People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
M'Hamed BOUGARA University – Boumerdes

Institute of Electrical and Electronic Engineering
Department of Electronics

Final Year Project Report Presented in Partial Fulfilment for the Requirements of the Degree of

MASTER

In **Electrical and Electronic Engineering**
Option: **Telecommunications**

Title:

A Study and Optimization of the JPEG2000 Image Compression Algorithm

Presented by:

- **BOUMEDINE Ahmed Yassine**
- **NEHAR Sallah Eddine**

Supervisor:

Dr. Fatma KEROUH

Registration Number:………/2016

Registration Number:……..../2016

Dedication

I dedicate my work to my family and many friends. A special feeling of gratitude to my loving parents, they have never left my side. I also dedicate this work to my classmates who have supported me throughout the process. I will always appreciate all they have done.

Boumedine Ahmed Yassine

I dedicate this work to dear Mother, Father and family who never gave up on me. .

I also dedicate this work to my friends whom I spent with my best times.

Nehar Sallah Eddine

Acknowledgements

We are indebted to many people who has helped, supported and encouraged during the writing of this project.

Our deepest thanks goes to our advisor Dr. Fatma Kerouh, who remained patient with us and guided us all the way through the project, provided us all the necessary tools and documents to finish the work.

Our gratitude also goes to our classmate Bouiacha Abdenour, who supported us with all he could.

We would like to thank our institution and faculty members without whom this project would be far from reality. We also extend our heartfelt thanks to our families friends and wishers.

Abstract

The aim of this project is to study and optimize JPEG2000 compression algorithm. JPEG2000 is a compression standard that uses wavelet transform. A color space transformation is first achieved, then the compression process in the wavelet domain is applied. Our purpose is to find the optimal color space, optimal wavelet, and optimal resolution level to compress any image. To carry out the experiments, the basic JPEG2000 algorithm is first implemented using MATLAB software. Then, tests are performed on different images selected from a free database named LIVE. For the study, three color spaces have been considered, Red, Green, Blue (RGB), Luminance/ Chrominance (YCbCr), and Hue/ Saturation/ Value (HSV). Furthermore, different wavelets and resolution levels have been considered. The compressed images are evaluated in terms of compression ratio, Detail Index (DI), and three quality metrics a full reference spatial metric named PSNR (Peak Signal to Noise Ratio), structural quality metric named MS-SSIM, and a blur metric (IQA).

Key Words: Image Compression, JPEG2000, Optimized Algorithm, Discrete Wavelet Transform (DWT), Compression Ratio (CR), Detail Index (DI), PSNR, MS-SSIM, IQA.

Table of Content

List of Tables

List of Figures

Nomenclature

JPEG:	Joint Photographic Experts Group
PSNR:	Peak Signal-to-Noise Ratio
MSE:	Root Mean Square Error
MS-SSIM:	Multi Scale Structural Similarity Index Measurement
QM:	The Blur Image Quality Measure
DI:	Detail Metric
CR:	Compression Ratio
RGB:	Red Green Blue
YCbCr:	(Luminance - Chrominance)
HSV:	(Hue Saturation and Value)
DCT:	Discrete Cosine Transformation
DWT:	Discrete Wavelet Transform
Bior:	Biorthogonal
Db:	Daubechies
Coif:	Coiflets
LWT:	Lifting Haar wavelet transform
EBCOT:	Embedded Block Coding with Optimized Truncation
RLE:	Run-Length Encoding
LZW:	Lempel-ZivWelch
BMP:	Bitmap Windows
PNG:	Portable Network Graphics
TIFF:	Tagged Image File Format

General Introduction

JPEG 2000 (JP2) is an image compression standard and coding system. It was created by the Joint Photographic Experts Group committee in 2000 with the intention of superseding their original JPEG standard (created in 1992). It is used in many applications and standards such as JPIP protocol for streaming, and Motion JPEG 2000 (MJ2). When we compress an image with JPEG2000 in paint for example, we cannot control the parameters of the JPEG2000, the color space, the quantization step, the used wavelet and resolution level, so default values are considered and there are used independently from the image. The question is: Are these parameters provide the best performance in terms of compression ratio and image quality?

The purpose of our project is to optimize JPEG2000 algorithm by finding the optimal color space, optimal wavelet, and optimal resolution level. Our project is composed of three chapters:

Chapter I is an introduction to image compression. This chapter is divided into five parts; the first part is an introduction to the chapter, the second part is an overview of data compression, the third part explains the quality metrics used in our study, the fourth parts talk about the lossy and lossless image compression, the last part is about JPEG compression standard.

Chapter II is a study of JPEG2000. This chapter contains three parts, the first part is an introduction to the chapter, the second part explains the structure of the JPEG2000, the last part is a state of the art (related work).

Chapter III presents and explains the considered methodology for optimizing JPEG2000 algorithm. This chapter is divided into five parts; the first part is an introduction to the chapter and an overview of the main methodology; the second part represents the methodology followed to find the optimal color space and a discussion of the obtained results; the third part is an explanation of the method used for finding the best wavelet; in the fourth part our purpose is to find the best level of wavelet transform, the last one is devoted to tests and validations of the obtained result.

Chapter I: Introduction to Image Compression

1.1 Introduction

With today's technology, everything is stored in almost every field, engineering, astronomy, finance…etc, in order to be processed and digitally transmitted. There is a tendency among governments to digitize all documents, Algeria as an example is trying this year to digitize all students' folders in the education sector. The aim is to create a national database for all its students, and to be electronically accessible. The amount of information is going to be huge. Methods of compressing this tremendous information are of significant importance both academically and industrially.

Image compression is at the heart of this endeavor, because images and videos are the most likely to take most of the storage capacity of devices. Interest in image compression dates back more than 35 years. Basically, the field has undergone significant growth through the practical application of the theoretic work of C. E. Shannon and others [1] first formulated the probabilistic view of information and its representation, transmission, and compression.

1.2 Data Compression

In brief, "Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and hence communication costs. Reducing the storage requirement is equivalent to increasing the capacity of the storage medium and hence communication bandwidth" [2].

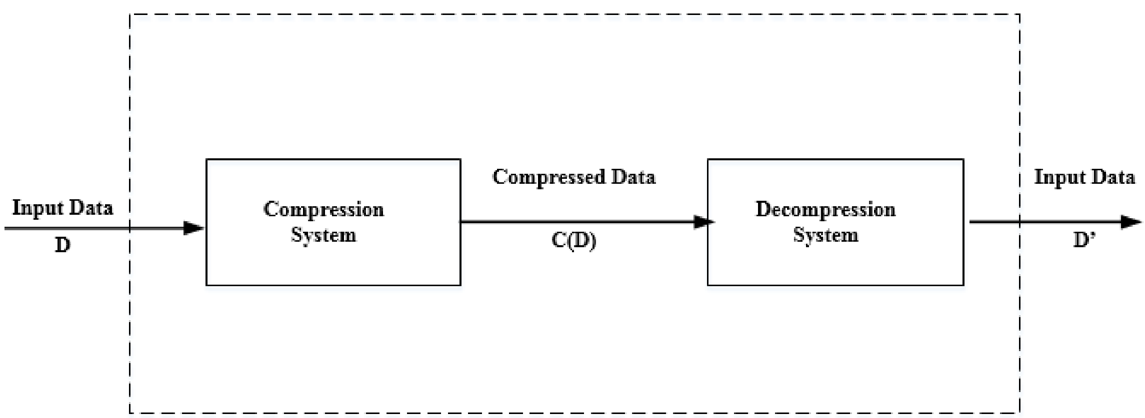


Figure 1. 1: Data Compression System [2]

There exist three types of redundancies, *coding redundancy,* which exist when less than optimal code words are used, *interpixel redundancy,* which result from correlations between pixels of an image and finally, *psychovisual redundancy,* which is due to data that is ignored by the human eyes (i.e., visually nonessential information) [3]. Figure 1.1 depicts a simple diagram of a compression system. *Coding* and *decoding* are the information theory terms of compression and decompression, Shannon's work has tremendously changed the communication realm in general, and image processing in specific.

Before we pass to JPEG2000 which is our main interest, we need to state and clarify some notions and concepts that will be used in the next chapters. For the sake of simplicity in explanation, we are going to assume gray-level images.

1.2.1 Coding Redundancy [3]

Let n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information, compression ratio C_R is defined as:

$$C_R = \frac{n_1}{n_2} \tag{1. 1}$$

A practical compression ratio, such as 10, means that the first data set has 10 information carrying units (say, bits) for every 1 unit in the second or compressed data set.

Let us assume, once again, that a discrete random variable r_k in the interval [0,1] represents the gray levels of an image and that each r_k occurs with probability $p_k(r_k)$:

$$p_r(r_k) = \frac{n_k}{n}, \quad k = 0,1,2,3 \ldots.,L-1 \tag{1. 2}$$

where L is the number of gray levels, n_k is the number of times that the k^{th} gray level appears in the image, and n is the total number of pixels in the image. If the number of bits used to represent each value of r_k is $l(r_k)$, then the average number of bits required to represent each pixel is:

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k)p_r(r_k) \tag{1. 3}$$

Thus the total number of bits required to code an M X N image is MNL_{avg}. [3].

Assigning fewer bits to the more probable gray levels than to the less probable ones achieves data compression. This process commonly referred to as *variable-length coding* which we will talk about in details later. In most images, certain gray levels are more probable than others (that is. the histograms of most images are not uniform). A natural binary coding of their gray levels assigns the same number of bits to both the most and least probable values. Thus failing to minimize the above equation and resulting in coding redundancy.

1.2.2 lnterpixel Redundancy [3]

The value of any given pixel can be reasonably predicted from the value of its neighbors, so the information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel to an image is redundant; it could have been guessed on the basis of the values of its neighbors. A variety of names including *spatial redundancy, geometric redundancy,* and *interframe redundancy,* have been given to refer to these interpixel dependencies. The term *interpixel redundancy* is used to encompass them all.

In order to reduce the interpixel redundancies or *runs* of identical intensities which are often called *run-length pairs* in an image, the 2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient (but usually "nonvisual") format. For example, the differences between adjacent pixels can be used to represent an image. Transformations of this type (that is, those that remove interpixel redundancy) are referred to as *mappings.* They are called *reversible mappings* if the original image elements can be reconstructed from the transformed data set.

1.2.3 Psychovisual Redundancy [3]

Certain information simply has less relative importance than other information in normal visual processing. This information is said to be *psychovisually redundant.* It can be eliminated without significantly impairing the quality of image perception.

Psychovisual redundancy is fundamentally different from the redundancies discussed earlier. Unlike coding and interpixel redundancy, psychovisual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing. Since the elimination of psychovisually redundant data results in a loss of quantitative information, it is commonly referred to as *quantization.*

1.2.4 Frequency Domain Coding

Frequency domain coding is essentially a “decorrelation” process between data, if for example a color in an image has a high probability, then its neighbor will most likely have a similar color, this is called “correlation”. Eliminating correlation between the pixels allow more efficient entropy coding, which we will talk about later. Frequency domain coding will allow us to know the distortion perceived by the image viewer which can be used to improve coder performance. For example, the low frequency elements from a continuous tone image are more important than the high frequency elements, so the quantization step for the high frequency coefficients can be larger. JPEG and JPEG2000 standards (which will be discussed in details in this and next chapter) they use the Discrete Cosine Transformation (DCT) and the Discrete Wavelet Transformation (DWT), which are frequency domain coding methods. These transformations decompose the two-dimensional pixel values from the image into basis signals and produce the coefficients of these basis signals as the outputs [4].

1.2.5 Quantization

In general, “the quantizer is a nonlinear system whose purpose is to transform the input sample x[n] into one of a finite set of prescribed values $\hat{x}[n]$. The operation is represented by ” [5] :

$$\hat{x}[n] = Q(x[n]) \tag{1. 4}$$

There are many types of quantizers, but they basically devise into *uniform* & *nonuniform* quantizers. Uniform quantizers are the ones that have equally spaced transition and reconstruction leves, whereas nonuniform quantizers does not have equally spaced reconstruction and transition levels, but rather dynamic quantization steps according to the histogram of the image.

Each compression standard uses a certain quantization algorithm including JPEG and JPEG2000 compression standards.

1.2.6 Entropy

“Entropy coding is a compression technique that uses the knowledge of the probabilities of all the possible data/symbols within the source image file. If a shorter codeword is assigned to a frequently occurring symbol instead of a rare symbol, the compressed file size will be smaller” [4]. Entropy H(z) or uncertainty is defined as [1]:

$$H(\boldsymbol{z}) = -\sum_{j=1}^{J} P(a_j) logP(a_j) \tag{1. 5}$$

Where z is a discrete random variable with possible values {a1, …, aJ}, $a_j$is the source symbol, $P(a_j)$ is the probability of the occurrence of the symbol.

Choice, uncertainty or entropy, as Shannon defined as a second theorem in his paper [1], is an important concept, and tool to remove redundancy from an image effectively, hence achieve maximum compression, and each image standard had a specific entropy method. For example JPEG2000 uses the so called Embedded Block Coding with Optimized Truncation (EBCOT) entropy algorithm, which we will talk about in detail later.

1.3 Color Spaces

A color space is a method by which we can specify, create, and visualize color. As humans, we may define a color by its attributes of brightness, hue, and color fulness. A computer may describe a color using the amounts of red, green, and blue phosphor emission required to match a color. A color is thus usually specified using three co-ordinates, or parameters. These parameters describe the position of the color within the color space being used. They do not tell us what the color is, that depends on what color space is being used. Three color spaces are commonly used.

RGB (Red Green Blue) this is an additive color system based on tri-chromatic theory. Often found in systems that use a CRT to display images. RGB is easy to implement but non–linear with visual perception. It is device dependent and specification of colors is semi–intuitive. RGB is very common, being used in virtually every computer system as well as television, video etc.

YCbCr (Luminance - Chrominance) this color space separates RGB into luminance and chrominance information. It is useful in compression applications (both digital and analogue). This space is device dependent but is intended for use under strictly defined conditions within closed systems.

YCbCr is a reversible transform of RGB and are computed by equations (1.6) and (1.7):

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} . \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{1. 6}$$

$$\begin{pmatrix} R \\ \mathrm{G} \\ \mathrm{B} \end{pmatrix} = \begin{pmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & -0.71414 \\ 1.0 & 1.772 & 0 \end{pmatrix} . \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} \tag{1. 7}$$

HSV (Hue Saturation and Value) is a common cylindrical-coordinate representations of points in an RGB color model. The two representations rearrange the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the cartesian (cube) representation.

In each cylinder, the angle around the central vertical axis corresponds to "hue", the distance from the axis corresponds to "saturation", and the distance along the axis corresponds to "lightness", "value" or "brightness".

HSV is a reversible transform of RGB computed as follow:

$$\mathrm{S} = \frac{max(RGB) - min(RGB)}{max(RGB)} \tag{1. 8}$$

$$V = max(RGB) \tag{1. 9}$$

The Hue, H, is then calculated as follows. First calculate R'G'B':

$$\mathrm{R'} = \frac{max(RGB) - R}{max(RGB) - min(RGB)} \tag{1. 10}$$

$$G' = \frac{max(RGB) - G}{max(RGB) - min(RGB)} \quad (1.\ 11)$$

$$B' = \frac{max(RGB) - B}{max(RGB) - min(RGB)} \quad (1.\ 12)$$

If saturation S is zero then Hue is undefined (i.e. the color has no Hue therefore it is monochrome) otherwise:

then, if R = max and G = min	H = 5 + B’
else if R = max and G ≠ min	H = 1 – G’
else if G = max and B = min	H = R’ + 1
else if G = max and B ≠ min	H = 3 – B’
else if R = max	H = 3 + G’
otherwise	H = 5 – R’

To convert back from HSV to RGB first take Hue, H, in the range 0 to 360 and divide by 60:

$$HEX = \frac{H}{60} \quad (1.\ 13)$$

Then the values of Primary Color (PC), Secondary Color (SC), *a*, *b* and *c* are calculated. The primary color is the integer component of Hex (e.g. in MATLAB floor(Hex));

$$SC = Hex - PC \quad (1.\ 14)$$

$$a = (1 - S) \times V \quad (1.\ 15)$$

$$b = (1 - (S \times SC)) \times V \quad (1.\ 16)$$

$$c = (1 - (S\,(1 - SC)))V \quad (1.\ 17)$$

Finally we calculate RGB as follows:

if primary colour = 0 then	R = V,G = c,B = a
if primary colour = 1 then	R = b,G = V,B = a
if primary colour = 2 then	R = a,G = V,B = c
if primary colour = 3 then	R = a,G = b,B = V
if primary colour = 4 then	R = c,G = a,B = V

if primary colour = 5 then R = V,G = a,B = b

1.4 Quality Metrics

In order to evaluate the quality of compressed images, quality metrics are required. The quality or fidelity metric is an important criterion for comparing the original image and the compressed one. Two general classes of metrics exist: *objective quality metrics* where the original input or compressed image is used to evaluate the loss and *subjective quality metrics* where a subjective entity is taken under consideration like the human eye. A subjective metric is the most faithful since the human visual system is the final judge of the image quality. However, it is not real-time so not useful for real application. In these situations, objective metrics can be used. Let's present some objective metrics.

1.4.1 Root Mean Square Error (RMSE)

Let $f(x,y)$ represents an input image and let $\hat{f}(x,y)$ denotes an estimate or approximation of $f(x,y)$that results from compressing and subsequently decompressing the input. For any value of x and y, the error between $f$and $\hat{f}$ at each position *(x, y)* can be defined as:

$$e(x,y) = \hat{f}(x,y) - f(x,y) \tag{1. 18}$$

So that the total error between the two images of size M * N is:

$$e = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y) - f(x,y)\right]}{M \times N} \tag{1. 19}$$

Hence the root-mean-square error would be

$$RMSE = \frac{\left[\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y) - f(x,y)\right]^2\right]^{\frac{1}{2}}}{M \times N} \tag{1. 20}$$

The RMSE represents the average of differences between two images pixel by pixel. It has the advantage of being simple and easy to implement.

