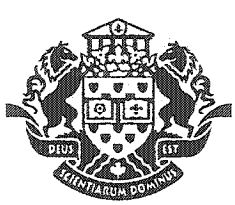



Université d'Ottawa · University of Ottawa

# Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

ZHENG, Dong

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

RST Invariant Digital Image Watermarking

J. Zhao

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

R. Goubran

R. Laganière

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

# RST invariant digital image watermarking

by

**Dong Zheng**

A thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of M.A.Sc

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa

Ottawa, Ontario, Canada

July 2003

*Your file Votre référence*
*ISBN: 0-612-90367-2*
*Our file Notre référence*
*ISBN: 0-612-90367-2*

# Abstract

Digital image watermarking has been proposed as a method to ensure the copyright protection and credibility of images by means of embedding a known piece of digital data into host images. To be useful, a good watermarking scheme should be robust against all known attacks to a certain degree based on requirements.

In this thesis, we propose a novel digital image watermarking scheme that is invariant to rotation, scaling, and translation (RST). We embed watermark in the log-polar mapping (LPM) of the Fourier magnitude spectrum of original image, and use the phase correlation between the LPM of the original image and the LPM of the watermarked image to calculate the displacement of the watermark positions in LPM domain. The exhaustive search method is used to retrieve the watermark if the original image is unavailable.

Spread spectrum technique and perceptual model are used to enhance the security and to achieve the optimum balance between invisibility and robustness. Stochastic analysis is used to determine the optimum threshold to minimize the total false probability of detection. In this thesis, we discuss all these in detail and show how they work with the main watermarking scheme to give better results.

We implement this watermarking algorithm to analyze its performance. The evaluations demonstrate that the scheme is invariant to rotation and translation, invariant to scaling when the scale is in a reasonable range, and very robust to JPEG compression and other attacks.

# Contents

# List of Tables

# List of Figures

# List of Acronyms

| | |
|---|---|
| LPM | Log Polar Mapping |
| ILPM | Inverse Log Polar Mapping |
| RST | Rotation, Scaling and Translation |
| DFT | Discrete Fourier Transform |
| IDFT | Inverse Discrete Fourier Transform |
| HVS | Human Visual System |
| DCT | Discrete Cosine Transform |
| FMT | Fourier Mellion Transform |
| DWT | Discrete Wavelet Transform |
| LSB | Least Significant Bit |
| PN | Pseudo Noise |
| MAP | Maximum Aposteriority Probability |
| ML | Maximum Likelihood |
| PSNR | Peak Signal to Noise Ratio |
| NVF | Noise Visibility Function |

This thesis is dedicated to my parents.

## Acknowledgement

I would like to deeply thank my supervisor, Professor Jiying Zhao, for bringing the problem of digital watermarking to me, and for his valuable guidance and feedback during every step of my work. I greatly appreciate his patience and confidence in my research abilities.

# Chapter 1

# Introduction

## 1.1 Digital watermarking

The rapid development of new information technologies has improved the ease of access to digital information. It also leads to the problem of illegal copying and redistribution of digital media. The concept of digital watermarking came up while trying to solve the problems related to the management of intellectual property in media. A conventional cryptographic system permits only valid key holders access to encrypted data. But once such data is decrypted, there is no way to track its reproduction. A digital watermark is intended to complement cryptographic processes. It is a visible or invisible identification code that is permanently embedded in the data and remains present within the data after any decryption process.

The concept of digital watermarking is derived from steganography. Both steganography and watermarking describe techniques that are used to convey information by embedding it into the cover data. However, steganography typically relates to cover point-to-point communication between two parties. Thus steganography methods are

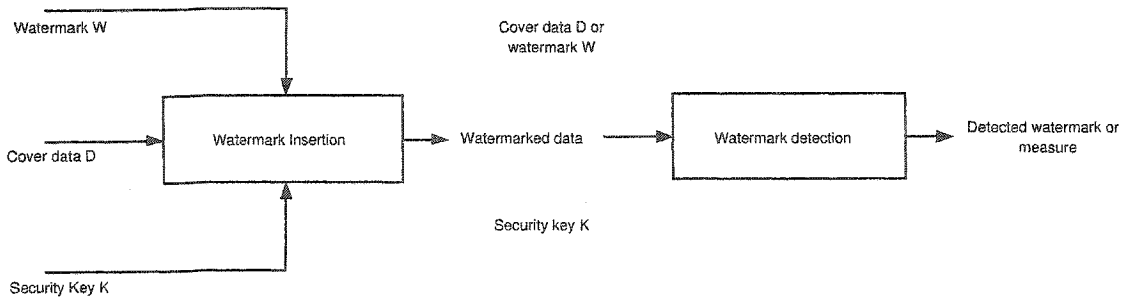



**Figure 1.1:** A typical watermarking system.

usually not robust against modification of the data, or have only limited robustness. Digital watermarking on the other hand should be robust against attempts to remove the hidden data. A popular application of watermarking is to give proof of ownership. It is obvious that for this application the watermark should be robust against manipulation that may attempt to remove it.

Fig. 1.1 is a typical watermarking system, which includes watermark embedding and watermark extraction. The inputs are the watermark, the cover media data and the embedding security key. The watermark can be a number sequence or a binary bit sequence. The key is used to enhance the security of the whole system. The output of the watermark embedding system is the watermarked data.

For the watermark extraction scheme, the inputs are the watermarked data, the security key and, depending on the method, the original data and/or the original watermark. The output is either the recovered watermark or some kind of confidence measurement indicating how likely the original watermark is present in the watermarked data.

Supposed that a watermark is defined as $W$, $D$ is the host data and $K$ is the security key. In watermarking, an embedder function $e(.)$ takes the watermark $W$, the host data $D$ and the security $K$ as input parameters and outputs the watermarked data $D'$.

$$D' = e(D, W, K) \tag{1.1}$$

The watermark is considered to be robust if it is embedded in a way such that the watermark can survive even the watermarked data $D'$ go through serious distortions. The watermark extracting procedure is depicted as following:

$$W' = d(D', K, ...) \tag{1.2}$$

$d(.)$ is the detector function. The $D$ and $W$ are the optional inputs for the detector function.

For a typical watermarking system, several requirements should be satisfied:

1. The watermark $W'$ can be extract from $D'$ with/without requiring explicit knowledge of $D$.

2. $D'$ should be as close to $D$ as possible in most cases.

3. If $D'$ is unmodified, then the extracted watermark $W'$ exactly matches $W$.

4. For robust watermarking, if $D'$ is modified, the $W'$ should still match $W$ well to give a clear judgment of the existence of the watermark.

5. For fragile watermarking, the $W'$ can indicate the possible tamping of the $D'$ and give information about the degradation of the $D'$.

There are different types of digital watermarking:

1. Digital image watermarking. Most of the research about digital watermarking are on image watermarking. The focus of all research on digital image watermarking might be due to that there are so many images available on World Wide Web free of charge and without any copyright protection.

2. Digital video watermarking. A video sequence consists of still images. Therefore all the watermarking methods applied on image can be applied on video. But video watermarking has other problems. For example, a video might be in a compressed format. To retrieve the watermark, we may need to decompress the video. In a real time environment, such a procedure might not to be feasible. So watermarking in the compressed domain might be needed. A video watermark should be able to resist different types of attacks such as frame averaging, frame dropping and frame swapping.

3. Digital audio watermarking. In case of audio signals, the term "watermarking" can be defined as "robust and inaudible transmission of additional data along with audio signals". Audio watermarking is based on the psychoacoustical approach of perceptual audio coding techniques. It exploits the properties of the human ear by embedding one or more key dependent watermark signals below the masking threshold.

4. 3D virtual objects watermarking. The most important component for watermark embedding in both VRML and MPEG4 is the 3D polygonal mesh. Shape of a 3D polygonal mesh is defined by two components, vertex coordinate and vertex topology. Vertex coordinate combined with vertex topology defines more complex geometrical primitives such as lines, polygons. These components are the most important targets for embedding in 3D mesh polygonal meshes.

In this thesis, most of the discussions are focused on the digital image watermarking.

