TEMPLATE BASED IMAGE WATERMARKING IN THE FRACTIONAL FOURIER DOMAIN

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TOLGA GÖKOZAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2005

Approval of the Graduate School of Natural and Applied Science.

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Gözde B. Akar
Supervisor

Examining Committee Members:

Prof. Dr. Murat Aşkar	(METU,EE)	
Assoc. Prof. Dr. Gözde B. Akar	(METU,EE)	
Assoc. Prof. Dr. Aydın Alatan	(METU,EE)	
Assoc. Prof. Dr. Tolga Çiloğlu	(METU,EE)	
Ersin Esen	(Tübitak)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

TEMPLATE BASED IMAGE WATERMARKING IN THE FRACTIONAL FOURIER DOMAIN

Gökozan, Tolga
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Gözde B. Akar

January 2005, 111 pages

One of the main features of digital technology is that the digital media can be duplicated and reproduced easily. However, this allows unauthorized and illegal use of information, i.e. data piracy. To protect digital media against illegal attempts a signal, called watermark, is embedded into the multimedia data in a robust and invisible manner. A watermark is a short sequence of information, which contains owner's identity. It is used for evidence of ownership and copyright purposes.

In this thesis, we use fractional Fourier transformation (FrFT) domain, which combines space and spatial frequency domains, for watermark embedding and

implement well-known secure spread spectrum watermarking approach. However, the spread spectrum watermarking scheme is fragile against geometrical attacks such as rotation and scaling. To gain robustness against geometrical attacks, an invisible template is inserted into the watermarked image in Fourier transformation domain. The template contains no information in itself but it is used to detect the transformations undergone by the image. Once the template is detected, these transformations are inverted and the watermark signal is decoded. Watermark embedding is performed by considering the masking characteristics of the Human Visual System, to ensure the watermark invisibility.

In addition, we implement watermarking algorithms, which use different transformation domains such as discrete cosine transformation domain, discrete Fourier transformation domain and discrete wavelet transformation domain for watermark embedding. The performance of these algorithms and the FrFT domain watermarking scheme is experimented against various attacks and distortions, and their robustness are compared.

Keywords: digital watermarking, fractional Fourier domain, template based recovery.

ÖZ

KESİRLİ FOURIER UZAYINDA ŞABLONA DAYALI İMGE DAMGALAMA

Gökozan, Tolga
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Gözde B. Akar

Ocak 2005, 111 sayfa

Sayısal teknolojinin temel özelliklerinden birisi de sayısal verilerin kolaylıkla kopyalanabilmesi ve çoğaltılabilmesidir. Fakat bu özellik, veri korsanlığına yani verinin yetkisiz ve yasadışı olarak kullanılmasına olanak sağlamaktadır. Sayısal verileri koruyabilmek için veriye, dayanıklı ve farkedilmeyen bir sayısal damga gömülür. Damga, veri sahibinin kimliğini barındıran kısa bilgidir. Çalınan bilginin telif haklarını alabilmek için kullanılır.

Bu tez çalışmasında damganın imgeye gömülmesi için uzamsal ve frekans bölgelerini birleştiren kesirli Fourier dönüşümü kullanılmış ve literatürde sıkça kullanılan güvenli yayılı izge damgalama yaklaşımı uygulanmıştır. Fakat güvenli

yayılı izge damgalama metodu ölçeklendirme ve döndürme gibi geometrik saldırılara karşı dayanıksızdır. İmgenin geometrik saldırılara karşı dayanıklılık kazanması için damgalı imgeye Fourier dönüşüm uzayında, yine farkedilmeyen bir şablon eklenmiştir. Şablon özünde bilgi barındırmaz ama imge üzerine uygulanan geometrik saldırıların belirlenmesinde kullanılır. İmgeye gömülü şablonun bulunmasıyla uygulanan geometrik dönüşüm hesaplanabilir ve bu dönüşüm tersine alınarak damganın çözülebilmesi sağlanır. Damgalama, insan görsel sisteminin maskeleme özellikleri dikkate alınarak gerçekleştirilmiş böylece damganın görünmezliği sağlanmıştır.

Tezde ayrıca değişik dönüşüm uzaylarının kullanıldığı damgalama algoritmaları da uygulanmıştır. Bu algoritmalar damga gömmek için ayrık kosinüs dönüşüm uzayı, ayrık Fourier dönüşüm uzayı ve ayrık dalgacık dönüşüm uzayı kullanan algoritmalardır. Kesirli Fourier dönüşümü damgalama algoritmasının ve bu algoritmaların performansları çeşitli saldırılara ve bozunumlara karşı denenmiş ve dayanıklılıkları karşılaştırılmıştır.

Anahtar Kelimeler: sayısal damgalama, kesirli Fourier uzayı, şablona dayalı düzeltim.

To My Parents

ACKNOWLEDGMENTS

I want to express my sincere gratitude to my supervisor, Assoc. Prof. Gözde B. Akar for her continuous help, guidance, understanding and interest throughout this work.

I would like to express my sincere appreciation to my colleagues in ASELSAN for their valuable friendship, help and support.

I would like to extend my special appreciation and gratitude to my parents and my friends for their encouragement, endless love and understanding of my spending lots of time on this work.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

ABBREVIATIONS

FrFT	Fractional Fourier Transform
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
DFT	Discrete Fourier Transform
LPM	Log-Polar Mapping
FMT	Fourier-Mellin Transform
FFT	Fast Fourier Transform
MSE	Mean Square Error
SNR	Signal-to-Noise Ratio
PSNR	Peak Signal-to-Noise Ratio
HVS	Human Visual System

CHAPTER 1

INTRODUCTION

During the last decade, digital technologies have grown tremendously and digital multimedia contents started to take place widely in our lives. However, a particular drawback of digital content is their ability to be volatile and easily processed. Since digital media can be easily duplicated without any loss of quality, the digital products attract the attention of hackers. On the other hand, the commercial exploitation of the Internet provides distribution of these information without too much cost. The ease by which a digital information can be duplicated and distributed has led to the need for effective copyright protection techniques.

One way to protect multimedia data against illegal recording and distribution is to embed information, called watermark, into the digital media that characterizes the person who applies it and, therefore, marks it as being his intellectual property. The embedded information can be extracted anytime to get the copyright information. Thus, the watermark must always remain in the data, and be detectable at anytime.

There are many application areas of the watermarks such as owner identification, proof of ownership, broadcast monitoring and etc. Each application needs its own requirements. Some of the requirements are robustness, fidelity, computational cost, and etc. The design of the watermark algorithm must provide these requirements.

Robustness is a major concern for most of the watermark algorithms. A robust watermark must resist to possible processings and remains detectable. These processings get a common name, *attack*. There are many kind of attacks and it is

probably impossible for a watermark to resist all kind of attacks, however, it is unnecessary and excessive. The robustness criteria is specific for the type of application.

A watermark needs a transformation domain for embedding it into digital media. This domain can be spatial domain [1,2] as well as frequency domain [17,18,19]. Several researches [3,4] show that it would be more robust to embed a watermark in frequency domain. Frequency domain techniques, mostly, depend on the spread spectrum approach, which suggests embedding the watermark in *spread* of frequencies. Therefore, the signal energy present in any signal frequency (thus the watermark) becomes undetectable.

In this chapter, first, we will give brief information on the different types of information hiding techniques. Then we will describe the main steps of the watermarking applications, such as watermark embedding and detection. The applications of different types of watermarks and the corresponding requirements are described later. Then, different types of attacks are introduced and they are categorized into four main groups. Finally, we will give the scope of the thesis.

1.1 DATA HIDING TERMINOLOGY

There are various techniques for information hiding into digital media. They are used for several purposes as well as copyright protection. In this section we will briefly give information about some data hiding terminology.

Two basic methods of information hiding are cryptography and steganography. The term steganography means “cover writing” and cryptography means “secret writing”**.**

Cryptography is a widely used method for protecting the digital content of the media. The message is encripted before transmission and decripted at the receiver side with the help of a key. Nobody, except the one having the key, can determine the content of the key. The message is caled the *plain text* and the encripted form is called the *cipher text* [9]. The information is protected at the time for transmission. However, after decryption, the information becomes unprotected

and it can be copied and distibuted. The schematic representation of the cryptography is given in Figure 1.1 (b).

In steganography, the message is embedded into the digital media rather than encrypting it. The digital media content, called the *cover*, can be determined by anybody , however, the message hidden in the cover can be detected by the one having the true key. The message stays in the message after the receiver gets the data. This allows steganography to protect the embedded information after it is decrypted. Steganography is therefore broader than cryptography. The schematic representation of the steganography is given in Figure 1.1 (a).

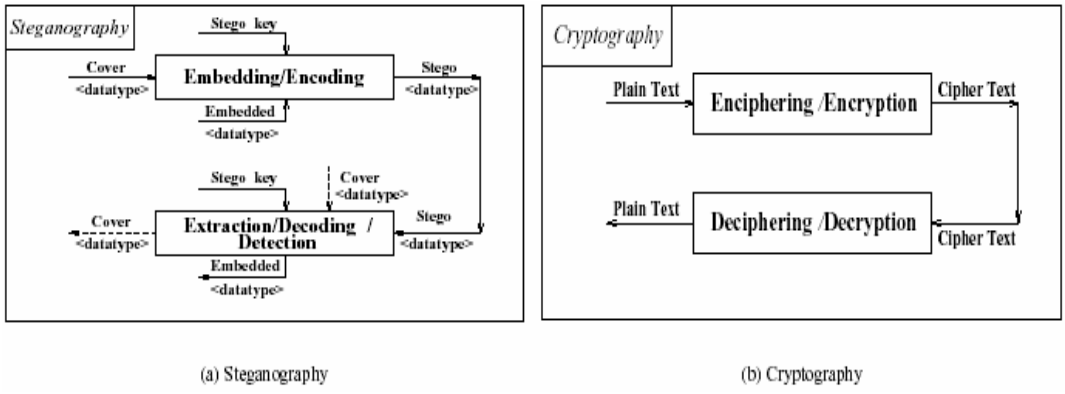


Figure 1.1 Steganography vs. Cryptography [8].

Watermarking techniques are particular embodiments of steganography. However, their usage aim is different. A watermark contains copyright information of the cover object. The robustness is a major concern for watermarking because the valuable data is protected (or the ownership is proved) as long as the watermark is present in it. On the other hand, hidden message may have have no value and no relationship with the cover in steganography.

The terms of watermarking and fingerprinting are sometimes confused. Fingerprinting involves hiding a unique identifier for the customer who originally acquired the file, and therefore is allowed to use it. Unlike watermarks, fingerprints identify the customer, not the copyright owner of the file. If the file is found in the

possession of somebody else, the copyright owner can use the fingerprint to identify the customer, which violated the license agreement by distributing a copy of the file.

1.2 THE WATERMARKING FRAMEWORK

Watermarking is the process of embedding a signal, called a watermark, into a multimedia object such that watermark can be detected or extracted later to make an assertion about the object. The object may be an image or audio or video.

In general, any watermarking scheme consists of three parts:

- The watermark signal,
- The encoder that embeds the watermark into the media
- The decoder and comparator that verifies the presence of watermark

Most watermarking techniques use a spread spectrum approach for embedding the signal, which is essentially the insertion of a pseudo-noise signal with a small amplitude into the content. The watermark can be embedded directly onto the content or onto its frequency domain. Let us denote an image by *I*, a signature by $W = \{w_1, w_2, ..., w_n\}$ the watermarked image by *I'*. *E* is an encoder function, it takes an image *I* and a signature *W*, and it generates a new image which is called watermarked image I', i.e.

$$E(I, W) = I'$$

The watermark process is shown in Figure 1.2.

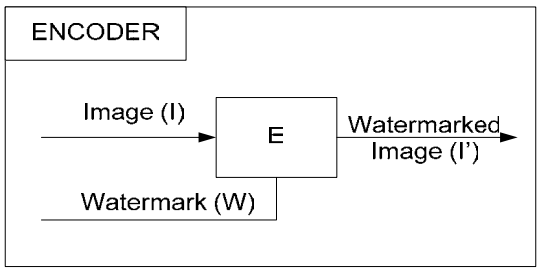


Figure 1.2 Watermark Encoding [9].

A decoder function *D* takes an image *J*, which can be watermarked, unwatermarked or possibly corrupted by any attack, whose ownership is to be determined and recovers a signature *W'* from the image. In this process original unwatermarked image may be used or may not be used according to the algorithm. If the detector does not need the original copy, watermarking scheme is called *public watermarking or blind watermarking*, if the detector needs the original image, then, it is called *private watermarking* or *non-blind watermarking* [10]. If the original image is used, the watermark can be extracted in its exact form (if the image is not corrupted). If it is a blind detection, we can determine whether a specific given watermarking signal is present in an image.

$$D(J, I) = W'$$

Figure 1.3 illustrates the watermark detection process.

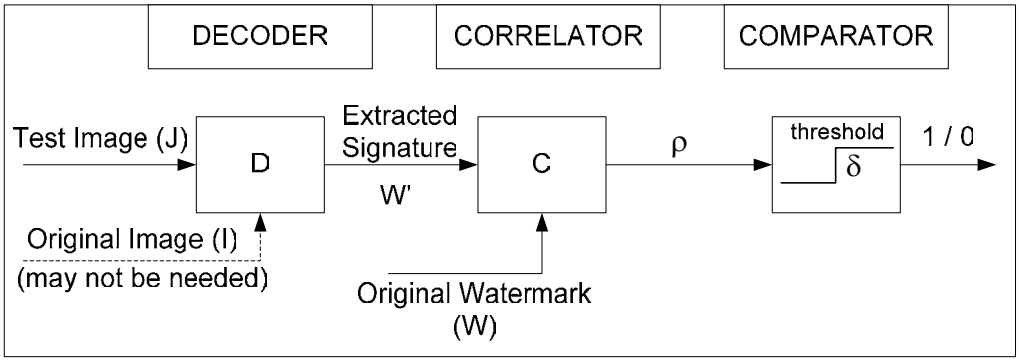


Figure 1.3 Watermark Detection [9].

The presence of the watermark can be proved using correlation methods. The correlator function *C* computes the correlation value, ρ. The computed correlation is compared with a detection threshold. If the correlation value exceeds the threshold value, the image is said to be watermarked. At the comparator, a binary output is generated, where binary 1 means watermark detected and 0 means not detected.

$$\rho(W', W) = \begin{cases} 1, & \rho \geq threshold \\ 0, & otherwise \end{cases}$$

1.3 TYPES AND APPLICATIONS OF WATERMARKS

Watermarking techniques can be categorized according to the application domain, according to the type of document, according to the human perception and according to the application [8,9,10]. Classification of watermarking techniques is shown in Figure 1.4.

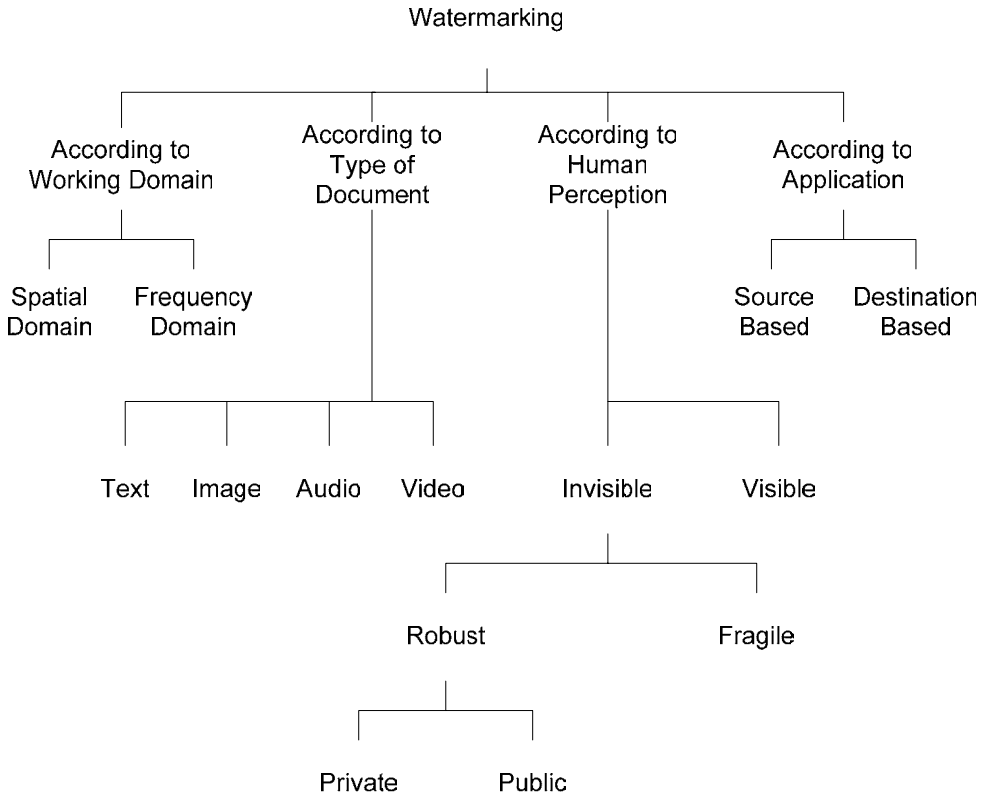


Figure 1.4 Types of watermarking techniques [8].

Watermarks can be embedded into the multimedia content in spatial domain or in frequency domain. Frequency domain watermarking methods may use several different domains, such as discrete cosine transformation (DCT) domain, discrete Fourier transformation (DFT) domain, discrete wavelet transformation (DWT) domain etc. In the literature, it is pointed that the frequency domain techniques are more robust then spatial domain techniques [3,4].

The watermarking algorithms can be named according to the embedded multimedia content, such as text, image, audio and video watermarking.

Another way to categorize the watermarks is made according to human perception. Visible and invisible watermarks are of this type. Logos are the examples of the visible watermarks that indicate the owner of the content [11]. A traditional way of visible image watermarking is to print *"©date,owner"* mark onto the image. One disadvantage of visible watermarks is that they can easily be removed from the digital cover.

Invisible watermarks change the media in a way that they are perceptually unnoticable. They can only be detected by an appropriate detection method. They identify the owner of the digital media. Unlike visible watermarks, the invisible watermarks could not be removed from the media because they became an integral part of the content after embedding. However, they can be made undetectable by some manupilations and distortions called *"attacks"*. The watermark, ideally, must stand all possible attacks. Proof of ownership is another application area for invisible watermarks, however, it needs a higher level security then owner identification. Craver *et al*. [12] proposed a watermarking scheme that can be performed on a watermarked image, to allow multiple claims of rightful ownership.

The two types of invisible watermarks are robust and fragile watermarks. The robust algorithms aim the watermark survival after possible distortions such as possible compressions, filterings and noise additions. However, the fragile watermarks are used to detect if there is any manipulation or modification on the digital content. These modifications would alter or destroy the watermark. Fragile watermarks can be used for content authentication such as *trustworthy camera*. A watermark is embedded into the frame when it is captured by the camera. The watermark will be lost if any alterings made so verifying if the frame is the original captured one or not.

The invisible robust watermarks are divided into two categories as private and public watermarks, as described in previous section. The private algorithms need the original content to detect the watermark where the public watermarks do

not need.

According to the applications, the watermark could be classified as source based and destination based watermarks. In the source based algorithms, all the copies are watermarked with a unique watermark and used for ownership identification or authentication. The watermark identifies the owner of the content. However, the destination based watermarks (fingerprints) are embedded uniquely to each copy and used to trace the buyer in the case of an illegal operation. Fingerprints can be used for broadcast monitoring. A unique watermark is put into each video or audio-clip prior to broadcast. Automated computers monitors the broadcast and detects when and where each clip is appeared [11] .

Another application area of the watermarks is copy control. The digital media can be copied without any quality loss. To prevent this, a watermark can be inserted in a media such that a recorder would not copy it if it detects a watermark that indicates copying is prohibited. However, this could be successful if all the manufactured recorders can implement watermark detection algorithms.

1.4 PROPERTIES OF WATERMARKS

Major properties of the watermarks are robustness, fidelity, computational cost and false positive rate [11]. However, a watermark may not satisfy all of these properties. In addition, that may be not required for all types of watermarks. For a visible watermark, fidelity is not an issue, however, for an invisible watermark it is one of the most important issues. The watermark is designed to fulfil the needed properties according to the type of the application. On the other hand, one property may challenge with another. To increase the strength of the watermark increases the robustness, whereas it decreases the fidelity. A trade-off must be made according to the applications. In this section, we will examine those properties.

1.4.1 Robustness

In most watermarking applications, the marked data is likely to be processed

in some way before it reaches to the watermark receiver. For example, in television and radio broadcast, the watermarked media should resist to lossy compression, D/A-A/D conversion applied on the transmitter and receiver side, and some small amount of horizontal and vertical translations. In addition, noise can added because of the transmission medium. Most images and videos on the web are subjected to compression, thus if a watermark is present in these objects, it must resist to compressions. Sometimes, one may want to use only some portion of the multimedia content, and hence crops and removes the other parts which requires robustness against cropping. The images may be printed and distrubuted as hardcopy. In this case, geometrical modification and some noise may occur on the image. The distributed copies have different watermarks in broadcasting applications. One may use these copies to provide an unwatermarked copy by averaging all copies which is called collusion attack.

A robust watermark must resist to possible attacks and remains detectable after applied attacks. However, it is probably impossible for a watermark to resist all kind of attacks, in addition, it is unnecessary and excessive. The robustness criteria is specific for the type of application.

On the other hand, the fragile watermarking idea contradicts with the robustness criteria. In these applications, the watermark must be changed or lost after any applied attack.

In many applications, when the signal processing between embedding and detection is unpredictable, the watermark may need to be robust to every conceivable distortions. This is the case for owner identification, proof of ownership, fingerprinting, and copy control. It is also true for any application in which hackers might want to remove the watermark.

1.4.3 Fidelity

High fidelity means that, the amount of degradion caused by the watermark in the cover is imperceptible for the viewer. It is a primary concern for invisible types of watermarks. However, in most applications increasing the robustness by

embedding a more powerfull watermark signal, may result in loss of fidelity. In this case a trade-off must be made and fidelity or robustness may be decreased to a required level. Some watermarking algorithms use visual masking property of the Human Visual System (HVS) and embeds the watermark to imperceptible regions in the cover object. This means embedding the most of the watermark in the speckled regions of the image.

For visible watermarks, it is meaningless to talk about fidelity. However, in this case the watermark may spread in a large or important area of the image in order to prevent its deletion by clipping.

A video signal, transmitted over NTSC, would not have very high quality. Hence, the watermark fidelity is not a big problem for the transmission using NTSC and can be low relatively. However, in HDTV and DVD video, the signals have very high quality and require much higher fidelity watermarks.

1.4.4 Computational Cost

Especially, in broadcast monitoring applications, the watermark embedding operation must not slow down the media production and the watermark detector must work in real-time while monitoring the broadcasts. This would require practical watermarking schemes, which would not create a lot of computational work. On the other hand, it is not very critical for a detector used for proof of ownership, because such a detector will only be used during ownership disputes.

1.4.5 False Positive Rate

A watermark detector may find a wrong watermark in the media or may not find the watermark, although there is. These are called false positives. The false positive rate is the number of false positives expected to occur in a given number of detector runs.

1.5 ATTACKS ON WATERMARKS

In watermarking terminology, an *attack* is any processing that may impair detection of the watermark or communication of the information conveyed by the watermark. The processed, watermarked data is then called *attacked data*.

Robustness against attacks is an important aspect for watermarking schemes. The usefulness of an attacked data can be measured by its perceptual quality and the amount of watermark impairment can be measured by criteria such as miss probability, probability of bit error, or channel capacity. An attack succeeds in defeating a watermarking scheme if it impairs the watermark beyond acceptable limits while maintaining the perceptual quality of the attacked data. [14]

The wide class of existing attacks can be divided into four main groups: removal attacks, geometrical attacks, cryptographic attacks and protocol attacks. Figure 1.5 summarizes the different types of attacks.

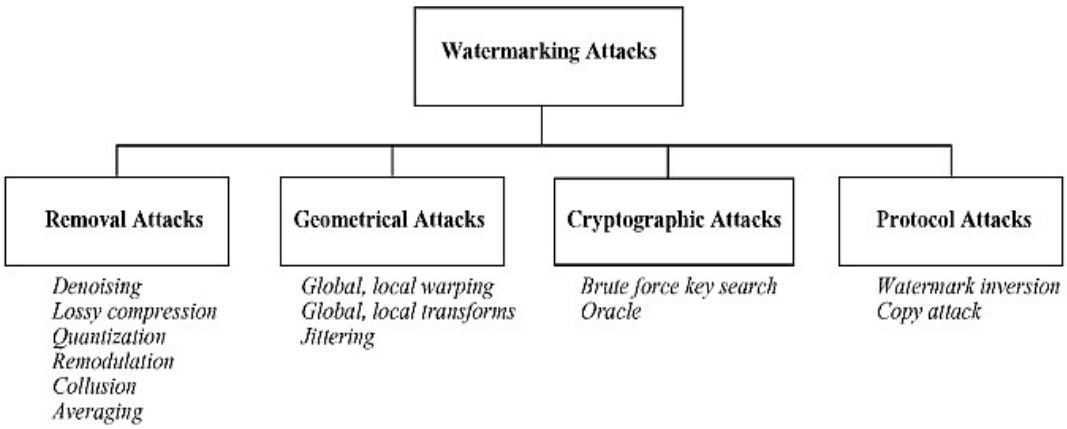


Figure 1.5 Classification of watermark attacks [13].

1.5.1 Removal attacks

Removal attacks attempt to weaken or completely remove a watermark from its associated content, while preserving the content so that it is not useless after the attack is over. This category includes denoising, quantization, remodulation, and

collusion attacks.

Denoising and quantization attacks impair the watermark quality as much as possible, while keeping the quality of the attacked data high enough. Lossy compression has the same effect as denoising.

The remodulation attack aims to predict the watermark. It may be implemented by subtracting the median filtered version of the watermarked image from the watermarked image itself. Then the predicted watermark is removed from the watermarked image, resulting with the median filtered version of watermarked data.

Collusion attacks are applicable when many copies of a given data set, each signed with a different watermark, can be obtained by an attacker. In such a case, a successful attack can be achieved by averaging all copies or taking only small parts from each different copy.

1.5.2 Geometric attacks

Geometric distortions are specific to videos and images including operations as rotation, scaling, translation, cropping etc. In contrast to removal attacks, geometric attacks do not actually remove the embedded watermark, but intend to distort the watermark detector synchronization with the embedded information. The detector could recover the embedded watermark information when perfect synchronization is regained. However, the complexity of the required synchronization process might be too great to be practical.

Recent watermarking methods try to survive from these attacks by use of templates, invariant domains, image feature dependent methods or self synchronizing watermarks to overcome the geometrical transformations inflicted by the attacker [15].

1.5.3 Cryptographic attacks

Cryptographic attacks aim at cracking the security methods in watermarking

schemes and thus finding a way to remove the embedded watermark information or to embed misleading watermarks. One such technique is the brute-force search for the embedded secret information. Another attack in this category is the so-called Oracle attack, which can be used to create a non-watermarked signal when a watermark detector device is available. Practically, application of these attacks is restricted due to their high computational complexity.

1.5.4 Protocol attacks

Craver *et al.* [12] described a method, called the watermark inversion attack or IBM attack, to provide a counterfeit watermarking schemes that can be performed on a watermarked image to create uncertainty about which watermark was inserted first.

