# Digital Watermarking Based on Fast Fourier Transformation Using Encryption

**By**

**Saba Riaz**

**MS-CS 04**


**Submitted to the Department of Computer Engineering in partial fulfillment of the requirements for the degree of**

**Master of Science**

**in**

**Computer Software Engineering**

**Thesis Supervisor**

**Brig. Dr. Yonus Javed**

**College of Electrical & Mechanical Engineering**

**National University of Sciences and Technology**

**2008**

**Acknowledgments**

*First of all, I would like to thank Almighty Allah, with whose blessings; I was able to do all my work.*

*I consider myself very fortunate that I was supported by several people and their support made it possible for me to complete this thesis work. . I would like to thank my parents whose prayers, guidance and continuous encouragement were a source of strength for me. I am deeply grateful to my teachers especially my project supervisor Dr. Younus Javed whose guidance and encouragement was a great source that made it possible for me to complete this thesis. My special appreciation and thanks goes to Dr Almas Anjum, for his guidance, support and encouragement throughout my thesis. I thank every person including my brother M. Raza, my sisters and all my friends for advising and counseling me during my studies and providing a friendly and supportive environment to carry out my MS research. I want to show cordial gratitude to my committee members Dr. Almas Anjum, Dr. Shaleeza Sohail, and Lt. Col. Sajid Nazir. Their dedication and expertise in their respective fields are and will always be a source of inspiration for me. I am thankful to everyone who helped me in this work.*

**Abstract**

Digital watermarking is a technique that embeds data called watermark in an image without knowing its existence to the human eye. However, digital watermarking is an emerging technology which is vulnerable to a number of attacks. Careful selection must be done while selecting a technique for embedding a watermark that might be a tradeoff between the robustness and quality of the information recovered. This selection depends on the need of the application and the use of the watermarked media.

In this thesis, Fourier Transform-based invariant is used for digital image watermarking. Middle frequencies are used to embed data in a circular ring. The embedded mark is designed to be robust against a number of attacks. The original image is not required for extracting the embedded mark.

Based on obscured watermark scheme based on the Fourier Transform, the targeted data is encrypted using an encryption key to enhance security of the secret information. The embedded marks are designed to be robust against a number of exploiting attacks like cropping, flipping, AWGN, histogram equalization and lossy compression etc. The thesis concludes with a discussion on the findings and outcomes when exposed to attacks and results are compared with LSB based spatial domain algorithm. Also future direction of the research is included at the end.

**List of Figures**



**List of Tables**

**Table of Contents**

**Chapter 1**



**Chapter 2**



**Chapter 3**

**Chapter 4**



**Chapter 5**



**Chapter 6**

**Chapter 7**

# Chapter 1

# Introduction

Digital watermarking is relatively a young and emerging research area. This chapter describes about an introduction to watermarking and its advantages in various applications. Also a brief history of watermarking will be presented here.

## 1.1 Introduction to Digital Watermarking

Digital watermarking is a method of embedding data (image or text) called a watermark into a multimedia object by using an insertion algorithm. Watermark object can be detected or extracted later that provides information about the multimedia object. For the detection or extraction process again some algorithm will be applied.

## 1.2 History of Information Hiding

Watermarking technique has evolved from steganography. Steganographic methods made their record debut a few centuries later in several tales by Herdoutous, the father of history [1]. The term of steganography came into use in 15th century after the appearance of Trithemius' book on the subject of "Steganographia".

Open and hidden form of code words to hide actual information was used in World War I. German spies used fake orders for cigar to represent various types of British warships-cruisers and destroyers. So they use the code words of 5000 cigars needed in Portsmouth to show five cruisers were in Portsmouth.

## 1.3 Information Hiding Techniques

Various techniques of information hiding techniques are available depending on different parameters. The classification of techniques is shown in Figure 1.

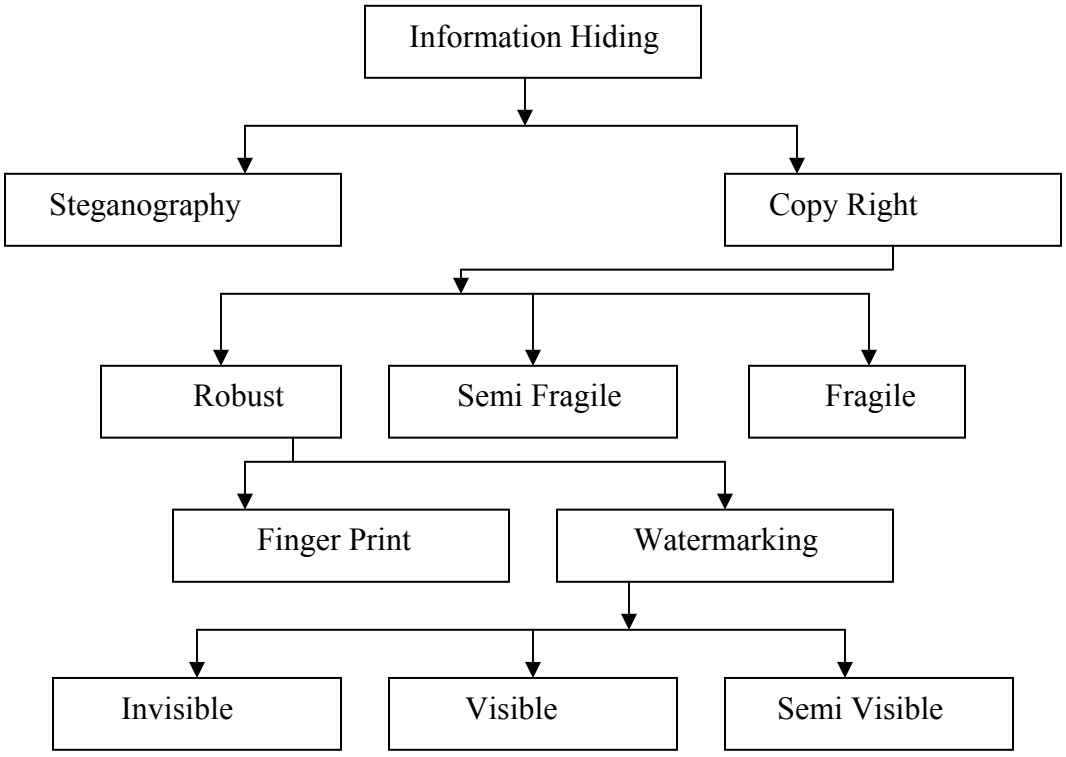



Figure 1.1: Information Hiding Information Techniques.

## 1.4 Steganography vs. Digital Watermarking

As the purpose of steganography is having a covert communication between two parties whose existence is unknown to a possible attacker, a successful attack consists in detecting the existence of this communication (e.g., using statistical analysis of images with and without hidden information). Watermarking, as opposed to steganography, has the (additional) requirement of robustness against possible attacks. In this context, the term 'robustness' is still not very clear; it mainly depends on the application.

## 1.5 Types of Watermarking

Watermarking and its techniques can be divided into categorizes depending upon several parameters. Different types of watermarking techniques are shown in the figure.

### 1.5.1 Domain Based Watermarking

Algorithms for inserting or extracting a watermark are based on two main domain types. These domain types are:

- Spatial Domain
- Frequency Domain

Many of current techniques used for blind watermarking are inspired by manipulating the frequency domain of the image.

### 1.5.2 Document Based Watermarking

Watermarking techniques can be divided into four categories according to the type of document to be watermarked as follows:

- Image Watermarking
- Video Watermarking
- Audio Watermarking
- Text Watermarking

The proposed scheme in this thesis is based on Image Watermarking.

### 1.5.3 Human Perception Based Watermarking

According to the human perception, the digital watermarks can be dividing into three different types as follows:

- Visible watermark

- Invisible-Robust watermark
- Invisible-Fragile watermark
- Dual watermark

### 1.5.4 Application Based Watermarking

Invisible robust watermarks find application in following cases:

- Invisible watermarking to detect misappropriated images. In this scenario, the seller of digital images is concerned, that his, fee-generating images may be purchased by an individual who will make them available for free; this would deprive the owner of licensing revenue.

- Invisible watermarking as evidence of ownership. In this scenario, the seller that of the digital images suspects one of his images has been edited and published without payment of royalties. Here, the detection of the seller's watermark in the image is intended to serve as evidence that the published image is property of seller.

## 1.6 General Applications of Digital Watermarking

Video and audio watermarking can be used in many areas of the production and broadcast chain for a wide range of copyright verification and monitoring applications. Combinations of the applications are allowed and, by using different keys, may coexist completely independently. Applications include:

- **Advertisement verification:** Adding an ID that is unique to each advertisement as a method of checking usage.

- **Broadcasting:** Adding copyright notices and then monitoring to check use (and abuse). Detectors in the field monitor broadcast channels and record details such as channel number, time and length of use. This data can be reported to initiate payments, etc.

- **Distribution (fingerprinting):** Adding copyright notices and identifying recipients; tracing the source of illegal copies.

- **Contents ID and archive:** Adding meta-data (e.g. owner, title, scene, director, cameraman, location, etc) to material for archive.

- **Mastering:** Adding copyright notices as proof of original ownership.

- **Medical Images:** Copy right protection of medical images is getting much of the importance in these days. In many instances, databases of medical images may require copyright protection. Watermarking these images help to protect the unauthorized distribution of medical images and keep a security of the patient.

Digital watermarking has a broad range of uses across many industries. Digital watermarking creates a digital identity for all media content, including photos, secure documents, advertisements, TVs, movies, and music, in digital or analog formats.
As such, digital watermarking has relevance across a broad array of applications from counterfeiting and piracy deterrence, media management and identification and authentication to monitoring and mobile e-commerce.

## 1.7 Attacks on Watermarks

Watermark schemes are often vulnerable to variety of attacks. A watermarked image is likely to be subjected to certain manipulations, such as compression, noise cropping, filtering, histogram equalization etc. Detail is shown here:

Lossy Compression: Many compression schemes like JPEG and MPEG can potentially degrade the data's quality through irretrievable loss of data.
Geometric Distortions: Geometric distortions are specific to images videos and include such operations as rotation, translation, scaling and cropping.
Common Signal Processing Operations: They include the followings.
– D/A conversion
– A/D conversion
– Resampling
– Requantization
– Dithering distortion
– Recompression
– Linear filtering such as high pass and low pass filtering
– Non-linear filtering such as median filtering
– Color reduction

All these attacks are not covered in this thesis. The emphasis is mainly on Compression, Geometric and Noise Attacks.

## 1.8 Overview of Thesis

Thesis comprises of some chapters. Chapter 1 is related to an introduction to the watermark. Chapter 2 is based on the literature review of the past techniques that were used for blind watermarking. Chapter 3 consists of a proposed approach. Chapter 4 is the comparison of the proposed approach with LSB based watermarking in spatial domain. Chapter 5 comprises of the results of the proposed scheme after the attacks are applied on the proposed scheme. Chapter 6 is a discussion on the results. The last Chapter 7 describes the future work.

### 1.8.1 Organization of Thesis

This thesis will be concentrating on the invisible, robust watermarking algorithm using Frequency Domain taking into consideration Geometric Attacks and security of the system along with Human Visual System (HVS) in consideration. The algorithm is compared with LSB based watermarking in spatial domain. Efficient implementation of the algorithms will be presented so that the algorithms can be applied in run time applications.

# Chapter 2

# Literature Review

This chapter will give the overview of the existing watermarking techniques and their advantages and discrepancy. At the end, the scope of this thesis will be described.

## 2.1 Overview of Existing Watermarking Techniques

The digital watermarking is a relatively young field, about 10 years old. Practically, in the context of digital image/video watermarking, the first papers appeared in the 1994, the debut being made by Matsui [2] and Schyndel [3]. The method proposed by Matsui is extension of Turner's [4] method which was proposed for audio data. They set the basis for the so called Least Significant Bit (LSB) watermarking. While this technique works in a noise free environment, it is totally useless when comes to robustness. Indeed, this method may easily be circumvented. For example, if it is known that the algorithm only affects the least significant bits of a word, then it is possible to randomly flip all such bits,

there by destroying any existing watermark. In spite of this drawback quite a few variants were developed during 1994 and 1995.

The real breakthrough came in 1995/1996 in digital watermarking domain. More sophisticated variants on this theme involved spread-spectrum techniques. Although these techniques have been used since the mid-fifties for military purposes because of their anti-jamming and low probability of intercept properties [5], their applicability to image watermarking were noticed quite late[6]. Since then a number of systems based on this technique have been proposed [6, 7, and 8]: typically a maximum length sequence is added to the signal in the spatial domain and the watermark is detected by using the spatial cross-correlation of the sequence and the watermarked frame.

Another category of marking techniques embed watermark in transformed domain, typically one that is widely used by compression algorithms. Thus, Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) domains can be used for watermarking. Langelaar et al, remove certain high frequency DCT coefficients [9], Cox et al. modulate the largest 1000 DCT coefficients of an image with a random vector [10], Koch et al. change the quantization of the DCT coefficients and modify some of them in such a way that a certain property is verified [11], while Ruanaidh et al. modulate the DCT coefficients with bi-directional coding [12]. Techniques of this kind can be combined with exploitation of the perceptual masking properties of human visual system (HVS) [13, 14, and 15]. The basic idea here is to amplify the mark wherever the changes will be less noticeable and also to embed it in the perceptually significant components of the signal.

