Mohamed Khider University of Biskra
Faculty of Science and Technology
Electrical Engineering Department

MASTER DISSERTATION

Science and Technology
Electrical Engineering
Field: Automatics
Option: Automatics and Industrial Computing

Ref.: ………./2022

Presented and supported by:
LEHRAKI Selsabile
The: ……………………….

Compression of color image by DCT

Jury:

Mrs.	MEGHERBI Hassina	Prof	University Of Biskra	President
Mr.	MESSAOUDI Abdelhamid	MCA	University Of Biskra	Supervisor
Mr.	BENCHAABAN Fateh	MCA	University Of Biskra	Examiner

University Year: 2021 - 2022

Mohamed Khider University of Biskra
Faculty of Science and Technology
Electrical Engineering Department

MASTER DISSERTATION

Science and Technology
Electrical Engineering
Automatic

Ref.: Enter the document reference

Compression of color image by DCT

Presented By:

LEHRAKI Selsabile

Favorable opinion of the supervisor:

Mr. MESSAOUDI Abdelhamid

Signature Favorable opinion of the President of the Jury

Mrs. MEGHERBI Hassina

Stamp and signature

DEDICATION

My great parents, the reason for my existence, who never stop giving in countless ways. and who believe in me and my constant source of support and encouragement.

To my sister Douaa who leads me through the darkness with light of hope and support.

To my brother Abdou and his wife and thier childrens: Tesnim, Mohamed, my soul Anas, Adam and Ibtehal.

To my littel brother Yahya one day you will also write a dedication.

To my sister Aya who left us too soon, i wish we meet in heaven.

My friends who encourage and support me: Yasmine, Doucha and Chaima.

To all the people in my life who touched my heart.

ACKNOWLEDGEMENT

I'm extremely grateful to god for giving me the ability to overcome the difficulties, and gave me hope after every disappointment.

I would like to thank all the teachers who guided me through my university journey.

I would like to express my gratitude, and deep respect to my supervisor Mr. MESSAOUDI Abdelhamid for helping, advising, and generously providing me with knowledge and expertise, for his unconditional patience.

I am deeply indebted to Mr.BENCHAABAN Fateh for his constant quest to make us distinguished, and for giving us a positive energy during his sessions. I could not have undertaken this journey without his wise advice, which helped me to change my thinking.

I would also like to present my deep gratitude to Mrs.MEGHERBI Hassina to be part of the members of the jury. for beaing such an amazing teacher for two years. I owe her my most sincere thanks.

I would like to express my gratitude to the people who contributed and helped me during the writing of this thesis: Chaima and Douaa

Abstract

The main goal of this work is the study of the effect of the color space in color image compression; i.e, which color space achieves better compression ratio. based on this, an algorithm has been developed containing a discrete cosine transform (DCT) and an arithmetic coding called TRE; Its idea is similar to JPEG. Applying the algorithm on color images, showed that better performance can be achieved with YCbCr transform compared to RGB color space, and blocks of size (16x16) are better than (8x8) with the quantization size being 8 to this compression scheme.

Key words: Color image compression, discrete cosine transform (DCT), PSNR,Color Spaces, Adaptive Scanning,Two Role Encoder (TRE).

Résumé

L'objectif principal de ce travail est l'étude de l'effet de l'espace colorimétrique dans la compression des images couleur ; c'est-à-dire quelle couleur permet d'obtenir un meilleur taux de compression. sur cette base, un algorithme a été développé contenant une transformée en cosinus discrète (DCT) et un codage arithmétique appelé TRE. son idée est similaire à JPEG.

L'application de l'algorithme sur des images couleur a montré que de meilleures performances peuvent être obtenues avec la transformation YCbCr par rapport à l'espace colorimétrique RVB, et des blocs de taille (16x16) sont meilleurs que (8x8) avec la taille de quantification étant de 8 pour ce schéma de compression.

Mots clés: Compression d'images couleur, transformée en cosinus discrète (DCT), PSNR, espaces colorimétriques, balayage adaptatif, codeur à deux rôles (TRE).

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
AI	Adobe Illustrator
BMP	Windows Bitmap
bpp	Bits Par Pixel
CMY	Cyan-Magenta-Yellow
CMYK	Cyan-Magenta-Yellow-Black
CR	Compression Ratio
CWT	Continuous Wavelet Transforms
DCT	Discrete Cosine Transforms
DPI	Dots Per Inch
DWT	Discrete Wavelet Transforms
EBCOT	Embedded Block Coding with Optimized Truncation
ESP	Postscript / Encapsulated Postscript
FDCT	Fast-DCT
GIF	Graphic Interchange Format

ICT	Irreversible Color Transform
IDCT	Inverse Discrete Cosine Transform
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
HSL	Hue-Saturation-Luminance
MSE	Mean Square Error
PDF	Portable Document Format
PPI	Pixels Per Inch
PSNR	Peak Signal to Noise Ratio
PRD	Percent Root-Mean-Square Difference
RCT	Reversible Color Transform
RLE	Run Length Encoding
ROI	Region Of Interest
SVG	Scalable Vector Graphics
TIFF	Tag Image File Format
TRE	Two-Role-Encoder
VQ	Vector Quantization
SQ	Scalar Quantization
YCbCr	Y: Luminance, Cb: Chrominance Bleu, Cr: Chrominance Red

General introduction

In the last decades, we saw the sweeping of images in all fields, in medicine, social media, and education.... As students, we note its importance and assistance to us in our studies. All these images are compressed according to certain algorithms, it is not possible to store 1000 GB in a 64 GB phone, for example, and send it within a few minutes.

Image compression, which for a long time was the domain of a small group of engineers and scientists, is now everywhere (ubiquitous). Image compression is the process of encoding or converting an image file in such a way that it consumes less space than the original file, and that is done by reducing the amount of data in an image (removing redundant data) with keeping the resolution and the quality of the image. The inverse process is decompression, it is applied to the compressed data to get the reconstructed image. The objective of image compression is to storage the images and without taking a big memory space, and minimize the duration of transmission or downloading it.

Generally, there are two types of data compression: lossless compression which do not cause a significant reduction in data size while ensuring high data quality (this type are often used in medical or military); And lossy compression which can significant reduction in data size but it Will produce a little confused image. These Data can be characters, sequences of numbers that are generated by a processes.

There are many images compression techniques available for compressing images, such as those based on DCT (Discrete Cosine Transform) like JPEG (Joint Photographic Experts Group), DWT (discrete wavelet transform) such as JPEG2000, quantization, etc. The common goal of all these techniques is to obtain a high compression rate.

Usually, color images are compressed after transforming color space; In this thesis the main objective is the study of the effect of color spaces in compression of color images.

This thesis is organized according to the following chapters:

Chapter I Is devoted to presentation of the different color spaces and the Characteristics of a digital image (Pixels, dimension, etc.), and the files of saving images.

Chapter II Focus on compression and its various methods: lossless compression (RLE, Huffman...), and lossy compression (quantization, transformation...).

Chapter III Highlighted JPEG and JPEG2000 standard and the different steps to how they work; Then the performance criteria (PRD, CR, bpp...).

Chapter IV Presents the used compression method while validating its performance by making a comparison with a method based on the DCT transform combined to an adaptive block scanning.

Chapter I

Color and image fundamentals

I.1 Introduction

The human perception of color is linked to the physiology of the eye. The retina is made up of two different types of cells: cones and rods. In its center, the vision is mainly ensured by cones, which are of very good quality and whose quality depends on the lighting. In the periphery, the vision is mediocre and there is no notion of color, especially when working in low light, guaranteed mainly by rods. Visual perception is a combination of sensors (eyes) and analytical systems (brain). Color perception is defined as consciousness [1].

A digital image can be generated in many ways. The most common methods use a digital camera, video recorder, or image scanner. However, digital images are also generated by image processing algorithms, by data analysis that yield two-dimensional discrete functions, and by computer graphics and animation.

I.2 Concept of light

Light is a form of energy emitted on an electromagnetic wave whose wavelength varies from 380 to 780 nm and it is what the human eye can see. outside this field there are ultraviolet rays if the wavelength is below 380 nm, and if above 780 there are infrared rays which the human eye cannot see. Moreover, Light has a speed of about 300,000 km/s and a frequency of about 600,000 GHz [1].

I.3 Concept of color

Color is the visible aspect (visible spectrum) of light,we can see the different colors that make up the visible aspect by separating the white light by a prism, and that is how the colors are spread from red, orange, yellow, green, and blue to purple, as shown in Figure I.1. In general, the human eye therefore breaks down the color into a limited number of components (red, green, and blue) and with mixing these three components we can produce multi and different colors [1].

Humans perceive and distinguish colors with their vision system which interprets colors through the amounts of light of various wavelengths. Objects can be defined through the reflection of the light on them. Also,Human beings can differentiate up to about two million different colors [2].

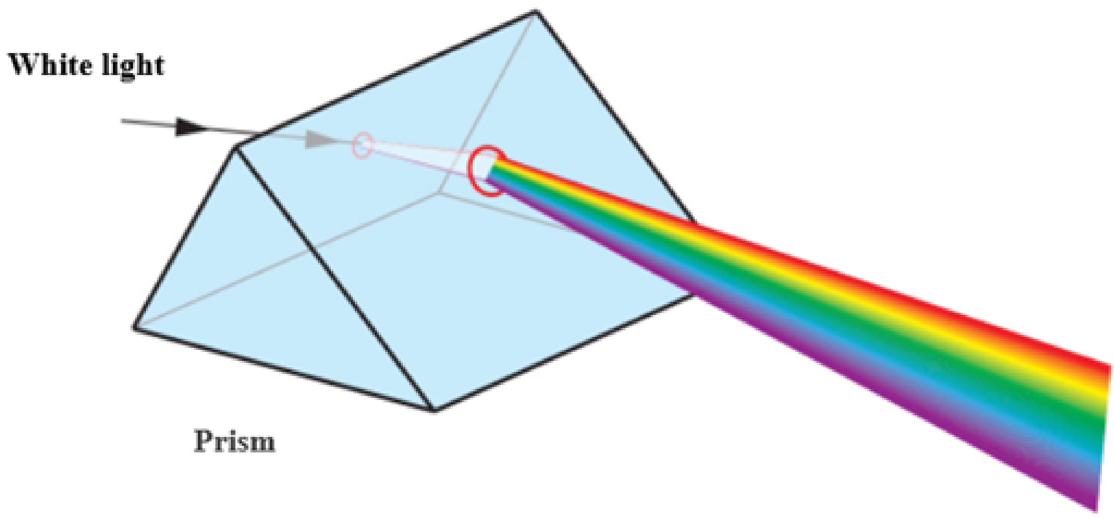


Figure I.1: A process of separating the white light by a prism.

I.4 Colors Spaces

Color space is a notation by which the human eye can perceive the visible electromagnetic spectrum, There are many color spaces working to include all perceptible colors, and it works to differentiate between colors in perception, but so far there is no agreement on the best color space yet. Here we will explain some of the following color spaces: RGB, CMY, HSV, YCbCr [2].

I.4.1 RGB space

This is a color system very common based on tri-chromatic Red, Green, Blue and it is the basis for calling it RGB. The RGB model proposes to encode each color component on one byte, which corresponds to 256 intensities of each of the three primary colors, if we take a value of each color and mix it, it produces 16777216 (256×256×256=16777216) different colors, that is more than the human eye can discern. this color space usually represented in 3D cube as figure I.2 shown.

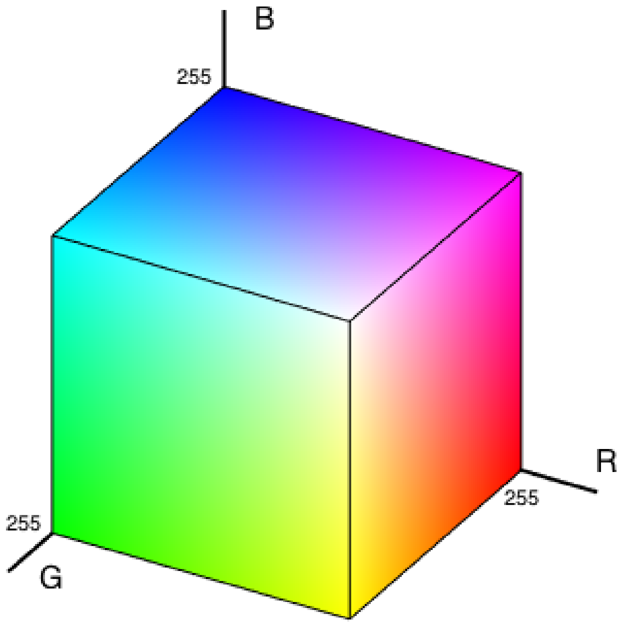


Figure I.2: An illustration of RGB color space as a 3D cube

Generally, white color appears if the value 256 of the three colors are mixed, but if the value of each color is zero, black appears, and this is what figure I.3 will show [1].

I.4.2 CMY space

The CMY color space is derived from the RGB space (figure I.4), which is an abbreviation for the three colors following: Cyan, Magenta et Yellow.this color space

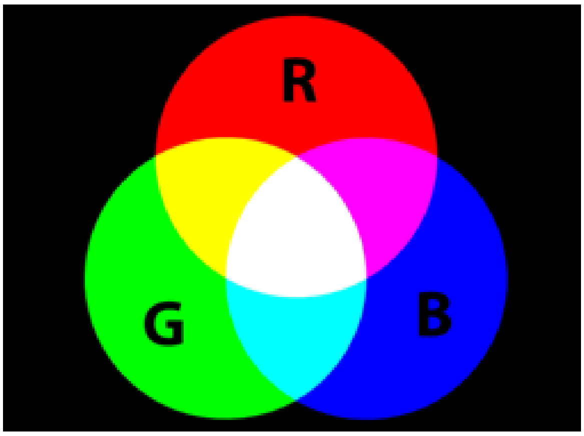


Figure I.3: representation of the RGB space.

is used in printing and hard copy output. One of the disadvantages of this model is the fact that it does not generate the color black exactly when this three color are mixed. To overcome this problem and closes this gap, the CMYK extension has been proposed where the K component encodes black [2].

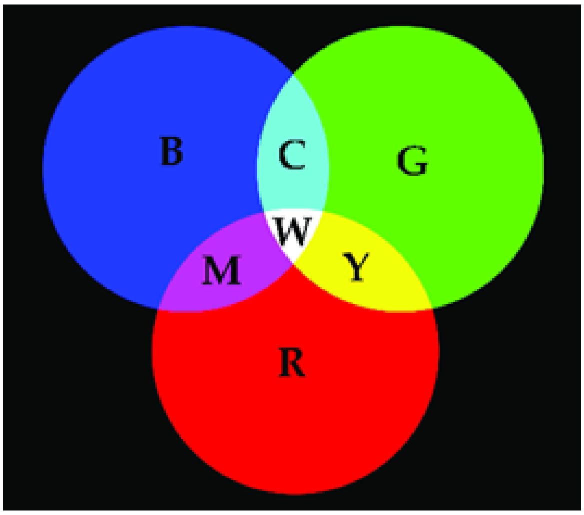


Figure I.4: representation of the CMY space derived from the RGB space.

I.4.3 HSL space

The HSL model (Hue, Saturation, Luminance), a model close to the physiological perception of color by the human eye. Although the RGB model is suitable for computer representation of color or display on output devices, it is difficult to select color through it. Indeed, the RGB color adjustment in computer tools is generally done using three sliders or three boxes with the relative values of each of the primary components, but the clarification of a color requires proportionally increase the respective values of each of the components. Thus the HSL model was developed to overcome this shortcoming of the RGB model. a space used in computer graphics applications. Figure I.5 shows an illustration of the HSV color space [3].

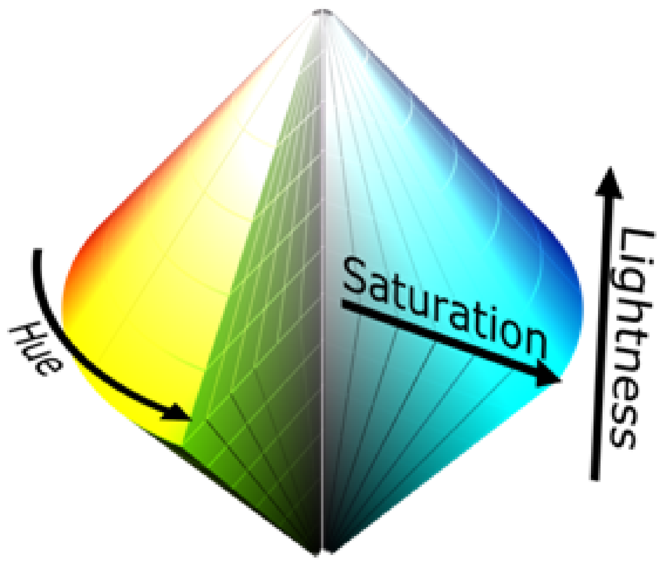


Figure I.5: representation of the HSL color space.

I.4.4 YCbCr space

Human vision is more sensitive to changes in the luminance than changes in chrominance this is why create the YCbCr space, which compatibility with the human vision system, it is exploiting its properties.