Another protocol attack is the copy attack. In this case, the watermark is estimated by using a watermarked data, and this estimated watermark is embedded into another data by adapting the local features to satisfy its imperceptibility.

1.6 SCOPE OF THE THESIS

In this thesis, we concentrate on a watermarking method for digital images that use fractional Fourier transformation (FrFT) domain, which is described by *Djurovic et al.*[5]. FrFT domain combines space and frequency domain. It indicates spectral content of the signal/image as well as the time location of the spectral components. While embedding the watermark signal, widely known spread spectrum approach is used [6,7]. The algorithm's robustness is tested against several attacks. The geometrical attacks, such as rotation and scaling, disturb the synchronization of spread spectrum signal, therefore the watermark could not be detected after geometrical transformations. To solve this problem, we implement a template addition algorithm in addition to the watermark. A template is a local peak in the frequency domain and it is used for detecting the transformations undergone. After detecting the template, the watermark signal becomes resynchronized with the

original and becomes detectable. Watermark embedding is performed by considering the masking characteristics of the Human Visual System, to ensure the watermark invisibility.

This thesis contains five chapters. The reader is first oriented with introduction to some watermarking knowledge such as basic watermark embedding and decoding schemes, types of watermarks, watermark properties and applications, and attacks applied to watermarks.

In Chapter 2, some background materials used in this research are introduced. These materials are properties of spread spectrum signals used in the watermarking applications. Analysis and calculation of detection threshold for determining the watermark presence is discussed later. Finally, image quality metrics used for calculating the amount image distortion are introduced. These metrics are used to calculate image distortion after watermark embedding and prepare a base for comparing different watermark schemes.

In Chapter 3, watermarking algorithms using different transformation domains are introduced. These are examples of DCT, DFT and DWT domain watermarking algorithms. These watermarking algorithms will be used to compare with FrFT domain watermarking algorithms in the experiments.

In Chapter 4, basic definition of FrFT domain will be introduced and the watermarking scheme is proposed. The FrFT domain watermarking scheme is strengthened by using a DFT domain template addition method and applying a visual masking algorithm.

In Chapter 5, the proposed algorithms are compared against several attacks and robustness issues are presented.

In Chapter 6, conclusions about the experiment results and the future work will be presented.

CHAPTER 2

BACKGROUND

2.1 SPREAD SPECTRUM WATERMARKING

A watermark placed in high frequency regions of an image are not robust to removal attacks such that it can be easily eliminated by filtering operations. However, if the watermark embedded in low frequency regions that are perceptually significant, the watermark become more robust but it creates fidelity problem. The problem then becomes how to insert the watermark into perceptually significant regions while preserving the high fidelity [6,7].

To solve this problem, the frequency spectrum of the image is viewed as a communication channel and the watermark is viewed as a signal through it. In spread spectrum communications, the transmitted narrowband signal (the message to be transmitted) is modulated by a broadband carrier signal which broadens (spreads) the narroband signal. So that, the signal energy present in any signal frequency (thus the watermark) is undetectable. Since the locations of the watermark is known, the detection of the watermark is possible. However, to destroy such a watermark would require a lot of distortions, which makes the multimedia content useless.

A watermark that is well placed in the frequency domain, makes it imperceptible to the viewer. This will always be the case if the watermark energy in one single frequency component is sufficiently small. Moreover, by using visual masking properties of the HVS, it is possible to embed the watermark in less perceptual regions, which allows to increase the watermark energy.

2.2 WATERMARK DETECTION

While detecting the presence of the watermark (Figure 1.3), the correlation between the extracted watermark and the original watermark is compared with the threshold value. If it exceeds the threshold, then we can say that the watermark is present. However, selecting a threshold value is a critical issue. If a very low value is selected, it may detect watermarks whether there is not any embedded, or it can detect wrong watermarks as well as the true one. On the other hand, if it is selected as a very high value, than the correlation value may not exceed the threshold value (especially after some attacks). In addition, public watermarking algorithms do not use original image for watermark detection, so the threshold value must be calculated for possibly attacked image. Another point of concern is that, defining a constant value for detection threshold is not practical because the detection process must be generalized and automated.

In this section, we will present a watermark detection threshold calculation method for correaltion-based watermark algorithms described by Pive *et al* [16]. The threshold is chosen according to a fixed constraint on the maximum probability of false positive errrors in watermark detection.

Here, the watermark consists of a set of *n* normally distributed samples $\{w_1, w_2, ..., w_n\}$ which are selected to modify a set of coefficients such as DCT or DWT coefficients according to the following rule :

$$v_i' = v_i + \alpha |v_i| w_i \tag{2.1}$$

where v_i is the original coefficient, w_i is a watermark sample, v_i' is the modified coefficients, and α is a properly chosen parameter for tuning the watermark energy; the higher α, the more robust and the more visible the watermark is. Describing Equation 2.1 by using vector multiplication, we get :

$$V' = V + \alpha |V| W \tag{2.2}$$

If we denote the possibly corrupted and watermarked coefficients as V^*, during the detection phase, the correlation between W and V^* is computed and used as a measure of the presence of *W*. More precisely, given a mark *W* and a set of possibly corrupted and watermarked coefficients V^*, the correlation $\rho(W, V^*)$, which is defined as

$$\rho(W, V^*) = \frac{W.V^*}{n} \tag{2.3}$$

can be used to determine whether a given mark is present or not, by simply comparing it to a predefined threshold.

To decide the presence of a mark, *W*, one of the following situations is possible:

Hypothesis 0 : $V^* = V$ or $V^* = V + \alpha|V|Y$ i.e. the image is not marked or a different mark, Y, is present;

Hypothesis 1 : $V^* = V + \alpha|V|W$ i.e. the mark *W* is present;

To discriminate between *Hp.0* and *Hp.1*, the decoder computes $\rho(W, V^*)$ and compares it with the threshold T_ρ. To determine the value of T_ρ, the decoder error probability can be taken into account. The error probability P_e, i.e. the probability of deciding the wrong hypothesis, can be written as:

$$P_e = P(0|1)P(1) + P(1|0)P(0) \tag{2.4}$$

where $P(0|1)$ is the probability of missing the presence of the mark (false negative), and $P(1|0)$ is the probability of revealing the presence of *W* when *W* is not actually present (false positive), $P(0)$ and $P(1)$ are the priori probability of *Hp.0* and *Hp.1*. By assuming that *Hp.0* and *Hp.1* are equiprobable, and by taking into account the particular decoding strategy, eqn. 2.4 can be put in the form

$$P_e = \frac{1}{2}\left[P(\rho < T_\rho \mid 1) + P(\rho > T_\rho \mid 0)\right] \tag{2.5}$$

where $\rho = \rho(W, V^*)$.

If hypothesis *Hp.0* holds

$$\mu_{\rho|Hp.0} = 0 \tag{2.6}$$

$$\sigma^2_{\rho|Hp.0} = (1+\alpha^2)\frac{\overline{\sigma_v^2}}{n} \tag{2.7}$$

and if *Hp.1* holds

$$\mu_{\rho|Hp.1} = \alpha\,\overline{\mu_{|v|}} \tag{2.8}$$

$$\sigma^2_{\rho|Hp.1} = (1+2\alpha^2)\frac{\overline{\sigma_v^2}}{n} + \alpha^2\frac{\overline{\sigma^2_{|v|}}}{n} \tag{2.9}$$

where

$$\overline{\mu_{|v|}} = \frac{1}{n}\sum_{i=1}^{n} E\left[\left|v_i\right|\right] \tag{2.10}$$

is the average value of $\mu_{|v_i|}$ over the set of marked coefficients, and

$$\overline{\sigma_v^2} = \frac{1}{n}\sum_{i=1}^{n} E\left[v_i^2\right] \tag{2.11}$$

is the average value of $\sigma^2_{v_i}$ over the set of marked coefficients. In Figure 2.1, the pdf's of ρ under hypotheses 0 and 1 are shown. In order to minimize the error probability, a threshold T_ρ has to be chosen such that error probability would be minimum. The optimum threshold is between zero and midway between $\mu_{\rho|Hp.1}$, that

is

$$T_p = \frac{\alpha}{n}\overline{\mu_{|v|}} \tag{2.12}$$

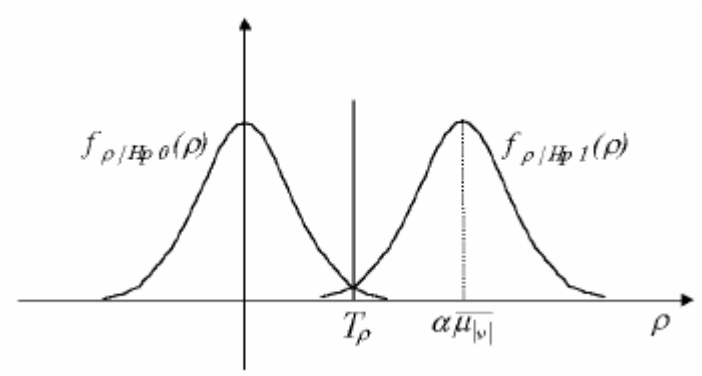


Figure 2.1 The pdf's of ρ under hypotheses of 0 and 1. Attacks are not considered [16].

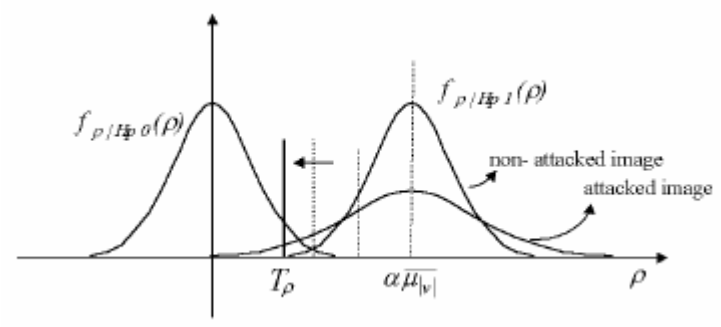


Figure 2.2 The pdf's of hypotheses 0 and 1. Attacks are considered [16].

However, if an image is attacked, the pdf's becomes as shown in Figure 2.2, therefore an error is likely to occur when comparing $\rho(W,V^*)$ with T_ρ. In practical applications, it is better to use a threshold T_ρ that is estimated on the marked image.

$$\overline{\mu_{|v|}} \cong \frac{1}{n}\sum_{i=1}^{n}\left|v_i^*\right| \tag{2.13}$$

An attack applied on the image will alter the mean value and variance

of $\rho(W,V^*)$. In general it can be stated that in presence of attacks the $\sigma_\rho(W,V)$ and $\sigma_\rho(W,V_Y)$ should remain approximately the same, whereas $\sigma_\rho(W,V_W)$ is likely to increase significantly. Therefore because the attacks, two Gaussians are still present, but the one centered in $\overline{\mu_{|V|}}$ has now significantly large variance. This suggested that T_ρ should be set closer to zero, instead of midway between zero and $\overline{\mu_{|V|}}$, so that T_ρ has been fixed to

$$T_p = \frac{\alpha}{3}\overline{\mu_{|v|}} \tag{2.14}$$

However, Piva *et al*. [16] stated that experimental results have shown that when the watermarked image is attacked the proposed threshold leads to a higher watermark-missing rate than was expected. In particular, the probability of missing an embedded watermark results to be considerably higher than the probability of false positive detection. This can be explained by the fact that under attacks it usually happens that $\mu_{\rho|Hp.1} < \alpha\overline{\mu_{|v|}}$ (Figure 2.3). To solve this problem, a different approach for threshold selection has been found. In this case, instead of trying to minimize the error probability P_e, it is chosen to fix a constraint on the maximum false positive probability (e.g. 10^{-6}), so that the threshold is moved leftmost (Figure 2.3).

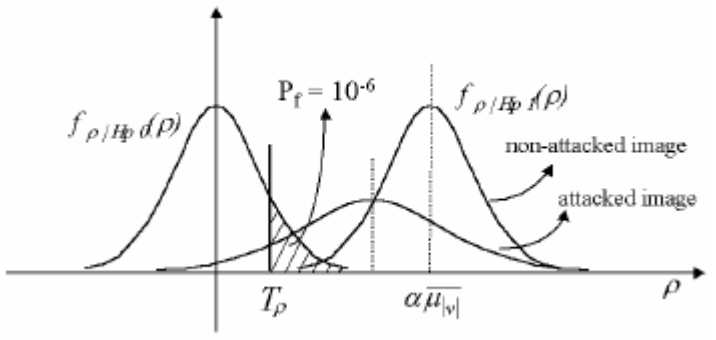


Figure 2.3 The choice of threshold based on a constraint on the maximum false positive probability [16].

In particular, given $P_f = P(\rho > T_\rho \mid 0) = \frac{1}{2} erfc\left(\frac{T_p}{\sqrt{2\sigma_\rho^2}}\right) = 10^{-6}$ the following relation holds:

$$\frac{T_p}{\sqrt{2\sigma_\rho^2}} \geq 3.3 \tag{2.15}$$

in such a way that a new threshold is obtained:

$$T_p = 3.3\sqrt{2\sigma_\rho^2} = 3.3\sqrt{\frac{2(1+\alpha^2)\overline{\sigma_v^2}}{n}} \tag{2.16}$$

Once again, the threshold can be evaluated directly on the watermarked and possibly corrupted image: the value $\left(1+\alpha^2\right)\overline{\sigma_v^2}$ corresponds, in fact $\overline{\sigma_{v^*}^2}$ so that we have:

$$T_p = 3.3\sqrt{\frac{2\overline{\sigma_{v^*}^2}}{n}} \tag{2.17}$$

While calculating the correlation, the average of the linear correlation is used by Piva *et al*. However, if linear correlation will be used in watermarking applications, the algorithm will not be robust to brightness changes in the case of image watermarking. This problem can be eliminated by normalizing the vectors before the correlation. This correlation type is called normalized correlation and formulated as:

$$\rho_{nc}\left(W, V'\right) = \frac{W.V^*}{\sqrt{(W.W)(V^*.V^*)}} \tag{2.18}$$

Another form of correlation is called correlation coefficient. It is obtained by subtracting out the means of the vectors before computing the normalized

correlation.

$$\rho_{cc}\left(W,V^*\right)=\frac{\widetilde{W}.\widetilde{V}^*}{\sqrt{(\widetilde{W}.\widetilde{W})(\widetilde{V}^*.\widetilde{V}^*)}}$$
$$\widetilde{W}=W-\overline{W} \qquad (2.19)$$
$$\widetilde{V}^*=V^*-\overline{V}^*$$

Correlation coefficient provides robustness against changes in DC term o the work, such as the addition of a constant intensity to all pixels of an image.

We have defined threshold formula as in eq. 2.17. However, this threshold is true when using average linear correlation. The following threshold formula can be used when using correlation coefficient.

$$T_{\rho}=3.3\sqrt{\frac{2n.\overline{\sigma_{v^*}^2}}{\left(\widetilde{V}^*.\widetilde{V}^*\right)\left(\widetilde{W}.\widetilde{W}\right)}} \qquad (2.20)$$

We will use correlation coefficient for finding correlation between watermarked coefficients and watermark signal and above formulation for decision threshold for the algorithms described in this thesis.

2.3 IMAGE QUALITY MEASURES

Objective image quality measures play an important role in various image processing applications such as in digital image watermarking. There are basically two classes of objective quality or distortion assessment approaches. The first are mathematically defined measures such as the widely used mean square error (MSE), peak signal to noise ratio (PSNR), and signal to noise ratio (SNR). The formulations for these are:

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[x(i,j) - x'(i,j)\right]^2}{MN}$$

$$PSNR = 20 * \log\left(\frac{255}{\sqrt{MSE}}\right) \tag{2.21}$$

$$SNR = 10 * \log\left(\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}x(i,j)^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[x(i,j) - x'(i,j)\right]^2}\right)$$

where M, N stands for the size of the image in both horizontal and vertical axes, $x(i,j)$ stands for the pixel values for the original image and $x'(i,j)$ corresponds to the pixel values of the distorted image.

MSE stands for the amount of error between two images, PSNR stands for error variance against the maximum possible image variance and SNR stands for the variance of the signal against the variance of the noise. Mathematically defined measures are still attractive because of their calculation simplicity. Additionally, they are independent of viewing conditions and individual observers. Although it is believed that the viewing conditions play important roles in human perception of image quality, in most cases, they are not fixed and specific data is generally unavailable to the image analysis system. If there are N different viewing conditions, a viewing condition method will generate N different measurement results that are inconvenient to use.

The second class of measurement methods consider human visual system (HVS) characteristics in an attemp to incorporate perceptual quality measures. Unfortunately, these complex metrics do not show any clear advantage over mathematical measures such as PSNR and SNR under strict testing conditions and different image distortion environments.

Wang et al. [30,31] proposed a mathematically defined universal image quality index. The quality measurement approach does not depend on the images being tested, the viewing conditions or the individual observers. More importantly, it

provides meaningful comparison across different types of image distortions.

To find the quality index (Eqn. 2.22), first, the original ($x = \{x_i \mid i = 1,2,...,N\}$) and the test ($y = \{y_i \mid i = 1,2,...,N\}$) images are subjected to a 8×8 sliding window and for each position of the window, the formula below is calculated, where bars over letters designate average and σ stands for the variance of the pixel values within the window.

The sliding window calculations results in a quality map of the image where the dynamic range of the map is [-1, 1]. The best value 1 is achieved if and only if $y_i = x_i$ for all *i*. The overall quality index value is the average of the quality map. The quality index can be stated as:

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}.\bar{y}}{\bar{x}^2 + \bar{y}^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \qquad (2.22)$$

The first component is the correlation coefficient between *x* and *y*, which measures the degree of linear correlation between *x* and *y*, and its dynamic range is [-1,1]. The best value 1 is obtained when $y_i = ax_i + b$ for all I where a and b are constants and a is positive. Even is x and y are linearly related, there still be relative distortions between them, which is evaluated in the second and third components. The second component, with a value range of [0,1], measures how close the mean luminance is between x and y. It equals one if and only if $\bar{x} = \bar{y}$. σ_x and σ_y can be viewed as estimate of the contrast of x and y, so the third component measures how similar the contrasts of the images are. Its range lies between 0 and 1, where the best value 1 is achieved if and only if $\sigma_x = \sigma_y$.

Perceptual models are used not only to measure the perceptual impact of a watermark but also to control it during the watermark embedding process. Most watermarking systems attempt to shape the added pattern according to some perceptual model to achieve automatic adjustment of the embedding strength and obtain a desired perceptual distance.

CHAPTER 3

REVIEW OF WATERMARKING ALGORITHMS

In this section, we review public watermarking algorithms, which use different domains for watermark embedding. However, they all use secure spread spectrum scheme and they are all frequency domain techniques. These algorithms are examples of DCT domain, DFT domain and DWT domain techniques. The algorithms try to embed one-bit watermark and the detection of a watermark is described by true (or binary 1). In DCT domain watermarking scheme, a visual masking algorithm is implemented to increase the fidelity of the watermark. The DFT domain algorithms are popular nowadays, because of their ability to resist geometrical attacks because of the natural properties of Fourier transform. Since the DWT domain contains sub-bands indicating the low and high frequencies, there is no need to apply a visual masking. In addition, to detect the watermark the locations of the marked coefficients must be known for DCT and DWT domain approaches, however, for DWT domain approach, the embedded coefficients are detected automatically.

3.1 DCT DOMAIN APPROACH

The described algorithm below is a public, one-bit watermarking scheme, which uses DCT domain for watermark embedding by the spread spectrum approach. It is described by M. Barni, F. Bartolini, V. Cappellini and A. Piva [17]. A sequence of random real numbers is embedded in selected DCT coefficients. Embedding is performed by exploiting the masking characteristics of the Human Visual System, to ensure the watermark invisibility.

The watermark consists of M randomly generated real numbers $W = \{w_1, w_2, ..., w_M\}$; each value w_i is a Gaussian random variable having zero mean and unity variance. To gain more robustness, a longer sequence is used as a watermark. However, this introduces some problems from the point of view of mark visibility, which has been solved by properly choosing the set of DCT values the watermark is superimposed to, and by perceptually hiding it in image areas with higher luminance variance.

For watermark embedding, the DCT of a gray scale image is computed and the coefficients are ordered by using zigzag scan. Thus, the DCT coefficients form a vector, which the low frequency components are involved in the first region and the high frequency regions involved in the last region of the vector. Always the same components of the vector are selected to be marked to prevent the need for the original image. In order to obtain the perceptual invisibility without loss of robustness, the first L coefficients of the vector are skipped; then the watermark is embedded into the next M coefficients which are $V = \{v_{L+1}, v_{L+2}, ...v_{L+M1}\}$. In this way a new vector is obtained, $V' = \{v_1, v_2, ..., v_L, v'_{L+1}, ..., v'_{L+M}, v_{L+M+1}, ...\}$, according to the following rule:

$$v'_{L+i} = v_{L+i} + \alpha \left| v_{L+i} \right| w_i \qquad i = 1,2,...,M$$

The vector V' is then reinserted in the zigzag scan and inverse DCT is performed, obtaining the watermarked image I'. The bold strip shows in Figure 3.1 shows the embedded M coefficients in DCT domain. The upper-left corner is the omitted L coefficients corresponding to the low frequency regions in the image.

Figure 3.1 The bold strip shows the locations of the watermarked DCT coefficients.

For the watermark detection algorithm, the DCT coefficients of the possibly corrupted image, I^* are computed. Then it is ordered by using zigzag scan and the M coefficients next to the L skipped coefficients are selected to generate a vector $V^* = \{v^*_{L+1}, v^*_{L+2}, ..., v^*_{L+M}\}$. The average linear correlation can be calculated as:

$$\rho = \frac{1}{M}\sum_{i=1}^{M} v^*_{L+i} w_i$$

If the watermarked image has not been corrupted; then $v^*_i = v'_i$ and, for the true watermark, ρ becomes:

$$\rho = \frac{1}{M}\sum_{i=1}^{M}\left(v_i w_i + \alpha |v_i| w_i^2\right)$$

The coefficients w_i can be modelled as independent and identically distributed random variables, having symmetrical probability density function and zero mean. Property that different vectors W are orthogonal; it is possible to demonstrate that two Gaussian random variables ρ_1 (if the watermark detected does not match the embedded) and ρ_2 (if the watermark matches the embedded one), with same variance $\sigma_\rho^2 = M\sigma_v^2$ and mean respectively $\mu_1 = 0$ and $\mu_2 = \alpha M \mu_{|v|}$ are

obtained, where $\mu_{|v|} = E[|v|]$ and $\rho_v^2 = E[v^2]$. In order to get a low detection error probability, the factor $k = \mu_v / \sigma_v$, i.e. the distance between Gaussian curves must be large enough. The factor μ_v increases with the random sequence length *M* and with α; the factor σ_v decreases when the number of skipped coefficients *L* increased. Therefore, increasing *L* and *M* increases the factor, *k*. However, increasing *L* and *M* will embed the watermark in the higher frequency regions. This will decrease the robustness of the algorithm against attacks such as compression, and low pass filtering. Therefore, optimum values must be chosen for *L* and *M* to increase *k* sufficiently and guaranteeing the algorithm is still robust to attacks. In the experiments we will use *L*=*M*=16000.

Barni *et al.*[17] indicate that, the correlation is computed for 1000 different watermark sequences and the sequence producing the highest correlation value is chosen as the embedded watermark. However, this may not be true after some distortions applied to the image. If the image is distorted enough, the watermark may be lost. Therefore, it would be better to define a threshold value for detection purpose. In addition, we will prefer using correlation coefficient in order to use linear correlation. We will compute the threshold value according to the method described in Chapter 2.2. According to equation 2.19 and 2.20:

$$\rho_{cc}\left(W, V^*\right) = \frac{\widetilde{W}.\widetilde{V}^*}{\sqrt{(\widetilde{W}.\widetilde{W})(\widetilde{V}^*.\widetilde{V}^*)}}$$

$$T_\rho = 3.3\sqrt{\frac{2M.\overline{\sigma_{v^*}^2}}{(\widetilde{W}.\widetilde{W})(\widetilde{V}^*.\widetilde{V}^*)}}$$

$$\widetilde{V}^* = V^* - \overline{V}^*$$

$$\widetilde{W} = W - \overline{W}$$

where $\sigma_{v^*}^2$ is the variance value over the set of possibly distorted watermarked DCT coefficients in vector V^*. The correlation coefficient distribution of 1000 different watermarks (while *L*=*M*=16000 for the 512x512 sized “Lena”

image) are given in Figure 3.2. The SNR value after embedding the watermark is 35dB. Here, half of the correlations are computed by using true watermarks and the others are computed by using wrong watermarks. The threshold (red distribution) is calculated for each experiment.

The same watermarked image (Figure 3.3(a)) is tested with 1000 different watermarks. Only the 250th watermark is true this time. The correlation coefficient values for these watermarks are show in Figure 3.3 (b). The true watermark gives higher correlation value and it is the only one, which exceeds the threshold value.

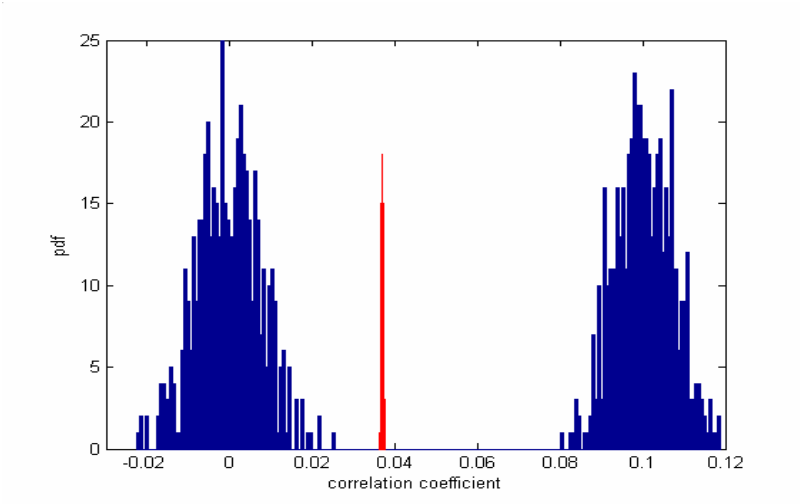
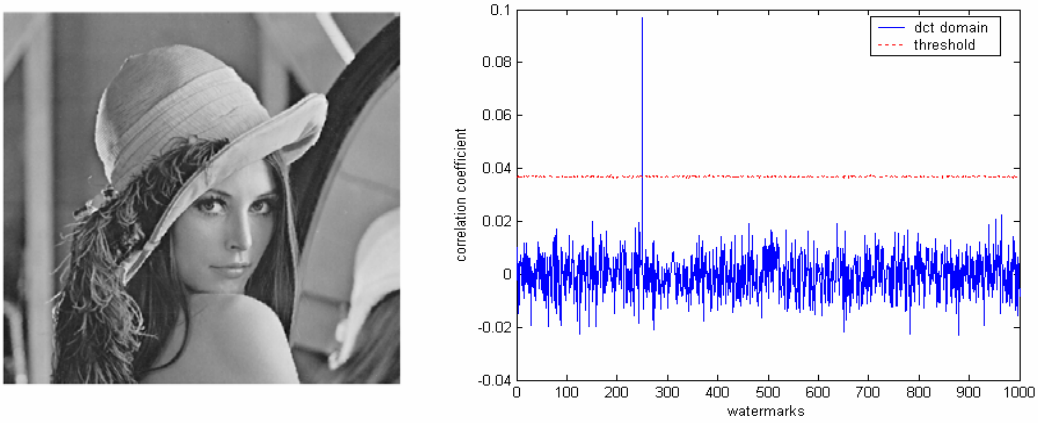


Figure 3.2 Distribution of the correlation coefficients and threshold values for DCT domain watermark detection for 1000 experiments. Half of them are computed with true watermarks and the other half with wrong watermarks.