## 2.2 Spatial Domain Watermarking Techniques

This is the first and the most straightforward way to add a watermark in any multimedia i.e.-e, audio, video, still images. Although, watermarking in spatial domain is not very robust but still several spatial domain digital watermarking methods have been proposed. Best results are obtained when transform domain is used for watermarking.

Turner [16] proposed a method for inserting an identification string into a digital audio signal by substituting the "insignificant" bits of randomly selected audio samples with the bits of an identification code. Bits are deemed "insignificant" if their alteration is inaudible. Such a system is also appropriate for two-dimensional (2-D) data such as images, as discussed in [17]. Unfortunately, Turner's method may easily be circumvented. For example, if it is known that the algorithm only affects the least significant two bits of a word, then it is possible to randomly flip all such bits, thereby destroying any existing identification code.

Carol1ni [18] suggests adding tags, small geometric patterns-to digitized images at brightness levels that are imperceptible. While the idea of hiding a spatial watermark in an image is fundamentally sound, this scheme may be susceptible to attack by filtering and re-digitization. The fainter such watermarks are, the more susceptible they are such attacks and geometric shapes provide only a limited alphabet with which to encode information. Moreover, the scheme is not applicable to audio data and may not be robust to common geometric distortions, especially cropping.

Brassil et al. [19] propose three methods appropriate for document images in which text is common. Digital watermarks are coded by I) vertically shifting text lines, 2) horizontally shifting words, or 3) altering text features such as the vertical end lines of individual characters. Unfortunately, all three proposals are easily defeated, as discussed by the authors. Moreover, these techniques are restricted exclusively to images containing text.

Tanaka et al. [20], [21] describe several watermarking schemes that rely on embedding watermarks that resemble quantization noise. Their ideas hinge on the notion that quantization noise is typically imperceptible to viewers. Their first scheme injects a watermark into an image by using a predetermined data stream to guide level selection in a predictive quantizer. The data stream is chosen so that the resulting image looks like quantization noise. A variation on this scheme is also presented, where a watermark in the form of a dithering matrix is used to dither an image in a certain way. There are several drawbacks to these schemes. The most important is that they are susceptible to

signal processing, especially re-quantization, and geometric attacks such as cropping. Furthermore, they degrade an image in the same way that predictive coding and dithering can.

Tanaka et al. [22] also propose a watermarking method for "color-scaled picture and video sequences". This method applies the same signal transform as the Joint Photographers Expert Group (JPEG) (discrete cosine transform of8 x 8 sub blocks of an image) and embeds a watermark in the coefficient quantization module. While being compatible with existing transform coders, this scheme may be susceptible to re-quantization and filtering and is equivalent to coding the watermark in the LSB's of the transform coefficients.

In a paper [23], Macq and Quisquater briefly discuss the issue of watermarking digital images as part of a general survey on cryptography and digital television. The authors provide a description of a procedure to insert a watermark into the least significant bits of pixels located in the vicinity of image contours. Since it relies on modifications of the least significant bits, the watermark is easily destroyed. Further, their method is restricted to images, in that it seeks to insert the watermark into image regions that lie on the edge of contours. Bender et al. [24] describe two watermarking schemes. The first is a statistical method called patchwork. Patchwork randomly chooses n pairs of image points, $(a_i, b_i)$, and increases the brightness at $a_i$ by one unit while correspondingly decreasing the brightness of $b_i$. The expected value of the sum of the differences of then pairs of points is then 2n, provided certain statistical properties of the image are true.

The second method is called "texture block coding," wherein a region of random texture pattern found in the image is copied to an area of the image with similar texture. Autocorrelation is then used to recover each texture region. The most significant problem with this technique is that it is only appropriate for images that possess large areas of random texture. The technique could not be used on images of text, for example, nor is there a direct analog for audio.

Rhoads [25] describes a method that adds or subtracts small random quantities from each pixel. Addition or subtraction is determined by comparing a binary mask of L bits with the LSB of each pixel. If the LSB is equal to the corresponding mask bit, then the random quantity is added, otherwise it is subtracted. The watermark is subtracted by first computing the difference between the original and watermarked images and then by examining the sign of the difference, pixel by pixel, to determine if it corresponds to the original sequence of additions and subtractions. This method does not make use of perceptual relevance, but it is proposed that the high frequency noise be pre-filtered to provide some robustness to low pass filtering. This scheme does not consider the problem of collusion attacks.

## 2.3 DCT Domain Digital Watermarking

Several techniques can transform an image into frequency domain, such as DCT, DFT and wavelet. Each transform has its advantages and disadvantages. First of all, we discuss the DCT approach.

***Coherent Decoding Method***

I.J. Cox et al. in [12] suggest another point of view that the watermarks should be embedded in the significant perceptual component of HVS. The major reason is most compression techniques try to reduce redundancy of images. In other words, they modify the insubstantial part, such as LSB in spatial domain and high frequency in frequency domain. This principle can explain why these early works are not robust.

Moreover, [12] supposes the original non-watermarked image can be obtained in the decoding part.

Therefore, they propose a spread spectrum watermarking technique as follows:

Encoding procedure:

1. Given an NxN image D, we implement NxN DCT and obtain NxN coefficients.
2. Pick the largest n coefficients as a vector $v$.
3. Generate a vector $x$ with length n by Gaussian N(0,1). Every key owns a unique $x$.
4. Select a scaling factor $\alpha$ and construct a new vector $v'$ with

$$v'_i = v_i(1 + \alpha x_i)$$

5. Replace the original $v$ by $v'$ and perform IDCT to obtain a watermarked image D'.

The decoding procedure:

1. Perform DCT with $D'$ and obtain $v^*$ whose elements correspond to the coordinates of $v$ in $D$.

2. Calculate $x^*=v^*-v$.

3. Measure the similarity between $x^*$ and original key $x$

$$sim(x, x^*) = \frac{(x^*)^T x}{\sqrt{(x^*)^T x^*}}$$

4. Verify whether the similarity is greater than a preset threshold and determine the copyright.

[12] has shown the strong robustness of this technique, even multiple-document attacks. As a matter of fact, no previous works can achieve its ability. However, the primary disadvantage is the need to synchronize with the unmarked image in the decoder. It's not practical that a client wants to confirm copyright of an image but he cannot obtain the original image.

***Non-coherent Decoder Method***

M. Barni et al. in [27] modify some details of the previous work under the same framework. This paper indicates that if we only select the largest n coefficients as a vector v" from the watermarked image D' without the help of original image D, it's difficult to guarantee v* = v". As a result, they propose that we can zigzag the order of DCT coefficients (like JPEG) and pick the (L+1)th to (L+M)th for watermarking. To decrease the error probability, they suggest the second term in watermarked vector should be proportional to the absolute value of coefficients.

$$v'_{L+i} = v_{L+i} + \alpha \mid v_{L+i} \mid x_{L+i}$$

The watermark detection can be first done by extracting a vector v** from the (L+1)th to (L+M)th coefficients of DCT D'. Calculate the similarity between v** and the key x by the following formula and compare it with a given threshold.

$$z = \frac{x^T (v^{**})}{M}$$

Experimental results also show its robustness.

### 2.3.1 Block DCT Domain Watermarking

Because of the robustness in [12], later researchers present some modified techniques. To increase the capacity of information and detection for cropping and localized signal processing, some works suggest blocking the images and performing DCT, which is the same way as JPEG. Experimental results also show its robustness.

### 2.3.2 Block DCT Domain Watermarking

Because of the robustness in [12], later researchers present some modified techniques. To increase the capacity of information and detection for cropping and localized signal processing, some works suggest blocking the images and performing DCT, which is the same way as JPEG.

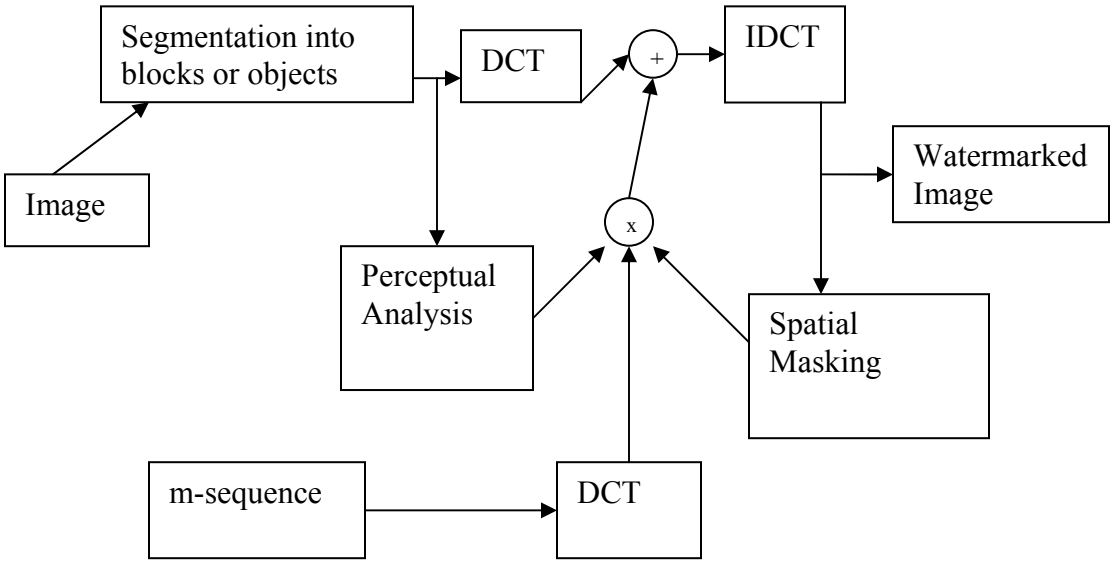



Figure 2.1: Encode procedure for DCT Domain

The above figure shows the structure of the encoder proposed by [28]. The watermark generation procedure is as follows:

1. Block the image into *Bi*, perform DCT of each block and obtain *Bi'*.
2. Calculate the frequency mask *FM* by a visual model.
3. Generate different m-sequences for each block and perform DCT to obtain *Wi*.

4. Masked coefficients in each *Bi'* are multiplied by *Wi*, i.e. *Mi= Wi *(FM*Bi')*.
5. Perform inverse DCT of *Mi* and scale it by spatial mask to control the invisibility of image.

The reason we adopt different m-sequences in this approach is that we can reduce unauthorized attack by removing cross-correlation of each block. Frequency and spatial masks assure the transparency of image. However, this method needs the original image in decoder to detect the watermark.

### 2.3.3 Image Adaptive Masking of DCT Block

In fact, the scheme in figure 1 is equal to frequency weighting in each DCT block. However, each block contains the same amount of information. To increase the capacity, in [29] the watermark can be adaptively embedded in each block according to local image characteristics. The measurement of characteristics is obtained from just noticeable differences (JND) of Waton's Model [30].

First, perform DCT of each block of original image and obtain coefficients $X_{u,v,b.}$ (u,v) is the coordinates of bth DCT block. Second, generate an independent Gaussian sequence, $w_{u,v,b}$, with N(0,1) according to a given user key. The embedding procedure is:

$$X^*_{u,v,b} = \begin{cases} X_{u,v,b} + t_{u,v,b} w_{u,v,b} & if\ X_{u,v,b} > t_{u,v,b} \\ X_{u,v,b} & otherwise \end{cases}$$

*where* $t_{u,v,b}$ *is the JND threshold calculated form Waton's Model. And the detection procedure:*

$$w^*_{u,v,b} = \frac{X_{u,v,b} - \hat{X}^*_{u,v,b}}{t_{u,v,b}} \qquad d_{WW^*} = \frac{w^* \cdot w}{\sqrt{E_{w^*} E_w}}$$

*where En=w.w and Ew*=w*.w*.*

After calculating the normalized correlation, *dww**, between original and watermarked images and comparing it with a threshold, we can determine whether the watermark is detected or not. Based on the framework, the authors also use wave let filter to replace

DCT. They obtain a much better improvement comparing with I.J. Cox in [12]. However, the common disadvantage of DCT domain techniques is the lack of resilient from geometric transform attacks.

## 2.4 DFT Domain Digital Watermarking

Fourier transform has some integral transform-based invariants properties. For instance, shifting in the spatial domain causes linear shifting in frequency domain, scaling the axes in the spatial domain causes an inverse scaling in the frequency domain, and rotating an image by an angle in the spatial domain causes the same angle in frequency domain. These properties could help watermark to resist geometric transform attacks.

### 2.4.1 Log-Polar Mapping Method

[31] proposes Fourier-Mellin Transform and log-polar mapping (LPM), In his experiments he has shown that the properties of transformed coordinate system has some useful properties. Based on the fundamental properties it was proposed that the framework can recover the rotated watermark. There were several disadvantages of the scheme. This approach requires original image while retrieving the embedded watermark. Also the proposed scheme has less ability to withstand JPEG compression and cropping attacks.

### 2.4.2 Template Based Method with Log-Polar and Log-log Mapping

The authors in [32] try to combine the good detection of rotation and scale of log-polar mapping and the scheme in [12]. They embed watermark in the DFT mid-band domain and a template in the low frequency part. Under this framework, the decoding procedure does not require the original image. It can detect the template by log-polar mapping and recover the coordinate system to extract the watermark. However, this framework cannot deal with aspect ratio attack. They propose another mapping, log-log mapping, to cope with this attack. Unfortunately, log-log mapping cannot resist flipping attack.

### 2.4.3 Template Matching Method

Instead of embedding template in the log-polar or log-log coordinate systems, [33] proposes a simpler but more robust method. Both the watermark (BCH codes) and template are embedded in the DFT domain.

