont normalisées suivant l'équation : [11]

$$\psi_{a,b}(t) = \frac{1}{|a|^{\frac{1}{2}}} \psi\left(\frac{t-b}{a}\right) \quad (\text{II.9})$$

Avec $a \in R^{+*}$ et $b \in R$

II.5.2.1. Transformation en ondelettes continue (TOC) :

La transformée en ondelettes d'une fonction est une représentation de $F \in L^2(R)$ celle-ci sur une base d'ondelettes choisie. Elle est dite continue car les paramètres de dilatation et de translation prennent des valeurs continues. Elle est définie par l'équation :

$$W(a, b) = TOC(f)(a, b) = \langle \psi_{a,b}, f \rangle = |a|^{-1/2} \int_{-\infty}^{+\infty} f(x) \bar{\psi}\left(\frac{x-b}{a}\right) dx \quad (\text{II.10})$$

Où $\bar{\psi}$: Représente le conjugué complexe de ψ

II.5.2.2 Transformation en ondelettes discrètes (TOD) :

La discrétisation de la DWT implique que les opérateurs de dilatation et de translation (a et b) prennent des valeurs discrètes suivant l'expression suivante : [11]

$$(a, b) = (a_0^j, kb_0 a_0^j) \quad (\text{II.11})$$

Ou $(j, k) \in Z^2$

Pour former ce qu'on appelle un réseau discret ; avec $a_0 > 1$ (le pas de dilatation) et $b_0 \neq 0$ (le pas de translation).

Ainsi, la nouvelle famille d'ondelettes s'établit par : [11]

$$\psi_{j,k}(t) = a_0^{-j/2} \cdot \psi(a_0^{-jt} - kb_0) \quad (\text{II.12})$$

Et la DWT s'établit comme suit : [11]

$$W(a, b) = \langle \psi_{j,k}, f \rangle = a_0^{-j/2} \int_{-\infty}^{+\infty} \psi(a_0^{-jt} - kb_0) f(t) dt \quad (\text{II.13})$$

II.6 Compression symétrique et asymétrique :

II.6.1 La Compression symétrique :

Si la quantité de travail pour compresser et décompresser est la même, alors la compression est symétrique car la même méthode est utilisée pour chacune ces opérations.[12]

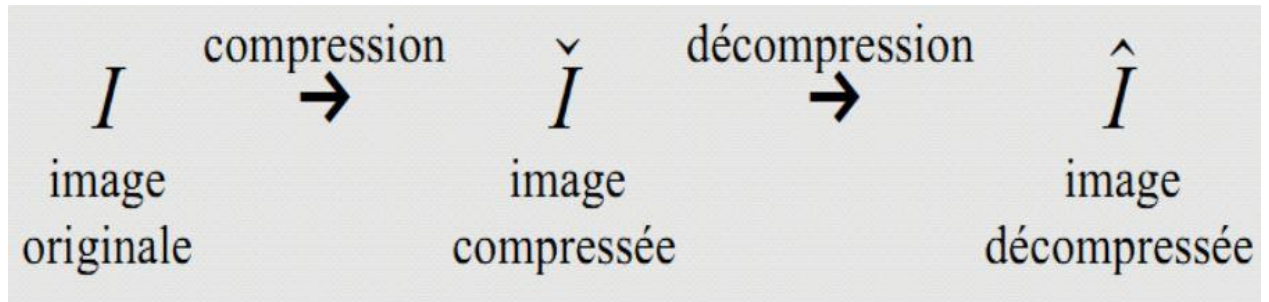


Figure II-3 : Schéma de la compression symétrique

II.6.2 La Compression asymétrique :

Si l'une des deux opérations demande plus de travail, on dit que la compression est asymétrique. Généralement la compression est plus lente que la décompression. [12]

II.7 La décompression :

Décompression 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.

II.8 Transformée en DCT :

La DCT est effectuée sur une matrice carrée $N \times N$ de valeurs de pixels et donne une matrice carrée $N \times N$ 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 plus petite $n \times n$ pixels. Ensuite la Transformée en Cosinus Discrète (DCT) est appliquée sur les pixels de chaque bloc. [13]

Transformée Directe :

$$F(u, v) = a(u)_0 a(v)_0 \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (\text{II.14})$$

Pour $u, v=0, 1, 2, 3 \dots N-1$.

Transformée Inverse :

$$F(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} a(u)_0 a(v)_0 F(u, v) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (\text{II.15})$$

Pour $x, y=0, 1, 2, 3 \dots N-1$.

Avec :

$$a(u) = \begin{cases} \left(\frac{1}{N}\right)^{\frac{1}{2}} & \text{pour } u = 0 \\ \left(\frac{2}{N}\right)^{\frac{1}{2}} & \text{pour } u = 0,1,2 \dots, N-1 \end{cases} \quad (\text{II.16})$$

II.9 Compression et décompression par la DCT :

II.9.1 Compression par la DCT :

Acquisition de l'image originale : Nous commençons en premier lieu par acquérir l'image originale (d'extension bmp) dans une matrice, pour lire et mémoriser l'image. Le chemin d'accès à l'image est bien spécifié en vue de la préparer pour les traitements.[13]

A. Transformée en DCT sur des blocs N*N :

L'image est découpée en bloc de N*N pixel. Ensuite la Transformée en Cosinus Discète (DCT) est appliqué sur les pixels de chaque bloc. A ce niveau, il n'y a pas encore de pertes de données [13].

B. Seuillage :

L'utilisation du seuillage est une caractéristique de débruitage. Celle-ci peut augmenter l'efficacité de la compression. Si les valeurs absolues des coefficients non nuls de la matrice DCT obtenus ne sont inférieurs à un certain seuil donné, ces derniers seront éliminés (mises à zéro). On définit le PCDZ comme le pourcentage de coefficients DCT nuls. Le seuillage

permet d'augmenter le nombre de répétitions de zéros. On peut dire qu'on perd les coefficients de faibles valeurs. La redondance des éléments nuls est importante dans l'étape de codage. Le seuillage est une étape non conservatrice.[13]

C. Quantification :

La quantification de chaque bloc regroupe les ensembles de valeurs proches. Ensuite, chaque amplitude originale sera remplacée par la valeur moyenne de l'intervalle, c'est-à-dire que l'étape de quantification consiste à diminuer la précision du stockage des entiers de la matrice DCT pour diminuer le nombre de bits occupés par chaque entier. C'est la partie non conservatrice de la méthode.[13]

II.9.2 Décompression par la DCT :

La phase de décompression est l'inverse de la phase de compression. Elle est divisée comme suit :

Extraction des paramètres de compression : Elle s'effectue par l'ouverture du fichier compressé afin d'extraire les paramètres (reconstruction de code binaire, les valeurs (DCTmax, DCTmin, Q, bloc, taille du fichier original, et le vecteur de répétition des éléments du vecteur SEQ)).[13]

A. Déquantification :

La déquantification nous permet d'extraire les coefficients DCT avec les moindres pertes possibles.[13]

B. Transformée en DCT inverse:

On applique la IDCT pour retrouver les éléments de la matrice représentant l'image à reconstruire. Ces coefficients servent à calculer l'erreur quadratique moyenne MSE et le rapport signal sur bruit crête PSNR [13].

C. Reconstruction de l'image :

Cette étape consiste à reconstruire l'image, en l'enregistrant dans un fichier de même extension que le fichier original (bmp) [13].

II.10. Les avantages et les inconvénients de DCT :

A. Les avantages :

- Performance de compression élevée.
- Sortie réel.
- Transformation rapide (surtout quand il est utilisé dans une image compressée).
- JPEG est destiné à des images de type photographique.

B. les inconvénients :

- Elle besoin de diviser le codage d'entrée en blocs 2d non chevauchants.
- Mauvaise transformation de l'ensemble de l'image.
- Le ratio de compression est bas.

II.11 Conclusion :

Après avoir étudié ce chapitre, des techniques de compression et des algorithmes ont été introduits. Photo, on peut tenir les points suivants :

On distingue deux types de compression en fonction de leur utilisation:

- Les données informatiques qui doivent rester les mêmes

(Textes, programmes informatiques, ...) utilisent une compression sans perte (RLE, LZW, Huffman, ...).