1.4.2 Peak Signal-to-Noise Ratio (PSNR)

A closely related objective fidelity criterion is the peak signal to-noise ratio of the compressed-decompressed image. The *PSNR* of the output image, denoted *SNR* is given by the following formula:

$$\text{SNR} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{RMSE} \tag{1. 21}$$

The PSNR is simply defined as:

$$PSNR = 10\ \log10(SNR) \tag{1. 22}$$

For a 8-bit gray-level image, the corresponding PSNR in dB would be:

$$PSNR = 10\ \log10(\frac{255}{RMSE}) \tag{1. 23}$$

PSNR is widely used as a quality score. It is the first and most known choice as a quality metric, and it is usually used instead of RMSE. In general a PSNR value greater than 30 informs about a good quality.

1.4.3 The Structural Similarity Index Measurement (SSIM), the Mean SSIM (MSSIM) and the Multi-Scale-SSIM (MS-SSIM)

Due to the limitations that the *objective quality metrics* offer, a new approach was proposed by Wang *et al.* [6]. They developed the new Structural Similarity (SSIM). It is "the measure of structural changes in image (luminance, contrast, and other errors)" [7]. For an input image *x* and output image *y SSIM* is defined as:

$$SSIM(x,y) \ = \ [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{1. 24}$$

Where $\alpha = \beta = \gamma = 1$, *x* and *y* are two discrete non-negative signals that have been aligned with each other, $l(x,y)$ is the luminance, $c(x,y)$ is the contract and $s(x,y)$ is the structural components which are defined as [6]:

$$l(x,y) = \ \frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1} \tag{1. 25}$$

$$c(x,y) = \ \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2} \tag{1. 26}$$

$$s(x,y) = \ \frac{\sigma_{xy}+C_3}{\sigma_x^2\sigma_y^2+C_3} \tag{1. 27}$$

where μ_x and μ_y are the mean of the original and compressed images, σ_x and σ_y are the standard deviations, , σ_x^2 and σ_y^2 are the variances, and σ_{xy} is the covariance of the two images. The constants C_1, $C_{2,}$ and C_3 are presented to "avoid instability when the denominator are without them are close to zero" [6] and are defined as [6] :

$$C_1 = (K_1 L)^2 \tag{1. 28}$$

$$C_2 = (K_2 L)^2 \tag{1. 29}$$

$$C_3 = {}^{C_2}/_{2} \tag{1. 30}$$

Where K_1 and K_2 are small constants << 1, and L is the dynamic range of the pixel values (which is 255 for 8-bit gray scale images).

Another related quality metric is used to evaluate the overall image quality. It is called the Mean Structural Similarity (MSSIM), which is defined as :

$$MSSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{N} SSIM(x_j, y_j) \tag{1. 31}$$

Where X and Y are the reference and the outputted images and N is the number of local windows of the image.

One of the advantages of the SSIM or (MSSIM) over PSNR is that it represents human visually system more accurately.

Equation (1.12) is considered as a single-scale method (single resolution). In order to "incorporate image details at different resolution" [8] Zhou Wang *et al.* [8] proposed yet another method for an image at different resolutions called *multi-scale SSIM*, the original image is indexed as scale 1, and the highest scale as M, its overall equation is defined as [8]:

$$MS - SSIM(x,y) = [l_m(x,y)]^{\alpha_M} . \prod_{j=1}^{M} [c_j(x,y)]^{\beta_j} \, [s_j(x,y)]^{\gamma_j} \tag{1. 32}$$

Where $\mathbf{x} = \{x_i | i = 1, 2, \cdots, N\}$ and $\mathbf{y} = \{y_i | i = 1, 2, \cdots, N\}$ are two image patches extracted from the same spatial location from two images being compared, $c_j(x,y)$ and $s_j(x,y)$ denote the contrast and the structure comparison respectively, at the j^{th} scale, $l_m(x,y)$ denotes the luminance comparison which is computed only at the last scale M. The coefficients α_M, β_j, and γ_j are "used to adjust the relative importance of different components" [8] which are similar to the ones in equation (1.12). As can be observed, equation (1.12) is considered as a special case of the more general case in equation (1.20).

We have chosen MS-SSIM to compare our results because it offers more flexibility than the normal SSIM, incorporates the variations of different viewing conditions and situations and relatively calibrates the different parameters [8]. One of its drawbacks is that at high scales, the single scale SSIM achieves better measurements, because JPEG and JPEG2000 "usually compress fine-scale details to a much higher degree than coarse-scale structure" [8], and the approach to calibrate the parameters is as Wang *et al* [8] described "crude". Hence, SSIM (or MSSIM) and MS-SSIM are important quality metric to evaluates the performance of compressed images using the JPEG2000 compression algorithm.

1.4.4 The Blur Image Quality Measure (IQA) or (QM)

The blur image quality metric is developed by F.Kerouh *et al.* [9] based on the idea of the exploitation of the edge width through a multi-resolution analysis using wavelet transform. It is defined as follows [9]:

$$IQA = 1 - \frac{\sum_{i=1}^{i=3} 2^{3-i} \times Q_i}{\sum_{i=1}^{i=3} 2^{3-i}} \qquad (1.\ 33)$$

Where the quality ratio Q_i represents the blur quantity in edges at each resolution level *I* and is defined as:

$$Q_i = \frac{NB_i}{NQ_i} \qquad (1.\ 34)$$

Where NQ_i & NB_i represent the number of edge pixels and the blured ones. The method takes into account all edge pixels detected by wavelet transform at each resolution level. IQA is normalized between 0 and 1, it attends almost 1 if the image is clear and decreases proportionally to blur amount.

1.5 Image Compression Models

There exist many standards, algorithms, and models for compressing images based primarily on eliminating the redundancies defined previously. They can be classified into two main categories Lossless and Lossy.

Lossless Compression which is, as the name suggests, compresses the image without lossing information. Most of the eliminated redundancies in lossless compression are coding redundancies. Some of the standards that uses lossless compression coding are *Windows Bitmap (BMP), Portable Network Graphics (PNG), Tagged Image File Format (TIFF) …etc* [3].

Lossy Compression which, as opposed to the lossless, compresses with loss. In other words, it compromises the quality with size. Some of the well known standards are that uses lossy coding methods are *Consulative Committee of the International Telephone and Telegraph (CCITT), Joint Photographic Experts Group (JPEG), JPEG2000…etc* [3]. Figure 1.2 depicts an overview of the most popular image standards, formats and algorithms. Figure 1.3 shows a general block diagram for compressing images that almost any standard follows which is essentially devised into two parts *encoding* and *decoding.*

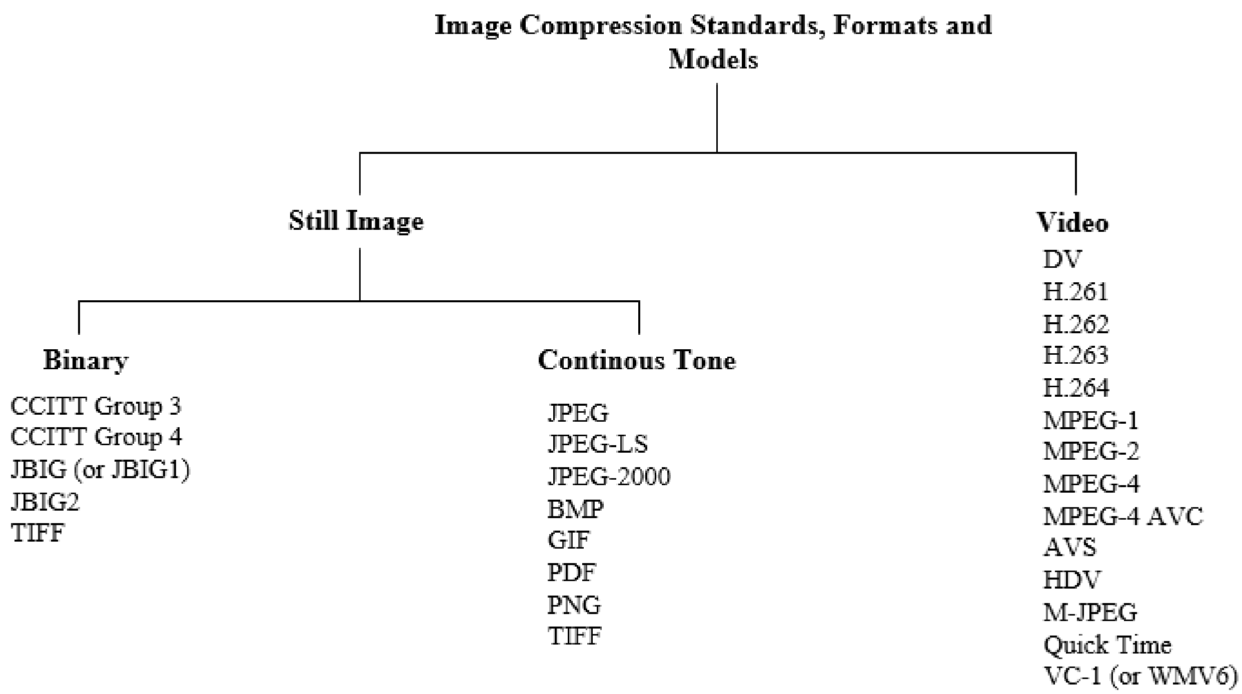


Figure 1. 2: Some popular Image compression Standards [3]

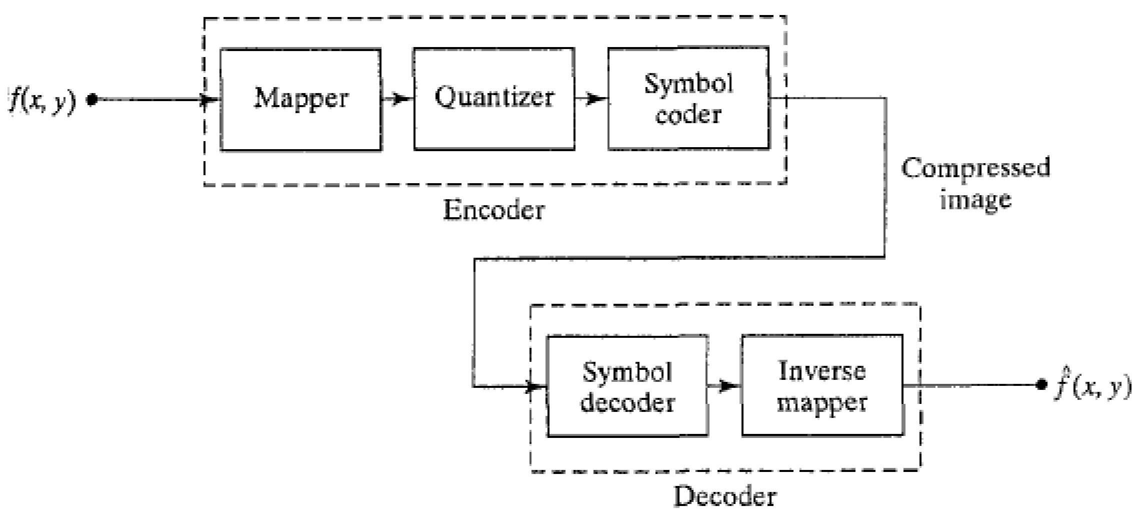


Figure 1. 3:A general image compression system block diagram [2]

1.5.1 Lossless Compression

There are many compression techniques for which the compression quality is perfect, no lost data. In this section we are going to explore only the most famous ones and leave the others for further reading and research.

1.5.1.1 Run-Length Encoding (RLE)

As we have seen in section 1.2.2, interpixel redundancy can be eliminated to achieve compression, and RLE is considered as one of simplest and oldest (it goes back to 1950 [3]) lossless compression technique to eliminated such a redundancy. The basic idea is of RLE is to code each *run* (group of repeated elements) encountered in a raw by its length and establish a way to determine those encoded runs (as decoding).

"For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. A hypothetical scan line, with B representing a black pixel and W representing white, might read as follows:

WWWWWWWWWWWWBWWWWWWWWW
WWWBBBWWWWWWWWWWWWWWW
WWWWWWWWWWBWWWWWWWWWW
WWWW

With a run-length encoding (RLE) data compression algorithm applied on the above hypothetical scanline, can be rendered as follows: 12W1B12W3B24W1B14W" [10].
RLE was used in the CCITT group 3 and 4, in facsimile (FAX) and also BMP uses RLE as a coding method. It was also used by an early black and white graphics file format supported by CompuServe [11]**.** Although RLE achieves a good compression, but further compression can be obtained using Variable length coding or Huffman procedure.

1.5.1.2 Bit-Plan Coding

It is a coding technique based on the idea of decomposing a multilevel image into a series of binary images and compressing each binary image with a known compression method as RLE. There are two approaches of decomposing the image, which we will not cover in our work, we refer the reader to the reference [3] for more comprehensive explanation and good examples.

1.5.1.3 Variable-Length coding (Huffman Coding)

"The simplest approach to lossless image compression is to reduce only coding redundancy. One method of doing so is using Huffman Coding algorithm. When coding the symbols of an information source individually, Huffman coding yields the smallest possible number of code symbols per source symbol the resulting code is optimal for a fixed value of *n*, subject to the constraint that the source symbols are coded *one at a time.*

The first step in Huffman's approach is to create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction.

The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to the original source. Figure 3 tabulates the steps with an example.” [3]

Original Source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Figure 1. 4: Huffman Code Assignement Procedure [3]

1.5.1.4 Arithmetic Coding

Unlike the variable-length codes described previously, *arithmetic coding* generates non-block codes. In arithmetic coding, a one-to-one correspondence between source symbols and code words does not exist. Instead, an entire sequence of source symbols (or message) is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between 0 and 1. As the number of symbols in the message increases. The interval used to represent it becomes smaller and the number of information units (say, bits) required to represent the interval becomes larger. Each symbol of the message reduces the size of the interval in accordance with its probability of occurrence. Figure 1.5 shows arithmetic coding example [3].

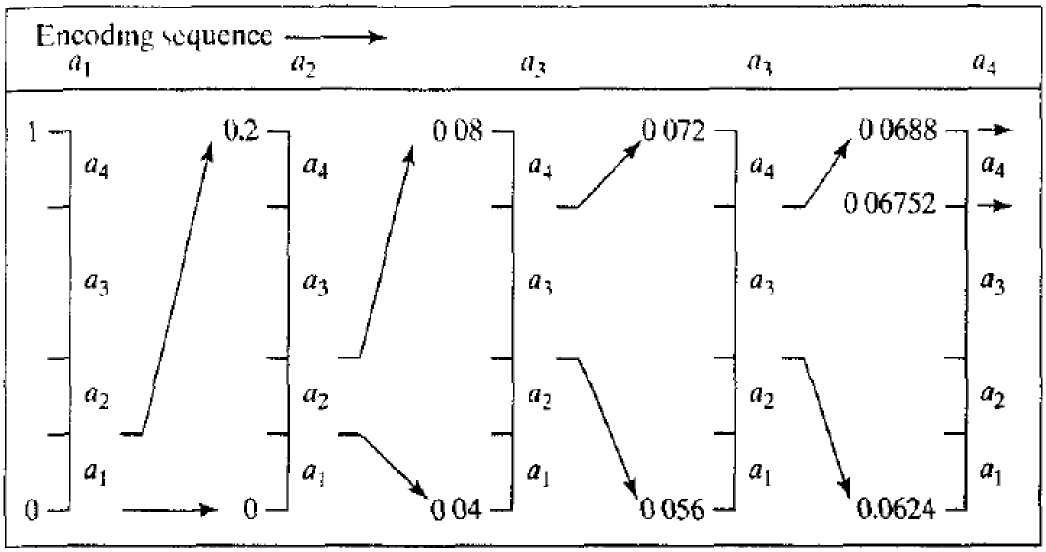


Figure 1. 5: Arithmetic Coding Process [3]

1.5.1.5 LZW Coding

Having examined the principal methods for removing coding redundancy, we now consider one of several error-free compression techniques that also attack an image's interpixel redundancies. The technique, called Lempel-ZivWelch (LZW) coding, assigns fixed-length code words to variable length sequences of source symbols but requires no *a priori* knowledge of the probability of occurrence of the symbols to be encoded. LZW coding is conceptually very simple. At the onset of the coding process, a codebook or "dictionary" containing the source symbols to be coded is constructed. As the encoder sequentially examines the image's pixels, gray-level sequences that are not in the dictionary are placed in algorithmically determined (e.g., the next unused) locations. If the first two pixels of the image are white the address following the locations reserved. The next time that two consecutive white pixels are encountered, the address of the location containing sequence is used to represent them. If it is too small, the detection of matching gray-level sequences will be less likely if it is too large. The size of the code words will adversely affect compression performance. A unique feature of the LZW coding just demonstrated is that the coding dictionary or codebook is created while the data are being encoded. For more examples and comprehensive explanation, the reader is referred to **[3]**.

Example [12]: let x be x=00101001000100101001101 01

i	log i	J=j	W	a	DICT[i]
1	0	empty	empty	0	0
2	1	$(1)_2=1$	0	1	01
3	2	$(10)_2=2$	01	0	010
4	2	$(11)_2=3$	010	0	0100

5	3	$(100)_2=4$	0100	1	01001
6	3	$(101)_2=5$	01001	1	010011
7	3	$(011)_2=3$	010	1	0101

Code of x: 0 (1,1) (10,0) (11,0) (100,1) (101,1) (011,1)
Code of x in a compact form (the true form): 0111001101001101101111.