## 1.2 Digital image watermarking

Digital image watermarking uses various image processing techniques to ensure the copyright protection and credibility of images. As a great deal of image data is stored in digital format, it has become easier to modify or forge information. Research in this field has focused on the design of robust techniques for the copyright protection of image data. In such methods, a watermark is imperceptibly or perceptibly embedded in a host image such that its removal using common distortions on the marked image is difficult with dramatically degrading the perceptible data content itself. This is always called "robust watermarking". Watermarking can also be used to address the problem of tamper proofing. For example, the image data are to be used as evidence, in this situation, the image data must be credible. By "credible", we mean that the image source is authentic and the information content has not been modified in transit to its destination. This can be called "fragile watermarking". For the fragile watermarking, it can serve the purpose of indicating that tamping of the original image data had occurred, or it can give more information about the attacks and degradation of the host image. ITU-T recommendation J.147 proposed an objective picture quality measurement method by use of in service test signals. It is based on semi-fragile watermarking. Fig. 1.2 is the classification of digital image watermarking.

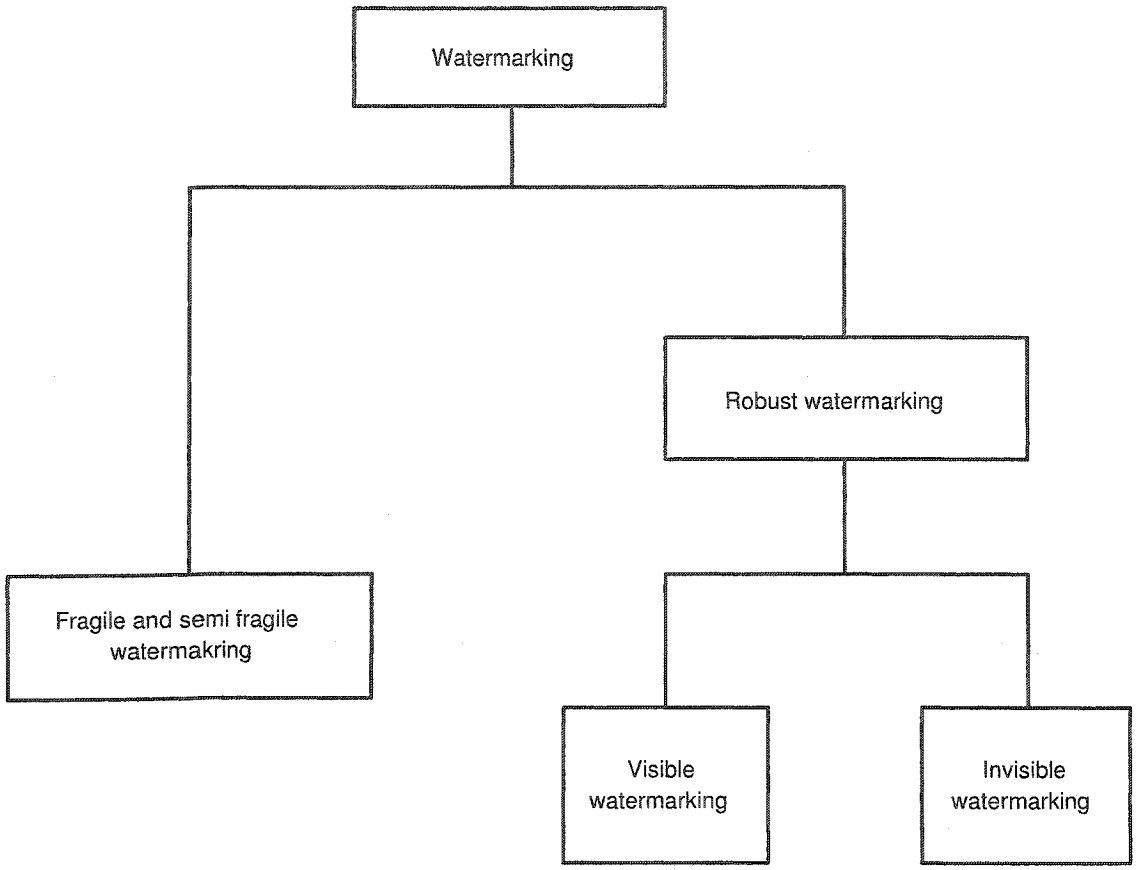



**Figure 1.2:** A classification of digital image watermarking.

## 1.3 A typical digital image watermarking system

Fig. 1.3 presents a simple block diagram of a typical digital image watermarking system. Orthogonal transform can be Discrete Fourier Transform, Discrete Cosine Transform or Discrete Wavelet Transform. The perceptual model is used to select those regions suitable for watermark embedding, which is very important for invisible watermarking.

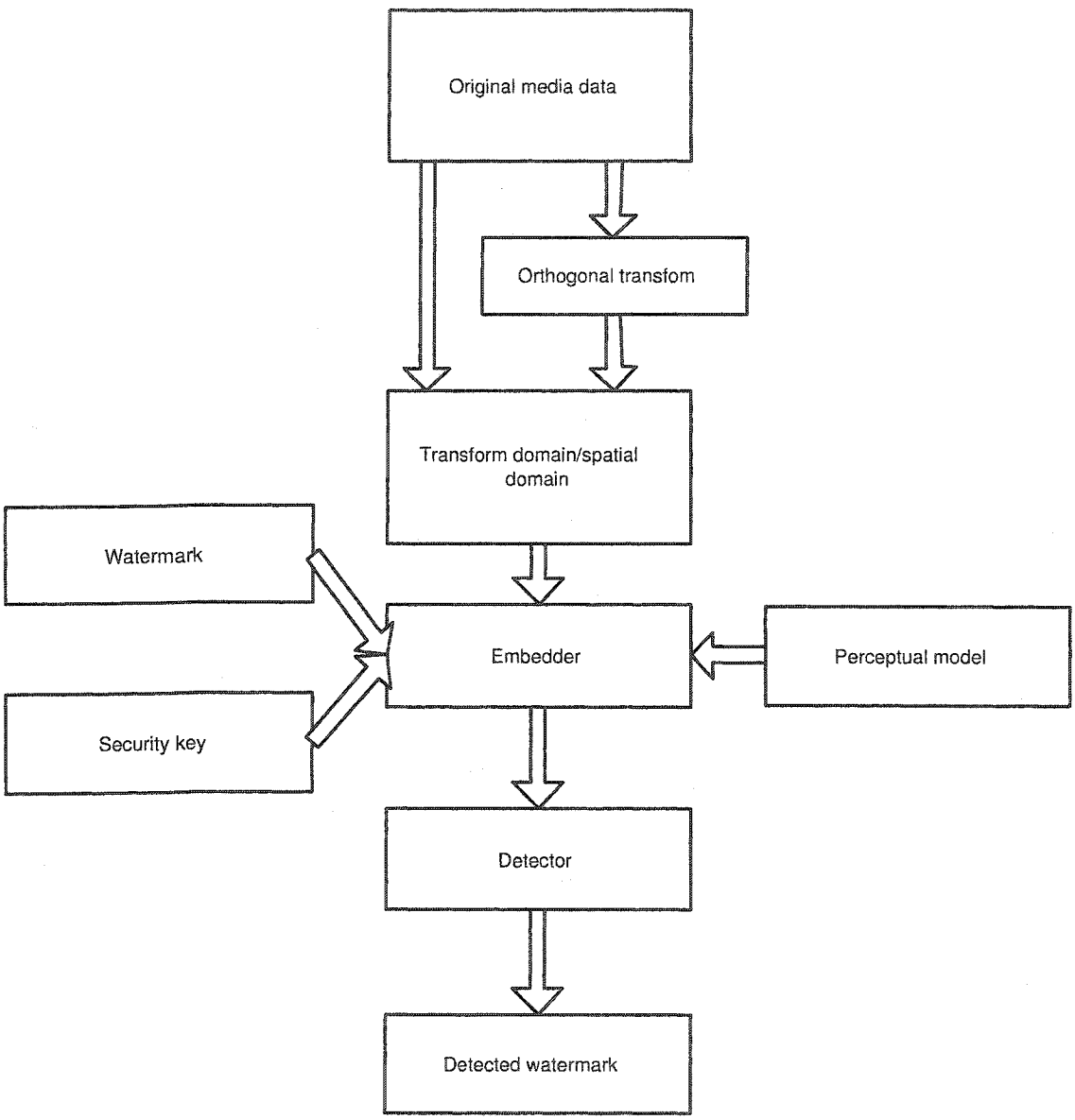



**Figure 1.3:** The block diagram of digital image watermarking.

## 1.4 The evaluation of the performance of a watermarking system

At present, the research on image watermarking is focused on the robust and invisible image watermarking. For such a image watermarking system, the following three requirements are often used to evaluate the performance of the system.

1. Invisibility. Invisibility means the watermark should be embedded into the host media data invisibly. In other words, we should keep the fidelity of the host image after the embedding processes.