The template consists of two lines (each line consists of 7 pixels) distributed from radius f1 to f 2 and angle è1 to è2. The radii and angles are generated by user key. Pixels on the lines have the values that are equal to the local average value of DFT points plus two standard deviations. The decoder can detect these two lines and extract watermarks by inverse rotation or scaling. Although this technique can perfectly extract watermark by rotation, its compression performance are not good.

### 2.4.4 Circularly Symmetric Watermark

Instead of embedding templates, [34] suggests we can embed a circularly symmetric watermark *W(k1,k2)* in DFT domain. Let *I(k1,k2*), *M(k1,k2)* denote the DFT of image and DFT magnitude respectively. The watermark consists of p rings with radii *Ri* and *Ri+1*, *i=1,2, ...p*. Each ring is distributed as follows:

$$W(r,\theta)=\begin{cases}0 & if\ \ r<R_i\ and\ r>R_{i+1}\\ \pm 1 & if\ \ R_i<r<R_{i+1}\end{cases}$$

$$r=\sqrt{k_1^2+k_2^2} \qquad \theta=\arctan(\frac{k_2}{k_1})$$

Furthermore, each ring is divided into s sectors. As a result, there are ps sectors in this watermark. However, to maintain the real value of image after inverse DFT, the watermark must be symmetric. Therefore, there are ps/2 sectors in the watermark (i.e. *W(k1,k2)=W(N-k1,N-k2)).* The watermarked image is simply to combine *M(k1,k2)* with *W(k1,k2), i.e. M'(k1,k2)= M(k1,k2) + W(k1,k2)*, and take inverse DFT of *M'(k1,k2).*

The detection procedure simply calculates the correlation, c, between *M'(k1,k2)* and *W(k1,k2)*, compares it with a given threshold, and determines whether it's a watermarked image or not.

$$c = \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} W(k_1,k_2)M'(k_1,k_2)$$

Since watermark is circularly symmetric, it is resilient to geometric attacks, such as rotation. In addition, its DFT properties help resist translation and scaling attacks.

## 2.5 Wavelet Domain Digital Watermarking

Wavelet plays a more and more important role in contemporary image processing field. It has lots of special advantages that conventional transforms, such as DCT and DFT, cannot achieve. Furthermore, it has become the fundamental transform in JPEG2000 standard.

### 2.5.1 Image Adaptive Masking of DCT Block

Authors in [29] also propose a wavelet version under the same framework. They hierarchically decompose an image into 4 layers by 9-7 orthogonal filters. The watermark is embedded as follows:

$$X^*_{u,v,l,f} = \begin{cases} X_{u,v,l,f} + t_{u,v,l,f} w_{u,v,l,f} & if\ X_{u,v,l,f} > t_{u,v,l,f} \\ X_{u,v,l,f} & otherwise \end{cases} \qquad for\ \ l = 1,2,3,4;\ \ f = 1,2,3$$

where $t_{u,v,l,f}$ is the JND threshold calculated from Waton's Model. And the detection procedure:

$$w^*_{u,v,l,f} = \frac{X_{u,v,l,f} - \hat{X}^*_{u,v,l,f}}{t_{u,v,l,f}} \qquad \rho_{WW^*}(l,f) = \frac{w_{l,f}^{\ *} \cdot w_{l,f}}{\sqrt{E_{wl,f^*} \cdot E_{wl,f}}}$$

$$\rho_{WW^*}(l) = \frac{1}{N_f}\sum_{f=1}^{N_f} \rho_{WW^*}(l,f) \qquad for \quad l = 1,2,3,4$$

$$\rho_{WW^*}(f) = \frac{1}{N_l}\sum_{l=1}^{N_l} \rho_{WW^*}(l,f) \qquad for \quad f = 1,2,3$$

$$\rho^*_{ww^*} = \max_{l,f}\{\rho_{WW^*}(l), \rho_{WW^*}(f)\}$$

where, $X_{u,v,l,f}$ is the coefficient with position $(u,v)$ in resolution level l and frequency orientation f. $X^*_{u,v,l,f}$ is the corresponding watermarked coefficients. $w_{u,v,l,f}$ is the watermark sequence. By

comparing $\tilde{n}^{*}ww^{*}$ with a given threshold, we can determine whether a watermark is detected or not.

As shown in the last three formulas, the authors calculate normalized correlations for each frequency and layer and select the maximal value from these correlations. The basic idea is to combine advantages of spatial properties with frequency ones. If a forger wants to crop the image, the higher-level detection suffers less attack than the lower-level one. On the other hand, if a forger wants to smooth the image, lower level detection can resist such an attack.

This approach performs much better than its DCT version, for example, scaling, even cropping with compression. However, it needs the original image and suffers the same fragility by rotation, the common drawback for DCT version.

### 2.5.2 Self-Similar Circularly Symmetric Watermarking

[35] suggests that both advantages of wavelet and circularly symmetric watermark can be combined together. First, a mother circularly symmetric pattern is generated with the structure in [34] and then shift and scale the mother function to obtain a family of patterns.

The image is decomposed into 4-level by Haar wavelet transform. The watermark is embedded in the level and a scaling ½ version of the same watermark is embedded in the 2nd level. The watermarked image is obtained after inverse wavelet transform. Figure 2.2 shows the whole framework. The detection procedure calculates the correlation between watermarked coefficients and watermarks itself.

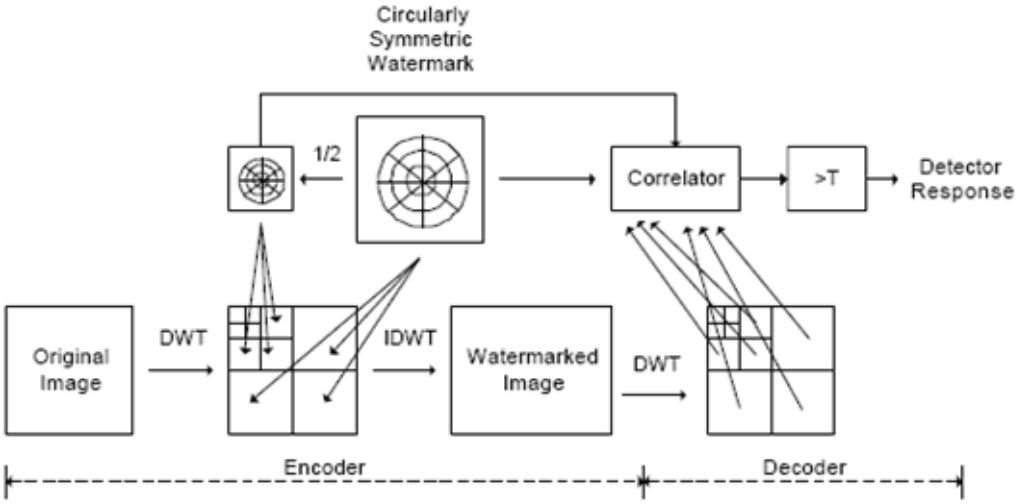



Figure 2.2: Frame work for Self-Similar Circular Symmetric Watermarking

# Chapter 3

# Fast Fourier Transform Based Watermarking Model

## 3.1 Introduction

This chapter describes an overview of the DFT and FFT invariant. The properties of FFT are also described. The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input

image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.

## 3.2 DFT in Image Processing

The DFT is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image, i.e. the image in the spatial and Fourier domain is of the same size.

For a square image of size $N \times N$, the two-dimensional DFT is given by:

$$F(k,l) = \frac{1}{N^2} \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} f(a,b)\, e^{-i2\pi\left(\frac{ka}{N}+\frac{lb}{N}\right)}$$

where *f(a,b)* is the image in the spatial domain and the exponential term is the basis function corresponding to each point *F(k,l)* in the Fourier space. The equation can be interpreted as: the value of each point *F(k,l)* is obtained by multiplying the spatial image with the corresponding base function and summing the result.

The basis functions are sine and cosine waves with increasing frequencies, i.e. *F(0,0)* represents the DC-component of the image which corresponds to the average brightness and *F(N-1,N-1)* represents the highest frequency.

In a similar way, the Fourier image can be re-transformed to the spatial domain. The inverse Fourier transform is given by:

$$f(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l)\, e^{i2\pi\left(\frac{ka}{N}+\frac{lb}{N}\right)}$$

To obtain the result for the above equations, a double sum has to be calculated for each image point. However, because the Fourier Transform is separable, it can be written as

$$F(k,l) = \frac{1}{N} \sum_{b=0}^{N-1} P(k,b)\, e^{-i2\pi \frac{lb}{N}}$$

where

$$P(k,b) = \frac{1}{N} \sum_{a=0}^{N-1} f(a,b)\ e^{-i2\pi \frac{ka}{N}}$$

Using these two formulas, the spatial domain image is first transformed into an intermediate image using N one-dimensional Fourier Transforms. This intermediate image is then transformed into the final image, again using N one-dimensional Fourier Transforms. Expressing the two-dimensional Fourier Transform in terms of a series of 2N one-dimensional transforms decreases the number of required computations.

Even with these computational savings, the ordinary one-dimensional DFT has $N^2$ complexity. This can be reduced to $N \log_2 N$ if we employ the Fast Fourier Transform (FFT) to compute the one-dimensional DFTs. This is a significant improvement, in particular for large images. There are various forms of the FFT and most of them restrict the size of the input image that may be transformed, often to $N = 2^n$ where n is an integer. The mathematical details are well described in the literature.

The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the real and imaginary part or with magnitude and phase. In image processing, often only the magnitude of the Fourier Transform is displayed, as it contains most of the information of the geometric structure of the spatial domain image. However, if we want to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, we must make sure to preserve both magnitude and phase of the Fourier image.

The Fourier domain image has a much greater range than the image in the spatial domain. Hence, to be sufficiently accurate, its values are usually calculated and stored in float values.

## 3.3 Fast Fourier Transform

An FFT (Fast Fourier Transform) is a faster version of the DFT that can be applied when the number of samples in the signal is a power of two. An FFT computation takes

approximately $N^* log_2(N)$ operations, whereas a DFT takes approximately $N2$ operations, so the FFT is significantly faster.

### 3.3.1 Applications of FFT

The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression. Several lossy image and sound compression methods employ the discrete Fourier transform: the signal is cut into short segments, each is transformed, and then the Fourier coefficients of high frequencies, which are assumed to be unnoticeable, are discarded. The decompressor computes the inverse transform based on this reduced number of Fourier coefficients.

Discrete Fourier transforms are often used to solve partial differential equations, where again the DFT is used as an approximation for the Fourier series.
The fastest known algorithms for the multiplication of very large integers use the polynomial multiplication method outlined above. Integers can be treated as the value of a polynomial evaluated specifically at the number base, with the coefficients of the polynomial corresponding to the digits in that base. After polynomial multiplication, a relatively low-complexity carry-propagation step completes the multiplication.

### 3.3.2 Watermarking in FFT Domain

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.

## 3.4 Difference between DFT and FFT

The difference between FFT and DFT is in the implementation method. FFT can reduce the amount of calculation by limiting the number of points (N in the above formula). The reduction ratio is approximately $N2$ versus $2Nxlog_2(N)$. When N is 1024, the ration is

about 1/100. This was really a landmark event in 1969 when the FFT was invented, considering the performance of computers at the time. Although the current computers are much, much and much faster than computers in 1969, importance of FFT still remains and will remain because more spectrum analysis in a limit time frame are required or real-time control system using FFT is becoming common.

## 3.5 Properties of DFT/FFT

A key property of the Fourier transform is that the multiplication of two Fourier transforms corresponds to the convolution of the associated spatial functions. This property, together with the fast Fourier transform, forms the basis for a fast convolution algorithm.

The Fourier transform can also be used to perform correlation, which is closely related to convolution. Correlation can be used to locate features within an image; in this context correlation is often called template matching.

The discrete Fourier transform is an invertible, linear transformation with denoting the set of complex numbers. In other words, for any N > 0, an N-dimensional complex vector has a DFT and an IDFT which are in turn N-dimensional complex vectors.

This orthogonality condition can be used to derive the formula for the IDFT from the definition of the DFT.

Equation indicates that the Fourier transform of an image can be complex. This is illustrated below in Figures 4a-c. Figure 4a shows the original image $a[m,n]$, Figure 4b the magnitude in a scaled form as $log(|A(\Omega,\Psi)|)$, and Figure 4c the phase $\Phi(\Omega,\Psi)$.

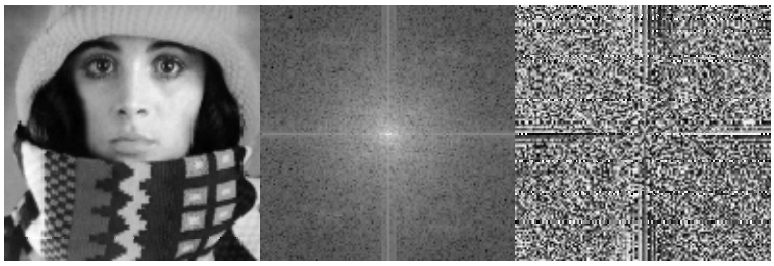


Figure 3.1: (a) (b) (c) **Original log(|A(Ω,Ψ)|) Φ(Ω,Ψ)**

Both the magnitude and the phase functions are necessary for the complete reconstruction of an image from its Fourier transform. Figure 3.1a shows what happens when Figure 3.1a is restored solely on the basis of the magnitude information and Figure 3.1b shows what happens when Figure 3.1a is restored solely on the basis of the phase information. Neither the magnitude information nor the phase information is sufficient to restore the image. The magnitude-only image (Figure 3.1a) is unrecognizable and has severe dynamic range problems. The phase-only image (Figure 3.1b) is barely recognizable, that is, severely degraded in quality.