According to ITU (International Telecommunication Union) standards ITU-R BT.601-5 and ITU-R BT.709-5, YCbCr defined as a color space for digital television systems and also defines the transformation of coefficients between YCbCr and RGB color spaces. it mainly used in the digital video paradigm and image compression schemes. This space is composed of Y which is the luminance component, Cb component represents blue and Cr represent red color [2].

The conversion between RGB color space and YCbCr color space is described by the following equations [2]:

- Transformation RGB ⇒ YCbCr:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.16875 & -0.33126 & -0.50 \\ 0.5000 & -0.41869 & 0.08131 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Transformation YCbCr ⇒ RGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34413 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}$$

I.5 Concept of image

An image is a visual or even mental representation of something (object, living being and/or concept), it preserves the shape of the model that is drawn from it, and this image can be natural as a reflection, shadow; or artificial like a painting (drawing, synthetic image, etc.), or photography (video, radiography, etc.) [3].

I.6 Digital images

A digital image is a two-dimensional representation of three-dimensional objects [4]. The term "digital image" designates any image (drawing, icon, photograph, etc.) which [3]:

- Acquired by analog-digital converters, for example, scanners, cameras.
- Created directly by computer programs using a mouse, graphic tablets, or by 3D modeling (so called "synthesis images").
- Processed using computer tools to transform it, and modify the size, colors, add or remove elements, apply various filters, etc.
- Stored in floppy disk, hard disk, CD-ROM, etc.

I.7 Characteristics of a digital image

We can define An image as a structured set of information with the following characteristics:

I.7.1 Pixel

The term pixel is an abbreviation of (PICture Elements). The pixel is the smallest constituent element of a digital image, these are numerical values representative of light intensities. these values are in a two-dimensional matrix through which the image is processed [5][6] .

I.7.2 Dimension (size)

The dimension of a digital image represents the number of lines (denoted by N) of the matrix of pixels which represents the image multiplied by the number of columns (denoted by M) of this matrix. NxM (Figure I.6), This equation gives us the total number of pixels in an image [6].

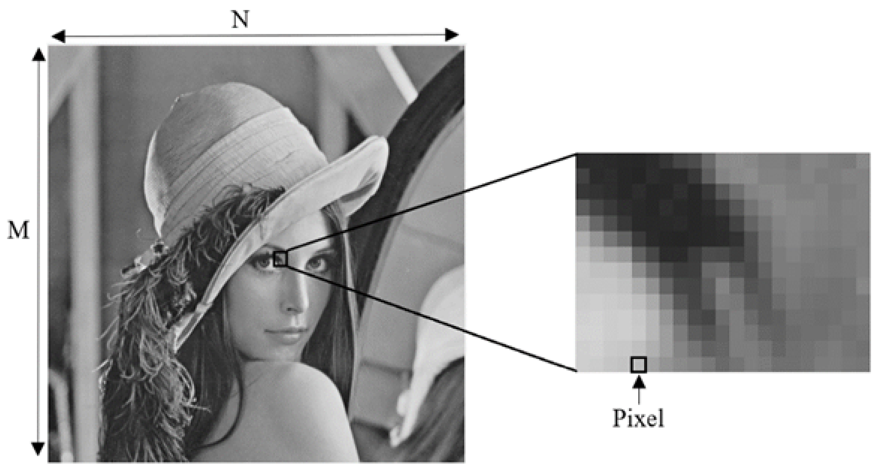


Figure I.6: Example showing each of the image's dimensions and pixels.

I.7.3 Resolution

The resolution of a digital image is defined by the number of pixels per unit area in the image. The higher the number of pixels in the unit area, the higher the resolution of the image. represent the resolution by dpi (dots per inch)or PPI (Pixels Per Inch). one inch representing 2.54 cm [5].

I.7.4 Histogram

A histogram is the distribution of pixel intensities in a digital image. with another form, is the number of pixels that represents each light intensity in the image. The horizontal axis of the graph represents variations in light intensity, while the vertical axis represents the number of pixels in the image. Figure I.7 represents an image with its histogram.

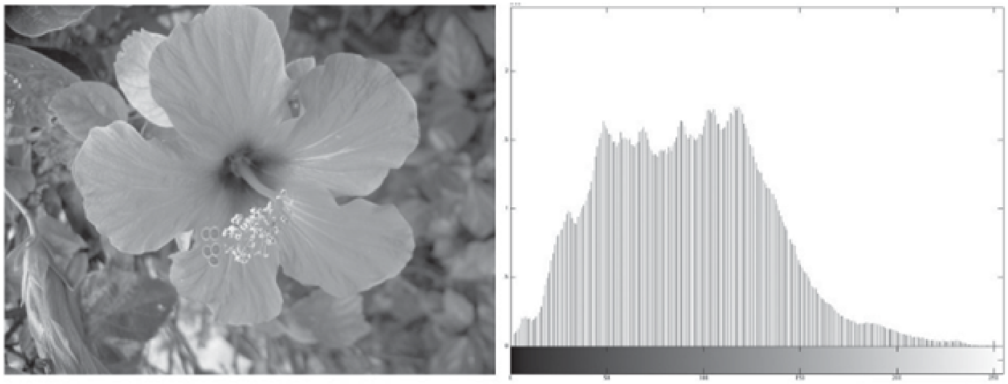


Figure I.7: Example showing a grayscale image with its histogram.

RGB images are represented by 4 histograms (Figure I.8) [6]:

- A histogram representing the luminance distribution.
- A histogram representing the distribution of the values of the red components.
- A histogram representing the distribution of the values of the green components.
- A histogram representing the distribution of the values of the blue components.

I.7.5 Luminance

Luminance is the degree of pixel brightness in the image. It is defined as being the quotient of the luminous intensity of a surface by the apparent area of this sur-

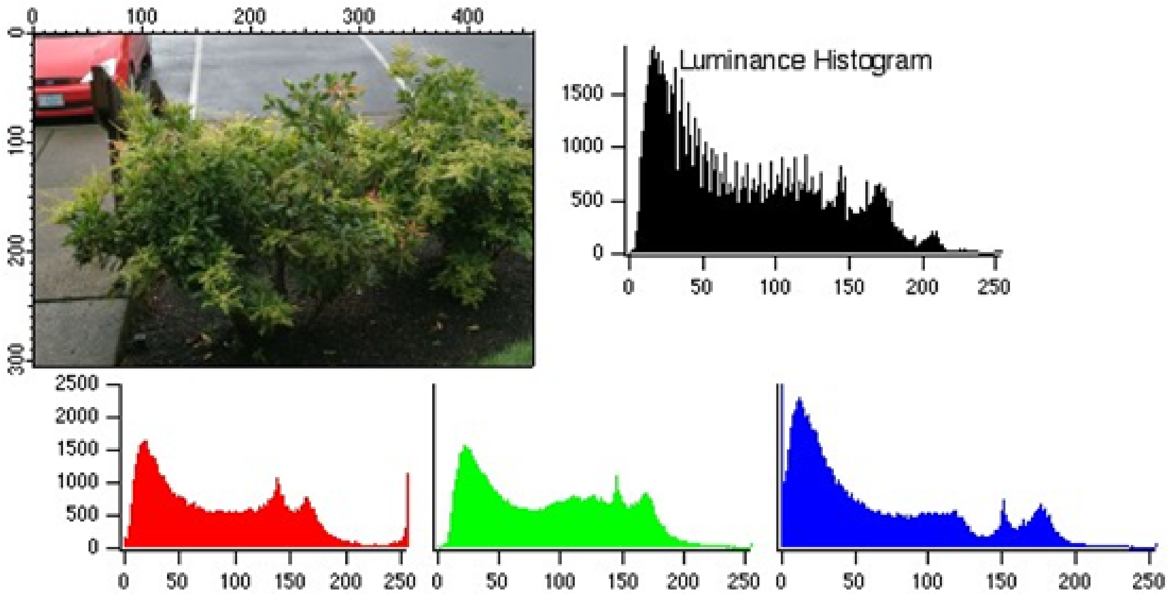


Figure I.8: Example showing a RGB image with its histogram.

face. A good luminance presents a bright image, A good contrast, and the absence of parasites [3].

I.7.6 Contrast

Contrast is the difference between dark regions and light regions in the image. the contrast C is defined by the ratio, and to calculate this last one this operation should be applied:

$$C = \frac{L1 - L2}{L1 + L2} \qquad (I.1)$$

With $L1$ and $L2$ are the degrees of the luminosity of two neighboring areas in the image [7].

A high-contrasted image presents a dynamic distribution of gray values over the entire range of possible values, with very clear whites and deep blacks. On the contrary, a low-contrast image has low dynamics (figure I.9) [8].

(a) (b)

Figure I.9: An example represents the difference between a low-contrast (a) and high-contrast image (b).

I.7.7 Depth

The Depth is the number of bits per pixel (bpp), this value reflects the number of colors or gray levels of an image, here are some values [2]:

- 8 bits/pixel = 256 colors
- 16 bits/pixel = 65,536 colors
- 24 bits/pixel = 16,7 million colors
- 32 bits/pixel = 1,07 billion colors

I.7.8 Contours and Textures

The contours represent the boundary between the pixels whose gray levels represent a significant difference.

The textures represent regions in the digital image that have homogeneous characteristics. The points which separate two textures in an image called "contours" [3].

I.7.9 Noise

In any digital image, the observed gray or color values have uncertainty. This uncertainty is due to the vagaries of counting the photons reaching each sensor. The measured color values are disturbed because the sensors pick up parasitic photons and are subject to electrostatic fluctuations during charging and discharging. When the sensor receives many photons from a well-lit scene, the noise is negligible compared to the actual photon flux. But even in an image with sufficient exposure, dark pixels receive very few photons and are therefore "noisy". in short, noise is the sudden difference in the intensity of a pixel compared to neighboring pixels in the image. generally, we can distinguish between two types of image noise that accumulates [3] :

- chrominance noise, which is the color component of noisy pixels: it is visible as random colored spots.
- luminance noise, which is the light component of noisy pixels: it is visible as darker or lighter spots giving the image a grainy appearance.

I.7.10 Image weight

The weight of the image determines according to the dimension and the depth, by counting the dimension $(N \times M)$ of the image multiplied by its depth [3].
example :
For a 640x480 image in true colors (True colors):
Number of pixels (dimension): 640 x 480 = 307200 pixels
Weight of each pixel (depth): 24 bits = 3 bytes
The weight of the image is equal to: 307200 x 3 = 921600 bytes.

I.8 Digital image types

Generally, there are three main types of digital images:

I.8.1 Binary image (black or white)

A binary image is a rectangular matrix associated with the value of the pixel. this pixel can only take the values 0 or 1(Figure I.10). The value 0 represents the color black and the value 1 represents the color white. therefore only one bit is used to code a level [3][4].

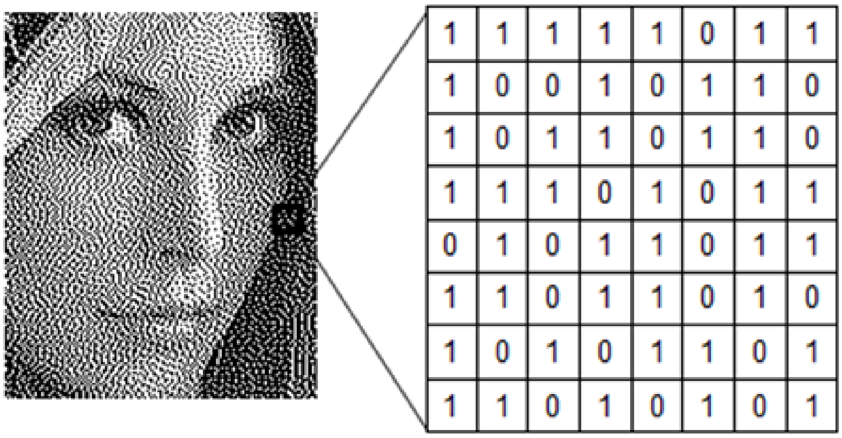


Figure I.10: a binary image.

I.8.2 Grayscale images

The gray level is the value of the light intensity at a point. if the intensity of the light is zero the value is zero which represents the color black, and if the intensity of the light is at the maximum value the value is 255 $(2^8 - 1 = 255)$ which represents the color white. the values between 0 and 255 represent the gradation of the two colors black and white, this color gradient can be seen in figure I.11. The value 256 is linked to the quantization of the image. to encode an image in grayscale it requires 8 bits (corresponding to 1 byte) [3] [4].

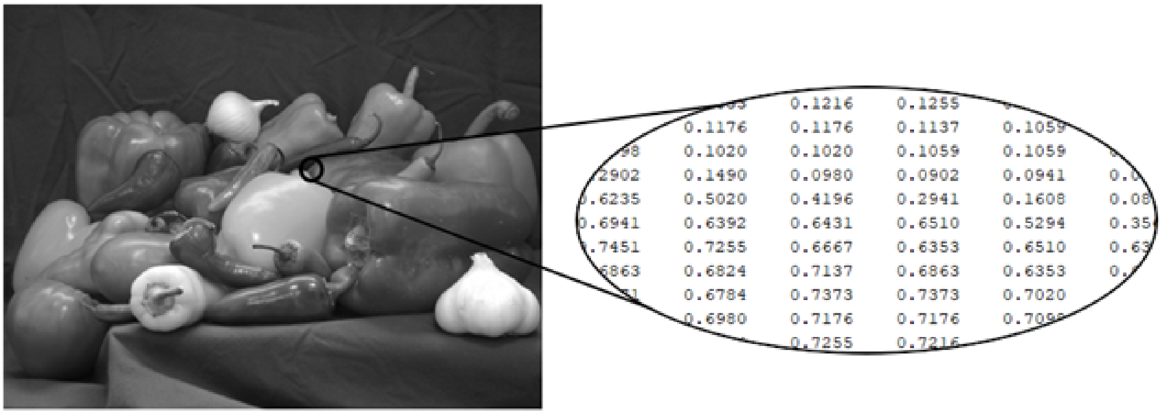


Figure I.11: a grayscale image

I.8.3 Color images

A color image is represented by three matrices red, green, and blue (RGB model) as showing in figure I.12 because the color space is the most used for the handling of digital images. Each color component is generally represented in 8 bits thus using 24 bits to represent each pixel of the image. The final color of each pixel is obtained by a mixture of these primary components which correspond to the dosage of the three basic colors (RGB) of each bit.

A color image can be coded in 4 bits, if the image is composed of 16 colors, 8 bits if the image is composed of 256 colors,and 24 bits if the image is composed of 16 million colors[3][4][3].

Figure I.12: a color image (RGB)

I.9 Vector and Bitmap image:

Generally, there are two main categories of digital images, bitmap images which are formed from a group of adjacent pixels, and victor images that represent by a set of geometric shapes. we will explain it in detail as follows:

I.9.1 Bitmap (Raster) Image

A bitmap or raster image is an image composed of little blocks of color called pixels organized in rows and columns (matrix). Each pixel is assigned a color code and a location, and this is how to form a picture. When we zoom in on a bitmap image, you can see the individual pixels (Figure I.13), so there's a loss of quality. This is commonly referred to as resolution-dependent since the quality or sharpness of the image depends on the resolution. we find this type in the domain of image processing and analysis [9]. Common bitmap image file types include:

- JPEG or JPG (Joint Photographic Experts Group): this format, the most widely used on the Internet, allows images to be displayed in 16 million color modes. it can compress an image to 50 KB which takes 1MB, so it gives the smallest file size, with the least loss of quality [10].

- Gif (Graphic Interchange Format): this format uses RGB encoding, the GIF format takes all 256 colors out of 16 million colors[11].

- BMP (Windows Bitmap): this is the format is the native image format in the Microsoft Windows operating systems. It compressed the images and gives it with a good quality[3].

- TIFF (Tag Image File Format): a format of excellent quality, but it presents compatibility problems due to the multiplicity of versions. There is also a compressed version which provides very compact files without noticeable loss of quality[3].

I.9.2 Vector image

The vector image is an image based on geometrical formulas (line segments, polygons, circles, rectangles, etc.), instead of pixels, to represent images. and This gives it the freedom to edit it resizing and changing colors without any losing resolution, making them ideal for icons, logos, and web-based imagery. When we zoom in on this type of image, we see a pure image without distortion (Figure I.13). This makes vector images resolution independent since the image quality is not affected by size or resolution settings.

To view this type of image on the computer screen, for example, you must convert to bitmaps first by converting the type of files. this type of image is used in the domain of graphics and computer-aided design [9]. Common vector image file types include:

- SVG (Scalable Vector Graphics): this format was developed to be used to compress or stretched images without losing the quality, and the element files are small [11].

- ESP (Postscript / Encapsulated Postscript): this is one of the best formats to export vector drawings, it encapsulates bitmaps and renders them in ESP format [12].

- PDF (portable document format): this format preserves the layout of a document and makes it small. also, this format is easy to deal with it, we can be viewed, printed, or electronically transmitted by uploading, downloading, or attaching it to a message or email [11].

- AI (Adobe Illustrator): is the proprietary Adobe file type for vector images. Other common vector image file types include PDF, EPS, and SVG. vector files don't lose resolution when compressed because they're built on a graph-like formula that is infinitely expandable [12].

