الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la RechercheScientifique

UNIVERSITE MOHAMED KHIDER BISKRA
Faculté des Sciences Exactes
et des Sciences de la Nature et de la Vie
Département DE MATHEMATIQUES

جامعة محمد خيضر
بسكرة
كلية العلوم الدقيقة وعلوم الطبيعة والحياة
قسم الرياضيات

THÈSE DE DOCTORAT

préparée au sein de

Laboratoire de Mathématiques Appliquées

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Spécialité : Mathématiques Appliquées

Soutenue publiquement à Biskra le 19/12/2016, par :

Amrane HOUAS

Ondelettes et Détection des Singularités pour le Traitement des Images

---

Devant le jury composé de :

Khaled MELKEMI,	Professeur, Université de Batna 2	President
Zouhir MOKHTARI,	Maitre de Conférences, Université de Biskra	Encadreur
Hamid BENSRIDI ,	Professeur, Université de Sétif 1	Examinateur
Mourad DILMI,	Maitre de Conférences, Université de Sétif 1	Examinateur
El Amir DJEFFAL,	Maitre de Conférences, Université de Batna 2	Examinateur
Naceur KHELIL,	Maitre de Conférences, Université de Biskra	Examinateur

DEDICATION

This thesis work is dedicated to my familly: ***parents***, ***brothers and my sister***, *who have always loved me unconditionally and have taught me to work hard for the things that i aspire to achieve.*

This work is also dedicated to my ***whife***, *who has been a constant source of support and encouragement during my research and preparation of thesis, and to my little daughters,* ***Asma Sajeda and Safwa Razane****. I am truly thankful for having you in my life*

I dedicate this work also to all my friends and colleagues of my department and colleagues of the University of Biskra

ACKNOWLEDGMENTS

First of all, *I thank* ***Allah***, *the almighty, for giving me strength and ability to complete this study.*

Second, *i would like to express my deepest sense of gratitude to my supervisor* ***Dr. Zouhir Mokhtari*** , *who offered his continuous advice and encouragement throughout the course of this thesis. I thank him for the systematic guidance and great effort he put into training me in the field, image processing.*

Thirdly, *i would like to express my sincere thanks to the* ***Professor. Khaled Melkemin*** *(University of Batna 1) because he agreed to chair the review committee and spend his time reading and evaluating my thesis, my profound gratitude is also expressed to the committee members,* ***Dr. Naceur Khelil*** *(University of Biskra),* ***Dr. El Amir Djeffal*** *(University of Batna 1),* ***Dr. Mourad Dilmi and Pr. Hamid Benseridi*** *(University of Setif 1), for generously offering their time, support and good will throughout the reading and review of this thesis.*

Finally, *i am greatly indebted and appreciate very much to my* ***wife***, *for her encouragement, support and sacrifices through out my research and preparation of this thesis.*

Contents

List of Figures

List of Tables

Introduction

The important role of images in the modern world is undeniable as they are intimately integrated into our lives. They are often present in our everyday life (video games, magazines, TV, ...), for our personal needs (medical imaging, biological imaging, photographs ...), and also in professional life (office, monitoring remote video conferencing, industrial vision ...).

They are not limited to the various technological areas, but they are useful tools for observation and investigation and frequently lead to major scientific discoveries in various fields of science. The world of processing and image analysis is very broad and multidisciplinary. It means all of the theories, methods, techniques, applications, software ... in connection with the information extracted from (qualitative or quantitative) images to survey, measure, understand, interpret and finally make a decision.

Image processing deals with the transformations of an image or multiple

images to an other group of one or more images ([22], [26], [39], and [43]). For image compression, the main purpose is to reduce the amount of data required to represent the image. This is done either by removing redundancies, or by using a basis which maps the minimum number of coefficients to reconstruct the image. Known methods are the representation of wavelets ([45],[38]) and *DCT* decomposition ([31], [13]). There are two techniques for compressing images: lossy and lossless (with and without loss).

Due to the huge expansion of images and multimedia use in current nowadays applications, the need for fast and secure representation, transmission and storage schemes become more and more crucial, especially because digital images can contain private and confidential information that may be associated with financial, medical or personal interest [34]. Encrypting images is a crucial tool for protecting information during communication in network, through the rapid development of computer network large sized images can be easily transmitted therefore, the encryption operation has become an important issue. The most classical encrypting techniques are well developed for the security of textual data, but these are not suitable with digital media such as images. The main constraint is that, the structure of image is complex compared to the text file, which implies that the size of image is much greater than the size of textual file. In this case the necessity of designing encryp-

tion and decryption algorithms with low complexity is very important. Many researches from different disciplines like mathematics, computer science and electrical engineering have focused for developing robust algorithms for encrypting images in order to offer a higher level of security in telecommunication networks.

Nowadays, different techniques of image encryption have been proposed. This is due to the proliferation of sophisticated sensors. By nature, the Internet by its TCP/IP protocol is a subject to any control, hence its vulnerability to hacker attacks. For this reason, the large number of researches in the field of visual cryptography have been developed. Exchange of secret digital images are frequently used worldwide in a second split on the Internet [42]. Therefore, it becomes very important to protect these information [47].

Cryptographic techniques can be divided into symmetric and asymmetric encryption [4]. As one of the important research topics, image encryption has been more developed. Due to its high processing speed and more degrees of freedom, the added value of image encryption is showed through the recent optical information processing technologies. Different optical techniques have been proposed for image encryption [21, 52].

As known, digital images have important proprieties like, redundancy

of data, less sensitive, correlation between pixels and massive capacity of data. Hence, many of image encryption algorithms have been proposed [18, 51] taking profit from these characteristics. Recently, Guomin Zhou et al. [51] proposeed a fast symmetrical image encryption algorithm based on skew tent map. Based on a new chaos based Line map, their proposed algorithm encrypts images with different size. In order to perturb the correlations between the R, G and B components of the true color image, these three components are encrypted at bit level and operated at the same time [51]. In fact, several classical encryption schemes like data encryption standard (DES) [8], triple data encryption algorithm (TDEA) [7], advanced encryption standard (AES) [6] and Rivest, Shamir and Adleman (RSA) [6, 16] have been developed. However, these algorithms are limited when they are applied in the encryption of digital images, especially for huge images [33].

Similar to DES algorithm but faster than DES, Nithin et al. [36] have proposed the fast image encryption algorithm (FEAL).

Using structurally random matrices and Arnold transform, Rawat et al. [41] have introduced a digital image encryption method based on a fast compressed sensing idea. Zhao et al [51] have recently presented a symmetric digital image encryption algorithm by a new improper fractional-order chaotic subsystem.

A binary image (bi-valued image) is the type of simple image that is

widely used in various electronic applications such as fingerprint analysis, robot vision, motion detection and character recognition. It often appears as cartoons in newspapers and magazines. Moreover, binary images frequently emerge as the result of many automatic tasks, such as binarisation, halftoning, edge detection, segmentation, and thresholding. Certain input/output devices and sensors, like for examples laser printers, fax machines, biometric devices, and bi-tone machine screens, can only handle bi-level images.

Due to their simplicity compared with gray level images; it is better to process binary images in real time. In the context of binary image encryption, many schemes have been proposed. Among the most published works, we can find in [10] a scan language is proposed by Bourbaki in 1986 as a language for efficient accessing of a two dimensional array. In [12] a parallel implementation version for the scan language is presented, which shows that the parallel expansion scheme is faster and requires less storage space. Bourbaki and Alexopoulos in [11] proposed a new encryption scheme for binary images using scan pattern. This algorithm is based on a family of 2D transposition which is produced by the scan language. In [14] Chung and Chang developed an encryption scheme for binary images with higher security, this approach sets the different scan patterns at the same level in the scan tree structure and uses the two dimensional run-encoding technique in order to ensure a higher security and a good compression ratio. In [27] a very simple

method for binary image encryption is reported based on inference of two phase-only masks, the main idea of this algorithm is that: the binary image is first modulated by a random phase mask and then separated into two phase-only masks. This approach offers a very low complexity and without any time consuming iterative computations. Most of the aforementioned algorithms, they have proved their effectiveness in the area of cryptography.

This work is organized as follows.

Chapter 1 resume the fundamentals concepts of image processing, where we given; definitions, types and properties of digital image, then we describes techniques of lossy and losseless image compression, after we resume the necessary mathematical background of image processing.

In chapter 2, we propose a new method of lossy compression made up of two steps. The first one is the transformation of a representation basis of d images to the one proposed by Melkemi and Mokhtari [35], with the aim to have a lot of redundancies. The second step is the application of the DCT transformation on the image obtained. When we get the different values of DCT coefficients, we perform a thresholding on these coefficients, depending on the probability of each coefficient, where we eliminate the coefficients that have low probabilities. We calculate the variation of the $PSNR$ of the reconstructed image, based on the percentage of non-zero coefficients obtained after thresholding, we plot the corresponding curve, analyze the results and finally compare the results

obtained with those achieved when we do the same steps, but replacing the DCT by the Haar DWT of the first and the second level.

In the last chapter, we propose an efficient encryption algorithm for binary images [24] which is based on dividing the original image into d blocks, then constructing new images of the same size as the original one and representing them in a new proposed basis.

We call key-image the matrix of parameters obtained using this transformation, and we call the encrypted images the represented images in this new basis.

In the proposed decryption algorithm, a subtraction between each encrypted image and the key-image is applied, then we sum them in an image to get the original one.

Moreover, in the same way, we use this new basis to encrypt a database of binary images. In fact, the idea of this new basis construction is inspired from the paper of Mokhtari and Melkemi [35].

Chapter 1

Fundamentals Concepts

1.1 Digital Image Fundamentals

1.1.1 Definitions

Digital images are made of picture elements called pixels. Typically, pixels are organized in an ordered rectangular array. The size of an image is determined by the dimensions of this pixel array. The image width is the number of columns, and the image height is the number of rows in the array. Thus the pixel array is a matrix of M columns $\times$ N rows. To refer to a specific pixel within the image matrix, we define its coordinate at x and y. The coordinate system of image matrices defines x as increasing from left to right and y as increasing from top to bottom. [3]

4*Having defined the number of pixels, $M \times N$, only provides a rectangular shape for our image. One more parameter, intensity, is needed to truly define an image. Each pixel has its own intensity value, or brightness. If all the pixels have the same value, the image will be a uniform shade; all black, white, gray, or some other shade. It is in the type of intensity used for each pixel that image types vary. Black and white images only have intensity from the darkest gray (black) to lightest gray (white). Color images, on the other hand, have intensity from the darkest and lightest of three different colors, Red, Green, and Blue. The various mixtures of these color intensities produces a color image. Thus the two most basic types of digital images, BW and Color, are known as gray-scale and RGB images. In addition to the intensity type of each pixel, the range of intensity values also varies. [3]

Intensity values in digital images are defined by bits. A bit is binary and only has two possible values, 0 or 1. An 8-bit intensity range has 256 possible values, 0 to 255. This can be seen mathematically by $2^{(\text{of bits})}$. For a 1-bit, or binary, image, $2^1 = 2$ possible values and for an 8-bit image, $2^8 = 256$ possible values. The standard digital photo uses an 8-bit range of values; RGB images use 8-bit intensity ranges for each color and BW images have a single 8-bit intensity range. Since RGB images contain $3 \times$ 8-bit intensities they are also referred to as 24-bit color images. [3]

Figure 1.1: 256 gray levels

1.1.2 Types of Digital Images

For photographic purposes, there are two important types of digital images : color and black and white. Color images are made up of colored pixels while black and white images are made of pixels in different shades of gray [44].

Black an White Images

A black and white image is made up of pixels each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays . Since the eye can barely distinguish about 200 different gray levels, this is enough to give the illusion of a stepless tonal scale as illustrated below in Figure (1.1):

Color Images

A color image is made up of pixels each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue (sometimes referred to as RGB) are the primary colors for mixing lightâthese so called additive primary colors are different from the subtractive primary colors used for mixing paints (cyan, magenta, and yellow). Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors.

Note that for images of the same size, a black and white version will use three times less memory than a color version.

Binary or Bilevel Images

Binary images use only a single bit to represent each pixel. Since a bit can only exist in two states on or off, every pixel in a binary image must be one of two colors, usually black or white. This inability to represent intermediate shades of gray is what limits their usefulness in dealing with photographic images.

1.1.3 Resolution

The more points at which we sample the image by measuring its color, the more detail we can capture. The density of pixels in an image is referred to as its resolution. The higher the resolution, the more information the image contains. If we keep the image size the same and increase the resolution, the image gets sharper and more detailed. Alternatively, with a higher resolution image, we can produce a larger image with the same amount of detail.

For example, the following images illustrate what happens as we reduce the resolution of an image while keeping its size the same the pixels get larger and larger and there is less and less detail in the image in Figure (1.2):

1.2 Mathematical background

1.2.1 The Discrete Cosine Transform (DCT)

Transform coding constitutes an integral component of contemporary image/video processing applications. Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels. Similarly in a video transmission system, adjacent pixels in consecutive frames show very high correlation.