# Chapter 4

# Design and Implementation of the Proposed Scheme

Digital watermarks are substantially imperceptible signals embedded into a host signal that may be used for broadcast monitoring, proof of ownership, prevention of unauthorized usage of copyrighted content, authentication, tamper-evidence, audience measurement, and transaction tracking.

## 4.1 Overview

Watermarking techniques described in the literature include manipulating the least significant bits of the host signal in time or frequency domains, spread spectrum, phase, amplitude or frequency modulation techniques, and insertion of watermarks.

Watermarking scheme is based on the technique used by [36]. Frequency domain is being used in this case. Frequency domain serves as a strong channel for transmitting information. A circular watermarking method is used in order to create a digital watermark as was used by [36]. Most of the watermarking schemes are not resistant to number of attacks. However, circularly embedding the watermark provides a resistant scheme to such attacks.

## 4.2 Proposed Scheme

In this scheme first the Fourier Transform of the image is taken to calculate the magnitude of two-dimensional image. Middle frequencies are selected to minimize the distortion because if a change occurs in the Low or High frequencies it brings changes in the image that is visible to the human eye.

The data is encoded, in the frequency domain, along a ring of middle frequencies. The middle frequencies are used to minimize noticeable distortion in the image quality. The human eye is able to catch modifications to the lower frequencies since most of the image's frequency content is located in this area. Modifying high frequencies can cause a multitude of local distortions along edges in the image. The frequencies used are specified by setting the radius of the ring (r). A value of 100 for a 256x256 image works effectively. The overall working of the proposed scheme is shown in the Figure 4.1.
Computational complexity of DWT is more compared to DCT. It only takes 54 multiplications to compute DCT for a block of 8x8, unlike wavelet calculation depends upon the length of the filter used, which is at least 1 multiplication per coefficient.

## 4.3 Steps of the Algorithm

Mainly, there are two parts of the algorithm.

Insertion of the Watermark

Extraction of the Watermark

Details of the each part are mentioned below:

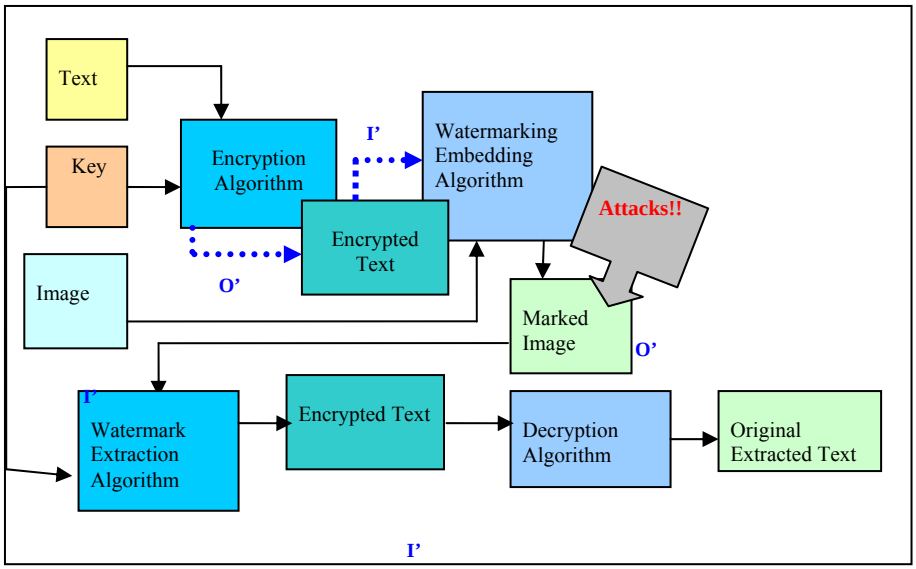



Figure 4.1 Overall Processing of Proposed Scheme

### 4.3.1 Insertion of Watermark

Insertion of watermark is further divided into phases. Encrypted message 'E' will be inserted into the image. The process involves a series of steps. For an image 'I' of size MxN denoted as I [M, N], let 'r' be the radius to create a circular Data Ring. Let 'α' be the scaling factor. The values of α and r are kept constant. The string of encrypted message will be converted into binary equivalent. Let E be a set of encrypted text having length L1. Let E be the input to convert the text into binary form. The output is a set of bits of length L2. The binary values of the encrypted message are stored in a single row vector which is padded with zeros to meet the circumference need of the circular data ring. The value of r is used to define the range of the extending the padding values in the vector.

$$C' = 2*(r-4)-L2 \qquad (2)$$

The row vector with binary values of the encrypted data is padded with zeros having length equal to the row dimension of the image. The values of the row vector are scaled by α factor. The vector is multiplied by a scalar value (α) such that the data values will not be "washed" out by the magnitude of the FFT and easily identified while reconstructing the image.

$$\text{datamark} = \alpha * [\text{binarytext } C'] \quad (3)$$

Another matrix ring[M,N] all filled with zeros is created. This ring matrix is a container to insert datamarked values in a ring form. The image is supposed to be divided into four quadrants with a center C. To find the coordinates where the datamark will be added, first the center C will be determined. To find the center the following equation will be used:

$$C = N/2 + 1 \quad \text{where } M=N, \text{ M and N are even numbers.} \quad (4)$$

Each datamark will be inserted alternatively in the two quadrants to enhance the retrieval capability of the algorithm. However if unique data is embedded in each quadrant it is possible to increase the data hiding capacity. The coordinates are determined using the equation of the circle.

C is considered to be a center point. From the equation of circle:

$$r^2 = (x-x_0)^2 - (y-y_0)^2 \quad (5)$$

X' is calculated from

$$x = X' + C \quad \text{where } C = N/2+1 \quad (6)$$

Y' is calculated from equation (5)

$$(y-y_0)^2 = r^2 - (x-x_0)^2 \quad (7)$$

$$(y-y_0) = \text{sqrt}(r^2 - (x-x_0)^2) \quad (8)$$

Interpreting the equation (6) and equation (7)

$$Y' = y, \; y_0 = 0 \text{ and } x = X' + C, \; x_0 = 0 \quad (9)$$

So the equation (7) becomes:

$$(Y'-0) = \text{sqrt}((r^2-(x-C-0)^2) \quad (10)$$

Finally,

$$Y' = \text{sqrt}((r^2-(x-C)^2) \quad (11)$$

The location for storing the datamark is (x,C+Y') , (C-X',C-Y'), (x,C-Y') and (C-X',C+Y')

Thus the 'ring' matrix is updated with data bits which are inserted from the 'datamark' row vector. The matrix is divided into four quadrants so that each data bit is inserted into two quadrants to improve the performance of extraction process.

Each data bit from the scaled row vector is inserted into two quadrants. The quadrants I and IV have the same bits whereas II and III have the same bits. Figure 4.2 shows the insertion process.

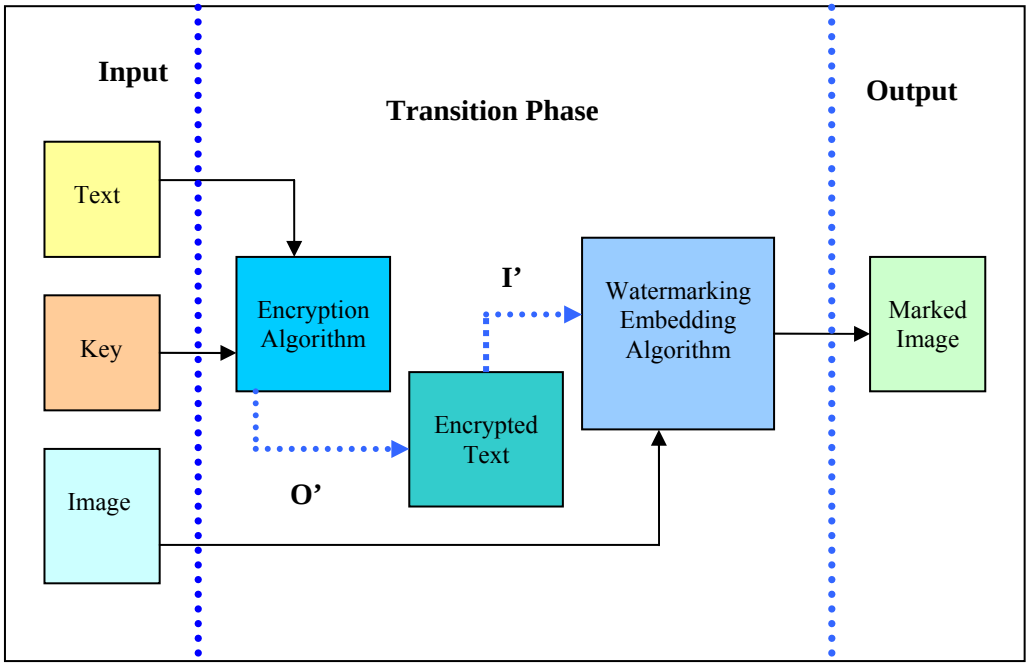



Figure 4.2: Insertion of Watermarking Scheme

4.3.1.1 Computation of Fast Fourier Transform

Frequency in image processing is a term to define the brightness of colors in an image. Fourier transform defines a function that is the sum of all increasing frequencies. The low frequencies have larger magnitude and therefore possess more information about the

image. Similarly, the higher frequencies contain information related to the sharp edges. Any change in these two extreme frequencies will be visible. Middle frequencies are considered to be best candidate for inserting data as human eye is unable to catch the modifications. The levels are shown in Figure 4.3.

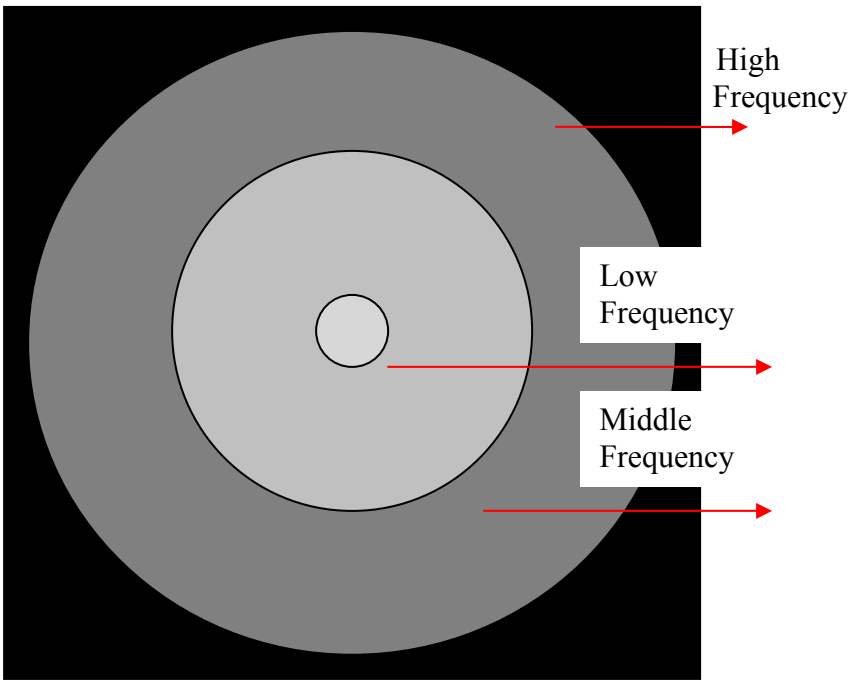



Figure 4.3: Three Levels of Frequencies

Fast Fourier Transform processing involves the following steps:

- Calculate the FFT of the image I[X, Y].
- Shift the frequencies to the center.
- Calculate the magnitude of FFT.
- Calculate the phase angle θ.
- Add the magnitude and ring matrix.
- Loop for the reducing the magnitude points where data bit are zero.

```
for i=1:N
   for j=1:M
      if(ring(i ,j) = = 0)
```

```
mag_img_marked (i ,j) = mag_img_marked (i , j)-
alpha/10;
end
```

For each ($i^{th}$, $j^{th}$) pixel value in the ring matrix, where a data bit "0" has been encoded, it is reduced by alpha/10 in order to minimize the chance that the value would be interpreted as a "1" (i.e. the magnitude of the FFT was initially large; the extractor interprets the pixel as logic "1").