I.10 Conclusion

In this chapter, we touched on the concept of light through which a person can perceive colors, and this is what allows the human eye to see things and differentiate between them. We also explained the concept of a digital image of all kinds that

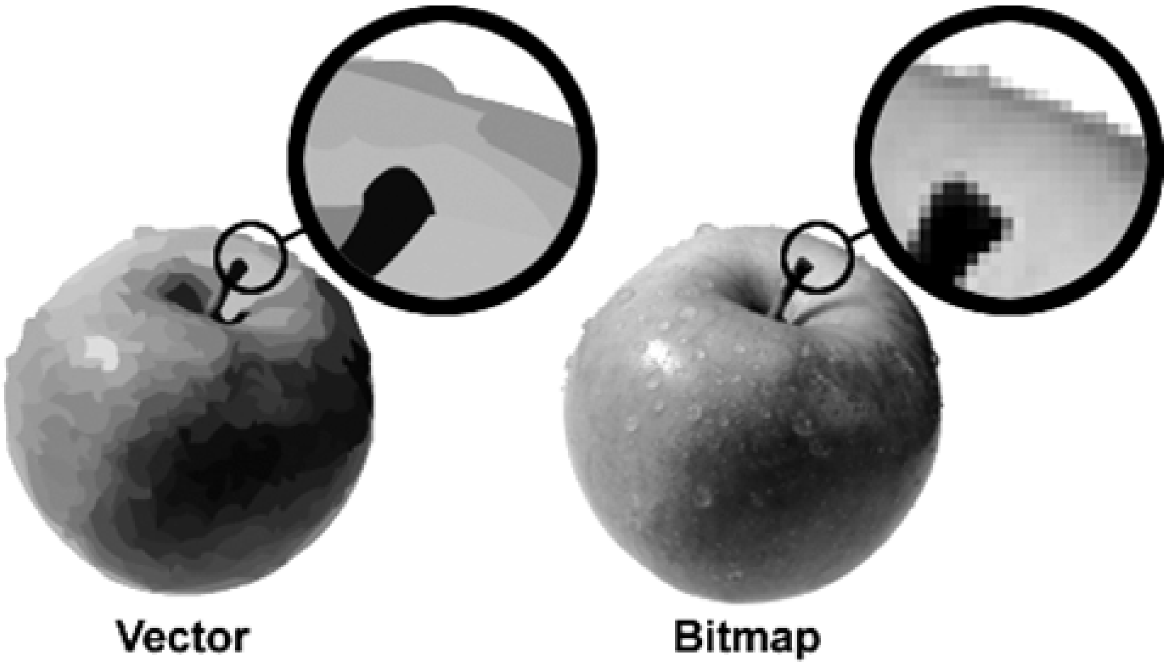


Figure I.13: An example of a bitmap and victor image

are formed through a group of adjacent pixels, as each pixel takes a specific digital value, and each of these values represents a specific color. Computers save images by compressing them into files such as GIF, JPEG, PDF, etc. to be easy to send and store, and to save space on these devices. In the next chapter, we will delve into the idea of compression.

Chapter II

Compression techniques

II.1 Introduction

the image Compression is a data compression application that encodes the original image with a few bits. The purpose of the image compression is to minimize the amount of storage quantity (redundancy in the image) and to transfer the data in an efficient form. the decoded image displayed on the monitor can be as close to the original image as possible. Image compression is done using coding algorithms based on mathematical transformations.

In general, there are two grand type of methods compression: lossless and lossy compression, in this chapter we will dive deep in this two methods and distinct between them.

II.2 Compression

Image compression means minimizing the size of an image file in bytes without degrading the image quality to an unacceptable level. Reducing the file size allows you to save more images on a given amount of disk, more memory space. It also reduces the time it takes to send the image over the Internet or download it from websites. In other words, compression is about reducing the physical size of blocks of information. The compressor uses an algorithm that is used to optimize data by applying appropriate considerations for the type of data to be compressed; The de-compressor is therefore necessary to reconstruct the original data using an algorithm that is opposite to the algorithm used for compression. i.e the complementary operation is decoding which recovers the image from the compressed representation [13].

In general, compression works to remove redundancy in an image. There are two main compression methods: lossless and lossy.

II.2.1 Lossless compression

Lossless compression or process provides an image that is identical to the original image after decompression. this type of compression is often used in medicine because of the ability to restore the original images in detail. Lossless compression takes place in two stages: decorrelation and entropy coding. The decorrelation step removes spatial redundancy by using techniques like run-length coding and transforms techniques. as for entropy is a measure of information. It removes coding redundancy by using some techniques.

We can use Huffman coding, arithmetic decomposition, and run-length encoding (RLE) to compress image without losing the information [14].

II.2.1.1 Huffman Coding

Huffman Coding is one of the most commonly implemented algorithmic techniques. it used in image formats like JPEG format. The idea of this coding is to assign variable length codes of the input, and create a table incorporating these codes. This table is known as frequency table. Then a binary tree is generated. in short, the Input is a set of variable along with their frequency of occurrences and output is Huffman Tree. a tree is created through the following stages [14]:

1. Search for two nodes having the lowest frequency which are not yet assigned to a parent node.

2. Couple these two nodes together to a new interior node.

3. Add both frequencies and assign this value to the two interior nodes.

4. Repeat until all nodes are combined together, and you get one node it is called root node, and the tree is completed.

This is what will be explained in details in the following example:
we have an input represented as:
input=0 10 110 1111 0 10 110 1111 0 10 11100 11101
This input is represented in the table II.1 with its frequency and symbol:

Symbol	Frequency	Code
c	3	0
h	3	10
i	2	110
n	2	1111
u	1	11100
a	1	11101

Table II.1: The input and what represents from a symbol and a frequency.

u and a are chosen as a per for having the smallest frequency. Here begins the process of linking nodes as shown in figure II.1

II.2.1.2 Arithmetic Coding

Arithmetic coding is another entropy coding technique which employs the idea of mapping a symbol sequence into a real number between zero and one based on the probability distribution of the source, and the real number is converted into a binary sequence as the codeword. It is proved that if the length of the binary sequence equals to $[-log_2 p(x)]$, the receiver is able to rebuild the message exactly without any side information, p(x) is the probability of the symbol sequence.

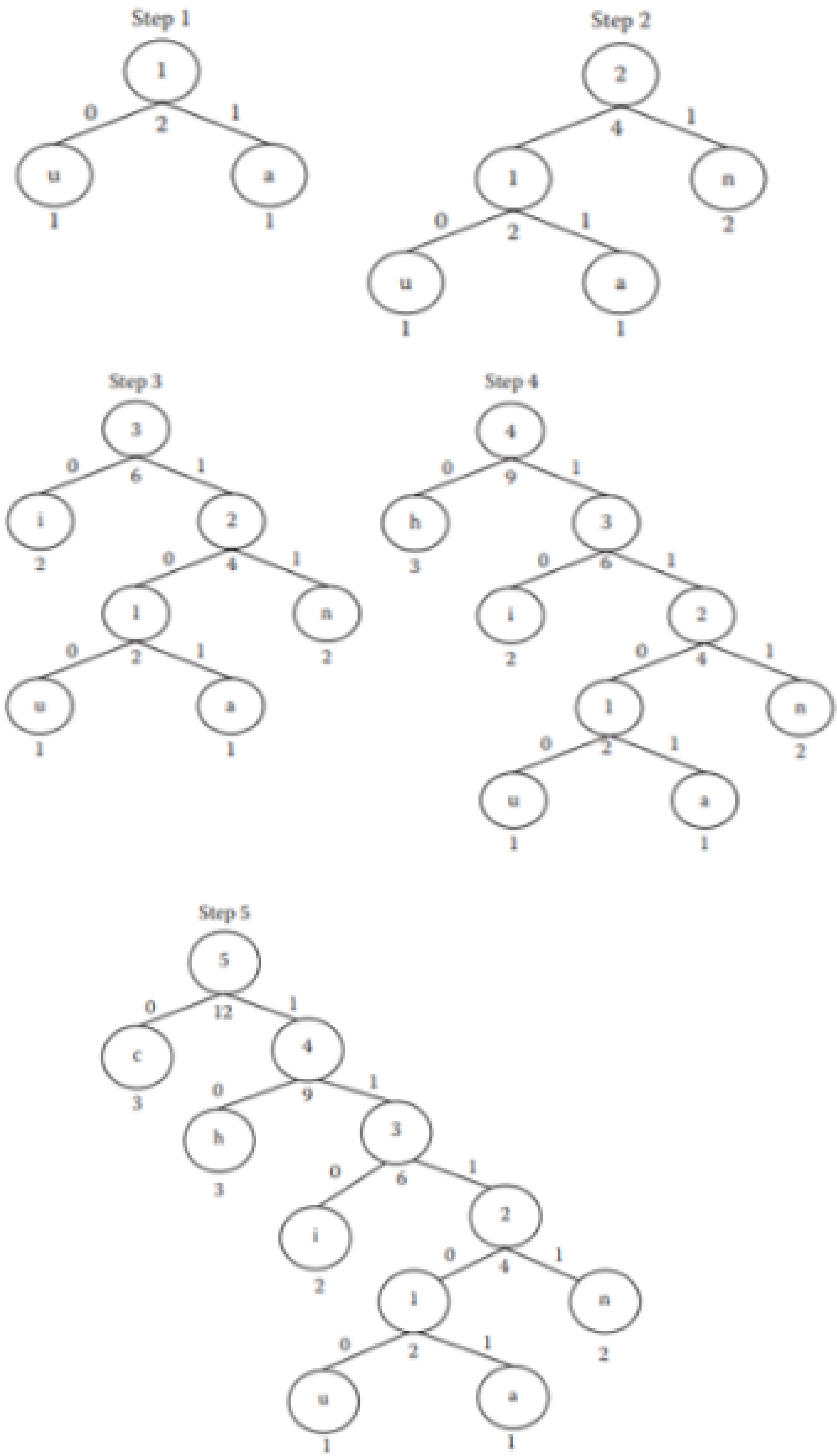


Figure II.1: The different steps of Huffman tree .

The process Arithmetic Coding is in two steps [15]:

First step: the partitioning stage, the interval [0, 1] is divided into several sub-intervals (noted by q) based on the probability distribution, and the symbols are assigned to intervals uniquely, defined as:

$$l_i = \sum_{k=1}^{i-1} p(a_k) \qquad (II.1)$$

$$h_i = \sum_{k=1}^{i} p(a_k) \qquad (II.2)$$

Second step: the encoding stage, the input data is encoded symbol by symbol (recursively) in the model and the final interval is converted into a certain length binary sequence. In detail, a finite length N sequence of X characterized by p(x) is encoded in Arithmetic coding. $(x_1, x_2...x_N)$ are encoded symbol by symbol in the algorithm described below:

1. Initialize the interval I_0 with $I_0 = [0, 1]$.
2. Choose the interval that is assigned to x_1, which is I_1 characterized by $l(x_1)$ and $h(x_1)$.
3. Divide the interval I1 into q sub-intervals with the same structure as that in the model, and choose the interval that is assigned to x2 which results in an interval I_2 characterized by $l(x_1, x_2)$ and $h(x_1, x_2)$ as :

$$l(x_1, x_2) = l(x_1) + l(x_2)p(x_1) \qquad (II.3)$$

$$h(x_1, x_2) = l(x_1) + h(x_2)p(x_1) \qquad (II.4)$$

4. Repeat step 3 until all the symbols are encoded. The final interval I_N is characterized by $l(x_1, x_2...x_N)$ and $h(x_1, x_2...x_N)$ which can be calculated in a recursive function :

$$l(x_1, x_2, ..., x_N) = l(x_1, x_2, ..., x_{N-1}) + l(x_N)p(x_1, x_2, ..., x_{N-1}) \qquad (II.5)$$

$$h(x_1, x_2, ..., x_N) = l(x_1, x_2, ..., x_{N-1}) + h(x_N)p(x_1, x_2, ..., x_{N-1}) \qquad (II.6)$$

5. Convert the final interval I_N into a length L binary sequence, L is defined as:

$$L = -log_2 p(x_1, x_2, ..., x_N) \qquad (II.7)$$

The length L binary sequence is transmitted to the receiver as the codeword. It is proved that the input sequence can be rebuild exactly by the receiver in this set up.
Example

An example illustrating the arithmetic coding process is shown in Figure II.2. The discrete random variable X takes the values from the alphabet $A = \{0, 1, 2, 3\}$ with the distribution listed in Table II.2.

X	0	1	2	3
P	0.6	0.2	0.1	0.1

Table II.2: the input values sequence X.

The process starts with the same interval used by the encoder: [0,1] and using the same model, dividing it into the same four sub-intervals that the encoder must have. The result after the encoding process is [0.534,0.54] as shown in Figure II.2. is converted to length $L = -[log_2 p_0 p_2 p_3] = 8bits$ of the binary string. The code word is 10001001.

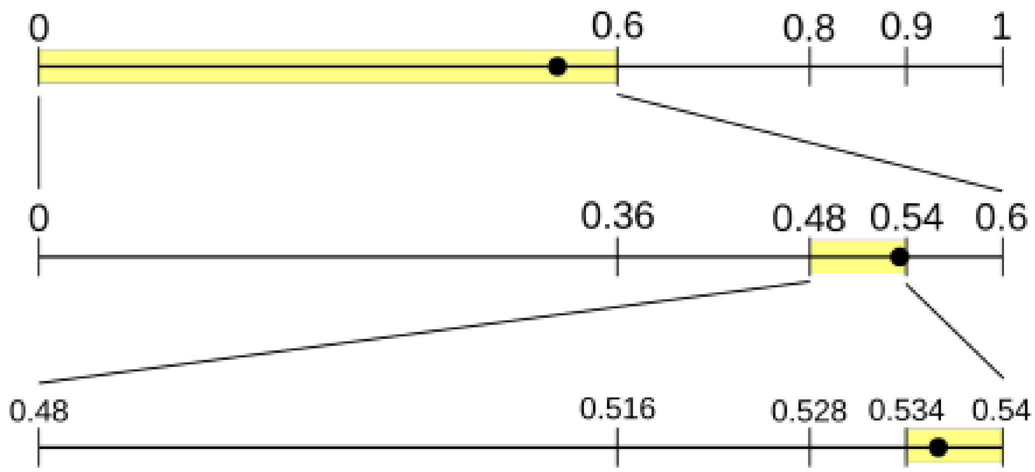


Figure II.2: Example of Arithmetic coding.

II.2.1.3 run-length encoding (RLE)

Among the most popular and simpler methods of coding of data compression can be mentioned Run Length Encoding which is often coded with RLE. it is adopted by most image formats such as TIFF and BMP format. Its idea is simple, as it replaces a long input with a shorter one, by reducing the interaction and attaching them to the number of their repetitions [16]. for example, an input consists of 10 numbers arranged as follows:
input = 128 128 128 0 0 127 127 127 127 127
With using the RLE algorithm the output is the number and is followed by the number of its frequency. So the output of the example be as the following form:
output= 128 3 0 2 127 5

II.2.2 Lossy Compression

Lossy Compression presents an image similar to the original but with some information lost that cannot be retrieved again. where it provides greater data compression. The basis for compression uses other characteristics of images such as filtering and look up tables (LUP), sub-sampling, bit allocation, and quantization such as scalar and vector, which are called importance-oriented techniques [14].

II.2.2.1 Quantization

In quantization, the source random variable taking values in an infinite set or a finite set of high cardinality is represented by the output akin values from a predetermined set of low cardinality. It is used to compress data for storage and transmission in digital form in communication systems. The mapping that performs the quantization process is called a quantizer. The set of source values is usually called the source alphabet, and the set of output values is selected from a set called the reproduction alphabet. There are two types of quantization: scalar quantization (SQ) and vector quantization (VQ) [17].

1. Scalar Quantization(SQ)

   Scalar quantization is defined as the association of each real value x with another value of q belonging to a finite set of values $\{q_1; q_2; ...; q_L\}$. At any value of x in the interval $[x_n; x_{n+1}]$, we match the quantized value of q_i in this range. This q value can be expressed according to the truncation used: either rounded up, rounded down, or rounded closer. Scalar quantization is performed independently for each value by [18][1]:

   $$SQ : \mathbb{R} \rightarrow \{q_1, q_2, ..., q_L\} \qquad (II.8)$$

   $$x \rightarrow q_i$$

   $$SQ(x) = \frac{x + 0.5}{q}$$

   (a) Uniform scalar quantization

   Uniform scalar quantization (USQ) is the earliest, simplest, and most common form of SQ, by which an input sample xi is mapped to a quantized value $\hat{x_i}$ by [19][20]:

   $$\hat{x_i} = [\frac{x_i}{Q_{step}}]Q_{step} \qquad (II.9)$$

   where Q_{step} is the quantization step size and [.] is the rounding operator. Generally speaking, USQ represented in a diagram as shown in figure II.3.

   (b) non-uniform scalar quantization

   A non-uniform scalar quantization can be seen as a uniform scalar quantization preceded by a non-linear transformation and followed by the inverse transformation. and we can say that it is a scalar quantization with steps that are not constant, as figure II.4 represent [21].