There exist a lot of other lossless coding techniques like *Bit-Plane Coding, constant area coding, Lossles predictive coding…etc* which we are not going to cover in this project. The reader has the choice to do further reading on those methods, a good source would be [3]. Basing on the presented methods, different standards exists (BMP, PNG, TIFF….).

1.5.1.6 Bitmap Windows (BMP)

BMP is a lossless image standard developed by Microsoft company. It supports images with 1 to 32 bits per pixel. BMP uses simple run-length encoding for 4 and 8 bits per pixel. However, BMP compression is only useful with large monochrome blocks, making it of limited potential [13].

1.5.1.7 Portable Network Graphics (PNG)

PNG is the most used file format in the internet, generally it uses LZW lossless algorithm as an encoder. It was first developed by Thomas Boutell [13] in 1996. The PNG format supports five different color types or methods for representing the color of pixels within an image, RGB Tripple , Palette, Grayscale, RGB with alpha channel and Grayscale with alpha channel.

1.5.1.8 Tagged Image File Format TIFF

It is a flexible file format supporting a variety of image compression standards including JPEG, JPEG2000…etc. TIFF supports LZW compression algorithm and it was first developed by Aldus company [3].

1.5.2 Lossy Compression

Unlike the lossless methods, the lossy compression method is based on the previously mentioned idea of compromising the accuracy of the image in exchange for increased compression. The outputted image which definitely distorted but this distortion may not be visual it must be tolerated in order to obtain a good image quality. In this section we are going to explore some lossy encoding techniques that are most famous in the compression realm.

1.5.2.1 Lossy Predictive Coding

Between successive pixels, there exists a mutual redundancy. *Predictive coding* is used to remove the mutual redundancy. Based on a sequence of reproduced pixels with an estimation rule. A pixel value can be predicted to replace the current pixel. This is the principle of predictive coding. If the current reproduced pixel is taken as the sum of the predicted pixel value and the quantized error value between the current pixel and the predicted pixel, the prediction method is called D*ifferential pulse code modulation (DPCM).* Since the prediction exploits the local correlation among the neighboring pixels, the predicted error can be encoded in fewer bits comparing to encoding the current pixel directly. Thus, in DPCM, rather than directly encoding the current pixel, the predicted error is encoded. The fact that DPCM utilizes the local correlation to reduce the redundancy in an image gives it an advantage over directly encoding the current pixel, which does not make use of the local correlation. Figure 1.6 shows a block diagram summarizing DPCM coding [14].

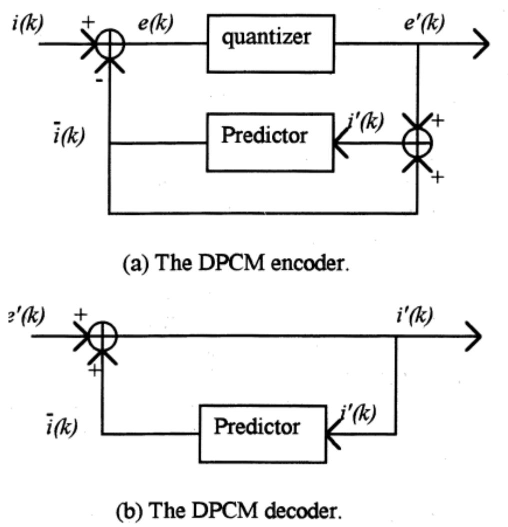


Figure 1. 6: Block diagram of DPCM coding [14]

1.5.2.2 Transform Coding

Separately encoding, each pixel is inefficient since it does not use the substantial correlation among pixels in a pixel block of a digital image. *Transform coding* is an effective way to encode a block of pixels through performing a linear transformation on these pixels and encoding the transform coefficients obtained from the transformation. The idea is that a suitable transformation produces fewer correlated transform coefficients than the original pixels, and the information may be concentrated into fewer transform coefficients. Thus, fewer bits can be used in the encoding process. Another factor supporting transform coding is that the human visual system has perceptual masking effects so that some frequency components are not as sensitive to encoding errors as others. By allocating fewer bits for those "masked" frequency coefficients, transform coding can produce an image that is perceived to be of superior quality [14].

1.5.2.3 Subband Coding

Subband coding is a coding method based on decomposition of the source representation in the frequency domain into relatively narrow subbands, source coding without this frequency band decomposition is sometimes called full band coding. In subband coding, since each of these bands has its own statistics, subjectively superior performance can be achieved over full band coding through an appropriate bit allocation strategy among the subbands. In the other words, subband coding does better resource allocation than full band coding. In the case of image compression, subband coding puts more effort into the frequency bands where image activity is apparent by allocating more bits to these bands.

A popular approach for subband coding is to use *linear phase quadrature mirror filters (QMF's)* to divide the full band into subbands. When channel noise and quantization noise are absent, using QMF's can result in near perfect reconstruction of the input signal without aliasing. There are four steps to design a subband coder:

1. Design QMF's,
2. Allocate bits among the subbands,
3. Design a quantizer for each subband, and
4. Design an entropy coder for each subband [14].

1.5.3 JPEG Image Compression Standard

It is a well known ISO/ITU-T standard created in the late 1980s. There are several modes defined in JPEG, however, here we refer to the most used namely baseline algorithm. The baseline mode is the most popular one and supports lossy coding only. Figure 1.7 shows the different steps for the JPEG baseline algorithm. In the baseline mode, the image is divided in 8x8 blocks and each of these is transformed with the DCT which is defined as

$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} S_{xy} \cos(\frac{(2x+1)u\pi}{16}) \cos(\frac{(2y+1)v\pi}{16}) \qquad (1.\ 35)$$

And the 8x8 Inverse Discrete Cosine Transform (IDCT) is defined by:

$$S_{yx} = \frac{1}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} C_u C_v S_{vu} \cos(\frac{(2x+1)u\pi}{16}) \cos(\frac{(2y+1)v\pi}{16}) \qquad (1.\ 36)$$

Where:

$$C_m = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } m = 0 \\ 1 & \text{otherwise} \end{cases} \qquad (1.\ 37)$$

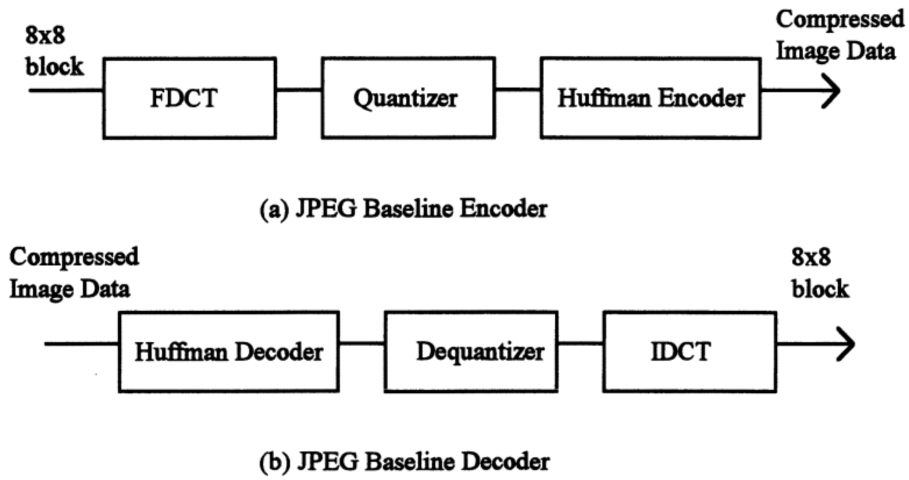


Figure 1.7: ***JPEG Baseline System Diagram [14].***

In the quantization step each of the 64 DCT coefficients is divided by a user-selectable quantizer step-size parameter and rounded to the nearest integer. This is a normalization process. The 64 quantizer step-size parameters form a matrix called the *Quantization Matrix* or *QM* (It should not be confused with the quality metric (QM) mentioned in section 1.4.4). The JPEG standard does not specify the QM. It up to the user to define the QM.

The quantization process introduces the distortion that is apparent in the decoded image. The magnitude of the distortion can be varied by changing the quantizer step-size parameters of the QM. This is because an error within the range of one half of a quantizer step-size is introduced during the normalization process. The larger the quantizer step-size, the greater the distortion; the greater the distortion, the smaller the amount of compressed data. This is a typical rate-distortion relationship for lossy compression [15] [14] [3].

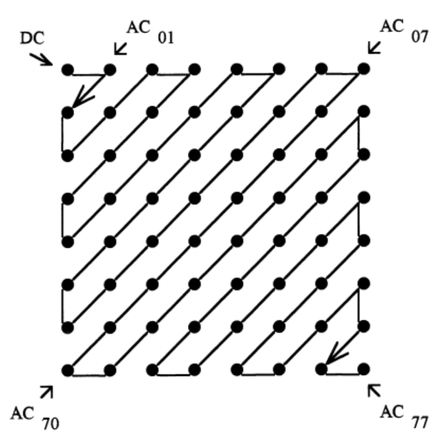


Figure 1.8: Zig-zag ordering. [14]

1.6 Conclusion

In this chapter we have defined the fundamental concepts of image compression, the different methods used to code to decode images, both lossless and lossy methods. We have defined some quality metric indexes that are going to be used in chapter three, so the reader would not be confused with the new concepts if new to the domain. At the end we have given a brief overview of the JPEG standard and the different steps involved in the algorithms to make it easy to see the development from JPEG to JPEG2000. In the next chapter we are going to give a brief study of the JPEG2000 standard along with the different works done to optimize and analyze the standard.

Chapter II: Study of The JPEG2000 Compression Standard

2.1 Introduction

JPEG2000 is a wavelet-based standard for the compression of digital images. It was developed by the ISO JPEG committee to improve on the performance of JPEG while adding significant new features and capabilities to enable new imaging applications. The JPEG2000 compression algorithm is part of a multi-part standard that defines compression architecture, file format family, client-server protocol and other components for advanced applications. Instead of replacing JPEG, JPEG200 has created new opportunities in geospatial and medical imaging, digital cinema, image repositories, and networked image access. These opportunities are enabled by the JPEG2000 feature set:

- A single architecture for lossless and visually lossless image compression
- A single JPEG200 master image can supply multiple derivative images
- Progressive display, multi-resolution imaging and scalable image quality
- The ability to handle large and high-dynamic range images
- Generous metadata support.

With JPEG2000, an application can access and decode only as much of the compressed image as needed to perform the task at hand. This means a viewer, for example, can open a gigapixel image almost instantly by retrieving and decompressing a low resolution, display-sized image from the JPEG2000 code stream [16].

JPEG2000 is divided into 13 parts, since the end of the first part which is the most important, many have been added into it. The following table lists the parts and a brief description of each one.

Table 2. 1: Different parts of JPEG2000 algorithm [16]

Part	Description
1	Core Coding System, including optional file format (JP2)
2	Extensions to Part 1 algorithm, including extended file format (JPX)
3	Motion JPEG2000 file format (MJ2) for timed sequence of images, such as video
4	Conformance Testing
5	Reference Software for implementing JPEG200
6	Compound Image File Format (JPM) for paged sequence of compound images
8	Secure JPEG200 (JPSEC) for the encryption, authentication and conditional access of JPEG200 image data and metadata; an amendment due in 2008 will extend the notion from the codestream to the file formats
9	Interactivity tools, APIs and protocols (JPIP) - a client-server protocol for JPEG200 files and data
10	JPEG200 3D (JP3D) - extensions to volumetric data
11	Wireless (JPWL) - extensions for error protection and correction in wireless and error-prone networks.
12	ISO Base Media File Format - Joint JPEG-MPEG6 activity that defined a base media format, MJ2 in Part 3 is an application of it
13	An entry level JPEG200 encoder – an encoder equivalent of Part 1
14	XML Structure Representation of JPEG200 file formatsand codestreams

2.2 Structure and Algorithm of JPEG2000

As mentioned before, JPEG2000 is based on the discrete wavelet transform (DWT), scalar quantization, context modeling, arithmetic coding and post compression rate allocation. DWT is dyadic and can be performed with either a reversible or irreversible filter. The quantizer follows an embedded dead-zone scalar approach and is independent for each sub-band. Each sub-band is divided into blocks, typically 64x64, and entropy coded using context modeling and bit-plane arithmetic coding. The coded data is organized in so called layers, which are quality levels, using the post-compression rate allocation and output to the code stream in packets [15]. The different processes, algorithms and tools that enter in the JPEG2000 standard are summarized in figure 2.1.

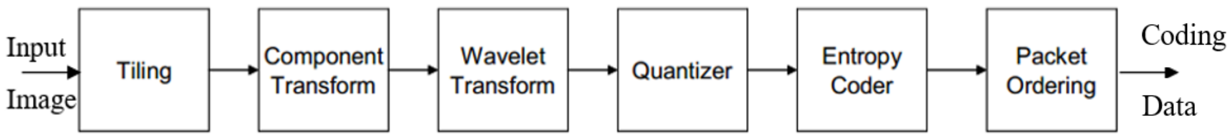


Figure 2. 1: Different steps of the JPEG2000 compression [16]

2.2.1 Preprocessing step

The first step in preprocessing is tiling. In this step, the source image is partitioned into rectangular non-overlapping tiles if the image is very large. All tiles have exactly the same dimensions. Each tile is compressed independently. Tiling is an optional process, where it is used for reduction of memory requirements and efficient extraction of a region from the image. The codec expects its input sample data to have a nominal dynamic range that is approximately centered on zero. Suppose that a particular component has P bits/sample. The samples may be either signed or unsigned, leading to a nominal dynamic range of $[-2^{P-1}; 2^{P-1} - 1]$ or $[0; 2^{P} - 1]$, respectively. If the sample values are unsigned, the nominal dynamic range is adjusted by subtracting a bias of 2^{P-1} from each of the sample values. If the sample values are signed, the nominal dynamic range is already centered on zero. By ensuring that the nominal dynamic range is centered on zero, a number of simplifying assumptions can be made in the design of the codec (e.g., with respect to context modeling, numerical overflow, etc…) [3] [17] [2].

In the encoder, the preprocessing step is followed by the component Transformations step. It is effective in reducing the correlations amongst the multiple components in a multi-component image. This results in reduction in redundancy and increase in compression performance. The JPEG2000 Part 1 standard supports two different transformations: (1) reversible color transform (RCT), and (2) irreversible color transform (ICT) [2]. ICT is used for lossy compression with the irreversible wavelet transform. The forward and the inverse ICT transformations are represented by the equations (1.6) and (1.7), respectively [3].

The reversible color transformation (RCT) is used for lossless compression with the reversible wavelet transform. The forward and inverse RCT transformations are represented by the equations (2.1) and (2.2) respectively:

$$\begin{pmatrix} Y_r \\ V_r \\ U_r \end{pmatrix} = \begin{pmatrix} [\frac{R+2G+B}{4}] \\ R-G \\ B-G \end{pmatrix} \tag{2. 1}$$

$$\begin{pmatrix} G \\ R \\ B \end{pmatrix} = \begin{pmatrix} Y_r - [\frac{U_r+V_r}{4}] \\ V_r + G \\ U_r + G \end{pmatrix} \tag{2. 2}$$

2.2.2 Discrete Wavelet Transform (DWT)

Instead of the DCT in JPEG, JPEG2000 uses DWT, after component transform, the tile-components are decomposed into different decomposition levels using discrete wavelet transform (DWT). These decomposition levels contain a number of subbands, which consist of coefficients that describe the horizontal, vertical, and diagonal spatial frequency characteristics of the original tile component. The default irreversible transform is implemented by Daubechies 9-tap/7-tap filter. The default reversible transformation is implemented by a5-tap/3-tap filter.

The standard can support two filtering modes, convolution based and lifting based. Convolution-based filtering consists in performing a series of dot products between the two filters and the signal.

The conventional convolution-based 1D DWT of is presented in figure 2.2. As shown in figure 2.2, this consists of two analysis filters, h (low-pass) and g (high-pass), followed by subsampling units. The signal *x[n]* is decomposed into the approximation (low-frequency) signal *lp[n]* and the detail (high-frequency) signal *hp[n]* . Note that in the structure of figure 2.2 the down sampling is performed after the filtering has been completed. This is clearly inefficient since, in this case, half of the calculated coefficients are redundant, and the filtering is realized at full sampling rate.

The convolution-based 1-D DWT suffers from high computational complexity and high memory utilization requirements [18].