2. Robustness. Robustness means that the embedded watermark should be robust against various attacks and processing techniques. For the digital image watermarking, a good watermarking scheme should be robust against filtering processing, noise addition, geometry transformation such as rotating, scaling and translating, lossy compression such as JPEG compression.

3. Capacity. Capacity means the watermark should be able to carry enough information for the purpose of copyright protection and authentication.

A good watermarking scheme should achieve a good trade-off among these requirements, refer to Fig. 1.4.

## 1.5 The application of digital watermarking

Digital watermarking systems are developed based on the applications:

1. Copyright protection and authentication. One of the main reasons for watermarking is for copyright protection. The idea is to embed information about the

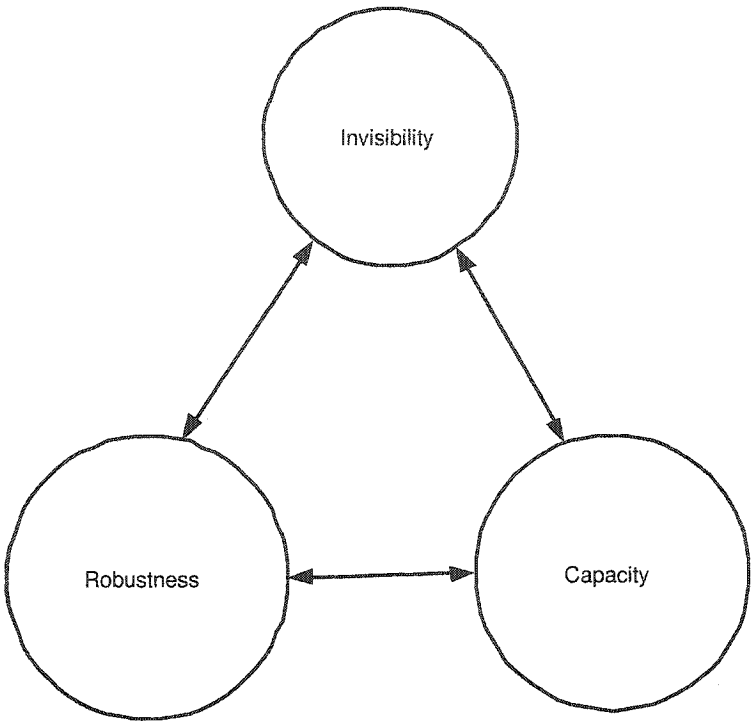



**Figure 1.4:** The requirements for a robust digital image watermarking system.

copyright owner into the data to prevent other parities from claiming to be the rightful owners of the data. The watermarks used for that purpose are supposed to be very robust against various attacks intended to remove the watermark.

2. Proof of ownership. The watermark should be unambiguous. The ambiguity attack is to embed additional watermark to claim to be the owner. The watermark used as a proof of ownership should be able to resolve rightful ownership.

3. Content authentication. To be able to authenticate the content, any changes or tampering with the content should be detected. This can be achieved through the use of "fragile/semi-fragile watermarking" which has low robustness to certain modifications such as lossy compression, noise pollution. Among all possible

watermarking applications, authentication watermarks require lowest level of robustness.

4. Content copy protection. It is desirable in some system to have a copy protection mechanism to prevent copying of the content. A watermark can be used in such systems to prevent the copying of the content data or limit the number of copying.

## 1.6 Watermarking benchmarking tools

1. Stirmark [1]. Stirmark is the first benchmarking tool developed at the University of Cambridge for digital watermarking technologies. Given a watermarked image, Stirmark can generate a number of image modifications which can be used to verify the watermarking scheme. Version 4 of Stirmark has evolved into a fully automated test bench tools. The attacks include: cropping, flip, rotation, sharpening, Gaussian filtering, line removal and JPEG compression.

2. Checkmark [2]. Checkmark is a benchmarking suite for digital watermarking developed at the University of Geneva. It provides efficient and effective tool to evaluate and rate watermarking schemes.

3. Certimark [3]. Certimark is a new benchmarking tool being developed. The focus of the design and development of Certimark is to develop a complete benchmarking suite for still image and video watermarking technologies.

Stirmark is very popular and has been used widely to evaluate the performance of the image watermarking schemes.

## 1.7 The contributions of the thesis

This thesis presnets a background review of RST invariant watermarking techniques. Based on the algorithm proposed by O'Ruanaidh [4], we propose in this thesis [5][6][7][8][9] a LPM and phase correlation based digital watermarking scheme that is invariant to RST transformations. We use phase correlation in LPM domain to rectify the watermark position and exhaustive search to detect the watermark if the original image is not available. The main contributions of the work include the idea of using phase correlation spectrum in digital image watermarking, and the simple and feasible implementation of RST invariant watermarking scheme in LPM domain.

We implement the algorithm and experiments show that it is robust against RST (rotation, scaling and translation) transformations, JPEG compression and other attacks, and that it can preserve the fidelity of the watermarked image.

### Publications generated from the research:

1. Dong Zheng, Jiying Zhao, and Abdulmotaleb El Saddik, RST Invariant Digital Image Watermarking Based on Log-Polar Mapping and Phase Correlation, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Authentication, Copyright Protection and Information Hiding, to appear in September 2003 issue.

2. Dong Zheng and Jiying Zhao, Apply phase information in RST image watermarking, IEEE International Conference on Consumer Electronics (ICCE2003), Los Angeles, California, USA, June 17-19, 2003.

3. Dong Zheng and Jiying Zhao, LPM-Based RST Invariant Digital Image Watermarking, IEEE Canadian Conference on Electrical and Computer Engineering

(CCECE) 2003, Montreal, Canada, May 4-7, 2003.

4. Dong Zheng and Jiying Zhao, RST Invariant Digital Image Watermarking: Importance of Phase Information, IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2003, Montreal, Canada, May 4-7, 2003.

5. Dong Zheng and Jiying Zhao, A Novel RST Invariant Digital Image Watermarking Scheme, Multispectral image processing and pattern recognition (MIPPR) conference 2003, Beijing, China, October 20-22, 2003.

## 1.8 Thesis structure

In Chapter 2, we give a brief introduction to various digital image watermarking techniques, in Chapter 3 we propose our scheme, in Chapter 4 we describe the watermark embedding, watermark extraction procedures and list several strategies used for implementing our scheme, in Chapter 5 we illustrate and evaluate the proposed scheme and in Chapter 6 we conclude the thesis and give some suggestions and ideas for future research work.

# Chapter 2

# Literature review

## 2.1 Introduction to various digital image watermarking techniques

According to the domain in which the watermark information is embedded in the image, digital image watermarking techniques can be classified as spatial domain and transform domain techniques as shown in Fig. 1.3.

One of the first used techniques for image watermarking appeared in 1993. Tirkel et al. [10] presented two techniques to hide data in the spatial domain of images. These methods were based on the pixel value's Least Significant Bit (LSB) modifications. A scheme proposed by Kurah and McHughes [11] known as image downgrading. Given two images of same size, one acts as cover and the other the watermark image. The most significant bits of the watermark image are taken and embedded in the least significant bits of the cover image. Extracting the least significant bits of the watermarked image can give a rough estimation of the watermark image. Bruyndonchx et al. [12] proposed a scheme based on the pixel region classification. Pixels are classified into homogeneous

luminances zones. Then the pixels have their gray levels changed following a rule that takes into the account where the pixel is inserted, and the value of the bit to be embedded.

Spread spectrum and transform domain watermarking was introduced by Cox et al. [13]. Cox's approach uses spread spectrum communication techniques to embed a single bit in the image. Pickholtz et.al [14] define spread spectrum communications as:

"Spread spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery."

Based on the Cox' work, J. O'Ruanaidh et al. [4] proposed a spread spectrum based watermarking approach. The watermark is embedded in the form of a pseudo-random sequence. In order to embed the watermark or to extract it, it is important to have access to the key which is simply the seed used to generate the pseudo-random sequences. A good spread spectrum sequence is one which combines desirable statistical properties such as uniformly low cross correlation with cryptographic security.

Most of the watermarking techniques in the spatial domain are easy to embed but does not provide much resistance against attacks. The transform/spectral domain watermarking has proved to be a better choice for robust image watermarking. Watermark has been embedded using the Discrete Cosine Transform (DCT) [15][16], Discrete Fourier Transform[17][4] and Discrete Wavelet Transform[18][19][20].

Bors et al. [16] proposed a DCT based watermarking scheme. The image is first divided into 8*8 pixel blocks. After DCT transform and quantization, the mid-frequency range DCT coefficients are selected based on a Gaussian network classifier. The mid-frequency range DCT coefficients are then used for embedding. Those coefficients are

modified using a linear DCT constraints. It is claimed that the algorithm is resistant to JPEG compression.