2. Vector Quantization (VQ)

   Vector Quantization is considered better than scalar Quantization because it makes full use of the interdependence of the components inside the

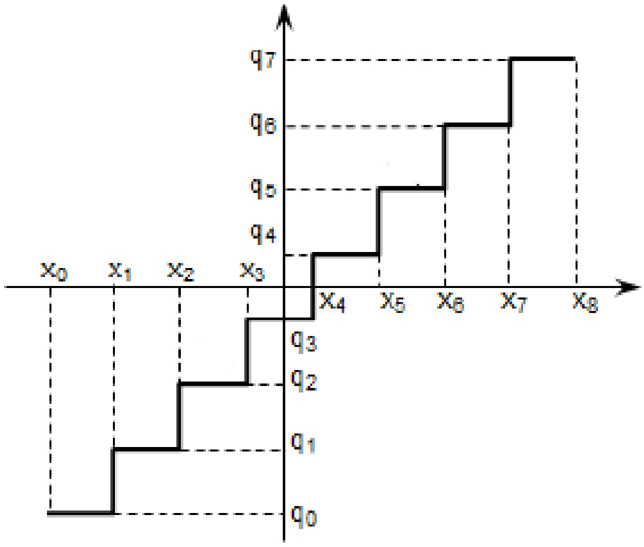


Figure II.3: graphic curve represent the Uniform scalar quantization.

vector, and it also achieves a higher-pressure ratio, and the closer the dimensions of the vector are to infinity, the more ideal the result will be. Decoding in VQ is very simple. But despite its perfection, it may become more complex as the dimensions of the vector increase, so it must be taken into account while designing the cipher and making it practical, and another problem of image VQ is the need of a codebook which causes several problems in practical application, such as generating a universal codebook for a large number of images, scaling the codebook to fit the bit rate requirement [18]. figure II.5 shown us a Block diagram of the steps of VQ for data compression.

a vector quantizer of dimension N and size L the map of $\mathbb{R}^{\mathbb{N}}$ in a finite set C containing L vectors of dimension N defined by [1]:

$$VQ : \mathbb{R}^{\mathbb{N}} \rightarrow C \; with \; C = \{\vec{C_1}, ..., \vec{C_L}\} \qquad (II.10)$$

$$\vec{X} \rightarrow \vec{C_i} = VQ(\vec{X})$$

II.2.2.2 Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) is the most commonly used transform for image and video coding. The DCT transforms the set of input values into a set of

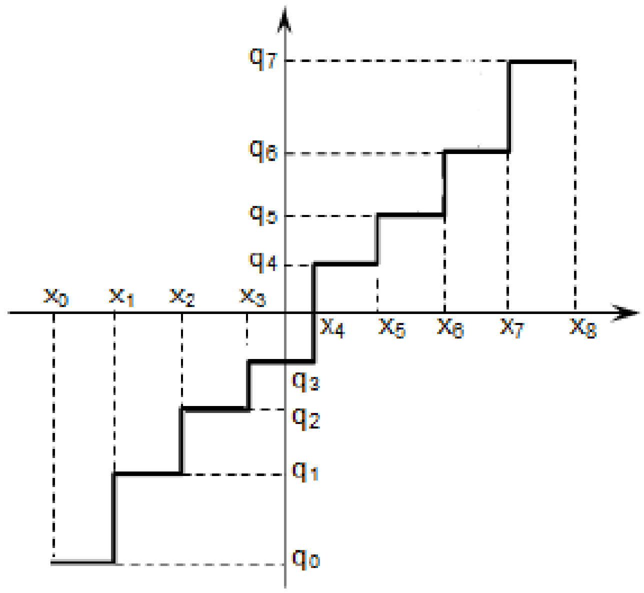


Figure II.4: graphic curve represent the non-uniform scalar quantization.

coefficients to cosine functions with increasing frequencies. The input is a set of numeric values and the output is a set of the same size.

The DCT is an invertible transform, which means that its output coefficients can be used to recreate the original input values. The reverse of the DCT is called the Inverse Discrete Cosine Transform it symbolized by IDCT. The DCT is commonly used to process data organized in either one or two dimensions [10].
The one-dimensional DCT of an array V of N numbers into an array T of N numbers is defined as:

$$T[i] = C(i)\sum_{n=0}^{N-1} V[n]\cos\frac{(2n+1)i\pi}{2N} \qquad (II.11)$$

where

$$C(0) = \sqrt{\frac{1}{N}}$$

$$C(k) = \sqrt{\frac{2}{N}}$$

$$k \neq 0$$

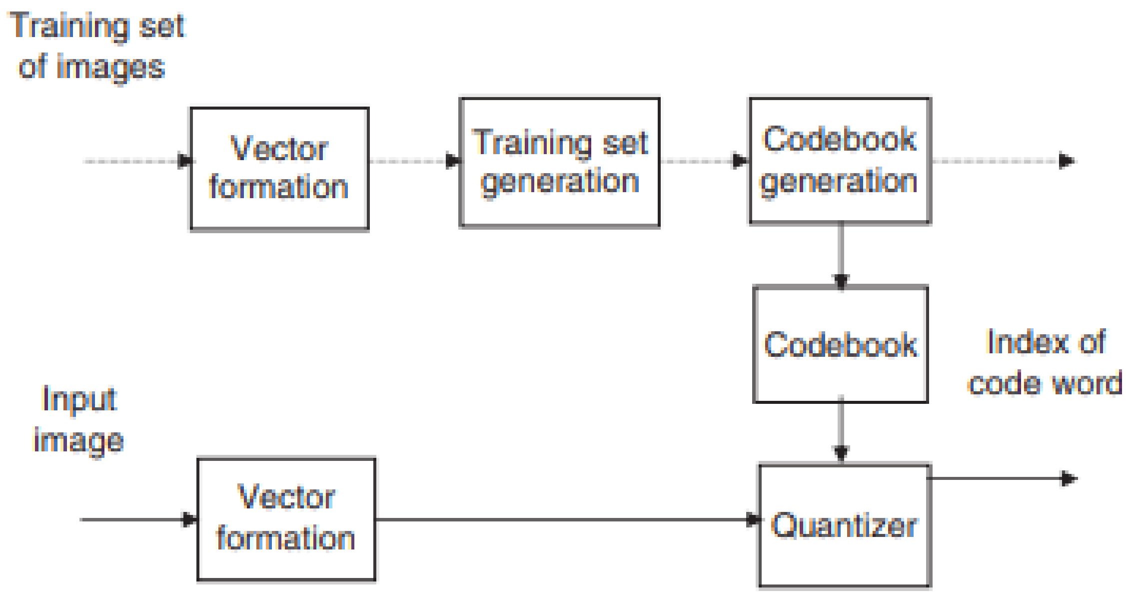


Figure II.5: A block diagram of the various steps involved in VQ.

The one-dimensional IDCT is defined as:

$$V[n] = \sum_{n=0}^{N-1} C(n)T[i] \cos \frac{(2n+1)n\pi}{2N} \qquad (II.12)$$

In this chapter we will deal with square matrix, i.e, the number of rows is equals the number of columns $(N = M)$. two-dimensional DCT is performed on square matrix of size N, it is defined as:

$$T[i,j] = C(i,j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} V[y,x] \cos \frac{(2y+1)i\pi}{2N} \cos \frac{(2x+1)j\pi}{2N} \qquad (II.13)$$

where

$$C(i,j) = \frac{1}{N} \text{ , } (i \text{ } or \text{ } j) = 0$$

$$C(i,j) = \frac{2}{N} \text{ , } (i \text{ } and \text{ } j) \neq 0$$

The two-dimensional IDCT is defined as:

$$V[y,x] = C(i,j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(i,j) T[i,j] \cos \frac{(2y+1)i\pi}{2N} \cos \frac{(2x+1)j\pi}{2N} \qquad (II.14)$$

II.2.2.3 Discrete Wavelet Transform (DWT)

DWT is an essential stage in the compression chain in many image compression algorithms. the process of this Transformation is done to compress the images, starting from the ranks of the image, that is, the DWT process is done horizontally, this is what divides the image into two sub-bands, one low frequency (L) and the other high frequency (h). The method is repeated again on the horizontal dwt output, the result is four sub-bands (LL, LH, HL, HH) (figure II.6). That process is repeated for the required number N on the LL sub-band depending on the image size [22].

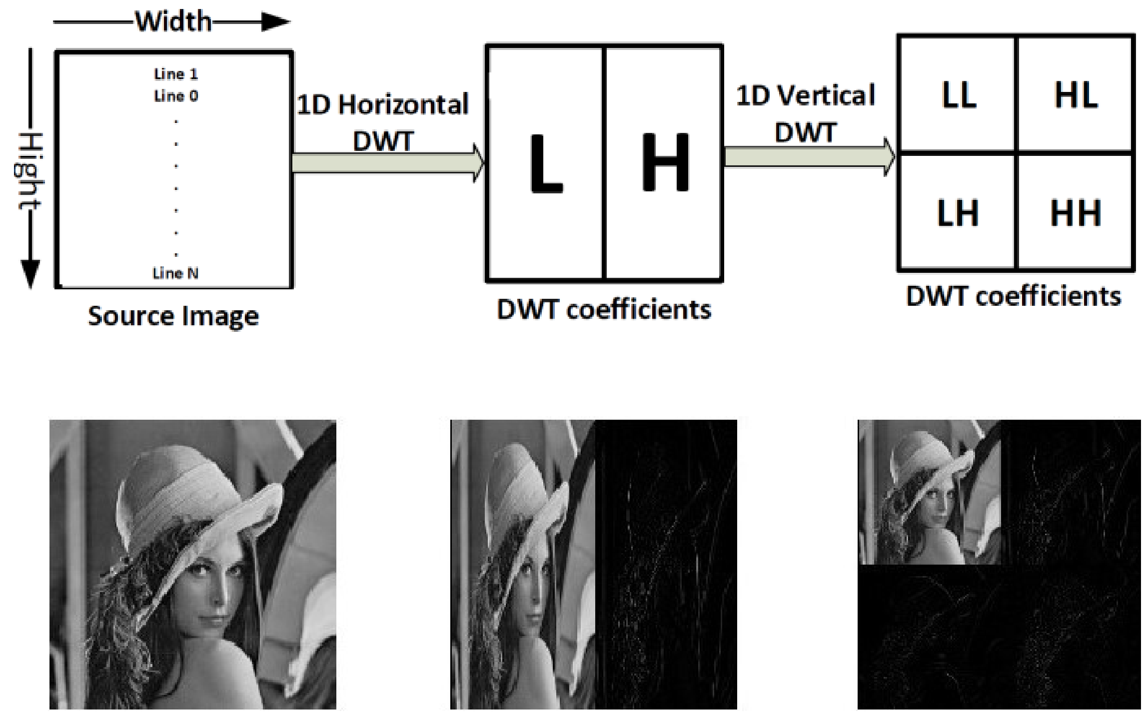


Figure II.6: an illustration of Tiling, DC level shifting and DWT in jpeg2000 standard.

The wavelet function is generated by dilations and translations of a function ψ which is defined as following [23]:

$$\psi_{a,b(x)} = |a|^{\frac{-1}{2}} \ \psi \ (\frac{x-b}{a}) \qquad (II.15)$$

Where (a, b) $\in \mathbb{R}$ and a $\neq 0$. High frequency wavelets correspond to a < 1 or narrow width, while low frequency wavelets have a > 1 or wider width. For a wavelet of orthogonal bases L_2:

$$\psi_{m,n(x)} = 2^{\frac{-m}{2}} \ \psi(2^{-m}x - n) \quad where \quad m, n \in \mathbb{Z}^{\not\models} \qquad (II.16)$$

The wavelet coefficients are given by:

$$C_{m,n(f)} = \sum f(x)\bar{\psi}_{m,n(x)}$$

One can represent any arbitrary function f as a superposition of wavelets as:

$$f(x) = C_{m,n(f)}\psi_{m,n(x)} \qquad (II.17)$$

II.2.2.4 Continuous Wavelet Transforms (CWT)

The continuous wavelet transform uses inner products to measure the similarity between a signal and an analyzing function. in this method the translation and dilation parameters vary continuously. in other words, the transform uses the functions:

$$\psi_{a,b(x)} = \frac{1}{\sqrt{|a|}} \ \psi \ (\frac{x-b}{a} \qquad (II.18))$$

With (a,b) $\in \mathbb{R}$ and a $\neq$ 0,where a is used to compress or extend the function f_i, and b is used to translate it. When we analyze a signal f(x) with these wavelets, we transform it into a function of two variables the time and the scale of analysis of the signal, which we can also note [24]:

$$W(a,b) = \frac{1}{\sqrt{|a|}} \int f(x)\psi_{a,b}(x)dx \qquad (II.19)$$

II.3 Conclusion

The large volume of image data makes it difficult to deal with (sending, downloading, etc.), for this reason researchers have resorted to invent ways to compress images in order to facilitate dealing with them. There are two types of compression: lossless compression, which we find it more in the field of medicine and military This is due to the importance of the image information and any deficiency may occur as a problem, because this compression does not support the loss of information or it may be a slight loss. As for the second type, it is lossy compression, which loses information with the inability to retrieve it, as it reduces the unimportant information, which by losing it does not distort the image to be compressed.

Chapter III

JPEG and JPEG2000 standards

III.1 Introduction

Some standards that we are very familiar with are used to measure size, weight, and distance. In this chapter we will discuss another type of standards, the standards of images. There are two popular standards: JPEG and JPEG2000.

JPEG is a standard for creating, storing, and using digital photo files. years later, the JPEG2000 standard was intended as a replacement for the JPEG digital image file format. Maybe they share the same name, but they are two different image file types. In this chapter, we explore these differences.

III.2 JPEG standard

In 1986, the Joint Photographic Experts Group (JPEG), officially known as ISO/IEC JTC 1/SC 29/WG 10, was formed with the purpose to establish a standard for the sequential progressive encoding of continuous tone grayscale and colour images. Its main requirements was the compression ratio should be good while maintaining the image quality, and the encoder should be parameterizable, also it can be applied to all types of digital image. JPEG standard includes four modes of operation it is [25]:

- Sequential DCT-based encoding (the most used): each image component is encoded in a single left-to-right, top-to-bottom scan.

- Progressive DCT-based encoding: the image is encoded in multiple scans for applications in which transmission time is long, and the viewer prefers to watch the image build up in multiple coarse-to-clear passes.
- Lossless encoding: the image is encoded to guarantee exact recovery of every source image sample value (even though the result is low compression compared to the lossy modes).
- Hierarchical encoding: the image is encoded at multiple resolutions, so that lower-resolution versions may be accessed without first having to decompress the image at its full resolution.

III.2.1 The Stages of the JPEG Standard

The lossy JPEG process is done according to the division of the grayscale image to be compressed into blocks of size 8x8 (block of 64 pixels). Color image compression it regarded as compression of multiple grayscale images, which are either compressed entirely one at a time, or are compressed by alternately interleaving 8x8 sample blocks from each in turn.

Then, each block is transformed by DCT or FDCT into 64 basis-signal referred to as DCT coefficients, and that is by taking FDCT to the input signal as its input and decomposes it into 64 orthogonal basis signals.

After output from the FDCT, each DCT coefficients is uniformly quantized, combined with a 64-elements quantization table, which must be specified by the application or user as input to the encoder. The purpose of this processing step is to discard visually unimportant information. Quantization is the main source of loss in DCT based encoders[26].

After quantization, the DC coefficient (the (0,0) coefficient of the transform) and the 63 AC coefficients are separately encoded. The DC coefficients are encoded in $DPCM$, using the prediction of the coefficient from the previous block:

$$DIFF = DC_i - DC_{i-1} \qquad (III.1)$$

The main energy of the image is concentrated in the DC coefficients. The DC coefficients are encoded separately from the AC coefficients to further exploit this fact. The 63 AC coefficients are encoded in RLE following a "zigzag" path starting from the $AC(0,1)$ coefficient [27]. In Figure III.1 (a) represents the Differential DC encoding and (b) Zig-zag sequence.

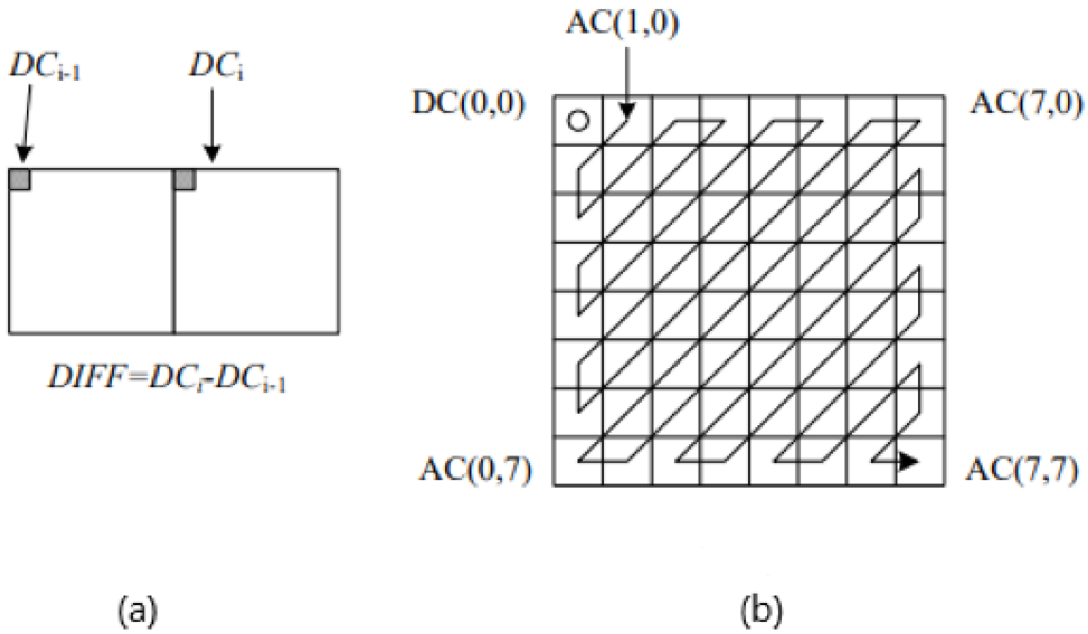


Figure III.1: Preparation of DCT coefficients for entropy coding.

