



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

MASTER DISSERTATION

Science and Technology
Electrical Engineering
Automatic

Ref.: Enter the document reference

Presented and supported by:

SEBÀA Mohamed

BROUTHEN Wail

The:

GUI MATLAB For Image Compression

Jury:

Dr. NEBBAR Hanane	MCA	University Of Biskra	President
Dr. MECHGOUG Raihane	MCA	University Of Biskra	Examiner
Dr. ZITOUNI Athmane	MCA	University Of Biskra	Protractor

University Year: 2021 - 2022



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

MASTER DISSERTATION

Science and Technology
Electrical Engineering
Automatic

Ref.: Enter the document reference

GUI MATLAB For Image Compression

The:

Presented By:

SEBÀA Mohamed

BROUTHEN Wail

Favorable opinion of the supervisor:

Dr.ZITOUNI Athmane

Signature Favorable opinion of the President of the Jury

Dr.NABBAR Hanane

Stamp and signature

Abstract:

There are numerous applications of image processing, such as satellite imaging, medical imaging, and video where the image size or image stream size is too large and requires a large amount of storage space or high bandwidth for communication in its original form. Image compression techniques can be used effectively in such applications. In this paper it has been proposed the compression methods using the *wavelet* 2-D with the toolbox of MATLAB called 'true image compression' '*wcompress*' this last one uses more than one embedded coding methods we mention *EZW*, *SPIHT* we are going to see the results based on their performance through calculation compression ratio *CR* and *PSNR*. For this we are going to create an interface using *GUI* MATLAB the interface has all the necessary component to do the compression more easily and efficiently.

Keywords: Image compression, wavelet, EZW, SPIHT, PSNR, CR, GUI

Résumé:

Il existe de nombreuses applications de traitement d'images, telles que l'imagerie par satellite, l'imagerie médicale et la vidéo, où la taille de l'image ou la taille du flux d'images est trop grande et nécessite une grande quantité d'espace de stockage ou une bande passante élevée pour la communication dans sa forme d'origine. Les techniques de compression d'image peuvent être utilisées efficacement dans de telles applications. Dans cet article, il a été proposé les méthodes de compression utilisant l'*ondelette* 2-D avec la boîte à outils de MATLAB appelée 'true image compression' '*wcompress*' ce dernier utilise plus d'une méthode de codage embarquée nous mentionnons *EZW*, *SPIHT* nous allons voir les résultats en fonction de leurs performances grâce au calcul des taux de compression *RC* et *PSNR*. Pour cela, nous allons créer une interface à l'aide de *GUI* MATLAB. L'interface possède tous les composants nécessaires pour effectuer la compression plus facilement et plus efficacement.

Mot de clé: compression d'image, ondelette, EZW, SPIHT, PSNR, RC, GUI

ملخص:

هناك العديد من التطبيقات لمعالجة الصور، مثل التصوير عبر الأقمار الصناعية والتصوير الطبي والفيديو حيث يكون حجم الصورة أو حجم دفق الصورة كبيرًا جدًا ويتطلب مساحة تخزين كبيرة أو نطاقًا تردديًا عاليًا للاتصال في شكله الأصلي. يمكن استخدام تقنيات ضغط الصور بشكل فعال في مثل هذه التطبيقات. في هذا البحث تم اقتراح طرق الضغط باستخدام الموجة ثنائية الأبعاد مع صندوق أدوات للماتلاب المسمى "ضغط الصورة الحقيقي" "wcompress" يستخدم هذا الأخير أكثر من طريقة للترميز مضمنة نذكر منها EZW، SPIHT التي سنرى نتائجهم على أساس أدائهم من خلال حساب نسبة الضغط ونسبة تشويه الصورة. لهذا، سنقوم بإنشاء واجهة باستخدام واجهة المستخدم الرسومية للماتلاب، تحتوي الواجهة على كل المكونات الضرورية للقيام بالضغط بسهولة وكفاءة أكبر.

الكلمات المفتاحية: ضغط الصورة، الموجة، موجة صفرية مضمنة، تعيين التقسيم في هرمية الأشجار، نسبة تشويه الصورة، نسبة ضغط الصورة، واجهة المستخدم الرسومية.

DEDICATIONS

To Allah Almighty Who paved the way for me to accomplish this humble work

To my very dear parents, source of life, love and affection

To my dear sisters, source of joy and happiness

To all my family, especially Ousama

To all my friends, especially Islam and kamal

To Mohamed, dear friend before being a partner

Best Regards Wail,

DEDICATIONS

To Allah Almighty Who paved the way for me to accomplish this humble work

To my very dear parents, without their support I wouldn't be here

To my dear brothers and sister, source of joy and happiness

To all my family, source of hope and motivation

To all my friends, especially Islam, Smail, Abderazzak and Sohaib

To Wail, dear friend before being a partner

Best Regards Mohamed,

THANKS

First and foremost, my heartfelt gratitude and praises go to Allah Almighty it is in the right of the grace to be mentioned, and less Reward is gratitude.

We would like to extend our sincere thanks and appreciation to our honorable supervisor *Dr. ZITOUNI Athmane*, for his guidance and patience to us.

We sincerely thank the members of the jury both the president *Dr. NABBAR Hanane* and the examiner *Dr. MECHGOUG Rihane*, for having wanted to accept to be part of the commission of examiner.

We also thank our mothers, our fathers, and our brothers and sisters for their invaluable support.

And for our honorable friend *ISLAM*, we wish him the best and the success in his life, you're the best, it has been 3 tough years for us, but we passed it successfully alhamdulillah.

SUMMARY

SYMBOLS & ANNOTATION..... V

LIST OF FIGURES AND TABLES VI

GENERAL INTRODUCTION X

CHAPTER I: GENERALITY ON DIGITAL IMAGE COMPRESSION..... 1

I.1 INTRODUCTION 1

I.2 DIGITAL IMAGE 1

 I.2.1 Definition..... 1

 I.2.2 Pixel 2

 I.2.3 Digital Image Resolution 2

 I.2.4 Types Of Digital Graphics..... 3

 I.2.4.1 Raster (Bitmap) 3

 I.2.4.2 Raster File Types..... 3

 I.2.4.3 Vector 4

 I.2.4.4 Vector File Types 4

 I.2.5 Color Coding 5

 I.2.5.1 Black & white Image..... 5

 I.2.5.2 Shade of grey 5

 I.2.5.3 Colored Image 5

 I.2.5.4 RGB Model 5

I.3 DATA COMPRESSION 6

I.4 IMAGE COMPRESSION..... 7

 I.4.1 Definition..... 7

 I.4.2 Image Compression Block Diagram..... 7

 I.4.2.1 Forward Transform..... 7

 I.4.2.2 Quantization 8

 I.4.2.3 Entropy coding 8

 I.4.3 Compression Technique 8

 I.4.4 Lossless Compression..... 9

 I.4.5 Lossless Methods..... 10

 I.4.5.1 Coding of Shannon-Fano..... 10

 I.4.5.2 Huffman coding..... 10

 I.4.5.2.1 LZW coding 12

 I.4.5.2.2 Area coding 12

 I.4.5.3 Arithmetic coding..... 13

Summary

I.4.5.3.1 LZW coding	14
I.4.5.3.2 Area coding	14
I.4.5.4 JPEG STANDARD	14
I.4.6 Lossy Compression.....	14
I.4.7 Lossy Methods.....	15
I.4.7.1 Transform based Methods	15
I.4.7.1.1 DWT-Based Compression System.....	15
I.4.7.2 Sub-band Coding	15
I.4.7.3 The predictive method.....	16
I.4.7.4 Fractal compression.....	16
I.4.7.5 Standard H-261 compression	16
I.4.7.6 Hybrid Methods.....	16
I.4.8 Embedded coding	16
I.4.8.1 JPEG 2000 Standard.....	17
I.4.8.2 Principle of JPEG2000	18
I.5 CONCLUSION	18
CHAPTER II: IMAGE COMPRESSION USING 2-D WAVELET	19
II.1 INTRODUCTION	19
II.2 FOURIER TRANSFORM.....	19
II.2.1 principle	20
II.2.2 Short-time Fourier transform (STFT).....	20
II.3 WAVELET TRANSFORM	21
II.3.1 Haar Wavelet	23
II.3.2 Daubechies wavelet	24
II.3.3 Biorthogonal wavelet.....	24
II.3.4 Morlet Wavelet	25
II.4 CONTINUOUS WAVELET TRANSFORM (CWT).....	26
II.5 DISCRETE WAVELET TRANSFORM	26
II.6 MULTIREOLUTION ANALYSIS.....	27
II.6.1 Approximation spaces.....	28
II.7 THE PYRAMID OF THE IMAGE.....	29
II.8 EMBEDDED CODING METHODS	31
II.8.1 Embedded Zerotrees Wavelet (EZW).....	31
II.8.1.1 Encoding a coefficient of the significance map.....	32
II.8.1.1.1 Zerotree root	32
II.8.1.1.2 Isolated zero.....	33

Summary

II.8.1.1.3 Positive significant coefficient	33
II.8.1.2 Defining Threshold.....	33
II.8.1.2.1 Initial threshold T_0	33
II.8.1.2.2 Threshold T_1	33
II.8.1.3 Scanning order for coefficients	33
II.8.1.3.1 Negative significant coefficient.....	33
II.8.2 Set Partitioning in Hierarchical Trees (SPIHT).....	34
II.9 CONCLUSION	34
CHAPTER III: GRAPHIC USER INTERFACE (GUI) MATLAB.....	35
III.1 INTRODUCTION	35
III.2 THE GUI PRESENTATION	35
III.2.1 The Components.....	36
III.2.2 The Figures	36
III.2.3 Callbacks (callback service)	36
III.2.4 Launch GUI MATLAB	36
III.3 THE LAYOUT WINDOW.....	38
III.4 PRESENTATION OF THE DIFFERENT INTERFACE OBJECTS	40
III.5 USED FUNCTIONS	41
III.6 CONCLUSION	42
CHAPTER IV: RESULTS AND DISCUSSION.....	43
IV.1 INTRODUCTION.....	43
IV.2 RESEARCH STEPS AND METHOD OF WORK	43
IV.2.1 Compression Ratio.....	44
IV.2.2 Mean square error (MSE)	44
IV.2.3 MATLAB wavelet toolbox.....	45
IV.2.4 MATLAB GUI (Graphic User Interface)	46
IV.2.4.1 The interface manipulation.....	47
IV.3 PRACTICAL WORK.....	51
IV.3.1 Wavelet Filters.....	54
IV.3.2 Decomposition levels.....	54
IV.4 RESULTS AND DISCUSSION	55
IV.4.1 Lena Image (256x256) 8-bit Results Table	55
IV.4.2 The Reconstructed Image Lena (256x256) 8-bit	56
IV.4.3 The Effect of wavelets on Lena Image (256x256) 8-bit.....	57
IV.4.4 Barbara Image (512x512) 8-bit Results Table.....	58

Summary

IV.4.5	The Reconstructed Image Barbara (512x512) 8-bit.....	59
IV.4.6	The Effect of wavelets on Barbara Image (512x512) 8-bit	60
IV.4.7	X-ray Image (512x512) 24-bit Results Table	60
IV.4.8	The Reconstructed Image X-ray (512x512) 24-bit.....	61
IV.4.9	The Effect of wavelets on X-ray Image (512x512) 24-bit.....	63
IV.4.10	Cameraman Image (256x256) 8-bit Results Table.....	63
IV.4.11	The Reconstructed Image Cameraman (256x256) 8-bit.....	64
IV.4.12	The Effect of wavelets on Cameraman Image (512x512) 24-bit	66
IV.4.13	Woman Image (256x256) 8-bit Results Table	66
IV.4.14	The Reconstructed Image Woman (256x256) 8-bit	67
IV.4.15	The Effect of wavelets on Woman Image (256x256) 8-bit	69
IV.4.16	Peppers Image (512x512) 24-bit Results Table	69
IV.4.17	The Reconstructed Image Peppers (512x512) 24-bit	70
IV.4.18	The Effect of wavelets on Peppers Image (512x512) 24-bit.....	72
IV.5	DISCUSSION AND CONCLUSION.....	72
IV.6	BASED ON RESULTS	73
IV.6.1	EZW methods performance	73
IV.6.2	SPIHT methods performance.....	73
IV.7	CONCLUSION	74
	GENERAL CONCLUSION	75
	BIBLIOGRAPHY AND WEBOGRAPHY	76

Symbols & Annotation

- A** : Approximation.
- DPI** : Dots Per Inch.
- PPI** : Pixel Per Inch.
- CR** : Compression Ratio.
- PSNR** : Peak Signal to Noise Ratio.
- MSE** : Mean Squared Error.
- FT** : Fourier Transform.
- DWT** : Discrete Wavelet Transform.
- CWT** : Continuous Wavelet Transform.
- DV** : Detail vertical.
- DH** : Detail horizontal.
- DD** : Detail Diagonal.
- JPEG** : Joint Photographic Expert Group.
- RGB** : Red-Green-Blue.
- STFT** : Short Time Fourier Transform.
- LZW** : Lempel–Ziv–Welch.
- MAR** : Multiresolution Analysis.
- RLE** : Run Length Encoding.
- BPP** : Bit-per-pixel.
- EZW** : Embedded Zerotree Wavelet.
- SPIHT** : Set Partitioning in Hierarchical Trees.
- GUI** : Graphic User Interface.

List Of Figures and tables

Figure I.1 Digital Image Representation	1
Figure I.2 Pixel Representation	2
Figure I.3 Digital Image Resolution in PPI & DPI	3
Figure I.4 Digital Graphic types Vector & Raster.....	3
Figure I.5 RGB Model.....	6
Figure I.6 A Representation for Data Compression	7
Figure I.7 A general block diagram of image compression systems: (a) encoder and (b) decoder.....	7
Figure I.8 Lossy and Lossless Compression	9
Figure I.9 Shannon-Fano coding	10
Figure I.10 Huffman coding.....	11
Figure I.11 Huffman coding Compression algorithm	11
Figure I.12 Arithmetic coding	13
Figure I.13 A general diagram of the CCSDS-IDC encoder.....	15
Figure I.14 The three-level wavelet decomposition of Lena image.....	17
Figure I.15 Accelerating JPEG 2000 Decoding for Digital Pathology and Satellite Images.....	17
Figure II.1 Fourier Transform	20
Figure II.2 Short-time Fourier transform STFT	21
Figure II.3 Time frequency tiling Wavelet Transform.....	23
Figure II.4 Haar Wavelet.....	23
Figure II.5 Daubechies Wavelet.....	24
Figure II.6 Biorthogonal Wavelet	25
Figure II.7 Morlet Wavelet	25
Figure II.8 Discrete Wavelet Transformation	27
Figure II.9 Multiresolution Analysis using wavelet transform	28
Figure II.10 Pyramid algorithm (Fast Mallat algorithm)	30
Figure II.11 Analysis of three level of Image decomposition.....	30
Figure II.12 Pyramid algorithm (Fast Mallat algorithm) the inverse.....	31

List of figures and tables

Figure II.13 Embedded Zerotrees of Wavelet Sub-band	32
Figure II.14 The repartition used by SPIHT algorithm.....	34
Figure III.1 MATLAB Command line.....	36
Figure III.2 Guide Quick Start.....	37
Figure III.3 The layout Window	37
Figure III.4 Layout window showcasing the interface component.....	39
Figure III.5 Different Object interface.....	40
Figure III.6 The generated interface	40
Figure III.7 The generated editor	41
Figure III.8 The generated editor	42
Figure IV.1 Wavelet Compression for Images.....	45
Figure IV.2 GUI (Graphic User Interface) MATLAB for the image compression	47
Figure IV.3 GUI (Graphic User Interface) MATLAB load image	48
Figure IV.4 GUI (Graphic User Interface) MATLAB set the Method Name.....	48
Figure IV.5 GUI (Graphic User Interface) MATLAB set the Filter Wavelet.....	49
Figure IV.6 GUI (Graphic User Interface) MATLAB set the number of encoding loop	49
Figure IV.7 GUI (Graphic User Interface) MATLAB results after click on compress button.....	50
Figure IV.8 GUI (Graphic User Interface) MATLAB save the compressed image ...	50
Figure IV.9 The Original Image Lena (512x512) 8-bit.....	51
Figure IV.10 The Original Image Barbara (512x512) 8-bit.....	52
Figure IV.11 The Original Image X-ray (512x512) 24-bit	52
Figure IV.12 The Original Image Cameraman (256x256) 8-bit	53
Figure IV.13 The Original Image Woman (256x256) 8-bit.....	53
Figure IV.14 The Original Image Peppers (512x512) 24-bit.....	54
Figure IV.15 The Reconstructed image Lena (265x256) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods	57
Figure IV.16 Comparing Different wavelets for Lena Image for both EZW & SPIHT	57
Figure IV.17 The Reconstructed image Barbara (512x512) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods	60
Figure IV.18 Comparing Different wavelets for Barbara Image for both EZW & SPIHT	60

List of figures and tables

Figure IV.19 The Reconstructed image X-ray (512x512) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods	62
Figure IV.20 Comparing Different wavelets for X-ray Image for both EZW & SPIHT	63
Figure IV.21 The Reconstructed image Cameraman (256x256) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods	65
Figure IV.22 Comparing Different wavelets for Cameraman Image for both EZW & SPIHT	66
Figure IV.23 The Reconstructed image Woman (256x256) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods	68
Figure IV.24 Comparing Different wavelets for Woman Image for both EZW & SPIHT	69
Figure IV.25 The Reconstructed image Peppers (512x512) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods	71
Figure IV.26 Comparing Different wavelets for Peppers Image for both EZW & SPIHT	72
Table IV-1 Types of wavelets used.....	54
Table IV-2 Lena Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps	55
Table IV-3 Lena Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps	55
Table IV-4 Lena Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps.....	55
Table IV-5 Barbara Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps	58
Table IV-6 Barbara Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps.....	58
Table IV-7 Barbara Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps	58
Table IV-8 X-ray Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps	60

List of figures and tables

<i>Table IV-9 X-ray Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps</i>	61
<i>Table IV-10 X-ray Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps</i>	61
<i>Table IV-11 Cameraman Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps</i>	63
<i>Table IV-12 Cameraman Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps</i>	64
<i>Table IV-13 Cameraman Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps</i>	64
<i>Table IV-14 Woman Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps</i>	66
<i>Table IV-15 Cameraman Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps</i>	67
<i>Table IV-16 Cameraman Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps</i>	67
<i>Table IV-17 Peppers Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps</i>	69
<i>Table IV-18 Peppers Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps</i>	70
<i>Table IV-19 Peppers Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps</i>	70

General Introduction

With the development that the world is witnessing thanks to the scientific revolution, especially in the field of communications, this technology has made the ability to store and process a large amount of data (text, image, audio or video content...) and with the large use of the digital image The amount of data has become so huge that it is necessary to find a solution Optimizing the reduction of data volume instead of increasing the storage capacity, and this led the researchers to resort to The compression method is an optimum solution for archiving images on the one hand, and liquify it and send it on the other.

Classification of compression methods according to the quality of the data into compression lossless. It is characterized by a high-quality image with low pressure rates and is suitable for sensitive applications such as medical images, text compression, and others, for example: RLE (Encoding Length Run) technology. Huffman coding, Welch-Ziv-Lempel (LZW) method, and when data require High pressure, we resort to the other method, which is pressure loss, which leads to Loss of information and maintaining.

In this thesis, we will mainly focus on the first of these problems, which is that of the representation of digital images. In most cases, problems related to its transmission or storage are assumed to be independent and fall within the scope of telecommunications. Compression consists in finding how to describe in the most the information as succinct as possible, possibly allowing themselves to downgrade it. This treatment not only reduces the number of elements needed to represent it, but also to simplify subsequent processing by condensing the information. In the case particular of the image, the compression endeavors to extract the visual information from the data raw received from the camera. Since simply scanning the image degrades it, another essential objective of compression is to find the best compromise between the quantity of information retained and the visual impact of the degradations made on the image. Finally, due to of the mass of data to be processed, a compromise is often necessary between the complexity of the operations carried out and the quality of the results obtained.

Several algorithms are proposed for applying pressure with loss, one of them who achieved advantages is wavelet transformation (small wave) Discrete Wavelet Transform

General Introduction

(DWT) that allows to represent a signal with a finite number of Values, such as the EZW (Wavelet Zerotree Embedded) encoding proposed by Shapiro in 1993

In this context, we have considered compression algorithms by wavelets and by embedded encodings (EZW, SPIHT). The embedded encoder offers the transmission property of the coded image while providing excellent, the results obtained by our new approach in terms of PSNR and compression ratio obtained are better especially for medium and high speeds without affecting the time Calculation. Finally, our results are comparing algorithms SPIHT and EZW

In this thesis, we proposed an easy-to-use interface for image compression using GUI MATLAB so we can have more control and functionality on hand, this can benefit a user who has no programming knowledge or MATLAB coding language to get more done in few click of a button.

Our thesis is organized to chapter such as:

In the first chapter, we talk about Generality on Digital Image compression,

- We'll mentioning digital images (pixel, resolution, ...etc)
- Data compression
- Image compression (quantification, encoding, lossless Methods lossy; ...etc)

The second chapter we'll talking about Image compression using 2-D wavelet.

- Fourier Transform, Wavelet transform, DWT
- Multiresolution Analysis
- Embedded coding, EZW, SPIHT

The third chapter is for Methodology Graphic user interface GUI MATLAB

The Final chapter is Results and discussing.

We end our thesis with a general conclusion and the perspectives envisaged about our work.

Chapter I: Generality on Digital Image Compression

I.

I.1 Introduction

Compression and encoding methods reduce the average number of bits per pixel to store or transmit, exploiting the informational redundancy of the image. For this purpose, several scientific researches have contributed to the birth of various methods of compression. Note that image compression is divided into two main axes: lossless and lossy compression. The first type of compression uses only the principle of the reduction of information and does not generate loss, the second type, defines an approximate representation of the information. In this chapter we will start by defining the notion of digital image and data compression. with its two types, then we will detail each type with its methods.

I.2 Digital Image

I.2.1 Definition

A digital image can be considered as a discrete representation of information having both spatial (layout) and escalated (color) data. ready to moreover consider it treating an image as a multidimensional signal [1].

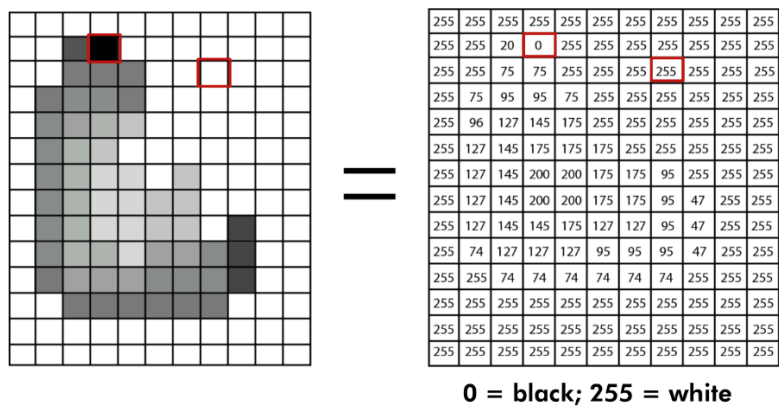


Figure I.1 Digital Image Representation

As you've seen in the top figure, the image is made up of pixels.

CHAPTER I : Generality On Digital Image Compression

I.2.2 Pixel

A pixel is the littlest unit of a digital image or graphic that can be shown and represented. A pixel is the essential consistent unit in digital graphics. Pixels are combined to make any obvious thing on a computer display. A pixel is additionally known as a picture element [2].