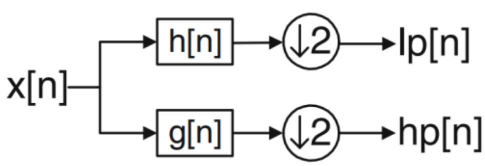


Figure 2. 2: Convolution-based implementation of the forward [18]

Lifting-based filtering consists of a sequence of very simple filtering operations for which alternately odd sample values of the signal are updated with a weighted sum of even sample values, and even sample values are updated with a weighted sum of odd sample values. The lifting-based filtering for the 5/3 analysis filter are computed by equations (2.3) and (2.4) [3] [2] [14] [7]:

$$Y(2n+1) = X(2n+1) - \left\lfloor \frac{X(2n)+X(2n+2)}{2} \right\rfloor \quad (2.3)$$

$$Y(2n) = X(2n) + \left\lfloor \frac{Y(2n-1)+Y(2n+1)+2}{4} \right\rfloor \quad (2.4)$$

Where X is the input signal and Y is the output signal. And the corresponding inverse transformation for the reversible 5/3 synthesis filter is computed by equations (2.5) & (2.6):

$$X(2n) = Y(2n) - \left\lfloor \frac{Y(2n-1)+Y(2n+1)+2}{4} \right\rfloor \quad (2.5)$$

$$X(2n+1) = Y(2n+1) - \left\lfloor \frac{X(2n)+X(2n+2)}{2} \right\rfloor \quad (2.6)$$

Where Y is the input signal, and X is the output signal. For irreversible 9/7 analysis filter, Equation 5 describes the 4"lifting" steps (STEPS 1 through 4) and the 2 "scaling" steps (STEPS 5 through 6) of the 1D filtering performed on the signal X to produce the coefficients of signal Y.

$$\begin{cases} Y(2n+1) \leftarrow X(2n+1) + (\alpha \times [X(2n) + X(2n+2)]) & step\ 1 \\ Y(2n) \leftarrow X(2n) + (\beta \times [Y(2n-1) + Y(2n+1)]) & step\ 2 \\ Y(2n+1) \leftarrow Y(2n+1) + (\gamma \times [Y(2n) + Y(2n+2)]) & step\ 3 \\ Y(2n) \leftarrow Y(2n) + (\delta \times [Y(2n-1) + Y(2n+1)]) & step\ 4 \\ Y(2n+1) \leftarrow -K \times Y(2n+1) & step\ 5 \\ Y(2n) \leftarrow \left(\frac{1}{K}\right) \times Y(2n) & step\ 6 \end{cases} \quad (2.7)$$

Where the values of the parameters are:

$$\begin{cases} \alpha = -1.586134342 \\ \beta = -0.052980118 \\ \gamma = 0.882911075 \\ \delta = 0.443506852 \end{cases} \quad (2.8)$$

And the scaling factor K is set to 1.230174105. Also, the irreversible inverse transformation of the Daubechies 9/7 synthesis filter is computed by equation (2.11)

$$
\begin{cases}
X(2n) \leftarrow K \times Y(2n) & step\ 1 \\
X(2n+1) \leftarrow -\frac{1}{K} \times Y(2n+1) & step\ 2 \\
X(2n) \leftarrow X(2n) - (\delta \times [X(2n-1) + X(2n+1)]) & step\ 3 \\
X(2n+1) \leftarrow X(2n+1) - (\gamma \times [X(2n) + X(2n+2)]) & step\ 4 \\
X(2n) \leftarrow X(2n) + (\beta \times [X(2n-1) + X(2n+1)]) & step\ 5 \\
X(2n+1) \leftarrow X(2n+1) - (\alpha \times [X(2n) + X(2n+2)]) & step\ 6
\end{cases}
\tag{2. 9}
$$

Where the values of the parameters and the scaling factor K are the same as used in equations (2.3) and (2.4). The lifting steps corresponding to the (5/3) filter-bank and (9/7) filter- bank are shown in Figure 2.2

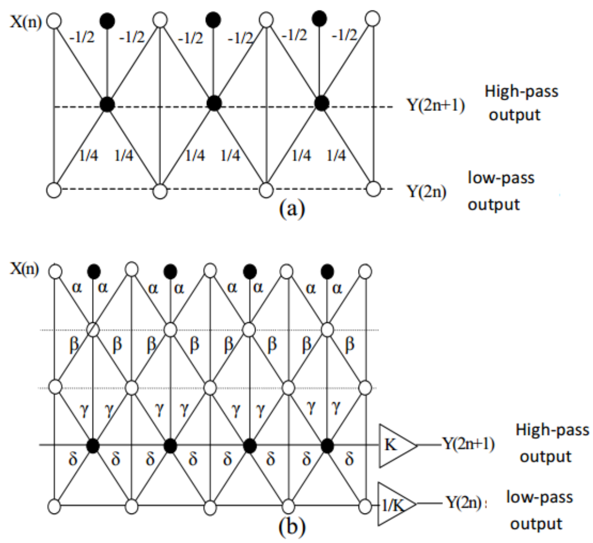


Figure 2. 3: Lifting prediction /update steps:(a).for the (5/3) filter-bank[32].

2.2.3 Quantization

The transform coefficients are quantized with a uniform scalar quantization with dead-zone. Quantization is the process by which the coefficients are reduced in precision. Each of the transform coefficients $c_b(u, v)$ of the subband b is quantized to the value $q_b(u, v)$ according to the equation (2.12):

$$
q_b(u,v) = signe\ (c_b(u,v)) \left[\frac{|c_b(u,v)|}{\Delta_b}\right] \tag{2. 10}
$$

For the reversible transform (5/3 filter), the quantization step size Δ_b for each subband b is required to be one, which means no lossy quantization performed. However, for the irreversible transform (9/7 filter), the quantization step sizes for all subbands are retrieved from the following equation (2.13):

$$\Delta_b = \left(1 + \frac{\mu_b}{2^{11}}\right).2^{Rb-\varepsilon b} \tag{2. 11}$$

It is represented relative to the nominal dynamic range R_b, the exponent coefficient ε_b, and the mantissa coefficient μ_b of each subband b.

2.2.4 Entropy Coding

The entropy coding and generation of compressed bitstream in JPEG2000 is divided into two coding steps: Tier-1 and Tier-2coding

- Tier-1 (EBCOT coder and BinaryArithmetic Coding-MQ-Coder) .
- Tier-2 (Organization of the bit-stream).

2.2.4.1 Tier 1 Coding

After quantization, each subband is divided into rectangular blocks, called code-blocks (see figure 2.3). These code-blocks are encoded independently. The code-block is decomposed into P bit-planes and they are encoded from the most significant bit-plane to the least significant bit-plane sequentially (see figure 2.4). Each bit-plane is first encoded by a fractional bit-plane coding (BPC) mechanism to generate intermediate data in the form of a context and a binary decision value for each bit position. In JPEG2000 the embedded block coding with optimized truncation (EBCOT) algorithm has been adopted for the BPC.

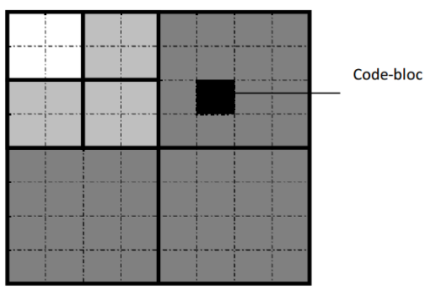


Figure 2. 4: Partitioning into code-Blocks[32]

2.2.4.1.1 Coder EBCOT

EBCOT encodes each bit-plane in three coding passes. The three coding passes in the order in which they are performed on each bit-plane are significant propagation pass, magnitude refinement pass, and cleanup pass. All three types of coding passes scan the samples of a code block in the same fixed order shown in figure 2.5. The codeblock is partitioned into horizontal stripes, each having a nominal height of four samples. As shown in the diagram, the stripes are scanned from top to bottom. Within a stripe, columns are scanned from left to right. Within a column, samples are scanned from top to bottom.

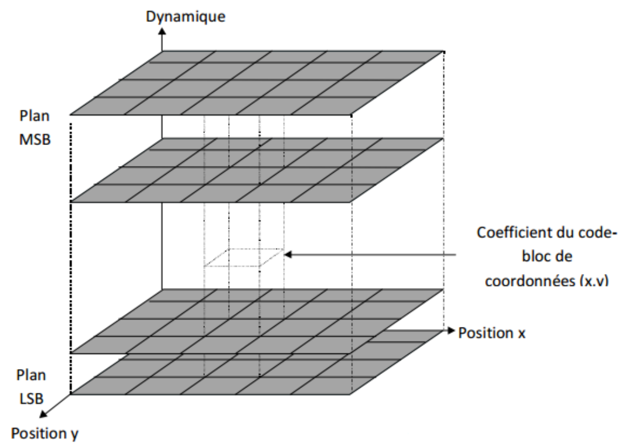


Figure 2. 5: Representatio of a Codeblock in bit plane[32].

Each coefficient bit in the bit plane is coded in only one of the three coding passes, and each coefficient in a block is assigned a binary state variable called its significance state that is initialized to zero (insignificant) at the start of the encoding. The significance state changes from zero to one (significant) when the first nonzero magnitude bit is found.
The context vector for a given coefficient is the binary vector consisting of the significance states of its eight immediate neighbor coefficients For each pass, contexts are created which are provided to the arithmetic coder.

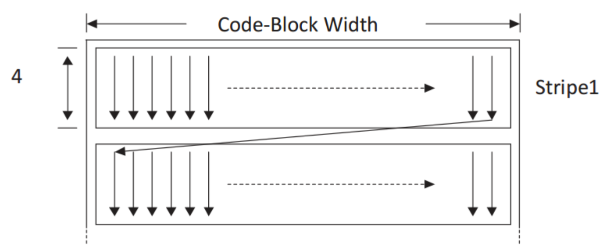


Figure 2. 6: Sample scan order within a code block[32]

In the following each coding pass is described:

a) Significance propagation pass

During the significance propagation pass, a bit is coded if its location is not significant, but at least one of its eight-connect neighbors is significant.

b) Magnitude refinement pass

During this pass, all bits that became significant in a previous biplane are coded. The magnitude refinement pass includes the bits from coefficients that are already significant.

c) Clean-up pass

The clean-up pass is the final pass in which all bits not encoded during the previous passes are encoded (i.e. coefficients that are insignificant and had the context value of zero during the significance propagation pass).The very first pass in a new code block is always a clean-up pass.

2.2.4.1.1 Binary Arithmetic Coding-MQ-Coder

As explained in the previous section, the fractional bit plane coding (EBCOT) produces a sequence of symbols, pairs of context and decision (CX, D), in each coding pass. The context-based adaptive binary arithmetic MQ-coder that is used in JBIG2 is adapted in JPEG2000 standard to encode these symbols.

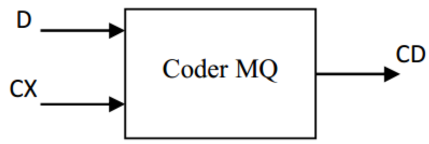


Figure 2. 7: Inputs and outputs of the encoder arithmetic[32]

2.2.4.2Tier-2 Coding

The final operation in a JPEG200 compression consists in generating the code stream. This one is a collection of code block's bit streams gathered into structures called packets and delimited by a header, containing all the signaling needed to decode. The various natural progression modes of JPEG200 code streams are obtained by ordering packets according to a specific progression. Five major progression modes are actually admissible, depending upon which information is interesting to be reconstructed first:

1) Layer-resolution-component-position-progression (LRCP)
2) Resolution-layer-component-position-progression (RLCP)
3) Resolution-position-component-layer-progression (RPCL)
4) Position-component-resolution-layer-progression (PCRL)
5) Component-position-resolution-layer-progression (CPRL)

2.3 Related Works

Since its first appearance 1996, JPEG2000 has been the main interest of many research centers and scientist due to the many benefits previously mentioned in section 2.1. In this section we are going to explore some of the related works concerning the compression techniques and the improvement made through the two decades by the scientific community. In our research and exploration of the many papers, books and reports we have notices that main works on JPEG200 were centered after the compression, in other words, researchers tried to improve the coding and decoding methods of the algorithm rather than trying to improve the transforms or the quantization steps. So we have had a pretty hard time finding papers that deal with the transforms (DWT) which our main interest in the work. The following papers are some the research done on the JPEG200 standard.

Diego Santa-Cruz *et al.* [19] has conducted an analytical study of JPEG2000 in which he defined and gave an overview of JPEG, JPEG2000, MPEG-4 VTC, JPEG-LS, and PNG. They talked briefly about each coding standard, its algorithm, a background history, and some of its known benefits. They also chose princely those standards over the other which was convincing (benefits, popularity, and industry incline…etc). In their comparison methodology they focused primarily on compression efficiency, and used The images “bike” (2048x2560) and “cafe” (2048x2560) , “cmpnd1” (512x768) “chart” (1688x2347) “aerial2” (2048x2048) “target” (512x512) “us” (512x448) as a test set, all these kinds are for the sake of diversity (Natural images, Computer Generated, Text, scanned…etc). They used different software that are we think no longer usable.

The authors summarized the results of their comparison in a quite clever way, stating which standard is better for which functionality, the results are in the table 2.2.

Table 2. 2: Comparative summary of the performance of JPEG2000 versus other Standards [19]

	JPEG2000	JPEG-LS	JPEG	MPEG-4 VTC	PNG
Lossless compression cerformance	+++	++++	+[a]	-	+++
Lossy compression cerformance	+++++	+	+++	++++	-
Progressive bitstream	+++++	-	++[b]	+++	+
Region Of Interest (ROI) coding	+++	-	-	+[c]	-
Arbitrary shaped objects	-	-	-	++	-
Random access	++	-	-	-	-
Low complexity	++	+++++	+++++	+	+++
Error resilience	+++	++	++	+++	+
Non-iterative rate control	+++	-	-	+	-
Genericity[d]	+++	+++	++	++	+++

[a] Only using the lossless mode of JPEG
[b] Only in the progressive mode of JPEG
[c] Tile based only
[d] Ability to efficiently compress different types of imagery across a wide range of bitrates.
Note:Functionality matrix. A “+” indicates that it is supported, the more “+” the more efficiently or better it is supported. A “-” indicates that it is not supported.

The authors conclude at the end that JPEG2000 offers the richest set of features and provides superior rate-distortion performance. However, this comes at the price of additional complexity when compared to JPEG and JPEG-LS, which might be perceived as a disadvantage for some applications, as was the case for JPEG when it was first introduced.

The paper is in overall complete in the essence of its aim, it really compared the four standards, in summarized tables and in detail in the paper, the only thing that we can add is more quality metrics like SSIM and other useful metrics.

Amr M Kishki *et al.* [20] have used the Haar wavelet transform in the JPEG2000 algorithm (HW-JPEG200) and compared it with Fast Zonal DCT, BTC, AMBTC, Enhanced BTC, 2D DWT Zonal, JPEG, and JPEG2000 using the quality metrics PSNR, Energy Consumption, Compression Ratio (CR), Bit Rate (BR), Correlation Coefficient (CC), and Spatial Frequency (SF). But before that they have showed the related work and the basic fundamentals behind each of the standard to keep the reader in track of recent studies. Their method of applying the Haar wavelet transform in JPEG2000 were devided into 5 steps which we are not going to mention them here, the reader is referred to [20] for detailed explanation of the method.

PSNR comparison (which is presented in bar chart) shows that HW-JPEG2000 is better than the others and scores 98.7404 dB while JPEG2000 scores 76.2964dB. For Energy Consumption the bar chart shows that the JPEG2000 and HW-JPEG2000 are the least energy consuming, and are quite the same of about 0.150 Joule. As for CR, BR, CC and CF tables 2.3 and 2.4 from [20] show the results in detail.

This work is actually similar to what we have done, but with more complex approach, ours was actually simpler in terms of complexity, and effectiveness. We could follow the same pattern that these authors have used in terms of quality metrics, and we can make a comparison between our approach and theirs.

Table 2. 3: CR & BR comparison

	CR	BR
Fast Zonal DCT	9.0903	0.8801
BTC	4	2
AMBTC	4	2
Enhanced BTC	4	2
2D DWT Zonal	3.0435	2.6286
JPEG	2.9981	2.6284
JPEG2000	1.9050	4.1994
HW-JPEG2000	1.7204	4.6502

Table 2. 4: CC & CF comparison

	CC	SF
Original Image	1	53.5091
Fast Zonal DCT	0.9991	54.0237
BTC	0.8901	32.7753
AMBTC	0.9049	26.0892
Enhanced BTC	0.9629	49.3303
2D DWT Zonal	0.9582	48.6785
JPEG	0.9353	55.5288
JPEG2000	0.9471	49.3920
HW-JPEG2000	0.9941	60.1078

Chih-Chang Chen *et al.* [21] proposed a new method for generating region-of-interest (ROI) mask in JPEG2000. In the bit plane coding, the DWT coefficients are evaluated to automatically generate the ROI mask by using three steps: sub-block classification, central point determination and mask generation. The data from the significant propagation pass in EBCOT procedure can be used for dynamically determining the ROI mask. The central point of ROI is defined as the geometric center for the high-frequency signals spread out in the image, these high-frequency signals are easily detected from the decomposed DWT component (HL, LH, HH). The generation of the ROI mask is searched by swing from the central point and going outward through dilation and erosion filter operations. From perceptual point of view, they group many small regions to become several close and large regions for the ROI mask.

This work proposes a low-complexity scheme to determine the ROI mask the proposed scheme can be easily realized with the JPEG200 at a low computational cost. As compared to the conventional ROI coding using the fixed and pre-determined area, the proposed scheme can provide more clear edges in multiple object shapes. Additionally, the ROI mask is updated by the picture content. The proposed scheme can be easily realized with the JPEG200 at a low computational cost. As compared to the fixed ROI mask, the proposed scheme can yield better perceptual quality at the same bit rate.

2.4 Conclusion

In this chapter we presented a brief study of the JPEG2000 standard with an overview of the related work in the papers that were most relevant to our project. In the next chapter we are going to propose our method for optimizing JPEG2000 compression standard along with a study of the different transforms. Also we are going to introduce a new quality index to evaluate obtained images.

Chapter III: Optimization of JPEG2000 Compression Standard

3.1 Introduction

In order to evaluate the performance of JPEG2000 with different color spaces, different types of wavelet transform, and different wavelet levels, we have implemented the MATLAB code of basic JPEG2000.

3.2 Methodology

Figure 3.1 summarizes the considered evaluation process (compression/decompression). After color space transformation, we apply wavelet transform on each component with level L, and then the result is quantized by the following formula:

$$Cq(m,n) = round\left(\frac{C(m,n)}{q}\right) \qquad (3.\ 1)$$

Where q is the quantization step, Cq is the quantized wavelet coefficient, and C(m,n) is the wavelet coefficient at the position (m,n). Then, each Subband is encoded separately using Huffman coding [22] [23] as shown in figure3.2.