Cox et al. [13] used the spread spectrum to embed the watermark in the frequency components of the host image. First the Fourier Transform is applied to the host image and a sequence of values $V$ from the magnitude components is selected. The watermark is inserted to obtain an modified values $V'$ using the following equation:

$$V' = V + \alpha \times W \tag{2.1}$$

The scaling parameter $\alpha$ is used to determine the embedding strength of the watermark. Different spectral components exhibit different tolerance to modification. To verify the presence of the watermark, the cross correlation value between the extract watermark $W'$ and the original watermark $W$ is computed as following:

$$sim = \frac{W' \times W^T}{\sqrt{(W' \times W'^T)(W \times W^T)}} \tag{2.2}$$

Here we call the cross correlation the similarity (sim). Experimental results showed that this method resists JPEG compression as a quality factor of 5%, scaling, dithering, cropping and collusion attacks.

Kundar and Hatzinakos [19] proposed a wavelet based watermarking scheme. The multiresolution data fusion is used for embedding where the image and the watermark are both transformed into the discrete wavelet domain. The watermark is embedded into each wavelet decomposition level of the host image. During extracting, the watermark is an average of the estimates from each resolution level of wavelet decomposition. This scheme is robust against JPEG compression, additive noise and filtering opera-

tions.

Contrary to the LSB approach, the key to make a watermark robust is that it should be embedded in the perceptually significant components of the image [13][18][21]. A good watermark is one which takes into account the behavior of human visual system. For the spread spectrum based watermarking algorithm, a scaling factor can be used to control the amount of energy a watermark has. The watermark energy should be strong enough to withstand possible attacks and distortions. Meanwhile a large watermark energy will affect the visual quality of the watermarked image. A perceptual model is needed to adjust the value of the scaling factor based on the visual property of the host image to achieve the optimal trade-off between robustness and invisibility.

The human visual system (HVS) shows variable sensitivities based on the properties of images. These properties include frequency, luminance sensitivity, color and contrast masking. A large smooth area of the image corresponds to low frequency component, while the heavily textured area corresponds to high frequency. In practice, an empirical perceptual model [4] that the watermark is embedded into the middle frequency component is widely used. The reason is that the watermark embedded in the high frequency is easily removed by attacks such as low pass filtering and JPEG compression, while embedding the watermark in the low frequency will affect the visual quality of the host image dramatically. It was also shown that the human eyes is less sensitive to the bright area of the image. So the watermark can be embedded with different strengths according to a luminance function to achieve the trade-off between robustness and invisibility [22]. This perceptual model can be used in the spatial domain watermarking scheme. More complicated perceptual models take luminance sensitivity and texture masking into account [23][24]. The watermark is embedded according to luminance and texture masking (such as edge detection). A set of empirically adjusted parameters is required.

For the color channel model, it is found that the human eyes is less sensitive to the changes in the blue channel. So for the color image watermarking, we can take advantage of the property of the human eye to used the blue color component for watermark embedding.

## 2.2 RST invariant digital image watermarking

In order for a watermark to be useful, it must be robust against a variety of possible attacks by pirates. These include robustness against compression such as JPEG, scaling and aspect ratio changes, rotation, cropping, row and column removal, addition of noise, filtering, cryptographic and statistical attacks, as well as insertion of other watermarks. While many methods perform well against compression, they lack robustness to geometric transformations [25]. Rotation and scaling attacks are considered more challenging than other attacks. This is due to the fact that changing the image size or its orientation, even by slight amount, could dramatically reduce the receiver ability to retrieve the watermark [26]. Recently it has been clear that even very small geometric distortions can prevent the detection of a watermark [27].

O'Ruanaidh et al. [4] first have outlined the theory of integral transform invariants and showed that this can be used to produce watermarks that are resistant to rotation, scaling, and translation. In their approach the Discrete Fourier Transform (DFT) of an image is computed and then the Fourier-Mellin transform is performed on the magnitude, the watermark is embedded in the magnitude of the resulting transform. The watermarked image is reconstructed by performing the inverse transforms (an inverse DFT and an inverse Fourier-Mellin transform) after considering the original phase [4][26]. Fourier-Mellin transform is a log-polar mapping (LPM) followed by a Fourier

transform, while an inverse Fourier-Mellin transform is an inverse log-polar mapping (ILPM) followed by an inverse Fourier transform. In the scheme, the embedded watermark may be extracted by transforming the watermarked image into RST invariant domain. However, they noted very severe implementation difficulties which might have hampered further work in this area [27]. The change of coordinate system means that the interpolation is needed. In our experiment, the LPM and ILPM will cause a unacceptable loss of image quality as shown in Fig. 3.9 even the bilinear interpolation is used.

O'Ruanaidh et al. [4] noticed this and suggested to embed watermark in the RST invariant domain independently of the original image, so that the original image will not suffer from LPM and ILPM transform theoretically. The problem is that the watermark still need to go through the ILPM and LPM which may cause the loss of the watermark data. The LPM is a sampling process, while oversampling in the center area and downsampling in the edge region. While the magnitude spectrum of an image generally has large magnitude components in the center (the low frequency areas), so the computation of the Fourier mellion transform is problematic because the interpolation only performs welll if the neighboring samples are of the same value level. So the scheme is difficult to implement according to our experiments.

Another strategy for detecting watermarks after geometric distortion is to identify what the distortions are, and invert them before applying the watermark detector. This can be done by embedding a template along with the watermark. Pereira et al. [25] proposed to embed two watermarks, a template and a spread spectrum message containing the information or payload. The template contains no information itself, but is used to detect transformations undergone by the image. Once the template is detected, these transformations are inverted and the spread spectrum signal is retrieved.

"One problem with this solution is that, because it requires the insertion of a registration watermark in addition to the data-carrying watermark, this approach is likely to reduce the image quality. A second problem arises because all image watermarked with this method will share a common registration watermark. This fact may improve collusion attempts to discern the registration pattern and, once found, the registration pattern could be removed from all watermarked images thus restricting the invertibility of any geometric distortions." [27].

Lin et al. [27] proposed a method that develops a watermark invariant to geometric distortions, and that eliminates the need to identify and invert them. The watermark is embedded into a translation and scaling invariant one-dimensional signal obtained by taking the Fourier transform of the image, resampling the Fourier magnitudes into log-polar coordinates, and then summing a function of those magnitudes along the log-radius axis. The scheme handles rotations by exhaustive search. We suspect that the probability of false positive of the algorithm is high due to the summing and exhaustive search, and that the exhaustive search is time-consuming. Also rotations of fractional degrees should be considered. And for this approach, it is difficult to utilize the perceptual model to achieve the trade-off between robustness and invisibility because of the one-dimensional projection and summing process.

Another approach for resisting geometric attacks is based on synchronizing the watermark that is embedded in an image with the correlating watermark using image features. This is achieved by geometrically transforming a reference watermark based on features of the original image content during the embedding process, while in the detection process, features of the watermarked and possibly distorted image content are used in order to geometrically transform the reference watermark or even the image itself. Such an approach was presented in [28]. Corner points and facial feature points

can be used for this purpose.

Some research are focusing on the Radon transform [29] in recent years. Two one-dimensional Generalized Radon transformations, the Radial Integration Transform and the Circular Integration Transform, can be used to extract characteristic values. Then the corresponding geometric transformation parameters can be calculated. In this way, the geometrically transformed watermark that is embedded in the image can be synchronized for detection. Some Radon transform based watermarking schemes [30] [31] have been proposed.

Hyung et al. [32] proposed a RST invariant watermarking scheme which uses the bispectrum feature vector of an image as the watermark. The bispectrum is the Fourier spectrum of the triple correlation of a signal. Phases of the integrated bispectra are invariant to translation and scaling. And the rotation invariance is achieved using the Radon transform. An image is decomposed into the 1-D projections and a feature vector is constructed. A watermark is embedded by modifying the vector.

It is widely accepted that phase plays an important, and often crucial, role in vision and image representation [33][34]. Experiments [35] show that images reconstructed from the original phase and the magnitude taken from another different source closely resemble the original ones, unlike the case of images reconstructed from magnitude only. Phase correlation based methods have been proposed to align two images which shifted relative to each other [36][37].