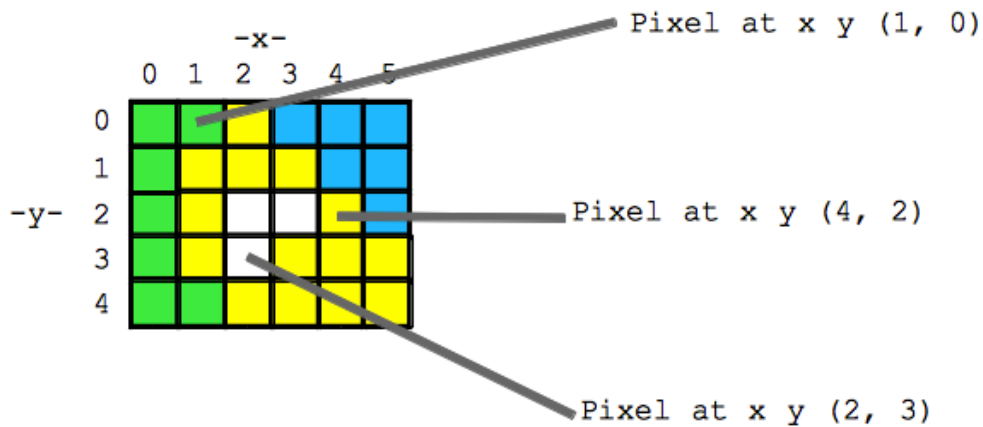


Figure I.2 Pixel Representation

Does the size number of these pixels matter?

I.2.3 Digital Image Resolution

The size of the 2-D pixel grid together with the data size stored for each individual image pixel determines the spatial resolution and color quantization of the image [1].

Millions of pixels that make up the image sensor in a digital camera are really little light-sensitive squares. These pixels in a photograph cannot be recognized unless you amplify a digital image. These boxes are known as the pixels.

Resolution is the camera's capacity to classify and successfully show discrete image data, such as details, patterns and textures within a given photographic image and it corresponds to how large a photo can become without becoming unacceptably blurry or grainy. Camera and image resolution is measured in Pixels Per Inch or **PPI** and the printer resolution is measured in Dots Per Inch or **DPI** [3].

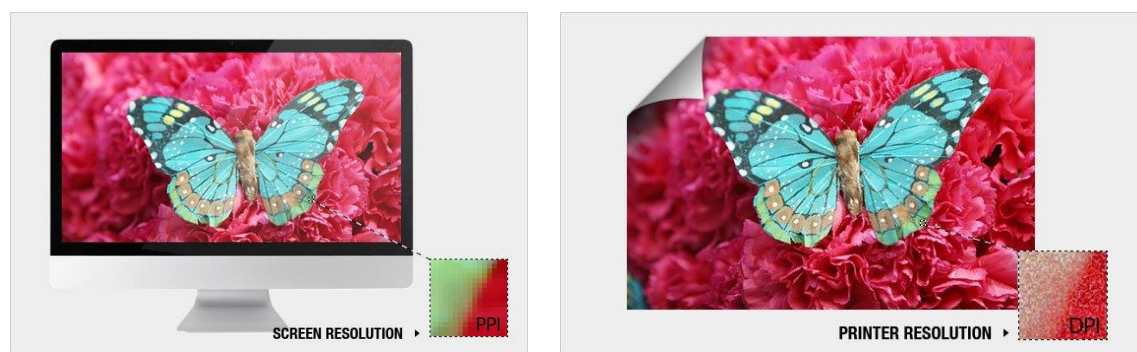


Figure I.3 Digital Image Resolution in PPI & DPI

I.2.4 Types Of Digital Graphics

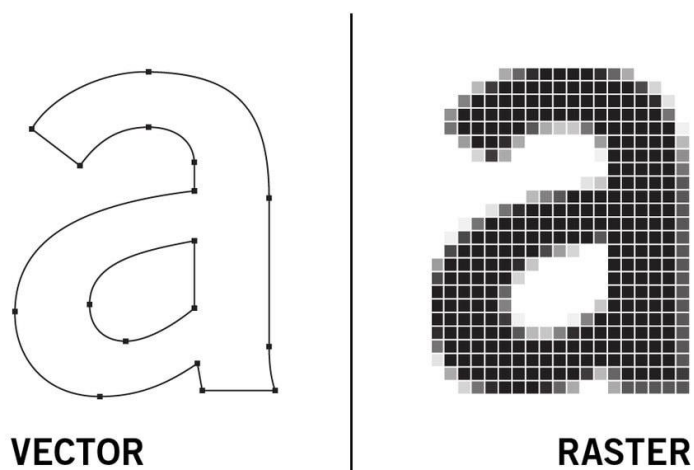


Figure I.4 Digital Graphic types Vector & Raster

I.2.4.1 Raster (Bitmap)

Raster images are often called bitmap images because they are made of millions of tiny squares [4], These squares matrix data structures are mainly a representation of a rectangular grid of pixels that can be viewed through a monitor screen, paper, or any other display medium.

I.2.4.2 Raster File Types

Raster files are mainly intended for computer users. That's why they are saved JPEG, PNG or GIF format [5].

- *JPG/JPEG – Joint Photographic Experts Group:*

CHAPTER I : Generality On Digital Image Compression

In most common use today; lossy compression; lossless variants exist [1], one of the most common image formats, JPEG are mainly seen online, JPEG offers compression without degrading the image and maintains a smaller file size that doesn't take much time to load [5].

- ***PNG – Portable Network Graphics:***

New lossless compression format; designed to replace GIF [1], PNG images are more commonly known for allowing transparent background as compared to JPEG images [5].

- ***GIF – Graphics Interchange Format:***

Unlike vector images, raster image quality is dependent on the resolution. This means that they cannot be scaled to randomly chosen resolutions without compromising on image quality [5]. Limited to only 256 colors (8-bit); lossless compression [1].

- ***TIF/TIFF – Tagged image (file) format***

Highly flexible, detailed and adaptable format; compressed/uncompressed variants exist [1]

- ***BMP – Bit map picture***

Basic image format; limited (generally) lossless compression; lossy variants exist [1]

I.2.4.3 Vector

Vector images, which are made of thin lines and curves known as paths, are rooted in mathematical theory [4]. Vector graphics or simply vectors are a computer graphic image type that is defined in terms of 2D pointers connected by lines and curves to form many different shapes [5].

I.2.4.4 Vector File Types

Nowadays, Vectors can be commonly found in numerous graphic file formats like SVG, EPS, PDF and AI [5].

- ***SVG – Scalable Vector Graphics:***

With newer technology being introduced with higher screen resolution, SVG is becoming more commercial in use [5].

- ***EPS – Encapsulated PostScript:***

CHAPTER I : Generality On Digital Image Compression

EPS is the most popular and widely used vector file format. The reason behind this particular type being the most popular is that the broadest range of software can open it.

I.2.5 Color Coding

An image appears as a matrix where each box contains associated numbers to one color. Usually there are 3 main types of colors for an image digital:

I.2.5.1 Black & white Image

Black and white is the simplest. The content of each box in the matrix is either a 0 (black) or 1 (white)[6]. The number of colors is only 2 and the rendering of the image the least effective but sometimes sufficient in the context of scriptural documents, for example.

I.2.5.2 Shade of grey

The so-called grayscale coding makes it possible to obtain more nuances than simple black and white. It offers additional possibilities to code the intensity level bright[6]. The color is often coded on one byte, i.e., 8 bits, which offers the possibility of obtaining 256 levels of gray (0 for black and 255 for white). We can also do it with 16 levels of gray (4 bits).

I.2.5.3 Colored Image

The color of a pixel is obtained, as a painter would, by mixing basic colors. It is not a question here of describing all the techniques used. We will describe one of the most commonly used principles which is that of synthesis additive.

I.2.5.4 RGB Model

The principle consists in mixing the 3 colors: red, green and blue (noted RGB or RGB in English), With the help of these 3 colors, we obtain a whole palette of shades ranging from black to White. Each color is associated with a byte (therefore 256 brightness levels) of each of the basic colors

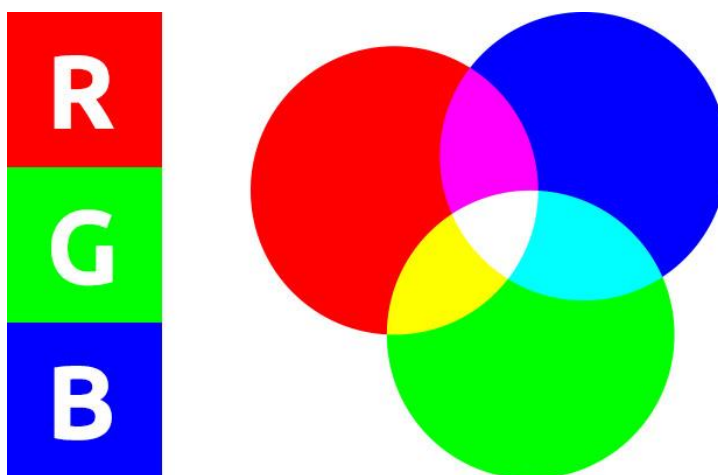


Figure 1.5 RGB Model

A 'color' pixel is then coded with 3 bytes and we then have the possibility of obtaining 224 color possibilities of the order of 16 million different colors. To reduce the space occupied by the color information, we use a palette of colors "attached" to the image. We then speak of indexed colors: the value associated with a pixel no longer conveys the actual color of the pixel, but returns the input corresponding to this value in a table (or palette) of colors, in which we have the complete representation of the color considered. Depending on the number of colors present in the image, we can thus gain a place not negligible: it is considered in practice that 256 colors among the 16 million colors 24 bits are sufficient

I.3 Data Compression

Compression can be characterized as a framework whose input is data, without compression and the output could be a generally brief computerized information stream representing the compressed data. The reverse process is called decompression allowing reconstruction of the picture from the advanced information stream. In some cases, the systems compression and decompression sets are called "Codec" (coding for compression and interpreting for decompression) [6].

CHAPTER I : Generality On Digital Image Compression

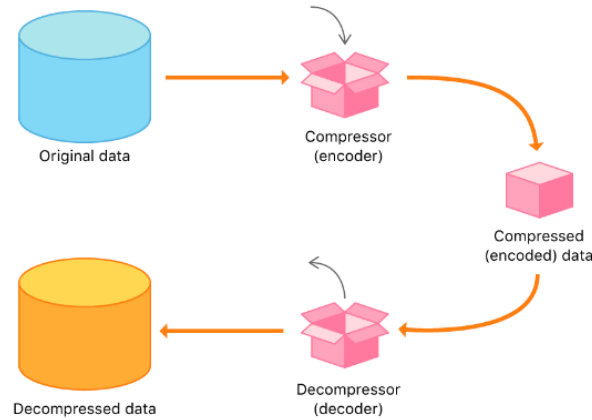


Figure I.6 A Representation for Data Compression

I.4 Image Compression

I.4.1 Definition

Compression and encoding methods reduce the number of bits per pixel to be stored or to transmit, by exploiting the informational redundancy in the image.

I.4.2 Image Compression Block Diagram

The block diagram of the compression is shown below:

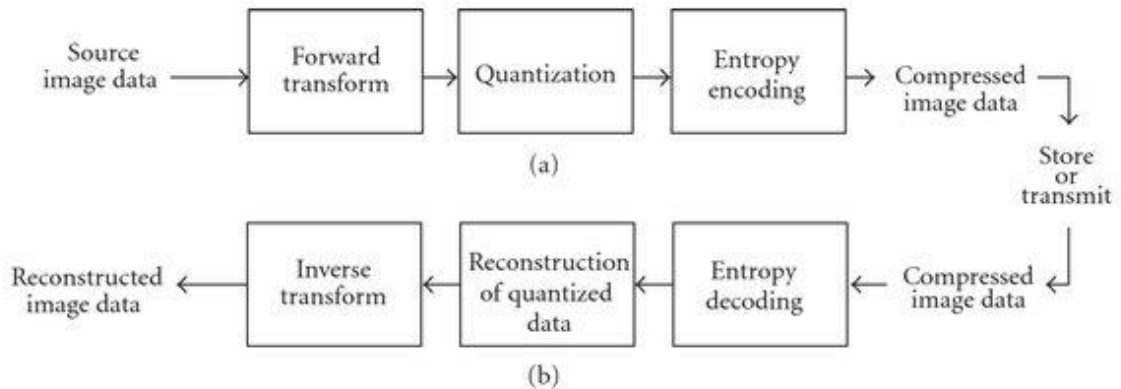


Figure I.7 A general block diagram of image compression systems: (a) encoder and (b) decoder.

I.4.2.1 Forward Transform

Transform-based compression frameworks are based on the understanding that the decorrelated coefficients of a change can be coded more effectively than the first picture pixels [7]. The existing reliance between each pixel and its neighbors (brightness shifts little

CHAPTER I : Generality On Digital Image Compression

from one pixel to an adjoining pixel) reflects an awfully solid relationship within the picture. So, we attempt to take a few of this relationship to decrease the sum of information by performing a de-correlation of the pixels.

The purpose of a transformation is to project the signal onto a basis of functions whose properties are adapted to the nature and characteristics of the signal or image that is want to analyze.

- Discrete Fourier Transform (DFT).
- Wavelet transformation (WT):
- Continuous wavelet transforms
- Discrete wavelet transforms.

I.4.2.2 Quantization

The point of quantizing the coefficients is to decrease the number of bits required for their representation. It could be a key step of compression. It approximates each esteem of a pixel by an integer numerous of an amount q , called basic quantum or quantization step. It may be scalar or vector. The leading execution is gotten by utilizing vector quantization.

I.4.2.3 Entropy coding

Entropic codes are based on the generation of words whose length depends on the probability of appearance of the symbols of the source that it represents (we also speak of variable length codes): a large number of bits will be used to code run symbol few probable while a redundant symbol will be coded on very few bits. This is the basic principle of information theory. The code thus obtained must be moreover only decodable.

Once the coefficients are quantized, they are coded. An encoder must fulfill a priority the following two conditions:

- Uniqueness: two distinctive messages ought to not be coded the same way.
- Decipherability: two progressive words codes must be particular and with no unambiguously.

I.4.3 Compression Technique

Image compression calculations are broadly classified into lossy and lossless technique as appeared in Figure. Misfortune less compression is appropriate for applications where picture debasement isn't tolerable i.e., field of therapeutic imaging [8]. The pictures which

CHAPTER I : Generality On Digital Image Compression

are compressed utilizing lossy techniques will not be remade as exactly as the initial picture. It is reasonable for applications where misfortune of redundant information is middle of the road. Lossy compression procedures are encourage classified into change based and non-transform-based techniques. The crude input picture is changed in to a comfortable shape so that machines can get to and read effectively. The spatial information show within the picture is gathered in to a frame based on the recurrence of event of pixel information. This approach is named as change-based compression.

As shown in the figure below:

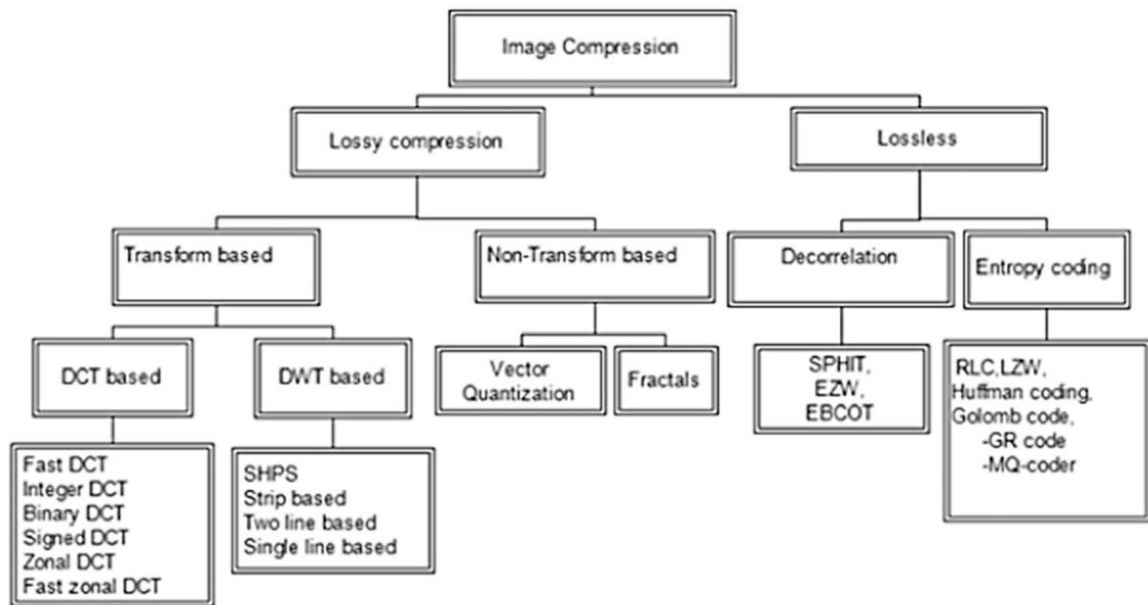


Figure I.8 Lossy and Lossless Compression

I.4.4 Lossless Compression

They allow recovering exactly the original pixels of the digital image [9]. Compression is said to be lossless when there is no loss of data on the original information. There is as much information after compression as before. The information to be compressed is seen as the output of a source of symbols which produces finished texts according to certain rules. The goal is to reduce the average size of the texts obtained after compression while having the possibility of finding exactly the original message.

I.4.5 Lossless Methods

I.4.5.1 Coding of Shannon-Fano

Used in the fifties, the Shannon-Fano code is the first code to have exploited the redundancy of a source[6]. All the symbols to be compressed are sorted according to their probability, and the sorted set of symbols is cut into two parts in such a way that the probabilities of the two parts are as close as possible to equality (the probability of a game being equal to the sum of the probabilities of the different symbols in this part). All the symbols of the first part are coded by a 0 followed by their Shannon-Fano code taking into account only the symbols of the first part, and all the symbols of the second part are coded by a 1 followed by their Shannon-Fano code taking into account only the symbols of the second part, recursively. When a part contains only one symbol, this one is represented by an empty code (zero length).

a_i	$p(a_i)$	1	2	3	4	Code		
a_1	0.36	0	00			00		
a_2	0.18		01			01		
a_3	0.18	1	10			10		
a_4	0.12		11	110			110	
a_5	0.09			111	1110			1110
a_6	0.07				1111			1111

Figure I.9 Shannon-Fano coding

I.4.5.2 Huffman coding

The most popular entropy-based encoding technique is the Huffman code. It provides the least amount of information units (bits) per source symbol. This short article describes how it works [10]. The first step in the Huffman algorithm consists in creating a series of source reductions, by sorting the probabilities of each symbol and combining the (two) least probable symbols into a single symbol, which will then be used in the next source reduction stage.

The Huffman coder is a binary tree ordered by all the symbols and by their occurrence frequencies. The two least frequent symbols in the string are connected by their ‘Parents’ by summing their frequencies. The symbols then take the values 0 and 1. The process is repeated

CHAPTER I : Generality On Digital Image Compression

on all the symbols until there are no more only one parent symbol forming the root of the binary tree.

Original source		Source reductions			
Symbol	Probability	1	2	3	4
a ₂	0.4	0.4	0.4	0.4	0.6
a ₆	0.3	0.3	0.3	0.3	0.4
a ₁	0.1	0.1	0.2	0.3	
a ₄	0.1	0.1	0.1		
a ₃	0.06	0.1			
a ₅	0.04				

Figure I.10 Huffman coding

To determine the code of a character, we start from the root and we go up until character. Each 0 or 1 encountered constitutes the code. For each character, the following codes are obtained:

C: 101 D: 110 B: 100 E: 111 A: 0

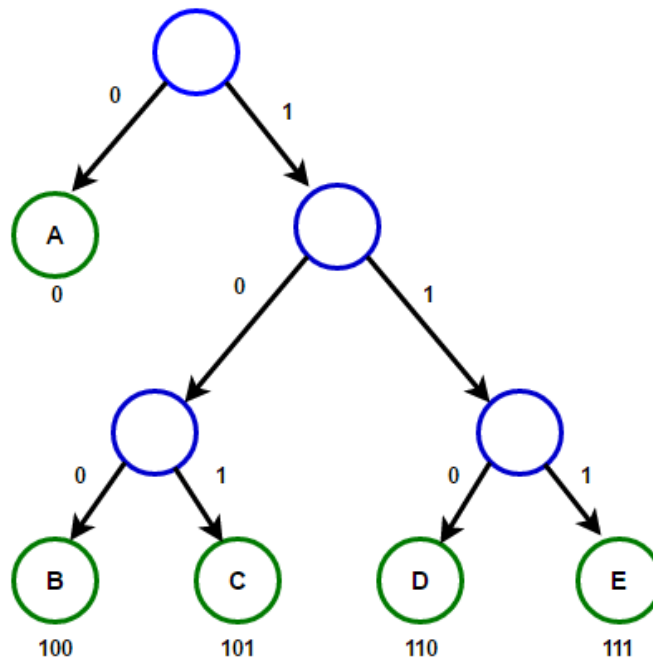


Figure I.11 Huffman coding Compression algorithm

The message therefore takes 23 bits, or an average of 2.1 bits/character. Without compression, with 8 bits per character, this message would fit on 184 bits[6]. However, the

CHAPTER I : Generality On Digital Image Compression

file will count more than 23 bits since we must send in the header of the file each character and the number of occurrences. Because of the header, very small files have a rate of uninteresting compression. However, large files will compress a lot because the size of the header is not proportional to the size of the file. A file of 1KB and a 1MB file have a good chance of having similar sized headers • Advantages and disadvantages Huffman encoding offers optimal character-by-character compression, i.e., shorter binary codes cannot be obtained for characters. He also has the advantage of being easy to implement by programming, and the execution time is rather fast.

Huffman coding can reduce the file size by 10% to 50% by removing the irrelevant information. In this technique smaller bit code is given to the pixel values which occur frequently and the higher bit code for repeated pixel value [11]. In order to encode images

- First of all, image is divided in to 8X8 blocks
- Then each block is coded with particular symbols • Huffman code is applied to each block
- Encoding all the blocks 2. Arithmetic Encoding Arithmetic encoding was introduced by Rissanen in which the last symbol is encoded and decoded first.
- Arithmetic encoding is based on following principle.
- The symbol alphabet should not infinite.
- All possible symbol sequence of give length should not infinite
- The number of real numbers in the interval $[0,1]$ can assign a unique subinterval for any given input sequence of symbols.

I.4.5.2.1 LZW coding

It is dictionary-based coding, in the static dictionary coding the dictionary is fixed during the encoding and decoding while in dynamic dictionary coding the dictionary is updated when new word is introduced.

I.4.5.2.2 Area coding

It is an enhanced version of the RLE. Area coding is highly effective and it can give better compression ratio but it has certain limitation that it can be applied to nonlinear transformation

- Encoding all the blocks

I.4.5.3 Arithmetic coding

Arithmetic coding is a data compression technique that encodes data by creating a code which represents a fraction in the unit interval [9]. While like the Huffman coder, unlike the latter, it produces a code for a whole sequence of symbols, not one code per symbol. Each new symbol read incrementally modifies the exit code. This code output is a floating-point number between 0 and 1, whose number of digits after the comma is the number of symbols. Unlike Huffman, he is not mandatory that each code has an integer number of bits. For example, a symbol of probability 0.9 has entropy 0.15, but Huffman will probably assign a one-bit code (or more), and the encoded sequence will have a longer number of bits than in theory. Moreover, the code for a sequence of symbols is an interval whose length decreases as we add more symbols to the sequence. This property allows us to have a coding scheme that is incremental, that is, the code for an extension to a sequence can be calculated simply from the code for the original sequence. As described in the above example, the interval in which the tag for a particular sequence resides is disjoint from all intervals in which the tag for any other sequence may reside[9]. Hence, any number in this interval can be used as a tag since the intervals for distinct sequences are disjoint, and one possible choice is the midpoint of the interval. In the following discussion, we will use the midpoint of the interval as the tag.