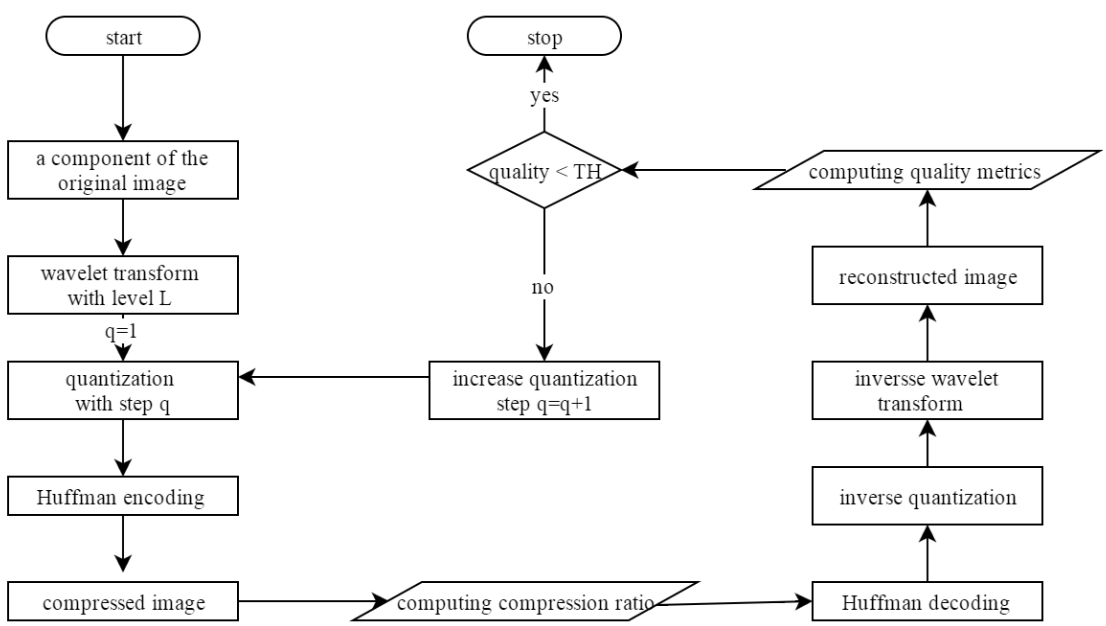


Figure 3. 1: Flowchart of the evaluation process

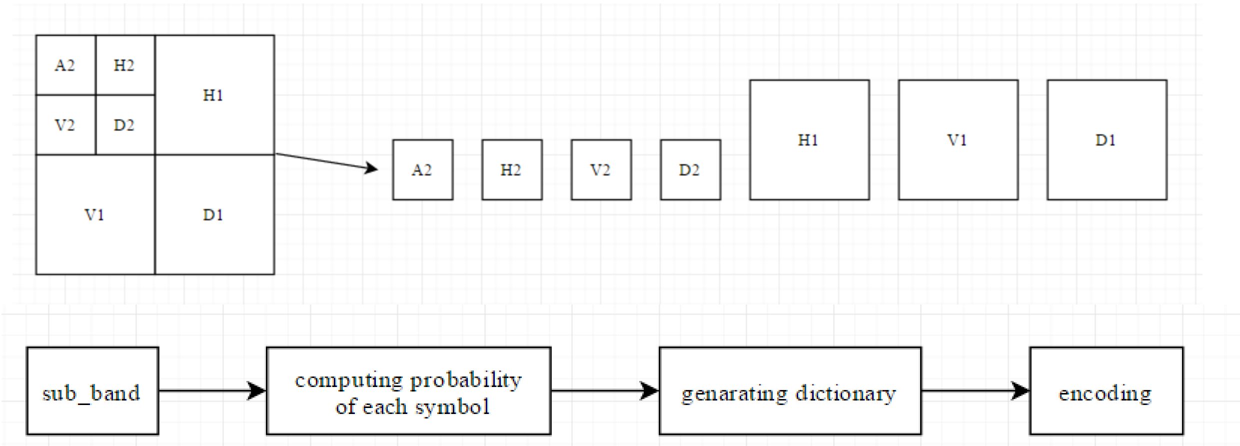


Figure 3. 2: Flowchart of the Huffman encoding process

After computing the probability of each symbol in the subband, a dictionary is generated and the image is coded. After Huffman encoding the Compression Ratio (CR) is computed by using equation (1.1).

In order to reconstruct the image from the compressed image, we follow the inverse process as shown in figure 3.1, and then we evaluate the quality of obtained images using quality metrics. To achieve that, we are going to use multiple quality metrics which are mentioned in detail in section (1.4). These metrics are:

PSNR: which is a widely used full reference quality index that measures the spatial similarity between two images and is given by the equation (a modified version of equation (1.11))

$$PSNR = 10 \times log10 \left(\frac{255^2 \times 3}{(MSE(R) + MSE(G) + MSE(B))} \right) \tag{3. 2}$$

MSE(R) is the mean square error of the R component of the image given by equation (1.7).

MS-SSIM: (multi-scale structural similarity for image quality assessment) is another full reference quality metric used to measure the structural similarity between the original and the decompressed image as explained in detail in section (1.4.3).

QM: because JPEG2000 introduces blur effect, so we have used a blur quality metric proposed by *F. Kerouh et al.* [9] to assess the introduced blur amount.

DI (Detail Index): is the mean of edges of the image. We propose to use this index in order to notice how the details are affected by the compression process. It is given by the following formula:

$$DI = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{n=1}^{N} \sqrt{H^2(m.n) + V^2(m,n)} \qquad (3.\ 3)$$

Herein, M and N stand for the dimension of the subbands, *H (m,n)* and *V(m,n)* are the horizontal and vertical wavelet details at the position *(m, n)*, respectively.

3.3 Optimal Color Space

To achieve the tests, we consider three images with different DI (detail index) values selected from a free database named LIVE [24], the selected images are presented in figure 3.3.

DI= 9.612 DI= 19.2935 DI= 5.2162

1 2 3

Figure 3. 3: Processed images

3.3.1 Methodology

Each image from figure 3.3 is presented by the three color spaces (RGB, YCbCR, and HSV), then each color space is compressed using JPEG2000 by fixing the wavelet transform to db2 and level L=6.

For the RGB color space each component is compressed iteratively as shown in figure 3.1, the stopping criterion (TH) is an index about the PSNR experimentally set to 31.

For the YCbCr color space we use the same process, except that the stopping criterion for the Y component is different from the one of Cb and Cr. This can be explained by the fact that most of the information is contained in the Y component [3], unlike the RGB where the information is equally distributed between the components. So a highest PSNR value is used for the luminance component compared to chrominance. For Y component, the PSNR is set experimentally to 40, and for the Cb and Cr components it is set to 36.

For the HSV only the S and V component are compressed, and the stopping criterion for both of them is set to PSNR =40.

3.3.2 Results and discussion

The explained methodology is applied on the three considered images. Obtained results in terms of PSNR, MSSSIM, QM, CR, and DI are depicted on figures 3.4, 3.5, and 3.6 respectively.

For the obtained results, we can notice that compressing the YCbCr color space provides the highest compression ratio with best quality in terms of PSNR, MS-SSIM, and QM. The lowest compression ratio is given by HSV color space. Furthermore, we can notice that details are very well conserved in the YCbCr color space compared to other color spaces. Moreover, we remark that for high detailed images the compression ratio is lower and vice versa.

We conclude that the optimal color space for JPEG2000 compression standard is the Luminance/Chrominance (YCbCr).

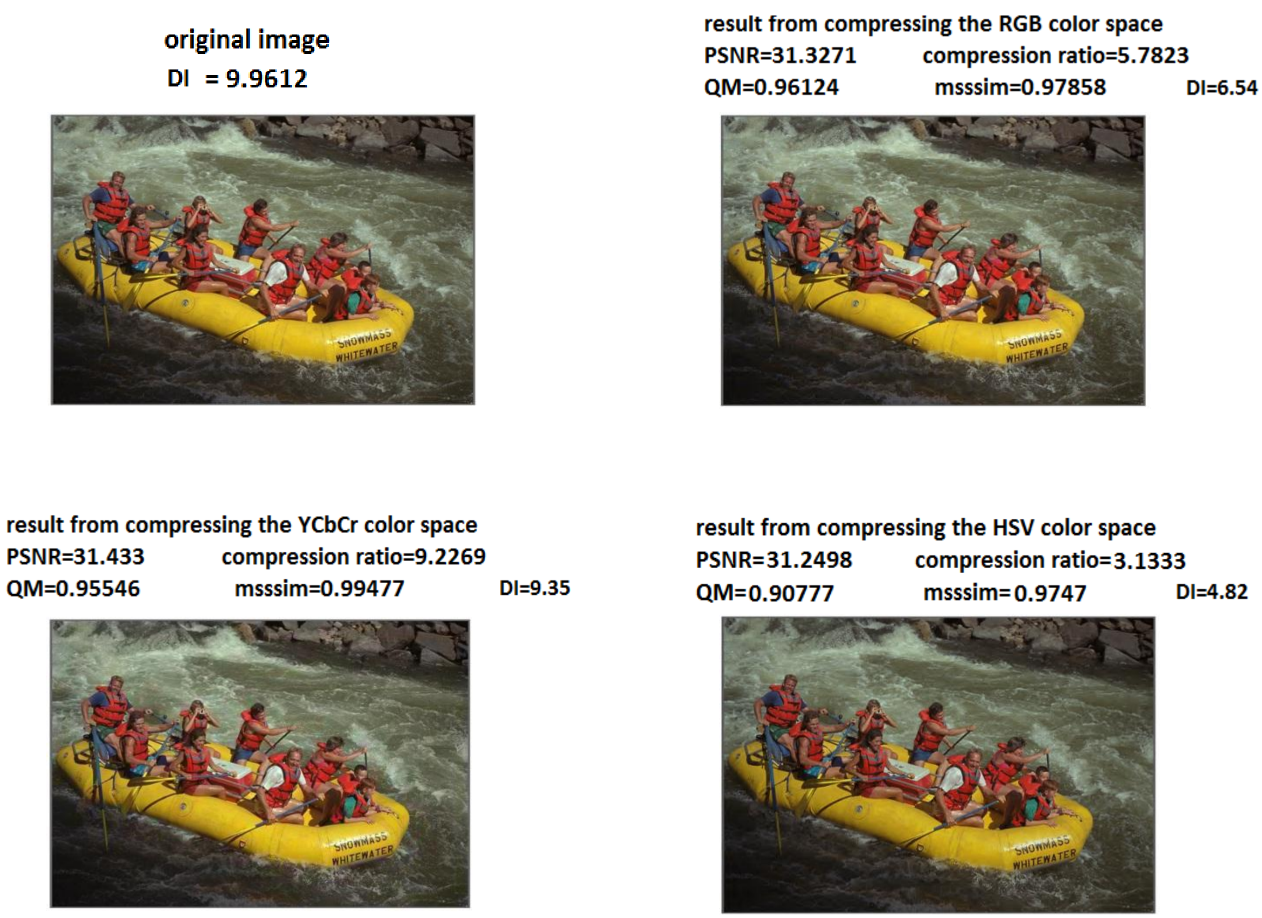


Figure 3. 4: Compression results for different color spaces of image 1

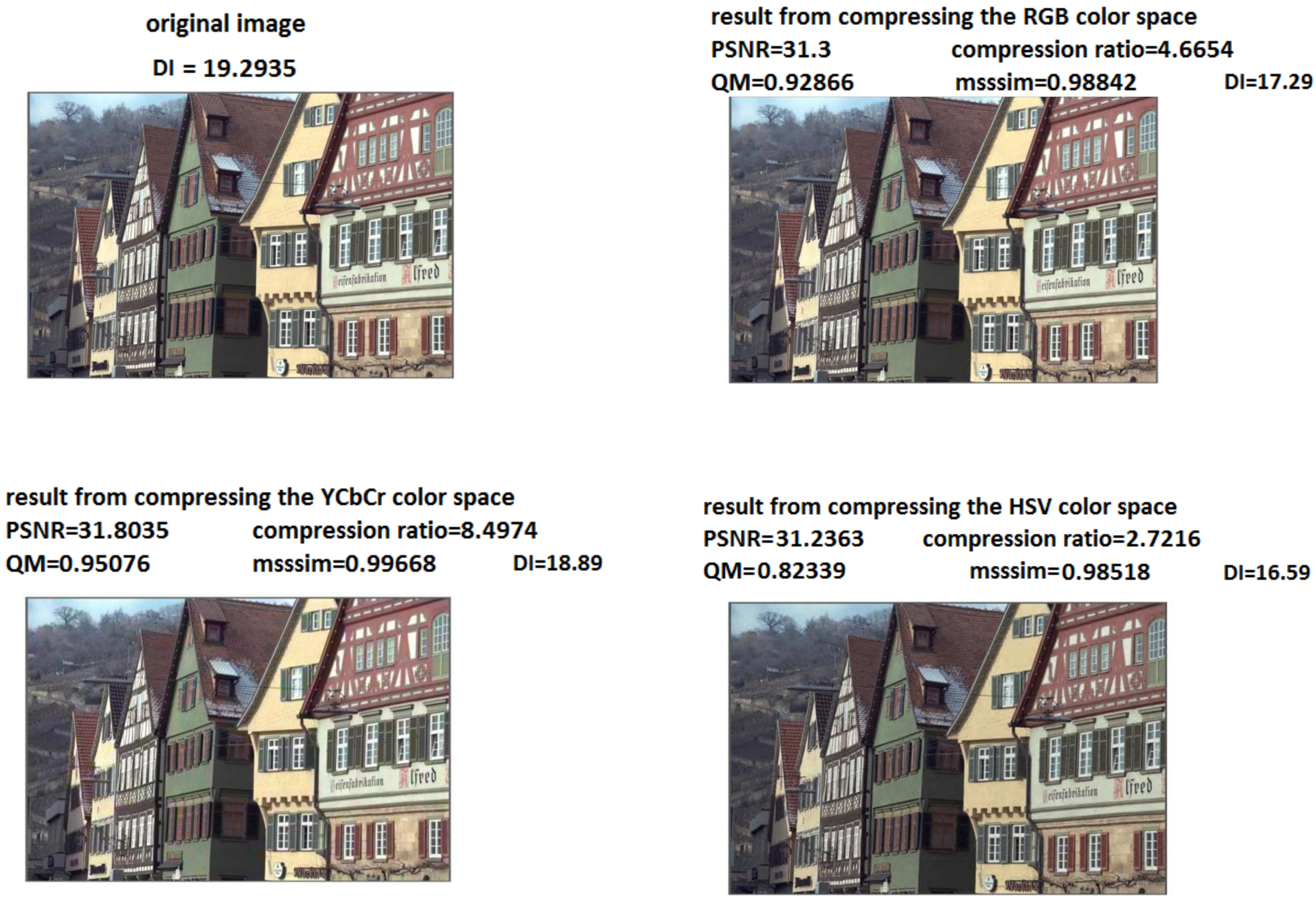


Figure 3. 5: Compression results for different color spaces of image 2

original image

DI = 5.2162

result from compressing the RGB color space

PSNR=31.3936 compression ratio=6.8109

QM=0.96474 msssim=0.9848 *DI=2.74*

result from compressing the YCbCr color space

PSNR=31.7282 compression ratio=14.2282

QM=0.94768 msssim=0.99303 *DI=3.54*

result from compressing the HSV color space

PSNR= 31.0875 compression ratio=3.6571

QM=0.88948 msssim=0.97636 DI=2.75

Figure 3. 6: Compression results for different color spaces of image 3

3.4 Optimal Wavelet

To achieve the tests, we consider four images with different DI (detail index) values. The luminance components of the selected images are presented in figure 3.7.

DI= 5.2162 DI=9.612 DI=19.2935 DI=35.079

1 2 3 4

Figure 3. 7: Processed images

3.4.1 Methodology

For tests we use two wavelet algorithms at a fix resolution level set to 6. The discrete wavelet transform (DWT) with four different wavelets (db1, db2, coiflet1, and bior2.2), and the lifting wavelet transform (lwt). Note that there is only one wavelet for lwt, which is Haar wavelet (db1).

Each Y component of the selected images is compressed iteratively by JPEG2000 algorithm.

3.4.2 Results and discussion

The evolution of the compression ratio and the quality metrics (PSNR, MS-SSIM, and QM) versus the quantization step for each considered image are recorded and plotted in graphs of figures 3.8, 3.10, 3.12, and 3.14. Images 1 and 2 shown in figure 3.7 are less detailed images (DI=5.21 and 9.96, respectively) compared to others. From figure 3.8 and figure 3.10 we notice that db2 provides the highest compression ratio and the lowest QM, which means that db2 introduces more blur effect on this type of image, which result in low image quality. Otherwise, compressed images using bior22 have the highest MS-SSIM, high QM (less blur effect), and high compression ratio, which result in high image quality.

Image 3 of figure 3.7 is a high detailed image (DI=19.29), from figure 3.12 we see that db1 provides the highest compression ratio with high MS-SSIM and high QM (less blur effect), which result in high image quality, bior22 gives the highest MS-SSIM, high QM (less blur effect), and high compression ratio, which result in high image quality.

Image 4 shown in figure 3.7 is a high detailed image (DI=35), from figure 3.12 we notice that db1 gives the highest compression ratio with highest MS-SSIM and high QM (less blur effect), which result in high image quality.

For the four images the lifting Haar wavelet transform gives the lowest compression ratio and the lowest MS-SSIM.

From all the obtained results, we conclude that for images that have less details the bior2.2 dwt is the best one. In fact, it gives the highest image quality and high compression ratio. For high detailed images db1 dwt provides the highest compression ratio and high image quality.