- Les données dont la qualité est limitée aux perceptions humaines (images, Utiliser des vidéos, des sons,) est une faible compression

Différents algorithmes de compression sont sélectionnés selon les critères suivants:

- Taux de compression (le rapport entre la taille du fichier compressé et Taille initiale du fichier)
- Qualité de la pression (sans / avec perte, puis pourcentage de Pertes)
- compression et élimination de la compression.

Chapitre III

Application Et Implémentation

III.1 Introduction :

L'objectif de ce travail est dans un premier niveau le développement d'une approche générale pour compression et de décompression à base de la transformée en cosinus discrète (DCT) sur les images numériques. Dans un deuxième niveau, extraire les critères de performances (Le taux de compression CR, La distorsion, l'erreur quadratique Moyenne.....).

Notre but c'est de trouvé un bon compromis entre ces mesures, c'est à dire un fort PSNR et un CR minimal, et bien sûr sans pour autant on perdre de la qualité visuel de L'image original, tout ça en respectant la contrainte du temps, plus le temps est cour meilleur est le système.

Dans ce chapitre, nous avons appliqué la compression/décompression d'image Dans le programme MATLAB et diverses simulations obtenues.

III.2 Définition de Matlab :

MATLAB est un langage de haute performance pour l'informatique technique. Il intègre le calcul, la visualisation et la programmation dans un environnement facile à utiliser où les problèmes et les solutions sont exprimés dans une notation mathématique familière.

Les utilisations typiques comprennent :

Mathématiques et calcul.

- Développement d'algorithme.
- Modélisation, simulation et prototypage.
- Analyse, exploration et visualisation des données.
- Graphiques scientifiques et techniques.
- Développement d'applications, y compris le bâtiment d'interface utilisateur graphique.

III.3 L'interface de programme d'application MATLAB :

Nous allons utiliser la version R2011a dans notre projet

Dans le premier, nous devons télécharger le logiciel Matlab à partir du site

Application et Implémentation

Officiel de Math Works, et l'installer.

Après l'installation de Matlab, nous l'ouvrons, l'image au-dessous illustre l'interface De Matlab.

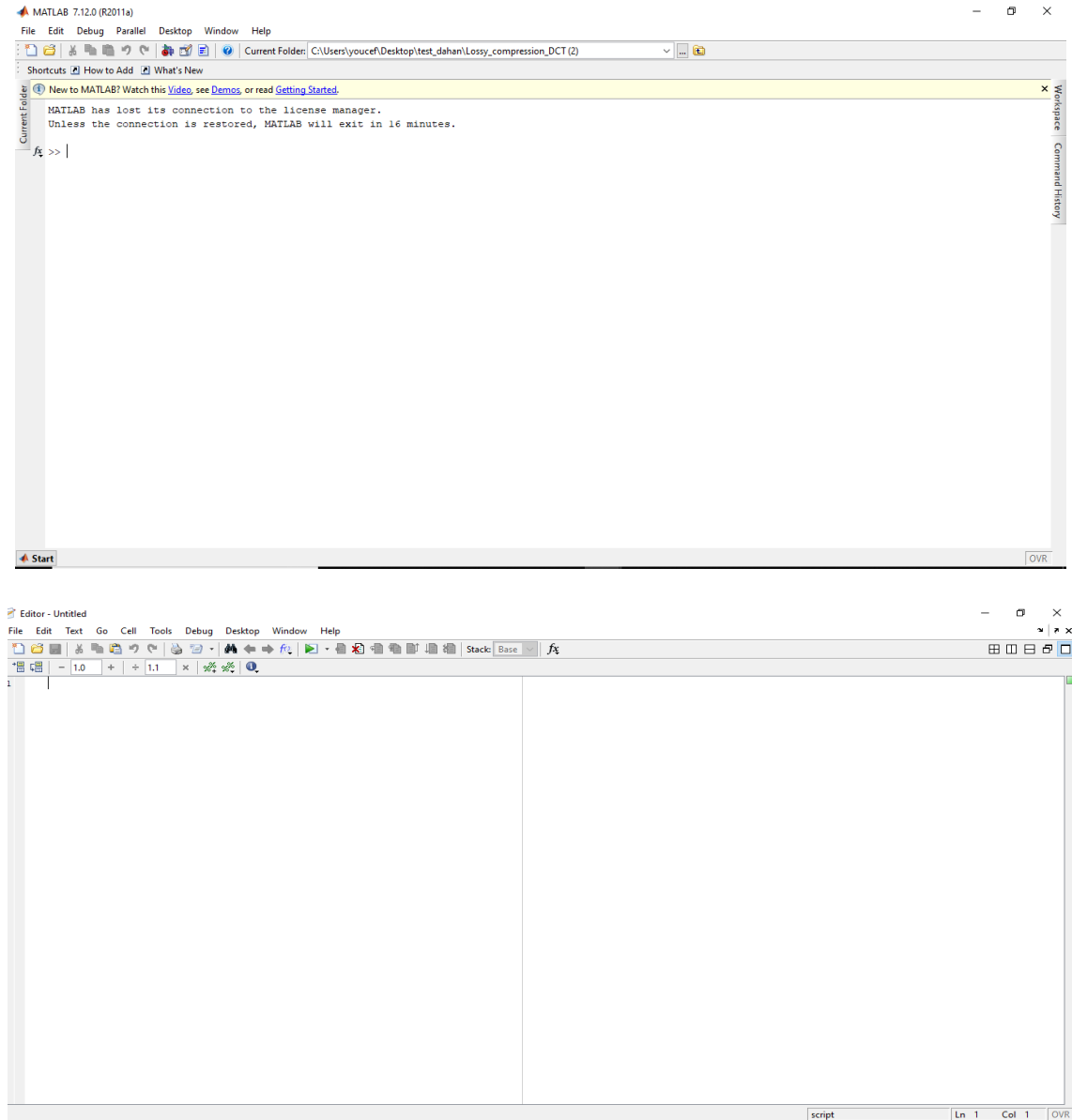


Figure III.1 : Interface graphique et IDE de MATLAB R2011a command Windows et fenêtre editor

III.4 L'implémentation de l'algorithme :

L'algorithme proposé pour cela se compose d'un certain nombre de fonctions, qui correspondent à chaque étape de compression/décompression. Ces étapes sont illustrées par la figure (III.2)