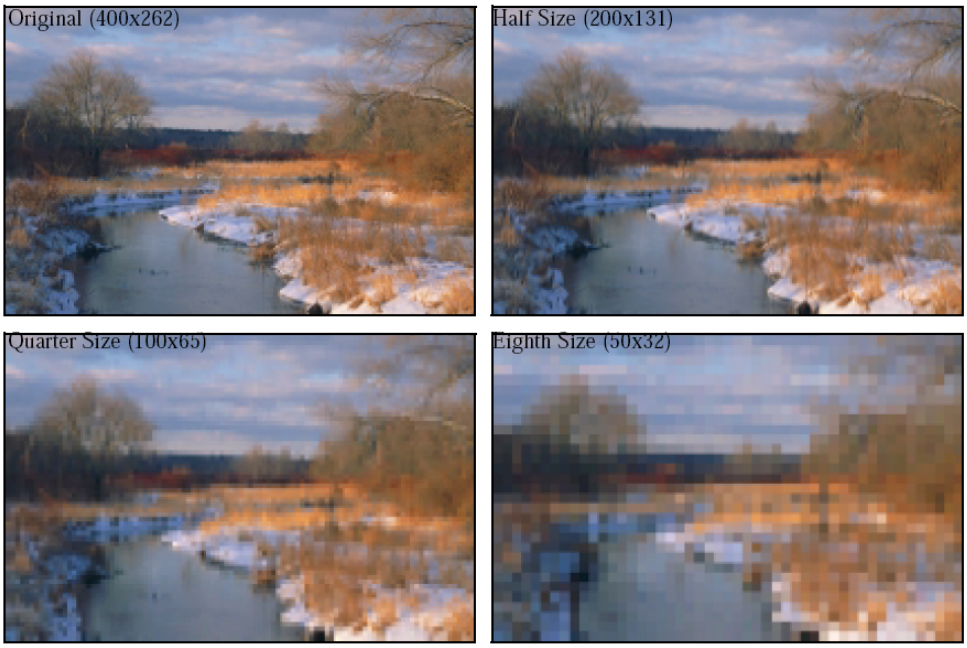


Figure 1.2: The resolution of an image

Consequently, these correlations can be exploited to predict the value of a pixel from its respective neighbors. A transformation is, therefore, defined to map this spatial (correlated) data into transformed (uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the information content of an individual pixel is relatively small i.e., to a large extent visual contribution of a pixel can be predicted using its neighbors. [28]

The Discrete Cosine Transform (*DCT*) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency.

The One-Dimensional *DCT*

The most common *DCT* definition of a $1-D$ sequence of length N is

$$C(u) = \alpha(u) \sum_{x=0,N-1} f(x) cos(\frac{\pi(2x+1)u}{2N}) \tag{1.1}$$

for $u = 0, 1, 2, ..., N-1$. Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0,N-1} \alpha(u) C(u) cos(\frac{\pi(2x+1)u}{2N}) \tag{1.2}$$

for $x = 0, 1, 2, ..., N-1$. In both equations 1.1 and 1.1 $\alpha(u)$ is defined as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for} \quad u = 0 \\ \sqrt{\frac{2}{N}} & \text{for} \quad u \neq 0 \end{cases} \tag{1.3}$$

It is clear from 1.1 that for $u = 0$ $C(0) = \sqrt{\frac{1}{N}} \sum_{x=0,N-1} f(x)$. Thus, the first transform coefficient is the average value of the sample sequence. In literature, this value is referred to as the *DC Coefficient*. All other transform coefficients are called the *AC Coefficient*.

The Two-Dimensional *DCT*

The study of the efficacy of *DCT* on images, necessitates the extension of ideas presented in the last section to a two-dimensional space. The $2-DDCT$ is a direct extension of the $1-D$ case and is given by

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0,N-1} \sum_{y=0,N-1} f(x,y) cos(\frac{\pi(2x+1)u}{2N}) cos(\frac{\pi(2y+1)v}{2N}) \tag{1.4}$$

for $u, v = 0, 1, 2, ..., N-1$, $\alpha(u)$ and $\alpha(u)$ are defined in 1.3. The inverse transformation is defined as

$$f(xny) = \sum_{u=0,N-1} \sum_{u=0,N-1} \alpha(u)\alpha(v) C(u,v) cos(\frac{\pi(2x+1)u}{2N}) cos(\frac{\pi(2y+1)v}{2N}) \tag{1.5}$$

for $x, y = 0, 1, 2, ..., N-1$.

Properties of *DCT*

In this outlineswe present some properties of the *DCT* which are of particular value to image processing applications.

1. **Decorrelation.** The principle advantage of image transformation is the removal of redundancy between neighboring pixels. This leads to uncorrelated transform coefficients which can be encoded independently.

2. **Energy Compaction.** Efficacy of a transformation scheme can be directly gauged by its ability to pack input data into as few coefficients as possible. This allows the quantizer to discard coefficients with relatively small amplitudes without introducing visual distortion in the reconstructed image. *DCT* exhibits excellent energy compaction for highly correlated images.

   Examples of the energy compaction property of *DCT* with respect to some standard images are provided in Figure 1.3 below. Figure refig:ec above reveals that it comprises of four broad image classes. (a) and (b) contain large areas of slowly varying intensities. These images can be classified as low frequency images with low spatial details. A *DCT* operation on these images provides very good energy compaction in the low frequency region of the transformed image. (c) contains a number of edges (i.e., sharp intensity variations) and therefore can be classified as a high frequency image

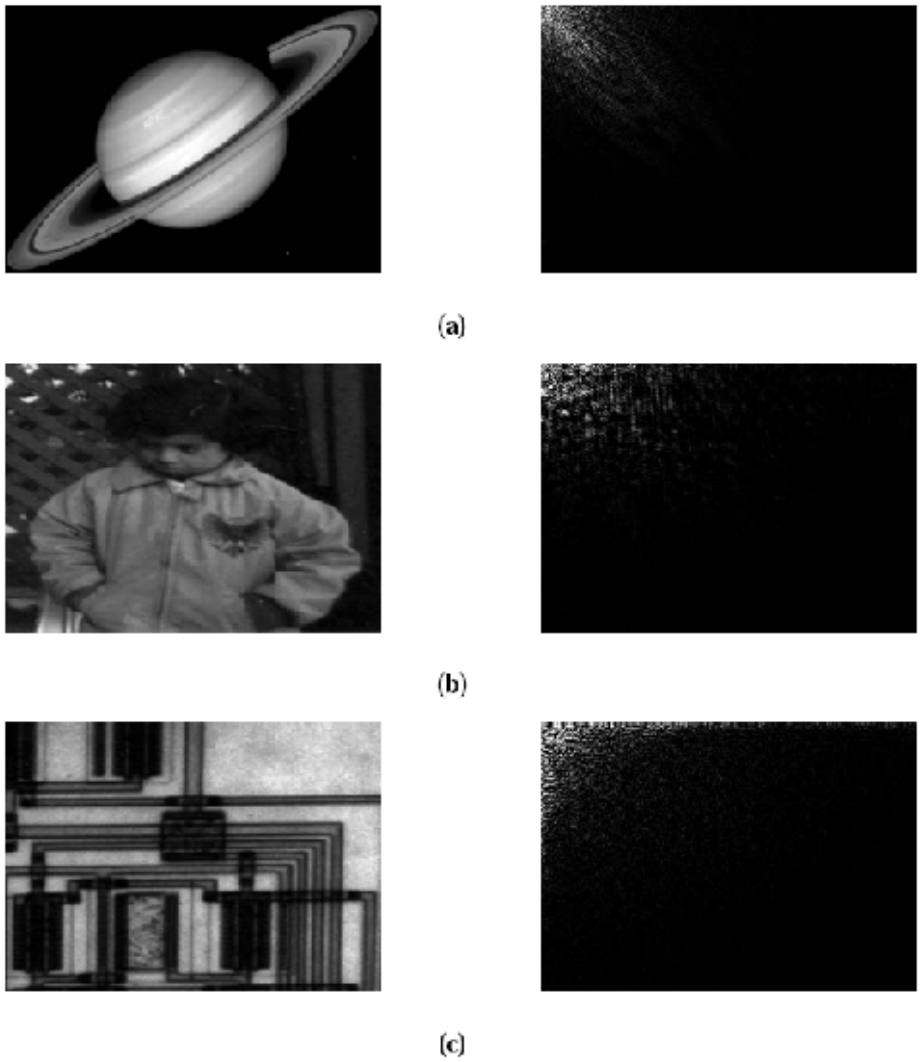


Figure 1.3: Energy compaction property of *DCT*

with low spatial content. However, the image data exhibits high correlation which is exploited by the *DCT* algorithm to provide good energy compaction.

DCT renders excellent energy compaction for correlated images. Studies have shown that the energy compaction performance of *DCT* approaches optimality as image correlation approaches one i.e., *DCT* provides (almost) optimal decorrelation for such images [15].

3. **Separability.** The DCT transform equation 1.4 can be expressed as,

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1} cos(\frac{\pi(2x+1)u}{2N})\sum_{y=0}^{N-1} f(x,y)cos(\frac{\pi(2y+1)v}{2N}) \tag{1.6}$$

for $u, v = 0, 1, 2, , N1$. This property, known as separability, has the principle advantage that $C(u,v)$ can be computed in two steps by successive $1-D$ operations on rows and columns of an image. The arguments presented can be identically applied for the inverse *DCT* computation 1.5.

4. **Symmetry.** Another look at the row and column operations in Equation 1.6 reveals that these operations are functionally identical. Such a transformation is called a symmetric transformation. A separable and symmetric transform can be expressed in the form

[40]

$$T = AfA \tag{1.7}$$

where A is an N symmetric transformation matrix with entries $a(i,j)$ given by

$$a(i,j) = \alpha(j) \sum_{j=0,N-1} cos(\frac{\pi(2j+1)i}{2N})$$

and f is the N image matrix.

1.2.2 The continuous/discrete Wavelet transform

The continuous Wavelet transform CWT

Given ψ in $L^2(\mathbb{R})$. Introduce a family of functions $\psi_{(a,b)}$ where $a > 0$ and $b \in \mathbb{R}$ as follows [48]

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}}\psi(\frac{(t-b)}{a})$$

$t \in \mathbb{R}$ and $\|\psi_{(a,b)}\| = \|\psi\|$

The continuous wavelet transform $F(a,b)$ of a function f is defined by

$$F(a,b) = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\psi(\frac{(t-b)}{a})dt$$

$$\langle f, \psi_{a,b} \rangle = \frac{1}{2\pi} \langle \hat{f}, \hat{\psi}_{a,b} \rangle$$

where

$$\hat{\psi}_{a,b}(\omega) = \sqrt{a}e^{-i\omega b}\hat{\psi}(a\omega)$$

The inverse wavelet transform

$$f(t) = C_{\psi}^{-1} \int_{-\infty}^{+\infty} \int_{0}^{+\infty} \frac{1}{a^2} F(a,b)\, \psi_{a,b}(t)\, dadb$$

$$C_{\psi} = \int_{0}^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega$$

Needed $\hat{\psi}(0) = 0$, i.e.,

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0$$

This is the reason why the functions $\psi_{a,b}$ are called wavelets. ψ is called the Motherwavelet.

The discrete Wavelet transform

DWT A huge amount of data are represented by a finite number of values, so it is important to consider a discrete version of the *CWT*. Generally, the orthogonal(discrete) wavelets are employed because this method associates the wavelets to orthonormal bases of $L^2(\mathbb{R})$. In this case, the wavelet transform is performed only on a discrete grid of the parameters of dilation and translation, i.e., a and b take only integral

values. Within this framework, an arbitrary function $f(t)$ of finite energy can be written using an orthonormal wavelet basis:

$$f(t) = \sum_m \sum_n d_n^m \psi_n^m(t) \tag{1.8}$$

where the coefficients of the expansion are given by

$$d_n^m = \int_{-\infty}^{+\infty} f(t)\psi_n^m(t)dt \tag{1.9}$$

The orthonormal basis functions are all dilations and translations of a function referred as the analyzing wavelet $\psi(t)$, and they can be expressed in the form

$$\psi_n^m(t) = 2^{\frac{m}{2}}\psi(2^m t - n) \tag{1.10}$$

with m and n denoting the dilation and translation indices, respectively. The contribution of the function at a particular wavelet level m is given by

$$d_m(t) = \sum_n d_n^m \psi_n^m(t) \tag{1.11}$$

which provides information on the time behavior of the function within different scale bands. Additionally, it provides knowledge of their contribution to the total function energy.

In this context, Mallat (1999) [32] developed a computationally efficient method to calculate 1.8 and 1.8. This method is known as multiresolu-

tion analysis (MRA). The MRA approach provides a general method for constructing orthogonal wavelet basis and leads to the implementation of the fast wavelet transform (FWT). This algorithm connects, in an elegant way, wavelets and filter banks. A multiresolution function decomposition of a function f is based on successive decomposition into a series of **approximations** and **details**, which become increasingly coarse. Associated with the wavelet function $\psi(t)$ is a corresponding scaling function, $\phi(t)$, and scaling coefficients, a_n^m (Mallat, 1999 [32]). The scaling and wavelet coefficients at scale m can be computed from the scaling coefficients at the next finer scale $m+1$ using

$$a_n^m = \sum_l h[l-2n] a_l^{m+1}, \tag{1.12}$$

$$d_n^m = \sum_l g[l-2n] a_l^{m+1}, \tag{1.13}$$

where $h[n]$ and $g[n]$ are typically called lowpass and highpass filters in the associated filter bank. Equations 1.12 and 1.13 represent the fast wavelet transform (FWT) for computing 1.9.