Finally, the entropy coding. This step enables additional lossless compression by encoding the quantized DCT coefficients more compactly based on their statistical properties. The JPEG standard addresses two types of entropy coding: Huffman coding and arithmetic coding. It is useful to think of entropy encoding as a two-step process. The first step converts the sawtooth sequence of quantized coefficients into a sequence of intermediate symbols. The second step converts the symbols into a data stream, where the symbols no longer have externally identifiable boundaries [26]. Figure III.2 shows the processing steps of the DCT-based modes of operation, and it can summarize all of the above about the stages of the JPEG standard.

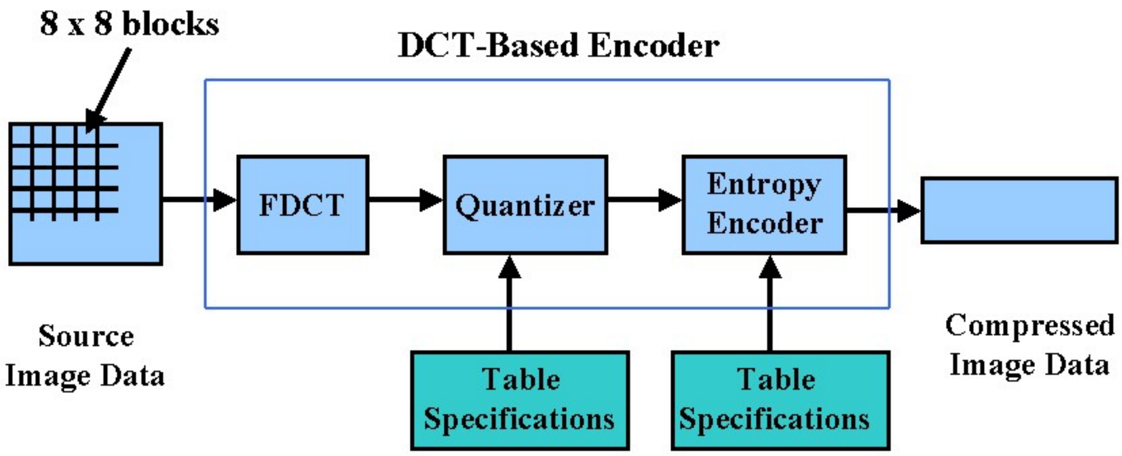


Figure III.2: shows a simplified of JPEG encoding of lossy method with DCT.

like any standard in which it is found weaknesses and strengths, it is founded that the JPEG standard, despite its wide spread and strength, has weaknesses represented in [28]:

- it is restricted to 8 bits/pixel inputs
- it provides a low-quality image at low bit rates.
- the maximal image size is 64k x 64k in the JPEG standard
- it suffers of a block affect as the DCT is applied on 8x8 block.

III.3 JPEG2000 standard

The JPEG2000 standard is a new international standard for encoding and decoding static pictures. it has been designed to be a JPEG successor. The JPEG2000 uses the DWT to transform the input image. the DWT eliminates the blocking artifacts, it can provide an excellent coding efficiency and wonderful spatial and quality scalable functionality. but it requires a long computation time. also JPEG2000 depends on the arithmetic coding. The main requirements for the JPEG2000 are [28][18][29]:

- Superior low bit-rate performance: This standard should provide better performance than the current standard at low bit-rates (ie, below 0.25 bpp (bits per pixel) for high detail grayscale images). This shall not be achieved with sacrificing performance on the rest of the rate-distortion spectrum. This feature is needed by applications like network image transmission and remote sensing.

- Lossless and Lossy Compression: Some applications require lossless compression. so, it is providing the ability to generate embedded bitstreams and allow for progressive lossy to lossless construction.

- Multiple resolution representation: This feature allows the user to sequence the stream to decode images as the resolution increases. This means that decoding of a small overall picture starts, and as long as decoding is running, the resolution increases to the original picture resolution.

- Tiling: it split the large image into smaller pieces that are encoded independently.

- Region of Interest Coding: Often some areas of an image are more interesting than others, so this standard is designed to allow users to define certain Region Of Interests (ROI) in an image that will be displayed with better quality and less distortion To encode and transmit it, this is the rest of the picture.

- Error resilience: Robustness to bit errors needs to be considered when designing the code-stream. One application where this feature is important is transmission over wireless communication channels.

- Random code-stream access and processing: This feature allows random access/decompression of custom ROIs in the image with less distortion than the rest of the image. In addition, random code stream processing can realize operations such as rotation, translation, filtering, feature extraction and scaling.

- A more flexible file forma.

- Improved performance to multiple compression/decompression cycles: This feature means that the new standard has better multiple compression/decompression loop performance than the old standard in the case of lossy transformations.

- Large and complex image support: the maximal size of a picture is $(2^{32}-1)\times(2^{32}-1)$ with a maximal bit-depth of 38 bits and the maximal number of components is 16384.

III.3.1 The Stages of the JPEG2000 Standard

The main steps of a JPEG2000 encoding process are described in figure III.4. In the first step, it is applied according to the color of the image. This step simplifies some implementation issues such as numerical overflows but has no effect on the coding efficiency. There are two transformations are available in the baseline JPEG2000 from the RGB to YCbCr color space [29]:

- irreversible color transform (ICT) that is represented by:

$$Y = 0.299 \times (R-G) + G + 0.114 \times (B-G) \qquad (III.2)$$

$$Cb = 0.546 \times (B-Y) \qquad (III.3)$$

$$Cr = 0.713 \times (R-Y) \qquad (III.4)$$

- reversible color transform (RCT) that is represented by:

$$Y = [\frac{R+2G+B}{4}] \qquad (III.5)$$

$$U = R-G \qquad (III.6)$$

$$V = B-G \qquad (III.7)$$

Then, it partition of the source image into rectangular non-overlapping blocks (tiles) (see Figure III.3); this process is referred to as a term "tiling". This is used to reduce the memory needs during the encoding.

Next, the tile components are decomposed, by inverse DC level shifting is performed on reconstructed samples of components that are unsigned only. then by different decomposition levels using a Discrete wavelet transform. These decomposition levels contain a number of subbands, which consist of coefficients that describe the horizontal and vertical spatial frequency characteristics of the original tile component (see Figure III.3) [30].

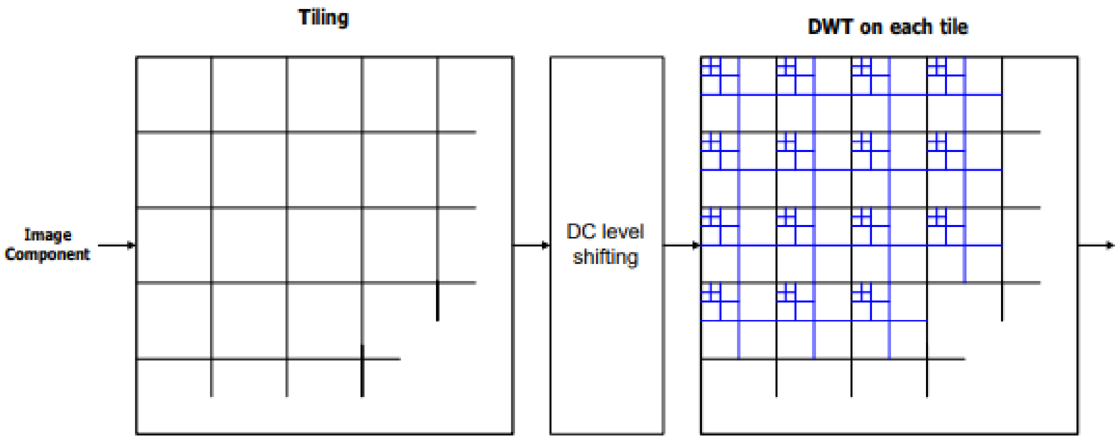


Figure III.3: shows a simplified of JPEG encoding of lossy method with DCT.

After transformation, all coefficients are quantized. this process by which the coefficients are reduced in precision, and it is lossy operation. Every subband at each level of decomposition shall have their quantization step, and this according to the formula:

$$q_b = sign(a_b(u,v))\ [\frac{|a_b(u,v)|}{\Delta_b}] \qquad (III.8)$$

The quantization step b is represented relative to the dynamic range Rb (Rb depends on the number of bits used to represent the original image tile component) of subband b, as:

$$\Delta_b = 2^{R_b - \varepsilon_b}\ (1 + \frac{\mu_b}{2^{11}}) \qquad (III.9)$$

After the quantization, we find the entropy coder. The indices of the quantized coefficients in each subband are entropy coded to create a compressed bitstream. For JPEG2000 is proposes the coding "embedded block coding with optimized truncation" or EBCOT. In EBCOT, each sub-band of an image tile is repartitioned into rectangular blocks which are called “code blocks”. These blocks of code are individually entropy coded [27].

Finally, component transformations. In JPEG2000, the bit-stream is organized as a succession of layers. Each layer contains the additional contributions from each code-block. For each code-block, a separate bit-stream is generated. No information from other blocks is utilized during the generation of the bitstream for a particular block. Truncation points to each code block are allocated using rate distortion optimization. Approximately 50 layers are supported in Part one of the standard.

If the bit-stream is truncated exactly on a layer point, it will be optimal in the rate distortion sense. If the bitstream is truncated part way through a layer, then it will not be optimal, but since many layers are used, the result will be close to optimal[30].

The EBCOT is one of the main resources intensive components of JPEG 2000, it represents the most critical part in the design and implementation of the JPEG2000 standard. The EBCOT Tier-1 encoding scheme consists of two processing stages: context modeling (CM) and matrix quantizer (MQ). Based on algorithm evaluation, CM operates on data stored in a CB and produces context decision (CX/D) pairs. The MQ coder consumes these (CX/D) pairs and generates an embedded bitstream [31].

III.4 Compression performance criteria

Compression performance measures how much the data size is reduced by the compression algorithm. There are standards for measuring compression performance which will be mentioned here :

III.4.1 Compression ratio (CR)

Compression ratio is a measure of the reduction of the detailed coefficient of the data. In the process of image compression, it is important to know how much detailed coefficient one can discard from the input data in order to sanctuary critical information of the original data. The Compression ratio is presented by the quantity

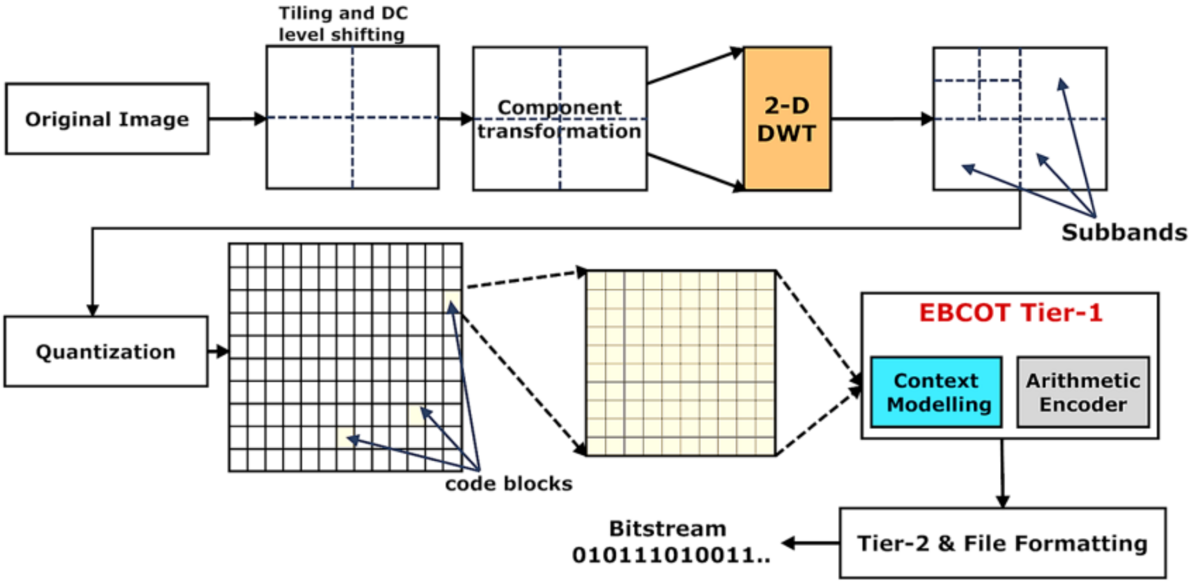


Figure III.4: Block diagram of the JPEG 2000 encoder.

of data of the compressed image and the quantity of data of the original image. It is defined as [32]:

$$CR = \frac{compressed\ image\ size\ (in\ bits)}{original\ image\ size\ (in\ bits)} \qquad (III.10)$$

Achieving high CR will result of less quality of reconstructed images.

III.4.2 Bits Per Pixel (bpp)

Bits per pixel is commonly used in image compression. To measure the compaction power of a compression method, the concept of bit rate is used. Which gives a measurement in bit per pixel. The bpp is defined by [1]:

$$bpp = \frac{number\ of\ bits\ necessary\ to\ code\ the\ image}{number\ of\ pixels\ in\ the\ image} \qquad (III.11)$$

To calculate the bpp for an image in gray levels we can use this equation:

$$bpp = \frac{8\ bits}{CR} \qquad (III.12)$$

And to calculate the bpp for colors image we can use this equation:

$$bpp = \frac{24\ bits}{CR} \qquad (III.13)$$

III.4.3 Percent Root-Mean-Square Difference (PRD)

The way of quantifying the difference between the original and the reconstructed signal called distortion. The most prominently used distortion measure is the Percent Root Mean Square Difference (PRD), it is defined by [33]:

$$PRD = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}[x(i,j) - \hat{x}(i,j)]^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}[x(i,j)]^2}} \times 100 \qquad (III.14)$$

N: The number of rows.
M: The number of columns.
$X(i,j)$: The samples of the original image.
$\hat{x}(i,j)$: The samples of the reconstructed image.

III.4.4 Mean Square Error (MSE)

MSE is another criterion for comparing the compression and enhancement techniques. It is a metric that is used to evaluate the accuracy of denoising. The lower the value of MSE, the closer is the denoised signal to the original, hence better denoising. It is the average of the square of the difference between the original image and the reconstructed image. It is defined by [34]:

$$MSE = \frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M}[x(i,j) - \hat{x}(i,j)]^2 \qquad (III.15)$$

$x(i,j)$ Represents the original image.

$\hat{x}(i, j)$ Represents the reconstructed image.
M and N are the number of rows and columns.

III.4.5 Peak Signal to Noise Ratio (PSNR)

PSNR is the criterion that is selected to measure the comparison between the compression and enhancement techniques. It is the ratio between the maximum possible power of a signal and the power of a corrupting noise. This metric is used as a measure of quality of reconstruction in image compression and image enhancement. The higher the value of PSNR, the more accurate is the denoising, as figure III.5 shown. PSNR can be defined as [34]:

$$PSNR = 20 \times log_{10}(\frac{255}{\sqrt{MSE}}) \qquad (III.16)$$

Figure III.5: Illustration of the PSNR measure and how does it the change effect the image.

III.5 conclusion

In this chapter, we gave an overview of the JPEG and JPEG2000 standards and their different stages for compressing an image. Then we discussed compression performance criteria (PSNR, CR, bpp....) which the designers take into account in the compression process. We found that there is an inverse relationship between PSNR and MSE. consequently, the higher the value of MSE, the lower the value of

PSNR. The lower value of MSE indicates better picture quality, and the high CR will result in less quality. PRD is the distortion measure.

Chapter IV

Used method

IV.1 Introduction

Color images are usually stored in RGB format. In this chapter, we are going to do a comparative study between RGB and YCbCr space, also the difference between the block size 16×16 and 8×8, also the size of quantization (Q), and this by propose a method of compression based on a quantifier and DCT as well as the use of entropic coding Two-Role encoder (TRE). The latter was carried out on the set of known images, and the results obtained and commented will be presented.

IV.2 Compression with color space transformation

The stages of the method will be divided into two main stages (figure IV.1). In this part, we will try the compression with color space transformation. we know that the color image consists of three planes, that means each process will repeat for each plane.