(a) (b)

Figure 3.3 (a) Watermarked "Lena" image. (b) Correlation coefficient values for 1000 different watermark signals in DCT domain. The 250th signal is the true one.

In order to enhance the robustness of the watermark by increasing the value of α, the masking characteristics of the Human Visual System can be exploited to adapt the watermark to the image being signed: the watermark can be computed by subtracting the watermarked image from the original one, which corresponds to the watermark in spatial domain. Then this watermark is re-embedded into the image by multiplying a variance matrix β as:

$$y''_{i,j} = y_{i,j} + \beta_{i,j}(y'_{i,j} - y_{i,j}) \qquad (3.1)$$

The variance matrix is calculated according to the local variance values in the image. A nxn sized window is slit on the image and the variance of each window is computed, resulting with the local variances of each pixel. Then, the variance matrix is calculated by using a map function, which maps the local variance values into the range specified for the variance matrix. The map function is shown in Figure 3.4. The pixels within the high textured regions have high variances, which correspond to high β values, and the pixels within uniform regions have low variances, which correspond to low β values. By using this technique, it is possible

to increase the watermark strength α, so that the error probability is further diminished.

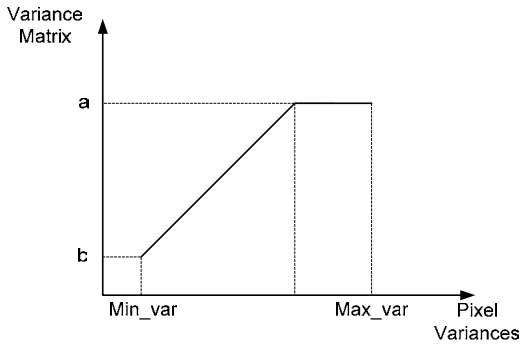


Figure 3.4 Map function for computing variance matrix. The local variances are mapped into the values of variance matrix.

3.2 DFT DOMAIN APPROACHES

The DCT and DWT domain watermarking techniques are not robust against geometrical attacks such as rotation, scaling and translation. However, the DFT domain watermarking algorithms takes the advantage of Fourier transformation properties against geometrical attacks. The DFT properties are described below :

- The Translation Property : Shifts in spatial domain cause a linear shift in the phase component. That is, the magnitude components of Fourier transformation do not effected from linear shifts in saptial domain.

$$f(x_1 + a, x_2 + b) \xleftrightarrow{FT} F(k_1, k_2)\exp[-j(ak_1 + bk_2)] \qquad (3.2)$$

- Reciprocal Scaling Property : Scaling the axes in the spatial domain causes an inverse scaling in the frequency domain.

$$f(\rho x_1, \rho x_2) \xleftrightarrow{FT} \frac{1}{\rho} F(\frac{k_1}{\rho}, \frac{k_2}{\rho}) \qquad (3.3)$$

- Rotation Property : Rotating the image through an angle θ in the spatial domain causes the Fourier representation to be rotated through the same angle.

$$f(x_1\cos\theta - x_2\sin\theta, x_1\sin\theta + x_2\cos\theta) \xleftrightarrow{FT} F(k_1\cos\theta - k_2\sin\theta, k_1\sin\theta + k_2\cos\theta) \quad (3.4)$$

In this section, three different watermarking algorithms, which uses DFT domain for watermark embedding are introduced. They all use the advantages of Fourier Transform to stand against geometrical attacks in different ways; in addition, they are completely different approaches.

3.2.1 RST Invariant Domain Watermarking

The aim of this algorithm is to embed the watermark in a transformation domain which is not affected from rotations, scaling and translations occurred in spatial domain. The algorithm is described by O. Ruanaidh and T. Pun [18].

Figure 3.5 illustrates the process of obtaining the RST transformation invariant from a digital image. The watermark takes the form of two dimensional spread spectrum signal in the RST transformation invariant domain.

A Fourier transform (FFT) is first applied which is then followed by a Fourier-Mellin transform (FMT- A log-polar mapping (LPM) followed by a Fourier transform). The invariant coefficients selected for their robustness to image processing are marked using a spread spectrum signal. The inverse mapping is computed as an inverse Fourier transform (IFFT) followed by an inverse Fourier-Mellin transform (IFMT- An inverse log-polar mapping (ILPM) followed by an inverse FFT). The inverse transformation from RST invariant domain to the image domain uses the phase computed during the forward transformations from image domain to the RST invariant domain.

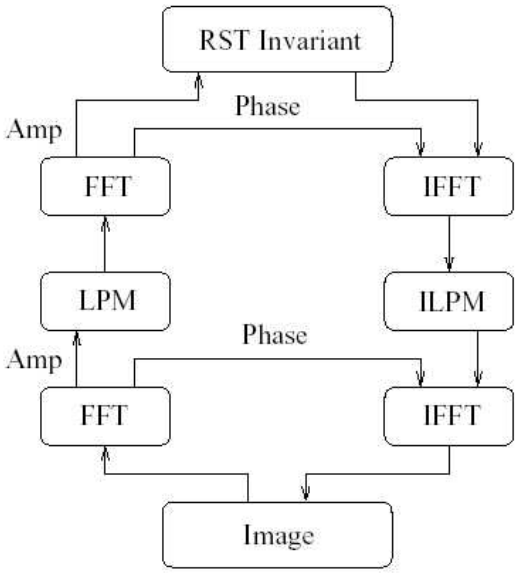


Figure 3.5 Diagram of RST invariant watermarking scheme [18]

To detect the watermark, the image is transformed to the RST invariant domain and the watermark is decoded.

The main idea behind log-polar mapping is to find a representation in which rotation and scaling operations are converted to linear shifts. This transformation maps the spatial coordinate axis (x, y) to polar axis (μ, θ) using the forward (eqn. 3.5) and inverse transformation (eqn. 3.6) equations:

$$\mu = \frac{1}{2}\ln\left(x^2 + y^2\right)$$
$$\theta = \tan^{-1}\frac{y}{x} \tag{3.5}$$

$$x = e^{\mu}\cos\theta$$
$$y = e^{\mu}\sin\theta \tag{3.6}$$

The forward transform converts a scaling in the form of $(\lambda x, \lambda y)$ into a translation (shift) expressed in terms of polar coordinates as $(\mu + \log\lambda, \theta)$, and a rotation of δ degrees is converted to a shift in the θ axis stated as $(\mu, \theta + \delta)$. Hence,

if we apply Fourier transform to the log-polar representation, we obtain a rotation and scale invariant domain because of the shift invariance property of Fourier transform.

The problem in this theoretically elegant method lies in its implementation. When applied on a digital image, the transformations require a lot rounding because of the trigonometric and logarithmic operators. This rounding causes a large amount of loss in the data, which results in huge amount of image quality loss.

256x256 sized *Lena* image (Figure 3.6 (a)) is transformed to RST invariant domain and then transformed back to image domain (Figure 3.6 (b)). The SNR value between these two images is 9.5dB. This shows the amount of data loss between transformations. To reduce the amount of data loss, the size of the transformations must be enlarged, which increases the computational cost a lot. In ideal case, where the size of the transformations are infinite, hence they appear as continuous transformations, there would not be any data loss, however to reach infinite sizes is impossible. Therefore, we will not implement this algorithm in the thesis, however, it creates a good base when dealing with geometrical distortions.

Figure 3.6 (a) Original *Lena* image. (b) Image after transformed to RST invariant domain and then back to spatial domain.

3.2.2 Circular Symmetric Watermarking

Another watermarking application in DFT domain is constituting the watermark signal in circularly symmetric form. The algorithm is described by I.Pitas and V.Solachidis [19]. As we explained, the usage of DFT domain makes the algorithm robust against translation. By embedding circularly symmetric watermark, it gains robustness against rotation attack. In addition, the algorithm is robust against scaling.

For the NxN image, let , $V_M(k_1,k_2)$ be the magnitude, $V_P(k_1,k_2)$ be the phase of Fourier transform. Let also $W(k_1,k_2)$ be the watermark. The watermark consists of a 2-D circularly symmetric sequence taking the values 1 or -1 and has zero mean value and is embedded to the magnitude coefficients of the Fourier transform. The watermark should affect neither low frequencies (in order to be visible) nor the high frequencies (in order to be robust against compression). By assuming that the zero frequency term $I(0,0)$ is in the center of the transform domain, the region in which the watermark is embedded should be a ring covering the middle frequencies. Thus,

$$W(r,\theta)=\begin{cases}0 & if \quad r<R_1 \; and \quad r>R_2\\ \pm 1 & if \qquad R_1<r<R_2\end{cases}$$

where

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

The ring is seperated in *S* sectors and *N* homocentric sub-rings of radius $r\in[R_1,R_2]$. The resulting ring is formed of *S.N* pieces. For each piece the same value, 1 or -1, is assigned. Neighboring pieces takes different values where one piece takes the value of 1, and the other takes -1. The shape of the ring is shown in

Figure 3.7. Black regions take the value of -1, white regions are 1 and the grey regions are zero which means there is no watermark.

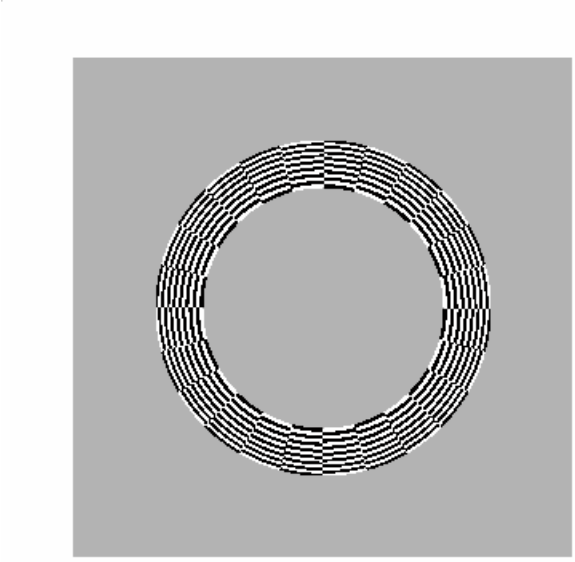


Figure 3.7 Shape of the watermark in DFT domain

The coefficients of the watermarked magnitude V'_M is:

$$V'_M(k_1,k_2) = V_M(k_1,k_2) + \alpha W(k_1,k_2)$$

α is a factor which determines the strength of the watermark. If the magnitude becomes negative, it is rounded to 0.

The DFT of a real signal has certain conjugate symmetry properties. The addition of a watermark to the magnitude of the DFT of the image does not ensure that the inverse DFT will produce a real image. To ensure that the DFT is real, the watermark must possess the following symmetry:

$$W_{k,l} = W_{N-k,N-l}, \quad \forall k,l \in [1,N]$$

Thus, the sectors must be selected properly to provide the symmetry. The watermarked image is obtained by computing the inverse DFT:

$$i' = IDFT(I'), \quad I' = (V'_M, V_P)$$

While detecting the watermark, the correlation ρ between the possibly corrupted coefficients V^*_M and the watermark W can be used to detect the presence of the watermark.

$$\rho = \sum_{i=1}^{N}\sum_{j=1}^{N} W(k_1,k_2)\, V^*_M(k_1,k_2)$$

Assuming that W and V_M are independent, identically distributed random variables and W has zero mean value, the mean of ρ is:

$$\mu = \begin{cases} \pi(R_2{}^2 - R_1{}^2)\alpha & if \quad W = W' \\ 0 & if \quad W \neq W' \\ 0 & if \quad no \ \ watermark \end{cases}$$

Again, we will use correlation coefficient (eqn. 2.19) for calculating the correlation value between watermark and the DFT coefficients. However, the threshold value formulation described in Chapter 2.2 (eqn. 2.20) was obtained for random watermark signals. The situation for symmetric watermark coefficients is different. Each symmetric DFT coefficients (which have the same magnitudes) are multiplied with same valued watermark key (because of the symmetry) to find the correlation value. This doubles the correlation between watermark and DFT coefficients. Therefore, for circularly symmetric watermarking in DFT domain, we will use two times of the threshold value described in equation 2.20.

$$\rho_{cc}\left(W, V_M^*\right) = \frac{\widetilde{W}.\widetilde{V}_M^*}{\sqrt{(\widetilde{W}.\widetilde{W})(\widetilde{V}_M^*.\widetilde{V}_M^*)}}$$

$$T_\rho = 2 \times 3.3 \sqrt{\frac{2M.\overline{\sigma^2_{\widetilde{V}_M^*}}}{(\widetilde{W}.\widetilde{W})(\widetilde{V}_M^*.\widetilde{V}_M^*)}}$$

$$\widetilde{V}_M^* = V_M^* - \overline{V}_M^*$$

$$\widetilde{W} = W - \overline{W}$$

where M is the number of watermarked coefficients.

The algorithm is tested with 1000 different (and symmetric) watermarks. Half of the experiments are made with true watermarks and the other half with wrong watermarks. The distribution of the correlation coefficient values are shown in Figure 3.8. The threshold is calculated for each experiment and the distribution of the threshold is shown on the figure.

The same watermarked image (Figure 3.9(a)) is tested with 1000 different watermarks. Only the 250th watermark is true this time. The correlation coefficient values for these watermarks are show in Figure 3.9 (b). The true watermark gives higher correlation value and it is the only one, which exceeds the threshold value.

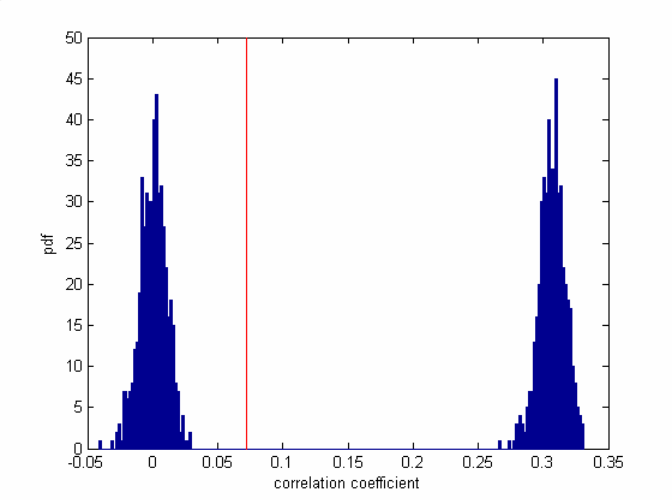
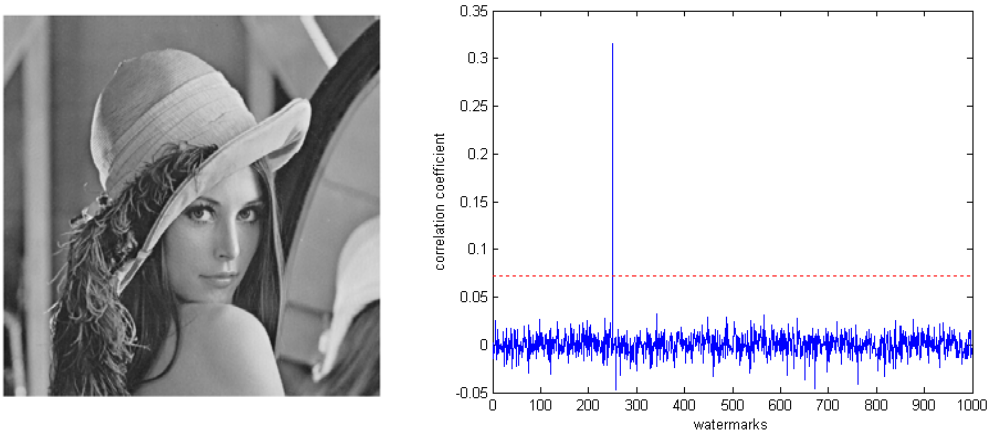


Figure 3.8 Distribution of the correlation coefficients and threshold values for DFT domain watermark detection for 1000 experiments. Half of them are computed with true watermarks and the other half with wrong watermarks.



(a) (b)

Figure 3.9 (a) Watermarked Lena image. (b) Detector response against 1000 different watermarks in DFT domain. Only the 250th watermark signal is true. Red line is the threshold value.

Since translations do not affect the DFT magnitude (eqn. 3.2), this algorithm is robust to translations. Rotation in spatial domain causes rotations in DFT magnitude by the same angle (eqn. 3.4). Since the watermark consists of *S* sectors having identical values, this allows the detection of the watermark even after a rotation in the range $\left[-\frac{\pi}{S},\frac{\pi}{S}\right]$ of the watermarked image. If a search of optimal rotation is performed that maximizes correlation value, the detection algorithm can be robust to any rotation angle. From geometrical transformation point of view, rotation around an arbitrary center is equivalent with rotation around the center of the image and translation. Thus, the algorithm is robust to rotation around an arbitrary center.

Scaling in the spatial domain causes an inverse scaling in the frequency domain (eqn. 3.7). Thus, if *NxM* is the size of the initial image and $[R_1,R_2]$ is the size of the watermarked ring (in the frequency domain), the size of the scaled image becomes $aN \times aM$ and the size of the watermark of the scaled image in the frequency domain remains unaltered, i.e. it still remains in $[R_1,R_2]$ while the total

size becomes $aN \times aM$. Thus, the mean value of the correlation does not change.

However, if we bring the scaled image to its original size (cropping if it is scaled up, or padding with zeros if it is scaled down), then the synchronization between watermark and watermarked coefficients will change and the watermark cannot be detected.

The circularly symmetric watermarking method is much simpler then RST invariant domain watermarking technique because no Fourier-Mellin transform is employed.

Licks *et al.* [20] uses random numbers in order to predefined ones and minus ones as the watermark signal values and they do not use sectors, but just a ring. However, in this case the watermark would not be circularly symmetric. In this approach, the algorithm is robust against linear translations only. They use a search procedure for detecting the watermark when the image is rotated, which is not a practical scheme because of the huge number of possibilities. The algorithm is also robust to scaling attack because of the same reasons as described for circularly symmetric watermarking algorithm.

3.2.3 Template Based Algorithms

Another method to gain robustness against geometrical attacks is to embed a synchronizer into the image, which is called a *template* in this method. The following algorithm is described by S. Pereira *et al* [21,22]. The watermark is composed of two parts, a template and a spread spectrum message containing the information. The template contains no information in itself, but is used to detect transformations undergone by the image. Once detected, these transformations are inverted and then the spread spectrum signal is decoded.

The watermark is embedded into the DFT coefficients between the radii R_1 and R_2 as a ring. As described in the previous algorithms, watermark embedding into the low frequencies creates fidelity problem, where embedding into the high frequency component will make it fragile against attacks such as low-pass filtering

or compression. Thus, R_1 and R_2 must be selected carefully to be robust against attacks while preserving the watermark invisibility. The watermark must fulfil the symmetry constraints to ensure the real valued image after inverse Fourier transform.

The watermark consists of a pseudo-randomly sequence generated by a secret key. Then the spread spectrum message inserted into the points in the DFT coefficients. The watermark embedding procedure is similar with the previously defined DFT algorithms. However, the difference is in the template algorithm, which is used to detect the amount of rotation and scaling.

The points of the template are uniformly distributed in the DFT domain with the low frequencies being excluded. The points are chosen pseudo-randomly as determined by a secret key. The strength of the template is determined adaptively as well. Pereira *et al*. [21] suggests that inserting points with equal strength to the local average value of DFT points plus one standard deviation yields a good compromise between visibility and robustness during decoding. Usually, the local average values at higher frequencies are lower then at higher frequencies. This makes the high frequency template points inserted less strongly.

To recover the image after possibly geometrical attacks, the local peak points are searched in the frequency domain. The template detection process is a point matching problem. Log-polar mapping and log-log mapping can be used in order to simplify the template detection algorithm. In the log-log mapping, the scaling and rotations are converted to translations. In log-log mapping the changes in aspect ratio becomes as translations.

In order to implement the whole algorithm, we will use the template scheme in FrFT domain watermarking algorithm, to gain robustness against geometrical attacks.

3.3 DWT DOMAIN APPROACH

In this section, we review a watermarking technique, which uses the DWT

domain for watermark embedding. This algorithm is developed by R. Dugad, K. Ratakonda, and N. Ahuja [23]. The watermark coefficients are added to the significant coefficients in the DWT domain and the method does not require the original image in the detection process. The amount of watermark added is adapted to the image.

The mark is a Gaussian sequence of psuedo-random real numbers matching size of the detail subbands. Although the watermark is embedded only to a few selected significant coefficients, using an image sized watermark fixes the locations that are manipulated.

Figure 3.10 shows a block diagram of the proposed algorithm. Three level DWT with Daubechies 8-tab filter is used. The low pass sub-band is picked out and watermark is added to the coefficients in the other (detail) sub-bands, which are above a given threshold T_1.

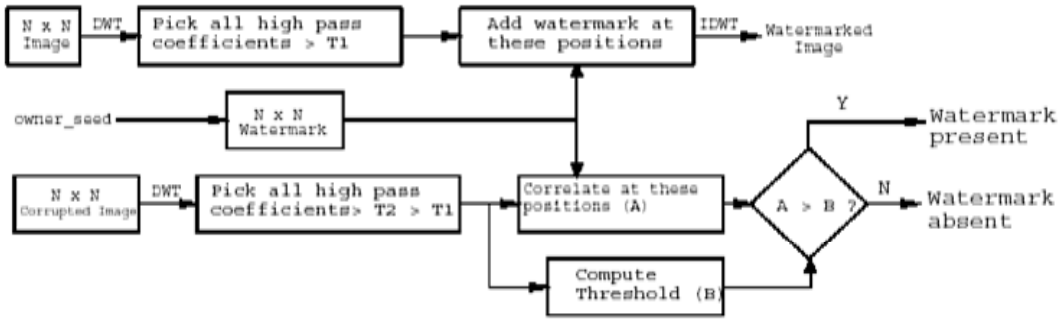


Figure 3.10 DWT domain watermarking algorithm. Top part shows the watermark casting and bottom part shows watermark detection [23].

This approach is different from Barni's [17], which fix the number of coefficients to be watermarked. This gives an adaptive selection to the amount of watermark added since smooth images have much fewer numbers of coefficients above a threshold compared to rough images.

No explicit visual masking is performed due to the time-frequency localization properties of DWT. The detail subbands, where the watermark is added, already contain edge related information of the image. Thus, adding the watermark

to significant coefficients in the detail subbands is equivalent to adding the watermark to only the edge areas of the image, which makes the the watermark invisible to the human visual system.

The watermark embedding equation is given as :

$$v_i' = v_i + \alpha |v_i| w_i \ ,$$

where *i* runs over all the DWT coefficients, which has magnitude above T_1 in detail sub-bands. v_i denotes the corresponding DWT coefficients of the original image and v_i' denotes the DWT coefficients of the watermarked image. w_i is the watermark value at the position of v_i.

For watermark detection, the correlation coefficient ρ_{cc} between the DWT coefficients of possibly corrupted watermarked image and the threshold, T_ρ,values are calculated as:

$$\rho_{cc} = \frac{\tilde{V}^* . \tilde{W}}{\sqrt{(\tilde{V}^* . \tilde{V}^*)(\tilde{W} . \tilde{W})}}$$

$$T_\rho = 3.3 \sqrt{\frac{2M . \overline{\sigma_{V^*}^2}}{(\tilde{V}^* . \tilde{V}^*)(\tilde{W} . \tilde{W})}}$$

$$\tilde{V}^* = V^* - \overline{\tilde{V}^*}$$

$$\tilde{W} = W - \overline{W}$$

where V^* denotes the vector of possibly corrupted DWT coefficients,W denotes the watermark vector and M is the number of such coefficients.

The algorithm is tested with 1000 different (and symmetric) watermarks. Half of the experiments are made with true watermarks and the other half with wrong watermarks. The distribution of the correlation coefficient values are shown in Figure 3.11. The threshold is calculated for each experiment and the distribution of the threshold is shown on the figure.

The same watermarked image (Figure 3.12 (a)) is tested with 1000 different

watermarks. Only the 250th watermark is true this time. The SNR value between the original and watermarked images is 35dB. The correlation coefficient values for these watermarks are show in Figure 3.12 (b). The true watermark gives higher correlation value and it is the only one, which exceeds the threshold value.

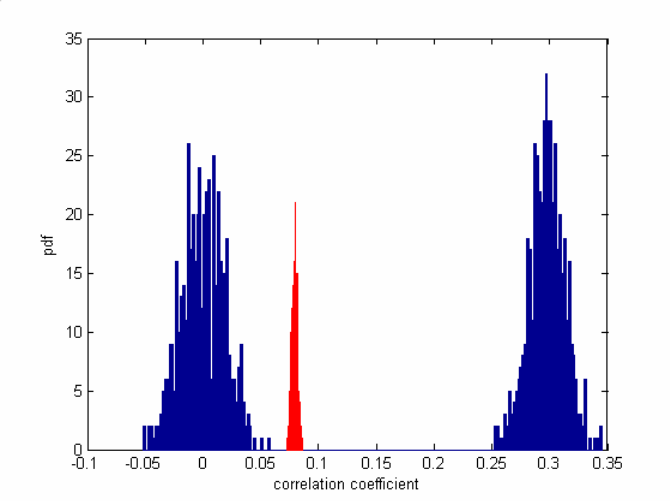
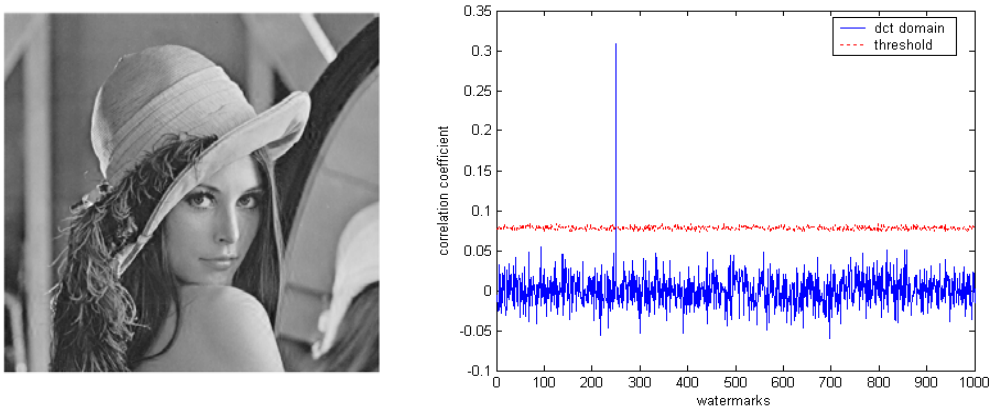


Figure 3.11 Distribution of the correlation coefficients and threshold values (red distribution) for DWT domain watermark detection for 1000 experiments. Half of them are computed with true watermarks and the other half with wrong watermarks.



(a) (b)

Figure 3.12 (a) Watermarked Lena image. (b) Detector response against 1000 different watermarks in DWT domain. Only the 250th watermark signal is true. Red line is the threshold value.