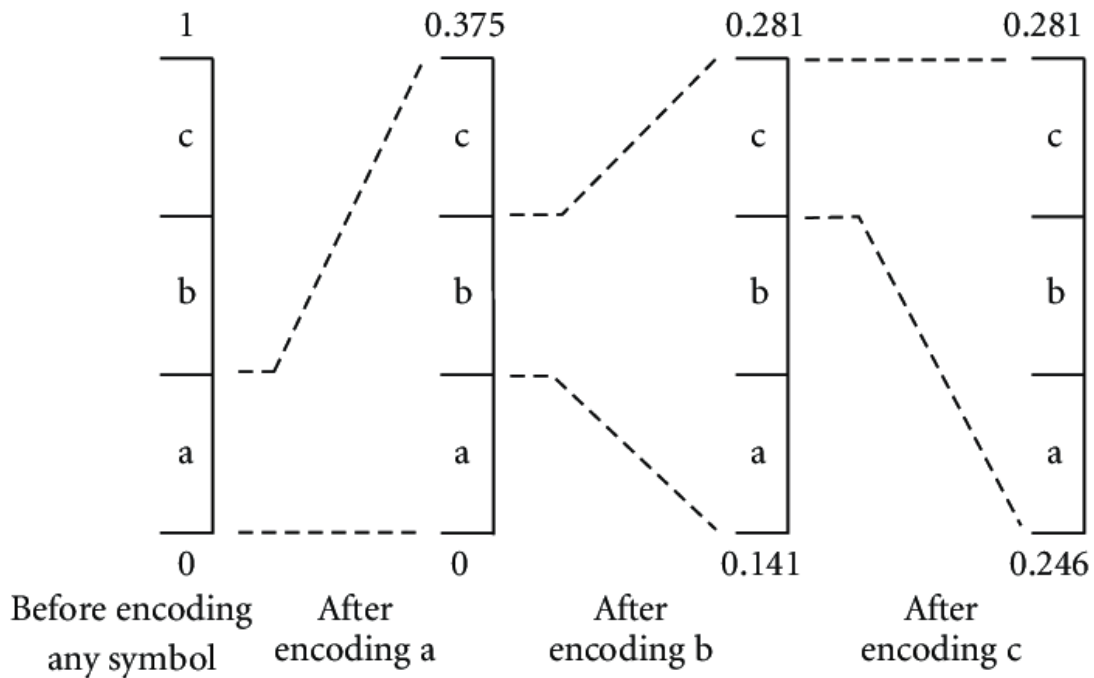


Figure I.12 Arithmetic coding

CHAPTER I : Generality On Digital Image Compression

- The symbol alphabet should not infinite.
- All possible symbol sequence of give length should not infinite
- The number of real numbers in the interval $[0,1]$ can assign a unique subinterval for any given input sequence of symbols.

I.4.5.3.1 LZW coding

It is dictionary-based coding, in the static dictionary coding the dictionary is fixed during the encoding and decoding while in dynamic dictionary coding the dictionary is updated when new word is introduced. 4. Area coding It is an enhanced version of the RLE[11].

I.4.5.3.2 Area coding

Is highly effective and it can give better compression ratio but it has certain limitation that it can be applied to nonlinear transformation

I.4.5.4 JPEG STANDARD

JPEG (Joint Photographic Experts Group) standard is designed by ISO (International Standards Organization) and the IEC group (International Electronic Commission). It is intended for compressing still images in color and grayscale for storage on digital media. It was conducted with a view to cover the most diverse applications, taking into account realistic constraints compared to the more visible applications, transmission, image banks. Techniques that define the JPEG standard are divided into two classes:

JPEG is a standard for continuous tone still images that allows lossy and lossless coding. Lossless JPEG [11] is an independent predictive coding compression technique that includes differential coding, run length, and Huffman code. Lossless JPEG is not largely used in space mission due to the low compression ratio. There are several modes defined for JPEG, including baseline, progressive, and hierarchical.

I.4.6 Lossy Compression

These methods are characterized by the loss of the original image [9]. Lossy compression only applies to "perceptual" data, usually sound or visual, which may undergo a modification, sometimes significant, without this is perceptible to a human. The loss of information is irreversible, it is impossible to recover the original data after such compression. Lossy compression is for this sometimes called irreversible or non-conservative compression.

I.4.7 Lossy Methods

I.4.7.1 Transform based Methods

Transform-based compression systems are based on the insight that the decorrelated coefficients of a transform can be coded more efficiently than the original image pixels. The transform typically results in some energy compaction (i.e., an energy redistribution of the original image into a smaller set of coefficients). Even though the energy is compacted into fewer coefficients, the total energy is conserved, resulting in a significant number of coefficients with values of zero or near zero. Several kinds of transforms, with different efficiencies, energy compacting methods, and computational complexities, are useful in compression systems. The most common data compression transforms are DWT (discrete wavelet transform).

I.4.7.1.1 DWT-Based Compression System.

The wavelet transform decomposes the original image into a sum of spatially and frequency localized functions, in a way that is similar to sub-band decomposition. The most important visual information tends to be concentrated into a reduced number of components (coefficients); therefore, the remaining coefficients can be quantized coarsely or truncated to zero with little image distortion. This is one of the reasons why wavelet-based compression schemes tend to produce superior image quality.

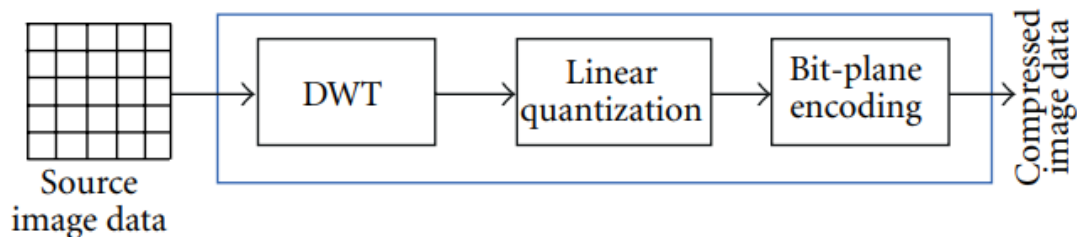


Figure I.13 A general diagram of the CCSDS-IDC encoder

I.4.7.2 Sub-band Coding

The sub band coding split the frequency bands of a signal and then each sub band is coded by encoder and bit rate which is related to that particular band.[12]. SBC is generally used in speech coding and image coding. At the decoder the sub band signals are decoded and un sampled and passed through a synthesis filter. Then all the sub band coefficients are properly summed up to yield the compressed image.

CHAPTER I : Generality On Digital Image Compression

I.4.7.3 The predictive method

The predictive method is one of the oldest, it is a de-correlation method where the principle is as follows[9]. In prescient coding of each pixel esteem is anticipated from previously encoded pixels. As it were the contrast between the anticipated esteem and the real esteem is quantified and after that encoded and transmitted.

I.4.7.4 Fractal compression

Is a process that only applies to images. It is based on the fractal geometry of Benoit Mandelbrot. The format of the images compressed by this process is currently only not standardized and therefore not recognized by browsers[6]. He is recognized by the. ff extension. • Principle of operation: The general principle is very simple to understand: the idea is that any image is a finite set of geometric transformations (rotations, translations, enlargements, reductions) applied to subsets of identical patterns and variable sizes that making up.

I.4.7.5 Standard H-261 compression

It is a standard developed by CCITT (International Consultative Committee on Telegraphy and Telephony). This standard is intended for the coding of moving for video calling (Visual Telephony) images. The H-261 is a hybrid combining the encoding and predictive coding. Is used to reduce spatial redundancy (intra-frame coding). The predictive coding for reducing the temporal redundancy between frames of the sequence (interframe coding).

I.4.7.6 Hybrid Methods

The term hybrid refers to the techniques combining predictive coding and transform coding. In the case of still images [9], a transformation is performed in a dimension along the line and then a prediction along the columns. For moving images, a combination is carried out between a two-dimensional transformation in the spatial domain and a prediction along the temporal component to exploit the temporal redundancy of the image signal.

I.4.8 Embedded coding

Wavelet transform generates sub-bands corresponding to the orthogonal projections of the vector space. There are structural similarities in the sub-bands' details in the same

CHAPTER I : Generality On Digital Image Compression

direction at successive resolutions (see Figure). Algorithms that exploit this property and are called embedded zero tree coding such as EZW, SPIHT and SPECK.

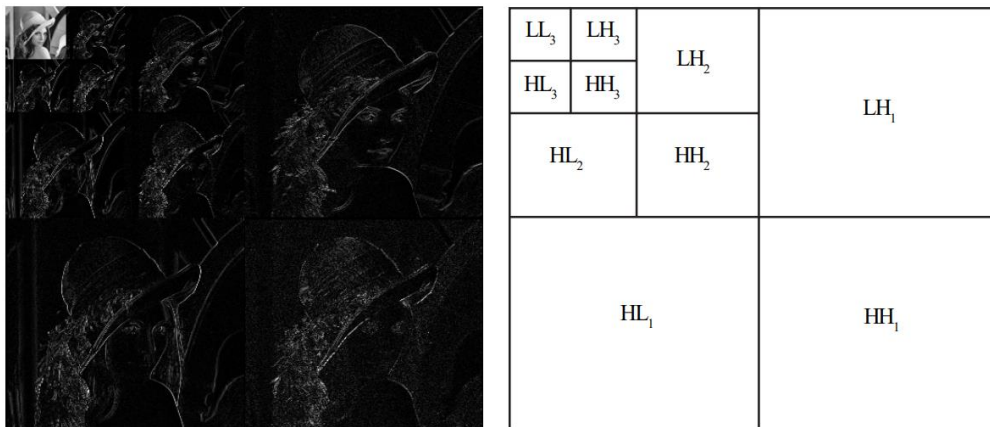


Figure I.14 The three-level wavelet decomposition of Lena image

I.4.8.1 JPEG 2000 Standard

After the JPEG standard, JPEG2000 now occupy a prominent place in the world of digital imaging: internet, medicine, digital cameras, image transmission on mobile networks ... etc. The JPEG2000 standard allowed to solve the disturbing problem of artifacts by providing a good visual quality at high levels of compression. Figure 1.7 shows an example of a reconstructed image by the standards JPEG and JPEG2000 for the same compression rate; the difference in the visual quality is clear.



Figure I.15 Accelerating JPEG 2000 Decoding for Digital Pathology and Satellite Images

I.4.8.2 Principle of JPEG2000

The standard JPEG2000 is based on the discrete wavelet transform DWT. The DWT allows a more localized analysis therefore finer, of the information. DWT methods are distinguished by the selection procedure of the coefficients of wavelet before the step of quantification. Two main approaches are used to eliminate the non-significant coefficients:

- The selection by sub-band which is to zero all coefficients nonsignificant lower than a threshold in each of the sub-bands separately. The inter-band correlation is therefore not taken into account.

I.5 Conclusion

Digital images, whether still or moving, are intended to be shared, therefore transmitted over communication networks. We must therefore see more far than simple source coding.

Compression is generally only one task particular that fits into systems where more constraints are present, than must be considered during development. Thus, we do not necessarily have to put puts in place an algorithm which – as an independent method – produces a rate of maximum compression, but which in an overall system leads to the performance optimal. The performance criteria of course include the compression ratio. But if transmission is part of the services.

Chapter II: Image Compression Using 2-D Wavelet

II.

II.1 Introduction

In recent years, researchers have come up with a transformation wavelet to treat the digital signal has been the need for the emergence of the wavelet transform is a solution to the problem of the finiteness of the transformations Fourier.

Wavelet transformation, in modern transformations, is considered to have many advantages and disadvantages characteristics, making it relies on statistical guesses that lead to numerical processing, and these have been shown The last effective results in a lot of applications, such as: Compressing the image carries the sign or the image to Multiple levels of detail, and this is both the time domain and frequency, and from which we address in this chapter to Fourier transform (FT) and focus on its shortcomings, then identify the short-run Fourier transform (STFT) and Recognize the types of waveforms so that the transformations of continuous waveforms (CWT, DWT) are described, and then we present An algorithm for constructing an image hierarchy that allows obtaining the coefficients through a split wavelet transform.

II.2 Fourier transform

A Fourier transform (FT) is a mathematical transform that decomposes functions depending on space or time into functions depending on spatial frequency or temporal frequency. An example application would be decomposing the waveform of a musical chord into terms of the intensity of its constituent pitches. The term Fourier transform refers to both the frequency domain representation and the mathematical operation that associates the frequency domain representation to a function of space or time.

An absolutely integrable function f on \mathbb{R} is a real- or complex-valued function such that $\int_{-\infty}^{\infty} |f(t)| dt < \infty$. If f is an absolutely integrable function on \mathbb{R} [12], then the Fourier transform of for cyclic frequency ω , denoted $f(\omega)$ is defined as

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t} dt \quad (II.1)$$

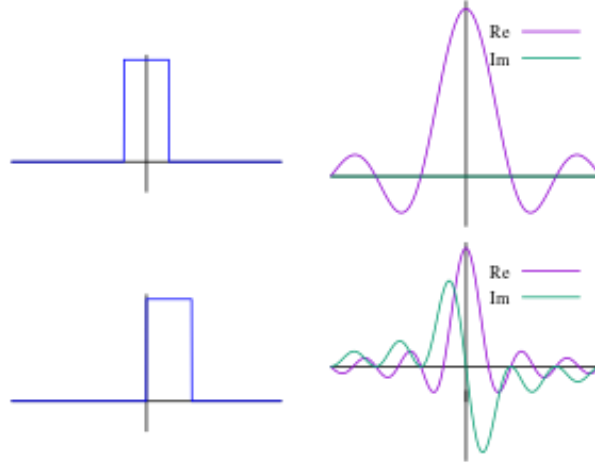


Figure II.1 Fourier Transform

II.2.1 principle

The principle of Fourier transform is to extract the fundamental frequency component of the fringe pattern in the 1- or 2-D frequency domain. An inverse transform of the filtered frequency domain signal then provides the modulo 2π phase of the fringe pattern[13]. For example, for a 2-D fringe pattern as described by Eq. (1), after the Fourier transform, filtering in the frequency domain, and inverse Fourier transform, the filtered exponential fringe pattern $\bar{f}(x, y)$ can be obtained as

$$\bar{f}(x, y) = \mathbf{F}^{-1}\{\mathbf{F}\{f(x, y)\}\} = a(x, y)\exp(j \times \varphi(x, y)) \quad (II.2)$$

where functions of \mathbf{F} and \mathbf{F}^{-1} represent forward and inverse Fourier transform, respectively.

II.2.2 Short-time Fourier transform (STFT)

Is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. In practice, the procedure for computing STFTs is to divide a longer time signal into shorter segments of equal length and then compute the Fourier transform separately on each shorter segment. This reveals the Fourier spectrum on each shorter segment.

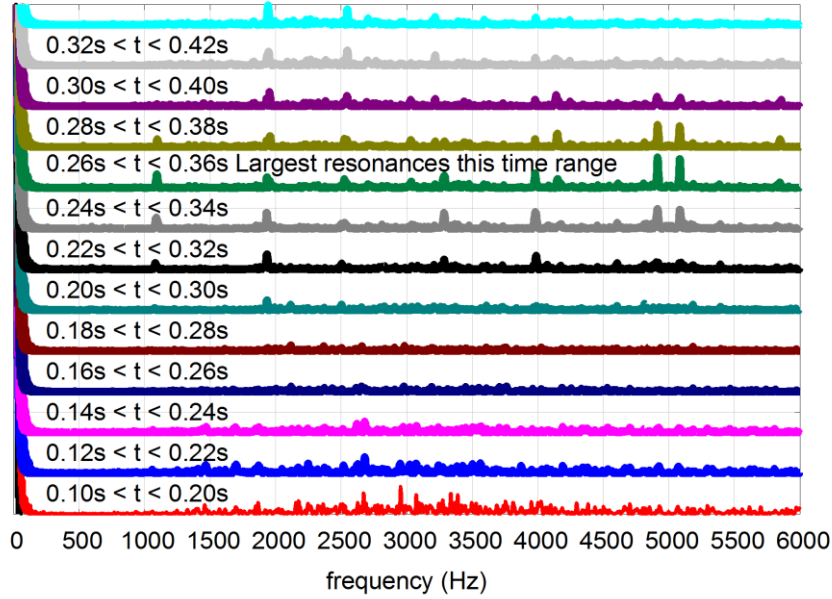


Figure II.2 Short-time Fourier transform STFT

Simply, in the continuous-time case, the function to be transformed is multiplied by a window function which is nonzero for only a short period of time. The Fourier transform (a one-dimensional function) of the resulting signal is taken as the window is slid along the time axis, resulting in a two-dimensional representation of the signal. Mathematically, this is written as:

$$STFT\{x(t)\}(\tau, \omega) \equiv x(\tau, \omega) = \int_{-\infty}^{+\infty} x(t) \omega(t - \tau) e^{-j\omega t} dt \quad (II.3)$$

II.3 Wavelet Transform

In order to overcome the problems of the short-time Fourier transform (STFT), which splits the time. Therefore, the theory of wavelet transformation or wavelet loading is developed, which depends on the frequency and frequency with uniform accuracy. Using a variable width window instead of a fixed width window along the signal, in order to obtain a localized loading of the signal (frequency - flux) and the treatment of high frequencies differently low frequencies.

$$T^{and} f(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) \psi\left(\frac{t - b}{a}\right) dt \quad (II.4)$$

Is the scale factor and b the translation parameter? The variable a plays the role of the inverse of the frequency: the smaller a is, the less the wavelet (the function analyzing) is extended in time, so the higher the central frequency of its spectrum[14]. We can also interpret this

CHAPTER II: Image Compression Using 2-D Wavelet

expression as a projection of the signal on a family of analyzing functions $\psi_{a,b}$, constructed from a “mother” function ψ in accordance with the following equation:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (\text{II.5})$$

Note that the norm is preserved when changing the scale factor:

$$\|\psi_{a,b}\|^2 = \int_{-\infty}^{+\infty} \frac{1}{a} \left| \psi\left(\frac{t-b}{a}\right) \right|^2 dx = \frac{1}{a} \int_{-\infty}^{+\infty} |\psi(x)|^2 a dx = \|\psi\|^2 \quad (\text{II.6})$$

We notice

$$T^{ond} f(a, b) = \langle f, \psi_{a,b} \rangle \quad (\text{II.7})$$

The spatial-temporal resolution is calculated in the same way as before: If the temporal “width” of ψ (the standard deviation) is taken as a unit:

$$\begin{aligned} \sigma_t^2 &= \int t^2 |\psi_{a,0}(t)|^2 dt \\ &= \int t^2 \frac{1}{a} \left| \psi\left(\frac{t}{a}\right) \right|^2 dt \\ &= \int_{-\infty}^{+\infty} a^2 x^2 \frac{1}{a} |\psi(x)|^2 a dx \end{aligned} \quad (\text{II.8})$$

It gives

$$\sigma t = a \quad (\text{II.9})$$

Unlike the STFT which has a fixed window size throughout, with wavelets we can analyze a signal at different frequencies with different resolutions. As seen in the figure below, wavelets can provide good frequency resolution and relatively poor temporal resolution at low frequencies, while good temporal resolution and relatively poor frequency resolution at high frequencies [15].

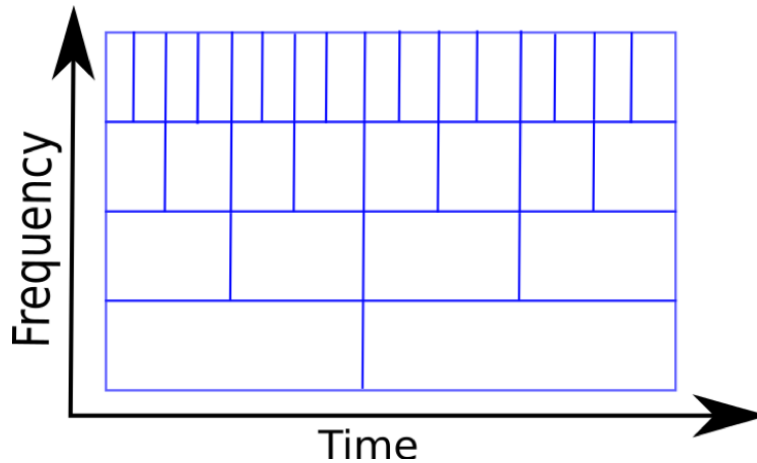


Figure II.3 Time frequency tiling Wavelet Transform

And among the examples we mention Haar wavelet and Morlet:

II.3.1 Haar Wavelet

In mathematics, the Haar wavelet is a sequence of rescaled "square-shaped" functions which together form a wavelet family or basis. Wavelet analysis is similar to Fourier analysis in that it allows a target function over an interval to be represented in terms of an orthonormal basis. The Haar sequence is now recognized as the first known wavelet basis and extensively used as a teaching example[16].

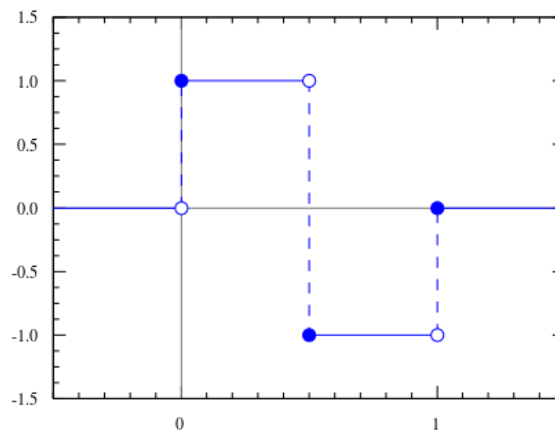


Figure II.4 Haar Wavelet

The study of wavelets, and even the term "wavelet", did not come until much later. As a special case of the Daubechies wavelet, the Haar wavelet is also known as Db1.

The Haar wavelet's mother wavelet function $\psi(t)$ can be described as

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2}, \\ -1, & \frac{1}{2} \leq t < 1, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{II.10})$$

II.3.2 Daubechies wavelet

The Daubechies wavelets, based on the work of Ingrid Daubechies, are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. With each wavelet type of this class, there is a scaling function (called the father wavelet) which generates an orthogonal multiresolution analysis[17], [18].

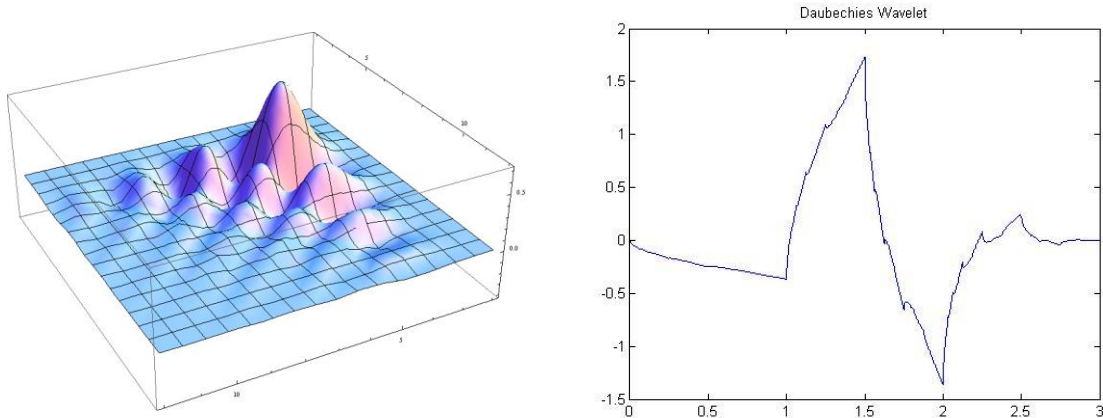


Figure II.5 Daubechies Wavelet

II.3.3 Biorthogonal wavelet

A Biorthogonal wavelet is a wavelet where the associated wavelet transform is invertible but not necessarily orthogonal[19]. Designing biorthogonal wavelets allows more degrees of freedom than orthogonal wavelets. One additional degree of freedom is the possibility to construct symmetric wavelet functions[20].