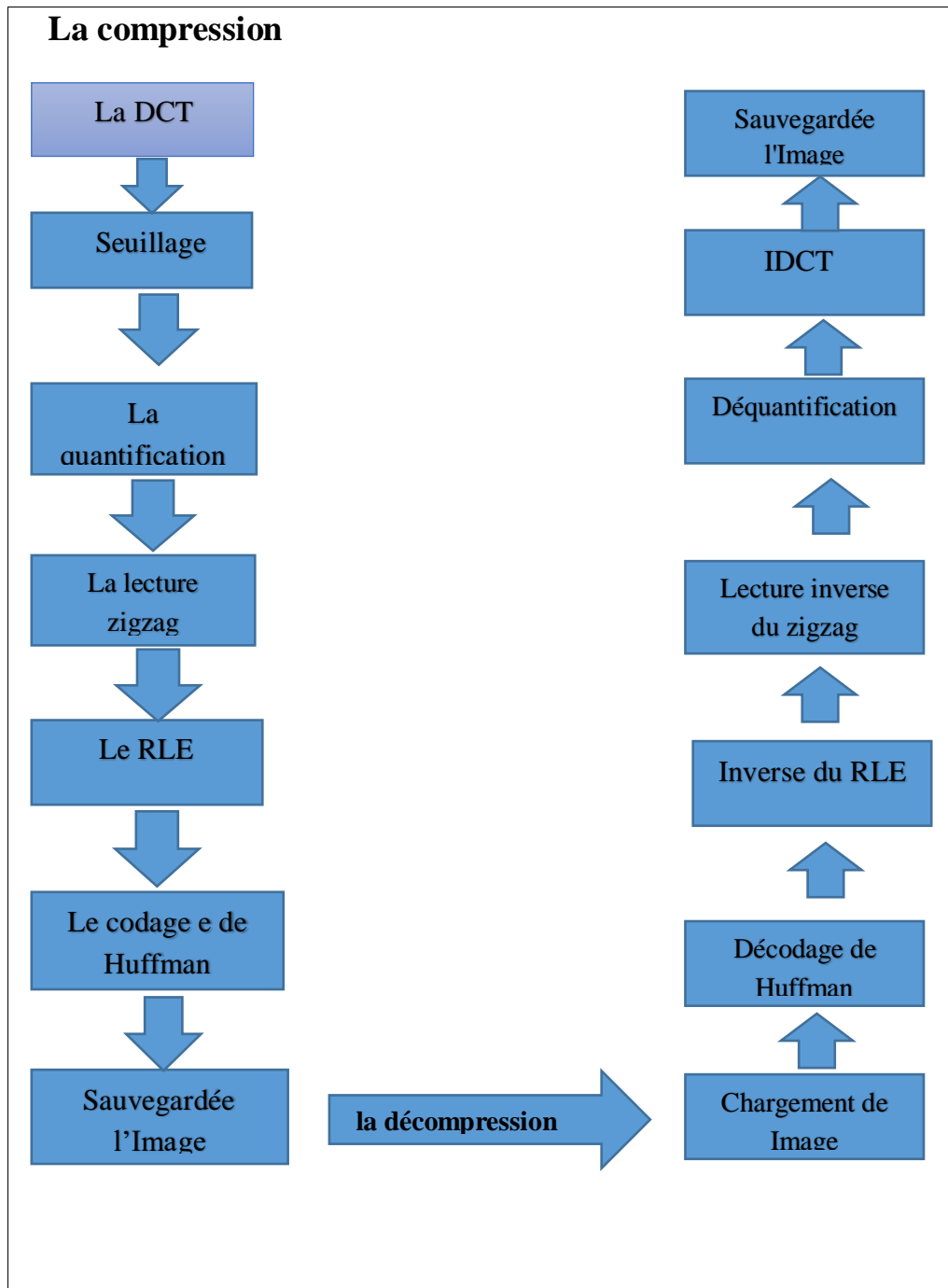


Figure III.2 Les étapes de compression/décompression

III.4.1 La compression et la décompression par la DCT :

Nous résumons les étapes de transformation, compression et décompression dans les sections suivantes.

III.4.1.1 Phase de Compression :

La phase de compression est divisée en plusieurs étapes :

Acquisition de l'image originale : Nous commençons en premier lieu par acquérir l'image originale (d'extension bmp) dans une matrice, pour lire et mémoriser l'image. Le chemin d'accès à l'image est bien spécifié en vue de la préparer pour les traitements.

Transformée en DCT sur des blocs $N*N$: L'image est découpée en bloc de $N*N$ pixel. Ensuite la Transformée en Cosinus Discrète (DCT) est appliqué sur les pixels de chaque bloc. A ce niveau, il n'y a pas encore de pertes de données.

de la DCT et de la IDCT (DCT inverse) données par :

$$F(u, v) = a(u)_0 a(v)_0 \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (\text{II.14})$$

Pour $u, v=0, 1, 2, 3, \dots, N-1$.

La transformée inverse de la DCT a pour équation :

$$F(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} a(u)_0 a(v)_0 F(u, v) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (\text{II.15})$$

Pour $x, y=0, 1, 2, 3 \dots N-1$.

$$a(u) = \begin{cases} \left(\frac{1}{N}\right)^{\frac{1}{2}} & \text{pour } u = 0 \\ \left(\frac{2}{N}\right)^{\frac{1}{2}} & \text{pour } u = 0, 1, 2 \dots, N-1 \end{cases} \quad (\text{II.16})$$

Application et Implémentation

62	65	65	68	72	84	42	18
48	57	60	60	65	60	48	26
45	54	58	60	65	60	48	26
58	64	72	57	74	49	50	64

37	48	53	56	53	48	40	33
58	50	52	64	60	51	42	31
82	72	65	76	76	72	53	38
79	71	60	67	55	58	48	40



Tableau III-1 Exemples la DCT

449.0000	50.1067	-56.6029	30.9237	-8.2500	4.6193	3.1508 -	10.8436
-24.9658	-8.6229	4.0178	-25.2594	-15.3529	-6.5918	-0.5251	6.3040
32.7241	18.7511	0.3395	16.4270	-5.8563	-1.5334	2.3195	-4.4199
19.9115	1.3888	-5.4081	9.1501	1.9265	1.2885	10.9566	-7.8130
-16.5000	6.5240	3.0587	-0.8197	2.2500	-0.0753	-0.4551	-1.9066
10.5129	1.1184	-7.4188	5.3972	-2.5611	-3.8771	2.2534	-1.6770
3.2223	-3.8759	1.5695	-0.8623	-4.3392	0.9649	7.4105	-6.7244
7.0866 -	8.9128	5.9762	1.6577	3.7771	1.2124	- 4.5621	4.3499

Application et Implémentation

Tableau III-2 Exemples du seuillage TH=4

449.0000	49.5114	-53.9994	25.2328	-5.7500	0	0	0
-11.5909	-9.7653	-22.0747	0	-15.3859	-7.3309	0	4.5435
25.0704	21.6779	-4.5267	24.7185	-7.2093	6.9869	0	-13.3481
-9.6728	0	-12.8502	9.8037	0	0	0	-6.2612
0	0	9.4890	-8.7474	0	-6.8972	5.2699	0
36.7894	4.8774	0	6.4322	4.9803	-6.3738	4.6962	0
-15.2553	0	0	4.2842	0	4.8197	0	-4.1917
-8.8719	0	-12.1790	-8.8954	8.7577	0	0	-7.6645

Quantification :

Les valeurs de la matrice DCT seuillée seront transformées pour être contenues dans l'intervalle $[0 \ 2^Q [$. La matrice de quantification sera calculée comme suit : On choisit la valeur de la quantification Q , (Q représente le nombre de bits nécessaires pour coder chaque élément de la matrice DCT). On détermine les deux valeurs MAX et MIN de la matrice DCT (DCT_{max} , DCT_{min}). Ensuite, on calcule la matrice $DCTQ$ par la formule suivante :

$$DCTQ(i, j) = \frac{(-1+2^Q)(DCT(i, j) - DCT_{min})}{DCT_{max} - DCT_{min}} \quad (III.4)$$

Avec $DCT(i, j) \neq 0$.

Tableau III-3 Exemples la matrice DCTQ

255	52	0	40	24	27	27	27
21	22	16	29	20	24	29	30
40	38	25	40	24	31	29	21
22	29	21	32	27	29	29	24
29	27	32	23	27	24	30	27
46	30	27	31	30	24	30	29

Application et Implémentation

20	27	26	30	27	30	27	25
23	27	21	32	23	29	27	23

La lecture zig-zag chaque masse de bloc est convertie en une ligne vectorielle comme le montre la figure suivante :

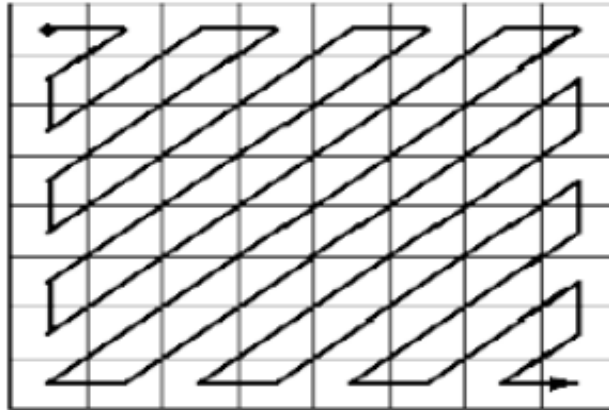


Figure III.3 Balayage Zig-zag du standard

Exemples :

255	52	21	40	22	0	40	16	38	22	29	29	25	29	24	27	20	40	21	27	46	20	30
-----	----	----	----	----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

32	32	24	24	27	27	29	31	27	23	27	27	23	27	26	31	27	29	29	30	21	29
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

24	30	30	21	32	27	24	30	24	27	30	30	23	29	27	27	29	25	27	23
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

le codage par RLE Le codage de Huffman est une méthode de compression statistique de données qui permet de réduire la longueur du codage d'un alphabet.