In fact, a_n^m and d_n^m are the convolutions of a_n^{m+1} with the filters $h[n]$ and $g[n]$ followed by a downsampling of factor 2 [32].

Conversely, a reconstruction of the original scaling coefficients a_n^{m+1} can

be made from

$$a_n^{m+1} = \sum_l (h[l-2n]a_l^m + g[l-2n]d_l^m \tag{1.14}$$

a combination of the scaling and wavelet coefficients at a coarse scale. Equation 1.14 represents the inverse of FWT for computing 1.8, and it corresponds to the synthesis filter bank. This part can be viewed as the discrete convolutions between the upsampled a_l^m and the filters $h[n]$ and $g[n]$, that is, following an âupsamplingâ of factor 2 one calculates the convolutions between the upsampled function and the filters $h[n]$ and $g[n]$. The number of levels in the multiresolution algorithm depends on the length of the signal. A signal with 2^k values can be decomposed into $k+1$ levels.

To initialize the FWT, one considers a discrete time function $f = f[1], f[2], ..., f[N]$ of length $N = 2^M$. The first application of 1.12 and 1.13, beginning with $a_n^{m+1} = f[N]$, defines the first level of the FWT of f. The process goes on, always adopting the $m+1$ scaling coefficients to calculate the $m+1$ scaling and wavelet coefficients. Iterating 1.12 and 1.13 M times, the transformed function consists of M sets of wavelet coefficients at scales $m = 1, ..., M$, and a function set of scaling coefficients at scale M. There are exactly $2^{(km)}$ wavelet coefficients d_n^m at each scale m, and $2^{(kM)}$ scaling coefficients a_n^M. The maximum number of iterations M_{max} is k. This property of the MRA is generally the key

factor to identify crucial information in the respective frequency bands. A three-level decomposition process of the FWT is shown in Figure 1.4 below

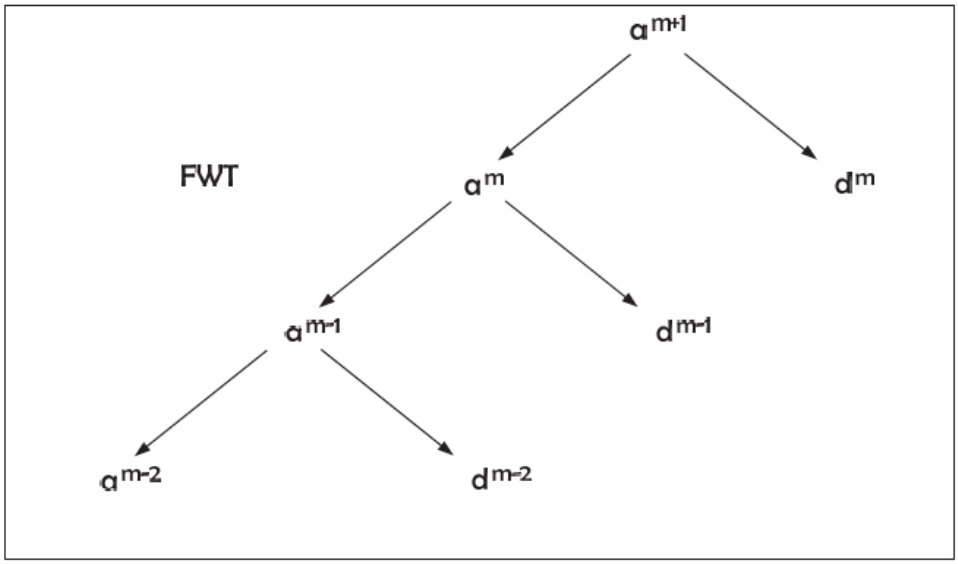


Figure 1.4: The structure of a three-level fast wavelet transform.

Examples

The Haar wavelet

The Haar wavelet's mother wavelet function $\psi(t)$ can be described as [29]

$$\psi(t) = \begin{cases} 1 & \text{for} \quad 0 \leq t < \frac{1}{2} \\ -1 & \text{for} \quad \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.15}$$

Its scaling function $\phi(t)$ can be described as

$$\phi(t) = \begin{cases} 1 & \text{for} \quad 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.16}$$

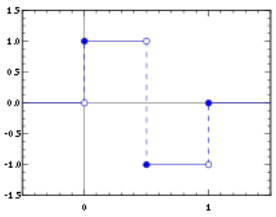


Figure 1.5: The Haar wavelet.

Haar Matrix

The 2×2 Haar matrix that is associated with the Haar wavelet is

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1.17}$$

If one has a sequence of length a multiple of four, one can build blocks of 4 elements and transform them in a similar manner with the 4×4

Haar matrix

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \tag{1.18}$$

which combines two stages of the fast Haar-wavelet transform.

Generally, the $2N \times 2N$ Haar matrix can be derived by the following equation.

$$H_{2N} = \begin{bmatrix} H_N \bigotimes [1, 1] \\ I_N \bigotimes [1, -1] \end{bmatrix} \tag{1.19}$$

where $I_N = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$ and $\bigotimes$ is the Kronecker product.

The Kronecker product of A where A is an m matrix and B is a p matrix, is expressed as

$$A = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \tag{1.20}$$

The Haar transform

The Haar transform is derived from the Haar matrix. An example of a

4×4 Haar transformation matrix is shown below.

$$H_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \tag{1.21}$$

The Haar transform y_n of an n-input function x_n is $y_n = H_n x_n$.

The Haar transform matrix is real and orthogonal. Thus, the inverse Haar transform can be derived by the following equations $H = H^*$, $H^{-1} = H^T$ i.e $HH^T = I$. Thus, the inverse Haar transform is $x_n = H_n^T y_n$.

The Meyer wavelet

Yves Meyer constructed a smooth othornormal wavelet basis as follows. First all, define the Fourier transform $\Phi(\omega)$ of a scaling function $\phi(t)$ as [29]

$$\Phi(\omega) = \begin{cases} 1 & \text{if} \quad |\omega| \leq \frac{2}{3}\pi \\ \cos[\frac{\pi}{2} v(\frac{3}{4\pi}|\omega| - 1)] & \text{if} \quad \frac{2}{3}\pi \leq |\omega| \leq \frac{4}{3} \\ 0 & \text{oterwise} \end{cases} \tag{1.22}$$

where v is a smooth function satisfying the following conditions:

$$v(t) = \begin{cases} 0 & \text{if} \quad t \leq 0 \\ 1 & \text{if} \quad t \geq 1 \end{cases} \tag{1.23}$$

with the additional property

$$v(t) + v(1-t) = 1 \tag{1.24}$$

In this case, the wavelet function ψ can be found easily from Φ. First, we find the fourier transform of ψ :

$$\begin{aligned} \Psi(\omega) &= \exp(\tfrac{i\omega}{2}) \textstyle\sum_{l\in\mathbb{Z}} \Phi(\omega + 2\pi(2l+1))\Phi(\tfrac{\omega}{2}) \\ &= \exp(\tfrac{i\omega}{2})[\Phi(\omega+2\pi) + \Phi(\omega-2\pi)]\Phi(\tfrac{\omega}{2}) \end{aligned} \tag{1.25}$$

ψ can be obtained by taking the inverse Fourier transform.

1.2.3 Peak Signal to Noise Ratio ($PSNR$)

The $PSNR$ is used to assess the quality of the recovered image.

Definition 1. *Mean Square Error (MSE) is defined as:*

$$MSE = \frac{1}{MN} \sum\sum (F(x,y) - F'(x,y))^2 \tag{1.26}$$

$F(x,y)$ and $F'(x,y)$ represent the pixel values of $M \times N$ original and reconstructed image.

Definition 2. *Peak Signal to Noise Ratio ($PSNR$) is defined as:*

$$PSNR = 10\log_{10}(255^2/MSE) \tag{1.27}$$

for color image we use following definition of MSE and $PSNR$ [17]

Definition 3. *Peak Signal to Noise Ratio ($PSNR$) is defined as:*

$$PSNR = 10\log_{10}((255^2\times 3)/(MSE(R)+MSE(G)+MSE(B))) \tag{1.28}$$

$$MSE = \frac{1}{MN}\sum\sum(F(x,y) - F'(x,y))^2 \tag{1.29}$$

$F(x,y)$ and $F'(x,y)$ represent intensities values of $M \times N$ original and reconstructed image belonging to R, G and B planes.

Compression Ratio

Definition 4. *Data compression ratio is defined as the ratio between the uncompressed size and compressed size*

$$Compression\ Ratio = \frac{Uncompressed\ Size}{Compressed\ Size} \tag{1.30}$$

1.2.4 Diffuse representation of an image

For a data-set of images, there exist a transform which give a diffuse representation of the image in some specific basis, it is based on the transformation of the representation basis of images given by Melkemi and Mokhatri in [35]; this transformation gives a representation that diffuses the images in the new basis.

This new transformation diffuses the images of a given data-set in a new basis, in order to share information quantity almost equal in different images of our data-set.

In [35], the authors have shown that if we have a database $\{I_k\}_{k=1,d}$ of d images, which are represented in the orthonormal basis $\{e_j\}_{j=1,n}$, such that for any value of k,

$$I_k = \sum_{j=1,n} a_{kj} e_j \tag{1.31}$$

There exists a new base $\{f_j\}_{j=1,n}$ where for all k:

$$I_k = \sum_{j=1,n} b_{kj} f_j \tag{1.32}$$

With

$$b_{kj} = \beta_j - a_{kj} \tag{1.33}$$

and

$$b_{kj} \sim \frac{\| I_k \|_1}{\sqrt{n}} \tag{1.34}$$

By choosing a suitable function and applying the method of least squares, they will get the optimal settings $j = 1, n$:

$$\beta_j^* = \frac{1}{d} \sum_{k=1,d} (a_{kj} + \frac{\| I_k \|_1}{\sqrt{n}}) \tag{1.35}$$

Chapter 2

A Lossy Compression Algorithm for Data Basis Images

In this chapter we present a new method to compress a data basis of images, in the first stage we present this data basis of images in new base given in 1.2.4 in order to reduce the amount of data required to represent the image. In the second stage we apply the DCT to the new representation and do a thresholding with different levels. Experimental results introduced at the end of the chapter, demonstrates the robustness of the proposed strategy in comparison with classical tools.

2.1 Image Compression

Image data compression is concerned with minimization of the number of information carrying units used to represent an image. Image compression schemes can be divided into two broad classes: lossless compression schemes and lossy compression schemes. Lossless compression techniques, as their name implies aim at exact reconstruction and involve no loss of information. Lossy compression techniques accept some loss of information, therefore images compressed using a lossy technique cannot be reconstructed exactly. The distortion in the image caused by lossy compression may be imperceptible to humans and we obtain much higher compression ratios than is possible with lossless compression [9].

2.1.1 Lossless Image Compression Techniques

Lossless data compression is a class of data compression algorithm that allows the exact original data to be reconstructed from the compressed data. Lossless data compression is used in many applications. For example, it is used in the ZIP file format [30].

Lossless compression is used in cases where it is important that the original and the decompressed data be identical, or where deviations from the original data could be deleterious. Typical examples are executable programs, text documents, and source code. Some image file

formats, like PNG or GIF, use only lossless compression, while others like TIFF and MNG may use either lossless or lossy methods [25].

Runlength encoding

Run-length encoding (RLE) is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could greatly increase the file size [25]. Run-length encoding performs lossless data compression and is well suited to palette-based iconic images. It does not work well at all on continuous-tone images such as photographs, although JPEG uses it quite effectively on the coefficients that remain after transforming and quantizing image blocks. The Run length code for a grayscale image is represented by a sequence Vi , Ri where Vi is the intensity of pixel and Ri refers to the number of consecutive pixels with the intensity Vi as shown in the Figure 2.1.