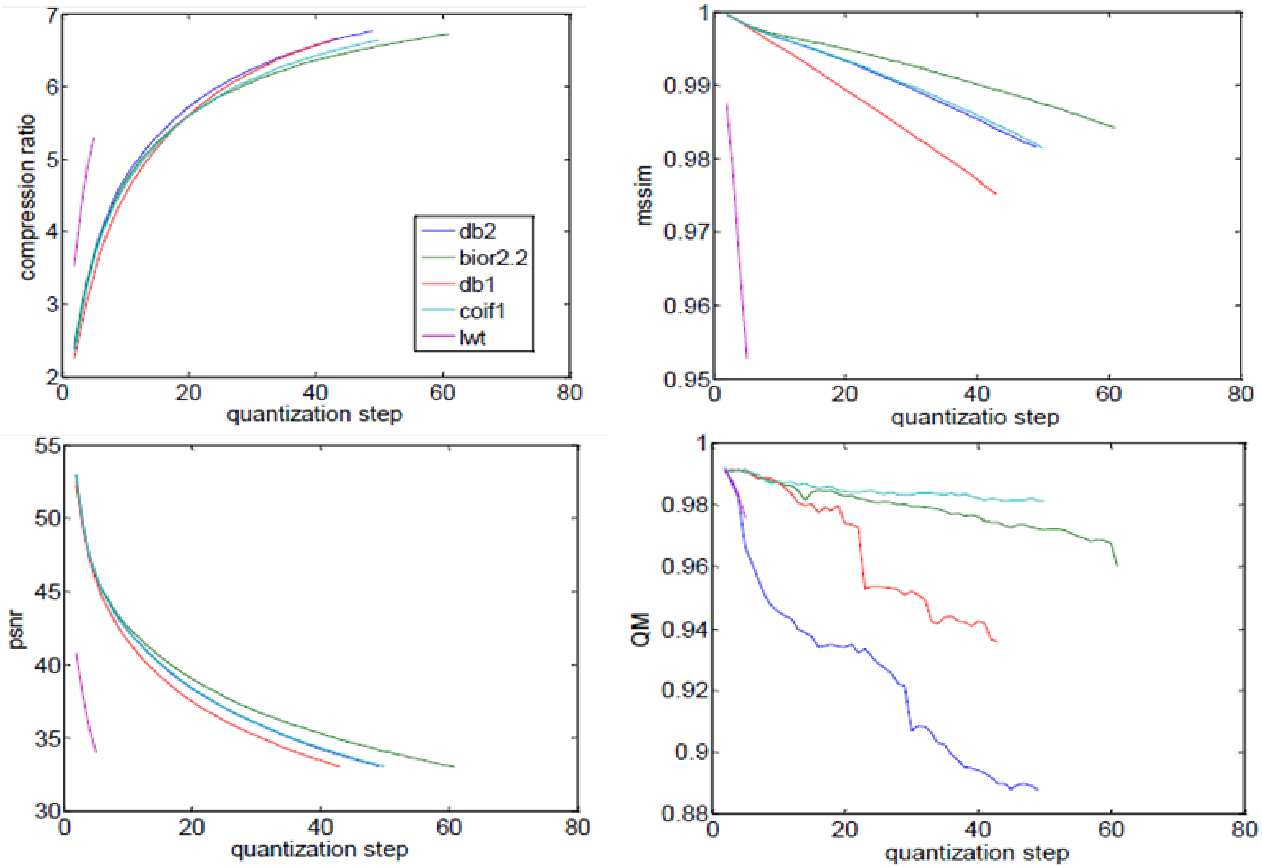


Figure 3. 8: Graphs of the CR and quality metrics of image 1 for different wavelet

Figure 3. 9: Resultant image 1 with highest compression ratio and quality

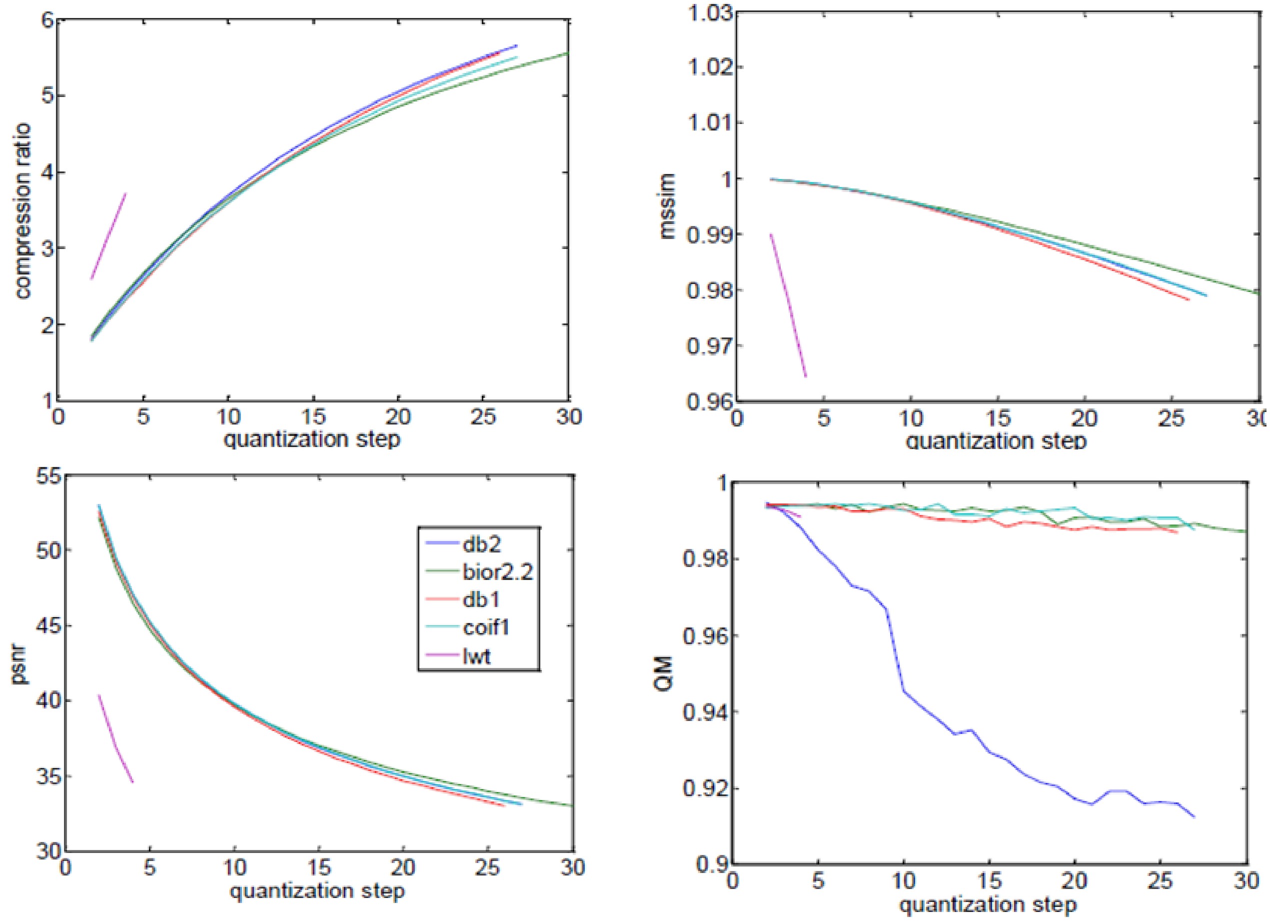


Figure 3. 10: Graphs of the CR and quality metrics of image 2 for different wavelet

original image
DI = 9.9612

result from compressing the Y component
PSNR=33.1273 **compression ratio=5.651**
QM=0.91237 **msssim=0.97896** **DI=10.21**

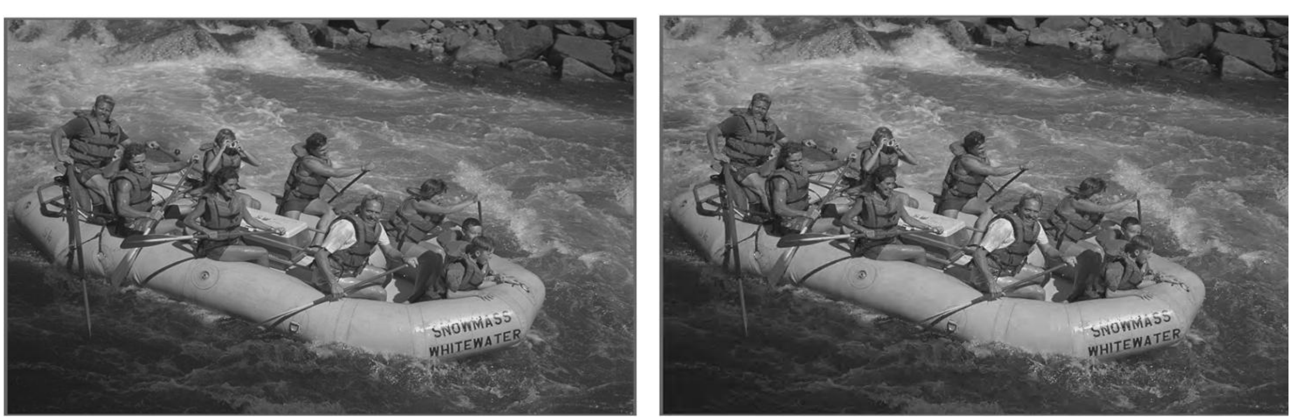


Figure 3. 11: Resultant image 2 with highest compression ratio and quality

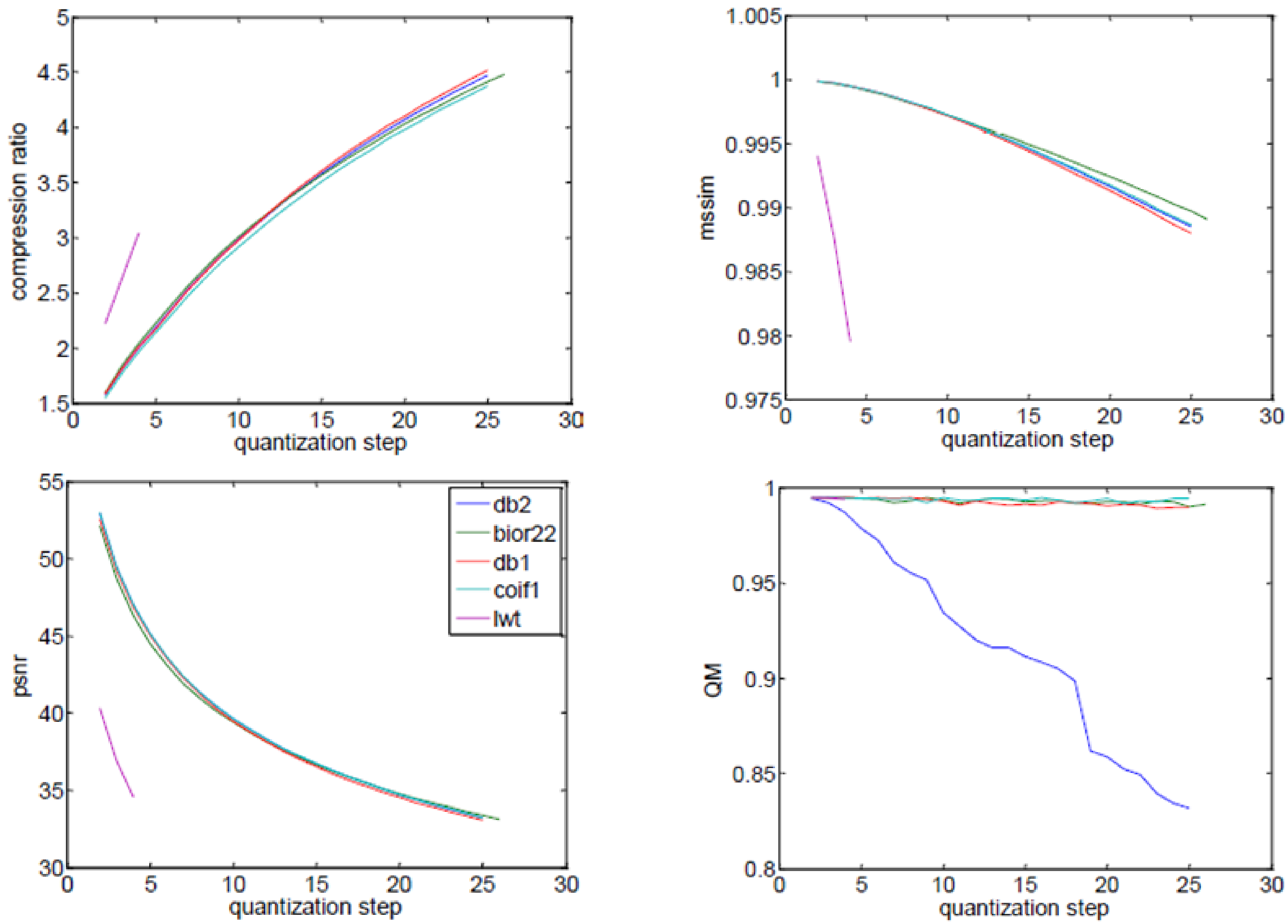


Figure 3. 12: Graphs of the CR and quality metrics of image 3 for different wavelet

Figure 3. 13: Resultant image 3 with highest compression ratio and quality

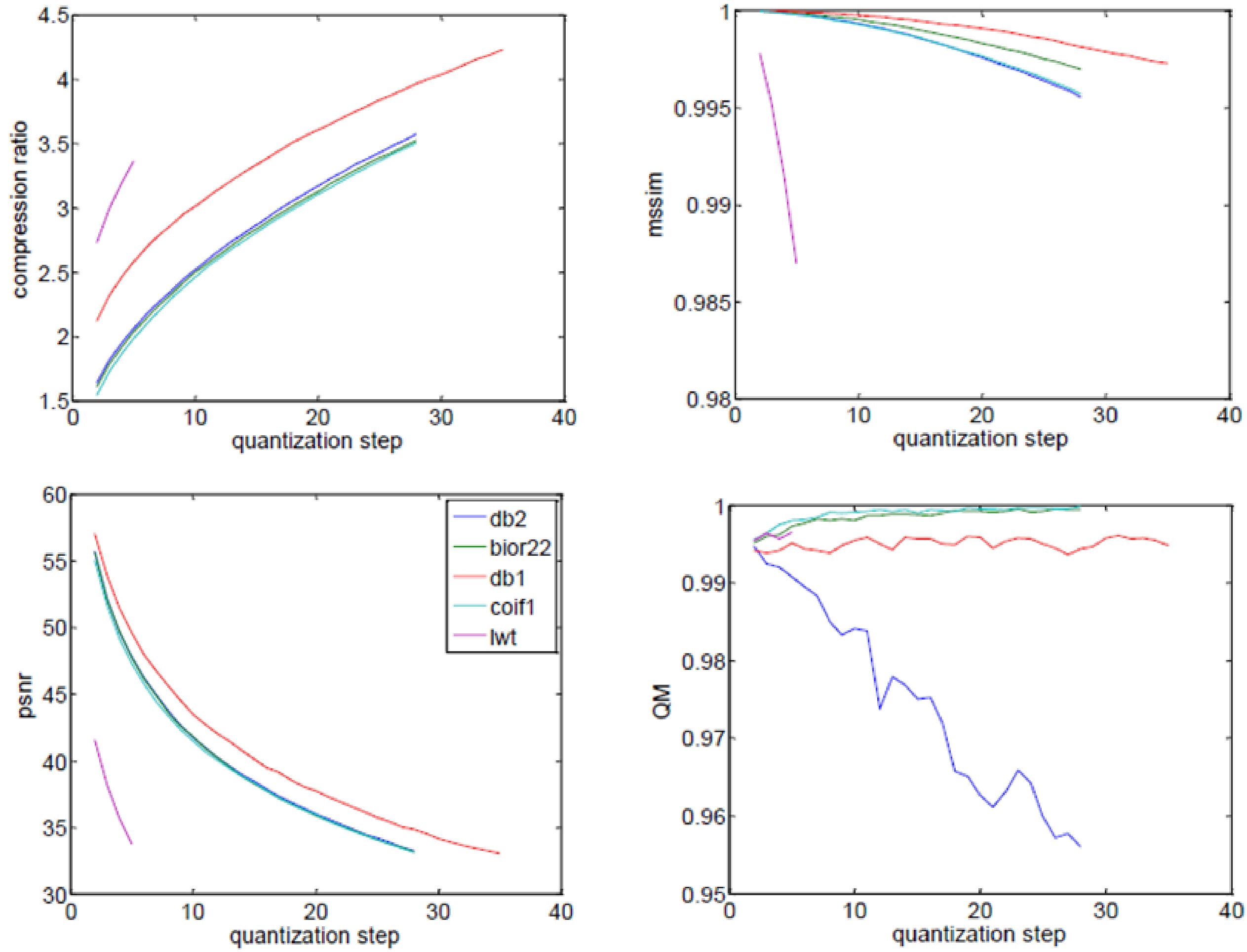
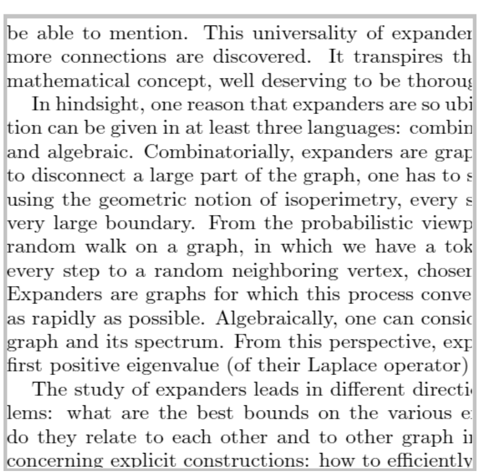


Figure 3. 14: Graphs of the CR and quality metrics of image 4 for different wavelet

original image
DI =35.079


be able to mention. This universality of expander
more connections are discovered. It transpires th
mathematical concept, well deserving to be thoroug
In hindsight, one reason that expanders are so ubi
tion can be given in at least three languages: combin
and algebraic. Combinatorially, expanders are grap
to disconnect a large part of the graph, one has to s
using the geometric notion of isoperimetry, every s
very large boundary. From the probabilistic viewp
random walk on a graph, in which we have a tok
every step to a random neighboring vertex, chosen
Expanders are graphs for which this process conve
as rapidly as possible. Algebraically, one can consid
graph and its spectrum. From this perspective, exp
first positive eigenvalue (of their Laplace operator)
The study of expanders leads in different directi
lems: what are the best bounds on the various e
do they relate to each other and to other graph in
concerning explicit constructions: how to efficiently

result from compressing the gray image
PSNR=33.0469 compression ratio=4.2267
QM=0.99484 msssim=0.99727 DI=35.56

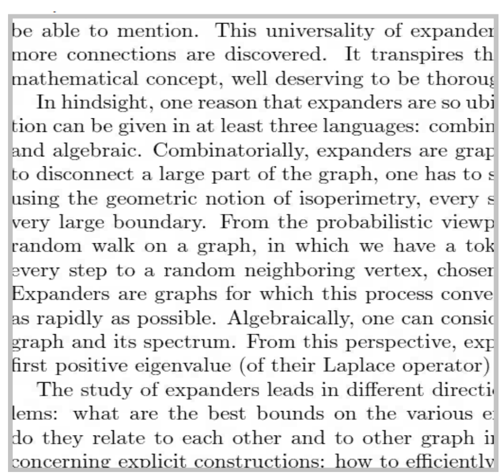

be able to mention. This universality of expander
more connections are discovered. It transpires th
mathematical concept, well deserving to be thoroug
In hindsight, one reason that expanders are so ubi
tion can be given in at least three languages: combin
and algebraic. Combinatorially, expanders are grap
to disconnect a large part of the graph, one has to s
using the geometric notion of isoperimetry, every s
very large boundary. From the probabilistic viewp
random walk on a graph, in which we have a tok
every step to a random neighboring vertex, chosen
Expanders are graphs for which this process conve
as rapidly as possible. Algebraically, one can consid
graph and its spectrum. From this perspective, exp
first positive eigenvalue (of their Laplace operator)
The study of expanders leads in different directi
lems: what are the best bounds on the various e
do they relate to each other and to other graph in
concerning explicit constructions: how to efficiently

Figure 3. 15: Resultant image 4 with highest compression ratio and quality

3.5 Optimal Wavelet Level

To achieve the tests, we consider the same four images with different DI (detail index) values selected previously (figure 3.7).