Codage RLE

Création d'une nouvelle séquence dans laquelle le deuxième élément correspond au niveau de gris et le premier élément correspond au nombre de pixels consécutif possédant ce niveau de gris. On code séparément le niveau de gris et l'occurrence de chaque pixel.

Exemple :

$v =$ 4 4 7 8 8 8 8 120 120 120 45 45 45 45 45 47 150 150 150 150 150
150 14

$RLE(v) =$ 4 2 7 1 8 4 120 3 45 5 47 1 150 6 14 1

Codage Huffman : Le codage de la matrice DCT quantifiée se fait comme suit : extraire les éléments non nuls de la matrice DCT quantifiée et leurs positions vers deux nouveaux

Application et Implémentation

vecteurs et matrices appelés respectivement NZ (non zéro) et (TAB). TAB indique par '1' la position spatiale d'un élément DCT non nul et par '0' également la position spatiale d'un élément DCT nul. TAB est transformé en un nouveau vecteur (8 bits par élément).

Les deux vecteurs NZ et TAB sont groupés en un seul vecteur SEQ pour être codés statiquement à l'aide du codage Huffman. - Création d'un fichier d'extension « DCT »: On mémorise les données obtenues dans un nouveau fichier dont l'extension choisie est DCT; ce qui nous permet d'identifier rapidement notre fichier compressé. Les tailles des fichiers original et compressé servent à calculer le taux de compression.

Exemple :

A1=0.2

A2=0.4

A3=0.2

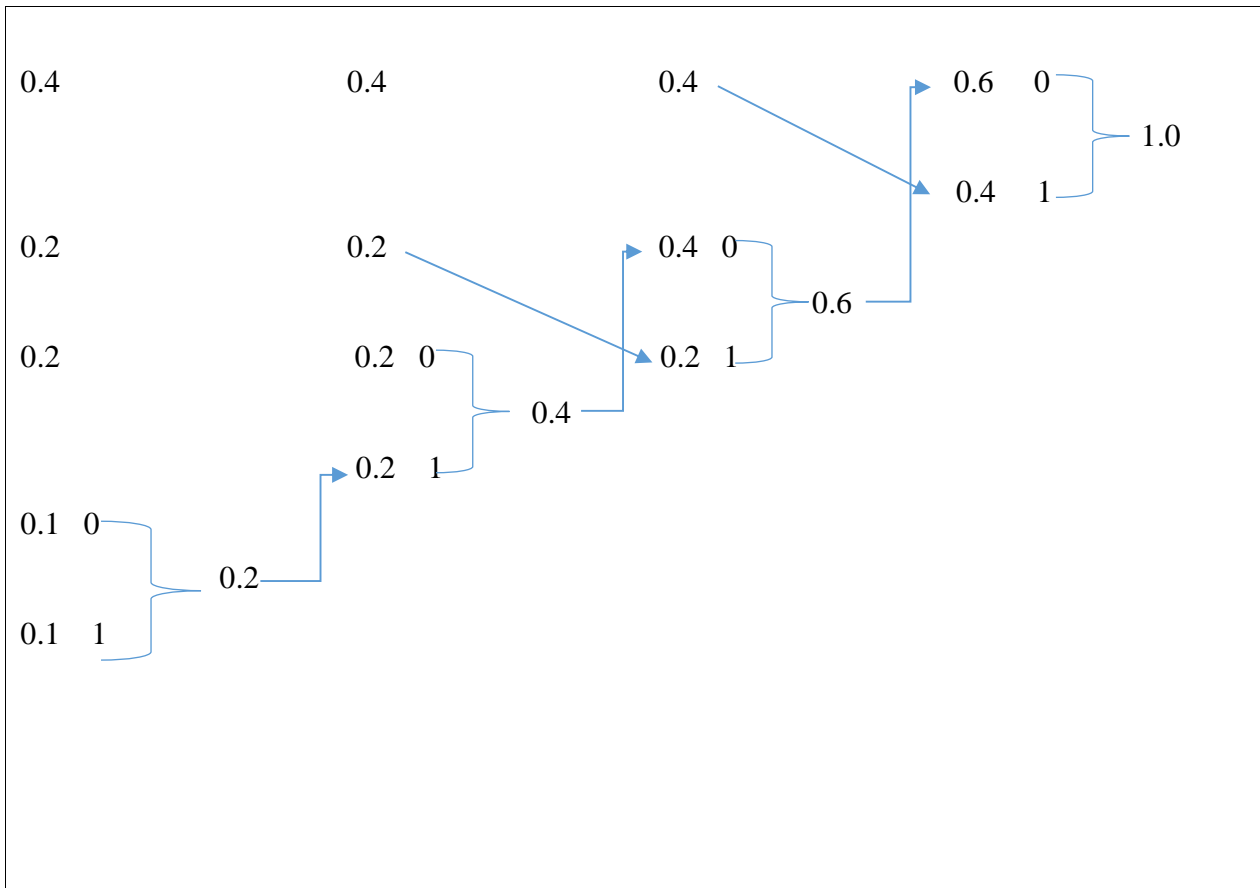
A4=0.1

A5=0.1

La solution :

		Codl	li
A2	0.4	1	1
A1	0.2	01	2
A3	0.2	000	3
A4	0.1	0010	4
A5	0.1	0011	4

Tableau III.4 exemple huffman code



Figures III.4 exemple huffman

$$\eta = \frac{H(x)}{\text{Larg}} * 100$$

$$H(x) = -[0.4 * \log_2(0.4) + 2 * (0.2) * \log_2(0.2) + 2 * (0.1) * \log_2(0.1)] = 2.12193$$

$$\text{Larg} = [1 * 0.4 + 2 * 0.2 + 3 * 0.2 + 4 * 0.1 + 4 * 0.1] = 2.2$$

$$\eta = \frac{2.12193}{2.2} * 100 = 96,45\%$$

III.4.1.2 Phase de décompression :

La phase de décompression est l'inverse de la phase de compression. Elle est divisée comme suit :

Extraction des paramètres de compression : Elle s'effectue par l'ouverture du fichier compressé afin d'extraire les paramètres (reconstruction de code binaire, les valeurs

(DCT_{max}, DCT_{min}, Q, bloc, taille du fichier original, et le vecteur de répétition des éléments du vecteur SEQ)).

Décodage Huffman : Le décodage statique est réalisé par le décodage Huffman sur le code binaire avec le vecteur de répétition des éléments SEQ reconstruits. Après cette opération, on obtient le vecteur SEQ qui contient les vecteurs NZ et TAB.

Déquantification: La déquantification nous permet d'extraire les coefficients DCT avec les moindres pertes possibles.

Transformée en DCT inverse: On applique la IDCT pour retrouver les éléments de la matrice représentant l'image à reconstruire. Ces coefficients servent à calculer l'erreur quadratique moyenne MSE et le rapport signal sur bruit crête PSNR.

Reconstruction de l'image: Cette étape consiste à reconstruire l'image, en l'enregistrant dans un fichier de même extension que le fichier original (bmp).

III.5 Critères de performances :

Des tests sur deux images médicales de type IRM ont été effectués en vue de vérifier les performances de la technique de compression étudiée. Ces performances sont basées sur les deux critères essentiels : Le taux de compression CR obtenu, et la qualité de l'image reconstruite.

Le taux de compression CR étant le rapport entre la taille de l'image après compression (Y) et celle de l'image originale (X):

$$CR = \frac{X}{Y} \quad (\text{III.5})$$

On définit le gain en compression par l'expression :

$$Gc(\%) = \left(1 - \frac{1}{CR}\right) * 100 \quad (\text{III.6})$$

Une compression efficace se traduit par une valeur de CR grand, et une compression peu efficace se traduit par un taux CR petit (ceci tout en gardant une bonne qualité visuelle). Par exemple, si la taille de l'image compressée représente 10% de la taille originale, nous avons un gain de 90% et la compression est efficace (CR = 10).