CHAPTER 4

FRACTIONAL FOURIER DOMAIN WATERMARKING

In this section, first, we will give brief information about fractional Fourier domain, which becomes more popular and find application areas especially in optics. Many researchers try to use different domains for watermark embedding and try to find solutions to the unresolved problems in this area. However, a perfect (or nearly perfect) algorithm for watermark embedding from the robustness point of view has not been found yet. In this chapter, we define a watermarking scheme that uses FrFT domain. In addition, we implement a masking algorithm by using the characteristics of the human Visual System, which increases the quality of the image after watermark embedding. Finally, we combine a template addition method with the watermarking algorithm to gain robustness against geometrical attacks.

4.1 FRACTIONAL FOURIER TRANSFORM

The Fourier transform (FT) is one of the most frequently used tools in signal analysis. It maps one-dimensional time signal $x(t)$ into a one-dimensional frequency function $X(w)$. Although the Fourier transform provides the signal's spectral content, it fails to indicate the time location of the spectral components, which is important, for example, when we consider non-stationary or time-varying signals. In order to describe these signals, time-frequency representations are used. The fractional Fourier transform (FrFT) is the general form of FT that maps a one-dimensional time signal into a two-dimensional function of time and frequency [24]. In recent years, the FrFT has attracted a considerable amount of attention, resulting

in many applications in the areas of optics and signal processing.

In time-frequency representations, two orthogonal axes, corresponding to time and frequency respectively, on a single plane are used (Figure 4.1). [25]. If a time-varying signal $x(t)$ is represented along the time axis, its Fourier transform $X(w)$ is represented along the frequency axis.

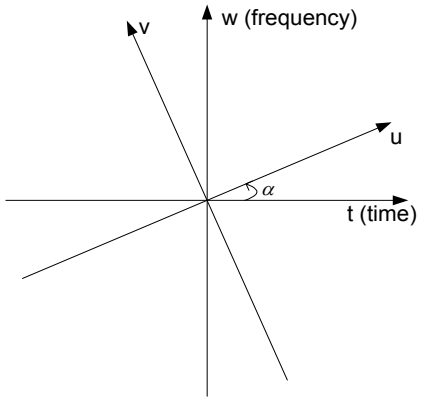


Figure 4.1 Time-frequency plane and a set of coordinates (u,v) rotated by an angle α relative to the original coordinates (t,w) [25].

The Fourier Transform operator, F, can be viewed as a change in the representation of the signal corresponding to a counter clockwise axis rotation of $\pi/2$ radians. Let $\{Ff\}(x)$ denote the Fourier transform of $f(x)$. Integer powers F^j of the operator $F \equiv F^1$ may be defined as its successive applications. Then, we have $\{F^2 f\}(x) = f(-x)$ and $\{F^4 f\}(x) = f(x)$. Then the a^{th}-order fractional Fourier transform $\{F^a f\}(x)$ of the function $f(x)$ may be defined for $0 < |a| < 2$ as [26]:

$$F^a[f(x)] \equiv \{F^a f\}(x) \equiv \int_{-\infty}^{\infty} B_a(x,x') f(x') dx' ,$$
$$B_a(x,x') \equiv A_\phi \exp\left[i\pi\left(x^2 \cot\phi - 2xx' \csc\phi + x'^2 \cot\phi\right)\right], \quad (4.1)$$
$$A_\phi \equiv \frac{\exp\left(-i\pi \operatorname{sgn}(\sin\phi)/4 + i\phi/2\right)}{|\sin\phi|^{1/2}}$$

where

$$\phi \equiv \frac{a\pi}{2} \quad (4.2)$$

and I is the imaginary unit. The kernel approaches $B_0(x,x') \equiv \delta(x-x')$ and $B_{\pm 2}(x,x') = \delta(x+x')$ for $a = 0$ and $a = \pm 2$ respectively. The definition is easily extended outside the interval $[-2,2]$ by remembering that the fractional Fourier transform operator is additive index, that is, $F^{a_1} F^{a_2} = F^{a_1+a_2}$.

The discrete form of fractional Fourier Transform is described in [26,27,28,29]. The fast computation of the fractional Fourier transform is described in [26,29].

Logarithmic magnitude of two-dimensional fractional Fourier transform applied image *Lena* is shown in Figure 4.2. The transformation angle $(a_1 = a_2)$ varies between zero and one. It can be seen that, at smaller angles, the image is closer to the time domain and the original image can be determined from the resulting two-dimensional coefficients. However, when the angle increases the transformation gets closer to the Fourier domain and the resulting coefficients show similarities with Fourier domain coefficients.

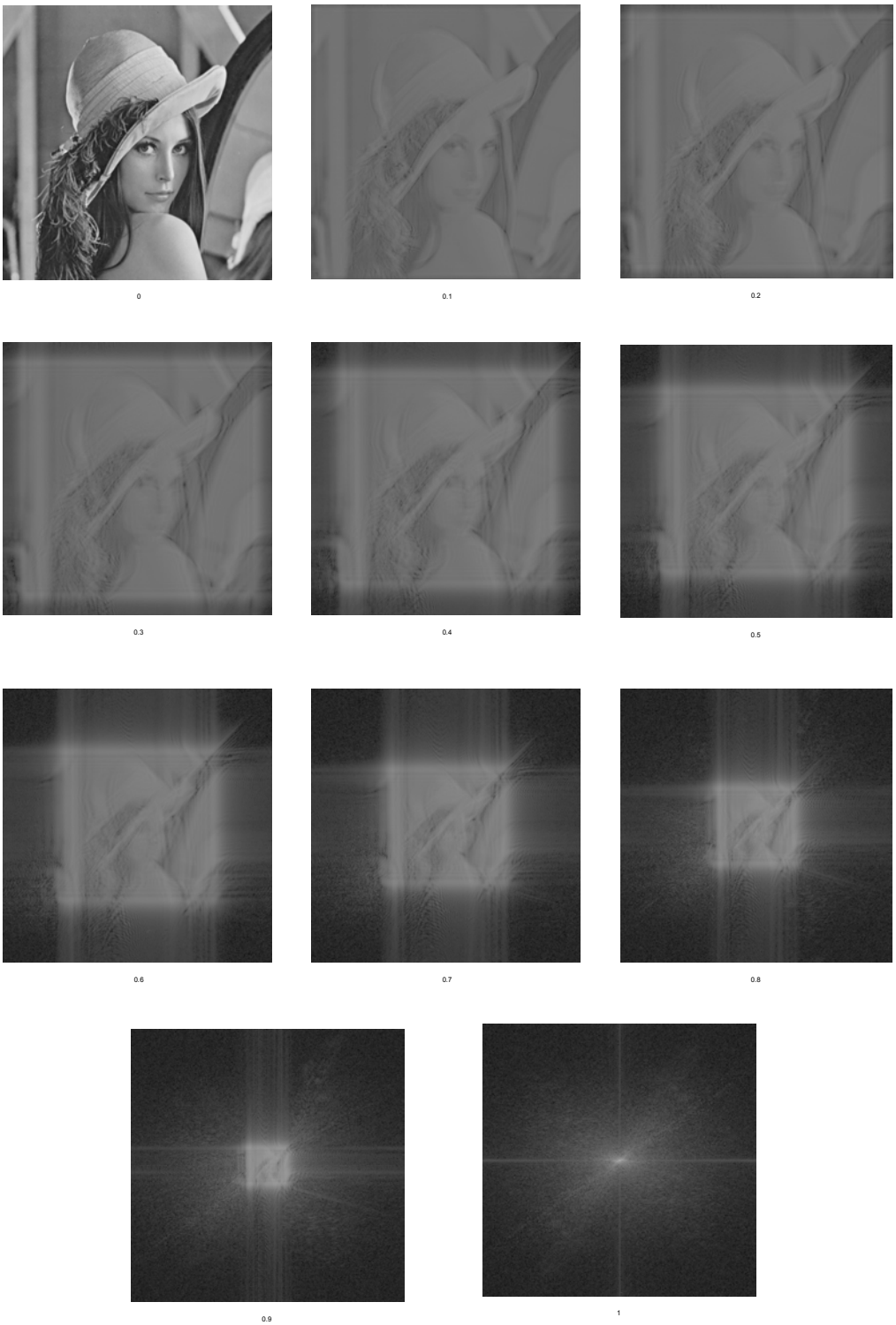


Figure 4.2 2D fractional Fourier computation of test image *'Lena'*. The transformation angles are $a_1 = a_2 = 0.1,0.2,....,1.0$ respectively.

4.2 WATERMARKING ALGORITHM

The first attempt to use the FrFT domain in watermarking area came from I.Djurovic, S. Stankovic and I. Pitas as described in [5]. In Chapter 3, we presented some frequency domain techniques, which use different transformation domains for watermark embedding. However, in this section, we will present an algorithm which uses a combination of time and frequency domain by using the FrFT.

It is a one-bit public watermarking scheme and the watermark is embedded according to the spread spectrum approach. The FrFT domain stands between time and frequency domain separated by transformation angles (a_x, a_y). Increasing the angles makes the time domain signals to get closer to the frequency domain. However, it is pointed that the frequency domain techniques are more robust according to the time domain techniques [3,4]. Thus, we will select the transformation angles to become closer to the frequency domain.

For watermark embedding, the two-dimensional FrFT for angles, (a_x, a_y), of a grey scale image is computed. Then the FrFT coefficients are ordered in increasing sequence. The first and highest L coefficients are omitted and the watermark is embedded in the next M coefficients. Selecting the coefficients for watermark embedding is a critical issue such as if the watermark were embedded in the highest coefficients, it would produce significant image deformation, while if it were embedded in the lowest coefficients it could be cleaned by lossy image compression or low pass filtering. The watermark is embedded as:

$$v_i' = v_i + \alpha \left(k_i' \left| \mathrm{Re}\{v_i\} \right| + j.k_i'' \left| \mathrm{Im}\{v_i\} \right| \right) \qquad i = L+1, L+2, ..., L+M \tag{4.3}$$

where (k_i', k_i'') represents the real-valued watermark key coefficients and v_i represents the FrFT coefficients for watermark embedding where $i = L+1, L+2, ..., L+M$. α is a factor which determines the strength of the watermark.

Original image is not needed for watermark detection. The knowledge of

watermark key and positions are needed for reliably detecting the mark. The correlation between the watermark and FrFT coefficients can be calculated as:

$$\rho = \sum_{i=L+1}^{L+M} \left[k_i' - jk_i''\right] v_i^* \tag{4.4}$$

with a chosen threshold. Here v_i^* denotes the FrFT coefficients of the possibly attacked target image. The threshold value can be selected as:

$$T_\rho = \frac{2}{M} \sum_{i=L+1}^{L+M} \left| v_i^* \right| \tag{4.5}$$

However, we will use correlation coefficient and the detection technique described in section 2.2 for computing the correlation and the corresponding threshold values as:

$$\begin{aligned} \rho_{cc}(W, V^*) &= \frac{\widetilde{W}.\widetilde{V}^*}{\sqrt{(\widetilde{W}.\widetilde{W})(\widetilde{V}^*.\widetilde{V}^*)}} \\ T_\rho &= 3.3\sqrt{\frac{2M.\overline{\sigma_{v^*}^2}}{(\widetilde{W}.\widetilde{W})(\widetilde{V}^*.\widetilde{V}^*)}} \\ \widetilde{V}^* &= V^* - \overline{V}^* \\ \widetilde{W} &= W - \overline{W} \end{aligned} \tag{4.6}$$

where W is the watermark vector and V^* is the possibly corrupted FrFT coefficients vector as:

$$\begin{aligned} W &= k_1' - j.k_2'' \\ V^* &= V^* + \alpha\left(k_1' \left| \mathrm{Re}\{V^*\} \right| + j.k_2'' \left| \mathrm{Im}\{V^*\} \right| \right) \end{aligned} \tag{4.7}$$

The algorithm is tested with 1000 different watermarks. (512x512 sized *Lena* image is used as stego-image. The watermarking variables are: *L=40000,*

$M=1000, \alpha = 14$, $a_1 = a_2 = 0.85$, k' and k'' have unit variances with zero mean). The SNR value becomes 35dB between original and watermarked images. Half of the experiments are made with true watermarks and the other half with wrong watermarks. The distribution of the correlation coefficient values are shown in Figure 4.3. The threshold is calculated for each experiment and the distribution of the threshold is also shown on the figure.

The same watermarked image (Figure 4.4 (a)) is tested with 1000 different watermarks. Only the 250th watermark is true this time. The correlation coefficient values for these watermarks are show in Figure 4.4 (b). The true watermark gives higher correlation value and it is the only one, which exceeds the threshold value.

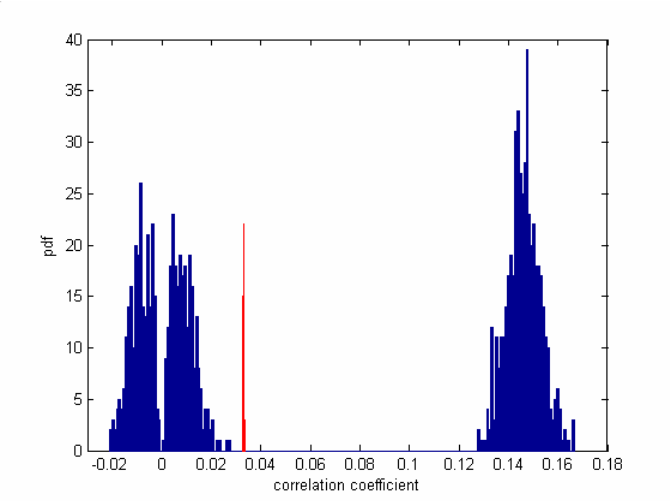
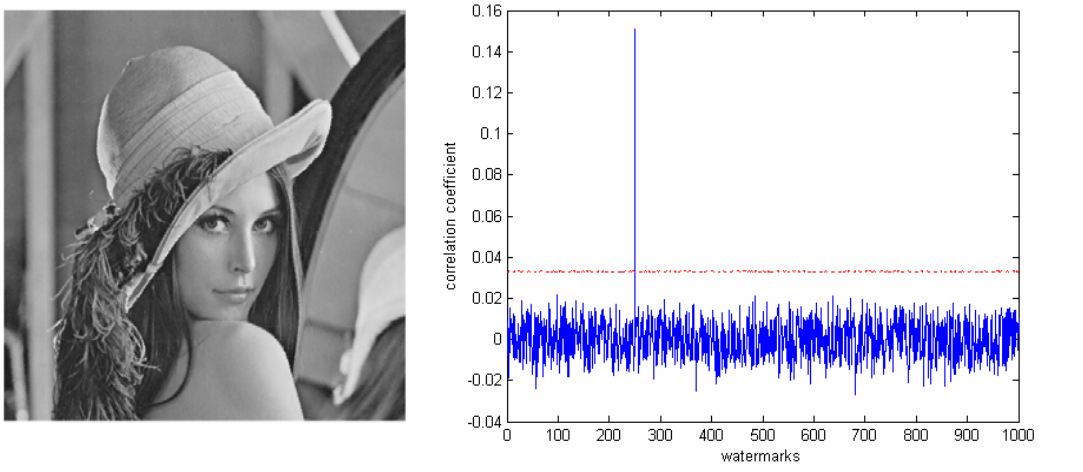


Figure 4.3 Distribution of the correlation coefficients and threshold values for FrFT domain watermark detection for 1000 experiments. Half of them are computed with true watermarks and the other half with wrong watermarks.



(a) (b)

Figure 4.4 (a) Watermarked Lena image. (b) Detector response against 1000 different watermarks in FrFT domain. Only the 250th watermark signal is true. Dotted line is the threshold value.

One advantage of the watermarking algorithm is the transformation angles where detection of the watermark signal requires the knowledge of angles as well as the watermark key. The detector response over transformation angles $a_1 = a_2 = 0.01,0.02,..,1$ with true watermark signal is shown in Figure 4.5. It is shown that the watermark detection can only be performed at the true angles. The watermark key consists of watermark keys (k_i', k_i''), positions of embedded coefficients and the transformation angles (a_1, a_2). By using different angles, more watermarks can be created than in DFT or DCT domain.

Calculation complexity of the procedure for watermark embedding and detection is not significantly increased, since there are standard fast algorithms for computing the FrFT [27].

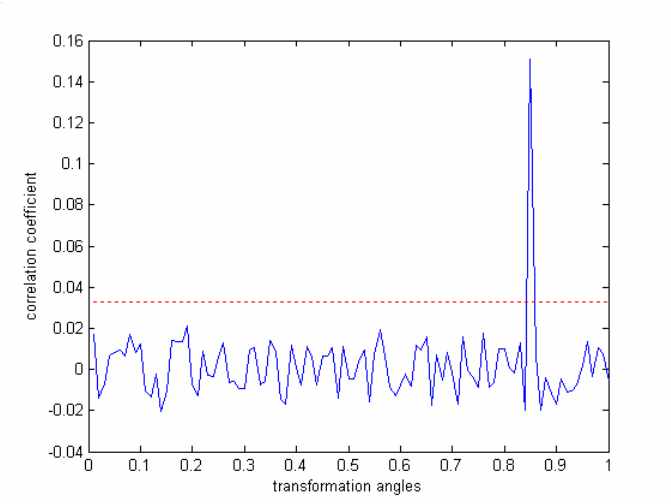


Figure 4.5 Detection of watermark signal using different transformation angles. The true transformation angles are 0.85. Dotted line shows the threshold.

To increase the perceptual quality, a visual masking is applied over the watermarked image. The masking algorithm is the same with the method as described in chapter 3.1, which is applied on the watermarked image in DCT domain. Image masking allows us to increase the watermark strength, α. To show the effect of visual masking, we make two experiments; watermarking with visual masking and without visual masking. While we were doing this, we kept the number of watermarked coefficients and their positions constant, change the watermark strength α and try to reach to the same SNR value. We compared the quality indexes, Q. While the SNR is 35dB for both experiments, the quality index become 0.996 (max quality index can be 1) with masking applied algorithm and become 0.884 without masking. Although the difference can be seen small, this difference make detectable perceptual degrading, which is not acceptable for watermarking algorithms.

4.3 TEMPLATE ADDITION ALGORITHM

The FrFT domain watermarking, itself, is not robust against geometrical attacks such as rotation and scaling because of the corrupted synchronization between embedded coefficients and watermark signal. However, to gain robustness

against geometrical attacks, a template is added to the watermarked image. Template addition mechanism is described in Chapter 4.3. However, we have changed the template addition and detection method while keeping the main idea behind the template.

A template is a local peak, which does not contain any information in itself, but is merely a tool used to recover possible transformations in the image. Once detected, these transformations are inverted then the spread spectrum signal is decoded.

A template addition algorithm is proposed in [21,22]. The template is inserted in Fourier Domain. Because of the rotation and scaling property (eqn. 3.7 and 3.8) of the Fourier Transform, the local peaks will rotate in the case of a rotation and they will reciprocally scale in the case of image scaling. By calculating the amount of rotation and scaling of the local peaks, anyone can determine the rotation and scaling attack on the image.

However, it is not the case for the translation attack. The local peaks do not effected from any translations because of the translation property (eqn. 3.6) of the Fourier domain. In [21,22], the watermark is also embedded in the Fourier Domain. This makes the algorithm naturally robust against translations. Since we embed the watermark in FrFT domain, use of template will not gain robustness to translations, because it will not supply any knowledge of translations.

Although in [21,22] it is said that 8 points and 25 points template works best, we have used 4 points as the template. Since each point needs its pair to not disturb the symmetry of the Fourier transform, these 4 points are formed as pairs and the template can be said to be formed from 2 pairs of local peaks.

The strength of the template is determined adaptively as well. The strength of the template is determined according to the magnitudes of the DFT coefficients around the template location. Note that that the template must be embedded in the mid-frequency region to be robust against filtering and compression attacks. Also it is critical that the points be inserted strongly enough so that they remain peaks after interpolation errors from possible transformations. The template embedding

algorithm is described as follows:

1. After obtaining the watermarked image, the magnitudes of the DFT coefficients are calculated.
2. The locations of template points are calculated. Each template point is inserted on one of the DFT coefficient's magnitude. The value of this coefficient is increased to become a local peak point between neighboring coefficients. One pair of template points are inserted on the first diagonal and the second pair are inserted on one the second diagonal. These pairs have different distances to the center. This provides detecting the peak points after transformation and determine which point it is. Figure 4.6 shows a diagram of the template points. T_1 and T_2 are the template points on the first diagonal and the distance between the points and the center is R_2. The location of the T_2 is the symmetric point of the location of the T_1. As similar, T_3 and T_4 are the template points on the second diagonal which are also symmetric Fourier coefficients.

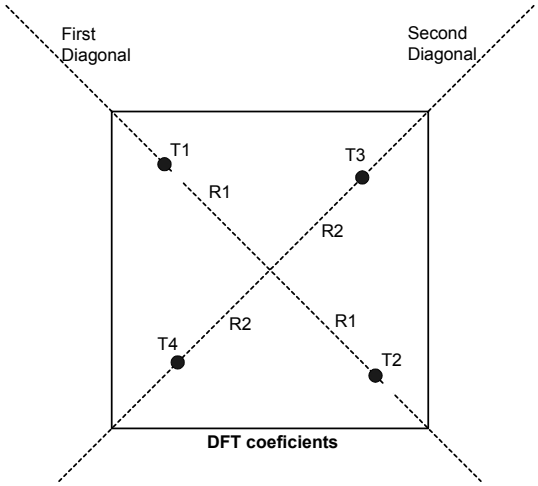


Figure 4.6 The locations of the template points on the magnitudes of the DFT coefficients. T_1, T_2, T_3 and T_4, are the template points located on the diagonals and R_1 and R_2 are the distances of these points to the center.

3. The magnitudes of the template points are increased to become a local peak at the neighboring area. This area is identified by α, d_1 and d_2 shown as

shaded region in Figure 4.7. The magnitude of the template point is increased to the n times of the highest coefficient's value in this region. All the template points undergone the same procedure except α, d_1 and d_2 change for each pair.

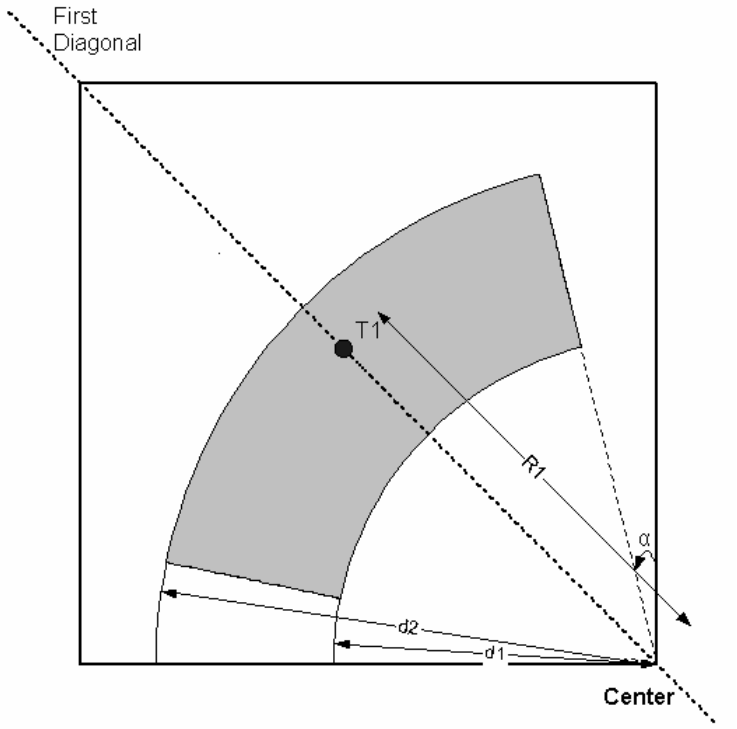


Figure 4.7 First quarter of the magnitudes of the DFT coefficients and the location of the template T_1. The magnitude of the template point is increased to become a local peak in the shaded region. This region is described as the inner area between the lines making α degrees form the center and between the circles with radiuses d_1 and d_2 from the center.

The logarithmic values of the magnitudes of the template inserted DFT coefficients are shown in Figure 4.8. We select $R_1 = 165, R_2 = 160, \alpha = 15^0, d_1 = 140$ and $d_2 = 200$ for the 512x512 sized image.

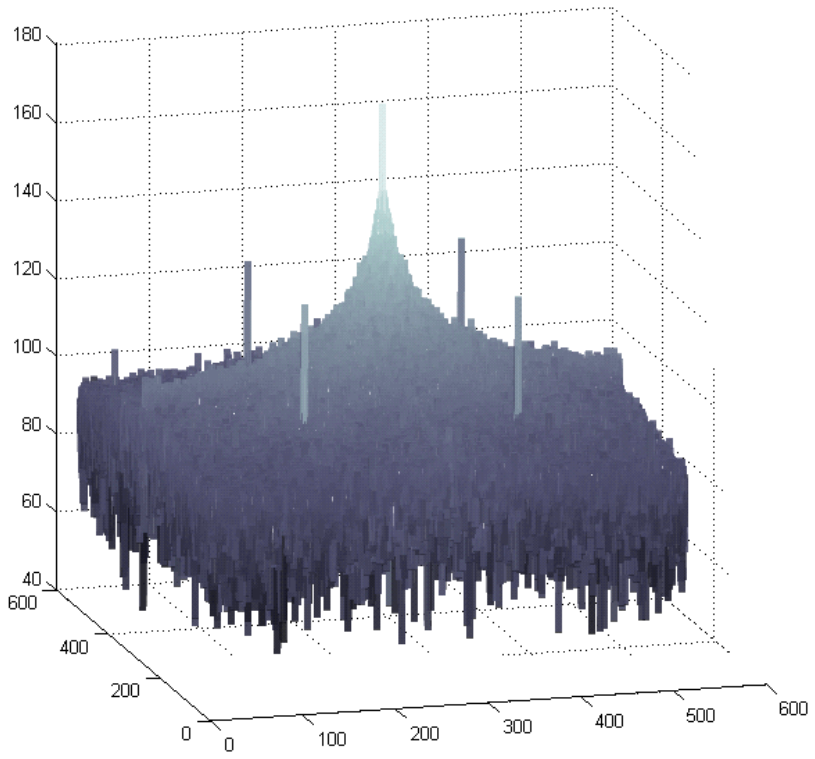


Figure 4.8 Magnitudes of the DFT coefficients of the template inserted image. The four peak points around the center (dc) coefficient shows the locations of template points.

4. Inverse DFT is applied to the template inserted coefficient to obtain the original image with template. In Figure 4.9 the watermarked version and the template inserted version of the Lena image are shown. The SNR value between the original image and the watermarked image is about 35dB where the SNR value between the original image and the template inserted copy is about 33dB. The 2dB difference between the watermarked and template inserted copies show that the template does not degrade the image perceptuality so much.

(a) (b)

Figure 4.9 (a) Watermarked image (b) Template inserted into the watermarked image. The SNR value of the watermarked image is about 35dB and the SNR value of the template inserted image is about 33dB.

To detect the template, the DFT magnitude of the image (which is possibly transformed) is searched to find the local peaks. After the template locations are detected, they are matched with the original locations and the transformations are calculated by comparing these two template's locations. The template detection process constitutes the most complex part of the template algorithm. Most of the algorithms are produced experimentally by eliminating the detection errors caused by geometrical translations. The template detection process is described below:

1. Calculate the DFT magnitude coefficients of the possibly transformed image.
2. Seperate the coefficient matrix into *m* by *m* sub-blocks. Find the mean and maximum values of each block. The blocks located at the center area are discarded because the peak located around the center area are arised from the dc and low frequency points which have very high magnitudes for most of the images. Calculate the max/mean value for all of the remaining blocks.
    i. Sort the blocks according to the max/mean values

ii. Sort the blocks according to the maximum values

iii. Take the *k* blocks having the maximum max/mean value.

iv. Find the maximum four blocks having the largest max/mean values and the four blocks containing the largest maximum valeus in these *k* blocks. If these double four blocks address the same locations the four maximums in these blocks are selected as template points.

v. If there are less than four blocks obtaining the above criteria, only these points are selected as template points and the algorithm continues to search the remaining template points.