Huffman encoding

Huffman encoding, an algorithm for the lossless compression of files based on the frequency of occurrence of a symbol in the file that is

32	32	32	40	40	40	40	56	56	56

$\{32, 3\}$	$\{40, 4\}$	$\{56, 3\}$

Figure 2.1: Example of RLE encoding

being compressed. The Huffman algorithm is based on statistical coding, which means that the probability of a symbol has a direct bearing on the length of its representation. The more probable the occurrence of a symbol is, the shorter will be its bit-size representation [2]. In any file, certain characters are used more than others. Using binary representation, the number of bits required to represent each character depends upon the number of characters that have to be represented. Using one bit we can represent two characters, i.e., 0 represents the first character and 1 represents the second character. Using two bits we can represent four characters, and so on. Unlike ASCII code, which is a fixed-length code using seven bits per character, Huffman compression is a variable-length coding system that assigns smaller codes for more frequently used characters and larger codes for less frequently used characters in order to reduce the size of files being compressed and transferred [2]. For example, in a file with the following data:

$$XXXXXXYYYYZZ$$

the frequency of "X" is 6, the frequency of "Y" is 4, and the frequency of "Z" is 2. If each character is represented using a fixed-length code of two bits, then the number of bits required to store this file would be 24, i.e.,

$$(2 \times 6) + (2 \times 4) + (2 \times 2) = 24.$$

If the above data were compressed using Huffman compression, the more frequently occurring numbers would be represented by smaller bits, such as:

$$\begin{array}{lllll} X & \text{by the code} & 0 & (1 & \text{bits}) \\ Y & \text{by the code} & 10 & (2 & \text{bits}) \\ Z & \text{by the code} & 11 & (2 & \text{bits}) \end{array}$$

therefore the size of the file becomes 18, i.e.,

$$(1 \times 6) + (2 \times 4) + (2 \times 2) = 18.$$

In the above example, more frequently occurring characters are assigned smaller codes, resulting in a smaller number of bits in the final compressed file [25].

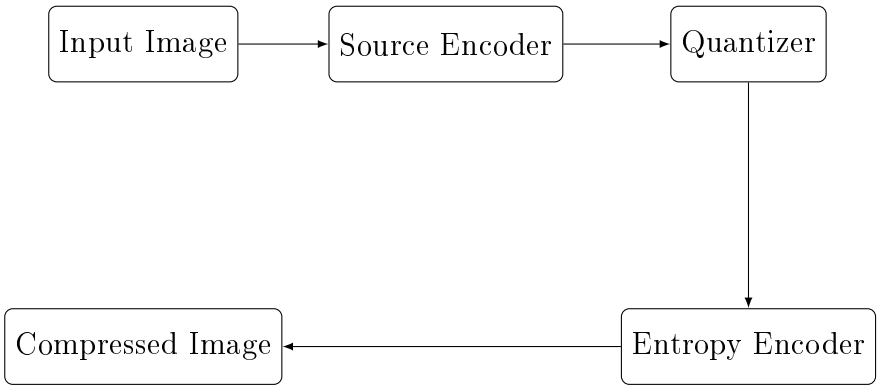


Figure 2.2: Lossy compression scheme

2.1.2 Lossy Image Compression Techniques

Lossy compression as the name implies leads to loss of some information. The compressed image is similar to the original uncompressed image but not just like the previous as in the process of compression some information concerning the image has been lost [37]. They are typically suited to images. The most common example of lossy compression is JPEG. An algorithm that restores the presentation to be the same as the original image are known as lossy techniques. Reconstruction of the image is an approximation of the original image, therefore the need of measuring of the quality of the image for lossy compression technique [37]. Lossy compression technique provides higher compression ratio compare to lossless compression.

Lossy compression scheme is shown in Figure 2.2

Transform Coding

Transform Coding algorithm usually starts by partitioning the original image into small blocks of smaller size. Then for each block related transform coefficients are obtained based on their transform, DCT and wavelet are the example of the transform coding. The resulting coefficients are then computed by quantization techniques and then the output of the quantizer is used for symbol encoding technique to produce the output. At the decoder the reverse process is obtained and image is reconstructed [46].

Block Truncation Coding

In this the image is divided into non overlapping blocks of pixels. Then the quantizer is used to find mean of the pixel values of the all the non overlapping blocks. After that thresholding is done so that the image pixels above the threshold values are set to zero or one. Then for each segment in the bitmap the related reconstruction value is obtained. Larger block size gives greater compression ratio but it reduces the quality of an image [46].

2.2 The proposed method

This section presents a class of methods to compress images. It is based on Diffuse representation of an image given in 1.2.4

2.2.1 Method based on DCT

We apply the algorithm below:

Algorithm 1 The proposed lossy compression code

1: *Input the data basis of d images;*
2: *Represent data basis of d images in new basis (compute* β_j *and* b_j*);*
3: *Apply DCT transform for each image of data basis;*
4: *Compute the probability of DCT coefficients ;*
5: *Tresholding;*
6: *Apply IDCT;*
7: *Return to the canonical basis;*
8: *Compressed Image;*

2.2.2 Method Based on DWT

We apply the same pattern, but replacing the DCT transform by the one corresponding to the Haar DWT of the first and second level.

2.3 Digital Results

We take 8 grayscale images (8 images RGB) of size $512 * 512$, in two different formats png and tiff. The selected images are:

Test, Man, Airplane, Fingerprint, Mandrill, lena, Peppers and Boat for png format (Women, Airplane, House, Lacke, Mandrill, Lena, Peppers, Milk for tiff format).

For images: Mandrill, Lena and Peppers, we plot the $PSNR$ according to the percentage of non-zero DCT coefficients for the method based on DCT and depending on the percentage of non-zero coefficients of the DWT for the one based on Haar DWT of the first and second level. Through this curves (figures (2.4,...,2.9)), we can see what process allows us to reconstruct images with only a small percentage We note that the $PSNR$ values for the images: Mandrill, Lena and Peppers, reconstructed by the method based on DCT in the new basis, are greater than those obtained by the method based on DWT for NNZ values (percentage of non-zero coefficients) less than 10.

So,we choose 4 NNZ values less than 10 for each method, we calculate the $PSNR$ associated with these values and we show the reconstructed images. We obtain the results shown in Figures (2.10, ..., 2.15). Numerical results are resumed in (table 2.1 , table 2.2).

type png

Type rgb

Figure 2.3: Images of png and tiff type

Image	Mandrill		Lena		Peppers	
	NNZ	$PSNR$	NNZ	$PSNR$	NNZ	$PSNR$
DCT	8.75%	43.31	6.17	30.33	8.78	53.8
	4.97%	39.47	3.12	32.02	4.96	51.34
	1.69%	36.73	2.50	28.90	1.86	42.99
	0.89%	30.89	0.71	31.67	0.88	36.66
Haar 2 niv	8.69%	33.01	6.17	5.83	8.78	50.03
	4.97%	7.27	3.12	11.17	4.96	9.21
	1.69%	14.27	2.51	13.59	1.87	19.82
	0.89%	18.36	0.71	19.89	0.88	26.71
Haar 1 niv	8.79%	13.08	6.21	17.87	8.80	17.87
	4.98%	16.90	3.13	22.70	4.94	23.69
	1.69%	23.99	2.55	23.83	2.55	23.83
	0.91%	27.75	0.0.71	20.13	0.71	30.13

Table 2.1: Numerical Results for Images of png type

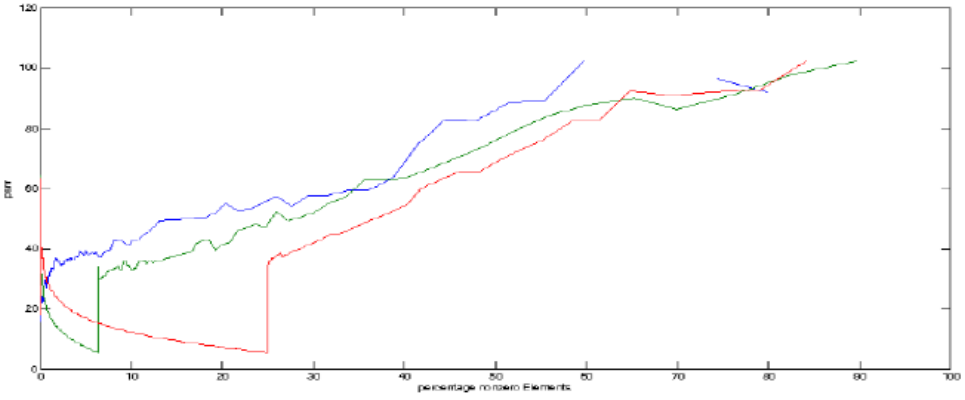


Figure 2.4: $PSNR$ curve of Mandrill.png, for DCT (blue), HAAR 2nd level (green) and HAAR 1st level (red).

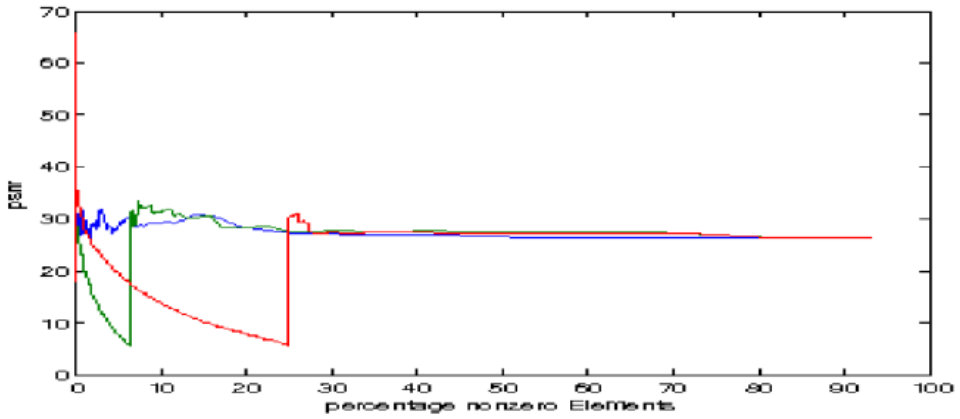


Figure 2.5: $PSNR$ curve of Lena.png, for DCT (blue), HAAR 2nd level (green) and HAAR 1st level (red).

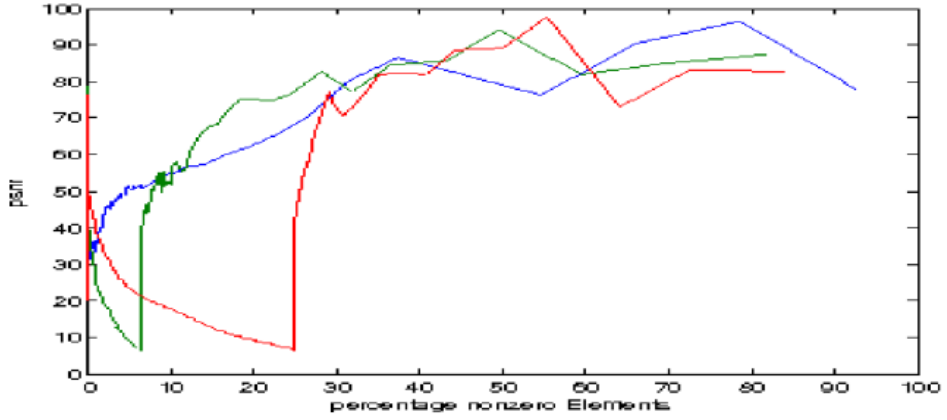


Figure 2.6: $PSNR$ curve of Peppers.png, for DCT (blue), HAAR 2nd level (green) and HAAR 1st level (red).

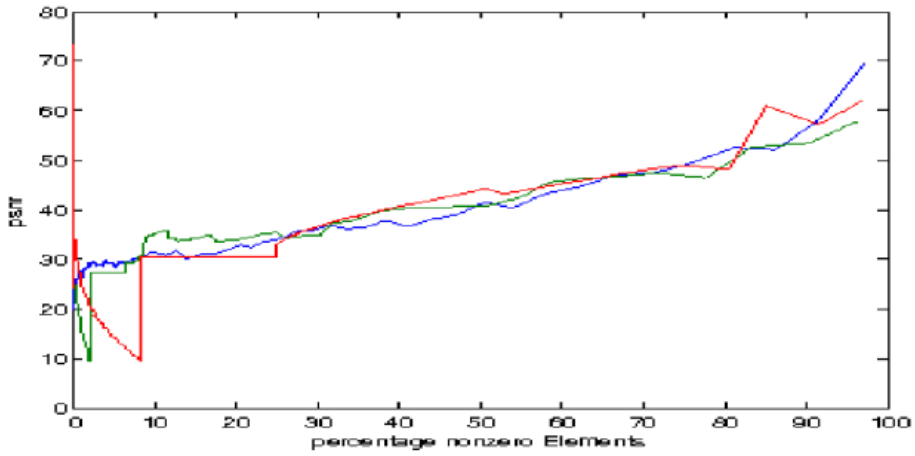


Figure 2.7: $PSNR$ curve of Mandrill.tiff, for DCT (blue), HAAR 2nd level (green) and HAAR 1st level (red).

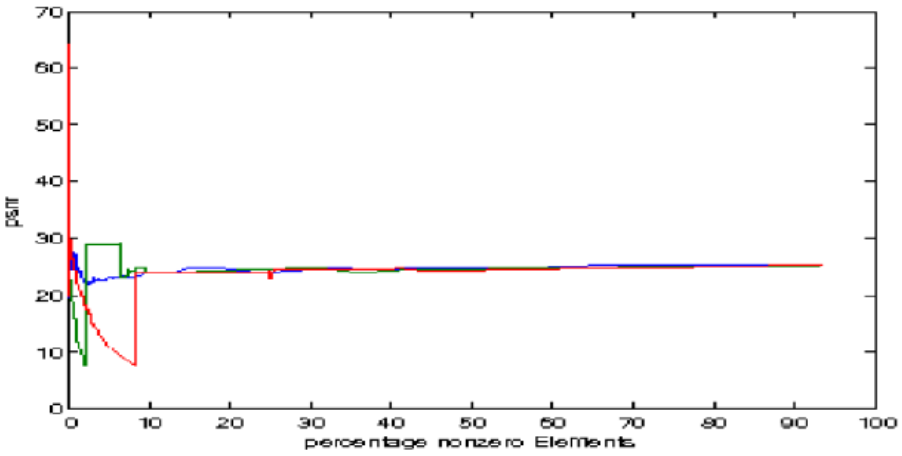


Figure 2.8: $PSNR$ curve of Lena.tiff, for DCT (blue), HAAR 2nd level (green) and HAAR 1st level (red).