Mesures de distorsion : La distorsion (D) est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement

identique à l'image originale. La mesure de distorsion utilisée généralement en compression d'image est l'erreur quadratique moyenne MSE. Cette grandeur est définie par la moyenne des écarts au carré e_{min}^2 entre le pixel (m, n) de l'image originale $I(m, n)$, et le pixel (m, n) de l'image reconstruite $I_r(m, n)$;

$$MSE = \frac{1}{M*N} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} [I(m, n) - I_r(m, n)]^2 \quad (II.4)$$

On référence cette erreur par rapport à la dynamique de l'image en décibels. On obtient un rapport signal sur bruit crête pour une image dont le maximum est $2^R - 1$ dénoté PSNR (Peak Signal to Noise Ratio). Si le minimum de l'image est nul (image bien cadrée) on obtient le rapport signal sur bruit crête à crête noté PPSNR (Peak to Peak Signal to Noise Ratio). Lorsque la reconstruction est parfaite, la MSE est nulle et le PPSNR tend vers l'infini.

$$PSNR \text{ ou PPSNR} = 10 \cdot \log_{10}(2^R - 1)^2 - 10 \cdot \log_{10}(MSE) \quad (II.5)$$

Ces mesures de distorsion sont objectives et simples à calculer. Certaines méthodes de compression recherchent le meilleur compromis entre la performance et la distorsion, et optimisent des courbes taux-distorsion ou R(D).

III.6 les images de type IRM :

III.6.1 Test sur l'Image IRM1 de taille 348 x 374 :

Ceci est une image de l'épaule, pour le patient avec une fracture osseuse (Figure III.4 Et III.5).

L'image originale et l'image après compression Le tableau 1 résume les valeurs d'erreur(MSE), rapport de pression(CR), rapport signal sur bruit (PSNR), etLe pourcentage de transactions est 0 DCT (PCDZ), et Gain Pressure (GC).



Figures III.5 image originale

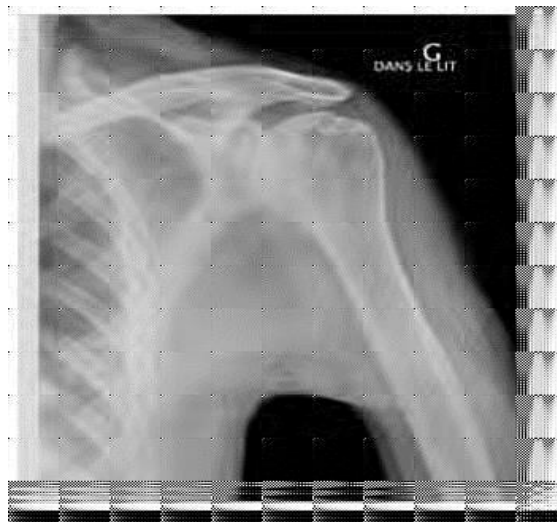


Figure III.6 Image Reconstituée

Image originale IRM1 et Figure III.5 image reconstruite, pour le bloc [32 32] et le codage Huffman. TH=30, CR= 57.8111, PSNR= 28.7946, G(%)=98.2702.

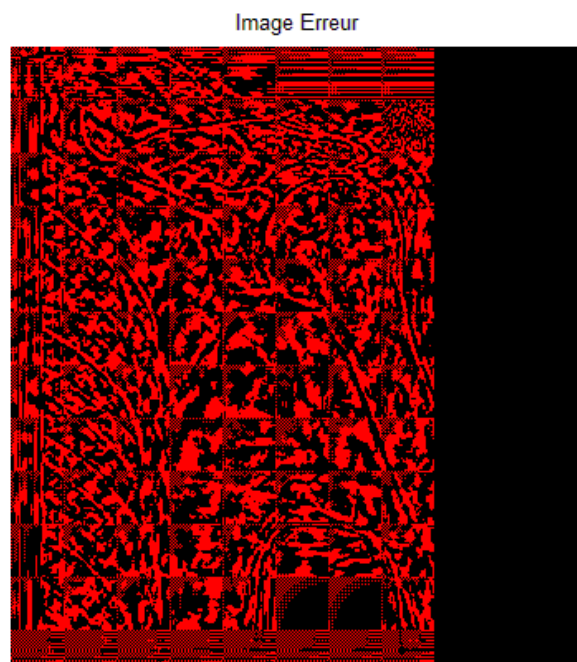
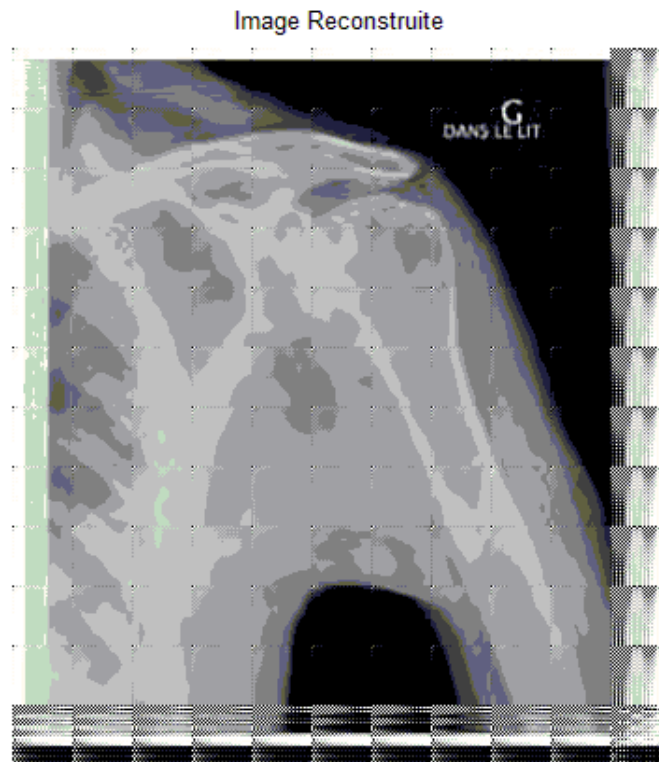


Figure III.7 Image reconstruite et image erreur, pour le bloc [32x32] et le codage Huffman. TH=30, CR= 57.8111, PSNR= 28.7946, G(%)=98.2702.

Application et Implémentation

Pour finir, dans le tableau ci-dessous sont résumés les temps de calcul mesurés pour Chaque étape de l'algorithme d'extraction que nous avons développé en *Matlab 2011* sur un *PC* de type Pentium 4 (*CPU 3 GHz*), fonctionnant sous Windows

TH	Les étapes	Temps (s)	Les étapes	Temps (s)
0	DCT	0.1106	Décod Huffman	24.2386
	Seuillage	0.0339	Décodage RLE	0.0324
	quantification	0.0033	zigzag inverse	0.3211
	Lecture zig zag	0.3658	Déquantification	0.0012
	Codage RLE	0.6783	IDCT	0.0450
	codage Huffman	3.0007		
2	DCT	0.1075	Décod Huffman	24.5671
	Seuillage	0.0265	Décodage RLE	0.0329
	quantification	0.0032	zigzag inverse	0.3237
	Lecture zig zag	0.3363	Déquantification	0.0012
	Codage RLE	0.6995	IDCT	0.0673
	codage Huffman	2.2212		
10	DCT	0.1061	Décod Huffman	23.8226
	Seuillage	0.0265	Décodage RLE	0.0326
	quantification	0.0036	zigzag inverse	0.3774
	Lecture zig zag	0.3316	Déquantification	0.0016
	Codage RLE	0.6684	IDCT	0.0495
	codage Huffman	2.0569		
20	DCT	0.1075	Décod Huffman	25.9343
	Seuillage	0.0271	Décodage RLE	0.0328
	quantification	0.0035	zigzag inverse	0.3589
	Lecture zig zag	0.3864	Déquantification	0.0013
	Codage RLE	0.5906	IDCT	0.0443
	codage Huffman	1.8836		
30	DCT	0.1076	Décod Huffman	23.9355
	Seuillage	0.0270	Décodage RLE	0.0323
	quantification	0.0034	zigzag inverse	0.3176