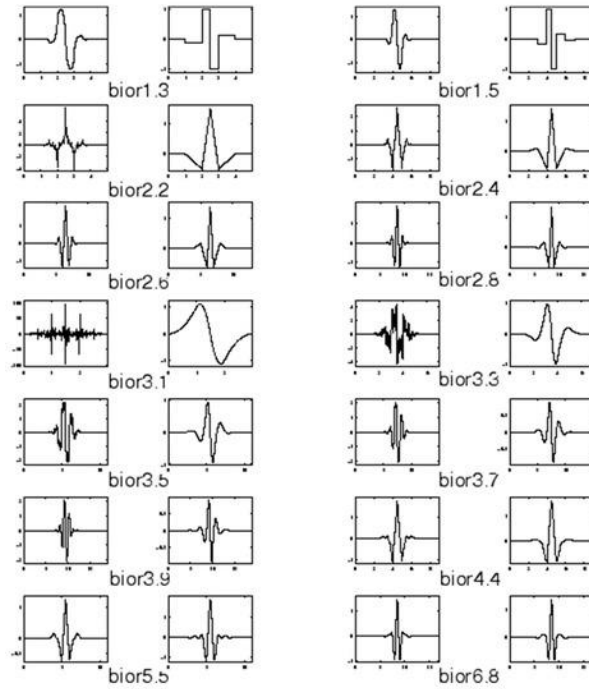


Figure II.6 Biorthogonal Wavelet

II.3.4 Morlet Wavelet

In 1946, physicist Dennis Gabor, applying ideas from quantum physics, introduced the use of Gaussian-windowed sinusoids for time-frequency decomposition, which he referred to as atoms, and which provide the best trade-off between spatial and frequency resolution. These are used in the Gabor transform, a type of short-time Fourier transform. In 1984, Jean Morlet introduced Gabor's work to the seismology community and, with Goupillaud and Grossmann, modified it to keep the same wavelet shape over equal octave intervals, resulting in the first formalization of the continuous wavelet transform.[21], [22].

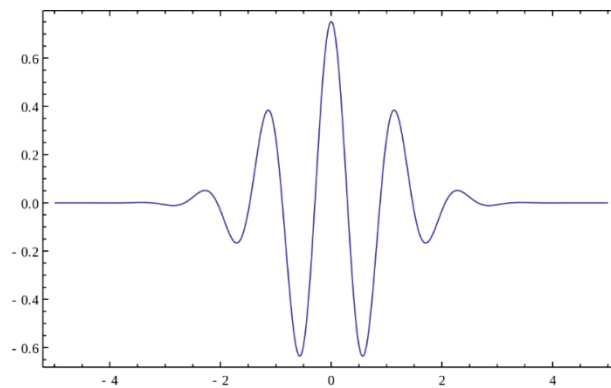


Figure II.7 Morlet Wavelet

Wavelets are used in many ways to process the signal through mathematical equations, and it is known as the transformation. The wavelet transformation splits the wavelet transform into continuous and discontinuous

II.4 Continuous Wavelet Transform (CWT)

In mathematics, the continuous wavelet transform (CWT) is a formal (i.e., non-numerical) tool that provides an overcomplete representation of a signal by letting the translation and scale parameter of the wavelets vary continuously[23].

The continuous wavelet transforms of a function $x(t)$ at a scale($a > 0$) $a \in \mathbb{R}^{*+}$ and translational value $b \in \mathbb{R}$ is expressed by the following integral

$$a X_{\omega}(a, b) = \frac{1}{|a|^{\frac{1}{2}}} \int_{-\infty}^{+\infty} x(t) \bar{\psi}\left(\frac{t-b}{a}\right) dt \quad (\text{II.11})$$

Where $\bar{\psi}$ is a continuous function in both the time domain and the frequency domain called the mother wavelet and the overline represents operation of complex conjugate. The main purpose of the mother wavelet is to provide a source function to generate the daughter wavelets which are simply the translated and scaled versions of the mother wavelet.

The problem with transforming continuous waveforms is the amount of data that is available to use all the values and gradations, which is What makes the processing process take a long time, so the optimal solution is sufficient to resort to wavelet transformation. Sheared DWT.

II.5 Discrete Wavelet Transform

In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms been temporal resolution: it captures both frequency and location information (location in time)[24].

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \phi(Sx - k) \quad (\text{II.12})$$

CHAPTER II: Image Compression Using 2-D Wavelet

where S is a scaling factor (usually chosen as 2). Moreover, the area between the function must be normalized and scaling function must be orthogonal to its integer translates e. g.

$$\int_{-\infty}^{\infty} \phi(x)\phi(x+l)dx = \delta_{0,l} \quad (\text{II.13})$$

After introducing some more conditions (as the restrictions above does not produce unique solution) we can obtain results of all these equations, e.g.

$$\psi(x) = \sum_{k=-\infty}^{\infty} (-1)^k a_{N-1-k} \psi(2x - k) \quad (\text{II.14})$$

where N is an even integer. The set of wavelets than forms an orthonormal basis which we use to decompose signal. Note that usually only few of the coefficient's a_k is non-zero which simplifies the calculations.

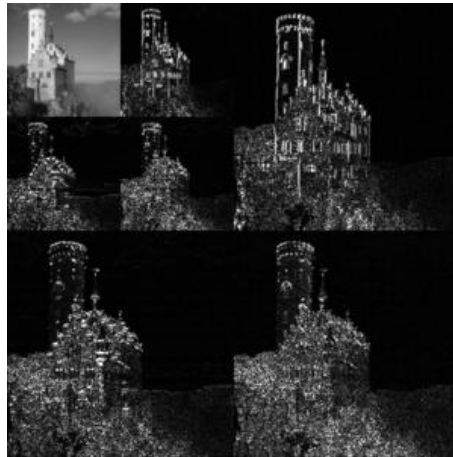


Figure II.8 Discrete Wavelet Transformation

II.6 Multiresolution analysis

A multiresolution analysis (MRA) or multiscale approximation (MSA) is the design method of most of the practically relevant discrete wavelet transforms (DWT) and the justification for the algorithm of the fast wavelet transform (FWT)[25].

The method of analyzing the signal into multiple levels of accuracy depends on the scale function or the wave father ϕ and the mother wave Ψ and that check[26]:

$$\psi_{n,m}(t) = 2^{-\frac{m}{2}} \psi(2^{-m}t - n) \quad (\text{II.15})$$

$$\varphi_{n,m}(t) = 2^{-\frac{m}{2}}\varphi(2^{-m}t - n) \quad (\text{II.16})$$

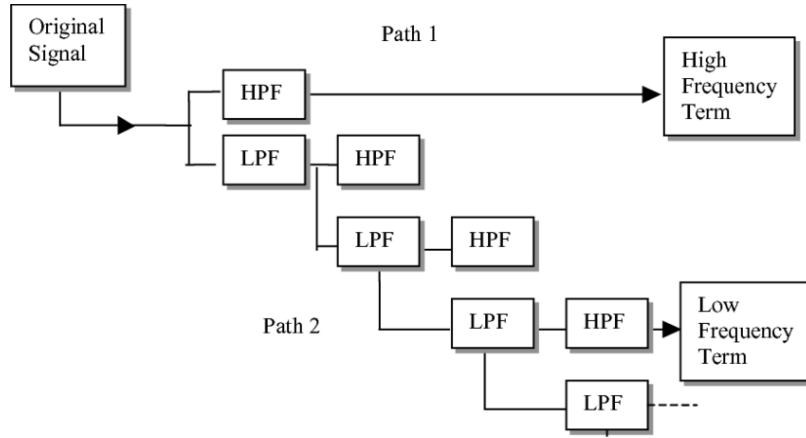


Figure II.9 Multiresolution Analysis using wavelet transform

II.6.1 Approximation spaces

We place ourselves in space $L^2(R)$ continuous functions of a real variable and of integrable square. An analysis at resolution j of the function f will be obtained by the action of a linear operator A_j on f , such that: $j \ V_j \ A \ f \in \ v_j$ being a subspace of L^2 , A will be a projector (idempotent). We will build a multi-resolution analysis using V_j subspaces nested together in the others, such that the passage from one to the other is the result of a change of scale (zoom). For example, in the dyadic case we have:

$$f(x) \in V_j \Leftrightarrow f\left(\frac{x}{2}\right) \in V_{j+1} \quad (\text{II.17})$$

This corresponds to a dilation of a factor of 2. The space V_{j+1} contains more signals” coarser” than the V_j space and it is clear that:

$$\bigcup_{j \in \mathbb{Z}} V_j = L^2(R) \quad (\text{II.18})$$

$$\bigcap_{j \in \mathbb{Z}} V_j = \{0\} \quad (\text{II.19})$$

II.7 The pyramid of the image

Loading the image with several levels of resolution (Multiresolution) using waveform transformation Discrete DWT (allows the signal to be split into a pyramid shape) The sub-bands of the signals (low frequencies and other high frequencies) are illustrated by Equations [A22:] The approximate coefficients of the signal () that have the lower frequencies are calculated by calculating The deflection is between two consecutive minutes Passing it through the filter, we illustrate this with the following equation:

$$A_j x = \sum_{-\infty}^{+\infty} \bar{h}(2k - n) A_{j-1} x \quad (\text{II.20})$$

Which,

$$h(n) = \langle \varphi_0^{-1}, \varphi_n^0 \rangle \quad (\text{II.21})$$

As for the details of the reference, it contains the missing information between successive approximations, which are as follows:

$$D_j x = \sum_{-\infty}^{+\infty} \bar{g}(2k - n) A_{j-1} x \quad (\text{II.22})$$

Which

$$g(n) = (-1)^{1-n} h(1 - n) \quad (\text{II.23})$$

Thus, the signal frequencies are separated by convolution by two filters to pass low in order to calculate the approximation and the other for high frequencies to find details (detail), so that it contains the output together on the same frequency content as the input signal, and thus the amount of data is doubled, which we resort to Reduce by a factor of 2, which is to reduce the number of samples (down sampling) by half[27].

CHAPTER II: Image Compression Using 2-D Wavelet

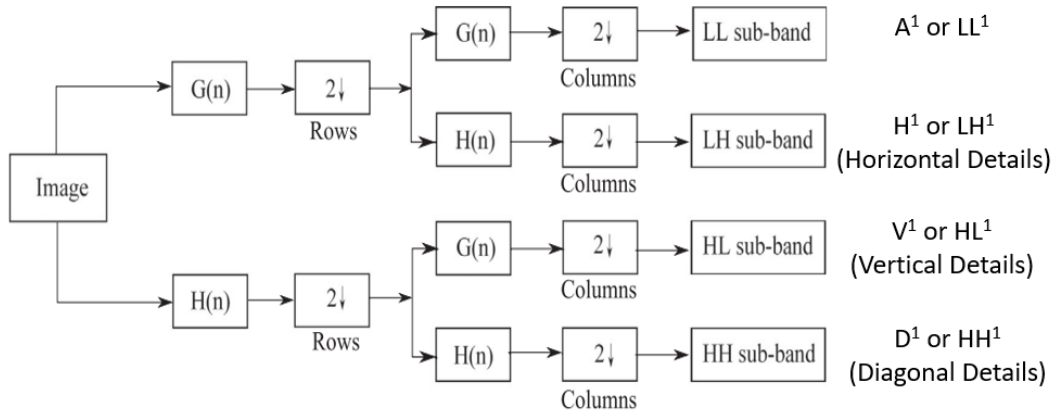


Figure II.10 Pyramid algorithm (Fast Mallat algorithm)

The image enters the upload process horizontally, and the low-pass filtering process is implemented, so that each filter is subjected to a sampling reduction process, then high pass is repeated Approximate image and the third process with the same principle in a vertical form so that we get four images we call other images called details contain horizontal details (DH), vertical details (DV) and details (even to get the smallest diagonal image (DD))[26], [27].

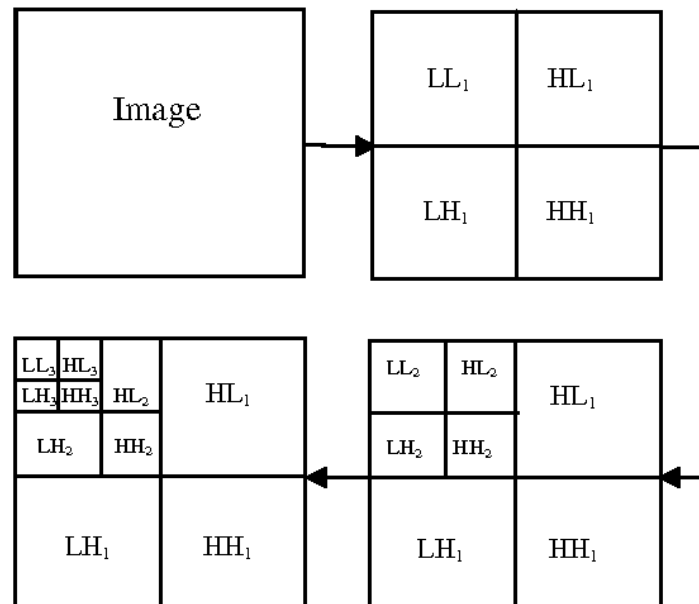


Figure II.11 Analysis of three level of Image decomposition

The image is restored by a reverse process of the loading stage so that the number of samples is increased Up-sampling (and denoted by $\uparrow 2$) and then the signal is passed using filters

CHAPTER II: Image Compression Using 2-D Wavelet

QMF (Filters Mirror Quadrature) which are built on the basis of the input filters[28], which are filters with Also, low and high pass, all filter outputs are summed to rebuild the signal. Explanation through the following figure

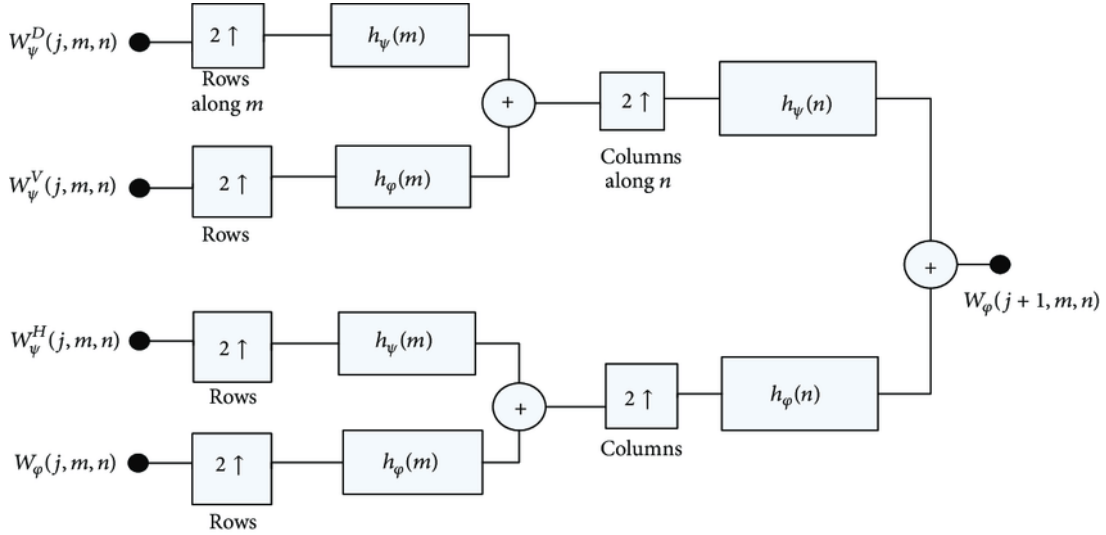


Figure II.12 Pyramid algorithm (Fast Mallat algorithm) the inverse

II.8 Embedded Coding methods

To better explain the concept of embedded coding, consider the following example. Assume that three users expect a compressed image to be sent. At the same time, they require different quality of the image. The first requires a 10 KB image quality, and the second and third user need a quality of 20 KB and 50 KB, respectively. Most methods of image compression with partial loss of data will have to compress the original image three times with different qualities, in order to generate three different files for the required qualities. A method based on embedded coding produces a single file, and users will receive three fragments of the file where lengths are equal to 10 KB, 20 KB and 50 KB.

II.8.1 Embedded Zerotrees Wavelet (EZW)

Embedded Zerotrees of Wavelet transforms (EZW) is a lossy image compression algorithm. At low bit rates, i.e., high compression ratios, most of the coefficients produced by a sub-band transform (such as the wavelet transform) will be zero, or very close to zero. This occurs because "real world" images tend to contain mostly low frequency information (highly correlated). However, where high frequency information does occur (such as edges in the image) this is particularly important in terms of human perception of the image quality, and thus must be represented accurately in any high-quality coding scheme[29], [30].

CHAPTER II: Image Compression Using 2-D Wavelet

The EZW algorithm can be summarized in three steps. As following:

- The definition of significant maps indicating the positions of the coefficients significant with respect to a given threshold.
- A successive approximation by displacements of the coefficients significant, which therefore allows a notion of progressiveness of the coding according to a rate-distortion stop criterion.
- A dynamic Symbol String arithmetic encoder[31].

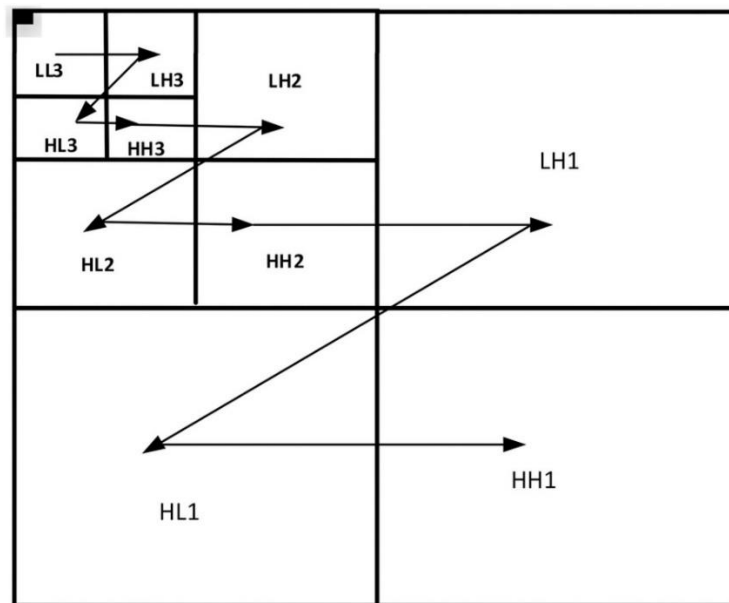


Figure II.13 Embedded Zerotrees of Wavelet Sub-band

II.8.1.1 Encoding a coefficient of the significance map

In a significance map, the coefficients can be representing by the following four different symbols. With using these symbols to represent the image information, the coding will be less complication.

II.8.1.1.1 Zerotree root

If the magnitude of a coefficient is less than a threshold T , and all its descendants are less than T , then this coefficient is called zerotree root. And if a coefficient has been labeled as zerotree root, it means that all of its descendants are insignificant, so there is no need to label its descendants.

CHAPTER II: Image Compression Using 2-D Wavelet

II.8.1.1.2 Isolated zero

If the magnitude of a coefficient that is less than a threshold T , but it still has some significant descendants, then this coefficient is called isolated zero.

II.8.1.1.3 Positive significant coefficient

If the magnitude of a coefficient is greater than a threshold T at level T , and also is positive, then it is a positive significant coefficient.

II.8.1.2 Defining Threshold

The threshold using above can be defined as the type below.

II.8.1.2.1 Initial threshold T_0

Assume C_{max} is the largest coefficient we get the below formula

$$T_0 = 2^{\lceil \log_2 C_{max} \rceil} \quad (\text{II.24})$$

II.8.1.2.2 Threshold T_i

It is reduced to half of the value of the previous threshold means T_0 and its formula is

$$T_i = \frac{1}{2} * T_{i-1} \quad (\text{II.25})$$

II.8.1.3 Scanning order for coefficients

Raster scanning is the rectangular pattern of image capture and reconstruction. Using this scanning on EZW transform is to perform scanning the coefficients in such way that no child node is scanned before its parent node. Also, all positions in a given sub-band are scanned before it moves to the next sub-band.

II.8.1.3.1 Negative significant coefficient

If the magnitude of a coefficient is greater than a threshold T at level T , and also is negative, then it is a negative significant coefficient.

II.8.2 Set Partitioning in Hierarchical Trees (SPIHT)

The SPIHT algorithm proposed by Saïd and Pearlman is an improvement of the EZW algorithm coding scheme. It is based on the same concepts: coding progressive by bit-plane and use of hierarchical dependencies between coefficients of the different sub-bands. However, a new protocol dependence between the coefficients is defined. The coefficients of the sub-band plus low frequency are grouped by four[32], and for each group, the progeny is as follows:

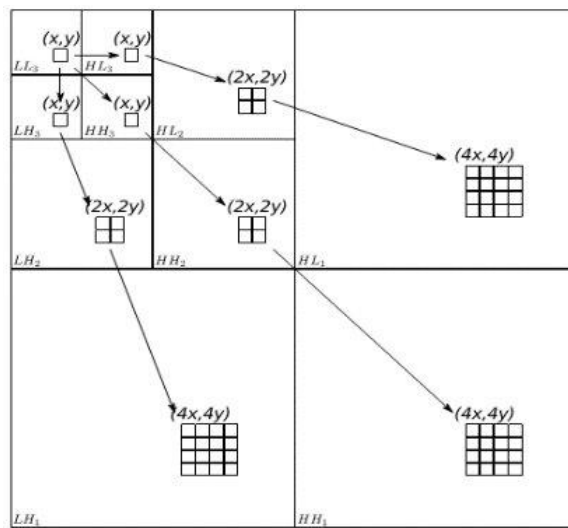


Figure II.14 The repartition used by SPIHT algorithm

II.9 Conclusion

In this chapter, we give an overview of wavelet transformation in general and discrete and continuous wavelet transformation in particular, the basic principles for this transformation were drawn up by Mallat Stéphane in 1989 Which was inserted next to the sign and the digital image, while it was enough just mathematical relationships, which Meyer put (1986-1988). So that the purpose behind the wave transformation theory was to overcome the shortcomings to convert.

Chapter III: Graphic User Interface (GUI)

MATLAB

III.

III.1 Introduction

It is an interactive and user-friendly system of numerical calculation and graphic visualization, intended to engineers and scientists. It has a programming language that is both powerful and simple of use. It allows problems and solutions to be expressed in an easy way, unlike other programming languages. It integrates functions of numerical analysis, matrix calculation, signal processing, 2D and 3D graphic visualization, etc. It can be used so interactively or in programming mode.

In this chapter we are going to discuss about GUI, with MATLAB you can create your own Graphical User Interface, or GUI, which consists of a Figure window, that a user can manipulate interactively with the mouse and keyboard.

III.2 The GUI presentation

The GUI (Graphical User Interface) makes it possible to create interfaces where the creator chooses several types of objects. The question that arises is: Why use a graphical interface in MATLAB? The main reason for using graphical interfaces is that they make easier things for the end users of the program. A good GUI can facilitate programs by providing them with a consistent look and intuitive controls, when a click of mouse over a push button occurs the GUI should launch the command carried by the label of this last[33].

The GUI provides the user with a familiar environment to work with. This environment contains *push buttons*, *toggle buttons*, *lists*, *menus*, *text boxes*, etc. suite, which are already well known by the user, so that he can concentrate on his application. Then, he must carry out the programming to obtain the interaction he wishes between these different objects. The three main elements to create a graphical interface in MATLAB are:

CHAPTER III : Graphic User Interface (GUI) MATLAB

III.2.1 The Components

Each item on a MATLAB GUI is a graphical component that includes controls and graphics. Graphical controls and static elements are created by the `ui-control` function and menus are created by the `ui-menu` and `ui-context menu` functions.