Based on the Fourier Mellion Transform, we proposed a simple and feasible RST invariant watermarking scheme in LPM domain which can achieve an optimal trade-off between robustness and invisibility and solve the problem mentioned above about the difficulty of the implementation. It is robust against various geometrical transformation attacks. Also this scheme uses phase correlation in LPM domain to rectify the water-

mark position. If the original image is not available for phase correlation computation, exhaustive search is used for watermark detection. The main contributions of the work include the idea of using phase correlation spectrum in digital image watermarking and the feasible implementation of the log-polar mapping based watermarking scheme.

# Chapter 3

# The proposed RST invariant image watermarking scheme

## 3.1 Discrete Fourier Transform

The DFT (Discrete Fourier Transform) of an image $f(x, y)$ of size $M \times N$ and the corresponding IDFT (Inverse DFT) are defined as follows [38]:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M+vy/N)} \tag{3.1}$$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M+vy/N)} \tag{3.2}$$

The Fourier magnitude spectrum and phase angle are defined as follows:

$$|F(u,v)| = [R^2(x,y) + I^2(x,y)]^{1/2} \tag{3.3}$$

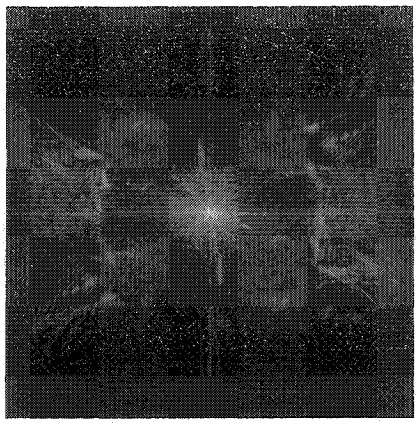


**Figure 3.1:** The magnitude spectrum of image Barbara.

$$\phi(u,v) \;=\; tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right] \tag{3.4}$$

where $R(u,v)$ and $I(u,v)$ are the real and imaginary parts of $F(u,v)$, respectively.

The magnitude spectrum of image Barbara image is shown in Fig. 3.1. The image reconstructed from the magnitude is shown in Fig. 3.2 and the image reconstructed from the phase is shown in Fig. 3.3. Here we can clearly see that the image reconstructed from the phase closely resemble the original image, unlike the image reconstructed from the magnitude.

## 3.2 The properties of Discrete Fourier Transform

In this section, we present some basic properties of the DFT related to geometric transformations on the spatial domain.

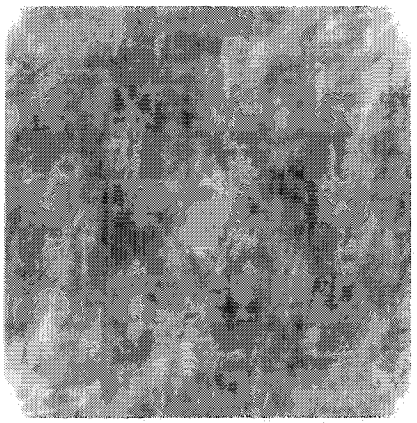


**Figure 3.2:** The reconstructed image using only the magnitude spectrum.

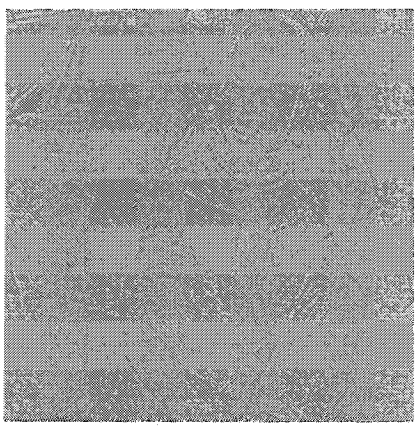
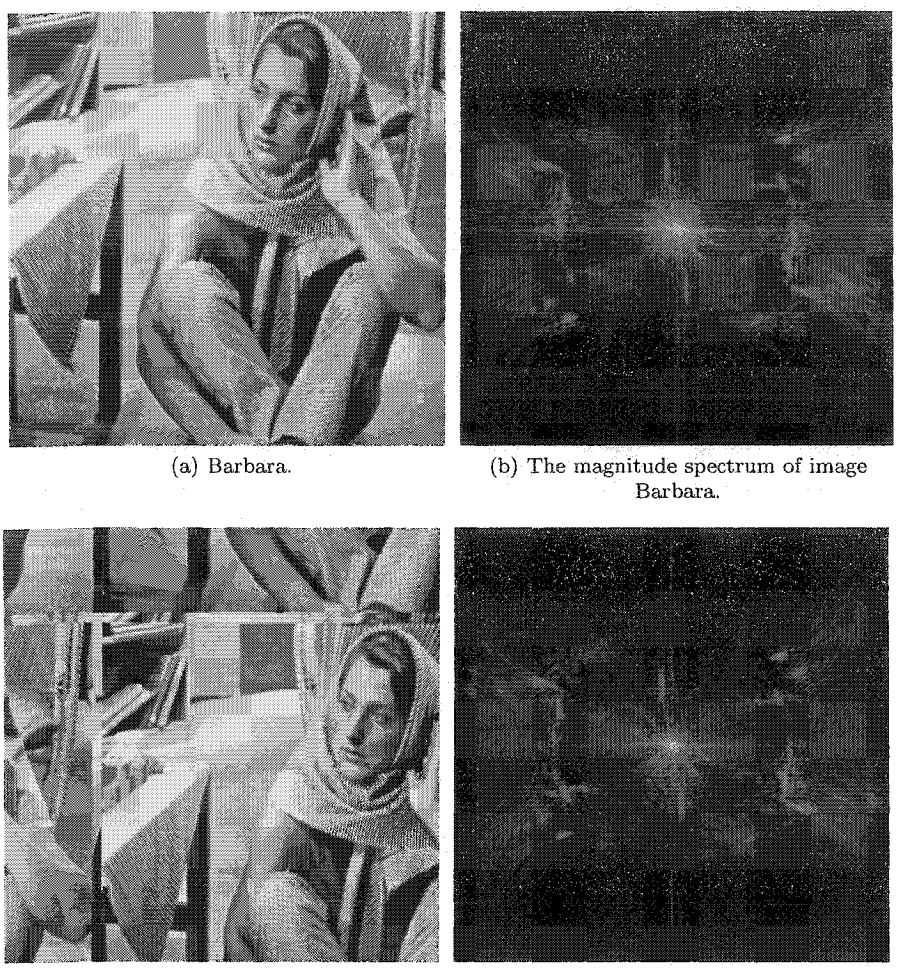


**Figure 3.3:** The reconstructed image using only the phase spectrum.

### 3.2.1 Translation

The $F(u,v)$ is Fourier transform of the image $f(x,y)$. A shift in the spatial domain of $f(x,y)$ will cause a linear shift in phase of $F(u,v)$ and will not change the magnitude spectrum as shown in Fig. 3.4.

$$F(u,v)e^{\left[\frac{-j2\pi}{N}\right](au+bv)} = f(x+a,y+b) \tag{3.5}$$



(a) Barbara.

(b) The magnitude spectrum of image Barbara.

(c) The circular translation of image Barbara.

(d) The magnitude spectrum of the circular translation.

**Figure 3.4:** Translation property of Fourier transform.

### 3.2.2 Rotation

Suppose $i_1$ is the rotated version of $i_0$ in spatial domain,

$$\begin{aligned} i_1(x,y) &= i_0((x\cos\alpha + y\sin\alpha), \\ &\quad (-x\sin\alpha + y\cos\alpha)) \end{aligned} \tag{3.6}$$

The Fourier transform of $i_1(x, y)$ and $i_0(x, y)$ are respectively $I_1(u, v)$ and $I_0(u, v)$, and their magnitudes are related by :

$$\begin{aligned} |I_1(u,v)| &= |I_0((u\cos\alpha + v\sin\alpha), \\ &\quad (-u\sin\alpha + v\cos\alpha))| \end{aligned} \tag{3.7}$$

As shown in Fig. 3.5, when the image is rotated in the spatial domain by some degree, its magnitude spectrum will be rotated by the same degree.