IV.2.1 Compression

In figure IV.1 (a) we give an illustration of the method of compression. We present in the following steps the image compression with details:

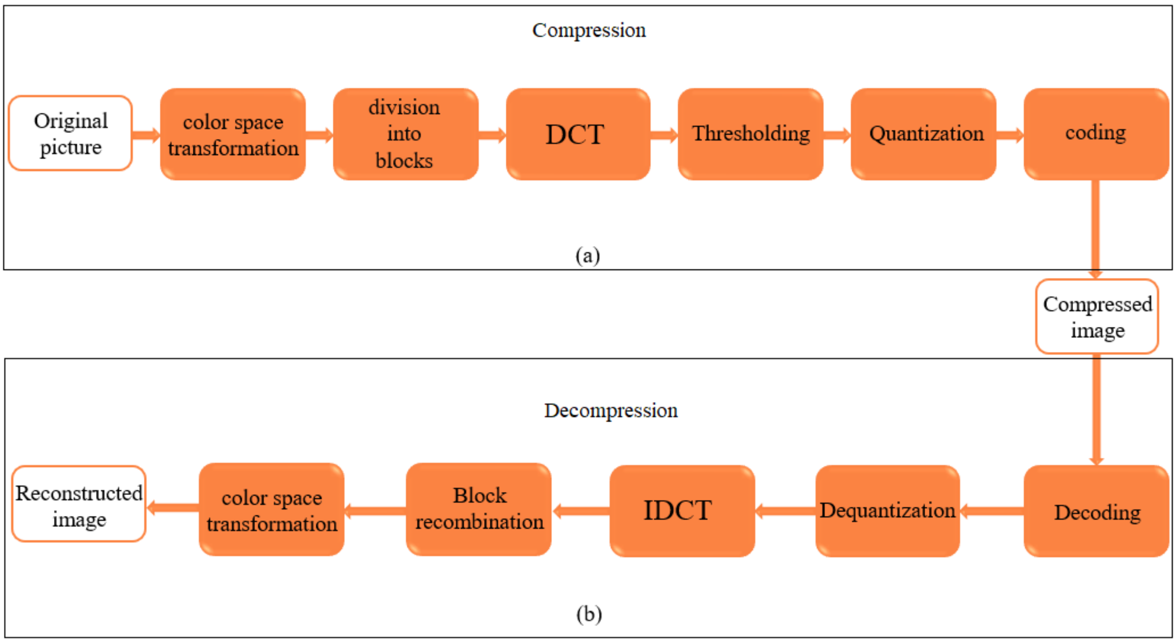


Figure IV.1: block diagram of the used compression steps and decompression steps of color image

IV.2.1.1 Color space transformation

In this step we are going to change of color space RGB to YCbCr; The reason for choosing this process is the RGB space in which the colorimetric information is more or less equally spread over the three planes. in this space the information is mainly stationed in the Y plane and it eliminates threading and redundancy between pixels in RGB space.

IV.2.1.2 Division into blocks

The division is an important step in compression because it limits the number of pixels to be processed and reduces the operations time (processed at the same time). We divide the converted image into blocks of 8×8 and 16×16 pixels. The next steps will be applied to each block separately.

IV.2.1.3 DCT

After the division, DCT will be applied to each block. the result will be a 8×8 and 16×16 block containing the corresponding frequency components of a 8×8 and 16×16 block of the transformed image. In this step, we will get the DCT coefficients of low frequencies and high frequencies. The majority of the energy of each block is concentrate in the low frequencies.

IV.2.1.4 Thresholding

Thresholding is a process of eliminating some values that are between the threshold values (absolute value). The threshold leads to a loss in the precision of the image, this loss is changed according to the value of the threshold. If the threshold value is high, the loss will be considerable, and vice versa.

We will offset the DCT coefficients confined between the threshold values with zero. We will get zero coefficients and non-zero coefficients.

IV.2.1.5 Quantization

After the threshold process, non-zero coefficients $NZDCT$ are quantified by a linear quantization of size Q bits. The quantization $QNZDCT$ of the coefficients $NZDCT$ is carried out by the following formula [40]:

$$QNZDCT = [1 + (\frac{NZDCT - NZDCT_{min}}{NZDCT_{max} - NZDCT_{min}})(2^Q - 2)] \qquad (IV.1)$$

Where: [.] represents the rounding to the nearest integer.
$NZDCT$: is the non-zero coefficients.
$NZDCT_{max}$: is the maximum value of $NZDCT$.
$NZDCT_{min}$: is the minimum value of $NZDCT$.
After this process, produces scalar values distributed between 0 and $2^Q - 1$, and many long runs of zeros. In the next step, we will see how we deal with it.

IV.2.1.6 Coding

Firstly, each 8×8 and 16×16 block is scanned by four paths Zigzag, Horizontal, Vertical, and Hilbert to find the best path with the most possible zeros at the end. the scanning is shown in the figure IV.2. these paths are represented by 2 bits as follows:

AS = 00 symbolizes to the Zigzag scan.

AS = 01 symbolizes to the Horizontal scan.

AS = 10 symbolizes to the Vertical scan.

AS = 11 symbolizes to the the Hilbert scan.

Then, the quantized coefficients we obtained are encoded by the four paths by a TRE (Two-Role-Encoder) lossless encoder, and this is by replacing each run of contiguouns zeros with TRE code which changes from 1 and $2^Q - 1$ depending on the long run of zeros.

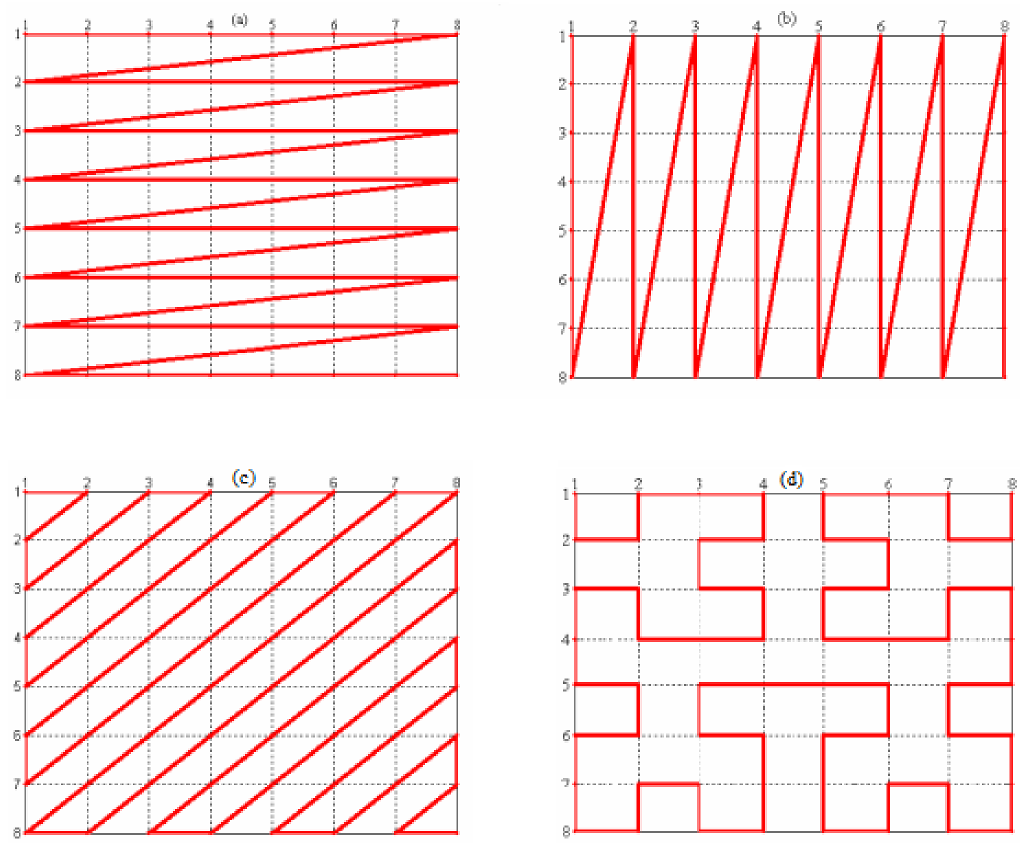


Figure IV.2: The different paths for scanning blocks: Horizontal (a), Vertical (b), Zigzag (c), and Hilbert (d) [41].

Example: We chose block X of 8×8 pixels represented from Lena's image in the grayscale level of size 512×512:

- block X

$$X = \begin{bmatrix} 103 & 100 & 104 & 105 & 105 & 105 & 104 & 110 \\ 103 & 100 & 104 & 105 & 105 & 105 & 104 & 110 \\ 103 & 100 & 104 & 105 & 105 & 105 & 104 & 110 \\ 103 & 100 & 104 & 105 & 105 & 105 & 104 & 110 \\ 103 & 100 & 104 & 105 & 105 & 105 & 104 & 110 \\ 101 & 101 & 98 & 100 & 108 & 104 & 106 & 106 \\ 97 & 96 & 96 & 103 & 104 & 109 & 111 & 104 \\ 92 & 100 & 97 & 99 & 105 & 110 & 106 & 109 \end{bmatrix}$$

- DCT Application

$$XDCT = \begin{bmatrix} 830 & -22 & 0 & 0 & 6 & -2 & 3 & -1 \\ 7 & 9 & 2 & -6 & 6 & 2 & 5 & 1 \\ -3 & -6 & -1 & 2 & -4 & -1 & -1 & -2 \\ 0 & 2 & 1 & 2 & 1 & 1 & -2 & 2 \\ 1 & 0 & 0 & -3 & 0 & -2 & 3 & -1 \\ -1 & -1 & -1 & 2 & -1 & 4 & -1 & 0 \\ -1 & 1 & 1 & 0 & 0 & -4 & -1 & 1 \\ -1 & -1 & -1 & -1 & 0 & 3 & 1 & -1 \end{bmatrix}$$

- Thresholding: With choosing THR = 3 we get the following block:

$$XDCT_{THR} = \begin{bmatrix} 830 & -22 & 0 & 0 & 6 & 0 & 0 & 0 \\ 7 & 9 & 0 & -6 & 6 & 0 & 5 & 0 \\ 0 & -6 & 0 & 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- quantization: By selecting Q=8 we will get the following block:

$$QXDCT = \begin{bmatrix} 147 & 51 & 0 & 0 & 54 & 0 & 0 & 0 \\ 54 & 54 & 0 & 53 & 54 & 0 & 54 & 0 \\ 0 & 54 & 0 & 0 & 53 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 54 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 53 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

As a result: the zero coefficients are not quantized and the maximum value of the thresholded DCT block 830 becomes 147 and the minimum value of -22 becomes 51.

Also we found that the global minimum values are -465 and maximum values are 1792 of DCT blocks.

The results of the adaptive scan step to the different paths are shown below:

zigzag scan (AS=00)
147 51 54 0 54 0 0 0 54 0 0 0 0 53 54 0 54 0 0 0 0 0 0 0 0 53 0 0 0 54 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 54 0 0 0 0 53 0 0 0 0 0 0 0
After TRE
147 51 54 1 54 3 54 4 53 54 1 54 8 53 3 54 21 54 4 53

Table IV.1: The application of the Zigzag scan on the QXDCT and their TRE code.

Horizontal scan (AS=01)
147 51 0 0 54 0 0 0 54 54 0 53 54 0 54 0 0 54 0 0 53 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 54 0 0 0 0 0 0 0 53 0 0 0 0 0 0 0 0 0 0
After TRE
147 51 2 54 3 54 54 1 53 54 1 54 2 54 2 53 24 54 7 53

Table IV.2: The application of the Horizontal scan on the QXDCT and their TRE code.

Vertical scan (AS=10)
147 54 0 0 0 0 0 0 51 54 54 0 0 0 0 0 0 0 0 0 0 0 0 0 0 53 0 0 0 0 0 0 54 54 53 0 0 0 0 0 0 0 0 0 0 54 53 0 0 54 0 0 0 0 0 0 0 0 0 0 0 0 0 0
After TRE
147 54 6 51 54 54 14 53 6 54 54 53 10 54 53 2 54

Table IV.3: The application of the Vertical scan on the QXDCT and their TRE code.

Hilbert scan (AS=11)
147 51 54 54 0 0 0 54 0 0 0 0 53 0 0 0 54 54 0 0 0 0 0 54 0 0 0 0 0 0 53 0 0 0 54 0 0 0 0 0 0 0 0 0 0 53 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
After TRE
147 51 54 54 3 54 4 53 3 54 54 5 54 6 53 3 54 10 53

Table IV.4: The application of the Hilbert scan on the QXDCT and their TRE code.

IV.2.2 Decompression

In order to reconstruct the image, we perform the reverse process of compression as shown in figure IV.1 (b). i.e, decoding, dequantization, and the inverse discrete cosine transform are applied. Taking into consideration that the dequantization is done according to the following formula:

$$NZDCT = \frac{(QNZDCT - 1) \times (NZDCT_{max} - NZDCT_{min})}{2^Q - 2} + NZDCT_{min}$$

IV.3 Experimental results

In this section, we present the experimental results of the simulation using Matlab 2015a obtained from several images.

We used color images coded on 24 bits with different sizes to check this method. these images are created to allow researchers to compare their algorithms and results on the same images.

These images are: Airplane, Peppers, Lena of size 512×512, and Girl, Couple, House, and Zelda of size 256×256 (see figure IV.3).

Airplane

Peppers

Lena

Girl

Couple

House

Zelda

Figure IV.3: The different images we used to test.

The evaluation of a compression algorithm is commonly carried out according to the quality measurement as PSNR, bpp,.. and compression rate as well as the execution time. depending on this, we select bpp and PSNR as follows [40]:

$$bpp = \frac{24\ bits}{CR} \qquad (IV.2)$$

with:

$$CR = \frac{original\ image\ size\ in\ bits}{compressed\ image\ in\ bits} \qquad (IV.3)$$

$$PSNR = 10 \times log_{10}\left(\frac{255^2 \times 3}{MSE(R) + MSE(G) + MSE(B)}\right) \qquad (IV.4)$$

To numerically evaluate the performance, we calculated the PSNR and the number of bpp and CR; The results mentioned in table IV.5 are when we compressed with RGB space.

	RGB									
	PSNR	CR_1	bpp_1	CR_2	bpp_2	PSNR	CR_1	bpp_1	CR_2	bpp_2
Q=7	8×8					16×16				
Airplane	30.71	11.14	2.16	15.36	1.56	28.15	3.27	7.34	7.51	3.20
Peppers	28.27	13.15	1.83	16.51	1.45	28.41	8.91	2.69	14.35	1.67
Lena	32.69	3.16	7.60	5.60	4.28	28.16	3.08	7.80	6.60	3.64
Girl	35.15	7.60	3.16	12.12	1.98	31.49	6.53	3.67	13.15	1.83
Couple	31.19	9.38	2.56	12.02	2.00	30.72	3.49	6.88	6.61	3.63
House	30.84	9.37	2.56	12.51	1.92	29.30	3.48	6.89	7.51	3.20
Zelda	31.25	6.27	3.83	8.77	2.74	28.82	2.96	8.09	6.07	3.96
Average	31.44	8.58	3.385	11.84	2.28	29.29	4.53	6.19	8.82	3.01
Q=8	8×8					16×16				
Airplane	30.62	10.79	2.22	14.15	1.70	30.81	9.38	2.56	15.30	1.56
Peppers	28.28	12.21	1.97	15.02	1.59	28.21	11.30	2.12	16.94	1.42
Lena	31.09	7.76	3.09	10.75	2.23	30.68	7.95	3.02	13.02	1.84
Girl	35.34	11.30	2.12	14.95	1.61	34.99	5.72	4.19	10.72	2.24
Couple	31.51	8.11	2.96	10.18	2.36	31.32	7.18	3.34	10.66	2.25
House	30.99	8.89	2.70	11.21	2.14	30.97	8.13	2.95	12.30	1.95
Zelda	31.51	6.05	3.97	8.17	2.94	31.69	4.70	5.11	7.87	3.05
Average	**31.36**	**9.19**	**2.73**	**11.95**	**2.09**	**31.24**	**7.77**	**3.33**	**12.40**	**2.04**
Q=9	8×8					16×16				
Airplane	30.18	10.52	2.28	13.12	1.83	35.84	3.07	7.80	5.92	4.06
Peppers	28.33	10.80	2.22	12.95	1.85	28.20	10.78	2.28	15.29	1.57
Lena	31.61	6.21	3.86	8.57	2.80	31.40	6.34	3.79	10.39	2.31
Girl	35.16	10.83	2.22	13.10	1.83	35.72	7.72	3.11	12.55	1.91
Couple	31.59	7.14	3.36	8.65	2.77	31.18	6.96	3.45	9.93	2.42
House	32.81	5.43	4.42	7.29	3.29	32.68	5.41	4.43	8.59	2.79
Zelda	31.66	5.34	4.49	7.06	3.40	31.66	5.34	4.49	7.06	3.40
Average	31.62	8.04	3.26	10.10	2.53	32.38	6.52	4.19	9.96	2.64

Table IV.5: The performances measured on the reconstructed images (RGB).

CR_1 : represents the value of CR without arithmetic coding.