3. The *k* maximum blocks in the *k* sub-blocks are seperated into four regions according to their locations as shown in Figure 4.10. Since the template points are located in four different regions, the template points also spreaded in four different regions after any geometrical transformation. So the remaining template points are searched in the regions which are not in the same region with the detected template points. The maximum points in the *k* blocks matching this criteria are selected to be possible template points.

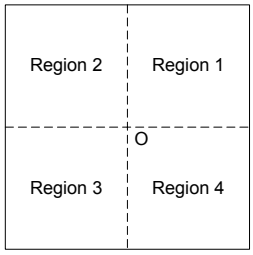


Figure 4.10 Four regions according to the center of the DFT coefficients.

4. Since we know the original template locations, we can calculate the distance of these locations to the center and the distance between these points. The ratios between these distances are calculated. For each possible template points, the distance between the found template points are calculated. The possible template points having the same ratio with the original template

points are selected as remaining template points. The four resolved template points are determined at the end of this step.

5. Finding the template locations are not enough to determine the amount of translation, we also need to match the original points with the detected points.By considering the distances of the original and detected template points, the matching can be estimated. After matching the points, determining the translation is simply solving linear equations for finding the amount of scaling and rotation of the image.
6. The image is transformed back to obtain the watermarked image by using the rotation angle and scaling ratio.

In the following experiment the template inserted image is rotated for 27^0 (Figure 4.11). It is not scaled and the outer side which exceeds the original side is cropped.

Figure 4.11 The "Lena" image is rotated for 27 degrees and the outer side which exceeds its original size is cropped.

The logarithmic magnitudes of the DFT coefficients are shown in Figure 4.12 (a). The white points show the high valued points. Because of the black region around the image occurred after rotation, continuous lines appear on the DFT image. This is because the sharp crossing from pixel values to the black region, which has zero magnitude. After the template detection algorithm, the recovered template locations are shown in Figure 4.12 (b).

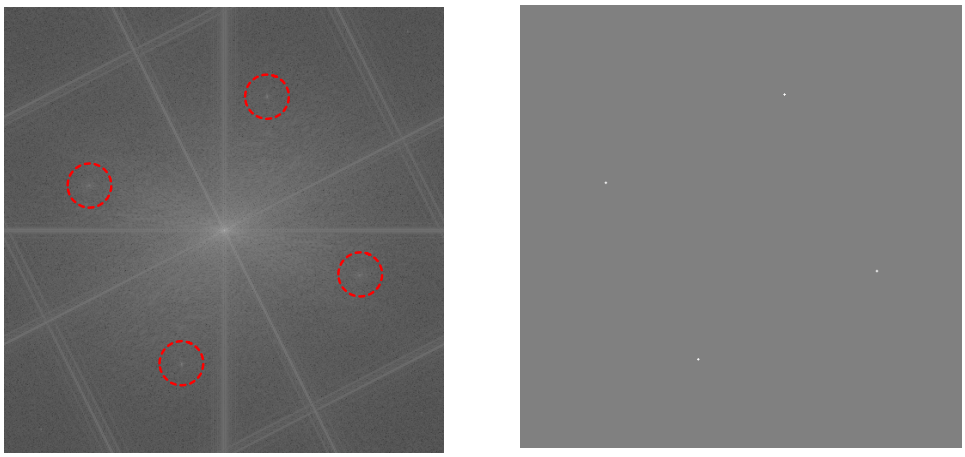
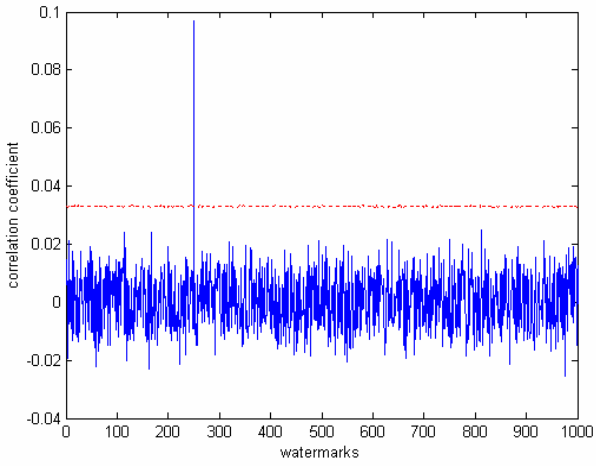


Figure 4.12 (a) The logarithmic magnitudes of the DFT coefficients of the rotated image. The red circles show the positions of the template points. (b) The positions of the recovered template points found by the template detection algorithm.

After recovering the template, the rotation angle and scaling factor is calculated as described in step 5. The algorithm finds the rotation angle as 26.95^0 and the scale factor as 1.002. These results show the success of the algorithm. After finding the rotation angle and scale factor values, these must be applied on the attacked image to fix the geometric distortion. In Figure 4.13 (a) the repaired image according to these values is shown. In Figure 4.13 (b) the detector response of the repaired image against 1000 different watermarks is shown. The watermark detector is successful to find the watermark after geometrical attack and recovery processes.



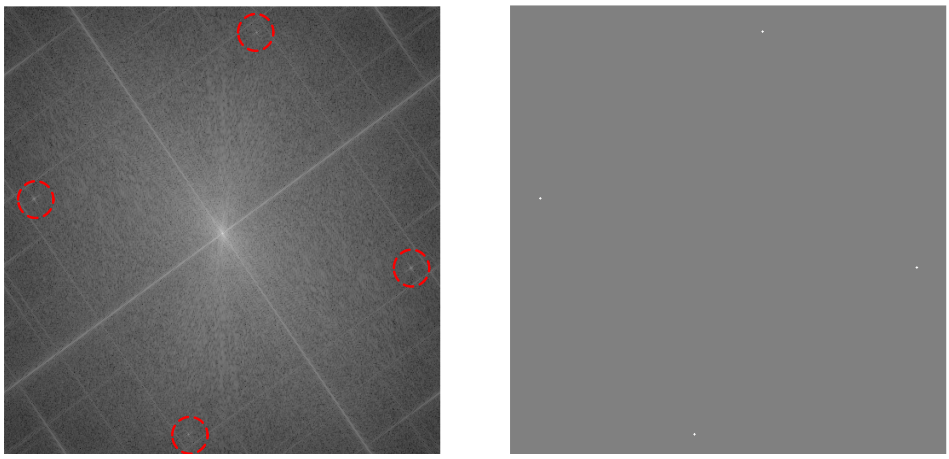
(a) (b)

Figure 4.13 (a) The attacked image is repaired according to the template. (b) Watermark detection after repairing the attacked image. The true watermark is the 250^{th} one.

In the following experiment the watermarked image is rotated for 55^0 but in this case we will not crop the outer region which exceeds the original size. Instead, we will resize the image to fit into its original size. The rotated and scaled image is shown in Figure 4.14.

Figure 4.14 The image is rotated for 55 degrees and it is scaled to fit its original size.

The logaritmic magnitudes of the DFT coefficients are shown in Figure 4.15 (a). The circled locations shows the positions of the template. After the template detection algorithm, the recovered template locations are shown in Figure 4.15 (b). Although the template points are inserted at the same location with the previous example, it is seen that the template points become closer to the outside of the DFT image. This proves the scaling property (eqn. 3.7) of the Fourier transform, which says if the size of the image is decreased, the DFT coefficients get away from the center in the same ratio. This ratio gives us the scaling factor of the translation.



(a) (b)

Figure 4.15 (a) The logarithmic magnitudes of the DFT coefficients of the rotated and scaled image. The circles shows the positions of the template points. (b) The positions of the recovered template points found by the template detection algorithm.

After recovering the template, the rotation angle and the scaling factor is calculated. The algorithm find the rotation angle as 55.00 degree and the scale factor as 0.72. The repaired image according to these values is shown in Figure 4.16 (a). The detector response against 1000 different watermarks are computed after repairing the image (Figure 4.16 (b)). The watermark detector is again successful to find the watermark after rotation and scaling attacks. The detection magnitude is smaller when it is compared to the previous example. As expected, it is because of the interpolations made when scaling and rescaling the image.

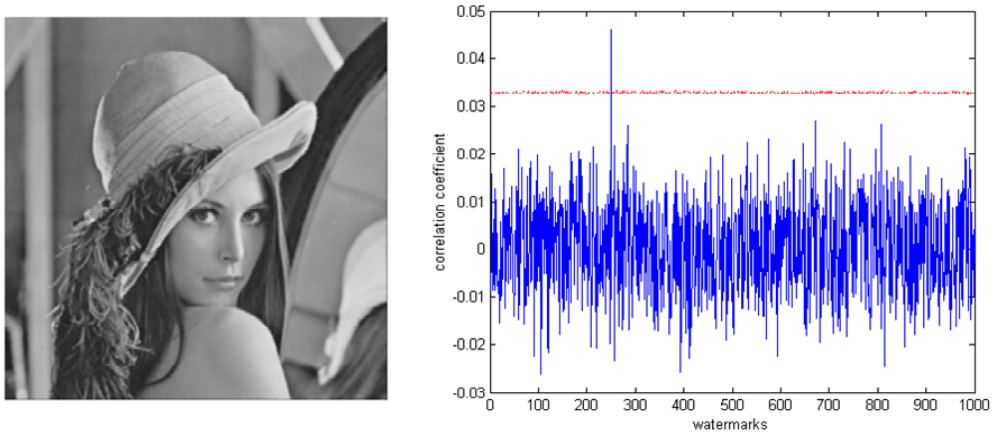


Figure 4.16 (a) The attacked image is repaired according to the template. (b) Watermark detection of 1000 different watermarks after repairing the attacked image. The true watermark is the 250th one.

In the next chapter the detection of the watermark is handled with more details against several types of attacks and the results are compared with other watermarking algorithms.

CHAPTER 5

EXPERIMENTAL RESULTS

In this Chapter, the experiment results for four watermarking algorithms are presented. These algorithms described in Chapters 3 and 4, which are DCT domain technique [17], circularly symmetric watermarking technique in DFT domain [19], DWT domain technique [23], and FrFT domain technique [5].

The experiments are involved in three groups. The first group includes the removal attacks such as compression and low pass filtering. The aim of these attacks is to decrease the energy of the watermark, thus, it cannot be detected by the watermark detector. In the second group, we will implement geometrical attacks on the algorithms. The geometrical attacks will not decrease the energy of the watermark; rather, it corrupts the synchronization between the watermark signal and its locations. In the third group of experiments, we will implement multiple attacks that are combination of more than one attack type. The details of these attacks are given in Table 5.2.

To provide a fair experiment, we keep the watermark energy constant by properly adjusting the watermark strength, α. After watermark embedding, the PSNR values become about 43.4dB. To test the efficiency of the template against geometrical attacks, we include template insertion scheme in experiments on geometrical attacks. The template is inserted only into the FrFT domain watermarked image. Although, the template creates some more attenuation after the watermarking process, we did not try to equalize the PSNR value (of the template inserted image), because it would not be fair while comparing the watermarking techniques. Therefore, we allow the template to decrease the PSNR value. The

parameters of the algorithms and the attenuation levels after watermarking process are given in Table 5.1. It is seen that, the template decreases the PSNR value by 2.9dB additionally after watermarking.

Table 5.1 Parameters of the watermarking algorithms and corresponding image quality metrics.

	Parameters	**PSNR (dB)**	**Quality Index, Q**	**SNR (dB)**	**MSE**
DCT	L = 16000 M = 16000 $\alpha = 0.68$	43.4	0.991	37.8	2.96
DFT	$R_1 = 70$ $R_2 = 150$ $S = 15^0$ $\alpha = 1950$	43.4	0.933	37.7	2.97
DWT	T1 = 40 T2 = 50 $\alpha = 0.149$	43.4	0.985	37.7	2.97
FrFT	L = 40000 M = 10000 $\alpha = 2$ $a_1 = a_2 = 0.85$	43.4	0.990	37.7	2.98
FrFT with template	$R_1 = 165$ $R_2 = 160$ $\theta = 15^0$ $d_1 = 140$ $d_2 = 200$	40.5	0.927	34.8	5.86

The effect of the visual masking in DCT and FrFT domain techniques on the fidelity can be detected from the quality index, Q. Although, Q values are close, small variances in quality are quite important for watermarking process.

Table 5.2 List of attacks performed on the test images.

Attacks		**Options**
Removal Attacks		
JPEG Compression		Quality : 100,95,90,…,10,5
Low Pass Filtering	Filter sizes: 3x3, 5x5, 7x7	Filter sizes: 3x3, 5x5, 7x7

	Filter sizes: 3x3, 5x5, 7x7	Filter sizes: 3x3, 5x5, 7x7
	Filter Size 3x3	Filter Size 3x3
Sharpening Filter		Filter Size 3x3
Gaussian Noise		Noise Variance : 50, 100, 150, 200, 250, 500, 1000, 2500, 5000, 7500, 10000
Geometric Attacks		
Cropping		% of original image: 5, 10, 15, 25, 35, 45, 60, 75, 90
Translation		Horizontal and vertical shift : (1,0), (0,1), (1,1), (5,5), (-10,10), (30,70), (-80,-175), (-120,50), (-256,-256)
Rotation & Cropping		Rotation angles (degree) : -80, -70, -60, -50, -40, -30, -20, -10, -5, -1, 1, 5, 10, 20, 30, 40, 50, 60, 70, 80
Rotation & Scaling		Rotation angles (degree) : -80, -70, -60, -50, -40, -30, -20, -10, -5, -1, 1, 5, 10, 20, 30, 40, 50, 60, 70, 80
Scaling		Scale Factor : 0.5, 0.8, 0.9, 0.95, 0.975, 1.025, 1.05, 1.1, 1.2, 1.5
Multiple Attacks		
Noise Addition with Filtering		Noise Variance : 50, 100, 150, 200, 250, 500, 1000, 2500, 5000, 7500, 10000 Filter : 3x3 median filter, 3x3 averaging filter
Rotation & Crop with Noise Addition		Rotation angle: 60 degree Noise Variance : 50, 100, 150, 200, 250, 500, 1000, 2500, 5000, 7500, 10000
Rotation & Crop with JPEG Compression		Rotation angles : : -80, -70, -60, -50, -40, -30, -20, -10, -5, -1, 1, 5, 10, 20, 30, 40, 50, 60, 70, 80 JPEG Quality : 70 %

We perform three sets of experiments for each attack. The first set aims to compare the algorithms. To compare the algorithms, we define a term named *margin*, which is formulated as:

$$margin = \frac{correlation \quad coefficient}{threshold} \tag{5.1}$$

The margin value provides a measure for available portion of the algorithm after the attack, in other words, if the margin is high, there is a long way to lose the watermark. If the margin is more than one, then it means the watermark is detected, however, if it is less then one, then it is not detected. It can be seen that, margin equalizes the threshold value to one and, by keeping the same ratio between the correlation coefficient and the threshold, the correlation value is adjusted. This helps us to compare the algorithms, because each algorithm would have different correlation and threshold values. On the other hand, since we are calculating the threshold value using the attacked image, we find different threshold values for the same algorithm against different attacks, which improves the need for such a metric.

In the second set, we give *confidence check* results [4]. The confidence check operation tries to find the correlation between a watermarked image and a set of possible pseudorandom watermark patterns. Ideally, there must be only one watermark pattern that the correlation value for it exceeds the threshold, which is the true watermark. It is a good measure to detect the false positives and negatives. For each attack, the confidence check of the algorithm, until the point that the watermark survives against the attack, is presented by plots. In the confidence check plots, we give the exact correlation coefficient and threshold values. Thus, the threshold value is computed by a small distribution.

The third set of experiments aims to show the effect of different FrFT angles on the performance of the FrFT domain algorithm against an attack. For an attack, we execute the algorithm for different transformation angles and present a plot of the margin against the transformation angles. To be fair for different transformation angles, we adjust the watermark strength α, to keep the PSNR value around a

constant value (43.4dB). For different transformation angles, the watermark strength is shown in Table 5.3.

Table 5.3 Watermark strength values for different FrFT angles.

Transformation angles, $a_1 = a_2$	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45
Watermark strength, α	0.223	0.229	0.227	0.219	0.209	0.202	0.194	0.187	0.179
Transformation angles, $a_1 = a_2$	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
Watermark strength, α	0.170	0.166	0.162	0.167	0.204	0.456	1.235	2.0	2.585
Transformation angles, $a_1 = a_2$	0.95	1.0	1.05	1.10	1.15	1.20	1.25	1.30	
Watermark strength, α	2.98	3.14	2.96	2.425	1.49	0.455	0.295	0.218	

There are some benchmarking tools, which are widely used by the watermarking researchers such as Stirmark [32]. These benchmark tools give out various attacked versions of the watermarked image automatically. However, in the experiments, we use MATLAB, since it is a very efficient tool in recursive operations. Thus, we were able to implement many types of attacks in a short time. However, we try to implement most of the attacks that are available in benchmarking tools.

We have also tested the execution time for watermark embedding and detection phases of each algorithm with the same computer to compare their computational cost. The results are given in Table 5.4. Visual masking operation in the DCT and FrFT domain algorithms are the main source of the lag in these algorithms.

Table 5.4 Execution times of algorithms.

Execution Time	DCT	DFT	DWT	FrFT
with visual masking	22.6 sec.	NA	NA	26.9 sec
without visual masking	2.6 sec.	1.3 sec.	2.2 sec.	5.6 sec

5.1 EXPERIMENTS ON REMOVAL ATTACKS

5.1.1 JPEG Compression

JPEG is currently one of the most widely used compression algorithm for images. Image watermarking systems should be resilient to some degree of compression.

In the experiments, we change the JPEG quality from 100 (no compression) to 5. A compressed watermarked image (by using FrFT domain algorithm) by quality factor 5 is given in Figure 5.1.

Figure 5.1 JPEG compression applied on FrFT domain watermarked image. JPEG Quality is 5.

The margin comparisons of algorithms are shown in Figure 5.2. Although, the DFT and FrFT domain techniques are more successful in higher compression qualities, it is very important to stand attacks with lower compression qualities. The DFT domain technique lost the watermark after the quality is lower than 15. Although, The FrFT domain technique can stand all experimented compression rates, the DCT and DWT domain techniques have more margins after experiments

with low compression quality.

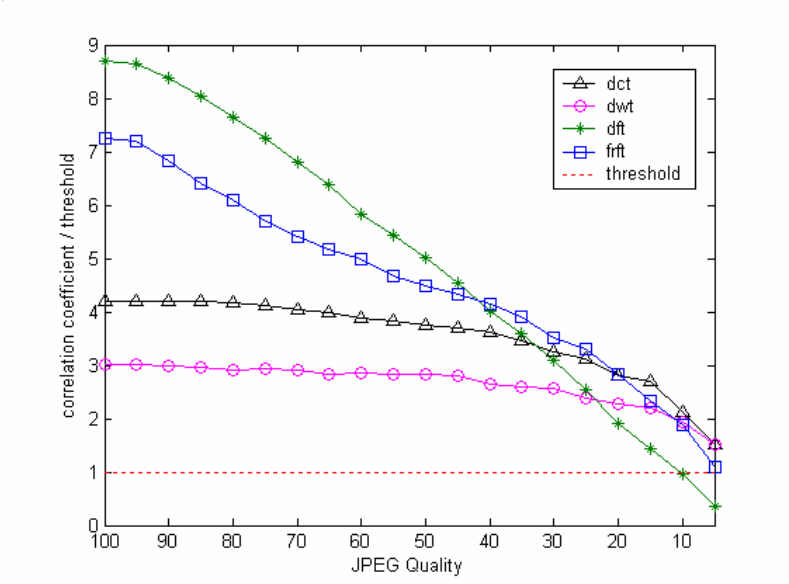


Figure 5.2 Comparison of watermarking algorithms against JPEG compression.

The DFT and FrFT domain watermarking algorithms show similar responses to the compression attack, i.e. the correlation decrease rate is more among other algorithms. Since we choose the transformation angles of the FrFT domain to be closer to the DFT domain, this similarity is expected.

The confidence check results show that our threshold calculation method is successful after attacks (Figure 5.3). The confidence check for DWT domain technique shows a more distributed threshold value. This is because, the algorithm does not know the exact locations of the watermark, but is try to find in the attacked image. Thus, it may find the true location or may not.

The results show that, the FrFT domain algorithm is robust to high compression rates. In addition, for small compression rates, it has a high margin. This would help us when we are considering multiple attacks such as JPEG compression and noise addition. In these experiments, margin gains more importance, because after the first attack, another attack type is able to use rest of the margin.

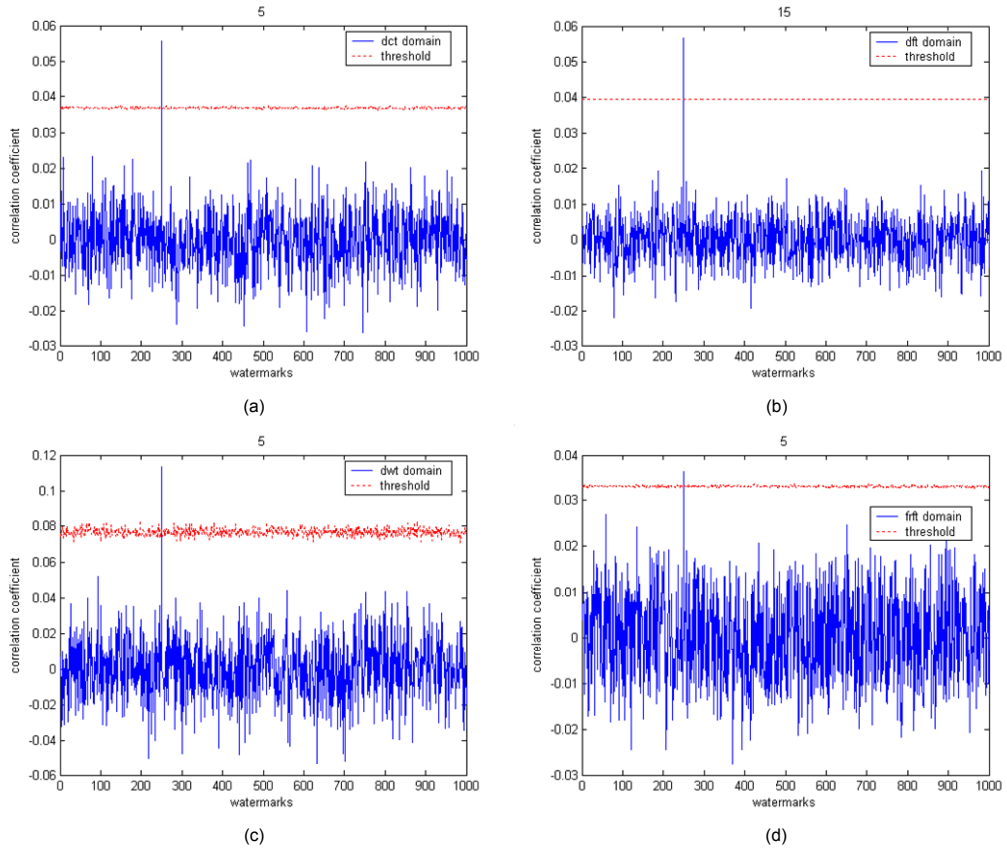


Figure 5.3 Confidence check of algorithms against JPEG compression. (a) DCT algorithm, JPEG Quality = 5, (b) DFT algorithm, JPEG Quality = 15, (c) DWT algorithm, JPEG Quality = 5, (d) FrFT algorithm, JPEG Quality = 5.

The FrFT domain algorithm with varying transformation angles is tested against compression attack. The results are shown in Figure 5.4. The algorithm could not detect the watermark below the transformation angles 0.8. Watermark strength values for these transformation angles are small to ensure the watermark imperceptivity, which support this result. Since small angle values transform the image into a domain closer to the spatial domain, watermark distortion on the image is more. To compensate this, small watermark strength is selected, however, this results in undetected watermark.

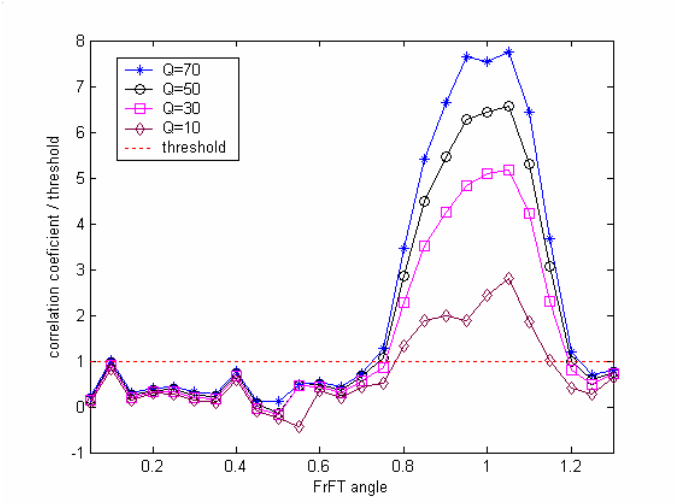


Figure 5.4 Margin values of the FrFT domain algorithm using different transformation angles against JPEG compression with qualities 70, 50, 30 and 10.

5.1.2 Low Pass Filtering

Low pass filtering includes linear and non-linear filters. Frequently used filters include median, Gaussian and standart average filters.

Median filtering is a non-linear operation useful in reducing impulsive, or salt-and-pepper type noise. Impulsive noise can occur due to a random bit error in a communication method [33]. It is also used in remodulation techniques to determine the watermark.

The 5x5 sized median filtered version of watermarked *Lena* image is shown in Figure 5.5, and the margin comparisons of algorithms are shown in Figure 5.6.

Figure 5.5 Median filtering on the FrFT domain watermarked image. Median filter size is 5x5.

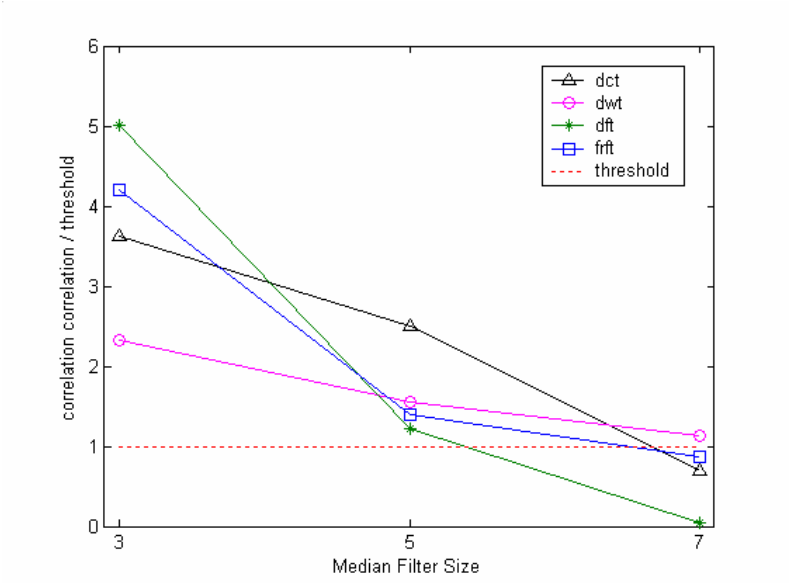


Figure 5.6 Comparison of watermarking algorithms against median filtering

The DCT domain algorithm stands to the 7x7 sized filtering while the other algorithms can stand to 5x5 sized. This result shows the DCT domain algorithm more robust among others. The confidence check plot of the algorithms is shown in Figure 5.7.