### 3.2.3 Scaling

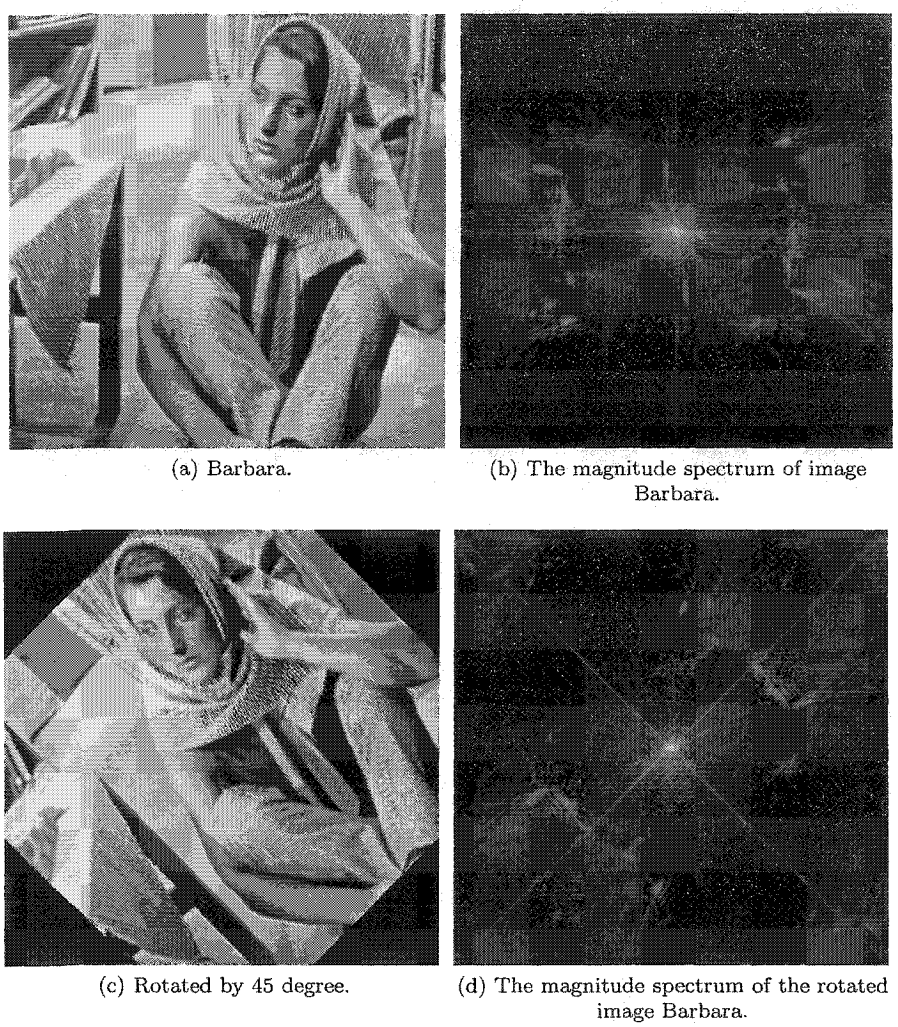
Scaling in the spatial domain can cause an inverse scaling in the frequency domain.

$$f(ax, by) = \frac{1}{|ab|} F(\frac{u}{a}, \frac{v}{b}) \tag{3.8}$$

This property is shown in Fig. 3.6.

## 3.3 Log-polar mapping and inverse log-polar mapping

The log-polar mapping is a conformal mapping from the points on the Cartesian plane (x, y) to points on the log-polar plane ($\rho$, $\theta$):



(a) Barbara.

(b) The magnitude spectrum of image Barbara.

(c) Rotated by 45 degree.

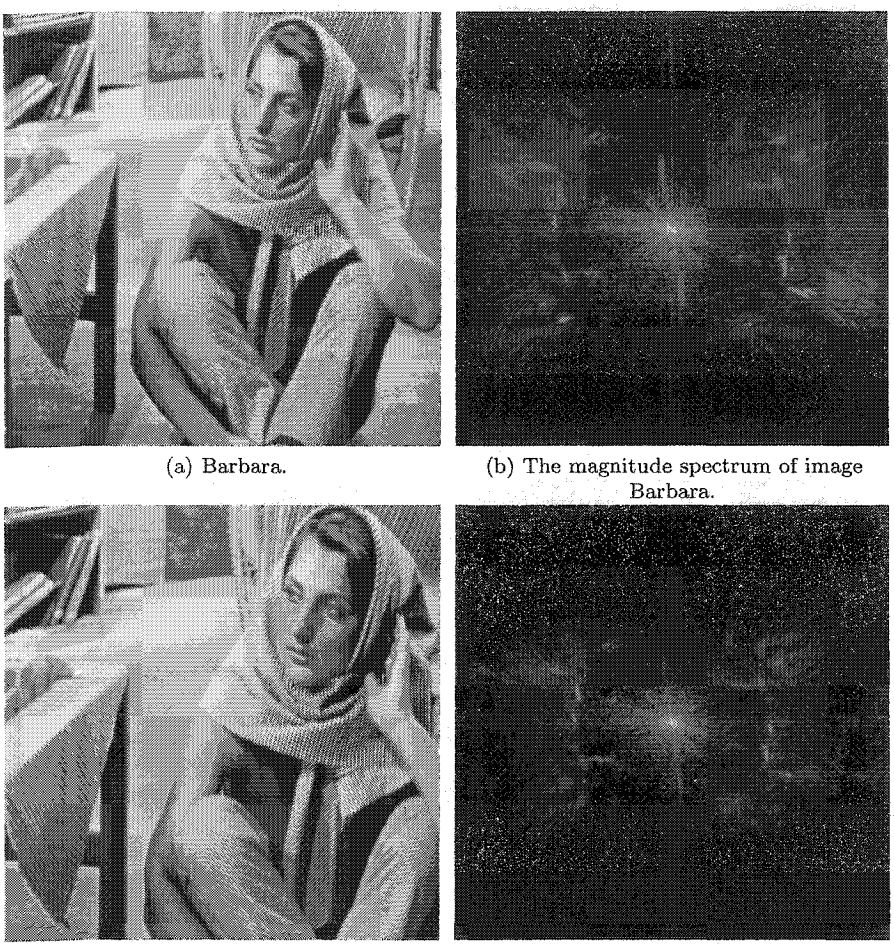
(d) The magnitude spectrum of the rotated image Barbara.

**Figure 3.5:** Rotation property of Fourier transform.

$$x = e^{\rho}\cos\theta \tag{3.9}$$

$$y = e^{\rho}\sin\theta \tag{3.10}$$

As shown in Fig. 3.7, the log-polar mapping is just like a sampling process. The sampling points on the Cartesian plane are used to construct the transformed image on



(a) Barbara.

(b) The magnitude spectrum of image Barbara.

(c) Scaled by 1.2.

(d) The magnitude spectrum of the scaled image Barbara.

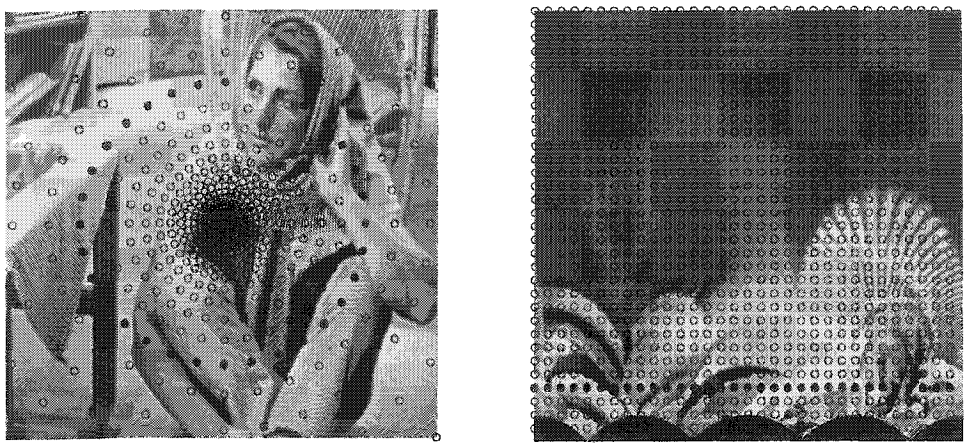
**Figure 3.6:** Scaling property of Fourier transform.

the log-polar plane. The bilinear interpolation is used to compute the values of those sampling points.

The inverse log-polar mapping is:

$$\rho = \ln(\sqrt{x^2 + y^2}) \tag{3.11}$$

$$\theta = tan^{-1}(\frac{y}{x}) \tag{3.12}$$



(a) Sampling points on the Cartesian plane.

(b) The corresponding points on the log-polar plane after log polar mapping.

**Figure 3.7:** Log polar mapping.

The log-polar mapping of Barbara and the inverse log-polar mapping are shown in Fig. 3.8 and Fig. 3.9. Bilinear interpolation is used in the computation of LPM and ILPM.