III.2.2 The Figures

The components of a GUI must be arranged in a figure which is a window on the screen of the computer. In the past, figures were created automatically whenever data was launched. However empty figures can be created with the function `figure` and can be used to hold any combination of components[34]

III.2.3 Callbacks (callback service)

There must be some way to perform an action if the user clicks the mouse or inserts information via the keyboard. For example, if a user clicks on a button, this event must cause the MATLAB code that implements the button's function to execute. There must be a callback to implement the function of each component graphic on the GUI

III.2.4 Launch GUI MATLAB

To launch GUI, you just head over to MATLAB and run 'guide' in the command line like this:

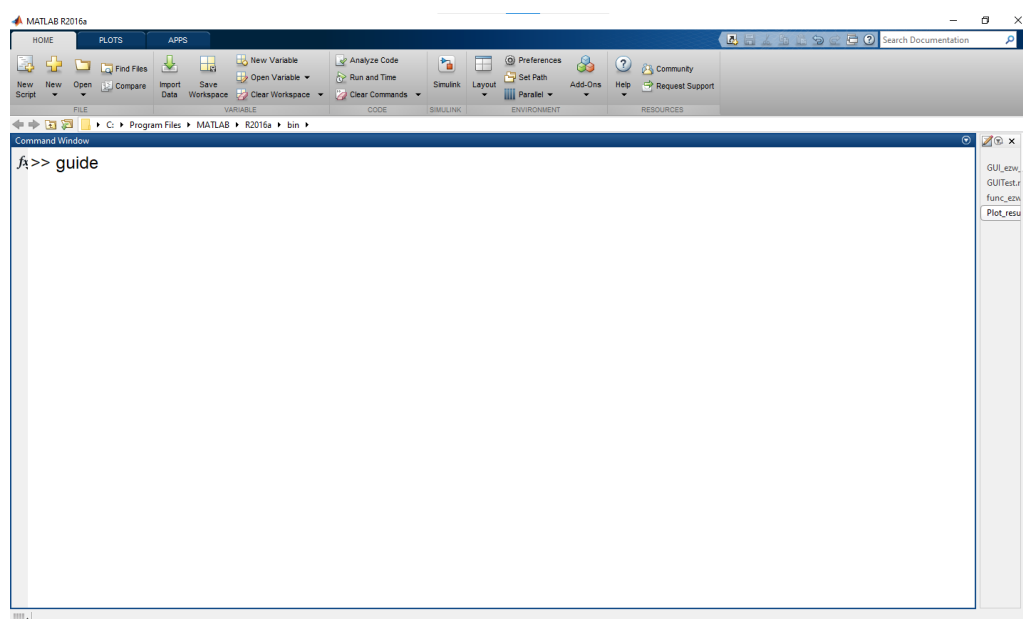


Figure III.1 MATLAB Command line

CHAPTER III : Graphic User Interface (GUI) MATLAB

It will pop up a window guide quick start where you can either create a new GUI or open an existing one as the figure below

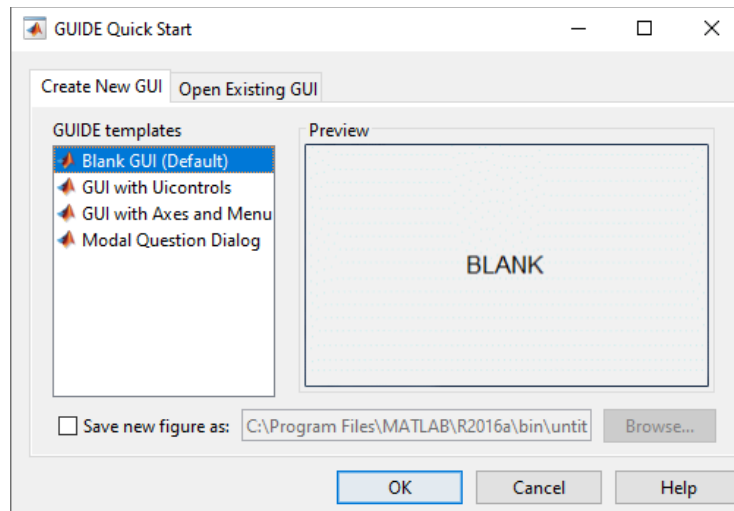


Figure III.2 Guide Quick Start

If you want to start a new interface you just have to go to *create New GUI* then you'll find options if you want to start from scratch, you choose *Blank GUI (default)*

After this it will open up the GUI working space the figure for you like below:

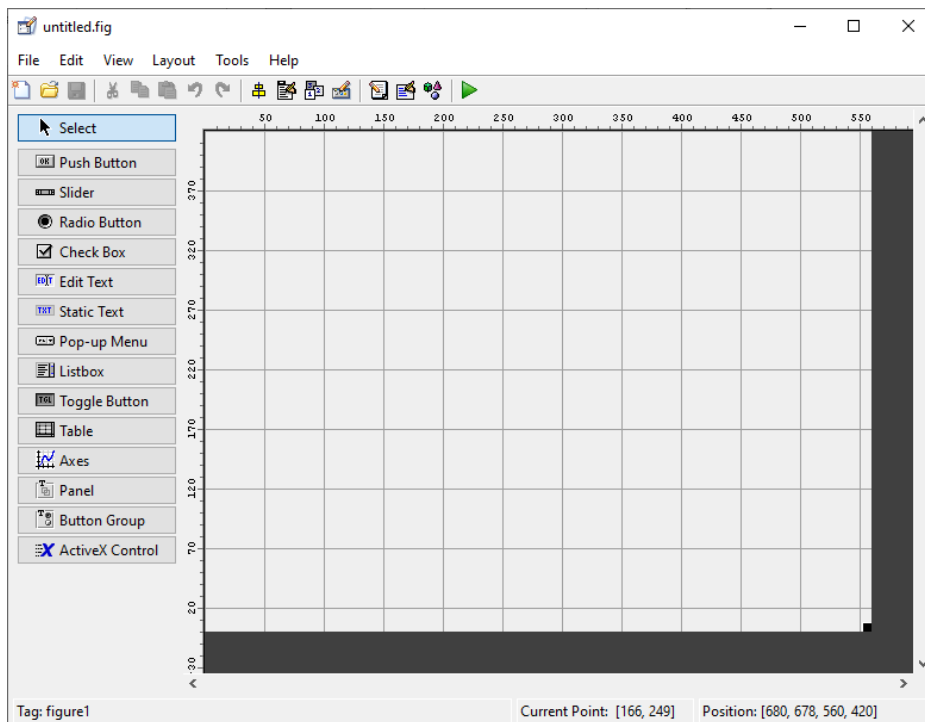


Figure III.3 The layout Window

III.3 The Layout Window

To have the components with their names on the tool's palette, just select the preferences in the MATLAB menu, and thus have the following layout below:

Name of Component and interface

1. New Figure
2. Open Figure
3. Save Figure
4. Cut
5. Copy
6. Paste
7. Undo
8. Redo
9. Align Objects
10. Menu Editor
11. Tab order editor
12. Toolbar editor
13. Editor
14. Property inspector
15. Object browser
16. Run Figure
17. Component Palette
18. Layout Area

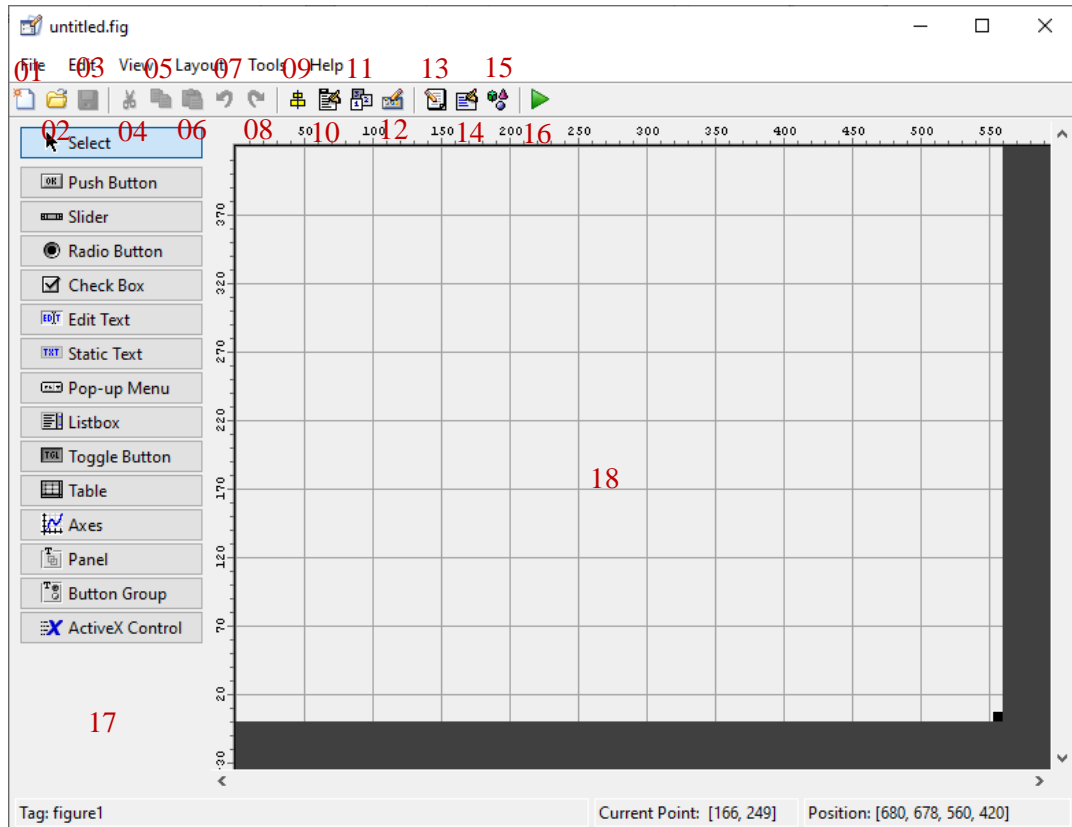


Figure III.4 Layout window showcasing the interface component

The main components of a graphical interface are:

1. Select
2. Push Button
3. Slider
4. Radio Button
5. Check Box
6. Edit Text
7. Static text
8. Pop-up Menu
9. Listbox
10. Toggle Button
11. Table
12. Axes
13. Panel
14. Button Group
15. Active X Control

III.4 Presentation of the different interface objects

In our interface we used the following objects:

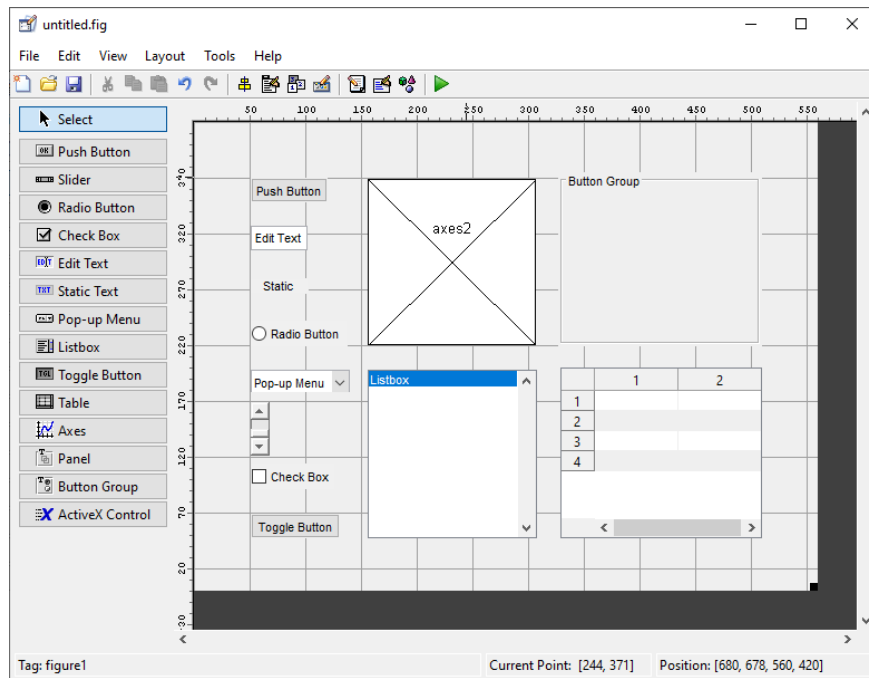


Figure III.5 Different Object interface

The generated interface in the figure below:

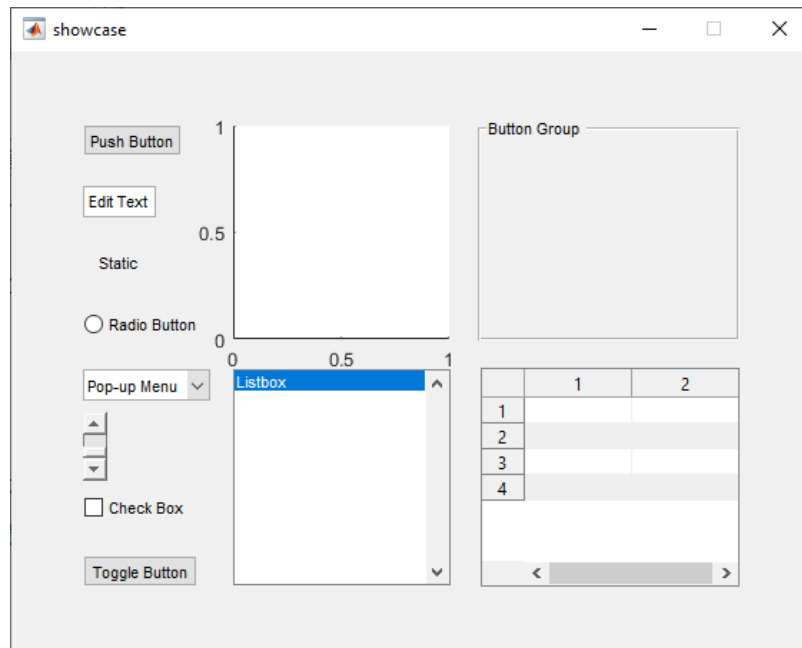


Figure III.6 The generated interface

III.5 Used Functions

When creating the GUI `untitled.fig` (extension file), a `.m` extension file is also generated and has the same name. For each object, in the `.m` file is declared a Callback function preceded by the name of the object (modifiable thanks to the Tag field in its properties). The code introduced in this function corresponds to the event that will occur when the mouse or cursor action on the object.

Then, to manipulate these objects, there are 2 important functions:

- `get` which allows to retrieve the values
- `set` which allows to put values

When you save the GUI, it will generate an editor file with 3 main callback functions (classes)

- Initialization Function can't be edit
- The opening function
- The output functions

```
1
2 function varargout = untitled1(varargin)
45 % End initialization code - DO NOT EDIT
46
47
48 % --- Executes just before untitled1 is made visible.
49 function untitled1_OpeningFcn(hObject, eventdata, handles, varargin)
61
62 % UIWAIT makes untitled1 wait for user response (see UIRESUME)
63 % uiwait(handles.figure1);
64
65
66 % --- Outputs from this function are returned to the command line.
67 function varargout = untitled1_OutputFcn(hObject, eventdata, handles)
75
```

Figure III.7 The generated editor

CHAPTER III : Graphic User Interface (GUI) MATLAB

The opening function is responsible for handles

Every Component in the GUI has its own Handles which is tagged by a name, means if you want to use any handles, you'll need the tag name like this:

➤ `handles.tagname`

```
1
2  function varargout = GUI_ezw_sphit(varargin)
46  % End initialization code - DO NOT EDIT
47
48
49  % -- Executes just before GUI_ezw_sphit is made visible.
50  function GUI_ezw_sphit_OpeningFcn(hObject, eventdata, handles, varargin)
51  % This function has no output args, see OutputFcn.
52  % hObject    handle to figure
53  % eventdata  reserved - to be defined in a future version of MATLAB
54  % handles    structure with handles and user data (see GUIDATA)
55  % varargin   command line arguments to GUI_ezw_sphit (see VARARGIN)
56
57  % Clear the image plot
58  InitImageFig(handles)
59
60  % Clear the image plot
61  InitImageInput(handles)
62
63  % Choose default command line output for GUI_ezw_sphit
64  handles.output = hObject;
65  handles.image_info = 0;
66  handles.imname = 0;
67  handles.image_rex = 0;
68  handles.if_dicom = 0;
69
70  % Update handles structure
71  guidata(hObject, handles);
72
73  % UIWAIT makes GUI_ezw_sphit wait for user response (see UIRESUME)
74  % uiwait(handles.figure1);
75
76
77  % -- Outputs from this function are returned to the command line.
78  function varargout = GUI_ezw_sphit_OutputFcn(hObject, eventdata, handles)
79  % varargout  cell array for returning output args (see VARARGOUT);
```

Figure III.8 The generated editor

III.6 Conclusion

MATLAB provides a good environment for creating the GUI. This is because it automatically generates the code for the design of the GUI. The generated code also contains comments that make it easy for the programmer to understand what the code entails and to better organize the callback functions. This makes the programmer's work easier and saves him/her from the bulky codes. This makes MATLAB an efficient tool for creating GUI. The purpose of the GUI is to create an interface to be used by everyone. This makes it an important tool for engineers implementing it in their projects.

Chapter IV: Results and Discussion

IV.

IV.1 Introduction

In this chapter, the compression results that were obtained using the Embedded coding for compression are presented.

So that the MATLAB program provides the possibilities to exploit the studied coding of the compression process and decompress using toolbox '*wcompress*' and '*GUI*' for creating an interface therefore to manipulate the image compression process more easily and efficiently.

A set of examples is selected for testing, including the reference images used on an extensive system for compressing the image, in order to evaluate the performance, we are going to apply methods such as *SPIHT*, *EZW* and other '*wcompress*' methods.

We are going also to apply wavelet for transformation in order to compare each of: *db4*, *bior3.3*, *Haar*.

The performance metric used concerns:

- Peak Signal-to-Noise Ratio *PSNR*
- Compression Ratio *CR*

IV.2 Research steps and method of work

In this work, I do the image compression by following these steps:

- Creating an interface with *GUI* MATLAB
- Use the MATLAB toolbox '*wcompress*' to do the compression

In the interface I can do the following:

- Load the image
- Specify the compression Methods (ex: *EZW*, *SPIHT*)
- Choose the wavelet filter (ex: *Haar*, *bior3.3*)
- Then I set a loop number to which the encoding process

CHAPTER IV: Results and Discussion

- And lastly, I press on *compress* to the process

IV.2.1 Compression Ratio

The compression ratio is defined as the ratio between the total number of bits necessary to represent the original information and the total number of bits of the binary file to store which results from the compression method:

$$\text{Compression ratio} = \frac{\text{Uncompress File Size}}{\text{Compressed File Size}} = CR(\text{bpp}) \quad (\text{IV.1})$$

In practice, the flow rate is rather used to measure the compaction power of a method. The bit rate is expressed in bits per pixel:

$$\text{Compression ratio} = \frac{\text{Uncompress File Size}}{\text{Compressed File Size}} \times 100 = CR(\%) \quad (\text{IV.2})$$

IV.2.2 Mean square error (MSE)

Is the squared norm of the difference between the data and the signal or image approximation divided by the number of elements. The MSE is defined by:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |X(i, j) - X_c(i, j)|^2 \quad (\text{IV.3})$$

$n \times m$: The size of the image

X, X_c : The values of the intensities of the original image and of the reconstructed image respectively

The other evaluation criterion, directly deduced from the EST is the signal-to-noise ratio (SNR):

$$SNR = 10 \log_{10} \left(\frac{1}{MSE} \right) \quad (\text{IV.4})$$

Is the variance of the signal (the image). The SNR quantity is measured in decibels (dB). Another widely used criterion is the peak signal-to-noise ratio or PSNR (Peak Signal to Noise Ratio) defined by:

$$PSNR(db) = 10 \log_{10}\left(\frac{255^2}{MSE}\right) \quad (IV.5)$$

For the compression and the decompression phase, this more complex process can be represented by the following figure.

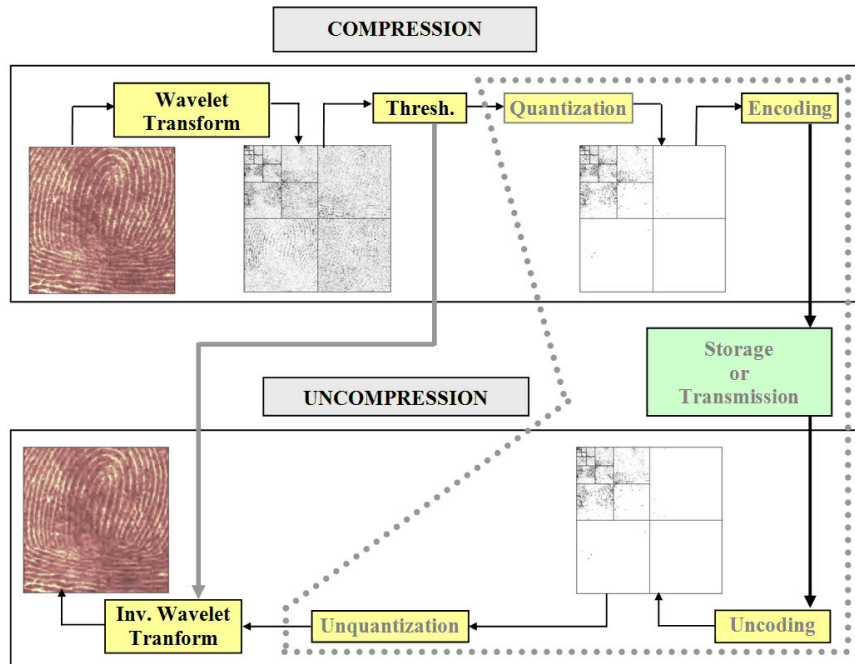


Figure IV.1 Wavelet Compression for Images

IV.2.3 MATLAB wavelet toolbox

We are going to use '*True compression of images using wavelets*' this is how is titled in MATLAB documentation

The context of '*wcompress*' tool looks like this

`wcompress('c', X, cname, compmthd)`

- **C:** stands for compression.
- **X, fname:** Input image to compress, specified as a 2-D array containing an indexed image or a 3-D array of uint8 containing a truecolor image. Both the row and column size of the image must be powers of two.
- **Cname:** Compressed image filename, specified as a character vector or string scalar. The `wcompress` function writes the compressed image to the file `cname` with extension. `wc` at the end.

CHAPTER IV: Results and Discussion

- **Compmethd:** Compression method, specified as a character vector or string scalar. The valid compression methods are divided into two categories **PCSM**, **CTM**.

'*wcompress*' tool can return two value such as compression ratio '*comprat*' and the bit-per-pixel ratio '*bpp*'. As we see below:

```
[comprat,bpp] = wcompress('c',X,'wpeppers.wtc','spiht','maxloop',11)
```

- **Maxloop:** Number of loops for compression algorithm encoding steps
- **11:** Is the number of steps

And for uncompressing image this is the context below

```
Xc = wcompress('u',cname)
```

- **U:** Stands for uncompressing
- **Cname:** Compressed image filename, specified as a character vector or string scalar. The *wcompress* function reads the compressed image from the file *cname* for uncompressing.

Example:

```
% Compress  
load mask;  
X = imread('wpeppers.jpg');  
[cratio,bpp] = wcompress('c',X,'wpeppers.wtc','spiht','maxloop',12)
```

Result:

```
cratio = 1.6549
```

```
bpp = 0.3972
```

```
% uncompressed  
Xc = wcompress('u','wpeppers.wtc');  
delete('wpeppers.wtc')
```

IV.2.4 MATLAB GUI (Graphic User Interface)

We use the GUI to create the interface for image compression for more efficiently result and the ease of use,

When you first open up the interface, you'll see 6 sections as you see in the figure below which contains:

- File
- File directory

- Compression Parameter
- Results
- Performance
- Process

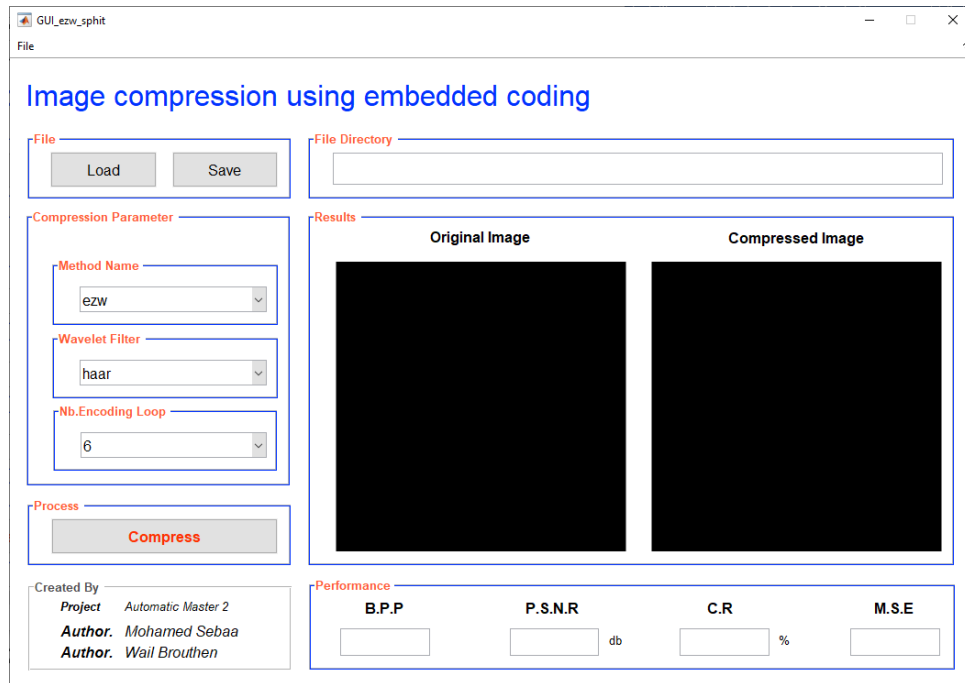


Figure IV.2 GUI (Graphic User Interface) MATLAB for the image compression

IV.2.4.1 The interface manipulation

The way to use the interface for image compression is to first click on **Load** button it will pop up a window directory where you choose the image, after this you specify **Method name** by clicking on the popup-menu, then you set the **Wavelet Filter** by clicking into the popup-menu, you set the **Nb.encoding Loop** to which the encoding steps, Finally, you hit **Compress** button.