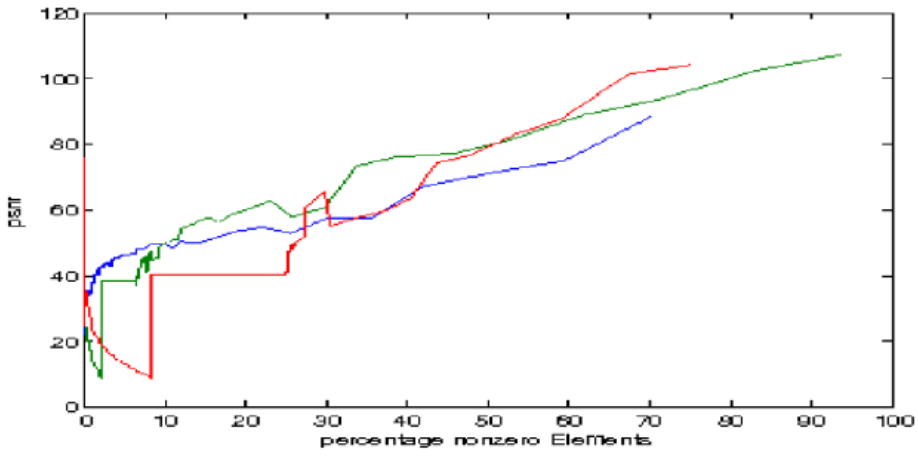


Figure 2.9: $PSNR$ curve of Peppers.tiff, for DCT (blue), HAAR 2nd level (green) and HAAR 1st level (red).

Image	Mandrill		Lena		Peppers	
	NNZ	$PSNR$	NNZ	$PSNR$	NNZ	$PSNR$
DCT	8.97%	30.40	6.00%	23.11	8.67%	49.68
	4.93%	29.39	3.11%	23.03	4.87%	46.26
	1.67%	28.25	2.54	22.02	1.88%	40.39
	0.92%	27.37	0.70%	26.85	0.87%	34.86
Haar 2 niv	8.88%	34.49	6.17%	29.00	8.69%	44.95
	4.93%	27.23	3.12%	29.00	4.94%	38.37
	1.67%	11.87	2.54	29.00	1.88%	9.74
	0.91%	16.38	0.71%	15.38	0.87%	15.57
Haar 1 niv	8.79%	30.55	6.16%	9.53	8.69%	40.13
	4.96%	14.93	3.11%	14.63	4.93%	12.72
	1.69%	21.91	2.54	16.41	1.89%	19.69
	0.91%	24.50	0.71%	24.18	0.86%	25.31

Table 2.2: Numerical Results for Images of tiff type

2.4 Discussion

Now let comment on the results obtained from the reconstructed images and the calculation of $PSNR$ and NNZ. For the image Mandrill.png (respectively: Mandrill.tiff), we note that for the method based on DCT, we can reconstruct the image with percentages of non-zero coefficients of the DCT up to 0.89 and $PSNR$ greater than 30 (respectively: up to 0.91 and $PSNR$ near 30). In the case of the method based on DWT (Haar of 2nd level), we can only reconstruct the mandrill.png image with 8.75 of non-zero coefficients and a $PSNR$ equal to 33 (respectively: 8.88 and $PSNR = 34$). About the image Lena.png (respectively: Lena.tiff), we find that through the method based on DCT, the image can be reconstructed with percentages of non-zero coefficients up to 0.71 and a $PSNR$ greater than 30 (respectively: up to 0.70 with $PSNR = 27$), in the case of the method based on DWT, the reconstructed images Lena.png and Lena.tiff are bad. Finally, for the last image, Peppers.png (respectively: Peppers.tiff), we note that using the method based on DCT, we can reconstruct the image Pepppers.png, with percentages of non-zero DCT coefficients up to 0.88 and a $PSNR$ of more than 30 (respectively: up to 0.87 with $PSNR = 34.86$). In the case of the method based on DWT (2nd level of Haar), we can only reconstruct the image Peppers.png with non-zero coefficients of 8.78 and a $PSNR = 50$ (respectively: 8.69 with $PSNR = 44.95$). We need to draw attention

here to the fact that through the method based on DCT, for non-zero coefficients of 8.78, we have a $PSNR = 53$ (respectively: for 8.69 we get $PSNR = 44.95$).

For the reconstruction of images from a data basis, we need to store the matrix of parameters β_j , which is a matrix of size $512 * 512$; its storage has no effect on the gain of bits even if we have a large data basis of images (d is large).

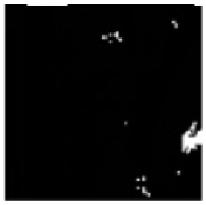
By DCT	By HAAR 1 level	By HAAR 2 level
		
PSNR=30.33 **NNZ=6.17** %	**PSNR=17.87** **NNZ=6.21**%	**PSNR=5.83** **NNZ=6.17** %
		
PSNR=32.02 **NNZ=3.12** %	**PSNR=22.7** **NNZ=3.13** %	**PSNR=11.17** **NNZ=3.12** %
		
PSNR=28.90 **NNZ=2.5** %	**PSNR=23.83** **NNZ=2.55** %	**PSNR=13.59** **NNZ=2.51** %
		
PSNR=31.67 **NNZ=0.71** %	**PSNR=20.13** **NNZ=0.71** %	**PSNR=19.89** **NNZ=0.71** %

Figure 2.10: Reconstructed images for lena.png

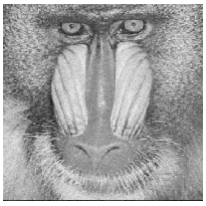
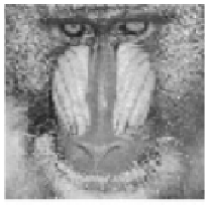
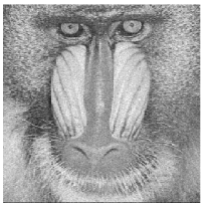
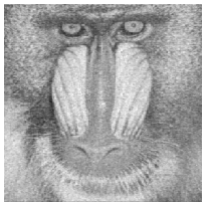
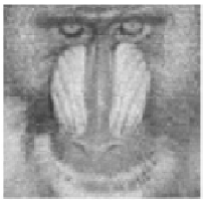
By DCT	By HAAR 1 level	By HAAR 2 level
		
PSNR=43.31 **NNZ=8.75** %	**PSNR=13.08** **NNZ=8.79**%	**PSNR=33.01** **NNZ=8.69** %
		
PSNR=39.47 **NNZ=4.97** %	**PSNR=16.90** **NNZ=4.98** %	**PSNR=7.27** **NNZ=4.97** %
		
PSNR=36.73 **NNZ=1.69** %	**PSNR=23.99** **NNZ=1.69** %	**PSNR=14.27** **NNZ=1.69** %
		
PSNR=30.89 **NNZ=0.89** %	**PSNR=27.75** **NNZ=0.91** %	**PSNR=18.36** **NNZ=0.89** %

Figure 2.11: Reconstructed images for mandrill.png

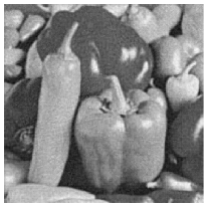
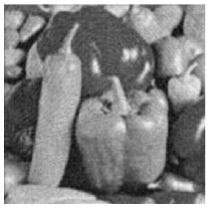
By DCT	By HAAR 1 level	By HAAR 2 level
		
PSNR=53.8 **NNZ=8.78 %**	**PSNR=17.87** **NNZ=8.80%**	**PSNR=50.03** **NNZ=8.78 %**
		
PSNR=51.34 **NNZ=4.96 %**	**PSNR=23.69** **NNZ=4.94 %**	**PSNR=9.21** **NNZ=4.96 %**
		
PSNR=42.99 **NNZ=1.86 %**	**PSNR=34.09** **NNZ=1.88 %**	**PSNR=19.82** **NNZ=1.87 %**
		
PSNR=36.66 **NNZ=0.88 %**	**PSNR=40.71** **NNZ=0.89 %**	**PSNR=26.71** **NNZ=0.88%**

Figure 2.12: Reconstructed images for pepers.png

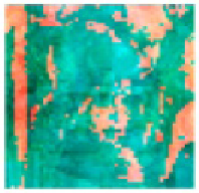
By DCT	By HAAR 1 level	By HAAR 2 level
		
PSNR=23.11 **NNZ=6.00** %	**PSNR=9.53** **NNZ=6.16**%	**PSNR=29.00** **NNZ=6.17** %
		
PSNR=23.03 **NNZ=3.11** %	**PSNR=14.63** **NNZ=3.11** %	**PSNR=29.00** **NNZ=3.12** %
		
PSNR=22.02 **NNZ=2.54** %	**PSNR=16.41** **NNZ=2.54** %	**PSNR=29.00** **NNZ=2.54** %
		
PSNR=26.85 **NNZ=0.70** %	**PSNR=24.18** **NNZ=0.71** %	**PSNR=15.38** **NNZ=0.71** %

Figure 2.13: Reconstructed images for lena.tiff

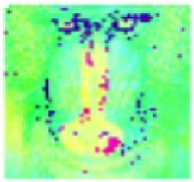
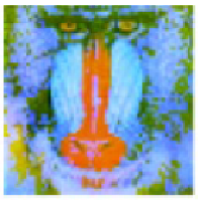
By DCT	By HAAR 1 level	By HAAR 2 level
		
PSNR=30.40 **NNZ=8.97** %	**PSNR=30.55** **NNZ=8.79**%	**PSNR=34.49** **NNZ=8.88** %
		
PSNR=29.39 **NNZ=4.93** %	**PSNR=14.93** **NNZ=4.96** %	**PSNR=27.23** **NNZ=4.94** %
		
PSNR=28.25 **NNZ=1.67** %	**PSNR=21.91** **NNZ=1.69** %	**PSNR=11.87** **NNZ=1.67** %
		
PSNR=27.37 **NNZ=0.92** %	**PSNR=24.50** **NNZ=0.91** %	**PSNR=16.38** **NNZ=0.91** %

Figure 2.14: Reconstructed images for mandrill.tiff

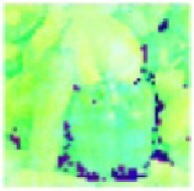
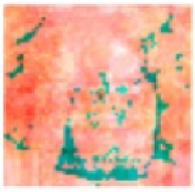
By DCT	By HAAR 1 level	By HAAR 2 level
		
PSNR=49.68 **NNZ=8.67 %**	**PSNR=40.13** **NNZ=8.69%**	**PSNR=44.95** **NNZ=8.69 %**
		
PSNR=46.26 **NNZ=4.87 %**	**PSNR=12.72** **NNZ=4.93 %**	**PSNR=38.37** **NNZ=4.94 %**
		
PSNR=40.39 **NNZ=1.88 %**	**PSNR=19.69** **NNZ=1.89 %**	**PSNR=9.74** **NNZ=1.88 %**
		
PSNR=34.86 **NNZ=0.87 %**	**PSNR=25.31** **NNZ=0.86 %**	**PSNR=15.57** **NNZ=0.87%**

Figure 2.15: Reconstructed images for pepers.tiff

Chapter 3

A novel binary image encryption algorithm

In this chapter, we propose a new algorithm to encrypt binary images. In the first step, the image is split into d blocks, which is used in new images of the same size as the original one, and represent them in the new basis given in 1.2.4 to obtain a key-image and encrypted images. The parameters obtained by this transformation are considered as key-image for the encryption and decryption algorithm. The decryption algorithm is performed by the subtraction between each encrypted image and key-image, then summing them in an image to obtain the original one. In the same way, we can apply our proposed algorithm to encrypt a database of binary images. Experimental results demonstrate the efficiency of the proposed approach.

3.1 Cryptography

Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, authentication and Non-repudiation. [1]

Confidentiality is the fundamental security service provided by cryptography. It is a security service that keeps the information from an unauthorized person.

Data integrity It is security service that deals with identifying any alteration to the data. The data may get modified by an unauthorized entity intentionally or accidently. Integrity service confirms that whether data is intact or not since it was last created, transmitted, or stored by an authorized user. Data integrity cannot prevent the alteration of data, but provides a means for detecting whether data has been manipulated in an unauthorized manner.

Authentication provides the identification of the originator. It confirms to the receiver that the data received has been sent only by an identified and verified sender.