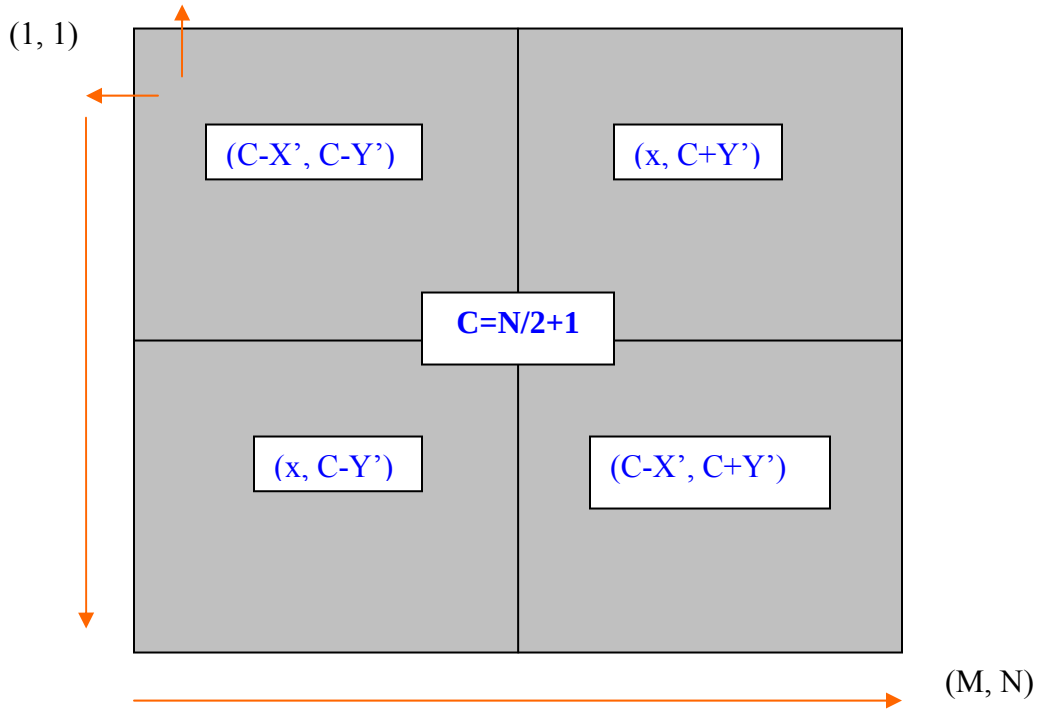



Figure 4.4: Location for insertion of the embedded text

At the pixels where a data bit "0" has been encoded, the matrix value is reduced by alpha/10 in order to minimize the chance that the value would be interpreted as a "1" (i.e. the magnitude of the FFT was initially large; the extractor interprets the pixel as logic "1").

4.3.1.2 Generation of embedded Image

For final generation of the watermarked image, each ($i^{th}$, $j^{th}$) element of the 'mag_img_marked' matrix is multiplied with the phase of the image so that the image in the original phase is obtained.

```
for i=1:N
```

```
    for j=1:M

image_marked_freq(i,j)=mag_image_marked(i,j)*(cos(phas
e_image(i,j))+sqrt(-1)*sin(phase_image(i,j)));
  end
```

For the new magnitude the frequency is shifted to the center and inverse fast Fourier transform is computed. The new image is generated with data embedded into it.

### 4.3.2 Extraction of Embedded Data

Extraction of the embedded data is also divided into phases. The process of extraction starts by reading the input image that is marked with hidden encrypted message and applies the secret key to decrypt the hidden text. The detail of these phases is as follows:

4.3.2.1 Intialization

- Initialize the matrix used in storing the extracted magnitude coefficients.

```
raw_data = [];
```

4.3.2.2 Computation of FFT

- Compute the magnitude of the FFT of the image.
- Calculate the size of the image.

```
[N,M] = size(f);
     C = N/2+1;
```

- Extract the magnitude coefficients along the circle.

```
for x = C+1:C+r-4
   X' = x - C;
   Y' = round(sqrt(r^2-(x-C)^2));
   raw_data(1,x-C) = (f(x,C+Y') + f(C-X',C-Y'))/2;
   raw_data(2,x-C) = (f(x,C-Y') + f(C-X',C+Y'))/2;
   end
```

- Subtract the alpha scalar.

```
raw_data = raw_data - (alpha);
```

- Convert to estimated data bits. The data bit is considered as 1 if the magnitude value is greater than or equal to zero else the data bit is 0.

```
bin_data = raw_data>=0;
```

- For robustness, the data bits were embedded in two quadrants of the FFT. This loop averages the bits to estimate the true data bit and reduce error.

```
k=1;
for x = 1:length(bin_data(1,:)),
    avg_bin_data(k) = bin_data(1,x);
    avg_bin_data(k+1) = bin_data(2,x);
    k = k + 2;
end
```

4.3.2.3 Decryption and Generation of the Text

- Convert the bits to a text message.

```
out_text=bin2text(avg_bin_data);
```

- The key will be used here to decrypt the output text. Actual message will be displayed. If an inappropriate key is entered, garbage data values will be shown. Overall extraction scheme is described below:

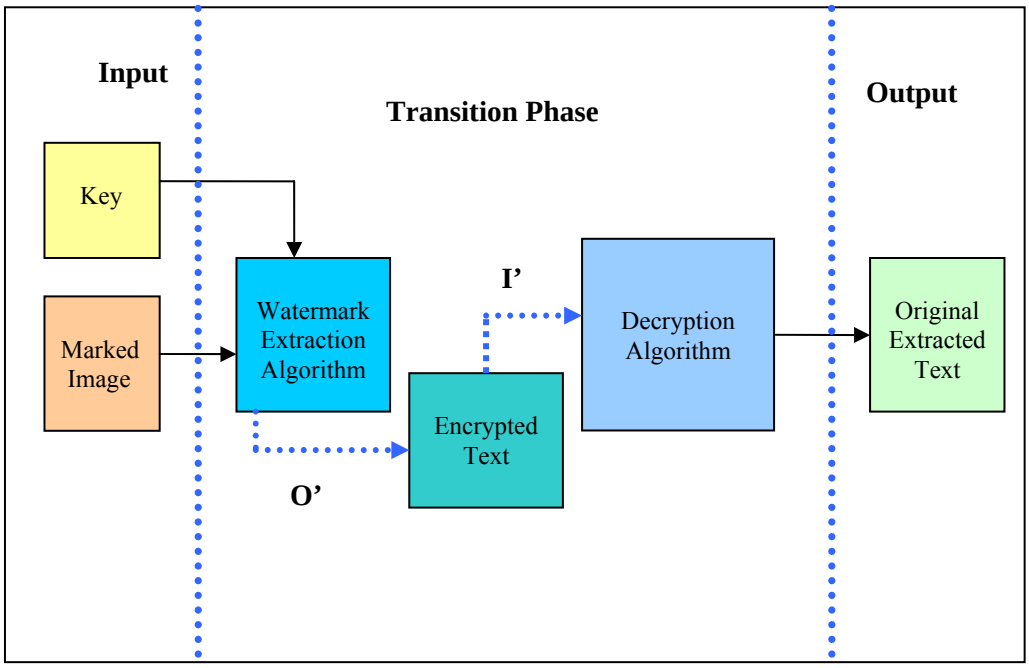



Figure 4.5: Extraction of Watermarking Scheme

## 4.4 Pseudo code

**Insertion of Watermark**

Insertion of watermark is further divided into phases. Each phase is described in detail.

-Encrypt (M, K) = E

-Creation of Data Mark:

```
bin_mesg ← Text2binary(E)
Padding(bin_mesg) by factor C'
                C'=2*(r-4)-L2          where L2 is
the length of the binary text. {L2 <100}
      bin_mesg←zeros(bin_mesg)
watermark = α*bin_mesg
      for k←1
```

For an image of size MxN center is determined as:

```
                          C=N/2 + 1
where M=N, M and N are in even numbers.
            for x=C+1:C+r-4
                    X'=x-C;
                Y'=round(sqrt(r^2-(x-C)^2));
                ring (x,C+B)=watermark(k)
                ring (x,C+B)=watermark(k)
                ring (x,C+B)=watermark(k+1)
                ring (x,C+B)=watermark(k+1)
      k=k+2
```

-Calculate the FFT of the image.

-Add the magnitude and Ring.

-Loop for the reducing the magnitude points where data bit are zero.

```
for i=1:N
   for j=1:M
      if(ring(i ,j) = = 0)
mag_img_marked (i ,j) = mag_img_marked (i , j)-
alpha/10;
end
```

**Generation of embedded Image**

```
for i=1:N
  for j=1:M
```

```
    imagem_marked_freq(i,j)=
mag_imagem_marked(i,j)*(cos(phase_imagem(i,j))+sqrt(-
1)*sin(phase_imagem(i,j)));

  end
```

- For the new magnitude shift the frequency to the center.
- Compute the Inverse Fast Fourier Transform
- Write the image in a file.

**Extraction of the Image**

Read the input image that is marked with hidden encrypted message and apply the secret key to decrypt the hidden text.

**Intialization**

-Initialize the matrix used in storing the extracted magnitude coefficients.

```
raw_data = [];
```

**Computation of FFT**

-Compute the magnitude of the FFT of the image.

-Calculate the size of the image.

-Extract the magnitude coefficients along the circle.

```
for x = C+1:C+r-4
   X' = x - C;
   Y' = round(sqrt(r^2-(x-C)^2));
   raw_data(1,x-C) = (f(x,C+Y') + f(C-X',C-Y'))/2;
   raw_data(2,x-C) = (f(x,C-Y') + f(C-X',C+Y'))/2;
   end
```

-Subtract the alpha scalar.

```
raw_data = raw_data - (alpha);
databit_data = raw_data>=0; converting into bits
```

This loop averages the bits to estimate the true data bit and reduce error.

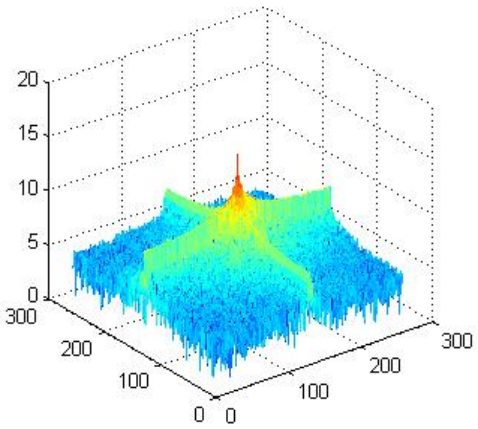
```
k=1;
for x = 1:length(bin_data(1,:)),
    avg_bin_data(k) = bin_data(1,x);
    avg_bin_data(k+1) = bin_data(2,x);
```

```
    k = k + 2;
end
out_text=bin2text(avg_bin_data);
original_text=decrypt(e_text,Key);
```

## 4.5 Selection of Fourier Transform

The theory behind Fourier Transform is that, it is possible to form a function f (a) as a summation of a series of sine and cosine terms of increasing frequency.

The term frequency in digital image usually refers to the variation in brightness or color across the image, i.e. it is a function of spatial coordinates, rather than time. For example, if an image represented in frequency space has high frequencies then it means that the image has sharp edges or details. Low frequency content defines the course image. To minimize distortion, then, the data should be inserted in the middle frequencies. The human eye is able to catch modifications to the lower frequencies since most of the image's frequency content is located in this area. Modifying high frequencies can cause a multitude of local distortions along the image's sharp edges. The Figure 4.5 shows the Fast Fourier Transform of the original image as well as for the embedded image.



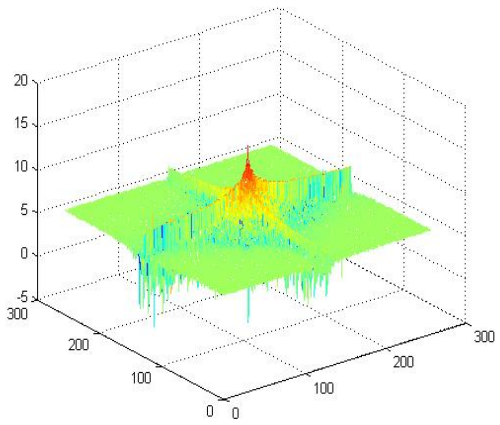




Figure 4.6: (a) FFT of the original Image (b) FFT of the embedded Image

# Chapter 5

# Spatial Domain Watermarking Scheme

As human eye perceives the image in way that it is unable to attune to small variations in color, so the image processing which adjusts the small differences between adjacent pixels will leave a resulting effect which is unnoticeable. Thus LSB technique takes this advantage and replaces the N least signification bits of each pixel in an unnoticeable way to the human eye.

## 5.1 Overview

As the least-significant bits of an 8-bit grayscale image encode the most minor variations in pixel color, they can be replaced with informational bits without altering the image in a perceptible way (provided that the number of bits replaced at each pixel is kept reasonably low). The LSB technique proves to be a rather well-rounded method and lends itself to a variety of information-hiding applications. Its first principle benefit is sheer volume. Since each pixel serves as a data carrier, a large quantity of imbedded information can be included in even the most modestly-sized images, such as the

256x256 pixel images used in this discussion. LSB also allows for the embedding of an interesting variety of hidden information. Where many other techniques only allow for the embedding of coded text or simple shapes, LSB can also allow for the hiding of photographic images and even audio recordings. This investigation will concentrate on embedding photographic images and encoded text using the LSB-replacement technique.