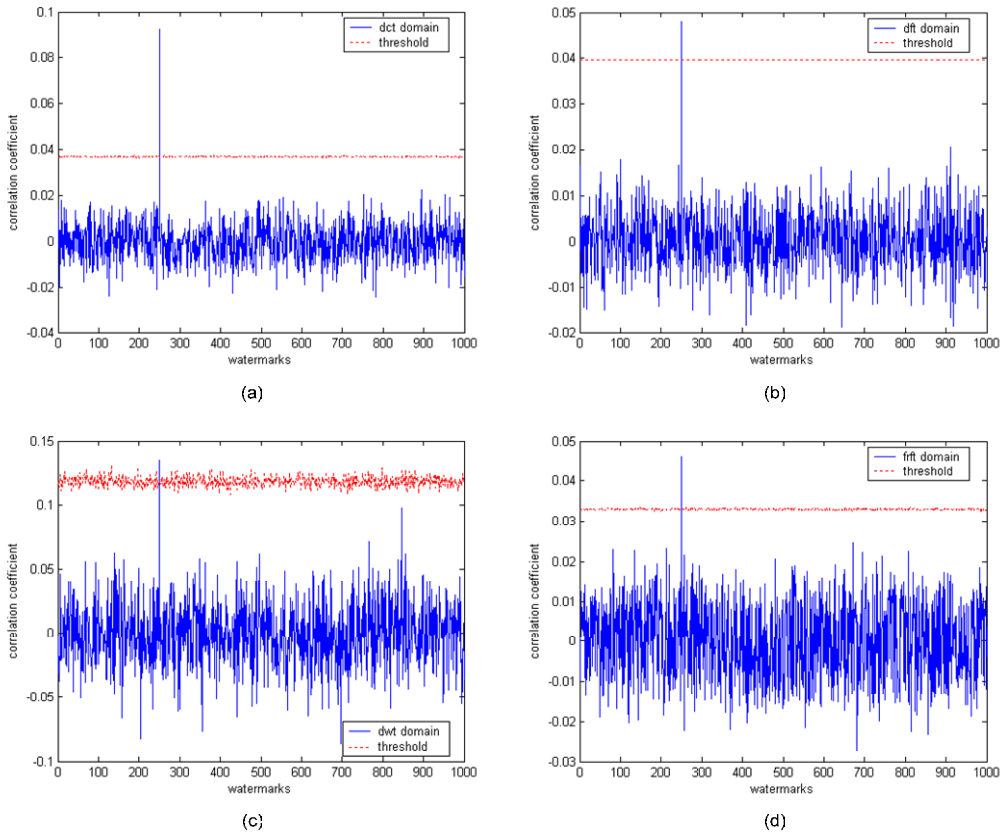


Figure 5.7 Confidence check of algorithms against median filtering. (a) DCT algorithm, Filter Size = 5x5, (b) DFT algorithm, Filter Size = 5x5, (c) DWT algorithm, Filter Size = 7x7, (d) FrFT algorithm, Filter Size = 5x5.

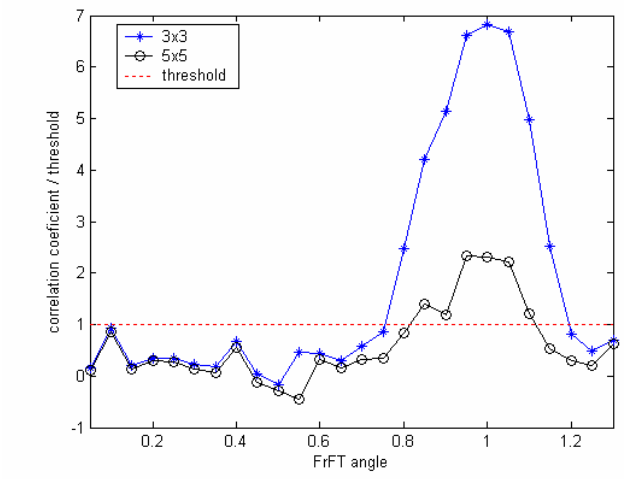


Figure 5.8 Margin values of the FrFT domain algorithm using different transformation angles against median filtering with filter sizes 3x3 and 5x5.

Test results for FrFT angles are shown in Figure 5.8. The results are similar with the experiments for compression attack. The algorithm is successful for the transformation angles above 0.8 and its performance is increasing while the angle increases. The performance is most when the transformation domain is around frequency domain.

Like median filtering, linear low pass filtering (standart average filtering, Gaussian filtering, etc.) operation smooths the image and is thus useful in reducing noise. However, median filtering differ in a way that it can preserve discontinuties in a step function and can smooth a few pixels whose value differ significantly from their surroundings without affecting the other pixels.

The magnitude of the frequency spectrum of a 3x3 averaging filter and the Lena image filtered with this filter are shown in Figure 5.9 and Figure 5.10.

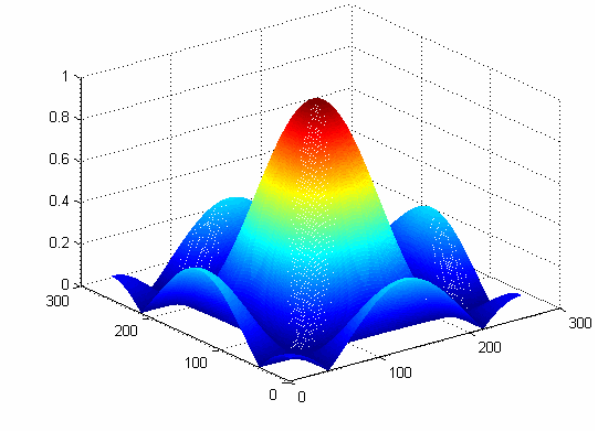


Figure 5.9 Frequency spectrum of 3x3 averaging filter.

Figure 5.10 Average filtering on the FrFT domain watermarked image. Flter size is 3x3.

The margin comparison plot and the confidence check results are given in Figure 5.11 and Figure 5.12. Performance results of the FrFT domain for different transformation angles are shown in Figure 5.13.

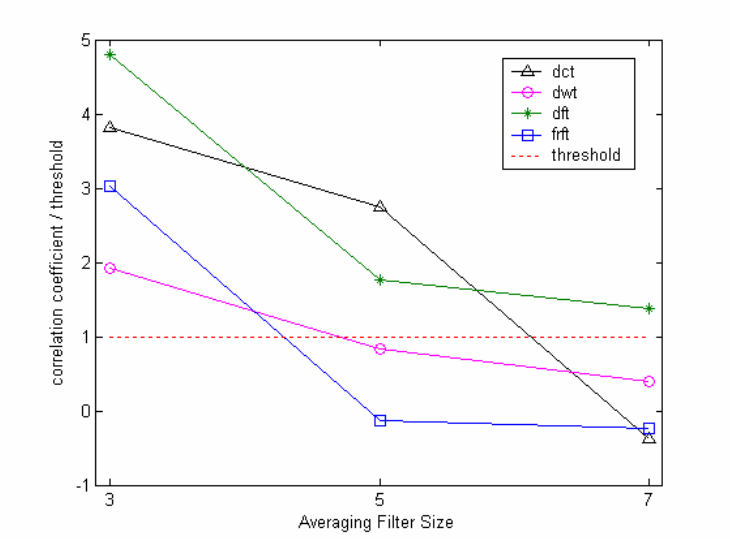


Figure 5.11 Comparison of watermarking algorithms against average filtering.

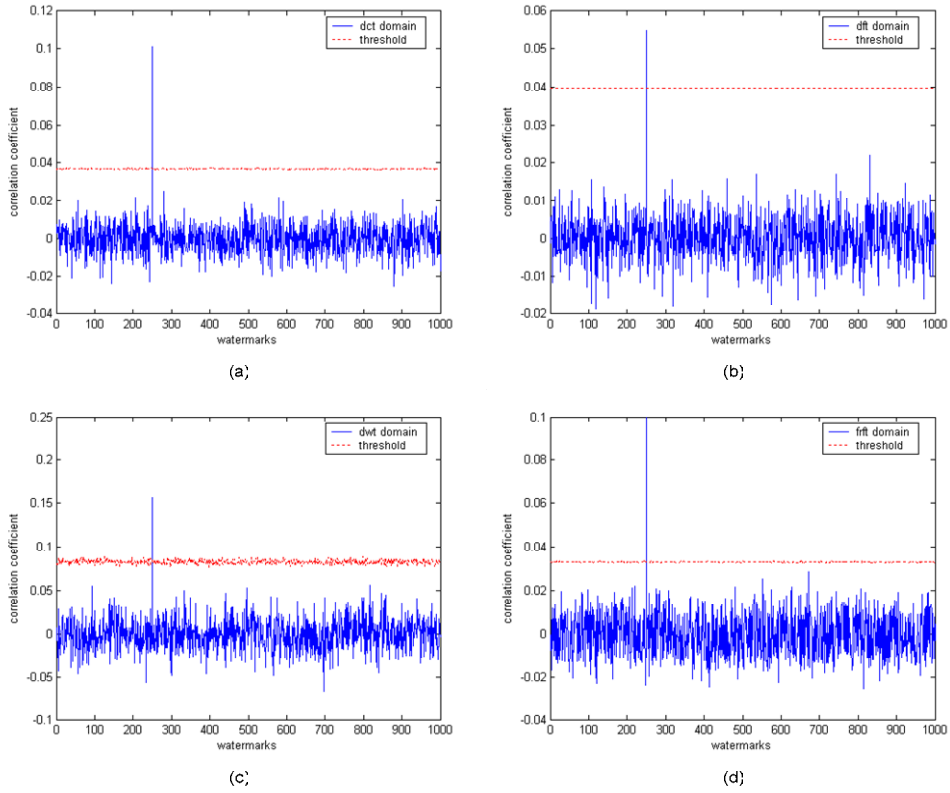


Figure 5.12 Confidence check of algorithms against average filtering. (a) DCT algorithm, Filter Size = 5x5, (b) DFT algorithm, Filter Size = 7x7, (c) DWT algorithm, Filter Size = 3x3, (d) FrFT algorithm, Filter Size = 3x3.

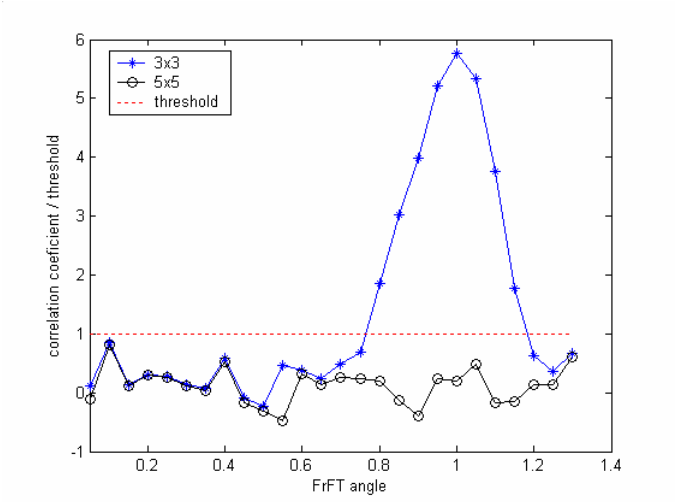


Figure 5.13 Margin values of the FrFT domain algorithm using different transformation angles against averaging filtering with filter sizes 3x3 and 5x5.

The impulse function of a Gaussian filter is given in equation 5.2. This attack type is included in Stirmark, and we implement the same filter. The frequency spectrum of the filter is shown in Figure 5.14. In Figure 5.15, the filtered Lena image is shown, which is watermarked in FrFT domain.

$$h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{5.2}$$

The confidence check results (Figure 5.15) shows that all implemented methods are robust against this attack. The test results for the averaging filtering was similar. All of the algorithms were survived from 3x3 sized filter, however, the difference was in the 5x5 sized filter.

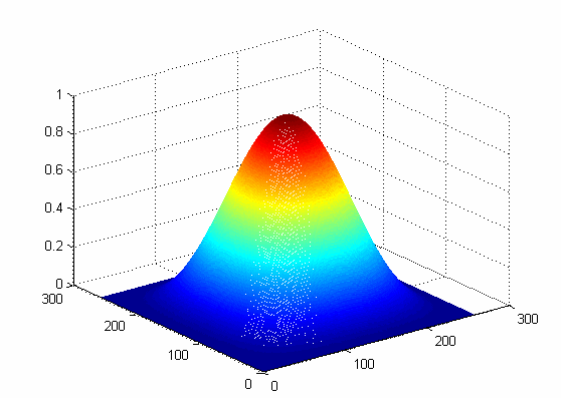


Figure 5.14 Frequency spectrum of a 3x3 Gaussian filter.

Figure 5.25 Gaussian filtering applied on FrFT domain watermarked image.

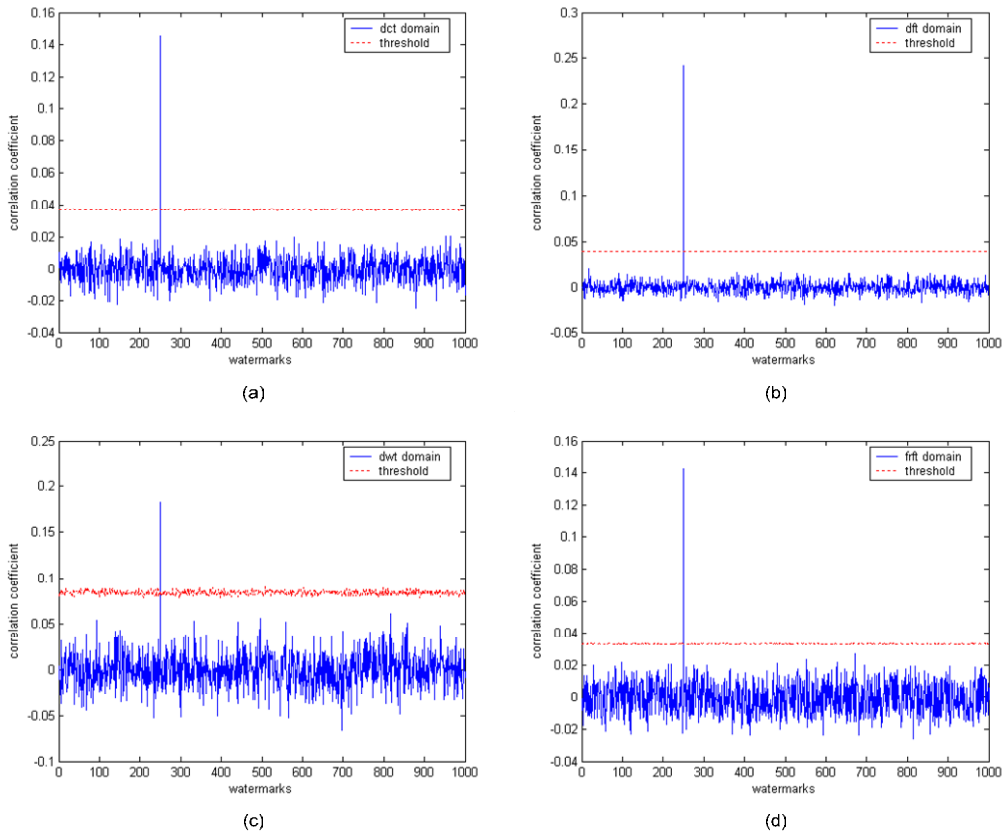


Figure 5.36 Confidence check of algorithms against Gaussian filtering. (a) DCT algorithm, (b) DFT algorithm, (c) DWT algorithm, (d) FrFT algorithm.

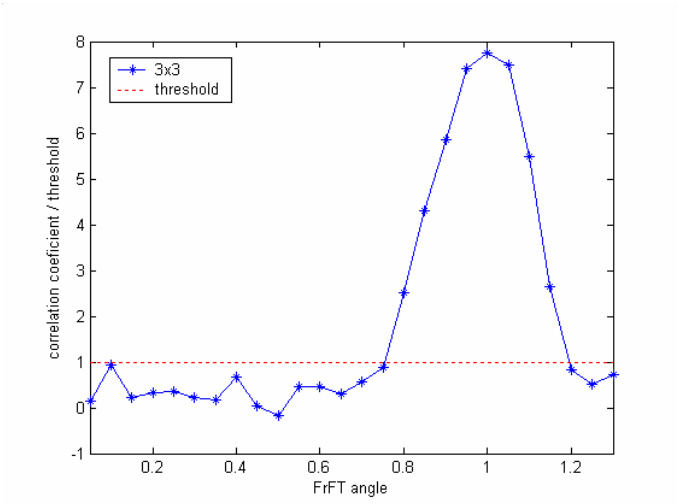


Figure 5.17 Margin values of the FrFT domain algorithm using different transformation angles against Gaussian filtering with filter size 3x3.

Test results shows that filtering is an important concern for watermarking algorithms. When the filter size is increased, the watermark could not be detected. Only DFT domain algorithm survives from 7x7 sized filtering. However, the situation is different for median filter. Since median filtering operation preserves the edges, the algorithms show better performance. Therefore, we can say that, the DFT domain shows more impulsive noise characteristics, where the other algorithms use mostly high frequency regions of an image with respect to DFT algorithm.

The effect of FrFT angles is similar with the previous experiments. The watermark can be detected for higher transformation angles and its performance is increasing while the transformation gets closer to the frequency domain.

5.1.3 Sharpening

Sharpening functions belong to the standart functionalities of photo edition shoftware. These filters can be used as an effective attack on some watermarking schemes because they are very effective at detecting high frequency noise introduced by some digital watermarking software. More subtle attacks are based on the Laplacian operator. In its simplest version the attacked image is $I' = I - \alpha\nabla^2(\nabla^2 I - I)$ where α is the strength of the attack.

In Stirmark, the sharpening filter is implemented by the filtering function given in equation 5.3. The magnitude of the frequency spectrum of this 3x3 sharpening filter and the filtered Lena image is shown in Figure 5.18 and Figure 5.19.

$$h = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{5.3}$$

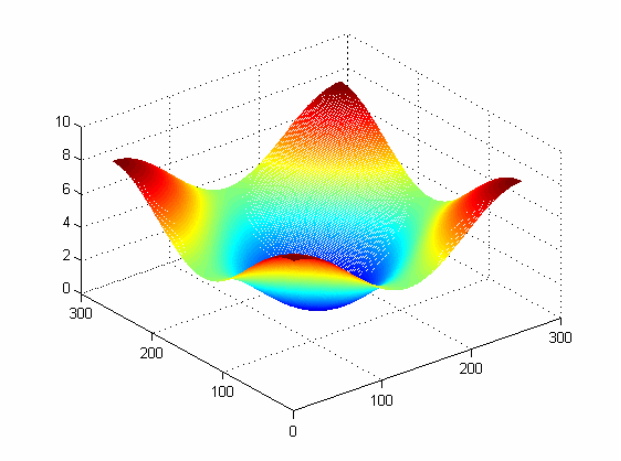


Figure 5.48 Frequency spectrum of a 3x3 sharpening filter.

Figure 5.59 Sharpening filter applied on the FrFT domain watermarked image.

The confidence check results (Figure 5.20) show that the algorithms are successful against sharpening attack. Since, these kind of attacks effect the low frequency regions, it is easier for a watermark to survive compared to the low pass filtering operations.

The effect of using different transformation angles for the FrFT domain is shown in Figure 5.21. As compared to previous examples, sharpening increases the performance of the algorithm and it becomes easier to detect the watermark.

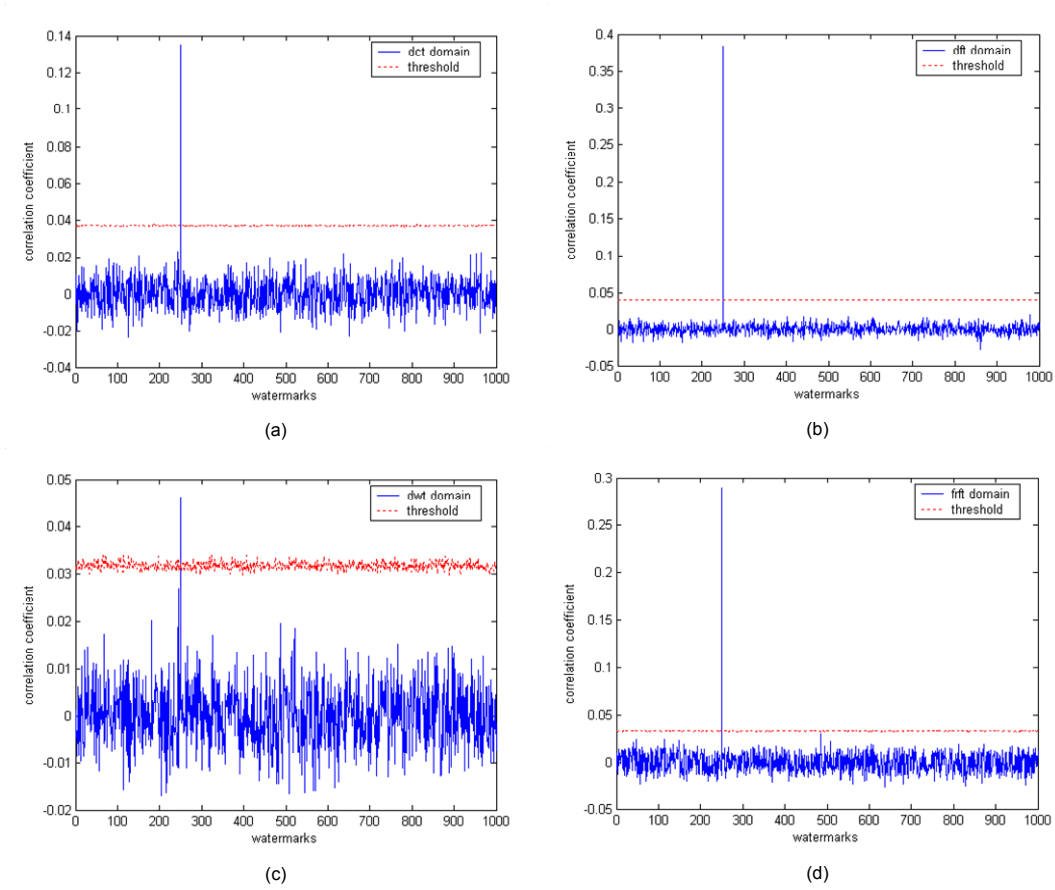


Figure 5.20 Confidence check of algorithms against sharpening filter. (a) DCT algorithm, (b) DFT algorithm, (c) DWT algorithm, (d) FrFT algorithm.

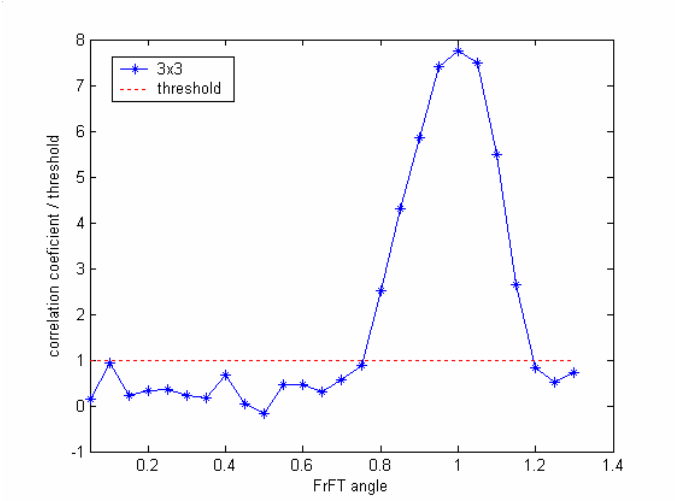


Figure 5.21 Margin values of the FrFT domain algorithm using different transformation angles against sharpening attack.

5.1.4 Gaussian Noise Addition

Additive noise has been largely addressed in the communication theory and signal processing theory literature. Authors often claim that their watermarking techniques survive this kind of attack but many forget to mention the maximum level of acceptable noise. Here, we will show the amount of noise that each algorithm survives.

In Figure 5.22, the noise added Lena image is shown. In the experiments, the noise variance has a range changing from 50 to 10000. In the Figure 5.23, the margin plot shows the amount of noise that the algorithms can withstan. The confidence check plots for the algorithms are given in Figure 5.24.

Figure 5.22 Gaussian noise added to the FrFT domain watermarked image. Noise variance is 10000.

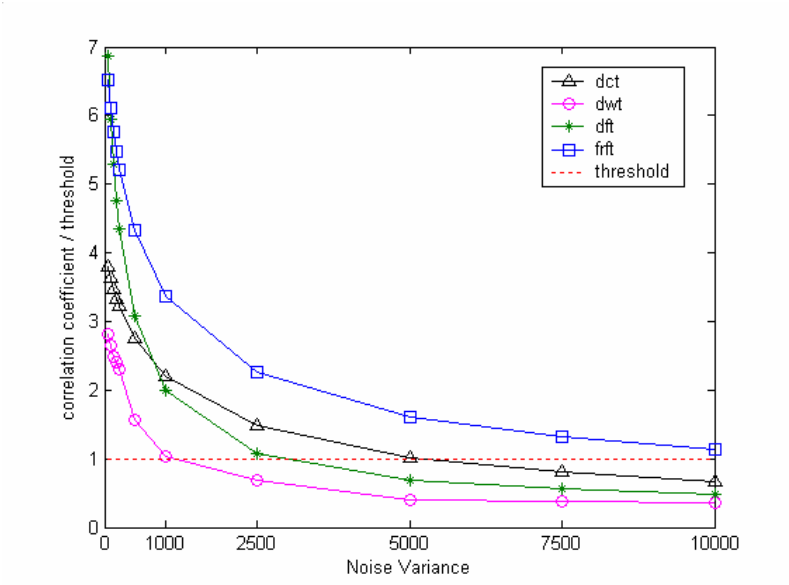


Figure 5.23 Comparison of watermarking algorithms against noise addition attack.