Non-repudiation is a security service that ensures that an entity cannot refuse the ownership of a previous commitment or an action. It is an assurance that the original creator of the data cannot deny the creation or transmission of the said data to a recipient or third party. Non-repudiation is a property that is most desirable in situations where

there are chances of a dispute over the exchange of data.

3.1.1 Cryptosystem

A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services. A cryptosystem is also referred to as a **cipher system**.

The basic model of a cryptosystem that provides confidentiality to the information being transmitted is illustrated in Figure 3.1 below:

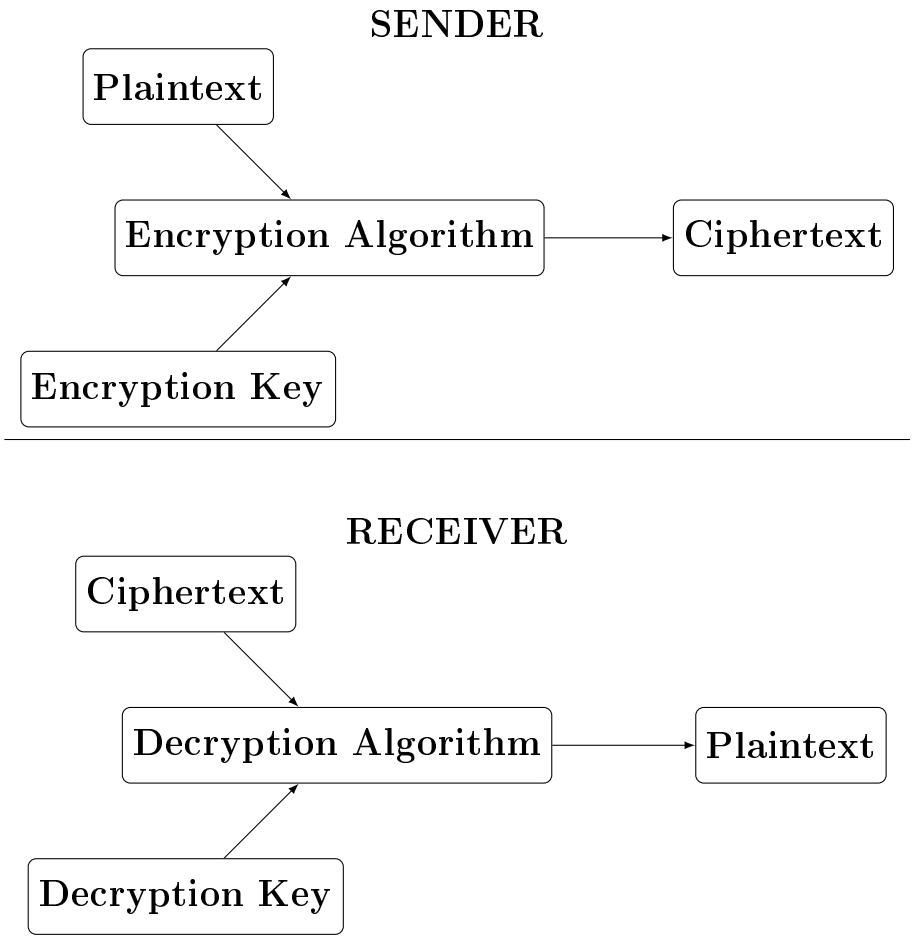


Figure 3.1: Cryptosystem scheme

The illustration shows a sender who wants to transfer some sensitive data to a receiver in such a way that any party intercepting or eavesdropping on the communication channel cannot extract the data.
The objective of this simple cryptosystem is that at the end of the process, only the sender and the receiver will know the plaintext. The various components of a basic cryptosystem are as follows:

1. **Plaintext.** It is the data to be protected during transmission.

2. **Encryption algorithm.** It is a mathematical process that produces a ciphertext for any given plaintext and encryption key. It is a cryptographic algorithm that takes plaintext and an encryption key as input and produces a ciphertext.

3. **Ciphertext.** It is the scrambled version of the plaintext produced by the encryption algorithm using a specific the encryption key. The ciphertext is not guarded. It flows on public channel. It can be intercepted or compromised by anyone who has access to the communication channel.

4. **Decryption algorithm.** It is a mathematical process, that produces a unique plaintext for any given ciphertext and decryption key. It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and outputs a plaintext. The decryption algorithm essentially reverses the encryption algorithm and is thus

closely related to it.

5. **Encryption key.** It is a value that is known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the ciphertext.

6. **Decryption key.** It is a value that is known to the receiver. The decryption key is related to the encryption key, but is not always identical to it. The receiver inputs the decryption key into the decryption algorithm along with the ciphertext in order to compute the plaintext.

For a given cryptosystem, a collection of all possible decryption keys is called a **key space**.

An **interceptor** (an attacker) is an unauthorized entity who attempts to determine the plaintext. He can see the ciphertext and may know the decryption algorithm. He, however, must never know the decryption key.

3.1.2 Types of Cryptosystems

Fundamentally, there are two types of cryptosystems based on the manner in which encryption-decryption is carried out in the system [1]:

1. **Symmetric Key Encryption.** The encryption process where same keys are used for encrypting and decrypting the information

is known as Symmetric Key Encryption.

A few well-known examples of symmetric key encryption methods are: Digital Encryption Standard (DES), Triple DES (3DES), IDEA, and BLOWFISH.

2. **Asymmetric Key Encryption.** The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption. Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible.

3.2 The proposed scheme

In the proposed idea, the transformation of an original binary image to another encrypted image is inspired from the work proposed by Mokhatri and Melkemi in [35] discussed in paragraph diffuse representation of an image in chapter 1.

First of all, let's start this section with an example that illustrates the application of the transform proposed by Mokhatri and Melkemi in [35], in order to apply it to the binary image encryption.

Example

The basis $\{e_j\}_{j=1,n}$ is the canonical basis of the vector space $\mathbb{R}^n$ with dimension n, in the following example we work with matrices with dimension 8×8 the dimension of our vector space is $n = 8 \times 8$, and the elements of this basis are matrices defined by :

$$e_{i,j}(s,l) = \begin{cases} 1 \text{ for } (s,l) = (i,j) \\ 0 \quad \text{otherwise} \end{cases} , s, l = 1, ..., 8$$

Let be I a matrix $n = 8 \times 8$ such that $I(i,j) \in \{0,1\}$.

$$I = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

If, we split matrix I vertically into $d = 2$ blocks, and construct 2

new matrices $I1$ and $I2$ with the same size as I such that each matrix contains one of the blocks and the remaining value is zero ($I = I1 + I2$), we get

$$I1 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$I2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

So, we have $a_1(i,j)$ the element of matrix I_1 and $a_2(i,j)$ the element of matrix I_2. Now, we transform the 2 matrices $I1$ and $I2$ into new basis. We obtain the matrix *beta* of parameters which are obtained by this transformation by applying equation 1.35

$$\beta\left(i,j\right)=\frac{1}{2}\left[\left(a_1\left(i,j\right)+\frac{\left\|I_1\right\|_1}{\sqrt{64}}\right)+\left(a_2\left(i,j\right)+\frac{\left\|I_2\right\|_1}{\sqrt{64}}\right)\right],i,j=1,...,8$$

and $b1$, $b2$; the representation of 2 matrices $I1$ and $I2$ in this new basis, according to the equation 1.33

$$b_1(i,j)=\beta(i,j)-a_1(i,j),i,j=1,...,8$$

$$b_2(i,j)=\beta(i,j)-a_2(i,j),i,j=1,...,8$$

$$\beta=\begin{pmatrix}2&2&2&2&2&2&2&2\\2&2&2&2&2&2&2&2\\2&2&2&2&2&2&2&2\\2&2&2&2&2&2&2&2\\2&2&2&2&2&2&2&2\\2&2&2&2&2&2&2&2\\2&2&2&2&2&2&2&2\\2&2&2&2&2&2&2&2\end{pmatrix}$$

$$b1 = \begin{pmatrix} 1 & 1 & 2 & 1 & 2 & 2 & 2 & 2 \\ 2 & 1 & 2 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

$$b2 = \begin{pmatrix} 2 & 2 & 2 & 2 & 1 & 2 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 \end{pmatrix}$$

We notice that almost coefficients of matrix $b1$ take the value $2 \sim \frac{\|I_1\|_1}{\sqrt{8\times 8}}$ and also almost coefficients of matrix $b2$ take the value $2 \sim \frac{\|I_2\|_1}{\sqrt{8\times 8}}$

We can return easily to the canonical basis by summation and subtrac-

tion

$$I = (\beta - b1) + (\beta - b2)$$

The proposed idea is applied for both a binary image and a database of d binary images. For a given binary image, we share the original image into d blocks (horizontally or vertically or both), then we construct new images of the same size as the original one, such that each image contains one of the blocks and the value of the remaining pixels is zero. After that, we transform them into the new basis. The matrix of parameters which is obtained by this transformation is called key-image and the images represented in the new basis are called encrypted images. In the decryption step, each encrypted image is subtracted with the key-image, and then all these new images are summed in an image to return-back to the original one.

The proposed encryption approach does not only encrypts a single binary image, but it can be used to encrypt a set of binary images having the same size. Indeed, a database of d binary images having the same size can be encrypted similarly using the proposed method. First, the d binary images are transformed into the new basis. The parameters matrix obtained by this transformation is called key-image and the new images based on this new basis are called encrypted images.

In the same way, the decryption process subtracts each encrypted image from the key-image in order to return back to the original database of d binary images. We explain the two proposed schemes of binary image encryption and decryption in the two pseudo-codes (see Algorithms 2,3).

3.2.1 The proposed algorithms

In this subsection, we present the two pseudo-codes. In the Algorithm 2, we describe the binary image encryption/decryption pseudo-code. The Algorithm 3 presents the binary image database encryption and decryption pseudo-code.

Let be I a binary image of size $m \times n$, to split I into d blocks vertically, we choose d integers $n_1, n_2, ..., n_d$ such that

$$\sum_{k=1}^{d} n_k = n \tag{3.1}$$

$$I = \left[\overbrace{Block1}^{n_1} \middle| \overbrace{Block2}^{n_2} \middle| \overbrace{Block3}^{n_3} \middle| ..... \middle| \overbrace{Blockd}^{n_d} \right] \tag{3.2}$$

then we construct d new images $I_1, I_2, ..., I_d$ of the same size as I, such that each image contains one of the blocks and the value of the remaining pixels is zero,

$$I_1 = \left[\overbrace{Block1}^{n_1} \middle| \overbrace{Zeros}^{n-n_1}\right], \; I_2 = \left[\overbrace{Zeros}^{n_1} \middle| \overbrace{Block2}^{n_2} \middle| \overbrace{Zeros}^{n-(n_1+n_2)}\right], \tag{3.3}$$

$$I_3 = \left[\overbrace{Zeros}^{n_1+n_2} \middle| \overbrace{Block3}^{n_3} \middle| \overbrace{Zeros}^{n-(n_1+n_2+n_3)}\right], ..., I_d = \left[\overbrace{Zeros}^{n-n_d} \middle| \overbrace{Blockd}^{n_d}\right] \tag{3.4}$$

finally, we transform them into the new basis, by computing matrix β and $b_1, b_2, ..., b_d$ representation of $I_1, I_2, ..., I_d$ in this new basis, using equations 1.33 and 1.35. Matrix β is used as key-image and $b_1, b_2, ..., b_d$ are used as d encrypted images. For decryption, we can easily obtain the original image I by computing

$$\sum_{k=1}^{d} (\beta - b_k) = (d \times \beta) - \sum_{k=1}^{d} b_k \tag{3.5}$$

For a database of d binary images having the same size, the d binary images are transformed into the new basis, using equations 1.33 and 1.35. The parameters matrix obtained by this transformation is called key-image and the new images based on this new basis are called encrypted images. In the same way, the decryption process subtracts

each encrypted image from the key-image in order to return back to the original database of d binary images.

Algorithm 2 The proposed binary image encryption pseudo code

Encryption:
Step 1. Initialization;
Input the image to encrypt;
Split the image in d blocks;
Form d images with the same size as the original one;
Step 2: Construction of the key-image and encrypted image;
Compute the new basis (compute β_j *and* b_j*);*
Save β_j *as the key-image;*
Save b_j *as the encrypted images;*
Decryption:
Step 1. Initialization;
Input the key-image and encrypted images;
Return to the canonical basis;
Step 2: Decryption step;
Sum the d splits;
Construct and display the decrypted image;

Algorithm 3 The proposed binary image database encryption pseudo code

Encryption:
Step 1. Initialization;
Input the database of d images of the database to encrypt;
Step 2: Construction of the key-image and the encrypted images of the database;
Compute the new basis (compute β_j and b_j);
Save β_j as the key image;
Save b_j as the database of encrypted images;
Decryption:
Step 1. Initialization;
Input the key-image and the encrypted images of the database;
Step 2: Decryption step;
Return to the canonical basis to get the decrypted images;

3.3 Encryption evaluations metrics

To evaluate the efficiency of our cryptography scheme, we will choose some basic parameters to evaluate our algorithm. One of the major parameters to examine the encrypted image is the visual inspection [23],[5], another parameter is the study of; characteristic diffusion [50],[19], which is measured to judge the randomization algorithm. If an algorithm has good diffusion characteristic, the relationship between the encrypted image and the original image is too complex and can not be easily predicted. In this work, we propose to study, in detail, the following metrics: the calculation of $PSNR$, the correlation between the key-image and the encrypted image; and in the end we evaluate the characteristics diffusion through the calculation of two parameters $NPCR$ and $UACI$.