Application et Implémentation

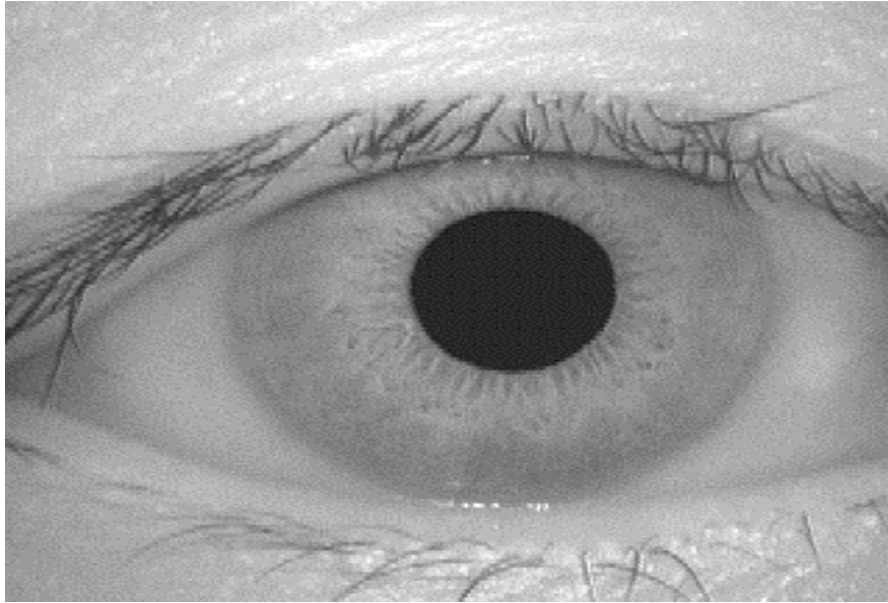
	Lecture zig zag	0.3310	Déquantification	0.0012
	Codage RLE	0.6033	IDCT	0.0435
	codage Huffman	1.6967		
50	DCT	0.1093	Décod Huffman	26.7770
	Seuillage	0.0313	Décodage RLE	0.0344
	quantification	0.0034	zigzag inverse	0.3498
	Lecture zig zag	0.3361	Déquantification	0.0012
	Codage RLE	0.5978	IDCT	0.0439
	codage Huffman	1.4016		
100	DCT	0.1531	Décod Huffman	22.1712
	Seuillage	0.0271	Décodage RLE	0.0326
	quantification	0.0029	zigzag inverse	0.3132
	Lecture zig zag	0.3495	Déquantification	0.0012
	Codage RLE	0.5655	IDCT	0.0422
	codage Huffman	1.0303		

Le Tableau III.5 mesuré le temps Les étapes de l'algorithme pour changer de seuillage l'Image IRM1

La comparaison entre les étapes de la chaîne algorithme montre que notre méthode présente un avantage certain concernant le temps de calcul particulièrement dans l'étape de la Décodage Huffman.

III.6.2 Test sur l'Image IRM2 de taille 320*380

C'est l'image de l'œil humain (Fig. III.7 et III.8). L'image originale et l'image après compression sont résumées dans le tableau 1, vitesse de pression et décompression. Tableau 1 Les valeurs d'erreur (MSE), le rapport de pression (CR), le rapport signal sur bruit (PSNR) et le pourcentage de transactions sont égaux à 0 DCT (PCDZ) et à la pression de gain (GC).



Figures III.8 image originale

Image Reconstituée

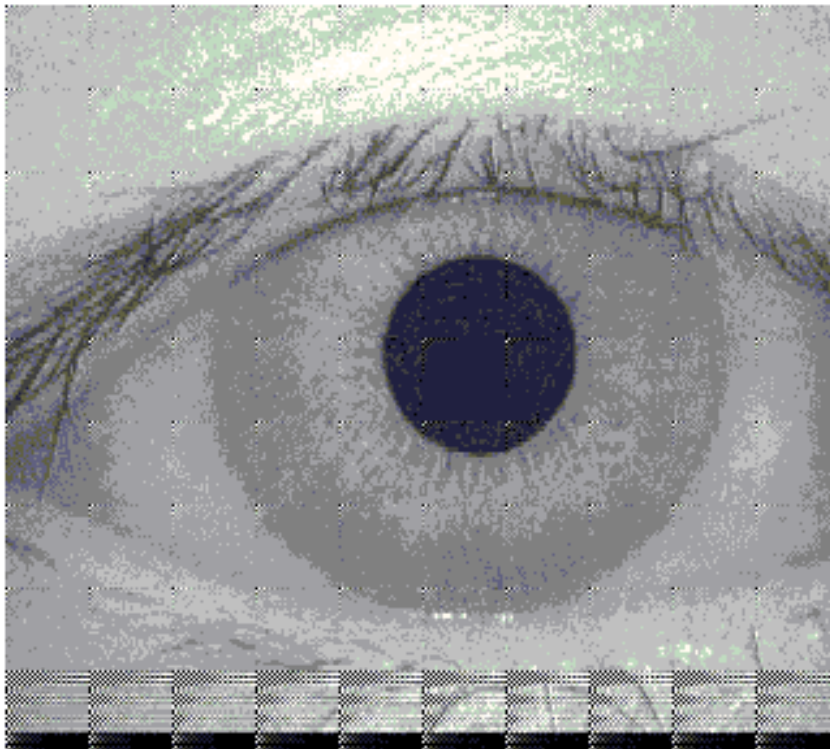


Figure III.9 Image Reconstituée

Image originale IRM1 et Figure III.8 image reconstruite, pour le bloc [32 32] et le codage Huffman. TH=10, CR= 11.3414, PSNR= 22.4017, G(%)=91.1827.

Application et Implémentation

Pour finir, dans le tableau ci-dessous sont résumés les temps de calcul mesurés pour Chaque étape de l'algorithme d'extraction que nous avons développé en *Matlab 2011* sur un *PC* de type Pentium 4 (*CPU 3 GHz*), fonctionnant sous Windows

TH	Les étapes	Temps	Les étapes	Temps
0	DCT Seuillage quantification Lecture zig zag Codage RLE codage Huffman	0.1441 0.0165 0.0021 0.4439 0.7590 4.3102	Décod Huffman Décodage RLE zigzag inverse Déquantification IDCT	24.9179 0.0344 0.3091 0.0012 0.0435
2	DCT Seuillage quantification Lecture zig zag Codage RLE codage Huffman	0.7205 0.0806 0.0175 1.6733 5.7081 31.5587	Décod Huffman Décodage RLE zigzag inverse Déquantification IDCT	23.4365 0.0332 0.3798 0.0012 0.0432
10	DCT Seuillage quantification Lecture zig zag Codage RLE codage Huffman	0.1752 0.0224 0.0020 2.5576 0.5837 2.5576	Décod Huffman Décodage RLE zigzag inverse Déquantification IDCT	23.8315 0.0342 0.3176 0.0012 0.0422
20	DCT Seuillage quantification Lecture zig zag Codage RLE codage Huffman	0.1970 0.0228 0.0020 0.3944 0.4546 1.5386	Décod Huffman Décodage RLE zigzag inverse Déquantification IDCT	23.7351 0.0344 0.3161 0.0012 0.0425
30	DCT	0.1580	Décod Huffman	24.1504

Application et Implémentation

	Seuillage	0.0227	Décodage RLE	0.0339
	quantification	0.0021	zigzag inverse	0.3584
	Lecture zig zag	0.3330	Déquantification	0.0012
	Codage RLE	0.3956	IDCT	0.0440
	codage	1.3465		
	Huffman			
50	DCT	0.1445	Décod Huffman	24.4560
	Seuillage	0.0229	Décodage RLE	0.0498
	quantification	0.0023	zigzag inverse	0.5936
	Lecture zig zag	0.3559	Déquantification	0.0012
	Codage RLE	0.3463	IDCT	0.0476
	codage	0.9568		
	Huffman			
100	DCT	0.1550	Décod Huffman	23.9764
	Seuillage	0.0308	Décodage RLE	0.0376
	quantification	0.0021	zigzag inverse	0.3433
	Lecture zig zag	0.3328	Déquantification	0.0012
	Codage RLE	0.3924	IDCT	0.0432
	codage	0.9412		
	Huffman			

Le Tableau III.6 mesuré le temps Les étapes de l'algorithme pour changer de seuillage l'Image IRM2

La comparaison entre les étapes de la chaîne algorithmique montre que notre méthode présente un avantage certain concernant le temps de calcul particulièrement dans l'étape de la Décodage Huffman.