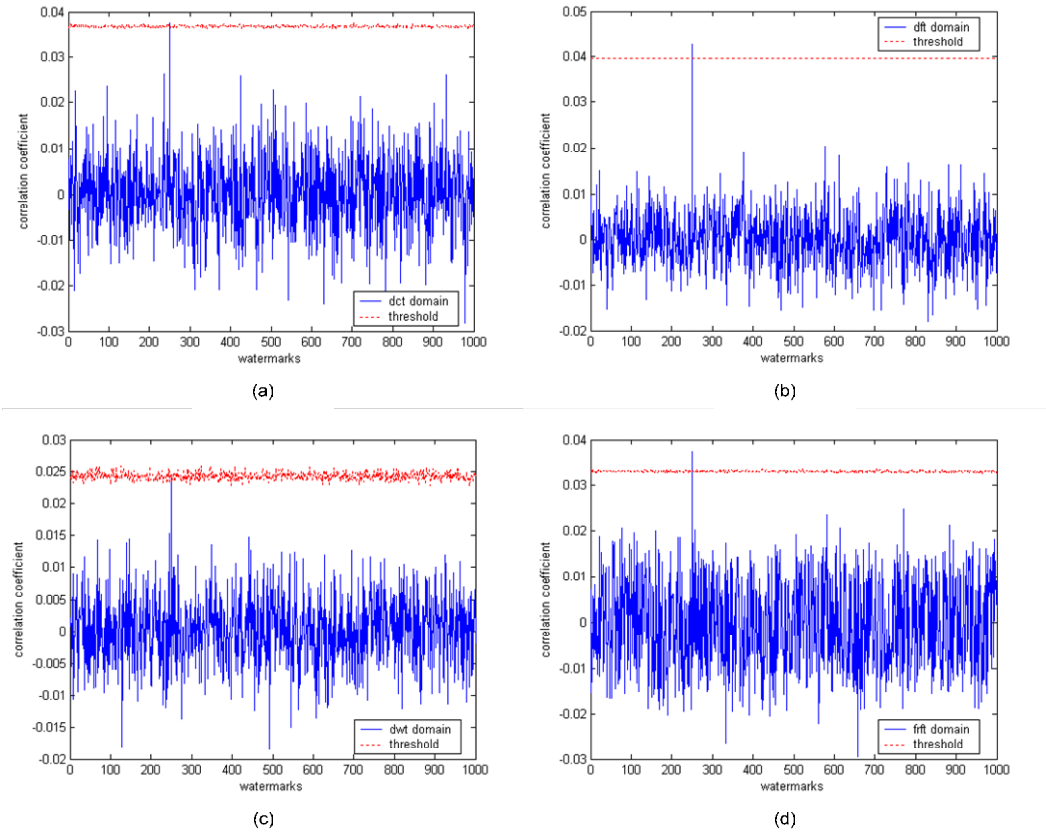


Figure 5.24 Confidence check of algorithms against noise addition. (a) DCT algorithm, noise variance = 5000, (b) DFT algorithm, noise variance = 2500, (c) DWT algorithm, noise variance = 1000, (d) FrFT algorithm, noise variance = 10000.

The results show that, the FrFT domain watermarking shows the best robustness among others against noise addition attack. Although the variance of the noise is increased to 10000, the mark continues to survive. This makes the algorithm suitable, when the image is transferred in a very noisy media.

The effect of using different transformation angles for the FrFT domain is shown in Figure 5.25. Again, the watermark can be detected for the transformation angles that brings the image close to frequency domain.

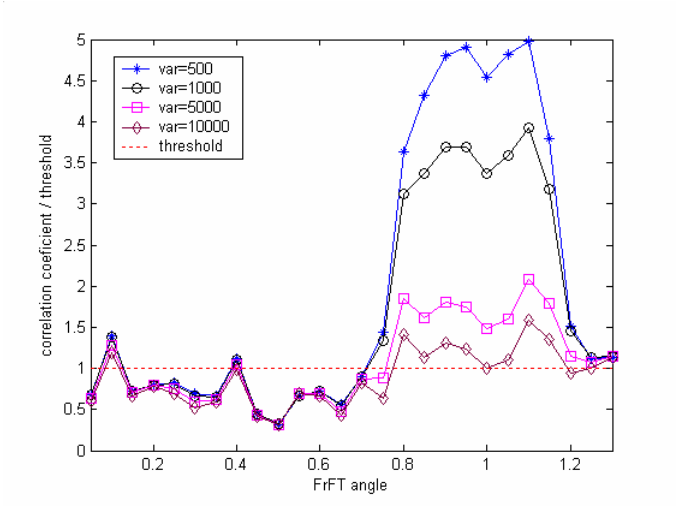


Figure 5.25 Margin values of the FrFT domain algorithm using different transformation angles against noise addition attack. Noise variances are 500, 1000, 5000 and 10000.

5.2 EXPERIMENTS ON GEOMETRICAL ATTACKS

In this section, different combinations of geometrical attacks (rotation, scaling and cropping) are applied. We will add the template inserted version to the experiments to show the abilities of the template algorithm.

5.2.1 Cropping

In some cases, pirates are just interested by the center part of the copyrighted

material; moreover more and more web sites use image segmentation, which is the basis of mosaic attack. Thus, the watermark must be detected from a small portion of the image.

While implementing the experiments, we crop the images with varying ratios of the original image. The subtracted part is padded with zeros. By doing this, we preserve the watermark synchronization with the watermark locations in the remaining portion. However, after the image is cropped, the remaining portion can be removed. This time, the image becomes a cropped and linearly translated version of the watermarked. This is out of our concern in this section, and we will only interest in cropping attack. Translation attack will be held individually later.

The experiments involve cropping range changing from 90% to 5% of the original image size. The cropped Lena image (crop ratio is 10%), the comparison of algorithms and the confidence check plots are given in Figures 5.26-28.

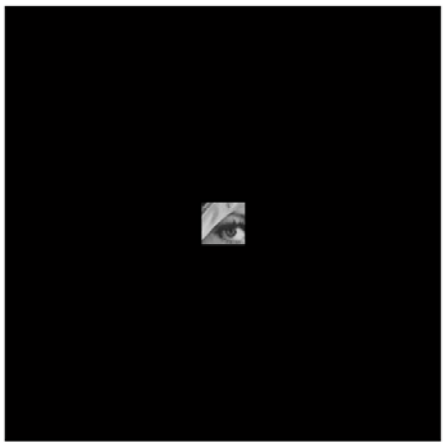


Figure 5.66 FrFT domain watermarked image is cropped. Cropping ratio is 10%.

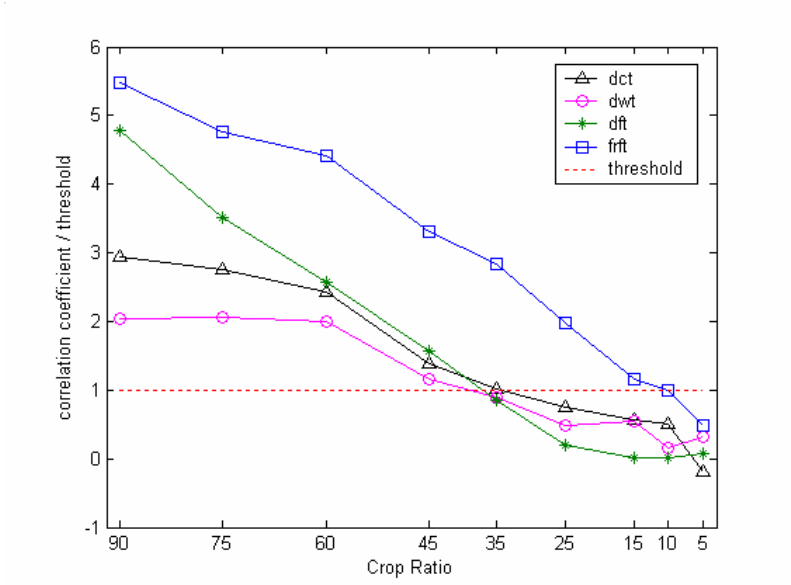


Figure 5.77 Comparison of watermarking algorithms against cropping attack.

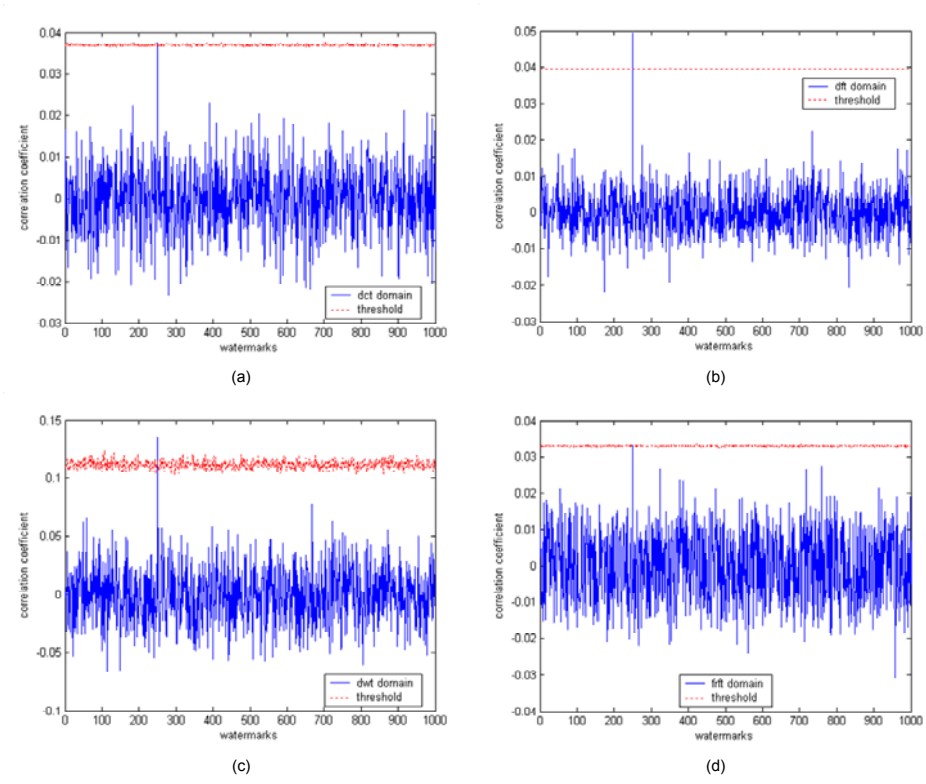


Figure 5.28 Confidence check of algorithms against cropping. (a) DCT algorithm, crop ratio = 35%, (b) DFT algorithm, crop ratio = 40%, (c) DWT algorithm, crop ratio = 40%, (d) FrFT algorithm, crop ratio = 10%.

From the results, it is shown that FrFT domain technique is robust to cropping attack until the image is cropped to obtain 10% of its original size. However, the other algorithms stand about 40% cropping ratio. Since the watermark is embedded in a combination of frequency and spatial domains, the mark is distributed all around the image. Thus, the detector can find the watermark in small portions. However, for the other algorithms, the watermark is inserted at selected frequencies, which may result with the loss of watermark, if these frequency regions are cropped.

The effect of using different transformation angles for the FrFT domain is shown in Figure 5.29. Compared to previous examples, the experiments for low transformation angles, which transform image into regions closer to the spatial domain, has increased performance. This results improve our decision about the performance of the FrFT algorithm against cropping attack. To become more successful against cropping attack, the watermark must embedded not only into the identified frequencies but it must also occupy all regions in the image, whetheher any region have components in the selected frequency region or not. This increases the chnace of the watermark to survive when any random region is selected to be cropped.

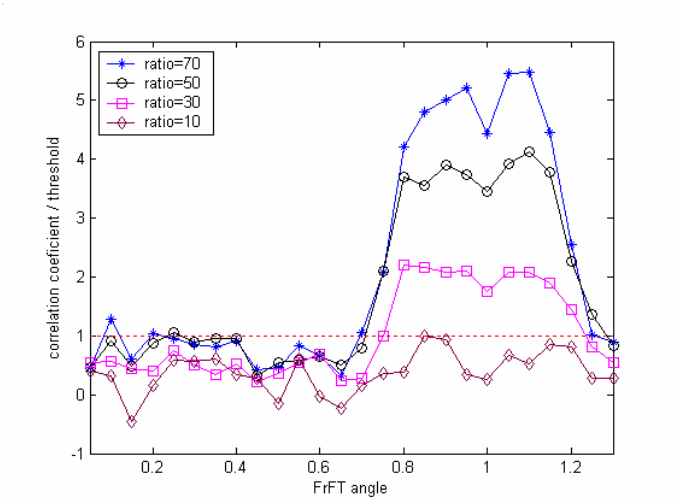


Figure 5.29 Margin values of the FrFT domain algorithm using different transformation angles against cropping attack. Crop ratios are 70%, 50%, 30% and 10%.

5.2.2 Translation

Translation can occur in filtering operations where the original size is not preserved. If a *mxm* sized image is filtered with a *nxn* sized filter, the resulting filtered image size becomes $m+n-1xm+n-1$. In addition, as we described in the above section, if only the cropped portion of the image is kept after cropping attack, this becomes a combination of cropping and translation attack.

The translated of the watermarked Lena image is shown in Figure 5.30. We preserve all of the image by padding the outer region into the translated region to prevent data loss. This will allow us to see the effect of translation individually. The comparison of algorithms and confidence check results are shown in Figure 5.31 and Figure 5.32.

Figure 5.30 DFT domain watermarked image is translated by 80 pixels in horizontal axis and 175 pixels in vertical axis.

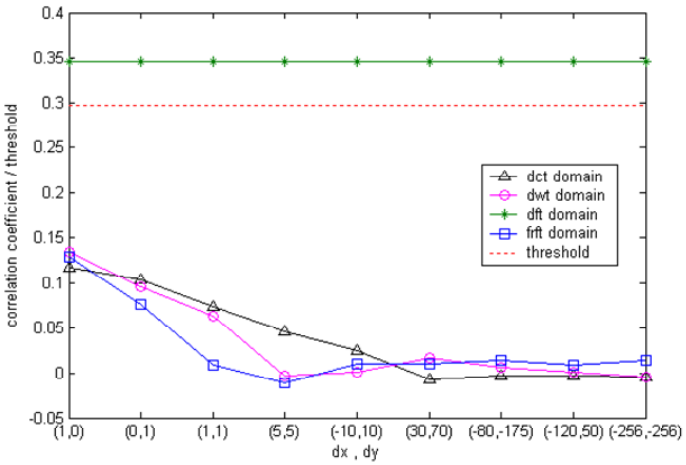


Figure 5.31 Comparison of watermarking algorithms against translation attack. Horizontal labels show the amount of translation in both axes.

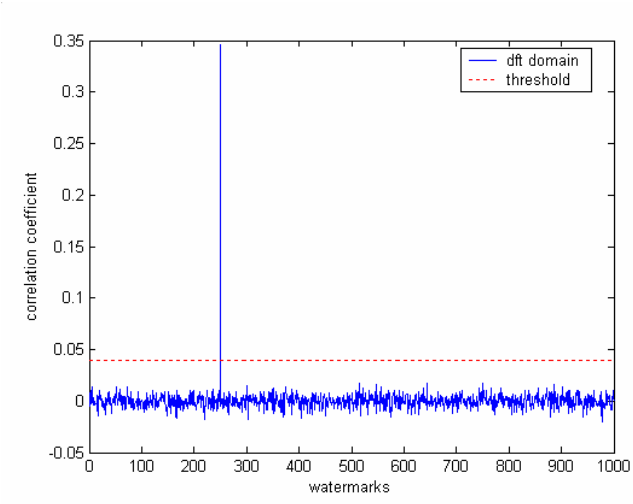


Figure 5.32 Confidence check of DFT domain watermarking algorithm against translation attack. The watermarked image is translated 80 pixels in horizontal axis and 175 pixels in vertical axis.

Only the DFT domain watermarking algorithm shows robustness to the attack. This is because of the property of the DFT domain that the magnitude coefficients of the DFT domain does not affected from linear shifts in spatial domain. Since no data is lost, the correlation value remains constant for all attacks.

The other algorithms lost the watermark if any translation (as small as one pixel shift) is applied. This shows the effect of synchronization loss. The template

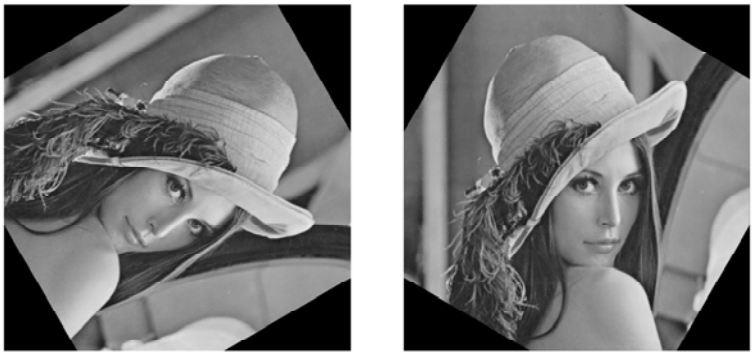
does not work in translation attack, because it is embedded into the magnitude coefficients and the locations of it remain unchanged. Therefore, it is not probable to detect the amount of translation with the help of the template.

However, it is possible to define a much more complex geometrical transformation by using projective polynomial transformations (where the image is transformed by a defined polynomial). In this case, the DFT domain also becomes vulnerable. Geometrical attacks are still an area of research and the researchers try to find algorithms that are robust to all kinds of geometrical attacks.

5.2.3 Rotation & Cropping

Small angle rotation, often in combination with cropping, does not usually change the commercial value of the image but can make the watermark un-detectable. Rotations are used to realign horizontal features of an image and it is certainly the first modification applied to an image after it has been scanned.

In this experiment, we rotated the image not only in small angles, but also in large angles to test the algorithm efficiency. 60 degree rotated Lena image is shown in Figure 5.33(a). After detecting the template, the rotation is reversed to recover the watermarked image (Figure 5.33(b)). The outer side which exceeds the image is cropped. The test results of the algorithms are shown in Figures 5.34 and 5.35.



(a) (b)

Figure 5.33 (a) Template inserted FrFT domain watermarked image is rotated by 60 degree. (b) The recovered image by using template detection mechanism.

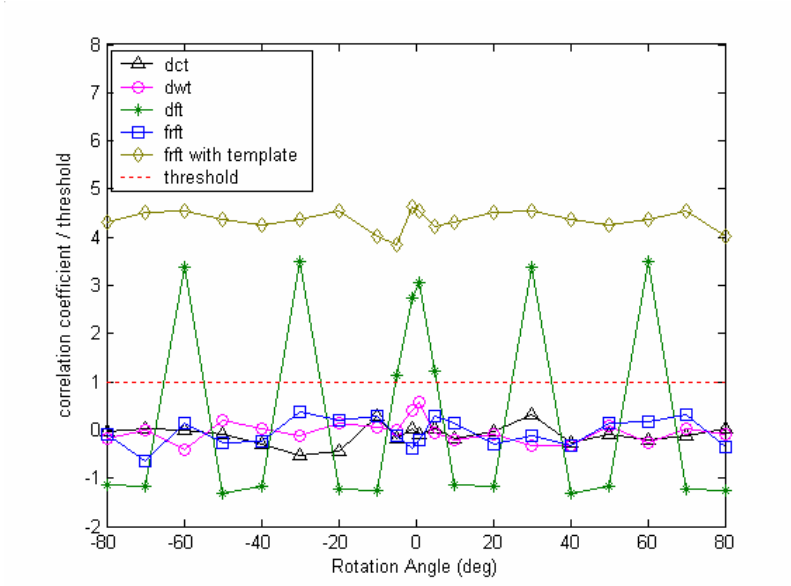


Figure 5.34 Comparison of watermarking algorithms against rotation and crop attack.

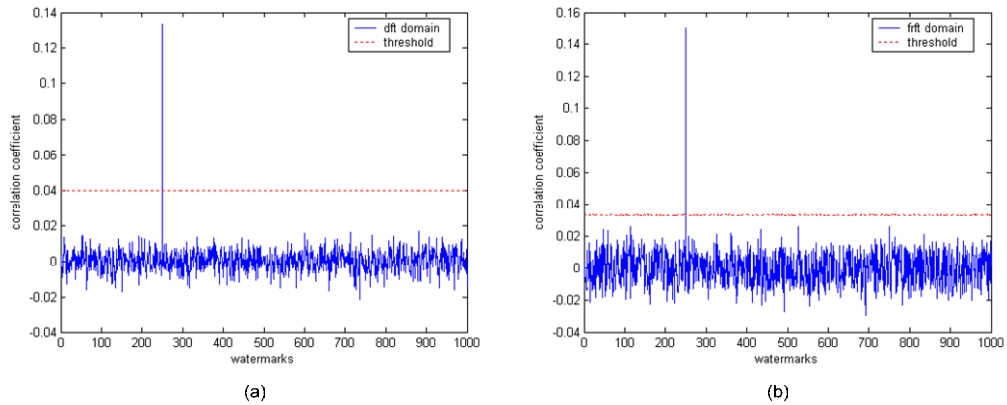


Figure 5.35 Confidence check of algorithms against rotation&crop attack. Rotation angle is 60 degrees. (a) DFT domain watermarking algorithm, (b) FrFT domain watermarking algorithm with template insertion scheme.

The results show that the synchronization is lost even if it would be a small angle. However, it is seen that, the FrFT domain algorithm gains robustness against the rotation attack by the help of the template. This does not surprise us, because the aim of the template is to eliminate the rotation and scaling attacks.

The DFT domain detects the watermark periodically. As we explained, the watermark is formed of sectors of ones and zeros, and they are separated by a

constant angle, which is 15 degree for our case. When the ones and zeros match with the same valued regions, the watermark can be found. In our case the exact matching will appear periodically in every 30 degrees. However, by implementing a simple search algorithm, the algorithm may become robust to all rotation angles.

5.2.4 Rotation & Scaling

Another kind of rotation experimented is the scaling applied after rotation is implemented. In this attack, no data loss occurs because of cropping, however the amount of interpolations increase.

The 60 degree rotated and scaled Lena image is shown in Figure 5.36 (a) and the recovered image is shown in Figure 5.36 (b). The comparison of algorithms are given in Figure 5.37. Only template inserted FrFT domain algorithm resist to this attack. Thus, only its confidence check plot is shown (Figure 5.38).

(a) (b)

Figure 5.36 (a) Template inserted FrFT domain watermarked image is rotated by 60 degree and it is scaled to fit its original size. (b) The recovered image by using template detection mechanism.

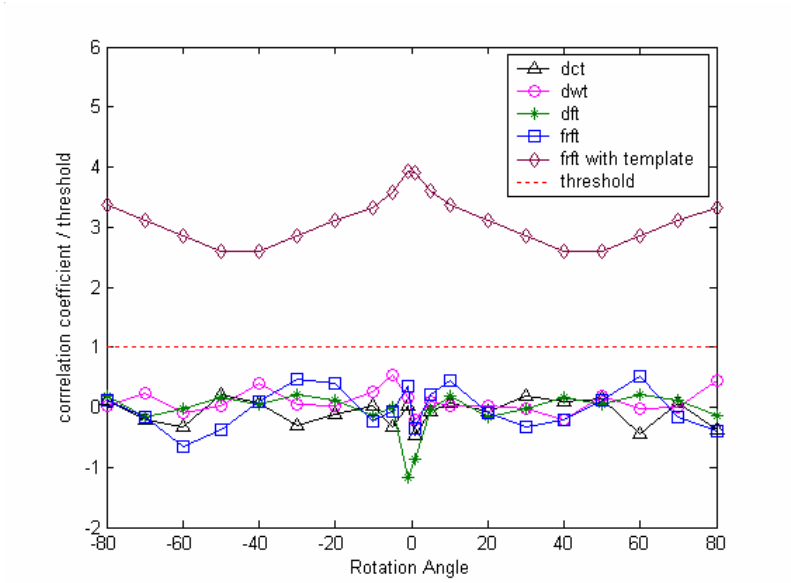


Figure 5.87 Comparison of watermarking algorithms against translation attack.

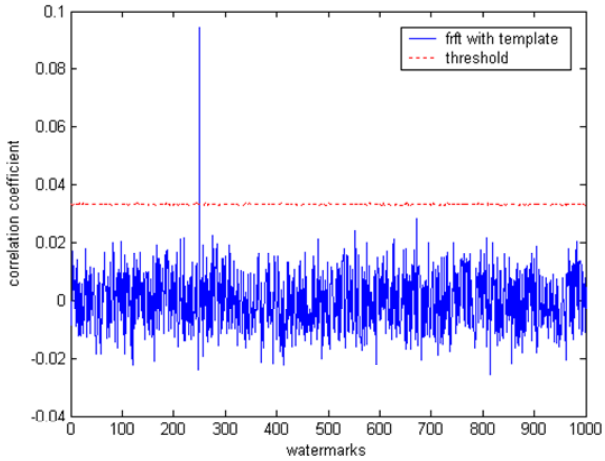


Figure 5.98 Confidence check of template inserted FrFT domain algorithm against rotation&scale attack. Rotation angle is 60 degrees.

The results show that, the template is successful to resolve both rotation and scaling. The margin value is minimum for the rotation angles around 45 degrees because the scale ratio is bigger at these values, which means more interpolation and more distortion on the watermark. On the other hand, this does not prevent detecting the template and finding the watermark.

5.2.5 Scaling

Scaling happens when a printed image is scanned or when a high resolution digital image is used for electronic applications such as Web publishing. This should not be neglected as we move more and more toward Web publishing. Scaling can be divided into two groups, uniform and non-uniform scaling. Under uniform scaling we understand scaling which is the same in horizontal and vertical direction. Non-uniform scaling uses different scaling factors in horizontal and vertical direction (change of aspect ratio). Very often digital watermarking methods are resilient only to uniform scaling. Uniform scaling is implemented in our experiments.

When the image is down-scaled; i.e. its size is shrinked, the outer region is padded with zeros to remain the original image size. However, if it is up-scaled, i.e. its size is increased, the regions that exceeds the original size is cropped. Two scaled versions are given in Figure 5.39 and Figure 5.40. First it is down-scaled to 50% of its original size. In the second it is up-scaled to 150% of its original size. The recovered images are also presented in the figures. The test results are shown in Figures 5.41-42.

(a) (b)

Figure 5.109 (a) Watermarked Lena image is down-scaled to 50% of its original size and the outer section is padded with zeros. (b) The image is recovered by the template algorithm.

(a) (b)

Figure 5.40 (a) Watermarked Lena image is up-scaled to 150% of its original size and the outer region is cropped. (b) The image is recovered by the template algorithm.

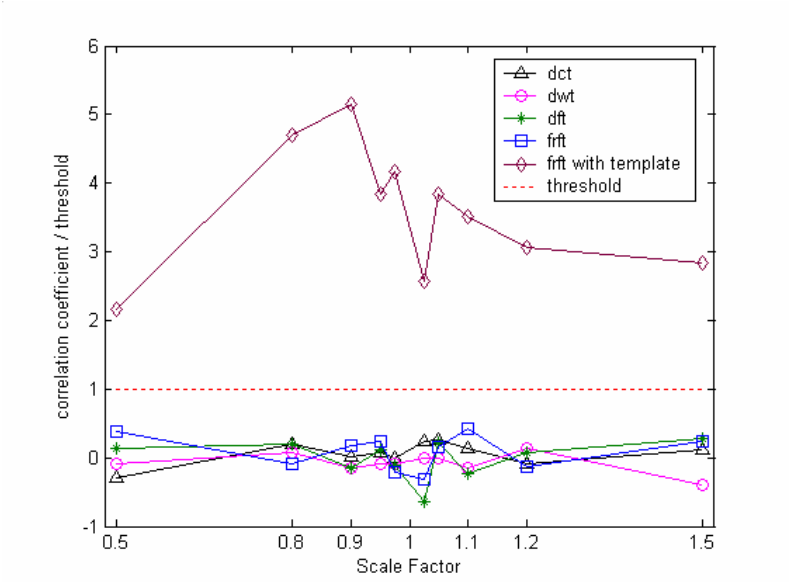


Figure 5.41 Comparison of watermarking algorithms against scaling attack.