3.3.1 Correlation Coefficient

Correlation is a measure of the relationship between two variables. If the two variables are images, and the correlation coefficient equals zero, then those images are totally different. If the correlation coefficient equals -1, this means that one of them is the negative of the other. So, success of the encryption process can be confirmed by smaller values of the correlation coefficient, which is given by the following equation :

$$corr = \frac{\sum_{i=1}^{N}(x_i - E(x)).(y_i - E(y))}{\sqrt{\sum_{i=1}^{N}(x_i - E(x)^2)}.\sqrt{\sum_{i=1}^{N}(y_i - E(y)^2)}} \tag{3.6}$$

Where

$$E(x) = (1/N).\sum_{i=1}^{N} x_i \tag{3.7}$$

3.3.2 Characteristics diffusion

Diffusion is an important parameter that must be measured to judge the encryption algorithm randomization. To test the security of the image encryption algorithm, two common measures may be used: Number of Pixels Change Rate ($NPCR$) and Unified Average Changing Intensity ($UACI$) [20],[49] Let $C1$ and $C2$ be two images with size $N \times M$, we

define an array, D, with the same size as images $C1$ and $C2$ by :

$$D(i,j) = \begin{cases} 0 \text{ if } C1(i,j) = C2(i,j) \\ 1 \text{ if } C1(i,j) \neq C2(i,j) \end{cases} \tag{3.8}$$

The $NPCR$ is defined as:

$$NPCR = \frac{\sum_{i,j=1}^{N \times M} D(i,j)}{N \times M} \times 100\% \tag{3.9}$$

$NPCR$ measures the percentage of different pixel numbers between these two images.

If $C2$ is the encryption image of $C1$, the $UACI$ is defined as:

$$UACI = \frac{1}{N \times M} \left[\sum_{i,j=1}^{N \times M} \frac{|C1(i,j) - C2(i,j)|}{MAX(C2)} \right] \times 100\% \tag{3.10}$$

Which measures the average intensity of differences between the two images. $NPCR$ and $UACI$ are used in security analysis of image encryption for differential attacks.

3.4 Experimental results

3.4.1 Numerical tests and visual results

We summarize our numerical tests and visual results in Tables 3.1, 3.2, 3.3, 3.4, and Figures 3.2, 3.3, 3.4, 3.5.

We perform the Algorithm 2 on three binary images: text, cartoon and Lena. Each image is split vertically into $d = 8$ blocks, key-image and encrypted images are saved in png and in jpeg2000 (jp2 in short) formats (see Figures 3.2 and 3.3). To assess the quality of the recovered image, $PSNR$ is used and calculated for these images in the two formats. To test the security of the image encryption algorithm. The parameters $NPCR$ and $UACI$ (see Table 3.1) and entropy values (see Table 3.4) are calculated. For correlation values, we note that for algorithm 1, all pixels of key-image are equals in the 3 images test (lena, cat and text), so correlation value between key-images and encrypted images is NAN.

For a database of images, we apply our proposed Algorithm 3 on a document containing $d = 6$ pages. The key-image and the encrypted images are saved in png and jp2 formats (see Figures 3.4 and 3.5). In the same way as presented above, the well-known $PSNR$, $NPCR$ and $UACI$ are computed in order to assess the performance of the proposed approach (see Table 3.2), also correlation values (see Table 3.3)and entropy values (see Table 3.4) are calculated.

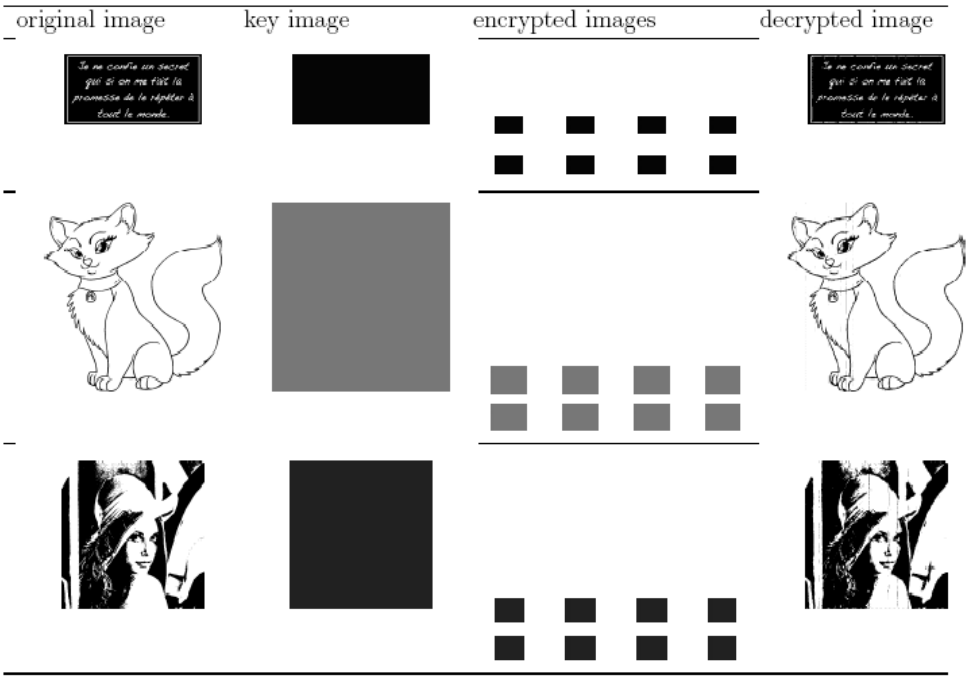


Figure 3.2: Visual results for Algorithm 2 applied on the images text, cat and Lena. Key-image and encrypted images are saved in jp2 format.

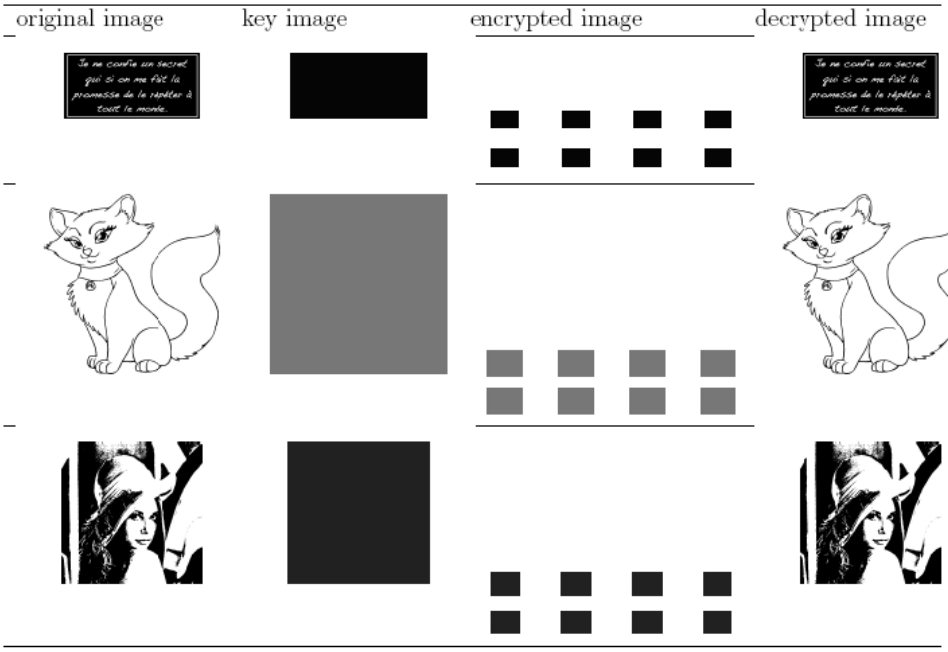


Figure 3.3: Visual results for Algorithm 2 applied on images, text, cat and Lena. Key-image and encrypted images are saved in png format.

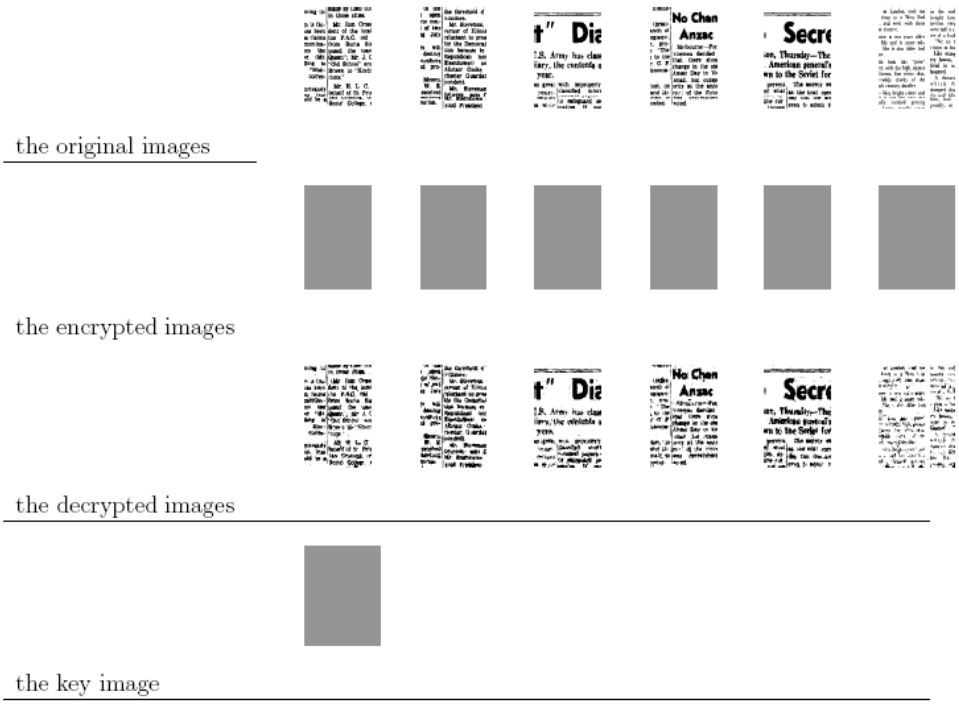


Figure 3.4: Visual results for Algorithm 3 applied on a dataset (document) containing 6 binary images (pages of the document). Key-image and encrypted images are saved in jp2 format.

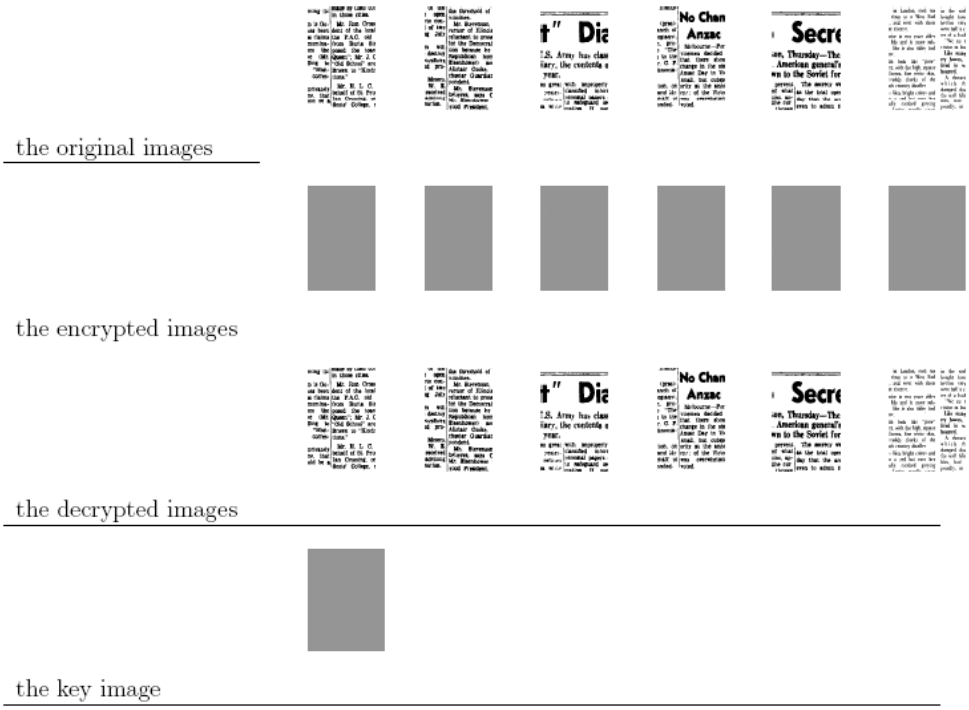


Figure 3.5: Visual results for Algorithm 3 applied on a dataset (document) containing 6 binary images (pages of the document). Key-image and encrypted images are saved in png format.