III.6.2 Test sur l'Image IRM3 de taille 256x256 :

Ceci est une image normale (Fig. III.4 et III.5). L'image d'origine et l'image sont résumées après la pression dans le tableau 1, vitesse de compression et de décompression. Tableau 1 Les valeurs d'erreur (MSE), le rapport de pression (CR), le rapport signal sur bruit (PSNR) et le rapport de transaction sont égaux à 0 DCT (PCDZ) et à la pression de gain (GC).

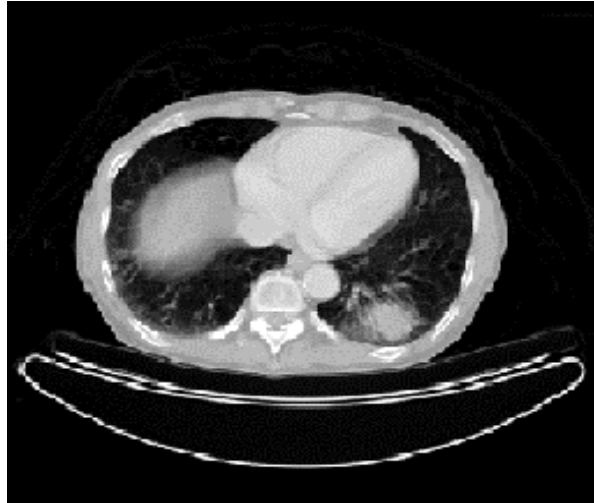


Image Reconstituée

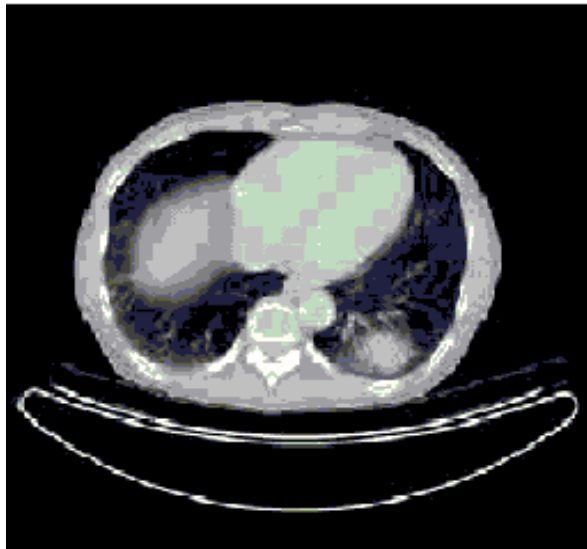
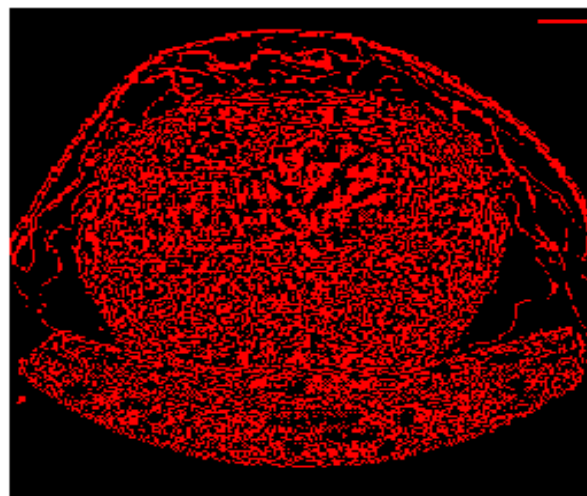


Image Erreur



Figures III.10 image originale reconstruite et erreur

Application et Implémentation

Pour finir, dans le tableau ci-dessous sont résumés les temps de calcul mesurés pour Chaque étape de l'algorithme d'extraction que nous avons développé en *Matlab 2011* sur un *PC* de type Pentium 4 (*CPU 3 GHz*), fonctionnant sous Windows

TH	Les étapes	Temps (s)	Les étapes	Temps (s)
0	DCT	2.2608	Décod Huffman	23.7675
	Seuillage	0.0496	Décodage RLE	0.0448
	quantification	0.0180	zigzag inverse	0.3734
	Lecture zig zag	3.9750	Déquantification	0.0012
	Codage RLE	5.2741	IDCT	0.0438
	codage	26.9100		
	Huffman			
2	DCT	2.2455	Décod Huffman	24.2864
	Seuillage	0.0793	Décodage RLE	0.0352
	quantification	0.0165	zigzag inverse	0.3213
	Lecture zig zag	4.6255	Déquantification	0.0012
	Codage RLE	5.7527	IDCT	0.0441
	codage	25.7604		
	Huffman			
10	DCT	2.2913	Décod Huffman	23.9980
	Seuillage	0.0953	Décodage RLE	0.0357
	quantification	0.0164	zigzag inverse	0.3441
	Lecture zig zag	4.3111	Déquantification	0.0012
	Codage RLE	5.4118	IDCT	0.0430
	codage	24.2723		
	Huffman			
20	DCT	2.2888	Décod Huffman	22.9382
	Seuillage	0.0942	Décodage RLE	0.0322
	quantification	0.0179	zigzag inverse	0.4208
	Lecture zig zag	4.1734	Déquantification	0.0012
	Codage RLE	4.9556	IDCT	0.0418
	codage	23.5560		
	Huffman			
30	DCT	2.2420	Décod Huffman	22.9399

Application et Implémentation

	Seuillage	0.0973	Décodage RLE	0.0329
	quantification	0.0179	zigzag inverse	0.3090
	Lecture zig zag	4.2048	Déquantification	0.0012
	Codage RLE	21.0357	IDCT	0.0438
	codage			
	Huffman			
50	DCT	2.2375	Décod Huffman	23.3672
	Seuillage	0.1104	Décodage RLE	0.0323
	quantification	0.0274	zigzag inverse	0.3256
	Lecture zig zag	4.3040	Déquantification	0.0012
	Codage RLE	4.4231	IDCT	0.0427
	codage	15.3372		
	Huffman			
100	DCT	2.5725	Décod Huffman	24.4899
	Seuillage	0.1088	Décodage RLE	0.0529
	quantification	0.0193	zigzag inverse	0.3278
	Lecture zig zag	5.2595	Déquantification	0.0012
	Codage RLE	4.4391	IDCT	0.0424
	codage	13.1238		
	Huffman			

Le Tableau III.7 mesuré le temps Les étapes de l’algorithme pour changer de seuillage l’Image IRM3

La comparaison entre les étapes de la chaîne algorithmique montre que notre méthode présente un avantage certain concernant le temps de calcul particulièrement dans l’étape de la Décodage Huffman.