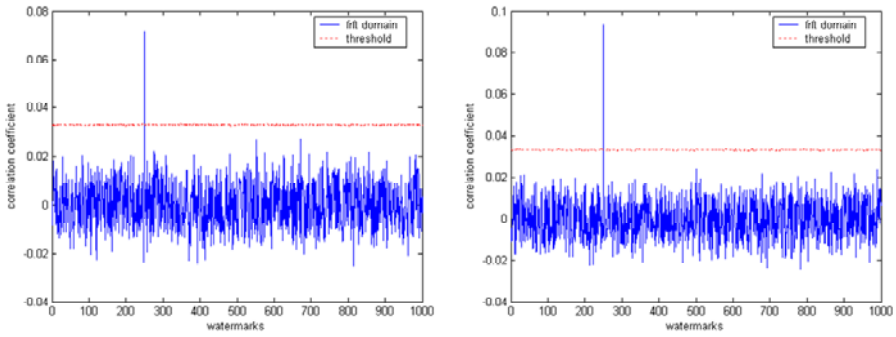


Figure 5.42 Confidence check of the template inserted FrFT domain algorithm against linear scaling attack. (a) Scale factor is 0.5. (b) Scale factor is 1.5.

The FrFT domain algorithm resist to this kind attack by the help of the template. The DFT domain watermark is not survived from this attack. It would survive if the size of the image will not be changed. By padding zeros or by cropping we changed the size; therefore, the synchronization is lost. However, the situation is opposite for the template algorithm. If the size will not change, it would be unable to detect the template. This situation can be resolved by applying small changes to the algorithms, which compares the original size and the resulting size of the image. Nevertheless, this would require the knowledge of the original size of the image.

5.3 EXPERIMENTS ON MULTIPLE ATTACKS

Sometimes, more than one type of attack is applied onto the image, which is called multiple attacks. In such cases, the watermark must stand against those attacks. In this section, some examples of multiple attacks will be presented.

5.3.1 Noise Addition with Low-Pass Filtering

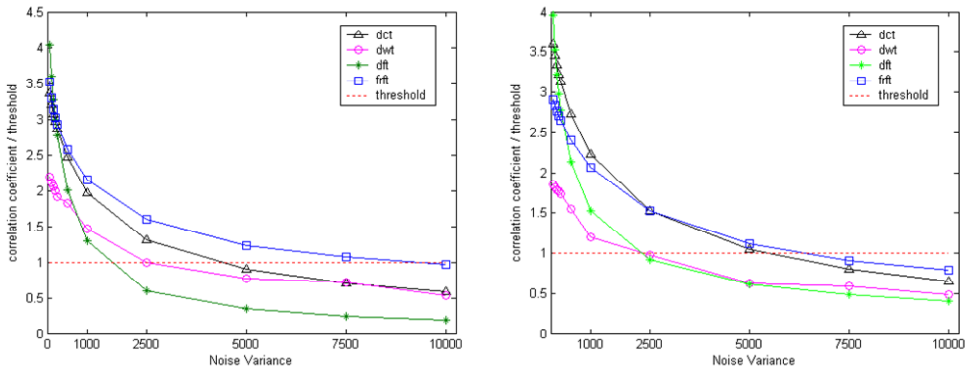
If an image becomes too noisy because of the noisy transmission media, the infringer may want to remove the noise before distributing the image. In the

experiments, the watermarked images are first corrupted by noise addition (with various variances) and then they will be filtered with a 3x3 sized filter (median and averaging filter). The noisy images filtered with median filter and averaging filter are shown in Figure 5.43.

(a) (b)

Figure 5.43 The FrFT domain watermarked Lena image is first corrupted by noise (with variance 5000) and then filtered by 3x3 sized (a) median filter, (b) averaging filter

The comparison plots of the algorithms are shown in Figure 5.44.



(a) (b)

Figure 5.44 Comparison of watermarking algorithms against noise addition and filtering attacks. (a) Filter is 3x3 median filter. (b) Filter is 3x3 averaging filter.

The results show the importance of the margin. The margins of the FrFT and DCT domain algorithms are more when compared with DFT and DWT domain algorithms. Although, there is some more attenuation because of the second attack, this high margin prevents the watermarks being lost. The advantage of the FrFT domain algorithm against noisy mediums is proved once more that, it could detect the watermark after one more attack is applied.

The confidence check results of the algorithms are shown in Figure 5.45.

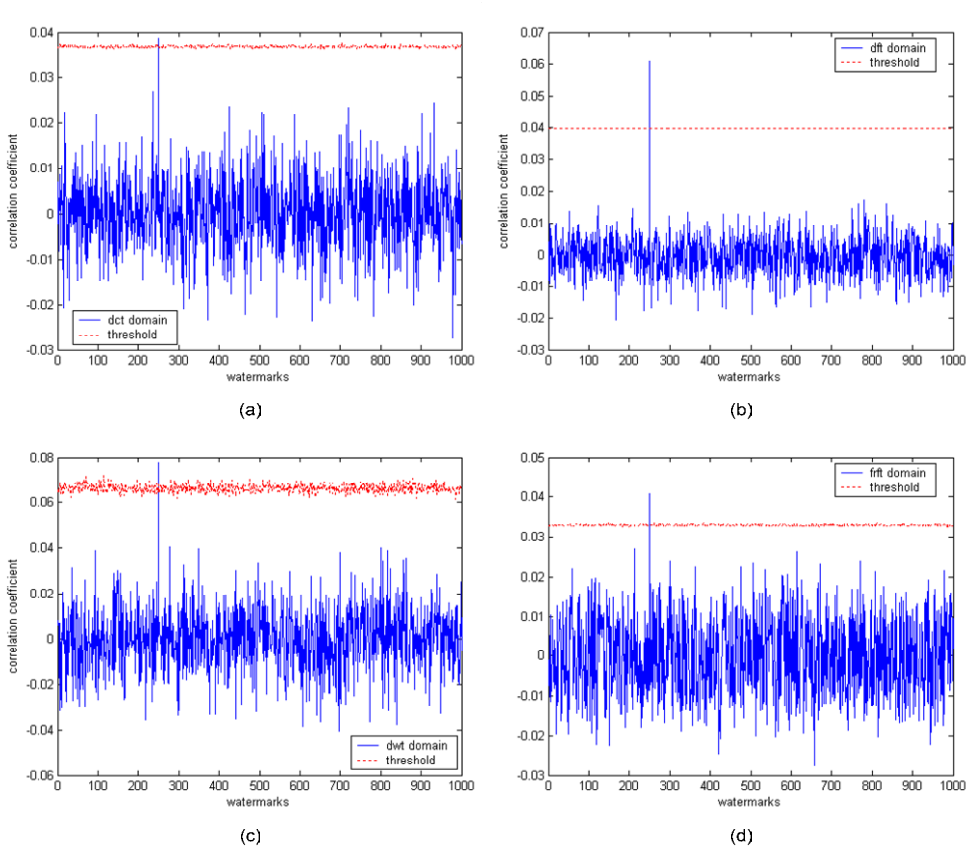


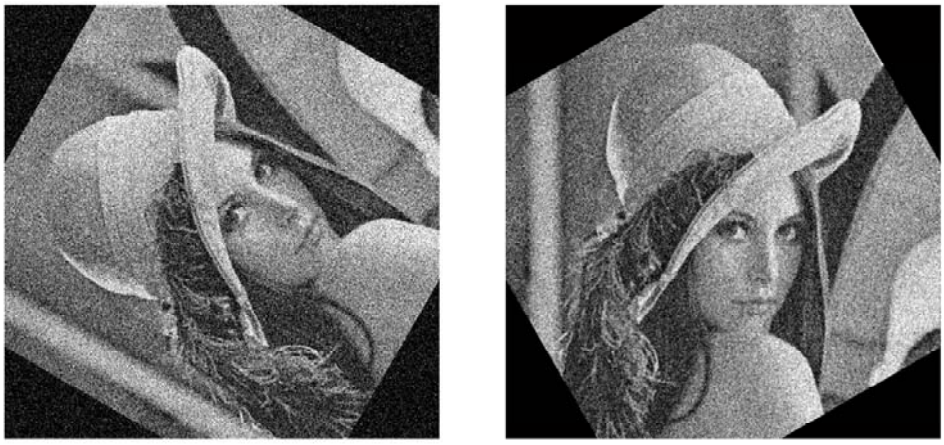
Figure 5.45 Confidence check of algorithms against noise addition and filtering attacks (3x3 averaging filter). (a) DCT algorithm, noise variance = 5000, (b) DFT algorithm, noise variance = 1000, (c) DWT algorithm, noise variance = 1000, (d) FrFT algorithm, noise variance = 5000.

5.3.2 Geometric Distortions with Noise Addition

Printing-scanning process introduce geometrical as well as noise-like distortions. In this section, first a rotation attack with cropping will be applied onto the image and then a Gaussian noise with varying variances will be applied additionally.

Since, DCT and DWT domain algorithms could not withstand to geometrical distortions, only the results of DFT and FrFT (template inserted) domain algorithms will be presented here. For the experiment, the rotation is chosen as 60 degree, thus, a search algorithm is not needed for the DFT algorithm. The noise variances are 50, 100, 250, 500, 1000, 1500 , 2000 and 2500 respectively.

In Figure 5.46, attacked Lena image and its recovered version is shown. Test results are presented in Figures 5.47 and 5.48.



(a) (b)

Figure 5.46 The watermarked and template inserted Lena image. (a) rotated by 60 degree and Gaussian noise with variance 2000 is added.

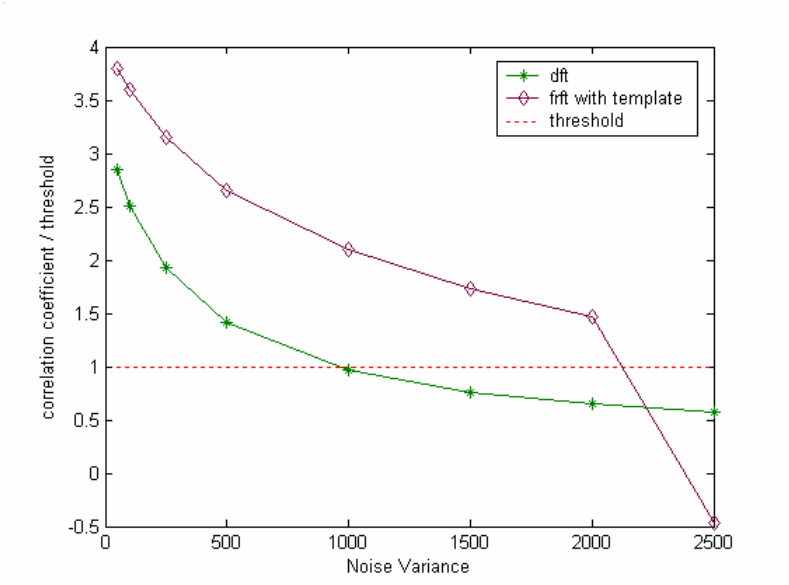


Figure 5.117 Comparison of DFT and FrFT domain algorithms against rotation&crop and noise attacks. The watermarked images are rotated by 60 degree and then noises with different variances are added.

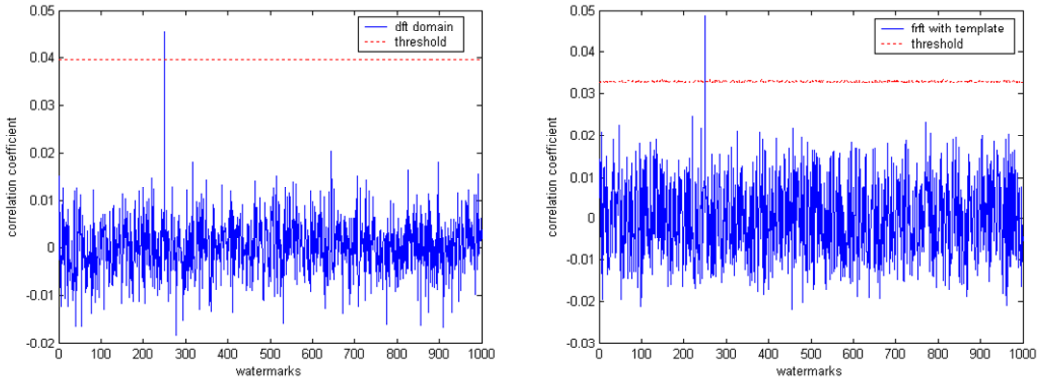


Figure 5.128 Confidence check of algorithms against rotation&cropping and noise addition attacks. (a) DFT domain, noise variance =750, (b) FrFT domain, noise variance = 2000.

As we explained, margin gets importance when we test the algorithm against multiple attacks. The results show that, the high margin obtained by the FrFT domain against noise addition attack give it additional performance among DFT algorithm. The template detector becomes unable to find the template locations when the noise variance becomes too high. With respect to this, a sharp decrease occured in the detection correlation of the FrFT algorithm, when the noise variance

increases to 2500. Since the template could not be detected, the synchronization could not be detected. This is an adventage of the DFT domain algorithm that, it does not need any other synchronization tools but the properties of the Fourier domain ensures the geometrical invariance. However, it need additional search algorithms for finding the locations.

5.3.3 Geometric Distortions with JPEG Compression

Rotation, and scaling alone are not enough they should be also tested in combination with JPEG compression. Since most pirates will first apply the geometric transformation and then save the image in a compressed format it makes sense to test robustness of watermarking system to geometric transformation followed by compression. However an exhaustive test should also include the contrary since it might be tried by willful infringers. It is difficult to chose a minimal quality factor for JPEG as artifact quickly appear. However experience from professionals shows that quality factors down to 70% are reasonable.

In the experiments the DFT and template inserted FrFT domain algorithms are tested. First, watermarked images are rotated by different angles, then the rotated images are JPEG compressed with a quality factor 70.

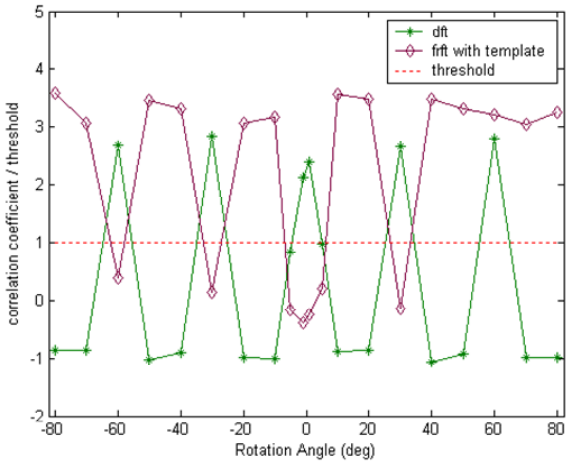


Figure 5.139 Comparison of DFT and FrFT domain algorithms against rotation&crop and JPEG compression. The watermarked images are rotated by various angles and then they are JPEG compressed with quality factor 70.

Figure 5.49 shows the comparison of two domains. As we explained, the DFT algorithm needs a search algorithm to become robust against all rotation angles. The results show that FrFT domain algorithm, sometimes, lost the watermark. This is because of the effect of compression on the template points. The compression (or other filtering operations) do not only affect the watermark, but also affect the template. Thus, when the template lost, the algorithm becomes vulnerable to geometric transformations. The confidence checks of the algorithms against the attacks are shown in Figure 5.50.

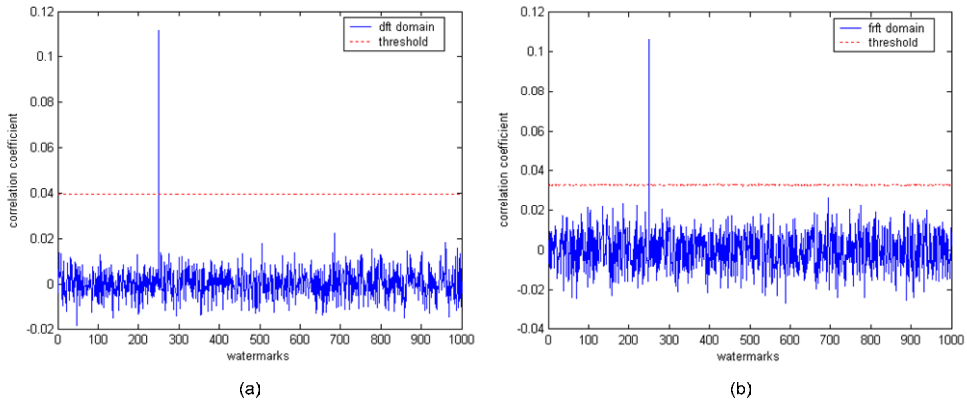


Figure 5.50 Confidence check of the algorithms against 60 degree rotation (and cropping) followed by a JPEG compression with quality factor 70. (a) DFT domain algorithm. (b) Template inserted FrFT domain algorithm.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

We have implemented and made experiments on four watermarking algorithms. They differ from each other by the transform domain used when inserting the watermark. These four algorithms are based on spread spectrum approach, thus the watermark is embedded at spread of frequencies with low energies, thus making them invisible by a pirate listening the channel.

For the two of the algorithms, which are DCT and FrFT domain techniques, we implement a visual masking method by considering the properties of the human Visual System. We separate the image into pieces and embed the watermark more into the regions, which have higher variance between pixel values. This increases the fidelity of the image. An image quality index is utilized to measure the image fidelity in terms of perceptual figures like means and variances.

To allow a fair comparison, we adjusted the watermark strength of the algorithms. After watermark embedding, these four algorithms give the nearly same PSNR value for an image. In addition to the PSNR values, we calculate the quality indexes, MSE and SNR values for all of the algorithms. These values are also close to each other, thus we ensured the experiment reliability.

In our experiments, we implement several attacks to the watermarked images. Attacked images fed to the watermark detector and correlation value between the watermark signal and watermarked coefficients are computed. Although many watermarking schemes use linear correlation or normalized correlation, we prefer to use correlation coefficient because it makes the algorithms robust to mean and variance changes in the watermark signal. The correlation coefficient is

compared with a threshold value again computed by using the attacked image. We did not define a constant detection threshold, rather describe a method to compute it, and implement this technique on all watermarking techniques.

We compared the algorithms by defining a term *margin*, which shows the amount of additional distortion to make the watermark undetectable. The margin helps us to easily compare the different algorithms, since it brings the threshold value to one and compute a detection correlation regarding to this threshold.

In addition, we presented the confidence check of algorithms against attacks, in which the algorithm is tested with many different pseudorandom generated watermark signals. This is useful to prove the algorithm, since it shows only the true watermark gives higher correlation values then the threshold value.

We have also tested the FrFT domain algorithm with different transformation angles to determine the effect of the angles. The test results show that, for lower angles, the transform domain is closer to the spatial domain and the watermark creates more attenuation on the image. To compensate this, we reduced the watermark strength, however, this time the watermark could not be detected. When the angles increase and the transform domain becomes closer to the frequency domain, the performance of the algorithm is also increases. The best performance is achieved when the transform domain is around the frequency domain.

For simple removal attacks, such as compression, filtering, noise addition etc., the algorithms are successful in great extend. However, it is not the case for the geometrical attacks. Geometrical attacks destroy the synchronization between the watermarks and the marked coefficients, results in the loss of watermark. The watermarks except the one embedded in DFT domain could not withstand to the geometrical attacks. Because of the properties of the Fourier domain, the DFT domain algorithm survived from these attacks. We also presented some multiple attack results, which are combination different types of attacks. The margin of the algorithms gains importance while dealing with multiple attacks. If an algorithm has more margin after one attack, it would be easier to withstand to the other attacks.

To gain robustness against geometrical attacks, we implemented a template

mechanism, which is used to determine the amount of geometrical translations undergone by the image. After the template is detected, the transformations are reserved and the synchronization is regained. The experimental results show that, the template is successful in determining the amount of rotation and scaling attacks.

The FrFT domain watermarking shows good robustness characteristics especially against compression, noise addition and cropping attacks. The use of the template makes it additionally robust to rotation and scaling attacks. Its high margin allows it to be robust against multiple attacks.

As a result, the FrFT domain technique is robust to many removal attacks, and gains additional robustness against some geometrical attacks by implementing a template addition method. However, it is hard to say that, it is a better algorithm then the other implemented techniques. This is because; the algorithms are tested with the same set of parameters. The parameters were selected as to provide more robustness to a type of attack; however, this would probably decrease its robustness against other sets of distortions. On the other hand, the FrFT domain algorithm has some advantages among other algorithms. First, it is not widely known domain. Thus, it is harder to determine the presence of the watermark for a pirate and implement a specific attack for it. Additionally, two more degrees of freedom are used to identify the algorithm, which are the transformation angles of the fractional Fourier transform. It is not enough to know the watermark locations and coefficients to detect the watermark, but also these transformation angles must be known. Another advantage of these angles is that, more watermarked versions of an image may be created by using these signals. As a result, the FrFT domain watermarking algorithm can be considered when there is a need for robust and secure watermarking scheme.

As a future work, we want implement the FrFT domain algorithm in a way to remove the need for watermark locations. That will remove most of the watermark payload for detection purpose. In addition, we will optimize and improve the template algorithm to become faster and more intelligent.

REFERENCES

[1] N. Nikolaidis and I. Pitas, “Robust image watermarking in the spatial domain” in *Signal Processing, Special Issue on Copyright Protection and Control*, vol 66, No.3, 1998, pp. 385-403.

[2] F. Sebe, J Domingo-Ferrer and Jordi Herrera, “Spatial-Domain Image Watermarking Robust against Compression, Filtering, Cropping, and Scaling” in *Information Security, ISW'00*, 2000, pp. 44-53. ISBN 3-540-41416-9.

[3] Peter Meerwald, ”Digital Image Watermarking in the Wavelet Transform Domain”, MSc thesis in University of Salzburg, 2001.

[4] S. Eren Balcı, “Robust watermarking of images”, MSc thesis in Middle East Technical University, 2003.

[5] I. Djurovic, S. Stankovic and I. Pitas, “Digital watermarking in the fractional Fourier transformation domain”, in *Journal of Network and Computer Applications*, 2001, pp. 167-173.

[6] I. Cox, J. Killian, T. Leighton and T. Shamoon, “Secure Spread Spectrum Watermarking for Multimedia”, in *IEEE Trans. On Image Processing*, 6, 12, 1997, pp. 1673-1687.

[7] I. Cox, J. Killian, T. Leighton and T. Shamoon, “Secure Spread Spectrum Watermarking for Images, Audio and Video”, in *International Conference on Image Processing, ICIP’96*, volume 3, pp. 243-246.

[8] Saraju P. Mohanty, “Digital Watermarking : A tutorial”. 2004, *http:// www.csee.usf.edu/~smohanty/research/Reports/WMSurvey1999Mohanty.pdf*

[9] Saraju P. Mohanty, “Watermarking of digital images”, MSc thesis, Indian Institue of Science, January 1999.

[10] December 2004, *http//www.watermarkingworld.org/*, “Digital Watermarking Frequenly Asked Questions”

[11] Ingemar J. Cox, Matt L. Miller, Jeffrey A. Bloom, “Watermarking applications and their properties”, in *International Conference on Information Technology*, *ITCC'2000*. 2000, pp. 6-10.

[12] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung, “Can Invisible Watermarks Solve Rightful Ownerships?” *IBM Technical Report RC 20509,* IBM Research, July 1996. IBM Cyberjournal: *http://www.research.ibm.*

[13] Petere Meerwald and Shelby Pereira, “Attacks, applications and evaluation of

known watermarking algorithms with Checkmark", in *Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents IV*, vol. 4675, no.4675, pp. 293-304, 2002.

[14] S. Volvoshynovsky, S. Pereira, T. Pun, J.J. Eggers and J.K. Su, "Attacks on digital watermarks: Classification, Estimation based Attacks and Benchmarks", in *IEEE Communications Magazine*, vol.39 no.8, pp.118-126, 2001.

[15] Vinicius Licks and Ramiro Jordan, "Geometric Attacks on Image Watermarking Systems: A Survey", Submitted To Journal Publication, 2003

[16] A. Piva, M. Barni, F. Bartolini, V. Cappellini, "Threshold Selection for Correlation-Based Watermark Detection", in *Proceedings of COST 254 Workshop on Intelligent Communications, 1998*, pp. 67-72.

[17] M. Barni, F. Bartolini, V. Cappelini and A. Piva, "Robust Watermarking of Still Images For Copyright Protection", in *Proceedings 13th International Conference On Digital Signal Processing DSP97*, 1997, pp. 499-502.

[18] Joseph J.K.O Ruanaidh and Thierry Pun, "Rotation, Scale and Translation Invariant Digital Image Processing", in *Proceedings IEEE International Conference on Image Processing 1997, ICIP 1997*, vol. 1, pp. 536-539, 1997.

[19] V. Solachidis and I. Pitas, "Circularly Symmetric Watermark Embedding in 2-D DFT Domain", *in IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'99*, vol.6, 1999,pp. 3469-3472.

[20] V. Licks and R. Jordan, "On digital image watermarking robust to geometric transformations", in *Procedings of 2000 International Conference Image Processing, ICIP'2000*, vol. 3, 2000, pp. 690-693.

[21] Shelby Pereira, Joseph J.K.O Ruanaidh and Frederic Deguillaume, "Template Based Recovery of Fourier-Based Watermarks Using Log-polar and Log-log Maps", in *IEEE International Conference on Multimedia Computing and Systems, ICMCS'99*, 1999, pp. 870-874.

[22] Shelby Pereira and Thierry Pun, "Fast Robust Template Matching for Affine Resistant Image Watermarking", in *International Workshop on Information Hiding, Lecture notes in Science,* 1999, pp. 200-210.

[23] R. Dugad, K. Ratakonda, and N. Ahuja, "A new wavelet-based scheme for watermarking images", in *Proceedings of International Conference on Image Processing, ICIP'98*, vol.2, 1998, pp. 419-423.

[24] Tatiana Alieva, Martin J. Bastiaans and Maria Luisa Calvo, "Fractional Cyclic Transforms in Optics: Theory and Applications", in *Recent Research Developments in Optics 1*, 2001, pp. 105-122.

[25] Luis B. Almeida, "The Fractional Fourier Transform and Time-Frequency Representations", in IEEE Transactions on Signal Processing, Vol. 42, No. 11,1994, pp. 3084-3091.

[26] Haldun M. Özaktaş, Orhan Arıkan, M. Alper Kutay and Gözde Bozdağı, "Digital Computation of the Fractional Fourier Transform", IEEE Transactions On Signal Processing, Vol. 44, No. 9, 1996, pp. 2141-2150.

[27] Çağatay Candan, M. Alper Kutay, Haldun M. Özaktaş, "The Discrete Fractional Fourier Transform" in IEEE Transactions On Signal Processing, vol. 48, no. 5, pp. 1329-1337, 2000.

[28] V. Ashok Narayanan and K.M.M. Prabhu, "The fractional Fourier transform: theory, implementation, and error analysis", in The Proceedings of Microprocessors and Microsystems, 27 , 2003, pp. 511-522.

[29] A. Bultheel and H. Martinez, "A shattered survey of the Fractional Fourier Transform", Report TW 337, Department of Computer Science,K.U.Leuven, 2002

[30] Zhou Wang and Alan C. Bovik, "A Universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, March 2002.

[31] Zhou Wang, Alan C. Bovik, and Ligang Lu, "Why is image quality assessment so difficult?".

[32] Martin Kutter and Fabien A. P. Petitcolas, "A fair benchmark for image watermarking systems," To in E. Delp et al. (Eds), in vol. 3657, *Proceedings of Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, 25-27 January 1999. The International Society for Optical Engineering.

[33] Jae S. Lim, "Two-Dimensional Signal and Image Processing", *Prentice Hall Signal Processing Series*, pp. 469-476, ISBN 0-13-935322-4