	Type	Text	Cat	Lena
$PSNR$	png	∞	∞	∞
	jp2	63.28	66.10	61.04
$NPCR$	png	100 %	100 %	100 %
	jp2	100 %	100 %	100 %
$UACI$	png	97.55 %	99.16 %	98.29 %
	jp2	81.28 %	98.33 %	95.40 %

Table 3.1: Numerical results of Algorithm 2.

	Type	Page 1	Page 2	Page 3	Page 4	Page 5	Page 6
$PSNR$	png	∞	∞	∞	∞	∞	∞
	jp2	57.91	57.68	59.62	58.42	59.16	57.37
$NPCR$	png	100 %	100 %	100 %	100 %	100 %	100 %
	jp2	100 %	100 %	100 %	100 %	100 %	100 %
$UACI$	png	98.96 %	99.02 %	98.25 %	98.98 %	98.96 %	98.84 %
	jp2	98.32 %	98.38 %	98.36 %	98.34 %	90.34 %	98.19 %

Table 3.2: Numerical results for Algorithm 3.

Encrypted page	1	2	3	4	5	6
Correlation	0.05	0.06	0.04	0.04	0.05	0.08

Table 3.3: Correlation values between key-image and 6 encrypted images for document of 6 pages.

3.4.2 Discussion

To demonstrate the efficiency of the proposed method, we have applied the Algorithms 2 on the three binary images with different sizes and the Algorithm 3 on a data-set called document containing 6 pages as binary images.

A text 227 $\times$ 467 pixels, cat 1024 $\times$ 994 pixels, Lena 512 $\times$ 512

Image	Origi	Encr1	Encr2	Encr3	Encr4	Encr5	Encr6	Encr7	Encr8	Key
Lena	0.9996	0.1122	0.3125	0.0541	0.0643	0.1302	0.1878	0.1864	0.9457	0
Text	0.5071	0.0640	0.0414	0.0759	0.0681	0.0642	0.0660	0.0872	0.3139	0
Cat	0.3117	0.1918	0.1896	0.2374	0.1888	0.1834	0.1822	0.2293	0.8486	0

Page	1	2	3	4	5	6
Original	0.7789	0.8431	0.7642	0.8021	0.8072	0.5822
Encrypted	0.7783	0.8463	0.7625	0.8006	0.8078	0.5903

Entropy of key-image (document containing 6 pages) = 0.0361

Table 3.4: Entropy values for original image, key-image and encrypted images

pixels, and the document of 6 pages with 256×256 pixels.

By analyzing the values of the calculated parameters, we can notice that in the case where the encrypted images and the key image are saved in the png format, we get the infinite value of $PSNR$, so the recovered image and the original one are identical. We can find it as well in other case (jp2 format), we got a big values for the $PSNR$, that we can confirm the good recovery of our images.

By using the $NPCR$ and $UACI$ criteria, we notice that the $NPCR$ is 100 per 100 for the both cases (png, jp2) and the two tests (images and document), that means that all the pixels change those values in

the encrypted images. For the $UACI$ test, the almost obtained values are near of 99 per cent.

By interpreting the correlation values in Table 3.3, we note that we got smaller values, this means that the encryption process is good.

By studying the entropy values obtained in Table 3.4, we confirm that key-image has 0 entropy for three tests; lena, cat and text, this means that key-image contains only one value, and we also note that the encrypted images, have a lower entropy than 1, so there is not a problem of storage.

Conclusion

From the proposed transformation of the representation basis of images given by Melkemi and Mokhatri in [35]; that gives a representation to diffuses the images in the new basis, in order to share information quantity almost equal in different images of data-set; we have succeeded to get two new results, the first concerns compression of an database of images, and the other concern the cryptography of a binary image or a binary database images.

In chapter 2, we are applied a method based on DCT and one based on DWT for the types of images: gray level (png) or colored (tiff). From the results obtained, we conclude that the method based on DCT is very efficient compared to the one based on DWT, because, using the DCT method, we were able to reconstruct the image with a percentage of pixels below 1, so it allows us to obtain a very significant gain in storage space.

In chapter 3, we have proposed a new encryption algorithm to protect binary images and databases of binary images having the same size. The main idea of this approach is based on the generation of a key as an image. This key-image is obtained from the proposed transformation, which get all the pixels values almost equal. Indeed, it helps us to present the encrypted and key images in one color (black sheet). In addition, this algorithm can encrypt not only an image but also a database of binary images having the same size.

We have shown with empirical evidence that this algorithm can be used efficiently in transferring a secret binary image, scanned document containing several pages, confidential queries containing binary images.

Bibliography

[1] Cryptography just for beginners. http://www.tutorialspoint.com /cryptography.

[2] Huffman compression. http://www.webopedia.com /Huffman$_c$*ompression.html*.

[3] what is a digital image. https://sites.google.com/site/learnimagej/ image-processing/what-is-a-digital-image.

[4] Anane, N., Anane, M., Bessalah, H., Issad, M., and Messaoudi, K. Rsa based encryption decryption of medical images. In *Systems Signals and Devices (SSD), 2010 7th International Multi-Conference on* (2010), IEEE, pp. 1–4.

[5] Bailey, K., and Curran, K. An evaluation of image based steganography methods. *Multimedia Tools and Applications 30*, 1 (2006), 55–88.

[6] Baran, N. News and views: Rsa algorithm in the public domain; woz joins the inventors hall of fame; entangled photons mean faster, smaller

ics; behemoth mothballed; advanced encryption standard selected; sgi releases sdk as open source; wsdl spec released. *Dr. Dobbâs Journal of Software Tools 25*, 12 (2000), 18.

[7] Barker, W. C., and Barker, E. B. Sp 800-67 rev. 1. recommendation for the triple data encryption algorithm (tdea) block cipher. *National Institute of Standards & Technology* (2012).

[8] Biham, E., and Shamir, A. *Differential cryptanalysis of the data encryption standard*. Springer Science & Business Media, 2012.

[9] Blelloch, G. E. Introduction to data compression. *Computer Science Department, Carnegie Mellon University* (2001).

[10] Bourbakis, N. A language for efficient accessing of a 2d array. In *IEEE Workshop on LFA, Singapore* (1986), pp. 52–58.

[11] Bourbakis, N., and Alexopoulos, C. Picture data encryption using scan patterns. *Pattern Recognition 25*, 6 (1992), 567–581.

[12] Bourbakis, N. G., Alexopoulos, C., and Klinger, A. A parallel implementation of the scan language. *Computer languages 14*, 4 (1989), 239–254.

[13] Chikouche, D., Benzid, R., and Bentoumi, M. Application of the dct and arithmetic coding to medical image compression. In *Information*

and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on (2008), IEEE, pp. 1–5.

[14] Chung, K.-L., and Chang, L.-C. Large encrypting binary images with higher security. *Pattern Recognition Letters 19*, 5 (1998), 461–468.

[15] Clarke, R. J. Transform coding of images. *Astrophysics 1* (1985).

[16] Das, A., and Adhikari, A. An efficient multi-use multi-secret sharing scheme based on hash function. *Applied mathematics letters 23*, 9 (2010), 993–996.

[17] Douak, F., Benzid, R., and Benoudjit, N. Color image compression algorithm based on the dct transform combined to an adaptive block scanning. *AEU-International Journal of Electronics and Communications 65*, 1 (2011), 16–26.

[18] Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *International Journal of Bifurcation and chaos 8*, 06 (1998), 1259–1284.

[19] Fu, C., Chen, J.-j., Zou, H., Meng, W.-h., Zhan, Y.-f., and Yu, Y.-w. A chaos-based digital image encryption scheme with an improved diffusion strategy. *Optics Express 20*, 3 (2012), 2363–2378.

[20] Fu, C., Chen, J.-j., Zou, H., Meng, W.-h., Zhan, Y.-f., and Yu, Y.-w. A chaos-based digital image encryption scheme with an improved diffusion strategy. *Optics Express 20*, 3 (2012), 2363–2378.

[21] Gong, L., Liu, X., Zheng, F., and Zhou, N. Flexible multiple-image encryption algorithm based on log-polar transform and double random phase encoding technique. *Journal of Modern Optics 60*, 13 (2013), 1074–1082.

[22] Gonzalez, R. C., and Woods, R. E. Digital image processing. *Nueva Jersey* (2008).

[23] Hashim, A. T. Measurement of encryption quality of bitmap images with rc6, and two modified version block cipher. *Eng. Tech. Journal 28* (2010).

[24] Houas, A., Mokhtari, Z., Melkemi, K. E., and Boussaad, A. A novel binary image encryption algorithm based on diffuse representation. *Engineering Science and Technology, an International Journal 19*, 4 (2016), 1887–1894.

[25] Jackson, D. J., and Hannah, S. J. Comparative analysis of image compression techniques. In *System Theory, 1993. Proceedings SSST'93., Twenty-Fifth Southeastern Symposium on* (1993), IEEE, pp. 513–517.

[26] Jain, A. K. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[27] Jia, W., Wen, F. J., Chow, Y. T., and Zhou, C. Binary image encryption based on interference of two phase-only masks. *Applied optics 51*, 21 (2012), 5253–5258.

[28] Khayam, S. A. The discrete cosine transform (dct): theory and application. *Michigan State University 114* (2003).

[29] Lee, D. T., and Yamamoto, A. Wavelet analysis: theory and applications. *Hewlett Packard journal 45* (1994), 44–44.

[30] Lim, J. S. Two-dimensional signal and image processing. *Englewood Cliffs, NJ, Prentice Hall, 1990, 710 p. 1* (1990).

[31] Makrogiannis, S., Economou, G., and Fotopoulos, S. Region oriented compression of color images using fuzzy inference and fast merging. *Pattern recognition 35*, 9 (2002), 1807–1820.

[32] Mallat, S. *A wavelet tour of signal processing*. Academic press, 1999.

[33] Mazloom, S., and Eftekhari-Moghadam, A. M. Color image encryption based on coupled nonlinear chaotic map. *Chaos, Solitons & Fractals 42*, 3 (2009), 1745–1754.

[34] Mohamed, F. K. A parallel block-based encryption schema for digital images using reversible cellular automata. *Engineering Science and Technology, an International Journal 17*, 2 (2014), 85–94.

[35] Mokhtari, Z., and Melkemi, K. A new watermarking algorithm based on entropy concept. *Acta applicandae mathematicae 116*, 1 (2011), 65–69.

[36] Nithin, N., Bongale, A. M., and Hegde, G. Image encryption based on feal algorithm. *International Journal of Advances in Computer Science and Technology 2*, 3 (2013), 14–20.

[37] Padmaja, V., and Chandrasekhar, B. Literature review of image compression effects on face recognition. *International Multidisciplinary Research Journal 2*, 8 (2012).

[38] Pearlman, W. A., Islam, A., Nagaraj, N., and Said, A. Efficient, low-complexity image coding with a set-partitioning embedded block coder. *IEEE transactions on circuits and systems for video technology 14*, 11 (2004), 1219–1235.

[39] Pratt, W. K. Image enhancement. *Digital Image Processing: PIKS Scientific Inside, Fourth Edition* (2001), 247–305.

[40] Radha, H. Lecture notes: Ece 802-information theory and coding, 2003.

[41] Rawat, N., Ni, P., and Kumar, R. A fast compressive sensing based digital image encryption technique using structurally random matrices and arnold transform. *arXiv preprint arXiv:1402.4702* (2014).

[42] Rodrigues, J. M. *Image Safe Transfer by Combination of Compression, Encryption and Data Hiding Techniques.* Theses, Université Montpellier II - Sciences et Techniques du Languedoc, Oct. 2006.

[43] Russ, J. C. *The Image Processing Handbook.* CRC Press, 2006.

[44] Sachs, J. Digital image basics. *Digital Light & Color 1999* (1996).

[45] Skodras, A. N., Christopoulos, C. A., and Ebrahimi, T. Jpeg2000: The upcoming still image compression standard. *Pattern Recognition Letters 22*, 12 (2001), 1337–1345.

[46] Sonal, D. K. A study of various image compression techniques. *COIT, RIMT-IET. Hisar* (2007).

[47] Soradge, N. V. A review on various visual cryptography schemes. *International Journal of Computer Science and Business Informatics 12*, 1 (2014).

[48] ter Morsche, H. Wavelet basics. www.win.tue.nl/casa/meetings/seminar/previous/wavelets1.pdf.

[49] Wu, Y., Noonan, J. P., and Agaian, S. Npcr and uaci randomness tests for image encryption. *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)* (2011), 31–38.

[50] Yan, X., Wang, S., Li, L., El-Latif, A. A. A., Wei, Z., and Niu, X. A new assessment measure of shadow image quality based on error diffusion techniques. *J. Inf. Hiding Multimedia Signal Process.(JIHMSP) 4*, 2 (2013), 118–126.

[51] Zhou, G., Zhang, D., Liu, Y., Yuan, Y., and Liu, Q. A novel image encryption algorithm based on chaos and line map. *Neurocomputing 169* (2015), 150–157.

[52] Zhou, N., Wang, Y., and Gong, L. Novel optical image encryption scheme based on fractional mellin transform. *Optics communications 284*, 13 (2011), 3234–3242.