3.5.1 Methodology

Each Y component of the images is compressed iteratively using JPEG2000 as shown in figure3.1. For images 1 and 2 we have used different levels of bior2.2 wavelet transform, and for image 3 and 4 we have used different levels of db1 wavelet.

3.5.2 Results and discussion

The values of the compression ratio and the quality metrics (PSNR, MS-SSIM, and QM) are recorded and plotted in graphs of figures 3.16, 3.17, 3.18, and 3.19.

The images 1 and 2 shown in figure 3.7 are less detailed images (DI1=5.21 and DI2=9.96), compressed by bior2.2, from figure 3.16 and figure 3.17 we see that l=6 gives the highest compression ratio with low MS-SSIM.

The images 3 and 4 shown in figure 3.7 are more detailed images (DI3=19.29 and DI4=35), compressed by db2. From figure 3.18 and figure 3.19, we remark that l=8 gives the highest compression ratio with low MS-SSIM.

For the four images we see that increasing the level results in decreasing the MS-SSIM (quality metric).

As a conclusion, the level L=6 of bior2.2 is the best level for images with less details, and level L=8 of db1 is the best level for images with high details.

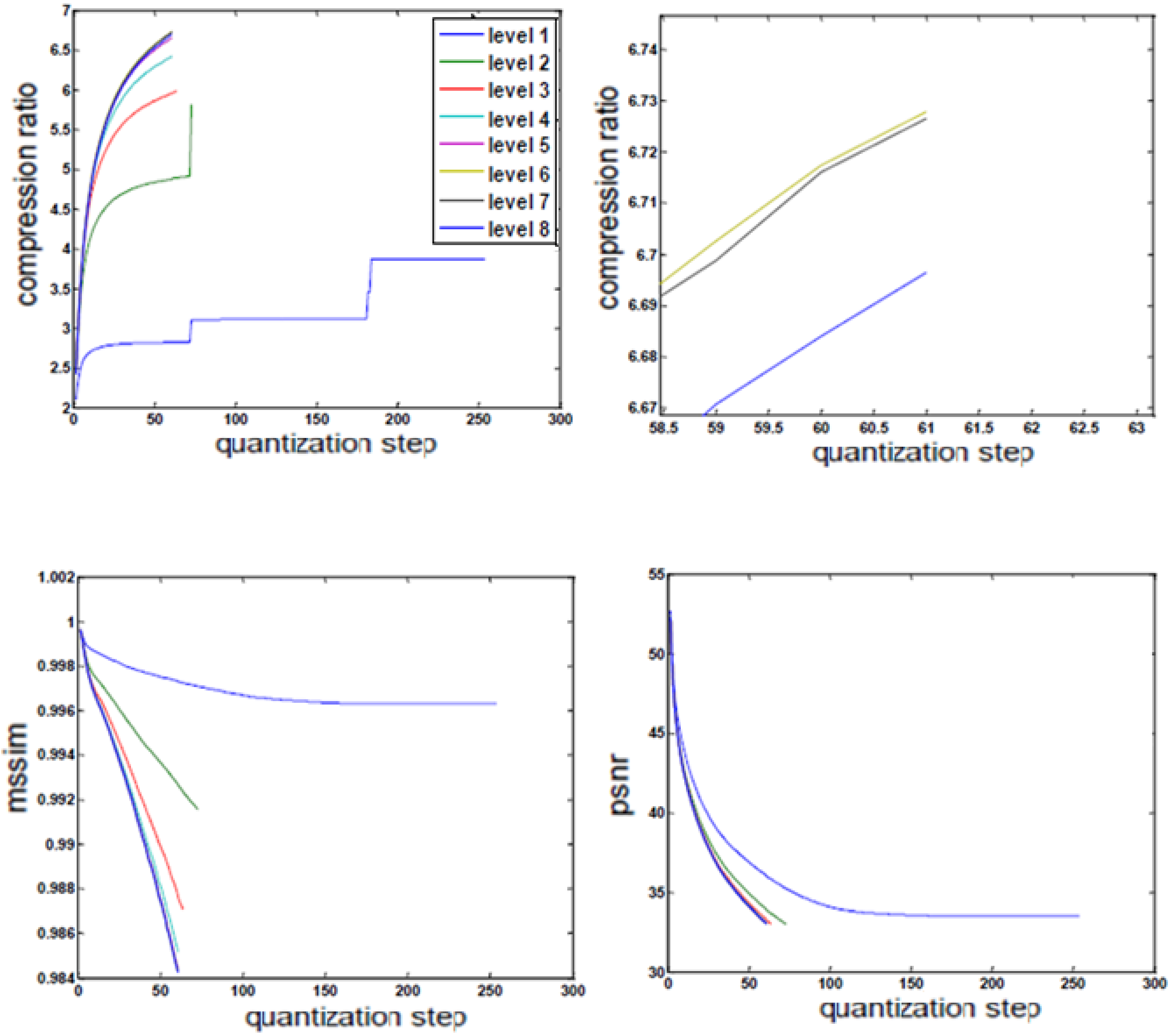


Figure 3. 16: Graphs of CR and the quality metrics of image 1 for different wavelet's levels

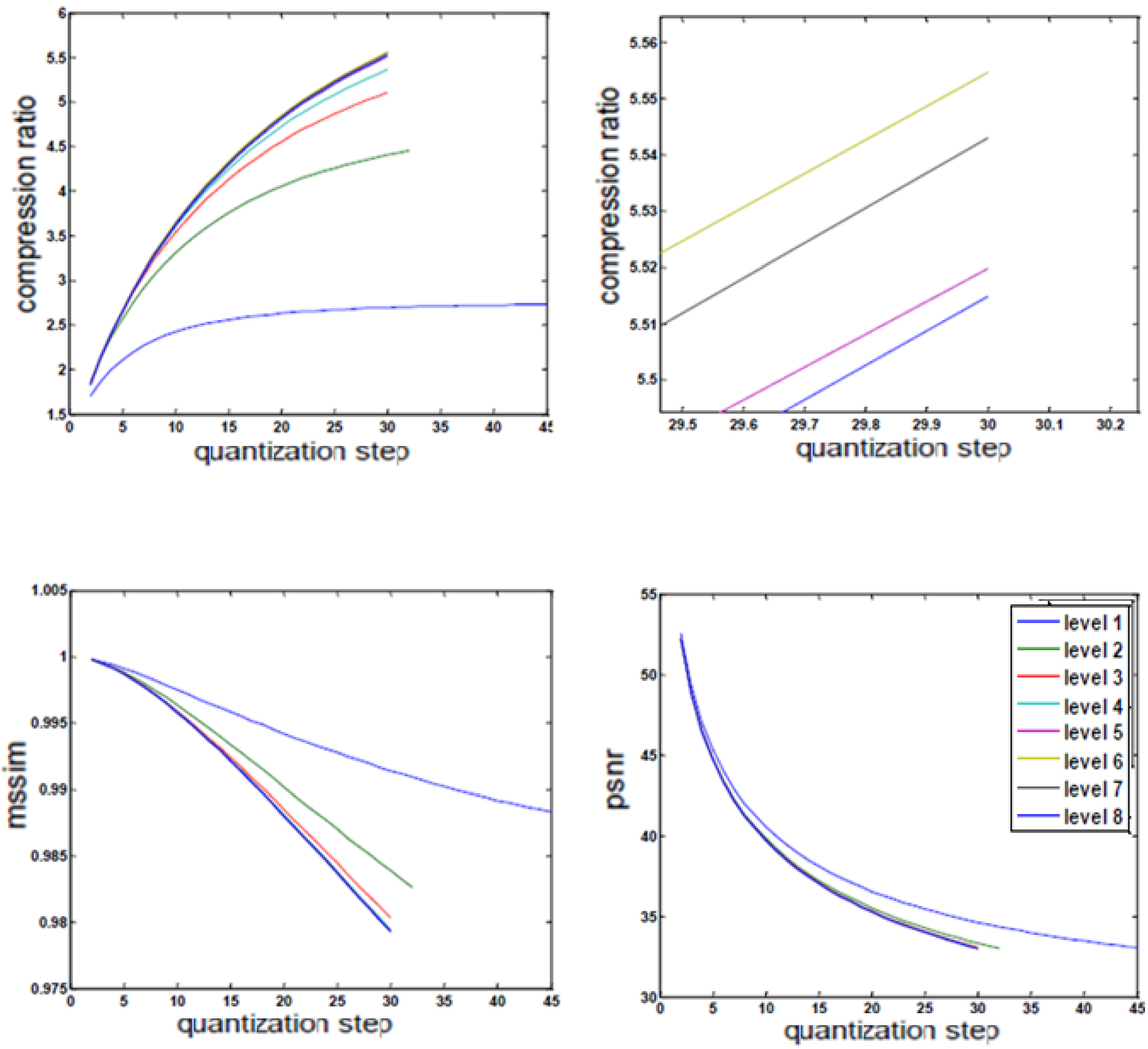


Figure 3. 17: Graphs of CR and the quality metrics of image 2 for different wavelet's levels

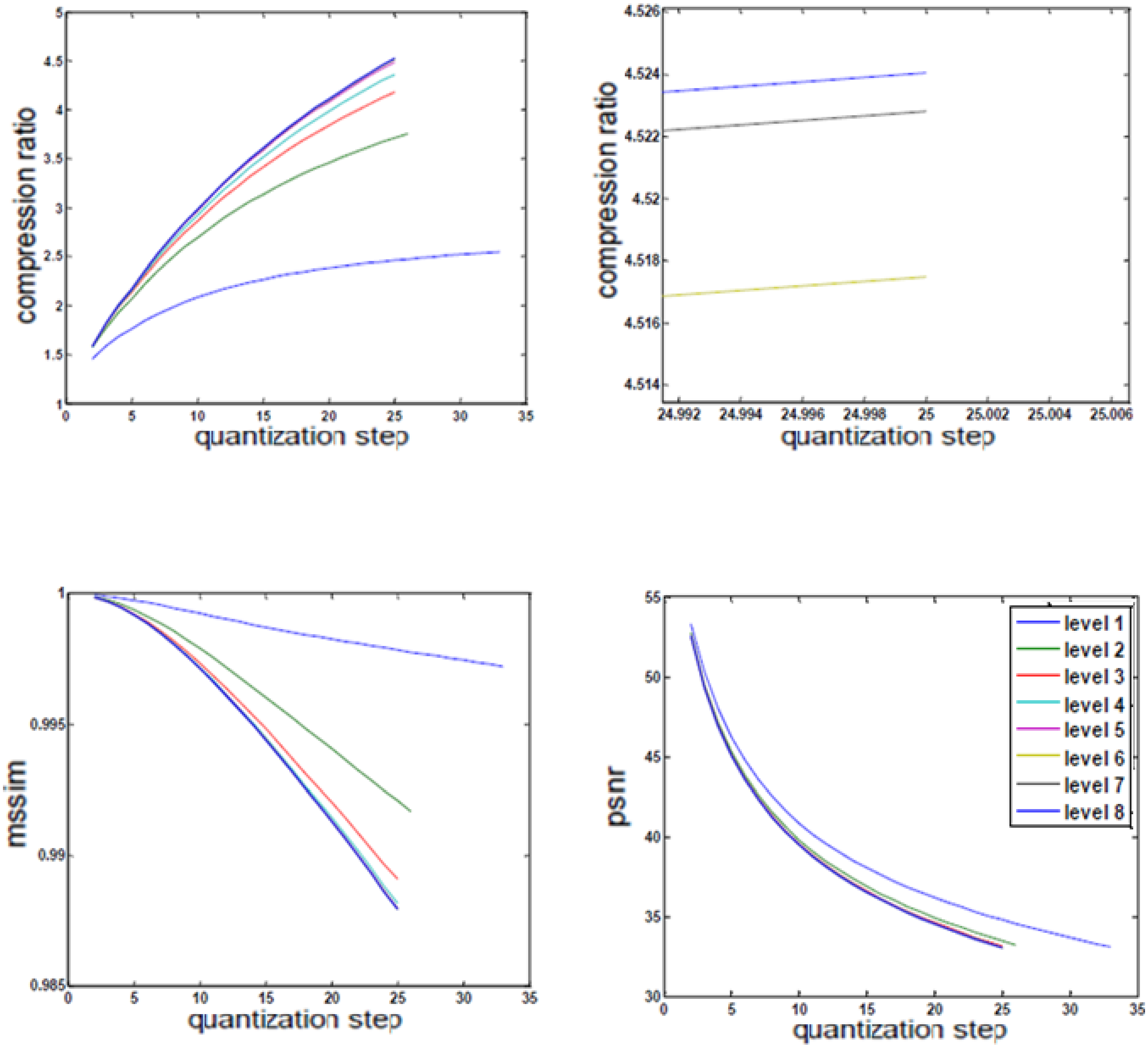


Figure 3. 18: Graphs of CR and the quality metrics of image 3 for different wavelet's levels

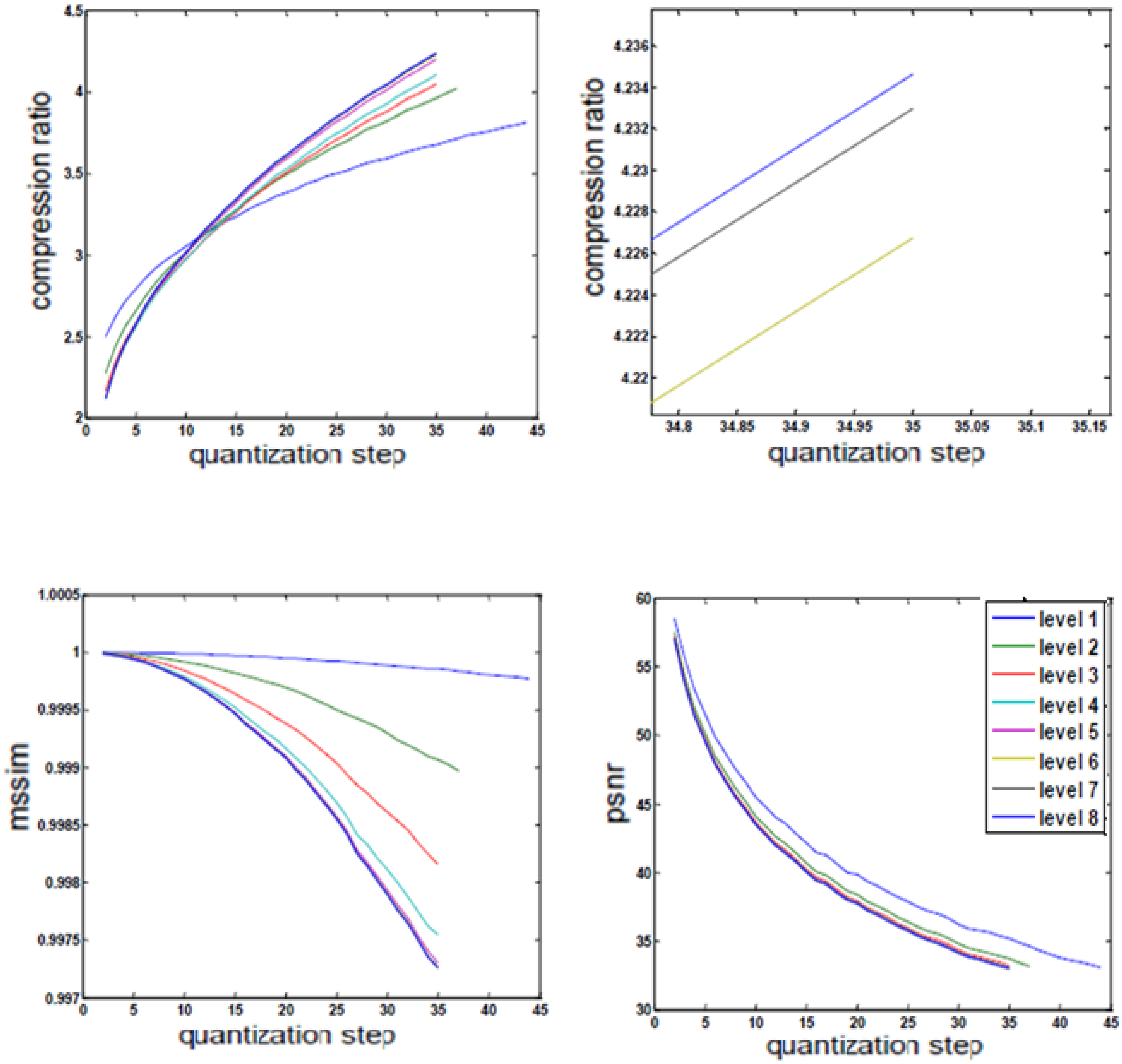


Figure 3. 19: Graphs of CR and the quality metrics of image 4 for different wavelet's levels

3.6 Test and validation

In the previous parts we have find that for detail index DI <19 bior2.2 provides best quality with resolution level set to 6, and for detail index DI > 30 db1 with 8 resolution levels gives better compression ratio, to validate the obtained results, let as consider 5 images (figure 3.20), finger print, astronomic, two medical imaging, and Lion.