After the compression is done the **compressed image** in the shows up with the **Performance** metrics **B.P.P**, **P.S.N.R**, **C.R**, **M.S.E**

If we want to save the compressed image, we click on **Save** button it will generate the compressed image as a bitmap **.bmp** extension file

➤ Figure below describe how to load image

CHAPTER IV: Results and Discussion

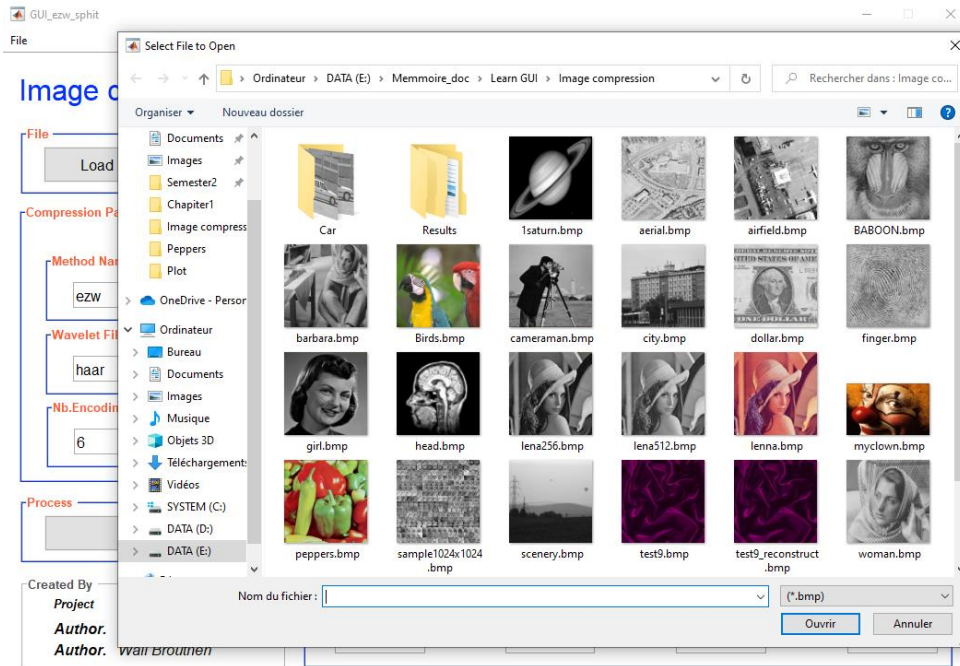


Figure IV.3 GUI (Graphic User Interface) MATLAB load image

- Figure below describe how to set the method name

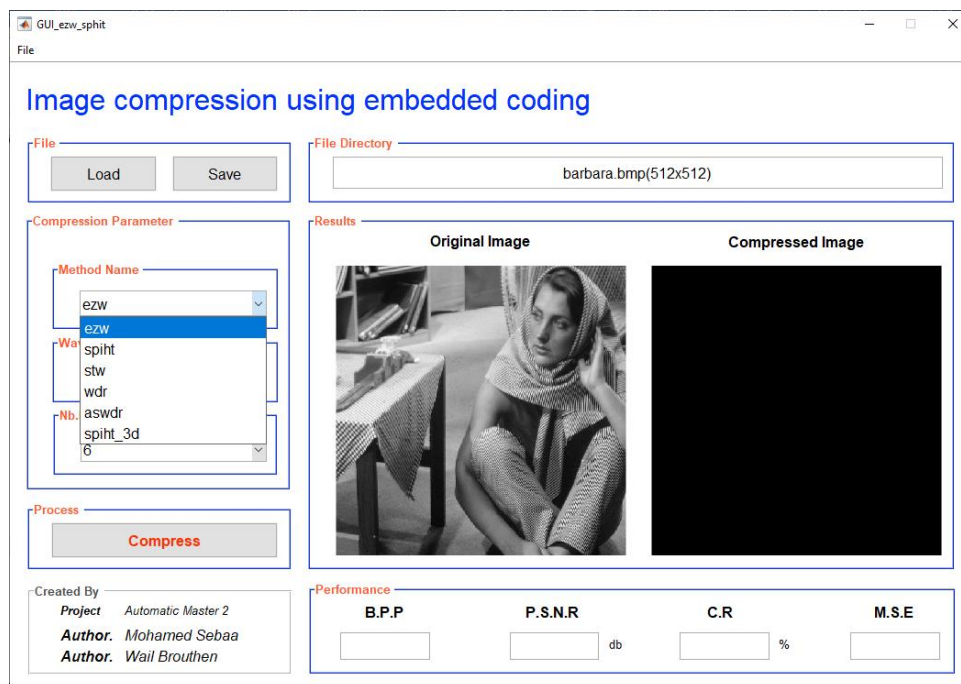


Figure IV.4 GUI (Graphic User Interface) MATLAB set the Method Name

- Figure below describe how to set the Filter Wavelet

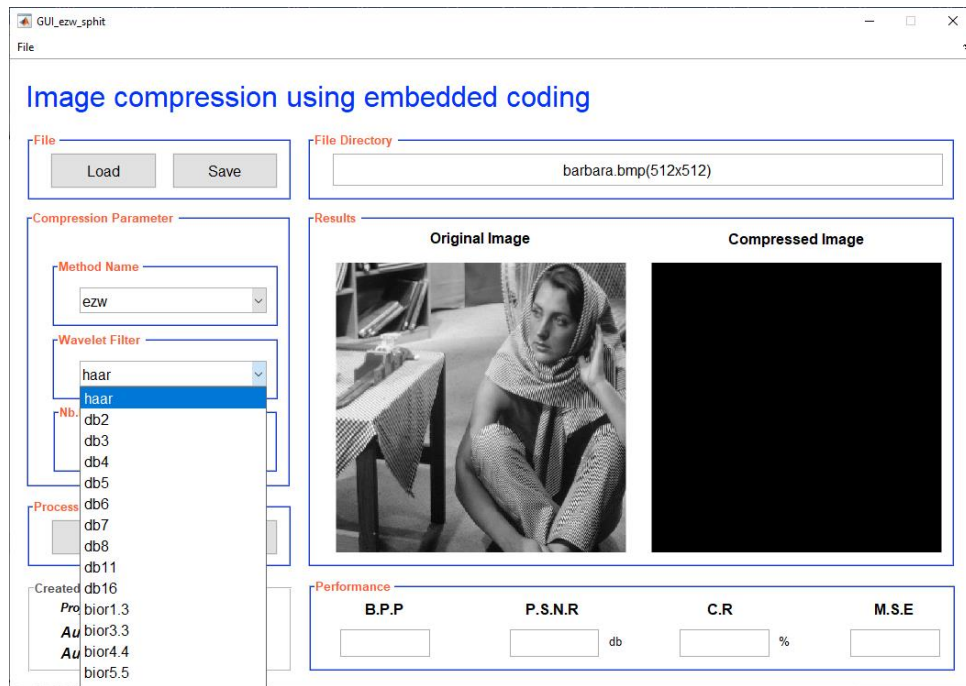


Figure IV.5 GUI (Graphic User Interface) MATLAB set the Filter Wavelet

➤ Figure below describe how to set the Number of encoding loop

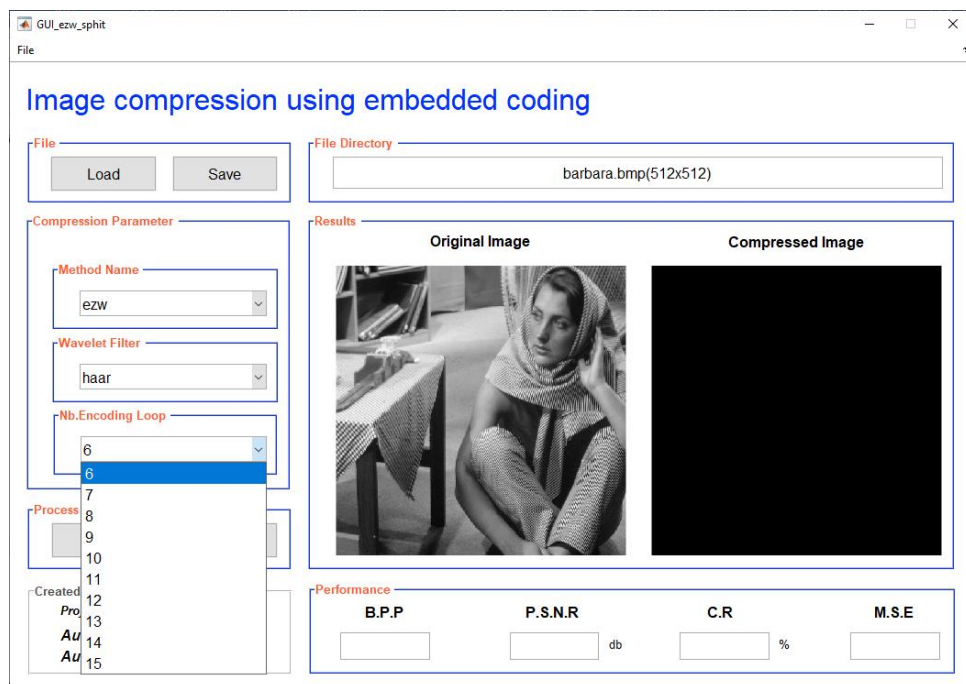


Figure IV.6 GUI (Graphic User Interface) MATLAB set the number of encoding loop

➤ Figure below describe the result after clicking on the compress button

CHAPTER IV: Results and Discussion

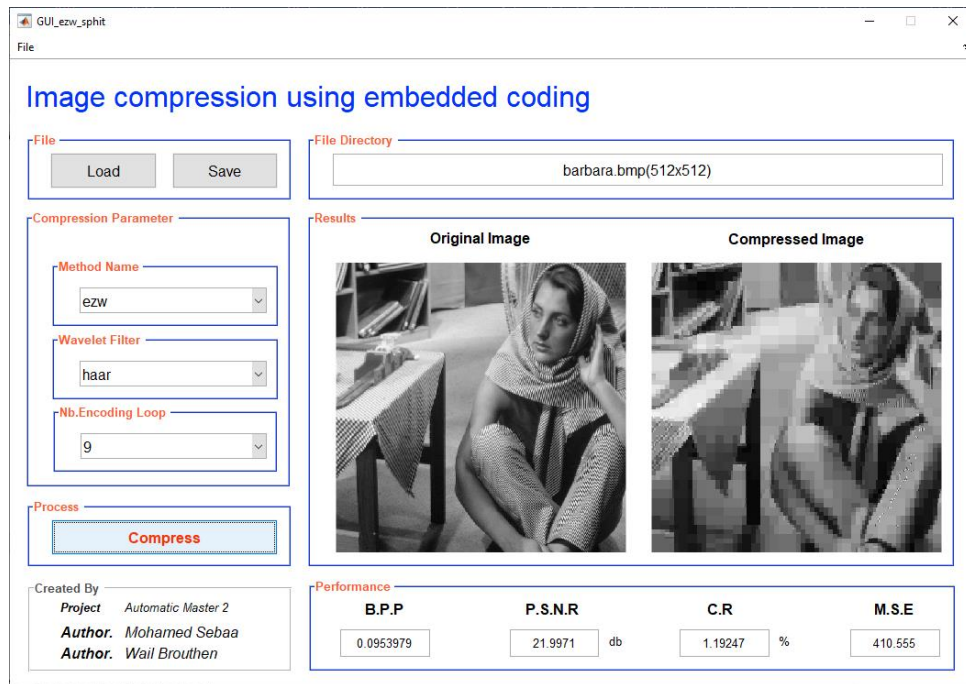


Figure IV.7 GUI (Graphic User Interface) MATLAB results after click on compress button

➤ Figure below describe how to save the compressed image

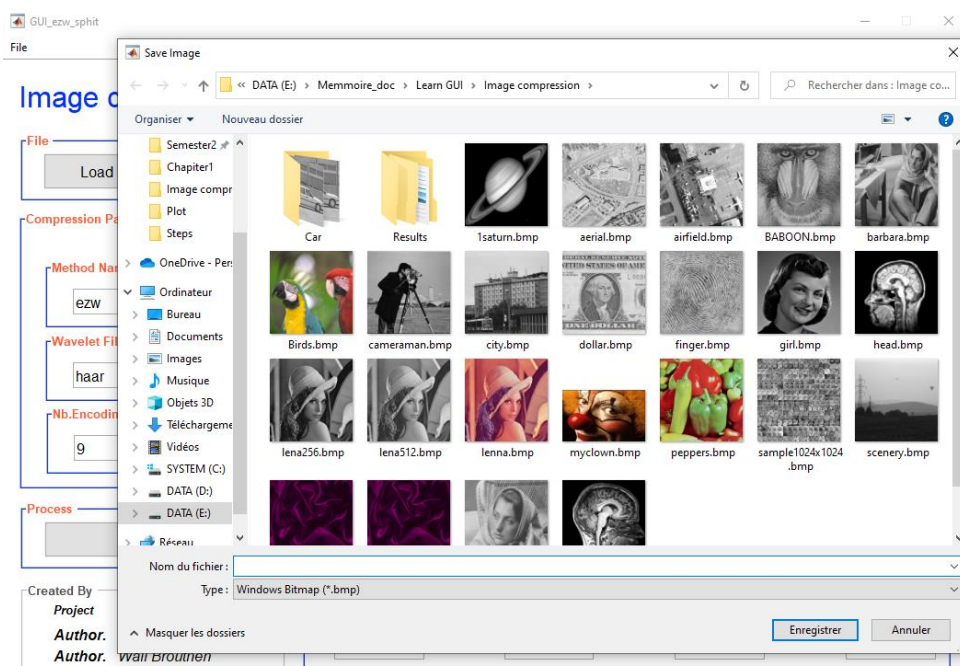


Figure IV.8 GUI (Graphic User Interface) MATLAB save the compressed image

IV.3 Practical Work

Our program engraved 4 grayscale images (8-bit) and 2 images in RGB (24-bit) And with different puzzles (512x512) & (256x256)

Image Name	Bit coding	size	Figure
Lena	8-bit	(256x256) pixel	Fig 1
Barbara	8-bit	(512x512) pixel	Fig 2
X-ray	24-bit	(512x512) pixel	Fig 3
Cameraman	8-bit	(256x256) pixel	Fig 4
Woman	8-bit	(256x256) pixel	Fig 5
Peppers	24-bit	(512x512) pixel	Fig 6