## 5.2 Proposed Scheme

The proposed scheme for LSB based method in spatial domain includes insertion of the watermark and its extraction. The process involves an input of a secret key which allows the user to extract the embedded information in a secure way. The key will not be used to encrypt any data that is embedded into the image however it will be used to identify the places (pixels) in the image to embed the binary form of the data. Suppose a grayscale value of a pixel is 123 and its binary equivalent is 0111 1011. Now if we want to add 1 based on storing the message bit then grayscale value will become 124. Its binary equivalent will be 0111 1100. Thus, the data is stored in the least significant bits. The following figure shows the working of LSB based algorithm:

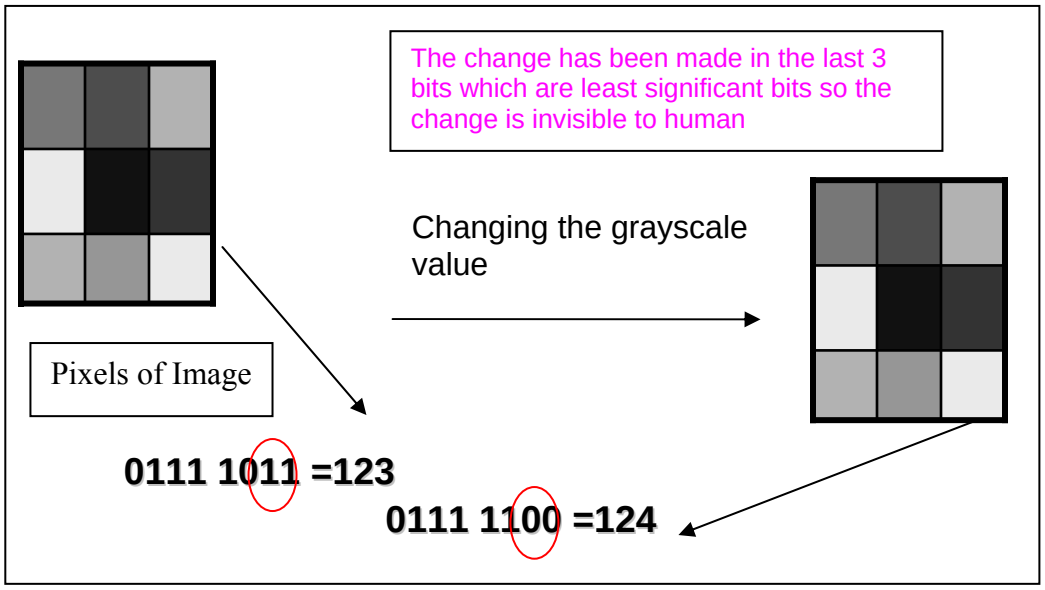



Figure 5.1: Working of LSB methods in an Image.

### 5.2.1 Working of the Proposed Scheme

The proposed scheme takes three inputs i.e. the image in which the data will be embedded, the key which will help to securely identify the place within the image and the text that will be embedded into the image. The reversed process required two inputs i.e. the watermarked image and the key to extract the hidden data from the points in the image. The working of the LSB based method is simpler as it requires the manipulation at the pixel level. The gray-level intensity however, will not be affected and thus it will not visible to the human eye. The scheme is very useful when large amount of data is needed and requires transmission over noiseless channels. The working is described in the following figure:

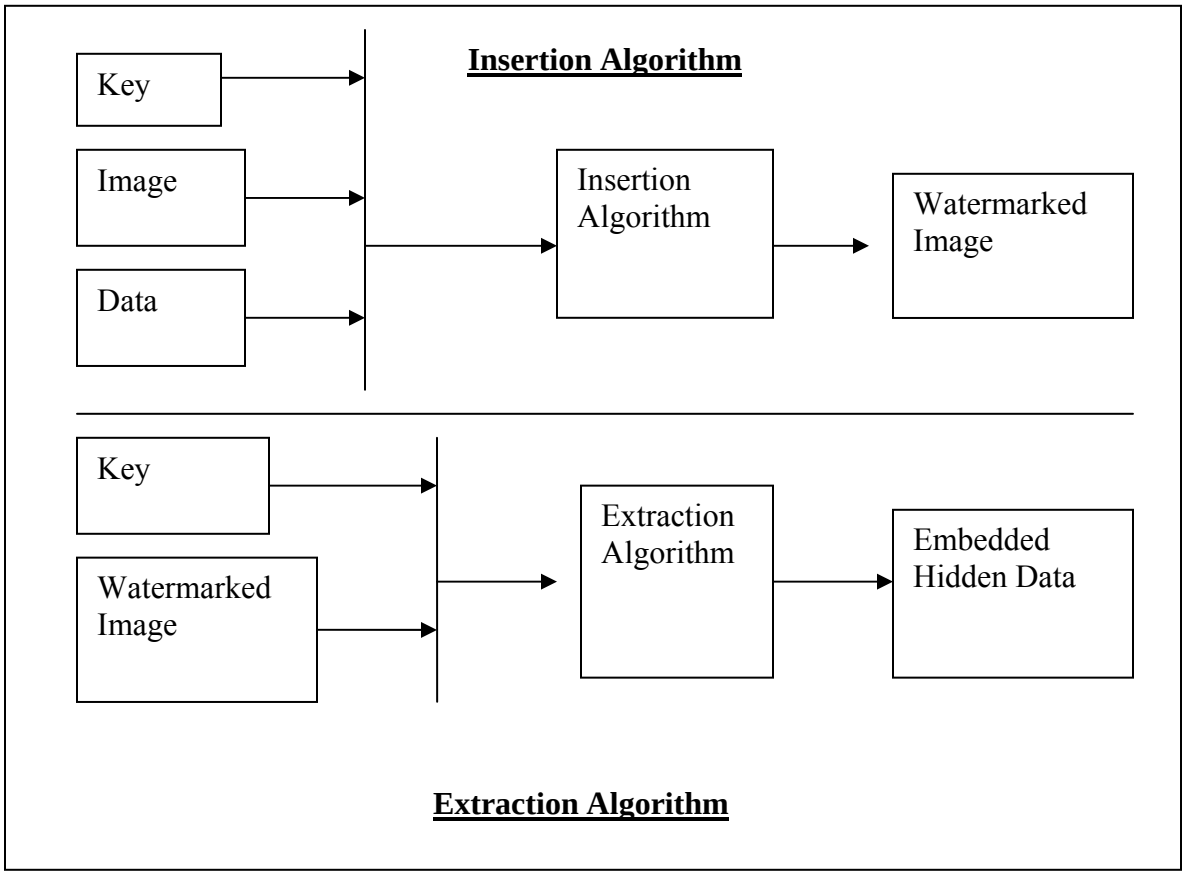



Figure 5.2: Working of LSB methods in an Image.

## 5.3 Steps of Algorithm

Mainly, there are three parts of the algorithm.

1. Generation of Key
2. Insertion of the Watermark
3. Extraction of the Watermark

Details of the each part are mentioned below:

### 5.3.1 Generation of Key

The key will be taken as an input from the user. It will be used to generate the coordinate pairs in the image for the insertion of the watermark data.

```
keyb = key(end:-1:1);
rows = cumsum(double(key));
columns = cumsum(double(keyb));    % Coord pairs for KEY
(rows,columns)
```

### 5.3.2 Insertion of Watermark

Insertion of watermark is further divided into the following steps:

-Create the data mark.

-Create a matrix of zero bits with size equal to that of the image. The key will be used to determine hiding points. For this, marked the hiding points with 1.

-Create an index of the points where data mark is 1.

-Size of the message is up to 1000.

Calculate the ASCII equivalent and represent them into binary form.

```
B = pic1(:,:,1);    [piclngth pichght] = size(B);   % Choose
the first page.
      dim1 = piclngth-2;   dim2 = pichght-3;
      idx = find(A==1);
      for vv = 1:1000
          for uu = 1:7
              if msgmat(vv,uu)==1;
                 if remainder (B(idx(uu+7*(vv-1))),2)==0
      then
                    Img(idx(uu+7*(vv-1))) =
      Img(idx(uu+7*(vv-1)))+1;
```

```
            end
        elseif remainder(B(idx(uu+7*(vv-1))),2)==1
then
              Img(idx(uu+7*(vv-1))) =
Img(idx(uu+7*(vv-1)))-1;
        end
    end
end
newImg = pic1;   newImg(:,:,1) = Img;
```

### 5.3.3 Extraction of the Image

Similarly for extraction the key will be used to find the hiding points. The index is again used to indicate the position where the data value is 1.

```
idx = find(A==1);   msgmat = zeros(1000,7);

for vv = 1:1000  % This is the decoder.
    for uu = 1:7
        if rem(B(idx(uu+7*(vv-1))),2)==1
           msgmat(vv,uu) = 1;
        end
    end
end
```

## 5.4 Pseudo Code

The pseudo code of the proposed LSB based method is described below:

### 5.4.1 Key Generation

The key will be used to extract the data embedded into the image. The key will be stored in an array and the cumulative sum both row and column will be used to identify the location of the storage in the image.

```
keyb = key(end:-1:1);
rows = cumsum(double(key));
columns = cumsum(double(keyb));   % Coord pairs for KEY
(rows,columns)
```

### 5.4.2 Insertion of Watermark

Insertion of watermark is further divided into the following steps:

- Create the data mark.
- Create a matrix of zero bits with size equal to that of the image. The key will be used to determine hiding points. For this, marked the hiding points with 1.
- Create an index of the points where data mark is 1.
- Size of the message is up to 1000.
- Calculate the ASCII equivalent and represent them into binary form.

The Index values generated on the base of key, the values are inserted into the image. Only the values which are equivalent to 1 are stored. For that, there are two choices if the value of the image is odd then it is incremented by one. Otherwise the value is decremented by 1.

The actual working of the algorithm is that grayscale images are ranged from 0-255 which represents the intensities. When the equivalent binary value of each pixel is calculated there will be 8-bit representation of it.
The bitmap image having intensities ranges from 0-255 when used to store the message in its noise.

### 5.4.3 Extraction of the Image

Similarly for extraction the key will be used to find the hiding points. The index is again used to indicate the position where the data value is 1.
The values in the *msgmat* will be converted back from binary to decimal and the ASCII equivalent alphabets will be the output of the hidden message extracted from the image.

## 5.5 Advantages and Disadvantages of the LSB Scheme

LSB based scheme is a smarter and simplest technique in the start of watermarking technique. The spatial domain watermarking interprets the pixels of the images. The noise is stored in the images and this noise is store as a room for insertion of data into it. Thus, noise is replaced with useful information leaving no effect on HVS. A large

amount of data, therefore, can be stored. The two main advantages of LSB scheme is that, a large amount of data can be stored easily and visually it is not visible to human eye.

The main disadvantage of LSB scheme is that, if noise is added somewhat while in communication or through any other way, it will destroy the data stored. So the information retrieved will not be correct. Also, LSB type of watermarking is vulnerable to attacks that include geometric attacks, noise attacks, jpeg compression and many other attacks.

# Chapter 6

# Experimental Results

This chapter will describe the effectiveness of the proposed approach against a variety of attacks. In experiments, the watermarked image was attacked using MATLAB with JPEG compression, Gaussian noise, blurring, resizing, histogram equalization, contrast adjustment, scaling, and cropping.

## 6.1 Comparison of Proposed Scheme with Past Techniques

More specifically, the datamark is created by scaling the binary data vector by a scalar input value $\alpha$. Alpha ranges from 10,000 to 25,000. From research of [36], a value of 12,000 balances the design trade-off of image distortion (alpha too big) and algorithm robustness. The scaled vector is applied along the circumference of the ring. The value of $\alpha$ and r were used to generate the pseudorandom keys [36]. However, here the values were kept constant as best results against a number of attacks were obtained by defining certain definite values.

The data bit is placed in two quadrants of the FFT to improve performance in the extraction process (the extracted bits are then averaged together.)

The operation outlined above is performed on the magnitude of the FFT of the image. To reconstruct, the phase of the image is multiplied with the newly modified magnitude, and the IFFT is computed.

### 6.1.1 Comparison of Proposed Scheme with DCT based Watermarking

Bors and Pitas [38] proposed an algorithm based on DCT that used 8x8 blocks. The algorithm is split into two main sections. Parameters are first used to find block locations. Secondly, parameters for constraint are imposed on the DCT coefficients. Zhao and Koch [39] use a similar technique but instead of calculating which blocks to use for transformation, blocks are selected at random and then quantized. This method is susceptible to geometric distortion. The proposed scheme is robust against a number of known geometric attacks.

### 6.1.2 Comparison of Proposed Scheme with other FFT based Watermarking

Some of the latest advanced techniques such as log-polar mapping along with FFT require the original image for the extraction purposes [40]. The proposed method however, does not require the original image thus found to be more robust and secured. Another method adopted in [41], where for minimal loss in image fidelity, the watermark is embedded in those DFT coefficients with highest magnitudes. The watermark detection is achieved without use of the original image by computing a similarity measure between the input watermark and the DFT coefficients of the attack image. However, this approach is less secure as compared to the proposed approach as the watermark is encrypted with security key.

### 6.1.3 Comparison of Proposed Scheme with DWT based Watermarking

The scheme proposed in [42] in DWT is vulnerable to JPEG compression, however the scheme presented here is robust against this attack and gives very acceptable results and bit error rate is negligible. Another scheme proposed in [43] has shown highly good results in JPEG compression, however the scheme was not simulated against other significant attacks like geometric and noise attacks. All these attacks have been discussed in the proposed scheme.

## 6.2 Selecting the parameters for the algorithm

8-bit gray images of size 256x256 are selected. For these images the value of r=100 and the value of alpha has been selected as 10000 to embed maximum characters. It has been observed that best results are obtained if the value of alpha is selected as α = 10000 or α = 12000. For image retrieval out of attacked image the results are obtained satisfactory at α =10000.

## 6.3 The quality of Watermarked Images

The phrase peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

The PSNR is most commonly used as a measure of quality of reconstruction in image compression etc. It is most easily defined via the mean squared error (MSE) which for two m×n monochrome images I and K where one of the images is considered a noisy approximation of the other is defined as:

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} \| I(i,j) - K(i,j) \|^2$$

$$\text{The PSNR} = 10.\log_{10}\left(\frac{Max_I^2}{MSE}\right) = 20.\log_{10}\left(\frac{Max_I}{\sqrt{MSE}}\right)$$

Here, MaxI is the maximum pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with B bits per sample, maximum possible value of MaxI is 2B-1.