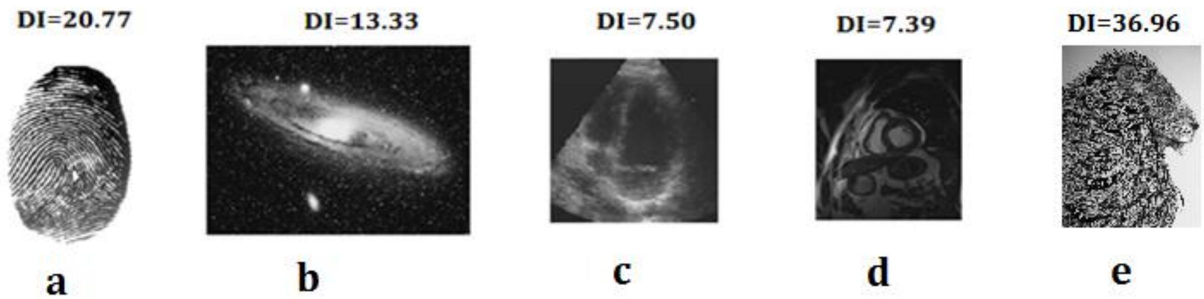


Figure 3. 20: Test images: a) Fingerprint, b) A Galaxy, c) Echography, d) IRM, e) Lion

Each images is compressed iteratively by JPEG2000 using the level L=6 for both db1 and bior2.2 as shown in figure 3.1.

Results and discussion

The values of the compression ratio and the quality metric MS-SSIM corresponding to PSNR=33 are recorded in table3.1, the obtained values in the previous part(3.4 optimal wavelet) in addition to the results of table 3.1 are plotted in function of detail metric DI as shown in figure 3.22.

From figure 3.22 we see that db1 has higher compression ratio than bior2.2, for images that have detail index DI < 30, the use of bior2.2 provides greater MS-SSIM value compared to db1 wavelet (it means that it gives better image quality).

From all the obtained results, we conclude that for images that have less details DI < 30 the bior2.2 dwt is the best one in terms of image quality. For high detailed images DI > 30 db1 is the best.

Table 3. 1: Results of testing process

Image	Detail Index	Db1 wavelet transform		Bior2.2 wavelet transform	
		Compression ratio	MS-SSIM	Compression ratio	MS-SSIM
A	20.77	3.2875	0.9973	3.3529	0.9975
B	13.33	5.5498	0.9827	5.1851	0.9841
C	7.50	6.7638	0.9679	6.2441	0.9727
D	7.39	6.0400	0.9731	5.5988	0.9715
E	36.96	4.2501	0.9974	3.6057	0.9972

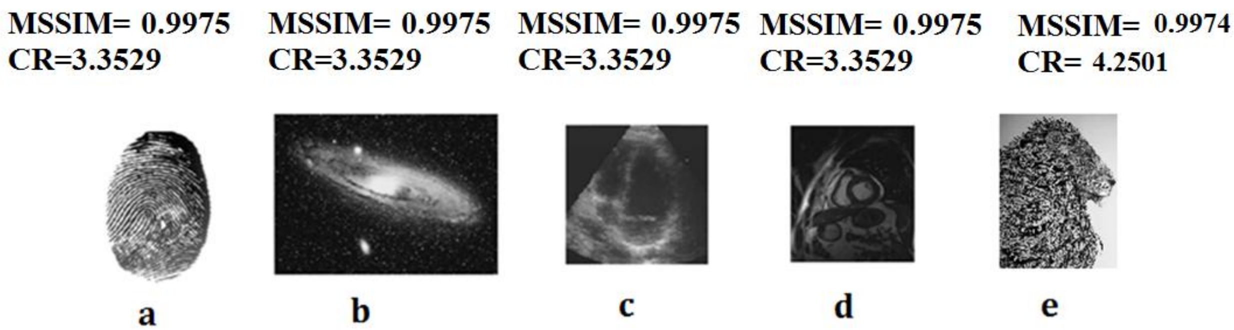


Figure 3. 21: Resultant compressed test images

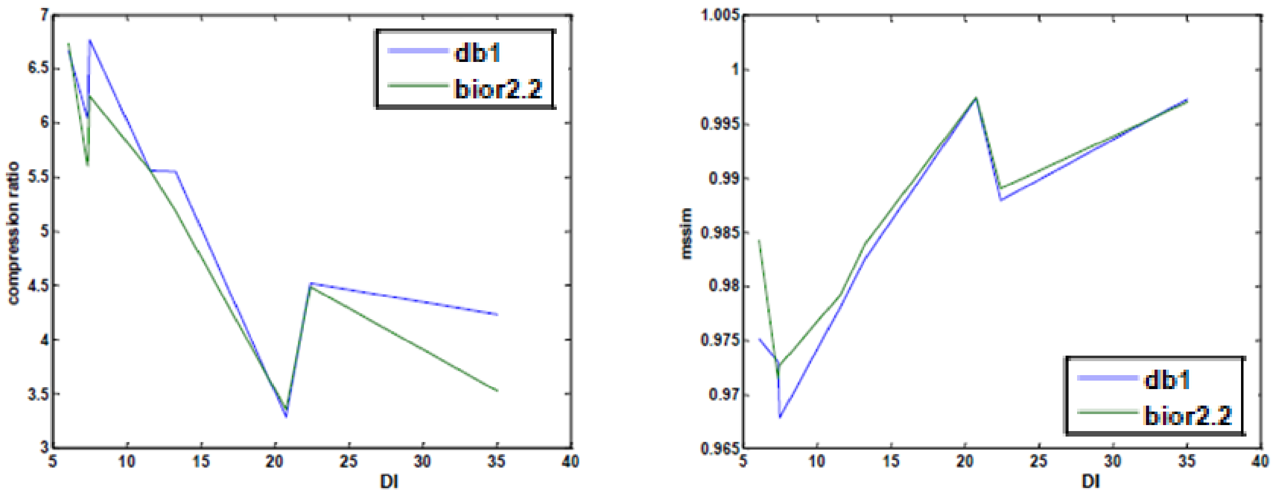


Figure 3. 22: Graph of MS-SSIM and CR in function of DI

3.7 Comparative study with the standards

In this part, we consider the images from figure 3.3; each image is compressed using the proposed optimized JPEG2000 version, JPEG2000 standard and JPEG standard with the same PSNR, using Adobe Photoshop CC 2015 software.

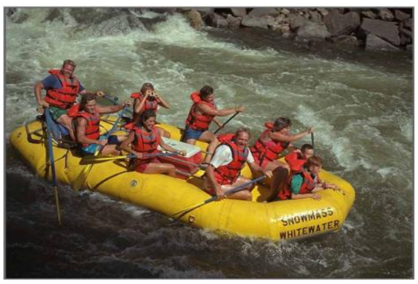
Results and discussion

The resulted images and their corresponding values of the quality metrics and compression ratio are shown in figure 3.23 and figure 3.24, in all the three images, we see that for the same PSNR, the images compressed using JPEG2000 have higher image quality in terms of the MSSIM and QM, but it has lower compression ratio compared to the JPEG, we see also that images compressed using the proposed optimized JPEG2000 have the highest image quality in terms of the MSSIM and QM, but it has lower compression ratio because the JPEG and JPEG2000 standards use more effective encoding process, so if we use the same encoding process, we expect that the proposed version will give the best image quality with the same compression ratio.

Result of compression with proposed JPEG2000
PSNR=32.1876 MSSIM= 0.9958 CR=9.0605
QM=0.99429 DI=10.2551

Result of compression with JPEG2000 standard
PSNR=32.2874 MSSIM= 0.98158 CR=13.7142
QM=0.99328 DI=10.2836

Result of compression with JPEG standard
PSNR=30.9952 MSSIM= 0.98833 CR=14.4
QM=0.98608 DI=9.0999



Result of compression with JPEG standard
PSNR=30.3193 MSSIM= 0.99221 CR=11.52
QM=0.98199 DI=18.4032

Result of compression with our algorithm
PSNR=31.9067 MSSIM= 0.9958 CR=8.5636
QM=0.99333 DI=19.4507

Result of compression with the standard
PSNR=31.9169 MSSIM= 0.98898 CR=11.52
QM=0.99304 DI=19.4164

Figure 3.23: Results of the comparative study for image 1 and 2

Result of compression with our algorithm
PSNR=32.01 MSSIM= 0.99556 CR=14.459
QM=0.9845 DI=4.8824

Result of compression with the standard
PSNR=32.1282 MSSIM= 0.98652 CR=24
QM=0.98652 DI=4.922

Result of compression with JPEG standard
PSNR=29.3879 MSSIM= 0.98273 CR=26.18
QM=0.96294 DI=4.3102

Figure 3.24: Results of the comparative study for image 3

3.8 Conclusion

From all the obtained results, we conclude that YCbCr is the optimal color space, it gives high quality in terms of PSNR, SSIM and QM, and high compression ratio.

For images that have fewer details Level L=6 of Bior2.2 DWT is the best one. In fact, it gives the highest image quality and high compression ratio.

For high detailed images Level L=8 of Db1 provides the highest compression ratio and high image quality.

The Compression Ratio (CR) increases while Detail Index (DI) decreases.

The image quality in terms of PSNR, SSIM and QM decrease as the level of the wavelet increase.

General Conclusion

The main aim of our study was to optimize the JPEG2000 compression standard along with a brief overview of the standard itself so we concluded that:

- YCbCr is the optimal color space; it gives high quality in terms of PSNR, SSIM and QM, and high compression ratio.
- The Compression Ratio (CR) increases while Detail Index (DI) decreases.
- Db1 Discrete Wavelet Transform (DWT) gives the highest Compression Ratio.
- For images that have Detail Index (DI) < 30, Bior2.2 gives the highest image quality.
- For high detailed images that have Detail Index (DI) > 30, Db1 Discrete Wavelet Transform (DWT) gives the highest high image quality.
- Level L=6 of Bior2.2 is the best level for images with Detail Index (DI) < 30.
- Level L=8 of Db1 is the best level for images with Detail Index (DI) >30.
- Increasing the level of Discrete Wavelet Transform (DWT) result in increasing the compression ratio, and in decreasing the image quality.

From these results, we conclude that optimal JPEG2000 algorithm is as follow, first the original image is transformed to the YCbCr color space, then if the user wants a high compression ratio, the images are compressed using high level of Db1 wavelet, if the user wants a high image quality, images with Detail Index (DI) < 30 are compressed using bior2.2 wavelet, and images with Detail Index (DI) > 30 are compressed using high level of Db1.

In our study most of the images have Detail Index (DI) < 23, in order to improve our work, we have test on more images with Detail Index (DI) > 23, so that the results will be more accurate. We can also add the execution time as criteria in order to see which wavelet is faster.

References

[1] C. E. SHANNON, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, October, 1948.

[2] A. Tinku and . T. Ping-Sing, JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures, New Jersey: John Wiley & Sons, Inc, 2005.

[3] C. G. Rafael and E. W. Richard, Digital Image Processing 2nd Edition, Prentice Hall, 2004.

[4] L. Clifford, "A Study of the JPEG-2000 Image Compression Standard," Master Degree Thesis to Queen's University , Ontario, Canada, 2001.

[5] A. V. Oppenheim, S. Ronald W and B. John R, Discrete-time signal processing 2nd Edition, Prentice-Hall, 1989.

[6] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing* , vol. 13, p. 600–612, 01 04 2004.

[7] A. J. Madhuri, S. R. Mehul, H. D. Yogesh, R. J. Kalyani and P. M. Shilpa, Image and video Compression Fundamentals, Techniques,and Applications, Florida: CRC Press, 2015.

[8] W. Zhou, P. S. Eero and C. B. Alan, "Multi-Scale Structural Similarity For Image Quality Assessment," in *the 37th IEEE Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2003.

[9] K. Fatma and S. Amina, "A No-Reference Blur Image Quality measure Based on Wavelet Transform," in *Digital Information Processing and Communications (ICDIPC), 2012 Second International Conference* , Klaipeda City, 2012.

[10] A. H. Robinson and C. Cherry, "Results of a prototype television bandwidth compression scheme," in *Proceedings of the IEEE* , 1967.

[11] S. Malkit, "Review Of Lossless Image Compression," *International Journal of Scientific Research Engineering & Technology (IJSRET)*, vol. 5, no. 2, pp. 53-55, 2016.

[12] Y. Abdou, "Lecture Notes: Lossless Compression, Bit plane Coding," The George Washington University, Washington DC, Fall 2011. [Online]. Available: https://www.seas.gwu.edu/~ayoussef/cs225/lossless.html#bitplane. [Accessed 18 May

2016].

[13] M. John, Compressed Image File Formats JPEG, PNG, GIF, XBM, BMP, ACM Press, 1999.

[14] K. Weidong, Digital Image Compression Algorithms and Standards, Springer Science & Business Media, LLC, 1995.

[15] S.-C. Diego and E. Touradj, "A Study Of JPEG2000 Still Image Coding Versus Other Standards," in *Proc. of the X European Signal Processing Conference*, Tampere, 2000.

[16] B. Robert, "Preservation, JPEG 2000 - a Practical Digital Standard," Digital Preservation Coalition, New York, 2008.

[17] C. G. Rafael, Digital image processing using matlab, 2nd Edition, Dorling Kindersley Pvt Ltd , 2006.

[18] S. Mallat, "A theory for multiresolution signal decomposition:," *IEEE Transactions on Pattern*, vol. 11, no. 7, pp. 674-693, 1989.

[19] S.-C. Diego and E. Touradj, "An Analytical Study Of JPEG 2000 Functionalities," in *International Conference on Image Processing*, Vancouver, BC, 2000.

[20] M. K. Amr , W. M. Nagy , E.-F. Nawal A, A. A. Abdelrahman and H. M. Ahmed, "C10. Haar Wavelet Image Compression Algorithm (HW-JPEG2000)," in *32nd NATIONAL RADIO SCIENCE CONFERENCE*, Cairo, 2015.

[21] C. Chih-Chang, C. Oscal and W. Horng-Hsinn, "Region of Interest Determined by Picture Contents in JPEG 2000," Chia-Yi, 2003.

[22] M. Walaa and Abd-Elhafiez, "New Approach for Color Image Compression," vol. 3, no. 4, 2012.

[23] A.-E. M and G. Wajeb, "Color Image Compression Algorithm Based on the DCT Blocks," College of Computer Science & Information Systems, Jazan University, Jazan, 2012.

[24] H. Sheikh and Z.wang, Live image quality assessment database release 2, 2010.

[25] M.Mozammel, C. Hoque and K. Amina, "Image Compression Using Discrete Wavelet Transform," vol. 9, no. 4, 2012.

[26] Y. Abdou, "Lecture Notes: Lossless Compression, Bit plane Coding," The George Washington University, Washington DC, Fall 2011. [Online]. Available: https://www.seas.gwu.edu/~ayoussef/cs225/lossless.html#bitplane. [Accessed 18 May

2016].

[27] S.-C. Diego and E. Touradj, "A Sudy Of JPEG 2000 Still Image Coding Versus Other Sandards," in *Proc. of the X European Signal Processing Conference*, Tampere, 2000.

[28] S.-C. Diego and E. Touradj, "An Analytical Study Of JPEG 2000 Functionalities," in *International Conference on Image Processing*, Vancouver, BC, 2000.

[29] S. Malkit, "Review Of Lossless Image Compression," *International Journal of Scientific Research Engineering & Technology (IJSRET)*, vol. 5, no. 2, pp. 53-55, 2016.

[30] C. G. Rafael, Digital Image Processing Using Matlab, 2nd Edition, Dorling Kindersley Pvt Ltd, 2006.

[31] A. H. Robinson and C. Cherry, "Results of a Prototype Television Bandwidth Compression Scheme," in *Proceedings of the IEEE*, 1967.

[32] S.Medouakh and Z-E. Baarir, "Study of the Standard JPEG2000 in Image Compression," *International Journal of Computer Applications (0975 – 8887), Vol 18, no.1, March 2011*