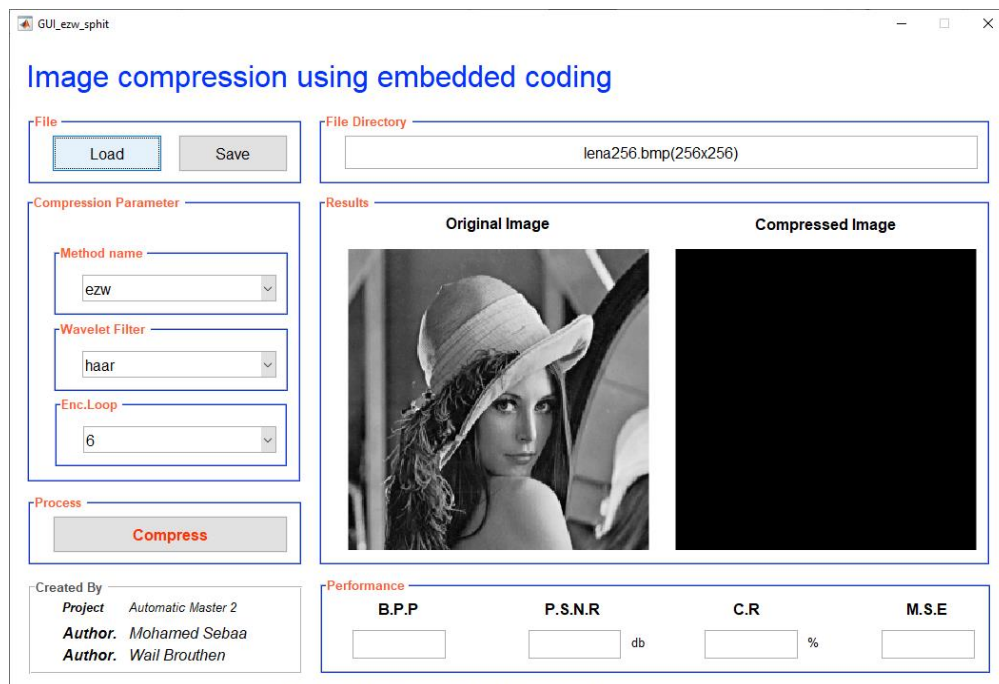


Figure IV.9 The Original Image Lena (512x512) 8-bit

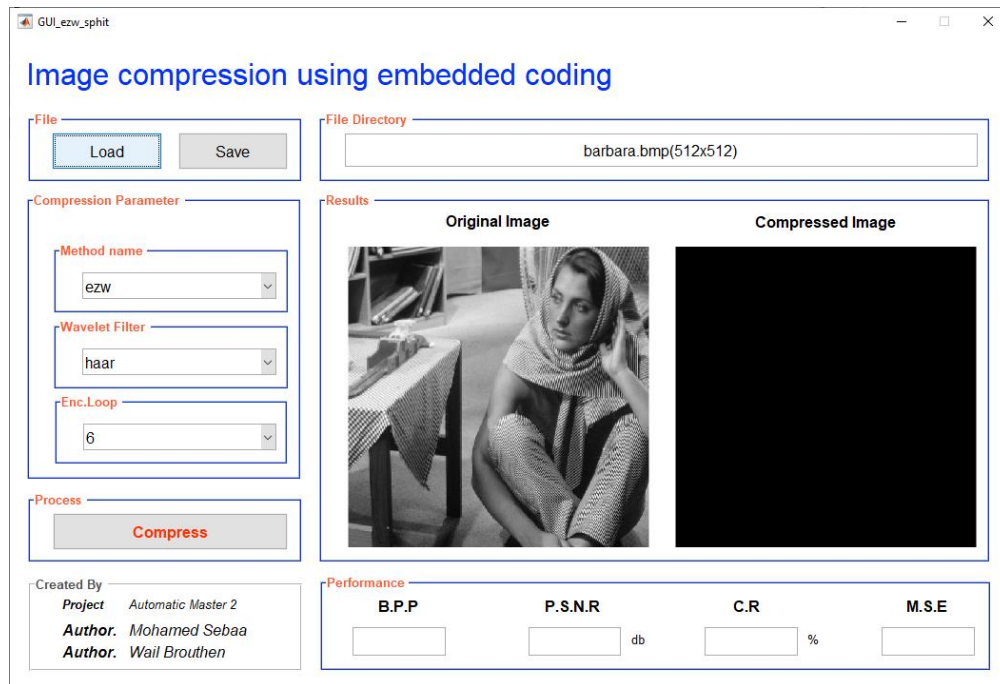


Figure IV.10 The Original Image Barbara (512x512) 8-bit

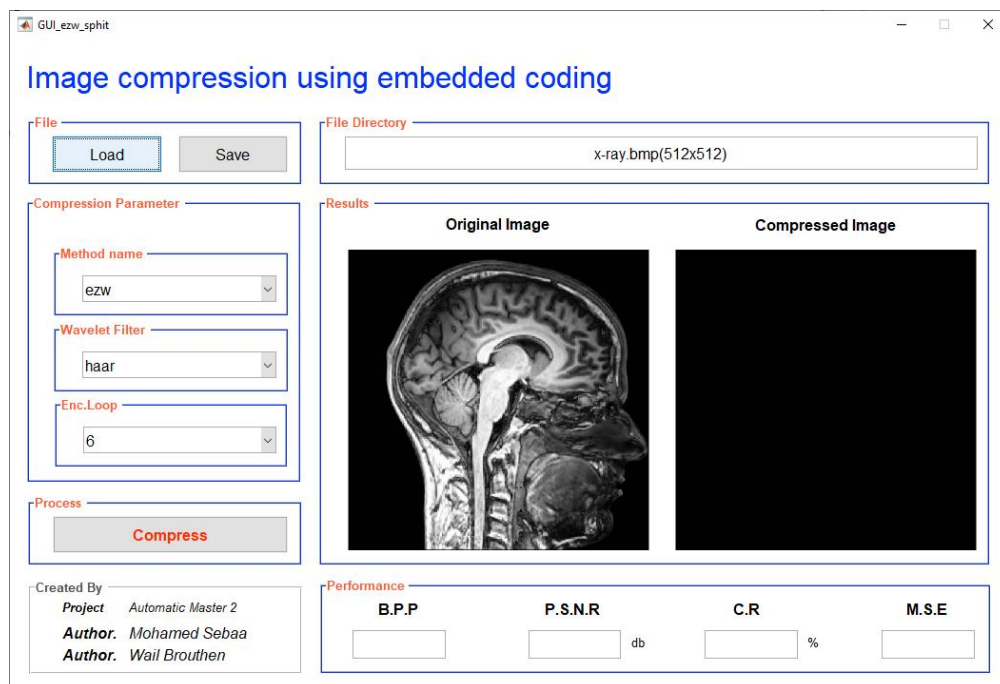


Figure IV.11 The Original Image X-ray (512x512) 24-bit

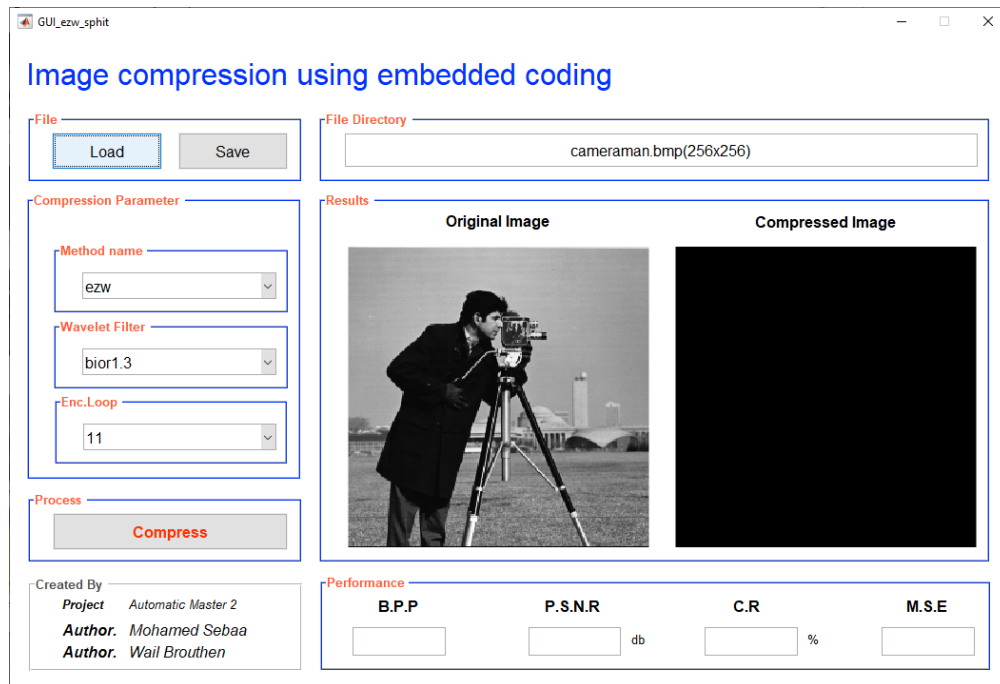


Figure IV.12 The Original Image Cameraman (256x256) 8-bit

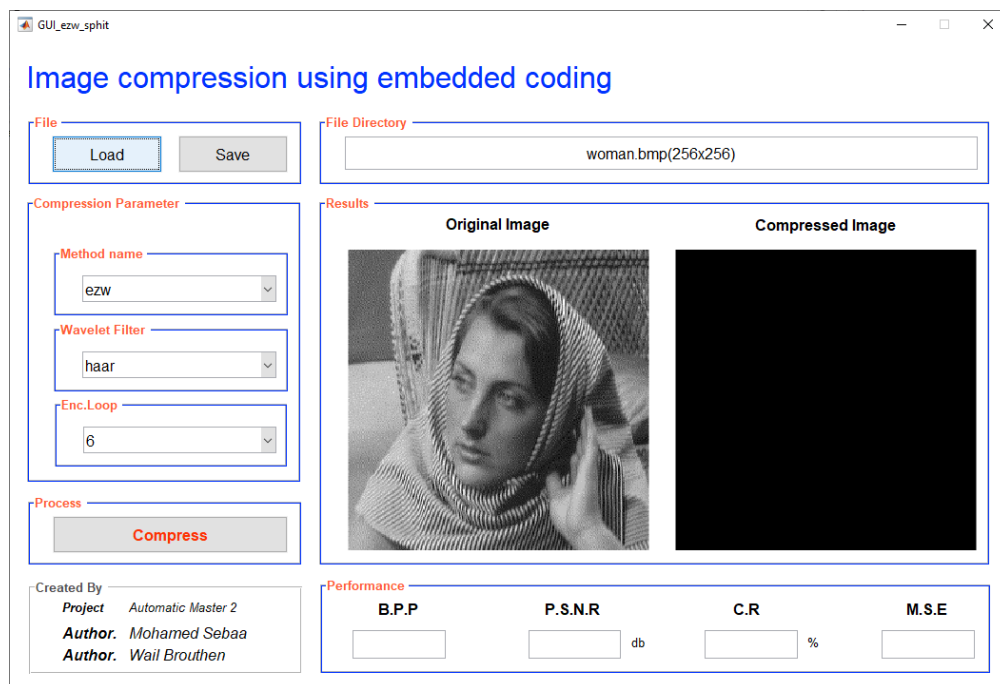


Figure IV.13 The Original Image Woman (256x256) 8-bit

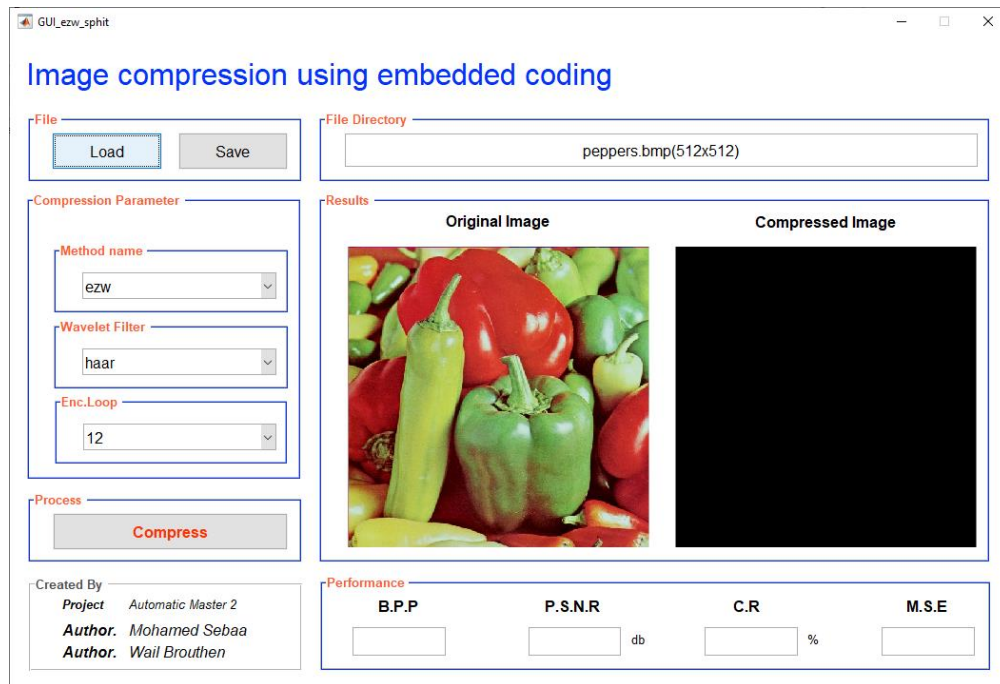


Figure IV.14 The Original Image Peppers (512x512) 24-bit

IV.3.1 Wavelet Filters

The Discrete Wavelet Transformation (DWT) is performed after selecting the main wavelet type that is used the perfect builder of an image.

Table IV-1 Types of wavelets used

Wavelet	Waves as they are termed in MATLAB
Haar	haar
Daubechies	db2, db3, db4, db5, db6, db7, ..., db45
Biorthogonal	bior1.3, bior2.2, bior3.3, bior4.4, bior6.8

IV.3.2 Decomposition levels

Another very important criterion in compression is the level of decomposition of the wavelet transform in our work the toolbox '*wcompress*' set the compression level to the max means it depends on every wavelet transform and the image size and for more efficiently.

CHAPTER IV: Results and Discussion

IV.4 Results and discussion

The tables below explain the *EZW* results which is been applied to many different sizes of images with changing wavelet type (Haar, bior3.3, db7) and different loop number.

IV.4.1 Lena Image (256x256) 8-bit Results Table

Table IV-2 Lena Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps

Enc.Steps	Haar					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	20.2231	0.78	0.06	17.7937	0.24	0.019
8	22.7607	1.92	0.15	19.8963	0.52	0.04
9	25.8549	4.56	0.36	22.3862	1.36	0.1
10	29.6065	10.01	0.80	25.2936	3.21	0.25
11	33.7327	19.23	1.53	28.8152	6.93	0.5
12	37.8203	33.01	2.64	32.7016	12.94	1.03

Table IV-3 Lena Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps

Enc.Steps	Bior3.3					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	22.6576	1.83	0.14	19.9942	0.58	0.04
8	25.258	3.78	0.30	22.3452	1.2	0.09
9	27.876	6.73	0.53	24.8014	2.44	0.19
10	30.95	11.59	0.92	27.4156	4.38	0.35
11	34.3085	19.36	1.54	30.4198	7.53	0.6
12	37.8909	30.92	2.47	33.6253	12.49	0.99

Table IV-4 Lena Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps

Enc.Steps	Db7					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP

CHAPTER IV: Results and Discussion

7	20.9364	0.78	0.06	18.0575	0.23	0.01
8	23.393	1.96	0.13	20.5557	0.94	0.03
9	26.4588	3.98	0.31	22.9666	1.09	0.08
10	30.2393	8.77	0.70	25.8929	2.56	0.2
11	34.3133	16.90	1.35	29.4307	5.66	0.45
12	38.204	28.91	2.31	33.2804	10.70	0.85

IV.4.2 The Reconstructed Image Lena (256x256) 8-bit

The Results are based on PSNR, in the range of 30.



EZW, Haar, 11 Step

PSNR: 33.7327, CR: 11.59, BPP:0.92



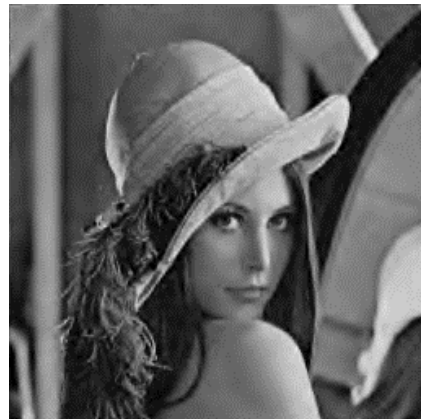
SPIHT, Haar, 11 Step

PSNR: 28.8152, CR:6.93, BPP: 0.3



EZW, Bior3.3, 10 Step

PSNR:30.95, CR:12.94, BPP:1.03



SPIHT, Bior3.3, 11 Step

PSNR: 30.4198, CR:7.53, BPP:0.3



EZW, Db7, 10 Step

PSNR: 30.2393, CR:8.77, BPP:0.70

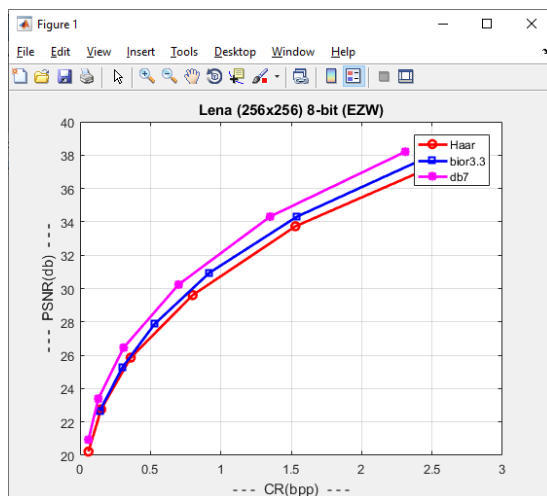


SPIHT, Db7, 12 Step

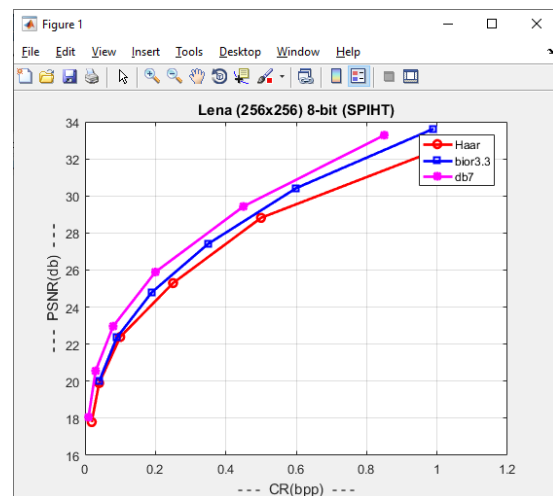
PSNR: 33.2804, CR:10.70, BPP:0.85

Figure IV.15 The Reconstructed image Lena (265x256) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods

IV.4.3 The Effect of wavelets on Lena Image (256x256) 8-bit



EZW



SPIHT

Figure IV.16 Comparing Different wavelets for Lena Image for both EZW & SPIHT

CHAPTER IV: Results and Discussion

IV.4.4 Barbara Image (512x512) 8-bit Results Table

Table IV-5 Barbara Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps

Enc.Steps	Haar					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	18.7618	0.2	0.01	16.7729	0.06	0.004
8	20.4734	0.5	0.04	18.5178	0.13	0.01
9	21.9971	1.19	0.09	20.2115	0.32	0.02
10	24.5702	3.65	0.29	21.7598	0.83	0.06
11	28.5555	10.19	0.81	24.2293	2.69	0.21
12	33.3786	21.46	1.71	27.8481	7.24	0.57
13	37.95	36.74	2.93	32.1442	14.36	1.14

Table IV-6 Barbara Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps

Enc.Steps	Bior3.3					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	20.6353	0.5	0.04	18.5872	0.14	0.01
8	22.0782	1.35	0.1	20.4621	0.32	0.02
9	23.5135	3.16	0.25	21.8799	0.85	0.06
10	26.0712	6.39	0.51	23.3409	2.04	0.16
11	29.6273	11.69	0.93	25.7718	4.28	0.34
12	33.2441	19.84	1.5	28.9978	7.71	0.61
13	37.0541	31.59	2.52	32.3973	12.84	1.027

Table IV-7 Barbara Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps

Enc.Steps	Db7					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP

CHAPTER IV: Results and Discussion

7	19.2769	0.18	0.01	17.3913	0.05	0.004
8	21.0662	0.42	0.03	19.0298	0.11	0.009
9	22.7468	1.09	0.08	20.789	0.28	0.02
10	26.011	3.46	0.27	22.4988	0.75	0.06
11	30.1196	8.34	0.66	25.533	2.46	0.19
12	34.38	16.27	1.3	29.2365	5.67	0.45
13	38.4845	28.30	2.26	33.2415	10.65	0.85

IV.4.5 The Reconstructed Image Barbara (512x512) 8-bit

The Results are based on PSNR, in the range of 30.



EZW, Haar, 12 Step

PSNR: 33.3786, **CR:** 21.46, **BPP:** 1.71



SPIHT, Haar, 13 Step

PSNR: 32.1442, **CR:** 14.36, **BPP:** 1.14



EZW, Bior3.3, 11 Step

PSNR: 29.6273, **CR:** 11.69, **BPP:** 0.93



SPIHT, Bior3.3, 13 Step

PSNR: 32.3973, **CR:** 12.84, **BPP:** 1.027



EZW, Db7, 11 Step

PSNR: 30.1196, CR: 8.34, BPP: 0.66

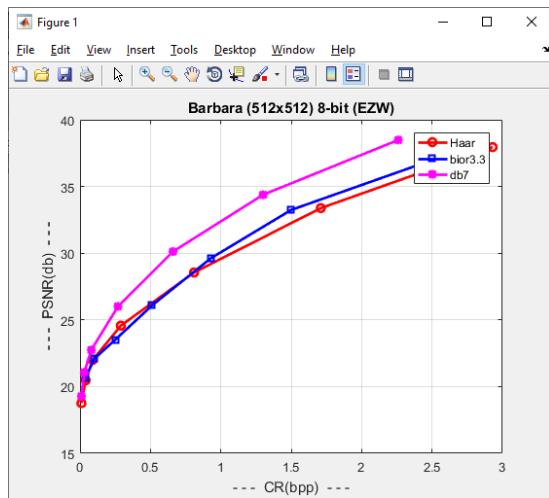


SPIHT, Db7, 12 Step

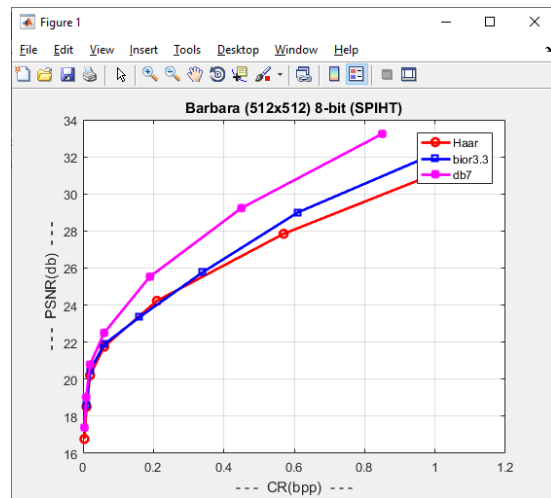
PSNR: 29.2365, CR: 5.67, BPP: 0.45

Figure IV.17 The Reconstructed image Barbara (512x512) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods

IV.4.6 The Effect of wavelets on Barbara Image (512x512) 8-bit



EZW



SPIHT

Figure IV.18 Comparing Different wavelets for Barbara Image for both EZW & SPIHT

IV.4.7 X-ray Image (512x512) 24-bit Results Table

Table IV-8 X-ray Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps

	Haar					
	EZW			SPIHT		
Enc.Steps	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
11	30.1196	8.34	0.66	29.2365	5.67	0.45

CHAPTER IV: Results and Discussion

7	29.8935	0.17	0.04	29.7133	0.13	0.03
8	30.4961	0.54	0.13	30.2319	0.39	0.09
9	31.1913	1.54	0.37	30.9719	1.06	0.25
10	32.322	3.66	0.8	31.9392	2.57	0.6
11	34.3014	8.14	1.95	33.7259	5.53	1.32
12	37.8967	15.82	3.79	36.7412	10.48	2.51

Table IV-9 X-ray Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps

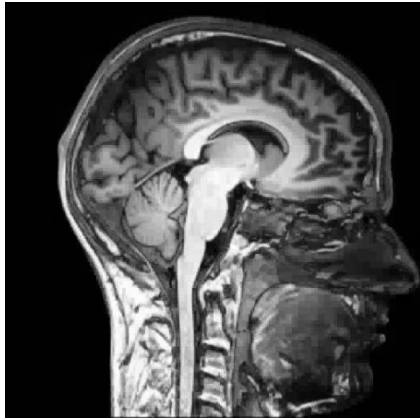
Enc.Steps	Bior3.3					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	30.5871	0.62	0.14	30.1569	0.17	0.04
8	31.2785	1.29	0.31	30.509	0.41	0.1
9	32.5264	2.59	0.62	31.153	0.87	0.21
10	34.3722	4.70	1.13	32.2096	1.70	0.4
11	36.3942	7.67	1.84	33.8474	3.05	0.73
12	39.3207	12.59	3.02	35.861	5.03	1.2

Table IV-10 X-ray Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps

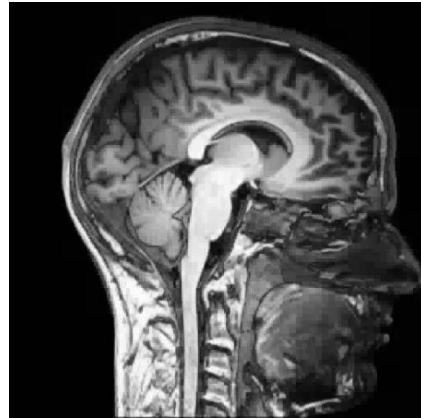
Enc.Steps	Db7					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	30.0701	0.19	0.04	29.8927	0.13	0.033
8	30.6431	0.56	0.13	30.4453	0.38	0.092
9	31.6667	1.36	0.32	31.3359	0.93	0.22
10	33.3472	3.08	0.74	32.8247	2.00	0.48
11	35.7079	5.88	1.41	34.9894	3.87	0.92
12	39.2826	11.04	2.65	38.212	7.24	1.73

IV.4.8 The Reconstructed Image X-ray (512x512) 24-bit

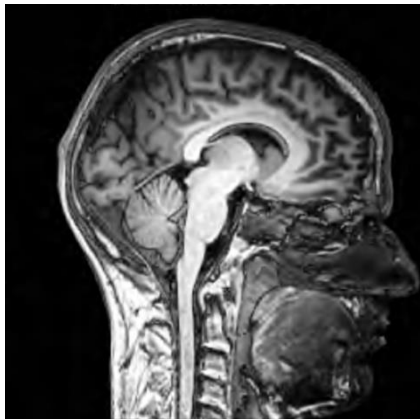
The Results are based on PSNR, in the range of 34.



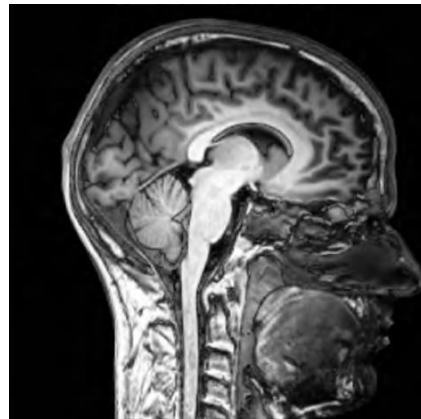
EZW, Haar, 10 Step
PSNR: 32.322, CR: 3.66, BPP: 0.8



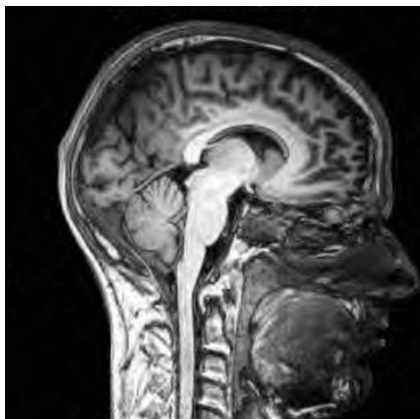
SPIHT, Haar, 11 Step
PSNR: 33.7259, CR: 5.53, BPP: 1.32



EZW, Bior3.3, 10 Step
PSNR: 34.3722, CR: 4.70, BPP: 1.13



SPIHT, Bior3.3, 12 Step
PSNR: 35.861, CR: 5.03, BPP: 1.2



EZW, Db7, 10 Step
PSNR: 33.3472, CR: 3.08, BPP: 0.74



SPIHT, Db7, 11 Step
PSNR: 34.9894, CR: 3.87, BPP: 0.92

Figure IV.19 The Reconstructed image X-ray (512x512) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods

IV.4.9 The Effect of wavelets on X-ray Image (512x512) 24-bit

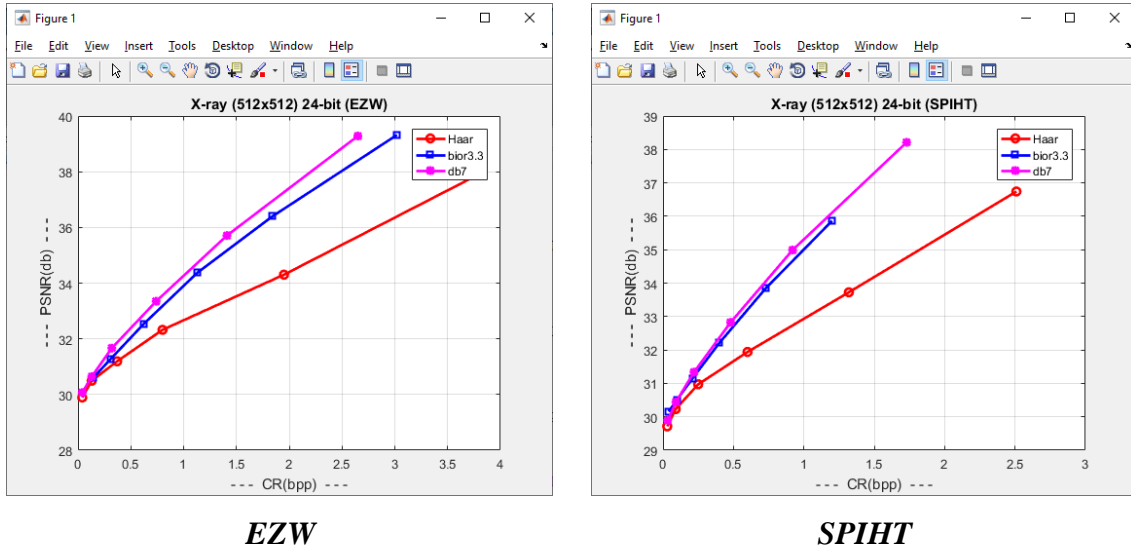


Figure IV.20 Comparing Different wavelets for X-ray Image for both EZW & SPIHT

IV.4.10 Cameraman Image (256x256) 8-bit Results Table

Table IV-11 Cameraman Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps

Enc.Steps	Haar					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	19.8037	0.76	0.06	17.7418	0.23	0.019
8	22.607	2.05	0.16	19.5346	0.52	0.04
9	26.3524	4.84	0.38	22.1894	1.50	0.12
10	30.3593	10.04	0.8	25.588	3.53	0.28
11	34.8477	18.38	1.47	29.3979	7.25	0.58
12	39.18	29.48	2.35	33.6228	12.95	1.03

CHAPTER IV: Results and Discussion

Table IV-12 Cameraman Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps

Enc.Steps	Bior3.3					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	21.2851	2.02	0.16	19.1902	0.58	0.04
8	23.8703	3.87	0.31	21.1138	1.26	0.10
9	27.099	7.23	0.57	23.5106	2.50	0.2
10	30.299	12.57	1.00	26.6391	4.71	0.37
11	34.0574	20.83	1.66	29.7151	8.23	0.65
12	38.0724	32.32	2.58	33.2372	13.54	1.08

Table IV-13 Cameraman Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps

Enc.Steps	Db7					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	19.6449	0.77	0.06	17.4422	0.22	0.017
8	22.682	2.10	0.16	19.3806	0.51	0.04
9	26.019	5.03	0.4	22.2493	1.43	0.11
10	29.7865	10.44	0.83	25.3458	3.29	0.26
11	34.3068	19.65	1.57	28.9422	6.82	0.54
12	38.8462	32.21	2.57	33.1209	12.75	1.02

IV.4.11 The Reconstructed Image Cameraman (256x256) 8-bit

The Results are based on PSNR, in the range of 30.