## 3.4 The proposed log-polar mapping based image watermarking scheme

We can write the relationship of an image $i_0(x, y)$, and a rotated, scaled, and translated version of the image, $i_1(x, y)$, as follows [4][27]:

$$\begin{aligned} i_1(x,y) &= i_0(\sigma(x\cos\alpha + y\sin\alpha) - x_0, \\ &\quad \sigma(-x\sin\alpha + y\cos\alpha) - y_0) \end{aligned} \tag{3.13}$$

**Figure 3.8:** LPM of image Barbara.

where the RST parameters are $\alpha$, $\sigma$, and $(x_0, y_0)$ respectively.

The Fourier transform of $i_1(x, y)$ and $i_0(x, y)$ are respectively $I_1(u, v)$ and $I_0(u, v)$, and their magnitudes are related by [4][27]:

$$\begin{aligned} |I_1(u,v)| &= |\sigma|^{-2} |I_0(\sigma^{-1}(u\cos\alpha + v\sin\alpha), \\ &\quad \sigma^{-1}(-u\sin\alpha + v\cos\alpha))| \end{aligned} \tag{3.14}$$

Equ. (3.14) is independent of the translational parameters $(x_0, y_0)$, which is the translation property of the Fourier transform [39].

Rewrite Equ. (3.14) by using log-polar coordinates:

**Figure 3.9:** The result of inverse LPM.

$$u = e^{\rho} \cos\theta \tag{3.15}$$

$$v = e^{\rho} \sin\theta \tag{3.16}$$

where $\rho \in \Re^2$ and $0 \leq \theta < 2\pi$. For how to calculate $\rho$ and $\theta$, refer to Section 4.3.1. Then the magnitude of the Fourier spectrum can be written as [27][4]:

$$\begin{aligned} |I_1(u,v)| &= |\sigma|^{-2} |I_0(\sigma^{-1} e^{\rho} \cos(\theta - \alpha), \\ &\quad \sigma^{-1} e^{\rho} \sin(\theta - \alpha))| \end{aligned} \tag{3.17}$$

or

$$|I_1(\rho,\theta)| = |\sigma|^{-2}|I_0(\rho - \ln\sigma, \theta - \alpha)| \tag{3.18}$$

Equ. (3.18) demonstrates that the amplitude of the log-polar spectrum is scaled by $|\sigma|^{-2}$, that image scaling results in a translational shift of $\ln\sigma$ along the log-radius $\rho$ axis, that image rotation results in a cyclical shift of $\alpha$ along the angle $\theta$ axis, and that image translation has no effects in LPM domain.

According to the translation property of the Fourier transform, the Fourier transforms of $I_1$ and $I_0$ is related by

$$F_1(\omega_\rho,\omega_\theta) = |\sigma|^{-2}e^{-j(\omega_\rho\cdot\ln\sigma+\omega_\theta\cdot\alpha)}F_0(\omega_\rho,\omega_\theta) \tag{3.19}$$

The Fourier magnitude of the two LPM mappings is related by

$$|F_1(\omega_\rho,\omega_\theta)| = |\sigma|^{-2}|F_0(\omega_\rho,\omega_\theta)| \tag{3.20}$$

where $F_1$ and $F_0$ are respectively the DFT of $I_1$ and $I_0$.

The phase difference between the two LPM mappings is directly related to their displacement, given by $e^{j(\omega_\rho\cdot\ln\sigma+\omega_\theta\cdot\alpha)}$.

Equ. (3.20) is equivalent to computing the Fourier-Mellin transform [4]. Equ. (3.20) demonstrates that the amplitude of Fourier-Mellin spectrum is scaled by $|\sigma|^{-2}$ caused by scaling transform, and is invariant to rotation and translation. $|\sigma|^{-2}$ will cause no

problem at all since we use normalized correlation to detect watermarks, so Fourier-Mellin transform is truly invariant to RST.

The original Fourier-Mellin based watermarking algorithm was proposed by O'Ruanaidh et al. [4]. One of the significant contribution of the paper is the novel application of Fourier-Mellin transform to digital image watermarking. Theoretically Fourier-Mellin domain is the best place to embed watermark, considering it is invariant to RST. In practice, the original image will need to endure both the LPM and ILPM, which make the image quality unacceptable. Ruanaidh et al. noticed the problem, so they proposed an alternative algorithm. Using this algorithm, only the watermark data goes through the ILPM, then it is inserted into the magnitude spectrum of the image. Applying the inverse DFT to the modified magnitude spectrum, one can get the watermarked image. For details of both the algorithms, refer to [4]. However there are several disadvantages in this algorithm, first the watermark data need go through the ILPM which will cause the distortion of the watermark signal. During the embedding process, it is almost impossible to find a good method to insert the watermark data (after the IDFT and ILPM) into the desired location in the image magnitude spectrum. So it is extremely hard to achieve the tradeoff between the invisibility of watermark and the robustness of the watermark.

Based on the second proposal by O'Ruanaidh et al., we propose a new scheme here, as shown in Fig. 4.1, Fig. 4.2 and Fig. 4.3. While Section 4.1 and Section 4.2 will explain the watermark embedding and extraction scheme in detail, the following outlines several important points of the scheme:

We embed watermark in LPM domain to simplify the effects of RST transformations into simple shifts (refer to Equ. (3.18)). For watermark embedding, the approximate ILPM is employed to replace ILPM, in order to eliminate the imprecision caused by

ILPM. Therefore actually watermarks are embedded in the Fourier magnitude spectrum of the original image, to achieve the effect of being embedded in LPM domain.

Since we do not apply the IDFT before the approximate ILPM, rotation and scaling operation in the spatial domain of the watermarked image will cause translation of the watermark positions in LPM domain, either a circular shift along the angle axis or the vertical shift along the log-radius axis (refer to Equ. (3.18)). Exhaustive search in the embedding area can be used to handle the shift of watermark positions caused by rotation and scaling [5]. However exhaustive search is time consuming and produces large correlation coefficient for unwatermarked images. Therefore, we can use phase correlation to rectify the watermark position to avoid exhaustive search [6] if the original image is available.

## 3.5 The phase correlation

Refer to Equ. (3.18) and Equ. (3.19), if we compute the cross-power spectrum of $F_1$ and $F_0$ as follows [36]:

$$C_{10} = \frac{F_1(\omega_\rho, \omega_\theta) F_0^*(\omega_\rho, \omega_\theta)}{|F_1(\omega_\rho, \omega_\theta) F_0^*(\omega_\rho, \omega_\theta)|} = e^{j(\omega_\rho \cdot \ln \sigma + \omega_\theta \cdot \alpha)} \tag{3.21}$$

where $F^*$ is the complex conjugate of $F$, the translation property guarantees that the phase of the cross-power spectrum is equivalent to the phase difference between the images. Furthermore, if we represent the phase of the cross-power spectrum in its spatial form, i.e., by taking the inverse Fourier transform of the representation in the frequency domain,

$$D_{10} = IDFT(angle(C_{10})) \tag{3.22}$$

where $IDFT$ is inverse Fourier transform, and $angle(C_{10})$ is the phase of $C_{10}$.

Based on the property of the Fourier transform, the Fourier transform of function $\delta(x-d)$ is $e^{-jwd}$. Equ. (3.22) gives a two-dimensional $\delta$ function centered at the displacement. So $D_{10}$ is a function which is an impulse, that is, it is approximately zero everywhere except at the displacement.

Our method determines the location of peak of $D_{10}$, and consequently calculate the watermarking position. Since the phase difference for every frequency component contributes equally, the location of the peak will not change if there are noises caused by watermark embedding, black pixel padding, and JPEG compression. The idea eliminates the need for exhaustive search, reduces the correlation coefficient calculated for unwatermarked images, and saves computation time.

# Chapter 4

# Implementation

## 4.1 Watermark embedding scheme

The procedure of embedding a watermark consists of the following steps (refer to Fig. 4.1):

1. First, use pseudo-noise (PN) code generator to generate a watermark data sequence, which is spread spectrum consisting of both positive and negative values.

2. Compute DFT of the original image. The magnitude spectrum of DFT is positive, while the watermark is a sequence of numbers that can be positive or negative. To be able to embed both positive and negative numbers, we need two numbers to represent one original watermark number. We encode positive numbers $x$ as $(x, 0)$ and negative numbers $x$ as $(0, x)$, so the length of watermark data sequence is doubled.