For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better.

In the proposed scheme the PSNR of the image is calculated. The two images are shown in the figure and the result of PSNR is shown in the table:

The PSNR, a commonly used measure for determining the quality of processed image, is used to determine the quality of the watermarked image. Experiment was conducted on a set of images and the result has been shown in the following table:

| No. | No. of Bits Inserted | PSNR in dB |
|---|---|---|
| 1. | 5 characters | ~29.998 |
| 2. | 10 characters | ~29.965 |
| 3. | 15 characters | ~29.958 |
| 4. | 20 characters | ~29.932 |
| 5. | 25 characters | ~29.930 |
| 6. | 30 characters | ~29.927 |

Table 1: PNSR of the watermarked images

The quality of the watermarked image is calculated based on the out comes of the peak signal-to-noise ratio. Experiment conducted on 20 images shows that the quality of the image lies in the acceptable range.

## 6.4 Robustness to Image Cropping

This is another important attack against which a number of watermarking schemes failed. Cropping involves removing of pixels in horizontal and/or vertical direction. The algorithm has been applied on a set of images. The result is shown in the figure below:

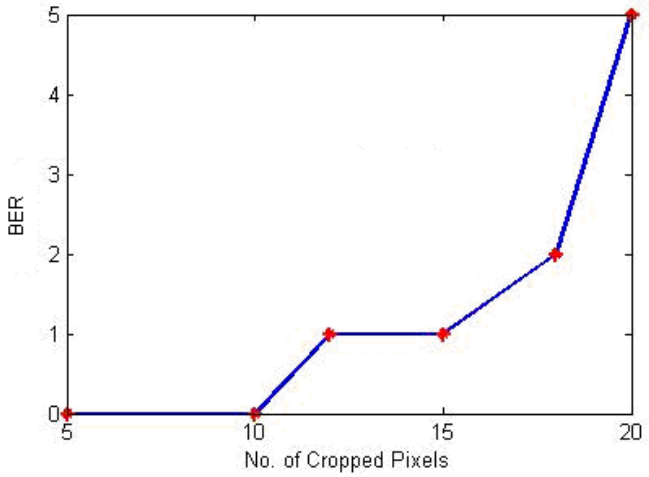



Figure 6.1: Cropping of image up to 10 pixels show a BER=0.

## 6.5 Robustness to Noise

Noise attack is one of the important attack against which the watermarking schemes are found vulnerable. Various noise attacks are applied to the watermarked image and the results are found satisfactory.

### 6.5.1 Additive White Gaussian Noise

Additive White Gaussian Noise channel adds wideband or white noise with a constant spectral density. The proposed scheme is exposed to AWGN with different values. These noise values are scaled with a parameter n. The graphical representation of the values and BER is shown in Figure.

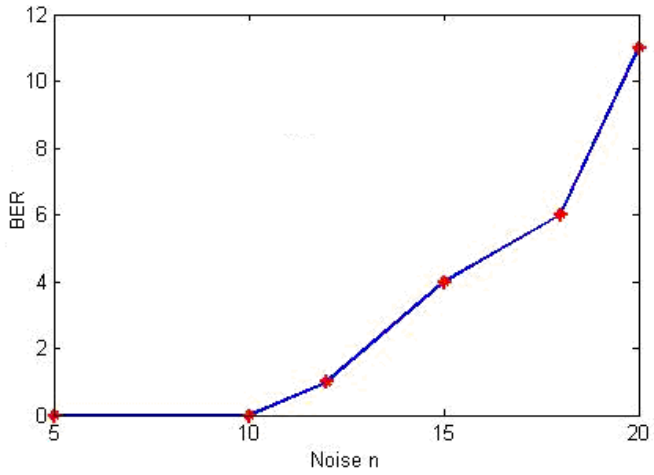



Figure 6.2: Graphical representation of adding AWGN noise to watermarked image shows BER=0 up to n=10.

### 6.4.1 Poisson and Speckle Noise

Speckle and Poisson noise can destroy the embedded data in the image. Against the Poisson noise the BER = 2. The same result has been observed when applied on set of 20 images. Speckle noise is another possible attack on the image. Speckle is an arbitrary,

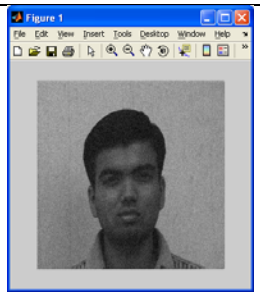
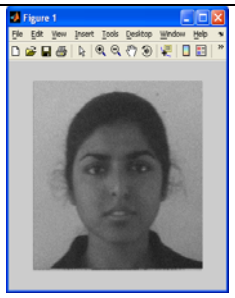
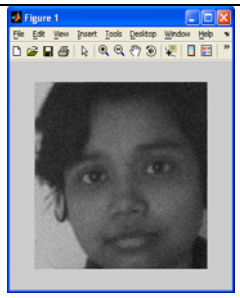
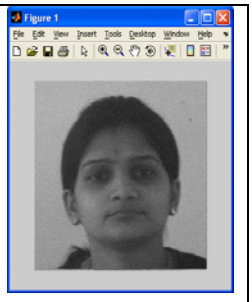
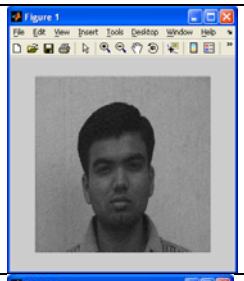
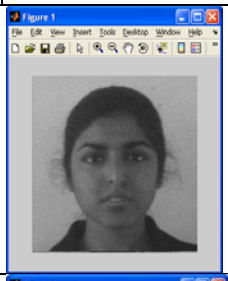
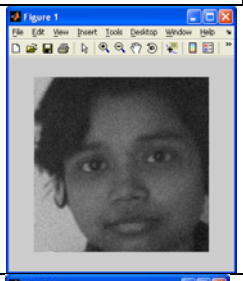
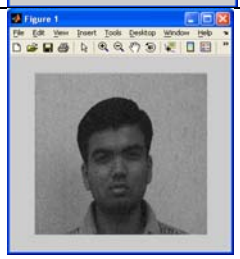
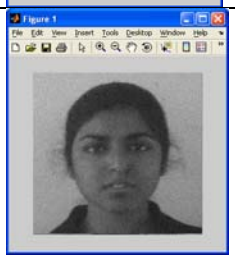
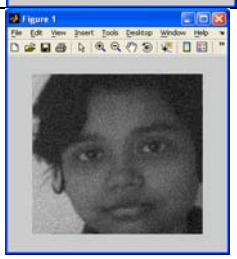
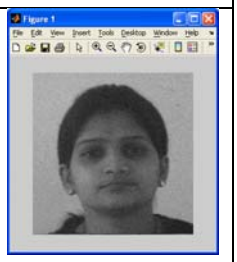
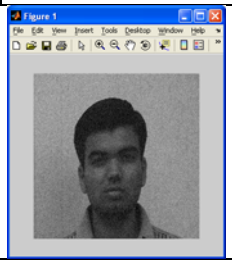
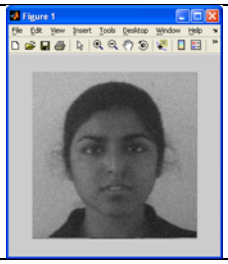
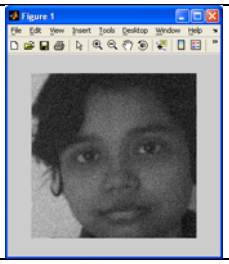
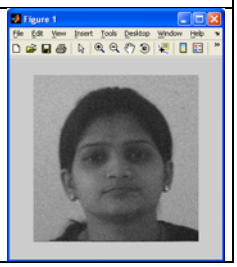
| | | | | |
|---|---|---|---|---|
| Noise error = 5<br>bit error =0 |  |  |  |  |
| Noise = 10<br>bit error =1 |  |  |  |  |
| Noise= 15<br>bit error=3-6 |  |  |  |  |
| Noise = 20<br>bit error =9 |  |  |  |  |

Figure 6.3: Images adding AWGN noise to watermarked image shows BER=0 up to n=10.

### 6.4.1 Poisson and Speckle Noise

Speckle and Poisson noise can destroy the embedded data in the image. Against the Poisson noise the BER = 2. The same result has been observed when applied on set of 20 images. Speckle noise is another possible attack on the image. Speckle is an arbitrary, deterministic, interference pattern in an image. The graphical representation of speckle noise when applied to the watermarked image is presented in the figure below:

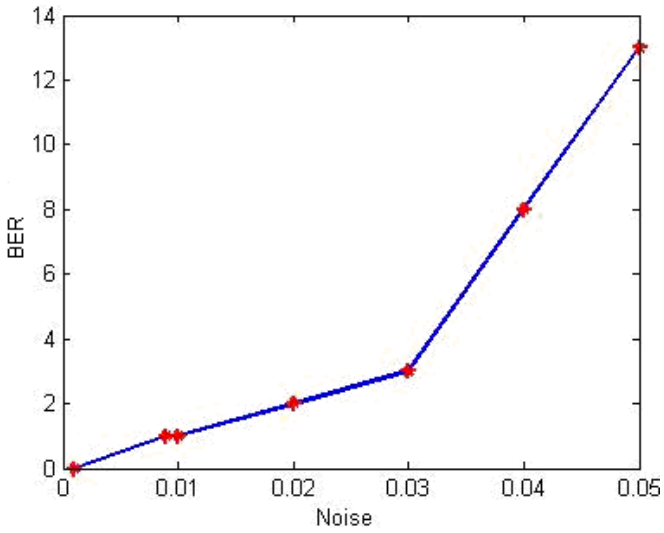



Figure 6.4: Graphical representation of adding speckle noise to Watermarked image and corresponding BER

### 6.4.2 Salt & Pepper Noise

In salt & pepper noise the random pixels are set to black and white throughout the image. The graphical representation of the salt & pepper noise attack on the watermarked image is shown in Figure 5.

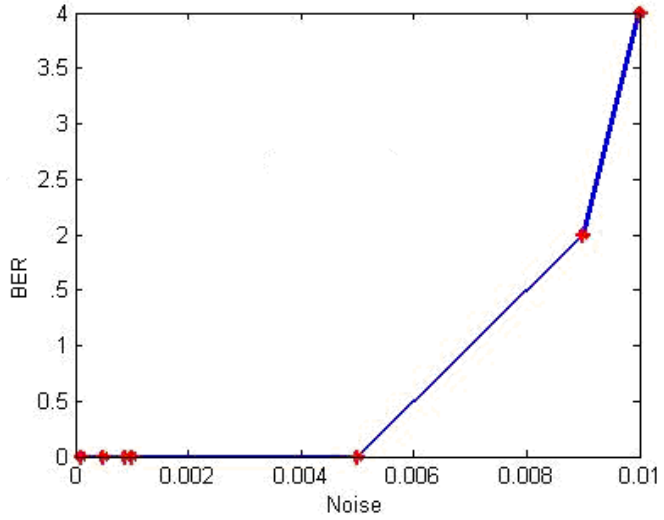



Figure 6.5: Graphical representation of adding salt & pepper noise to Watermarked image and corresponding BER

### 6.4.3 Robustness to Histogram Equalization

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram.
This method usually increases the local contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast without affecting the global contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. To illustrate the utility of histogram equalization, consider the watermarked image which shows an 8-bit grayscale image. The histogram confirms that this image has poor dynamic range. (Note that we can view this histogram as a description of pixel probability densities by simply scaling the vertical axis by the total number of image pixels and normalizing the horizontal axis using the number of intensity density levels (*i.e.* 256). However, the shape of the distribution will be the same in either case.)

In order to improve the contrast of this image, without affecting the structure (*i.e.* geometry) of the information contained therein, we can apply the histogram equalization operator. The resulting image is and its histogram is shown. Note that the histogram is not flat (as in the examples from the continuous case) but that the dynamic range and contrast have been enhanced. Note also that when equalizing images with narrow histograms and relatively few grey levels, increasing the dynamic range has the adverse effect of increasing visual graininess.

Histogram equalization adjusts the contrast of the image by using image’s histogram. Applying histogram equalization to a watermarked image can also be a source of embedded data distortion. Here in this paper, the watermarked images are exposed to histogram equalization. It has been observed through experiments on a set of images that the proposed scheme resists the histogram equalization attack.

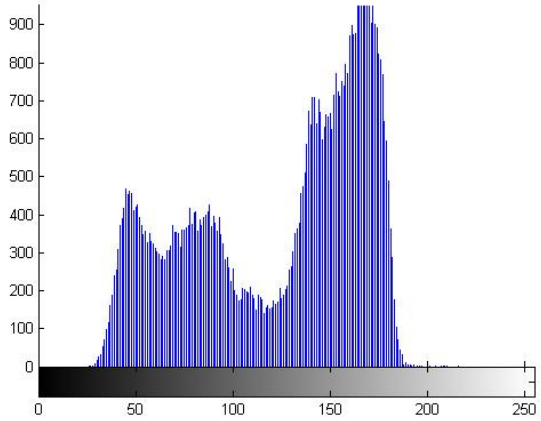
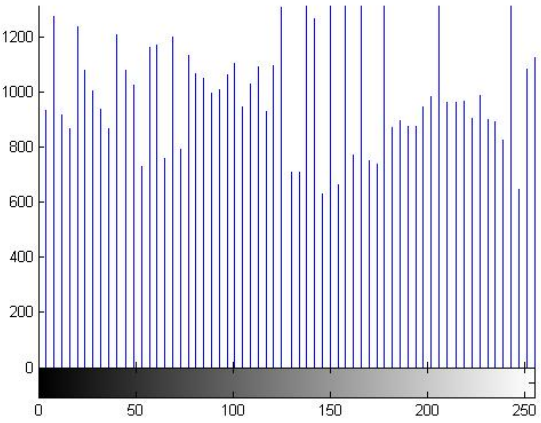



Figure 6.6: (a) Original Image

(b) Histogram for Watermarked Image before applying equalization




c) Watermarked image after applying equalization.

d) Histogram for Watermarked Image after applying equalization

### 6.4.4 Robustness to Image Compression

JPEG standard is a well know technique that allows image compression. The image can be compressed into a stream of bytes. The compression technique is usually lossy compression that may loss the visual quality of the image as well. If such compression is applied to a watermarked image it may lose the data embedded into the image. Experiments performed on the images show that the proposed scheme is resistant when watermarked image is compressed using JPEG standard. The results are shown in the table mentioned below:

In experiments the image is compressed up to 70% and watermark extraction is applied to it. The figures are shown along with their BER.