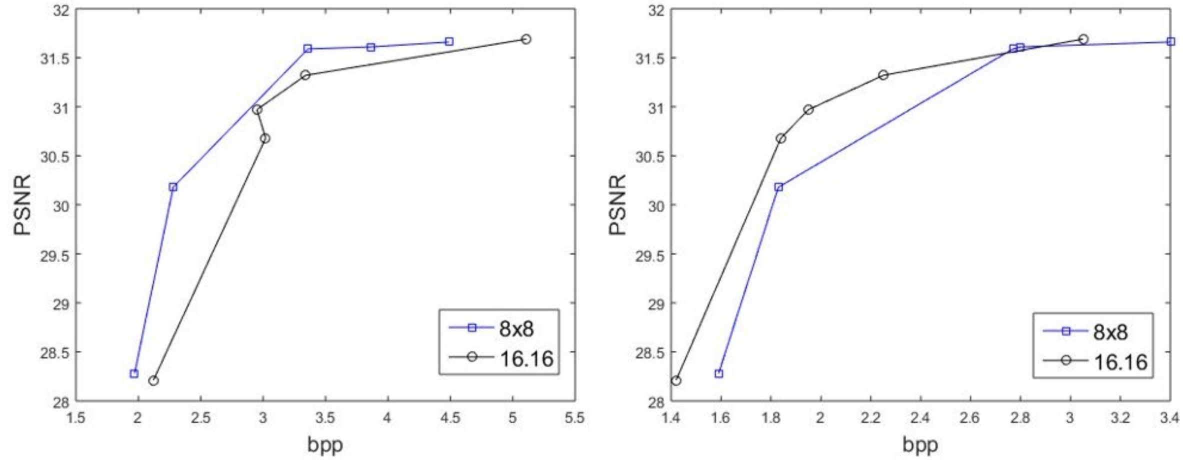
bpp_1 : represents the value of bpp without arithmetic coding.

CR_2 : represents the value of CR with an arithmetic coding.

bpp_2 : represents the value of bpp with Huffman coding.

As an observation, the quality of the restored images obtained with the 16×16 with Q = 8 is the best choice to compress.

The results obtained were explained in the following charts:



(a) (b)

Figure IV.4: Comparison of the compression performance for 16x16 and 8x8 blocks of compression without arithmetic coding (a) and with it (b).

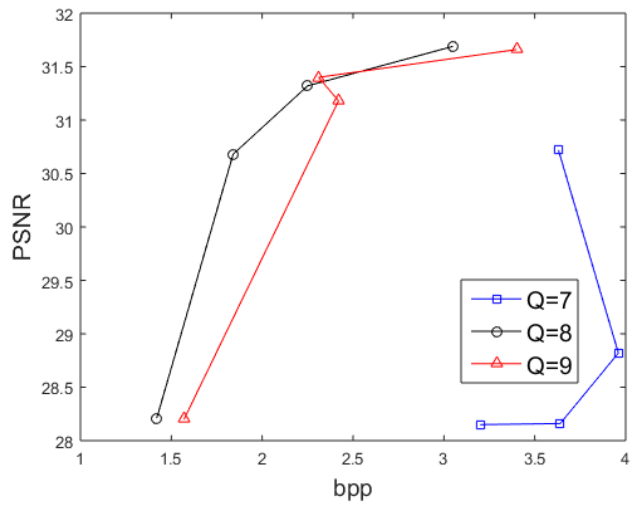


Figure IV.5: Comparison of the compression performance for 16x16 block with a 3 different size of quantization (Q).

The fee in figure IV.5 and figure IV.4 confirmed the superiority of pressure with huffman coding , and the results were better with Q = 8

The images reconstructed according to the proposed method are shown in the figure IV.6.

After a color space change from RGB to YCbCr, we calculated the PSNR and the number of bpp and CR one more time; And we represent the results in table IV.6.

Figure IV.6: The reconstructed images without transformation (RGB).

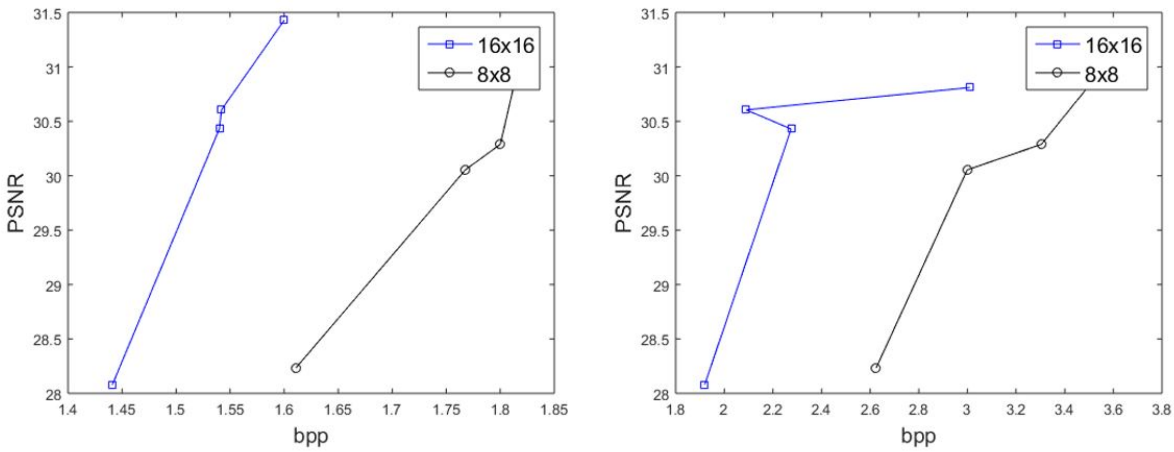
	YCbCr									
	PSNR	CR_1	bpp_1	CR_2	bpp_2	PSNR	CR_1	bpp_1	CR_2	bpp_2
Q=7	8×8					16×16				
Airplane	30.82	9.24	2.60	15.22	1.58	27.61	10.95	2.19	21.74	1.10
Peppers	28.87	3.91	6.13	6.93	3.46	27.07	10.11	2.37	17.24	1.39
Lena	30.43	8.15	2.94	13.10	1.83	28.12	14.86	1.62	25.85	0.92
Girl	33.49	17.98	1.33	28.71	0.84	32.15	21.08	1.14	38.19	0.63
Couple	31.61	8.78	2.73	13.44	1.79	28.73	6.60	3.64	12.77	1.88
House	30.78	8.22	2.92	12.99	1.85	27.51	8.23	2.92	15.20	1.57
Zelda	30.37	7.30	3.29	11.52	2.08	27.58	7.17	3.35	13.42	1.79
average	30.91	9.08	3.13	14.55	1.62	28.40	11.29	2.46	20.63	1.33
Q=8	8×8					16×16				
Airplane	31.28	12.92	1.86	19.05	1.30	31.42	6.64	3.61	13.28	1.81
Peppers	28.08	12.50	1.92	16.66	1.44	28.24	9.14	2.62	14.90	1.60
Lena	30.43	10.55	2.75	15.58	1.54	30.39	7.26	3.30	13.33	1.80
Girl	34.65	19.72	1.22	28.70	0.93	34.54	13.02	1.84	24.56	0.97
Couple	31.43	10.74	2.23	13.99	1.63	31.21	7.46	3.22	13.05	1.84
House	30.61	11.49	2.10	15.57	1.54	30.44	8.86	2.71	14.92	1.60
Zelda	30.81	7.97	3.01	11.73	2.07	30.17	8.00	3.00	13.58	1.77
average	**31.04**	**12.27**	**2.15**	**17.33**	**1.49**	**30.92**	**8.63**	**2.9**	**15.37**	**1.48**
Q=9	8×8					16×16				
Airplane	30.43	14.19	1.69	19.11	1.26	31.19	12.26	1.96	20.25	1.19
Peppers	30.96	4.11	5.84	6.60	3.63	28.80	8.13	2.95	13.02	1.84
Lena	31.81	6.31	3.81	9.84	2.44	31.34	6.44	3.72	11.73	2.05
Girl	36.03	13.94	1.72	19.84	1.21	35.34	13.99	1.72	23.96	1.00
Couple	32.96	6.75	3.56	9.80	2.45	32.38	6.61	3.63	11.31	2.12
House	30.80	10.20	2.35	13.04	1.84	30.62	10.76	2.23	16.23	1.48
Zelda	31.77	5.64	4.26	8.40	2.86	31.39	5.59	4.29	9.72	2.47
Average	32.11	8.73	3.32	12.38	2.24	31.58	9.11	2.93	15.74	1.74

Table IV.6: The performances measured on the reconstructed images (YCbCr).

As an observation, from the results of the table IV.6 that for the YCbCr space, the best result is for Q=8 and a block size of 16 × 16.

The superiority of the 16×16 split with Q=8 in compression with YCbCr transform as spotted on the graph of figure IV.7 and figure IV.8 .

Figure IV.10 represent 2 graph of lena image: (a) the different value in 8x8 block, (b) the different values in 16x16 block.



(a) (b)

Figure IV.7: Comparison of the compression performance for 16x16 and 8x8 blocks of compression with arithmetic coding (a) and without it (b).

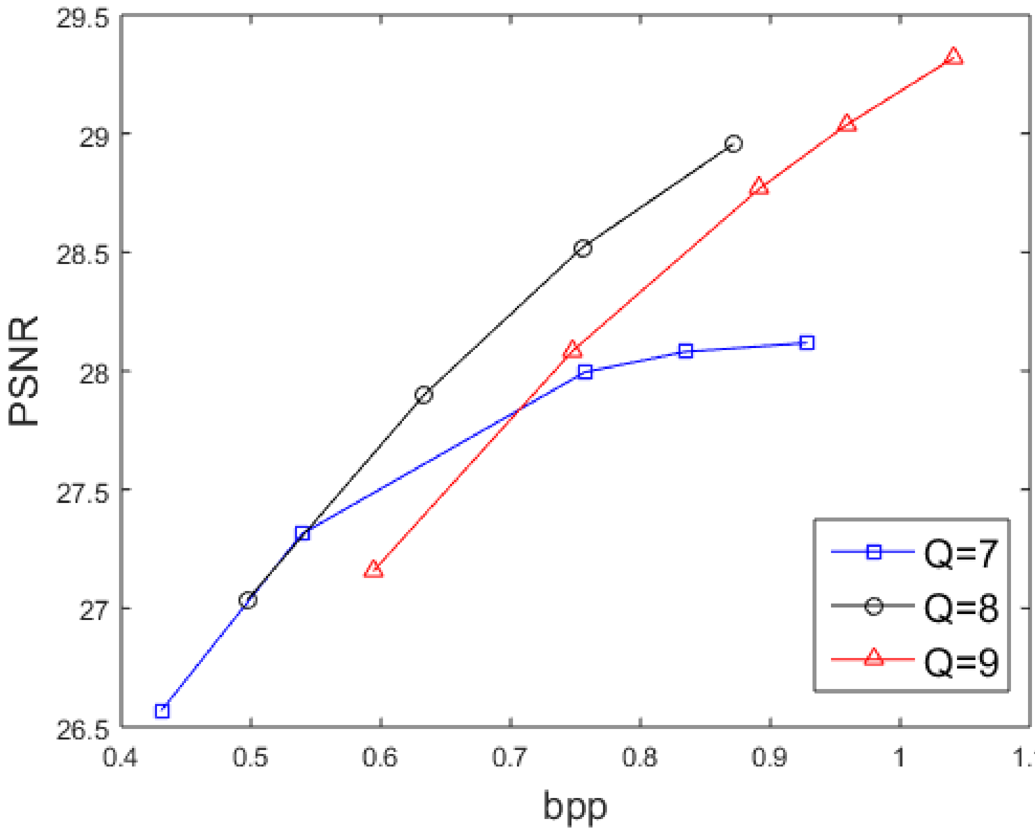
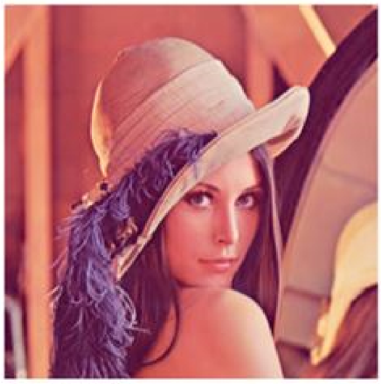


Figure IV.8: Comparison of the compression performance for 16x16 block with Q=7, Q=8 and Q=9.

PSNR = 31.42
bpp2 = 1.81

PSNR = 28.24
bpp2 = 1.60



PSNR = 30.39
bpp2 = 1.80

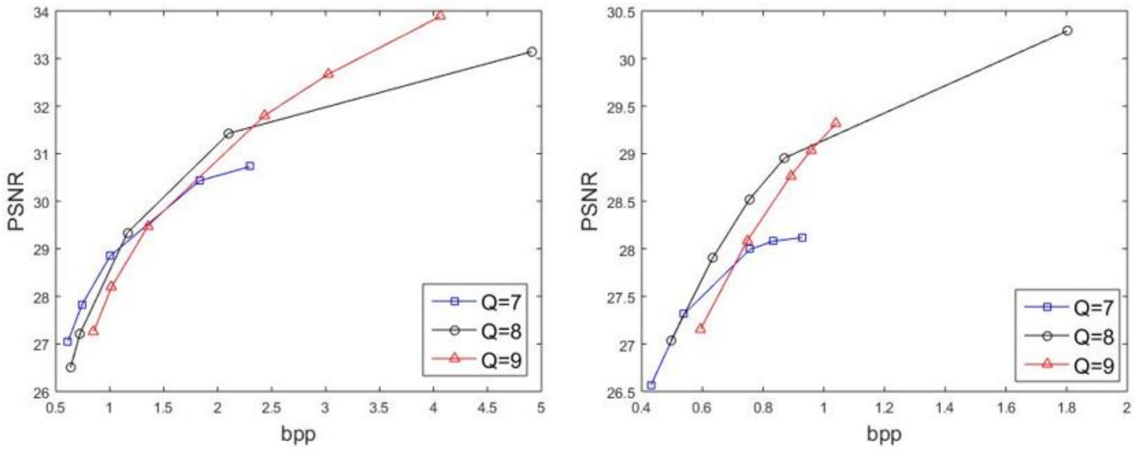
PSNR = 34.54
bpp2 = 0.97

PSNR = 31.21
bpp2 = 1.84

PSNR = 30.44
bpp2 = 1.60

PSNR = 30.17
bpp2 = 1.77

Figure IV.9: Images reconstructed with transformation (YCbCr).



(a) (b)

Figure IV.10: Diagram showing the change of bpp and PSNR for Lena image.

Figure IV.11 represents the reconstructed Lena image in YCbCr space for different values of PSNR and bpp.

As a note, that transformation (YCbCr space) allows more efficient compression. The results of the compared are reported in table IV.7.

	YCbCr			RGB		
	PSNR	CR	bpp	PSNR	CR	bpp
Airplane	31.42	13.28	1.81	30.81	15.30	1.56
Peppers	28.24	14.90	1.60	28.21	16.94	1.42
Lena	30.39	13.33	1.80	30.68	13.08	1.84
Girl	34.54	24.56	0.97	34.99	10.72	2.24
Couple	31.21	13.05	1.84	31.32	10.66	2.24
House	30.44	14.92	1.60	31.32	10.66	2.25
Zelda	31.21	9.00	2.66	30.97	12.30	1.95
average	30.17	13.58	1.64	31.69	7.87	3.05

Table IV.7: Compression performance of each test image in RGB and YCbCr spaces with block size 16×16 (summary of table IV.5 and IV.6).

It is clear that the results obtained after applying the YCbCr transformation in terms of compression and distortion rate are better than those obtained by applying compression directly to an RGB image.

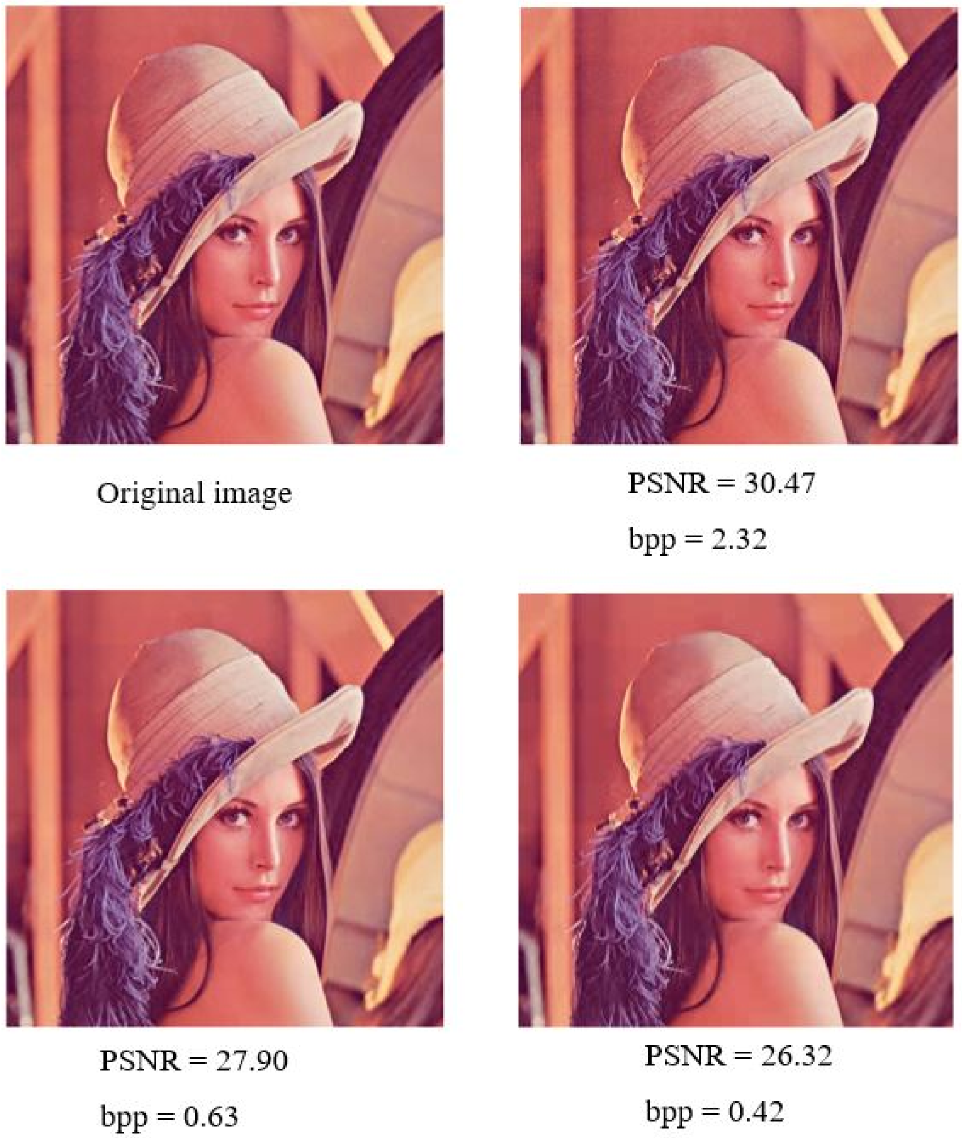


Figure IV.11: The performance of the Lena image compression in the YCbCr space.