EZW, Haar, 10 Step

PSNR: 30.3593, CR: 10.04, BPP: 0.8



SPIHT, Haar, 11 Step

PSNR: 29.3979, CR: 7.25, BPP: 0.58



EZW, Bior3.3, 10 Step

PSNR: 30.299, CR: 12.57, BPP: 1.00



SPIHT, Bior3.3, 11 Step

PSNR: 29.7151, CR: 8.23, BPP: 0.65



EZW, Db7, 10 Step

PSNR: 29.7865, CR: 10.44, BPP: 0.83



SPIHT, Db7, 12 Step

PSNR: 33.1209, CR: 12.75, BPP: 1.02

Figure IV.21 The Reconstructed image Cameraman (256x256) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods

CHAPTER IV: Results and Discussion

IV.4.12 The Effect of wavelets on Cameraman Image (512x512) 24-bit

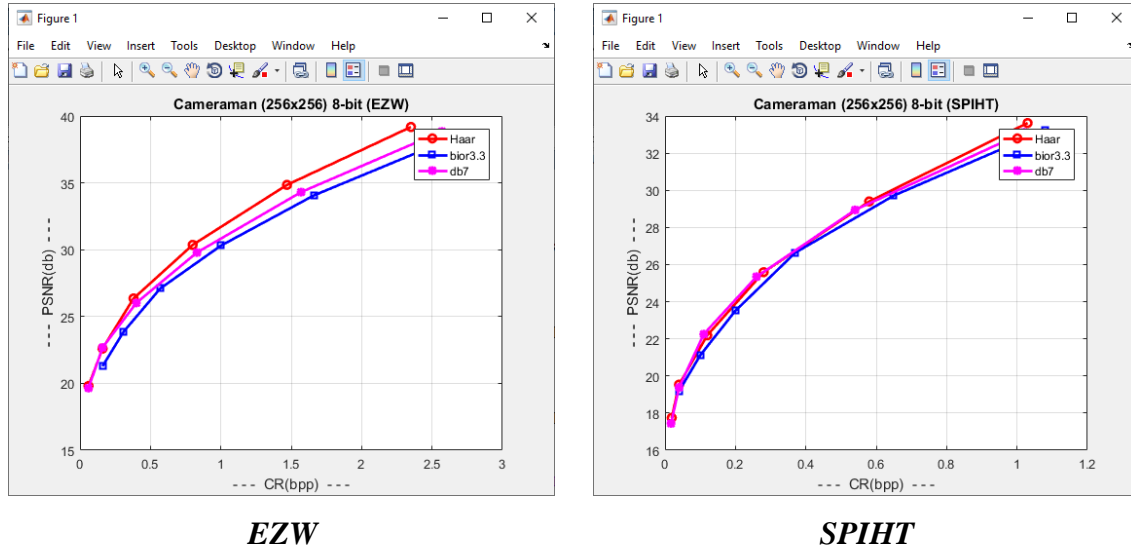


Figure IV.22 Comparing Different wavelets for Cameraman Image for both EZW & SPIHT

IV.4.13 Woman Image (256x256) 8-bit Results Table

Table IV-14 Woman Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps

Enc.Steps	Haar					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	18.5879	0.5	0.04	17.3768	0.18	0.01
8	19.9668	1.52	0.12	18.4485	0.34	0.02
9	22.5559	5.56	0.44	19.8102	1.08	0.08
10	25.8026	15.42	1.23	22.2248	3.93	0.31
11	30.6775	35.54	2.84	25.2649	10.93	0.87
12	36.5778	61.56	4.92	29.5823	25.00	2.00

CHAPTER IV: Results and Discussion

Table IV-15 Cameraman Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps

Enc.Steps	Bior3.3					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	19.5648	1.91	0.15	18.5311	0.48	0.03
8	20.904	4.33	0.34	19.493	1.17	0.09
9	23.4009	9.25	0.74	20.806	2.95	0.23
10	26.2309	18.58	1.48	23.1308	6.10	0.48
11	29.4667	37.74	3.01	25.7628	12.41	0.99
12	34.706	63.61	5.08	28.9121	25.21	2.017

Table IV-16 Cameraman Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps

Enc.Steps	Db7					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	18.9742	0.53	0.04	17.6761	0.18	0.01
8	20.4763	1.62	0.13	18.8181	0.35	0.02
9	23.39	5.52	0.44	20.3184	1.17	0.09
10	26.4403	13.61	1.08	22.9967	3.82	0.3
11	30.988	31.92	2.55	25.8834	9.49	0.75
12	36.975	57.77	4.62	30.0229	22.88	1.83

IV.4.14 The Reconstructed Image Woman (256x256) 8-bit

The Results are based on PSNR, in the range of 30.

CHAPTER IV: Results and Discussion



EZW, Haar, 11 Step

PSNR: 30.6775, CR: 35.54, BPP: 2.84



SPIHT, Haar, 12 Step

PSNR: 29.5823, CR: 25.00, BPP: 2.00



EZW, Bior3.3, 11 Step

PSNR: 29.4667, CR: 37.74, BPP: 3.01



SPIHT, Bior3.3, 12 Step

PSNR: 28.9121, CR: 25.21, BPP: 2.017



EZW, Db7, 11 Step

PSNR: 30.988, CR: 31.92, BPP: 2.55



SPIHT, Db7, 12 Step

PSNR: 30.0229, CR: 22.88, BPP: 1.83

Figure IV.23 The Reconstructed image Woman (256x256) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods

IV.4.15 The Effect of wavelets on Woman Image (256x256) 8-bit

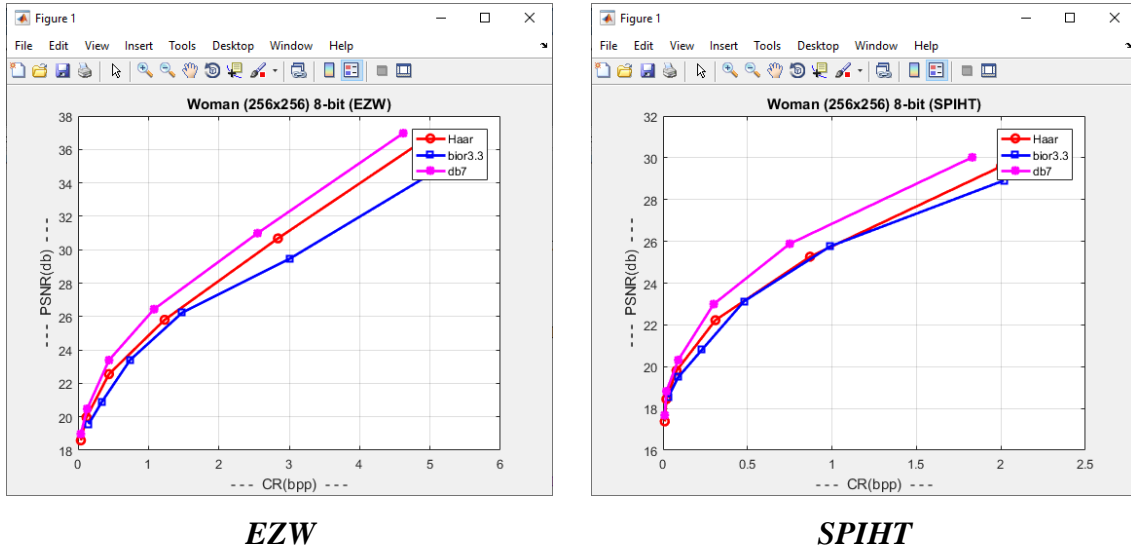


Figure IV.24 Comparing Different wavelets for Woman Image for both EZW & SPIHT

IV.4.16 Peppers Image (512x512) 24-bit Results Table

Table IV-17 Peppers Image Result value for EZW & SPIHT Embedded coding with Haar wavelet and different value of encoding steps

Enc.Steps	Haar					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	28.638	0.07	0.01	28.2931	0.05	0.01
8	29.5581	0.17	0.04	29.1164	0.12	0.03
9	30.696	0.43	0.10	30.2789	0.31	0.07
10	31.8601	1.10	0.26	31.5138	0.79	0.19
11	33.1851	2.55	0.61	32.7587	1.81	0.43
12	34.7334	5.42	1.30	34.3324	3.76	0.90

CHAPTER IV: Results and Discussion

Table IV-18 Peppers Image Result value for EZW & SPIHT Embedded coding with Bior3.3 wavelet and different value of encoding steps

Enc.Steps	Bior3.3					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	29.328	0.20	0.04	28.3463	0.06	0.01
8	30.3734	0.44	0.10	29.0743	0.13	0.03
9	31.6657	0.89	0.21	30.0117	0.28	0.06
10	33.0978	1.78	0.42	31.3626	0.59	0.14
11	34.622	3.38	0.81	32.7531	1.17	0.28
12	36.1241	6.28	1.50	34.2918	2.25	0.54

Table IV-19 Peppers Image Result value for EZW & SPIHT Embedded coding with Db7 wavelet and different value of encoding steps

Enc.Steps	Db7					
	EZW			SPIHT		
	PSNR (db)	CR (%)	BPP	PSNR (db)	CR (%)	BPP
7	28.6721	0.07	0.017	28.4148	0.04	0.01
8	29.5831	0.17	0.04	29.2542	0.11	0.02
9	30.6377	0.40	0.09	30.3543	0.26	0.06
10	31.9927	0.92	0.22	31.6347	0.61	0.14
11	33.5432	2.13	0.51	33.159	1.42	0.34
12	35.3138	4.47	1.07	34.8513	2.93	0.7

IV.4.17 The Reconstructed Image Peppers (512x512) 24-bit

The Results are based on PSNR, in the range of 34.



EZW, Haar, 12 Step
PSNR: 34.7334, CR: 5.42, BPP: 1.30



SPIHT, Haar, 12 Step
PSNR: 34.3324, CR: 3.76, BPP: 0.90



EZW, Bior3.3, 11 Step
PSNR: 34.622, CR: 3.38, BPP: 0.81



SPIHT, Bior3.3, 12 Step
PSNR: 34.2918, CR: 2.25, BPP: 0.54



EZW, Db7, 11 Step
PSNR: 33.5432, CR: 2.13, BPP: 0.51



SPIHT, Db7, 12 Step
PSNR: 33.159, CR: 2.93, BPP: 0.7

Figure IV.25 The Reconstructed image Peppers (512x512) using wavelet Haar, Bior3.3, Db7 with different steps of encoding for both EZW & SPIHT methods

IV.4.18 The Effect of wavelets on Peppers Image (512x512) 24-bit

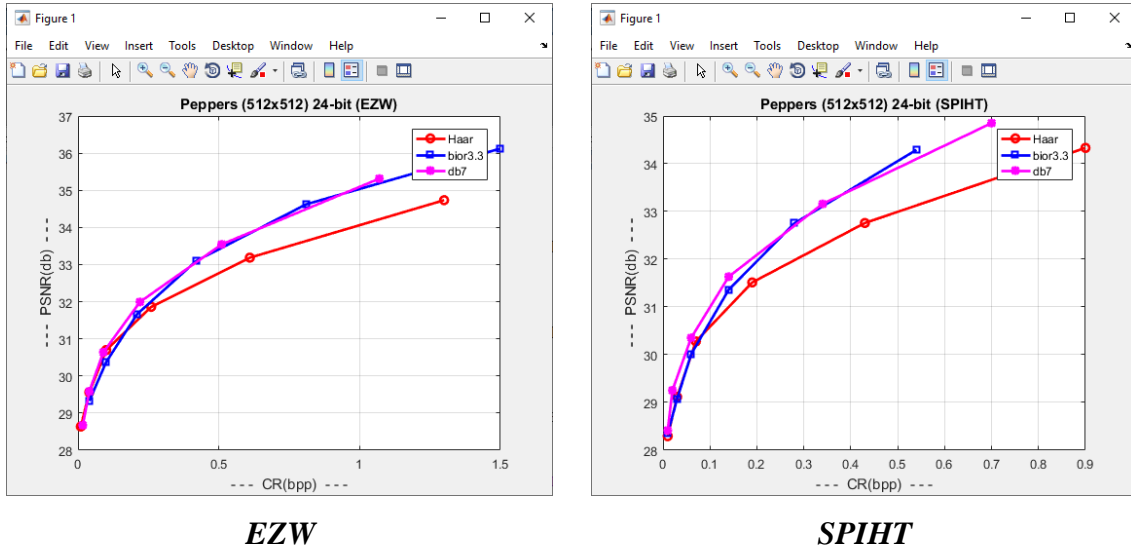


Figure IV.26 Comparing Different wavelets for Peppers Image for both EZW & SPIHT

IV.5 discussion and conclusion

The obtained results are based on MATLAB toolbox ‘*wcompress*’ this tool has two kind of methods CTM (Coefficient Thresholding Method) and PCSM (Progressive Coefficients Significance Methods), for our study we used PCSM which contains EZW, SPHIT methods and other advanced methods.

The key parameter is the *maxloop* (number of loops) stands for encoding steps; increasing it leads to better recovery, but a worse compression ratio.

‘*wcompress*’ output the compression ratio and Bit-per-pixel

Starting from the results shown in the tables and figures, using the *EZW* & *SPIHT* coding algorithm, and applying Three types of filters (Haar, bior3.3, db7) on images of different sizes and depending on the performance values of each *PSNR*, *CR*, *BPP*.

In the case of High number steps of encoding PSNR increase

Which represents the quality of the image reconstruction, so that it is

The recovered image is close to the original image, and this corresponds to an increase in the resulting Compression ratio.

CHAPTER IV: Results and Discussion

The low encoding steps condition indicates a low image recovery rate and thus a low compression ratio.

IV.6 Based on results

We've Applied filters Haar, bior3.3 and DB7 on different images with different steps and methods, we get the following based on performance metrics:

IV.6.1 EZW methods performance

For 8-bit images we are looking for PSNR around 30 low compression ratio and good quality image

- Lena image (256x256): Performed well when filter db7
- Barbara image(512x512): Performed well when filter db7
- cameraman image(256x256): Performed well when filter haar
- Woman image (256x256): Performed well when filter db7

For 24-bit images we are looking for PSNR around 34 low compression ratio and good quality image

- X-ray image (512x512): Performed well when filter db7
- Peppers image (512x512): Performed well when filter db7

IV.6.2 SPIHT methods performance

For 8-bit images we are looking for PSNR around 30 low compression ratio and good quality image

- Lena image (256x256): Performed well when filter haar
- Barbara image(512x512): Performed well when filter db7
- cameraman image(256x256): Performed well when filter haar
- Woman image (256x256): Performed well when filter db7

For 24-bit images we are looking for PSNR around 34 low compression ratio and good quality image

- X-ray image (512x512): Performed well when filter db7
- Peppers image (512x512): Performed well when filter bior3.3

As we saw EZW, SPIHT performs well on db7 filter

CHAPTER IV: Results and Discussion

These results are relative because there might be another filter version like in bior and db can perform differently but overall as we noticed that SPIHT perform better than EZW of course because it's more advanced.

IV.7 Conclusion

In this work, good results were obtained, based on several measures: PSNR, CR and BPP ratio. Compression is affected by the size of the image used, the type of filters used, and the steps of encoding (number of loops) value that is used specifically. The SPIHT methods performed better than EZW because its more advanced, there are other methods in '*wcompress*' tool we didn't mention but they are more advanced but for the purpose of the study we discussed about the famous one

So, the embedded coding make a huge difference in compressing image.

General Conclusion

The work carried out in this dissertation focused on the compression image using more advanced embedded coding with the MATLAB tool and creating an easy-to-use interface using MATLAB GUI.

The compression of the image does not necessarily mean a decrease in quality, but in fact it is the elimination of Data that is unnecessary and that cannot be seen by human eyes. The embedded EZW and SPIHT coding is among the compression methods that depend on the discontinuous wavelets that are addressed in this the note, which provides the feature of gradual transmission of an encrypted image while achieving distinctive performance and high efficiency.

The use of GUI in this work makes it easy for a regular person to get more done with just a click of a button, although the created interface still does not reach the professional look it has done the job of functionality.

On the other hand, it will a better choice to use the new interface creator called App designer it is integrated in MATLAB 2016 to recent version, it has more functionality than the simple GUI in terms of aesthetics and the handy code editor that comes with great features.

Bibliography and webography

- [1] C. Solomon and T. Breckon, “Fundamentals of Digital Image Processing A Practical Approach with Examples in Matlab.”
- [2] “What is a Pixel? - Definition from Techopedia.”
<https://www.techopedia.com/definition/24012/pixel> (accessed May 24, 2022).
- [3] “Resolution | Understanding Resolution in Digital Photography.”
<https://www.exposureguide.com/resolution/> (accessed May 24, 2022).
- [4] “Vector vs Raster Files - Graphic Design Services.”
<https://www.seekacreative.co.nz/resources/vector-vs-raster-files> (accessed May 24, 2022).
- [5] “Raster vs Vector Graphics: Ultimate File Type Guide.”
<https://justcreative.com/raster-vs-vector-graphics-ultimate-guide/> (accessed May 24, 2022).
- [6] BENLAHCENE Sarra, “Compression des images médicales Par ondelettes,” 2017. Accessed: May 24, 2022. [Online]. Available: <http://dSPACE.univ-tlemcen.dz/handle/112/12372>
- [7] L. N. Faria, L. M. G. Fonseca, and M. H. M. Costa, “Performance evaluation of data compression systems applied to satellite imagery,” *Journal of Electrical and Computer Engineering*, 2012, doi: 10.1155/2012/471857.
- [8] G. Suseela and Y. Asnath Vicky Phamila, “Energy efficient image coding techniques for low power sensor nodes: A review,” *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2961–2972, Dec. 2018, doi: 10.1016/J.ASEJ.2017.10.004.
- [9] AMRAOUI Merouane, “Etude de codage par blocs des algorithmes hiérarchiques EZW et SPIHT Study of embedded coding with blocks of EZW and SPIHT,” 2016. Accessed: May 24, 2022. [Online]. Available: <http://thesis.univ-biskra.dz/id/eprint/3929>

Bibliography

- [10] “Huffman Coding,” *Encyclopedia of Multimedia*, pp. 278–280, Jun. 2006, doi: 10.1007/0-387-30038-4_92.
- [11] “(PDF) A Review of Image Compression.”
https://www.researchgate.net/publication/257868354_A_Review_of_Image_Compression (accessed May 24, 2022).
- [12] E. Ghaderpour, S. D. Pagiatakis, and Q. K. Hassan, “A survey on change detection and time series analysis with applications,” *Applied Sciences (Switzerland)*, vol. 11, no. 13. MDPI AG, Jul. 01, 2021. doi: 10.3390/app11136141.
- [13] L. Huang, Q. Kemaio, B. Pan, and A. K. Asundi, “Comparison of Fourier transform, windowed Fourier transform, and wavelet transform methods for phase extraction from a single fringe pattern in fringe projection profilometry,” *Optics and Lasers in Engineering*, vol. 48, no. 2, pp. 141–148, Feb. 2010, doi: 10.1016/j.optlaseng.2009.04.003.
- [14] A. ZITOUNI, “Ondelettes et techniques de compression d’images numérique.”
- [15] “Time-Frequency Analysis and Wavelets - Sapien Labs | Neuroscience | Human Brain Diversity Project.” <https://sapienlabs.org/lab-talk/time-frequency-analysis-and-wavelets/> (accessed Jun. 14, 2022).
- [16] “Haar wavelet - Wikiwand.” https://www.wikiwand.com/en/Haar_wavelet (accessed Jun. 14, 2022).
- [17] J. Shen and G. Strang, “Asymptotics of Daubechies Filters, Scaling Functions, and Wavelets,” *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 312–331, 1998, doi: 10.1006/ACHA.1997.0234.
- [18] “Daubechies wavelet - Wikiwand.”
https://www.wikiwand.com/en/Daubechies_wavelet (accessed Jun. 14, 2022).
- [19] “Biorthogonal wavelet - Wikiwand.”
https://www.wikiwand.com/en/Biorthogonal_wavelet (accessed Jun. 14, 2022).
- [20] T. Enamamu, A. Otebolaku, J. Marchang, and J. Dany, “Continuous m-Health Data Authentication Using Wavelet Decomposition for Feature Extraction,” *Sensors 2020, Vol. 20, Page 5690*, vol. 20, no. 19, p. 5690, Oct. 2020, doi: 10.3390/S20195690.

Bibliography

- [21] N. Kumar and R. Kumar, "Wavelet transform-based multipitch estimation in polyphonic music," *Heliyon*, vol. 6, no. 1, p. e03243, Jan. 2020, doi: 10.1016/j.heliyon.2020.e03243.
- [22] "Morlet wavelet - Wikiwand." https://www.wikiwand.com/en/Morlet_wavelet (accessed Jun. 14, 2022).
- [23] C. Torrence and G. Compo, "A Practical Guide to Wavelet Analysis," *Bull Am Meteorol Soc*, vol. 79, no. 1, pp. 61–78, 1998.
- [24] S. A. Broughton, "Wavelet Based Methods in Image Processing," *www.rose-hulman.edu*, Accessed: Jun. 14, 2022. [Online]. Available: <http://www.rose-hulman.edu/~brought/Epubs/Imaging/waveimage.html>
- [25] C. K. Chui, A. K. Chan, and S. J. Liu, "An introduction to wavelets," 1992.
- [26] "Multi-resolution analysis using the wavelet transform [15]. | Download Scientific Diagram." https://www.researchgate.net/figure/Multi-resolution-analysis-using-the-wavelet-transform-15_fig6_233657890 (accessed Jun. 14, 2022).
- [27] "Wavelet Toolbox in MATLAB — Part 1 | by Ashkan Abbasi | Medium." <https://medium.com/@ashkan.abbasi/wavelet-toolbox-in-matlab-part-1-dfcabc68725b> (accessed Jun. 14, 2022).
- [28] "The inverse discrete wavelet transform. Here, $h \text{ í } \mu \text{ í }^{1/4}$ “ ($\text{í } \mu \text{ í } \pm \text{š}$), $h \text{ í } \dots$ | Download Scientific Diagram." https://www.researchgate.net/figure/The-inverse-discrete-wavelet-transform-Here-h-i-i14-i-i-h-i-i14-i-i-h-i-i_fig7_283882847 (accessed Jun. 14, 2022).
- [29] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993, doi: 10.1109/78.258085.
- [30] "Embedded Zerotrees of Wavelet transforms - Wikiwand." https://www.wikiwand.com/en/Embedded_Zerotrees_of_Wavelet_transforms (accessed Jun. 14, 2022).
- [31] R. Boujelbene, L. Boubchir, and Y. ben Jemaa, "Enhanced embedded zerotree wavelet algorithm for lossy image coding," *IET Image Processing*, vol. 13, no. 8, pp. 1364–1374, Jun. 2019, doi: 10.1049/IET-IPR.2018.6052.

Bibliography

- [32] M. REZKALLAH and N. YOUSFI, “Compression d’images par transformée en ondelettes.”
- [33] “A Guide to MATLAB (For Beginners and Experienced Users) || SIMULINK and GUIs | Hunt, Brian R.; Lipsman, Ronald L.; Rosenberg, Jonathan M. | download.” <https://booksc.org/book/63912867/cee5d5> (accessed Jun. 15, 2022).
- [34] B. R. Hunt, R. L. Lipsman, and J. M. Rosenberg, “SIMULINK and GUIs,” *A Guide to MATLAB*, pp. 121–135, Jun. 2012, doi: 10.1017/CBO9781139164801.009.