### 6.4.5 Image Flipping Attack

Vertical Image flipping is yet another very easy attack to perform, however it is so effective that it fails many existing watermarking algorithms.

A set of images are shown below, which includes original, vertical. Experiments show that the proposed approach is robust against vertical attack. The BER=0 in case of vertical flipping.

Figure 6.7: (a) Original Image

(b) Vertical Flipping

## 6.5 Security of the Embedded Data

The embedded data is secured and cannot be easily acquired. Only an authenticated user who knows the key to decrypt the data embedded can get the information stored in the image. Even the visual appearance of the image is quite good that no one can differentiate if any information is stored in an image. The present approach does not require the original image to retrieve the information stored in the image. If however, any attack on the data occurs it will not be correctly, unless key to decrypt the information is obtained.

## 6.6 Discussion

The scheme has been found robust against a number of well know attacks. There are two main attacks to which the watermarked scheme is vulnerable, one is the attack on the embedded data and the second is on the container image. The embedded data can be secured if it is encrypted using an encryption technique. Thus if an intruder has reached the embedded text, he will not be able to get the original text.

Selection of algorithm is therefore based on the nature of its application. There is always a tradeoff between the data hiding capacity and its robustness. For future work, the FFT can be further enhanced to cater the rotation attack on the image.

# Chapter 7

# Conclusion & Future work

This chapter will conclude the work presented in this thesis and give the direction towards further work in this area of research.

## 7.1 Conclusion

This thesis is based on blind watermarking system for static images. An enhanced way of embedding the data keeping the security against attacks on data as well as on the image has been presented here. Frequency domain has been selected for embedding the data in a secure way. The presented scheme had been tested against a number of known attacks and the results were found acceptable. In this thesis, spatial domain watermarking scheme had also been implemented and the same attacks were applied on these images. Over a set of large images the system of spatial domain watermarking was not resistant to these attacks. However, this scheme has a high data hiding capacity. Therefore, the selection of the scheme is based on the factors of required data hiding capacity and robustness. In frequency domain middle frequencies are selected as they provide a good level of visual acceptance even after distorted with the embedded data. The work presented is an

enhanced version as presented in [36]. The data has been secured using encryption in both domains.

The simplest spatial domain as well frequency based watermarking is discussed here. The Fast Fourier Based watermarking proves to be robust against a series of attacks and therefore considered to be a better candidate for a secured application of watermarking schemes. Encrypting the data before embedding adds a data security. Besides the classic use in the ownership protection and copy/access control, the schemes for blind watermarking, presented in the Chapter 4 and Chapter 5, can be a useful tool to send side information in communication. These schemes are workable schemes which can be used for data hiding (text) into a carrier (image) robustly. The proposed scheme can embed data with much more strength while remaining within HVS visibility. It has been tested for attacks such as Lossy Compression, Additive Noise, Histogram equalization and some of the geometric attacks against which the proposed scheme show very good resistance.

One has the option of inserting watermark efficiently, using the Spatial Domain Watermarking but with a compromise on the quality of results obtained at the reception, or robustly using Frequency Domain Watermarking with computational requirements as well as data hiding capacity.

## 7.2 Summary of Contribution

The frequency domain watermarking has been found a reasonably acceptable way for blind watermarking. The reason for selecting the middle frequency is that, changes in high and low frequencies produces visible changes to the images, thus middle frequency is the best carrier for embedding data into it. The data hiding capacity is however quite low, because data has to be repeated on multiple places in the circular ring so that when the data is retrieved there is less chance of errors. In this thesis, spatial domain watermarking has also been discussed. Spatial domain watermarking has a high capacity to store large amount of data as each data bit is stored in LSB. In an image, there is some room for noise. This noise has been replaced with data bits thus leaving no change on the

visual appearance of the image. The changes that are made to the noise part of the image are not visible to the human eye. All the data that is inserted is also encrypted before embedding. The encryption provides a reasonable level of security to the data.

The scheme presented in this thesis is motivated by frequency domain manipulation and has been found robust against a number of well know attacks. Encryption has added a more secure way for embedding the data. The proposed method is compared with latest techniques of watermarking in different domains. Attacking the frequency domain watermarking has shown that it is robust against the attacks.

## 7.3 Future Work

The system can be further enhanced to cater more geometric attacks and robustness. More robust techniques for the encryption can be used so that data can be embedded securely. Currently, the FFT based scheme is working for an image of size 256x256, further improvement can make the algorithm to work for different dimensions of images. Also, another improvement in the algorithm can allow a large amount of data to be stored in the image.

# References

[1] A.Papoulis, ”Probability, Random Variables and Stochastic Processes”, McGraw Hill Inc., 3rd Edn.,1991.
[2] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet and T. Pun “Second generation benchmarking and application oriented evaluation”.
[3] Rade Petrovic, Babak Tehranchi, Joseph M. Winograd by “Digital Watermarking Security Considerations”
[4] M.Matsui, “How to secretly embed a signature in a picture”, In Proceedings of IEEE, 1994.
[5] R.G.van Schyndel, “A digital watermark, In Proceedings of IEEE, Nov1994, pp 86-90.
[6] L.F.Turner, “Digital data security system,” Patent IPN WO 89/08915, 1989.
[7] R. Pickholtz, D. Schilling and L. Milstein, “Theory of Spread Spectrum Communication-A Tutorial”, IEE Trans. On Communications, Vol. Com-30, No. 5, pp.855-884, May 1982.
[8] R.G. van Schyndel, A.Z. Tirkel, and C.F. Osborne. “A digital watermark” In International Conference on Image Processing, volume 2, pages 86-90, Austin, Texas, USA, 1994. IEEE.
[9]. Raymond B. Wolfgang and Edward J. Delp. “A watermark for digital images”. In International Conference on Images Processing, pages 219{222, Lausanne, Switzerland, September 1996. IEEE.
[10]. Raymond B. Wolfgang and Edward J. Delp. “A watermarking technique for digital imagery: further studies”. In International Conference on Imaging, Systems, and Technology, pages 279{287, Las Vegas, NV, USA, 30 June{3 July 1997. IEEE.
[11]. Gerrit C. Langelaar, Jan C.A. van der Lubbe, and Reginald L. Lagendijk. “Robust labeling methods for copy protection of images”. In Sethin and Jain [62], pages 298-309.
[12] I.J. Cox, J. Kilian, T.leighton and T. Shamoon, “Secure spread spectrum watermarking for multimedia”, IEEE Transaction on Image Processing 8,pp.58-68,1999.
[12]. E. Koch and J. Zhao. “Towards robust and hidden image copyright labeling”. In Workshop on Nonlinear Signal and Image Processing, pages 452{455, Neos Marmaras, Greece, June 1995. IEEE.
[13]. Joseph J. K. _O Ruanaidh, W. J. Dowling, and F. M. Boland. “Watermarking digital images for copyright protection”. IEE Proceedings on Vision, Signal and Image Processing, 143(4):250{256, August 1996.
[14] Deepa Kundur and Dimitrios Hatzinakos,"A Robust Digital Image Watermarking Scheme Using the Wavelet-Based Fusion",IEEE ICIP'2001,October,Santa Barbara, California.
[15] Deepa Kundur and Dimitrios Hatzinakos.”Digital watermarking using multiresolution wavelet decomposition” In Proceedings of IEEE ICASSP 2002, volume 5, pages 2969 - 2972, Seattle, WA, USA.
[16] Ren-Junn Hwang, “A Digital image copyright protection scheme based on visual cryptography”, Temkang Journal of Science and Engineering, Vol. 3, No. 2, pp. 97-106, 2000.
[17] L. F. Turner, "Digital data security system," Patent IPN WO 89108915, 1989.
[18] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in 1nt. Conf Image Processing, 1994, vol. 2, pp. 86-90.
[19] G. Caronni, "Assuring ownership rights for digital images," in Proc.Reliable IT Systems, VIS'95.
[20] J. Brassil, S. Low, N. Maxemchuk, and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying," in Proc. Infocom '94, pp. 1278-1287.
[21] K. Matsui and K. Tanaka, "Video-steganography," in Proc. IMA Intellectual Property Project, 1994, vol. I, pp. 187-206.
[22] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multi-level image," in Proc. 1990 1EEE Military Communications Conf, 1990, pp. 216-220.
[23] B. M. Macq and J.-J. Quisquater, "Cryptology for digital TV broadcasting," in Proc. IEEE, vol. 83, pp. 944-957, 1995.
[24] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for data hiding," in Proc. SPlE, vol. 2420, p. 40, Feb. 1995.
[25] G. B. Rhoads, "Indentification/authentication coding method and apparatus," Rep. WIPO WO 95/14289, World Intellect. Property Org.,1995.
[27] Mauro Barni, Franco Bartolini, Vito Cappellini and Alessandro Piva, “A DCT-domain system for robust image watermarking,” in Signal Processing, vol 66 May 1998, pp.357-372
[28] M.D. Swanson, Bin Zhu, A.H.Tewfik, “Transparent robust image watermarking,” in Proceedings of International Conference on Image Processing, vol: 3, 1996, pp 211 -214

[29] C.I Podilchuk and Wenjun Zeng, “Image-adaptive watermarking using visual models,” in IEEE Journal on Selected Areas in Communications, vol: 16 Issue: 4 , May 1998, pp: 525 -539
[30] A. B. Watson, "DCT quantization matrices visually optimized for individual images," in Proc. SPIE Conf. Human Vision, Visual Processing and Digital Display IV, Feb. 1993, vol. 1913, pp. 202–216.
[31] J.J.K.Ó Ruanaidh and T. Pun, “Rotation, scale and translation invariant spread spectrum digital image watermarking,” in Signal Processing, vol 66 May 1998, pp. 303-317
[32] S. Pereira, J.J.K.Ó Ruanaidh, F. Deguillaume, G.Csurka and T. Pun, “Template based recovery of Fourier-based watermarks using log-polar and log-log maps,” in 1999. IEEE International Conference on Multimedia Computing and Systems, vol: 1 , 1999 pp: 870 -874
[33] Shelby Pereira and Thierry Pun, “Robust Template Matching for Affine Resistant Image Watermarks,” in IEEE Transactions on Image Processing, vol 9, no 6, June 2000, pp1123-1129
[34] V. Solachidis and I. Pitas, “Circularly symmetric watermark embedding in 2-D DFT domain,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing 1999, vol: 6 , pp: 3469 -3472
[35] S. Tsekeridou and I. Pitas, “Wavelet-based self-similar watermarking for still images,” in Proceedings of IEEE International Symposium on Circuits and Systems, 2000, vol: 1 pp: 220 -223
[36] V. Licks, R. Jordan, D.F.G. Azevedo, J.S. Correa, P.R.G. Frano, R.D.R Ragundes "Circular Watermark Robust Against Geometric Attacks", accepted for IEEE International Symposium on Information Theory and Its Applications, Hawaii, USA, (2000)
[37] W. Zhu, Z. Xiong and Y.-Q. Zhang, "Multiresolution Watermarking for Images and Video," in IEEE Transactions on Circuits and Systems for Video Technology, Volume 9, No. 4, June 1999, pp. 545--550.
[38] A. Bors ad I. Pitas, Image watermarking using DCT domain constraints, IEEE Int. Conf. on Image Processing (Sep 1996) pp 231-234.
[39] J. Zhao and E. Koch, Embedding robust labels into images for copyright protection, Proc. Int. Congression on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Vienna (Aug 1995) pp 242-251.
[40] Radovan RIDZOŇ, Dušan LEVICKÝ, “Robust Digital Watermarking Based On The Log-Polar Mapping”, in Radioengineering, Volume 16, No. 4, 2007, pp78-81
[41] Chi-Man Pun, “A Novel DFT-based Digital Watermarking System for Images” Signal Processing, 2006 8th International Conference on Volume 2, Issue, 16-20 2006.
[42] Guzman, V.V.F.; Miyatake, M.N.; Meana, H.M.H. “Analysis of a Wavelet-based Watermarking Algorithm” Electronics, Communications and Computers, 2004. CONIELECOMP 2004. 14th International Conference on Volume , Issue , 16-18 Feb. 2004 Page(s): 283 – 287
[43] Jui-Cheng Yen; Hun-Chen Chen; Jui-Hsiang Juan **Blind Watermarking Based on the Wavelet Transform** Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT apos;06. Seventh International Conference on Volume , Issue , Dec. 2006 Page(s):474 - 478