IV.4 Comparative study

In this section, we compare the performance of the proposed method with the image compression method based on DCT transform combined with an adaptive block scannin. This method uses the TRE and the DCT transformation. These comparisons are illustrated in table IV.8.

The results that we have found are not enough comparing to the Article [40], but it gave us a clear image that we can rely on.

	proposed method			DOUAK [40]		
	PSNR	CR	bpp	PSNR	CR	bpp
Airplane	30.31	23.71	1.01	30.25	38.42	0.62
Peppers	28.24	14.90	1.60	30.19	26.70	0.90
Lena	30.39	13.33	1.80	31.94	27.83	0.86
Girl	34.54	24.56	0.97	35.15	47.10	0.51
Couple	31.21	13.05	1.84	32.45	22.90	1.05
House	30.44	14.92	1.60	31.83	25.92	0.93
Zelda	31.21	9.00	2.66	31.27	25.78	0.93
average	30.91	16.21	1.64	31.87	30.66	0.83

Table IV.8: The performances measured on the reconstructed images (YCbCr).

IV.5 Graphic User Interface (GUI)

In order to present our explicit image compression method, we test a platform realized using Matlab R2015a software, where the input of the simulation interface is an RGB image, which will be compressed. The output is a reconstructed image. In Figure IV.12 we will present our simulator.

Figure IV.12: Simulator interface.

1. First of all select an image (Figure IV.13).

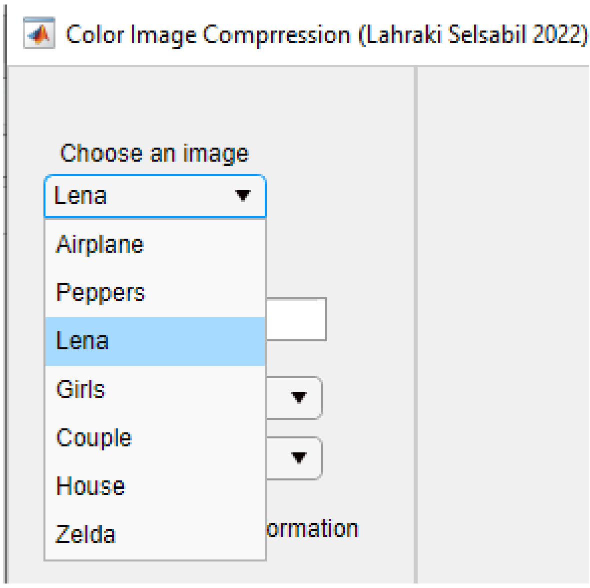


Figure IV.13: Select an image.

2. Secondly choose the value of the Thresholding (Figure IV.14)

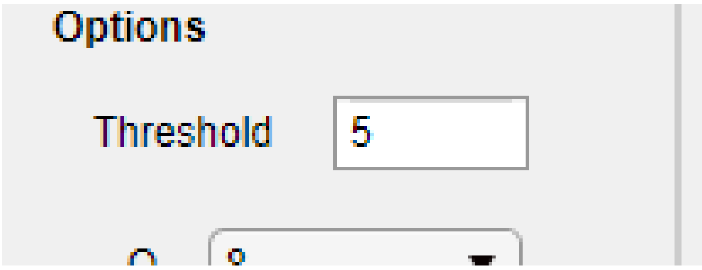


Figure IV.14: Thresholding value.

3. Thirdly pick the size of the quantization (Figure IV.15).

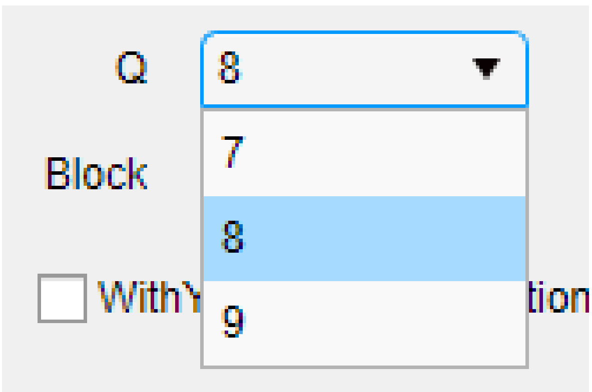


Figure IV.15: size of quantization.

4. Next choose the size of the block (Figure IV.16).

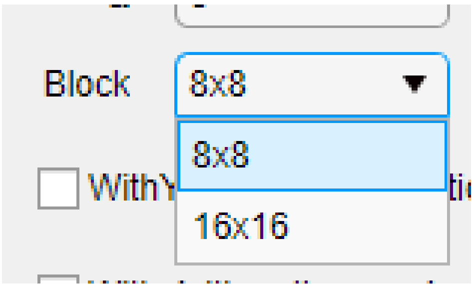


Figure IV.16: size of block.

5. Then select the style of color transformation (Figure IV.17 IV.18)

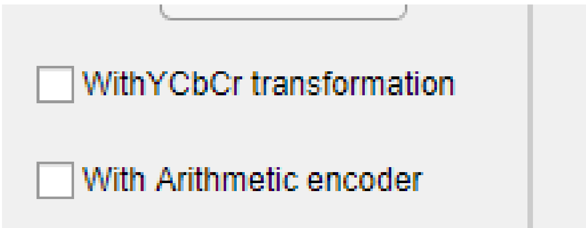


Figure IV.17: choice of color space (1).

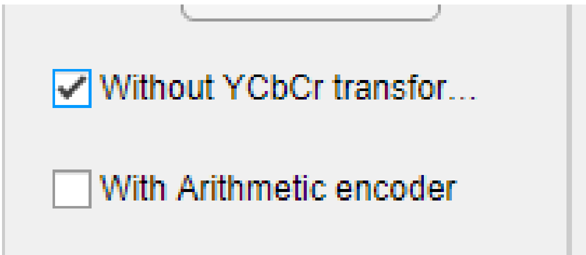


Figure IV.18: choice of color space (2).

IV.6 conclusion

In this chapter, we have proposed an image compression technique by cutting the image into blocks, and a change of space which achieves a better compression ratio. after we get the DCT coefficient applied The thresholding step (delete part of the DCT coefficient), at the end a lossless compression step is applied (arithmetic coding).

applying the algorithm on color images, showed that better performance can be achieved with YCbCr transform compared to RGB color space, and a block of size 16x16 is better than 8x8 with a quantization size equal to 8 to this compression scheme, and we noted that the compression with the arithmetic coding achieves better results than the compression without it.

The results that we have obtained in this chapter are rather satisfactory from the point of view of improving the performance of the encoder (PSNR, compression rate Cr, number of bits per pixel bpp); But compared to that the article [40] it was a little weak.

General conclusion

Compression is born from the development of digitization and multimedia techniques, consists of coding all of this data on a reduced number of bits. This is why the Compression is important therefore appears to be unavoidable today to fulfill the functions of archiving and rapid transmission. Indeed, it would be illusory to believe that compression becomes obsolete because of the capacity of memories which is constantly increasing, because the quantity of data to be stored grows in the same way, and networks evolve much more slowly. In image compression Redundant information is eliminated while the most essential information for obtaining of a good quality final image is preserved; And this by a coding scheme who consists of a quantization step and a coding step.

In the first chapter we presented the light and how human eye can see the different colors. In trying to explain the mechanisms that govern human vision. also several color models have been proposed and their range of systems. These systems are developed by different organizations. These various models, combined with technological constraints, linked to certain industries, have led to the coexistence of a large number color systems. as well we defineded images and we mentioned its characteristics.

In the second chapter, we approached the problem of the compression of color images; there are two major categories of techniques known as lossless and lossy techniques are distinguished. Lossless compression ensures full reconstruction of the original image following the decompression process. This type of compression uses only redundancies in the image which did not allow a significant reduction in the volume of the image. Lossy compression is defined to implement efficient compression, while controlling the quality degradation of the reconstructed image.

Thus, it seeks to guarantee a better bit rate-distortion compromise. Such a compression technique is necessary for applications requiring high compression rates.

In the third chapter we dealt with 2 different formats and their working mechanism, in which we find that the JPEG is based on DCT which transforms the image from the spatial domain to the frequency domain that produces low frequencies which symbolize DC and high frequencies which symbolize AC. JPEG2000 this format based on DWT that divides the block into 4 parts LL,HL,LH,HH and each block is divided to another 4 parts. then we represent the performance criteria that the scientists rely on to see the reliability of their algorithm on the image quality.

At the end we propose a method similar to JPEG based on DCT and TRE coding ; The main goal of this work is the study of the effect of the color space in color image compression; i.e, which color space achieves better compression ratio. based on this, an algorithm has been developed containing of discrete cosine transform (DCT). its idea is similar to JPEG. Applying the algorithm on color images, showed that better performance can be achieved with YCbCr transform compared to RGB color space, and blocks of size (16x16) is better then (8x8) with a quantization size equal to 8 to this compression scheme. Also, we noted that the compression with the arithmetic coding achieves better results than the compression without it.

Bibliography

[1] A. Anwander, Segmentation d'images couleur par un opérateur gradient vectoriel multiéchelle et contour actif: application à la quantification des phases minéralogiques du clinker de cimen. PhD thesis, Lyon, INSA, 2001.

[2] M. A. Bourenane, Un outil pour l'indexation des vidéos personnelles par le contenu. PhD thesis, Université du Québec à Trois-Rivières, 2009.

[3] M. Benabdellah, "Outils de compression et de crypto-compression: Applications aux images fixes et vidéo," 2007.

[4] M. A. Bourenane, Un outil pour l'indexation des vidéos personnelles par le contenu. PhD thesis, Université du Québec à Trois-Rivières, 2009.

[5] A. MESSAOUDI, Contribution à l'amélioration et la mise en oeuvre de nouveaux algorithmes pour la compression des signaux. PhD thesis, Université de Batna 2, 2017.

[6] M. Benabdellah, "Outils de compression et de crypto-compression: Applications aux images fixes et vidéo," 2007.

[7] A. Boucetta, Etude de l'effet des Transformées de Décorrélation en Compression des Images Couleurs RGB. PhD thesis, Université de Batna 2, 2010.

[8] M. Bergounioux, Introduction au traitement mathématique des images-méthodes déterministes, vol. 76. Springer, 2015.

[9] K. Taous and B. Souad, Segmentation marginale d'image couleur. PhD thesis, Université Mouloud Mammeri, 2011.

[10] C. Houassine, Segmentation d'images par une approche biomimétique hybride. PhD thesis, 2012.

[11] T. Mohamed and D. Djafar, "La découverte des concepts sémantiques cachés avec plusieurs niveaux d'abstraction pour la recherche d'images," Université kasdi merbah ouargla, 2013.

[12] H. Sarra, BENFRIHA et Asma, Segmentation d'Image par Coopération Régions-Contours. PhD thesis, Université Kasdi Merbah ouargla, 2016.

[13] O. R. Mustapha and C. Malis, Analyse D'Images Par Filtres De Gabor: Application Aux Images Médicales. PhD thesis, Université Mouloud Mammeri, 2017.

[14] "Vector images vs bitmap images." https://documentation.designer.io/working-with-images/vector-vs-bitmap/, 2022. [Online; accessed 17-05-2022].

[15] J. Miano, Compressed image file formats: Jpeg, png, gif, xbm, bmp. Addison-Wesley Professional, 1999.

[16] "Svg, pdf, jpg, png; what's the difference?." https://www.95visual.com/blog/svg-pdf-jpg-png-whats-the-difference. [Online; accessed 25-05-2022].

[17] "Ai files." https://www.adobe.com/creativecloud/file-types/image/vector/ai-file.html. [Online; accessed 25-05-2022].

[18] B. Mekki, M. OMARI, et al. Image Compression Using Logarithmic Functions with Pixels Shifting. PhD thesis, Ahmed Draia University-Adrar, 2016.

[19] M. A. Joshi, M. S. Raval, Y. H. Dandawate, K. R. Joshi, and S. P. Metkar, Image and video compression: Fundamentals, Techniques, and Applications. CRC press, 2014.

[20] Z. Wang, Y. Mao, and I. Kiringa, "Non-binary distributed arithmetic coding," in 2015 IEEE 14th Canadian Workshop on Information Theory (CWIT), pp. 5–8, IEEE, 2015.

[21] R. B. Ruchi Gupta, Mukesh Kumar, "Data compression - lossless and lossy techniques," 2016.

[22] N. Saldi, Optimal quantization and approximation in source coding and stochastic control. PhD thesis, Queen's University (Canada), 2015.

[23] Y. Q. Shi and H. Sun, Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards. CRC press, 1999.

[24] D. Hui and D. L. Neuhoff, "Asymptotic analysis of optimal fixed-rate uniform scalar quantization," IEEE Transactions on Information Theory, vol. 47, no. 3, pp. 957–977, 2001.

[25] Y. Xing, M. Kaaniche, B. Pesquet-Popescu, and F. Dufaux, “Chapter 3 - basic compression tools,” in Digital Holographic Data Representation and Compression (Y. Xing, M. Kaaniche, B. Pesquet-Popescu, and F. Dufaux, eds.), pp. 19–24, Academic Press, 2016.

[26] N. Moreau, “Outils pour la compression application à la compression des signaux audio,” Télécoms. Lavoisier, 2009.

[27] M. S. Ibraheem, Logarithmic Discrete Wavelet Transform For High Quality Medical Image Compression. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2017.

[28] E. Doutsi, Retina-inspired image and video coding. PhD thesis, Université Côted’Azur, 2017.

[29] S. A. Tahar, Etude de la compression des images de visages sous la norme JPEG 2000. PhD thesis, Alger, 2006.

[30] F. Galvan, A. Policriti, S. Battiato, and C. XXVII—AY, “First quantization table detection in double compressed jpeg images,” 2016.

[31] G. K. Wallace, “The jpeg still picture compression standard,” IEEE transactions on consumer electronics, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[32] E. Zeybek, Compression multimodale du signal et de l’image en utilisant un seul codeur. PhD thesis, Université Paris-Est, 2011.

[33] M. S. Ibraheem, Logarithmic Discrete Wavelet Transform For High Quality Medical Image Compression. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2017.

[34] D. Jérémie, JPEG 2000 Encoder Development. PhD thesis, KTH Department of Information Technology and Microelectronics, 2006.

[35] C. Christopoulos, A. Skodras, and T. Ebrahimi, “The jpeg2000 still image coding system: an overview,” IEEE transactions on consumer electronics, vol. 46, no. 4, pp. 1103–1127, 2000.

[36] R. Ghodhbani, T. Saidani, L. Horrigue, and M. Atri, “An efficient pass-parallel architecture for embedded block coder in jpeg 2000,” Journal of Real-Time Image Processing, vol. 16, no. 5, pp. 1595–1606, 2019.

[37] A. Deshlahra, Analysis of Image Compression Methods Based On Transform and Fractal Coding. PhD thesis, 2013.

[38] S. V. Tummala and V. Marni, “Comparison of image compression and enhancement techniques for image quality in medical images”. 2017.

[39] S. V. Tummala and V. Marni, “Comparison of image compression and enhancement techniques for image quality in medical images”. 2017.

[40] F. Douak, R. Benzid, and N. Benoudjit, “Color image compression algorithm based on the dct transform combined to an adaptive block scanning,” AEU- International Journal of Electronics and Communications, vol. 65, no. 1, pp. 16–26, 2011

[41] F. Douak, Reconstruction des images compressées en utilisant les réseaux de neurones artificiels et la DCT. PhD thesis, Université de Batna 2, 2008.