III.7 Les Tableaux : Valeurs de CR, PSNR, MSE, et le G(%) en fonction des différents paramètres avec le codage Huffman pour l’image IRM1 IRM2 et IRM3 :

Application et Implémentation

BLOC	TH	CR	PSNR (db)	MSE	GC(%)
[32x32]	0	34.2656	4.7696	85.0309	97.0816
	2	34.2656	4.7696	85.0309	97.0816
	10	39.4360	4.7790	84.8480	97.4642
	20	51.4910	4.7556	85.3057	98.0579
	30	57.8111	4.7292	85.8261	98.2702
	50	75.5088	28.7192	4.6538	98.6757
	100	101.4962	91.3066	4.4604	99.0147

Le Tableau III.8 Valeurs de CR, PSNR, MSE, et le G(%) pour l'image IRM1

BLOC	TH	CR	PSNR(db)	MSE	GC(%)
[16x16]	0	3.4405	1.5593	178.0790	70.9342
	2	3.4405	1.5593	178.0790	70.9342
	10	5.4581	1.5810	177.1896	81.6786
	20	11.8566	1.4640	182.0287	91.5658
	30	18.0791	1.3462	187.0320	94.4688
	50	28.2561	1.1246	196.8253	96.4609
	100	44.2033	0.7763	213.2584	97.7377

Le Tableau III.9 Valeurs de CR, PSNR, MSE, et le G(%) pour l'image IRM2

BLOC	TH	CR	PSNR (db)	MSE	GC(%)
[8x8]	0	5.4609	39.0102	0.0320	81.6880
	2	5.4609	39.0102	0.0320	81.6880
	10	5.8103	30.3509	0.2352	82.7893
	20	5.8103	30.3509	0.2352	82.7893
	30	7.5057	23.4838	1.1433	86.6767
	50	8.9078	19.0051	3.2065	88.7738
	100	14.1588	13.3825	11.7028	92.9372

Le Tableau III.10 Valeurs de CR, PSNR, MSE, et le G(%) pour l'image IRM3

III.8 Conclusion

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

Dans ce chapitre, nous avons testé les performances, cependant, une nette détérioration de la qualité les images reconstruites ont été observées le seuil est supérieur à 20. Cette étude de performance La technique de compression DCT peut être étendue à autres types d'images.

La compression en mode codage DCT est une compression avec perte non adapté aux besoins de précision, mais serait le format le mieux adapté pour les mémoires ou pour le web la rapidité de transmission, cela dit, un format nouveau a été développer pour les applications qui demande une fidélité maximum de l'image traité à l'image réel, comme l'imagerie médicale, c'est le format JPEG2000 qui a les mêmes avantage du JPEG mais basé sur une compression sans perte en utilisant une nouvelle transformé appelée la transformé en ondelettes.

En général, on peut dire que l'approche proposée permet d'atteindre de très bons résultats des points de vue qualité des images reconstruites d'une part et le taux de compression d'autre part. En effet, elle permet de réduire énormément la quantité de données contenues dans les images tout en conservant leurs propriétés originales. Ces résultats dépendent du type de l'ondelette, de l'algorithme de DCT et du type de codage entropique utilisés.

Conclusion

Général

Conclusion générale

Les techniques de compression avec pertes assurent généralement des taux de compression relativement élevés mais avec un certain degré de dégradation. Les objectifs des techniques de compression est donc de réduire plus et d'altérer les moins possible. Parmi les techniques de compression avec perte nous avons la méthode basée sur les transformations par la DCT.

L'objet de ce travail a été la recherche et l'évaluation de techniques de compression avec pertes pour améliorer le taux de compression et le rapport signal à bruit en pic. Nous avons choisi d'utiliser une méthode non conservatrice, car elle permet d'accéder à des taux de compression extrêmement élevés sans modifier l'image. Elle ne peut pas être détectée par l'œil humain. En outre, de nombreuses applications vous permettent de choisir la qualité d'image compressée à l'aide de matrices de quantification. . Cette méthode a l'avantage d'une grande précision de reconstruction.

Ce mémoire présente un système de compression/décompression d'image par la technique de transformée en cosinus discret, qui a été examiné en utilisant une ensemble des images à fond gris, afin de vérifier l'évaluation et l'efficacité de la chaine algorithmique proposée. Dans un premier niveau le développement d'une approche générale pour compression/décompression d'image par utilisation des techniques de traitement d'information et l'image (codage/décodage Huffman, DCT/ IDCT, RLE/IRLE).

Ensuite, notre expérimentation a été effectuée sur des images de différentes tailles. Les test ont été effectués sur les différentes formes de la quantification pour chaque image. Nous avons fait un prélèvement des mesures les performances de l'approche proposée (Le taux de compression CR, La distorsion, l'erreur quadratique Moyenne.) et le temps de chaque étape de l'algorithme de compression/décompression ainsi que le temps globale de calcul pour chaque test.

Bibliographie

- [1] C. Diltsm, Numériser vos collections, RCIP Octobre 2001.http://www.chin.gc.ca/Francais/Contenu_Numerique/Guide_Gestionnaires/
- [2] C.Houassine, segmentation d'images par une approche biomimétique hybride. université universite m'hamed bougara- boumerdes. 2012.
- [3] A.Medjaoui , f.fares , segmentation des images par contours actifs : application sur les Images Satellitaires à Haute Résolutions, Université Abou Bakr Belkaid– tlemcen.2012
- [4] T.Mohamed ,D.Djafar, la découverte des concepts sémantiques cachés avec plusieurs niveaux d'abstraction pour la recherche d'images,université kasdi merbah ouargla.2013
- [5] M Sandeli, traitement d'images par des approches bio-inspirées application à la segmentation d'images, université constantine 2. 2014.
- [6] N MERABET,M MAHLIA, recherche d'images par le contenu, université aboubakrbelkaid– tlemcen.2011
- [7] C. Lepec, Les Graphiques Sur Ordinateurs, N. RIMOUX & A. COLIN Editeurs. Paris, 1991.
- [8] Mark Nelson and Jean-loup Gailly, "The data compression book" second edition, League for Programming Freedom, Cambridge MA 02139, 1995
- [9] R. Sylvain, "La Compression de Données", Club Photoshop de Nantes. Conférence du 14 octobre 1999.
- [10] MEMOIRE DE PROJET DE FIN D'ETUDES Université Abou Bakr Belkaid– Tlemcen Pour l'obtention du diplôme de master en infor Thème Synthèse et Etude comparative sur les méthodes de compression d'images DCT et DWT
- [11] Master ACADEMIQUE Présenté par : ZEGHIB Maymouna SEROUTI Abla Intitulé Compression des images avec curvelet Soutenu le 2014-2015 UNIVERSITE ECHAHID HAMMA LAKHDAR – EL OUED
- [12] Mémoire MASTER Thème Étude et Simulation de L'Algorithme de Compression d'Images JPEG Par:MUSTAFA Abdelfattah BEN SAHA Abdelkader Soutenu publiquement le : 19/06/2018 Université de Ghardaïa
- [13] Département d'Electronique, Université de M'sila, Algérie teme Application de la DCT et le Codage de Huffman pour la Compression Djamel CHIKOUCHE* Ridha BENZID*and Khaled ROUABAH** des Images Médicales.

Résumé

Les images numériques rencontrent des problèmes avec le traitement le stockage et leurs transmissions en raison de leur grande taille. Le but de la compression des images est de réduire la quantité de volume que nous avons besoin de stocker ou d'envoyer comme l'image, audio et vidéo et il y a un ensemble d'algorithmes de compression qui répondent aux besoins de nouvelles applications.

Cette mémoire étudie la compression des Images à niveaux de gris, et fait avec un algorithme de conversion de section distinct et en imitant à l'aide du logiciel MATLAB et en discutant des différents résultats obtenus.

Mots-clés: compression d'image, Images à niveaux de gris, MATLAB, simulation, algorithmes.

ملخص

الصور الرقمية تواجه مشاكل في المعالجة التخزين و في ارسالها نتيجة حجمها الكبير الهدف من ضغط الصور هو التقليل من كمية حجمها التي نحتاج إلى تخزينها او ارسالها مثل الصورة والصوت والفيديو وهناك مجموعة من الخوارزميات الضغط التي تلبي احتياجات تطبيقات الجديدة.

هذه المذكرة تعمل على دراسة الضغط صور ذات مستوى رمادي، وتم اجراء عليها خوارزمية تحويل المقطع المنفصل وبواسطة محاكات باستخدام برنامج MATLAB ومناقشة مختلف النتائج التي تم الحصول عليها.

كلمات مفتاحية: ضغط الصورة، صور ذات مستوى رمادي، MATLAB ، محاكاة، الخوارزميات.