3. Select the desired locations in the LPM magnitude spectrum for embedding the watermark data sequence.

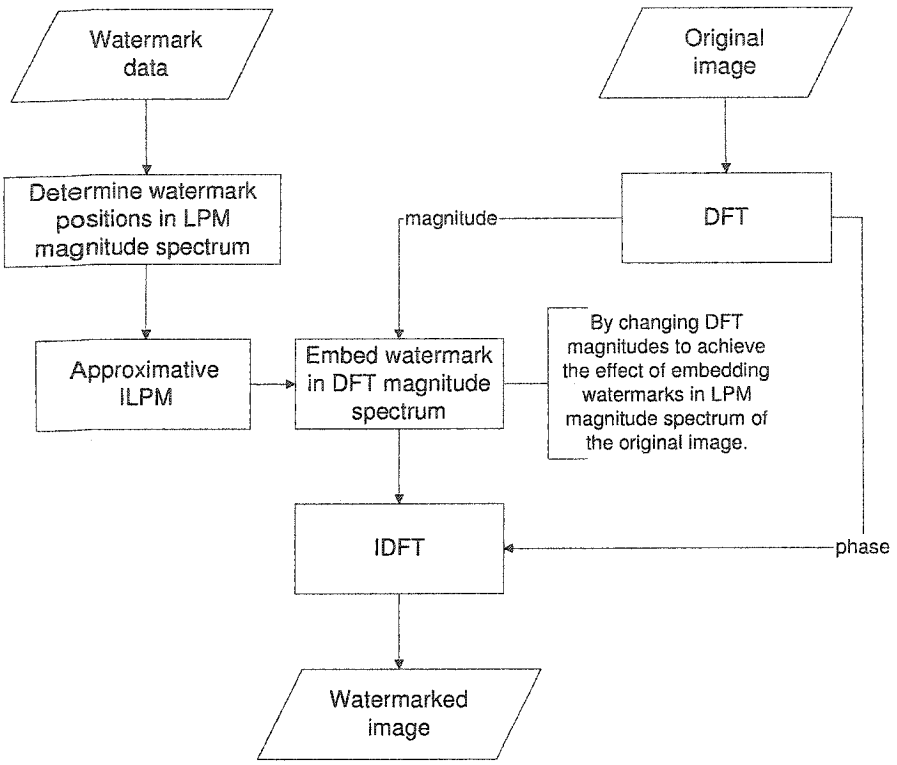



**Figure 4.1:** Watermark embedding scheme.

4. If we embed watermark in LPM domain, we need ILPM to transform back from LPM domain to DFT domain. To avoid computing ILPM that may bring unacceptable computational imprecision, we use the approximative ILPM and embed watermark in DFT domain. The watermarking locations in the Cartesian DFT magnitude spectrum is approximated from the watermark points in LPM domain selected in step 3.

   Naturally, if we want to change the value of one point in LPM magnitude spectrum for embedding watermark data, we only need to find the corresponding four points in Cartesian magnitude spectrum and change their values accordingly. For details, refer to Section 4.3.1.

5. Embed the watermark data into the DFT magnitude spectrum of the original image. Most algorithms use a simple embedding equation such as:

$$E' = E + \alpha * W \tag{4.1}$$

where $E$ is the DFT magnitude spectrum of the original image, $W$ is the watermark data, $E'$ is the modified DFT magnitude spectrum of the original image, and $\alpha$ is watermarking strength used to achieve the tradeoff between the robustness and the visibility of the watermark.

According to our experiments, by carefully selecting the watermarking positions and watermarking strength $\alpha$, the difference between the average value of $E$ and $\alpha * W$ can be small enough. So we can use Equ. (4.2) to replace the values of the embedding points by $\alpha * W$.

$$E' = \alpha * W \tag{4.2}$$

This embedding process will not change the amplitude values of those embedding points dramatically, therefore the goal of invisibility can be achieved. Meanwhile, the embedding method can simplify the extraction process.

The scaling operation will change the value of DFT magnitude spectrum, which is proportional to the scaling factor. The correlation function can be normalized for amplitude changes by using the correlation coefficient, which is in the range of $-1$ to 1, independent of scale changes in the amplitude (refer to Equ. (4.3)).

6. Finally apply inverse DFT to get the watermarked image.

During the embedding process, the symmetry of the DFT magnitude spectrum should be maintained, thus we must carefully select the desired points in the LPM magnitude spectrum. For details, refer to Section 4.3.3.

## 4.2 Watermark extraction scheme

### 4.2.1 Exhaustive search

Watermark extraction can be done with or without using original image. Exhaustive search can be used if original image is not available [5]. The rotation and scaling transformation in spatial domain result in a cyclically translational shift in LPM domain. So the watermark in LPM domain can only be shifted from its original place. We can use the exhaustive search method to simply shift the whole image in LPM domain vertically and horizontally pixel by pixel, retrieve the "watermark data" at the embedding position, compute the correlation, and choose the largest value as the extraction result.

Watermark extraction can be done using the exhaustive search method, without need of the original image, refer to Fig. 4.2.

1. Apply DFT and LPM to the watermarked image. We know that the rotation and scaling will only cause shifts in the LPM magnitude domain, therefore it is very easy to get the watermark by using exhaustive search.

2. We apply the exhaustive search to calculate the normalized correlation coefficient between the original watermark data and the extracted watermark data, by using Equ. 4.3. Here we call the normalized correlation coefficient the similarity.

3. If the value of similarity is larger than the threshold, the watermark is successfully extracted, otherwise, the watermark does not exist or we fail to detect it.

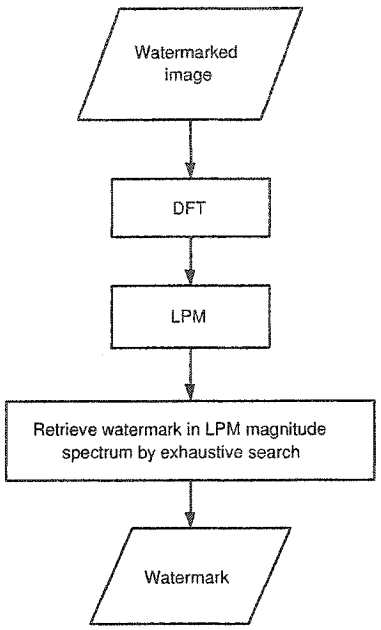



**Figure 4.2:** Watermark extraction scheme, exhaustive search

$$sim = \frac{W \times V^T}{\sqrt{(W \times W^T)(V \times V^T)}} \tag{4.3}$$

where $W$ and $V$ are respectively the original watermark vector and retrieved watermark vector, and $(\cdot)^T$ is the transpose operation of a matrix.

We carry out the exhaustive search in a predefined watermarking range. The extraction process does not need the original image. Because of the exhaustive search, the correlation values for unwatermarked images are random and in the region of 0.2 to 0.4. In all the cases, the correlation values for watermarked images are higher than 0.6. So the distance between the correlation values for watermarked images and the correlation values for unwatermarked images are big enough to give the correct indication whether watermark exists even under severe attacks.

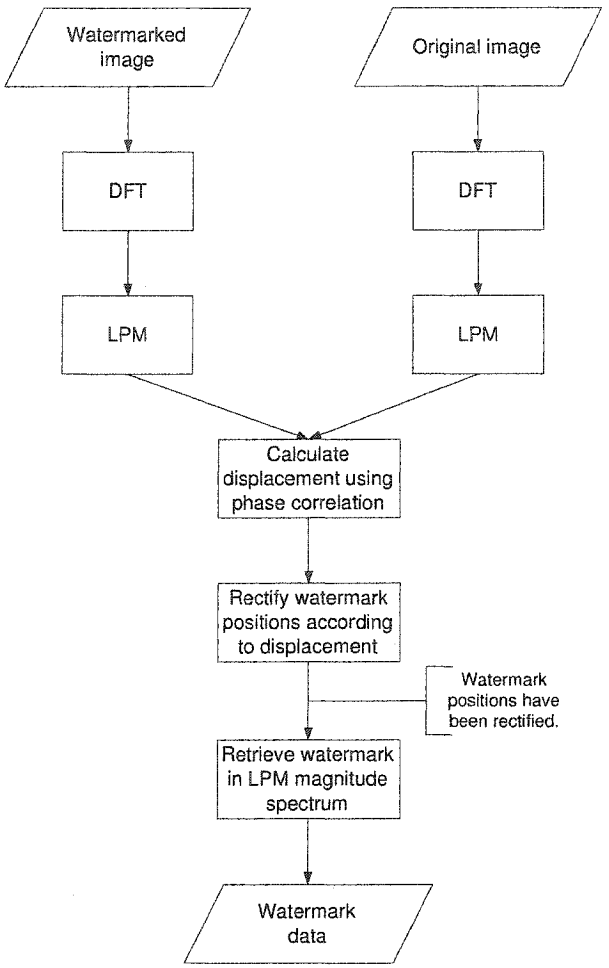



**Figure 4.3:** Watermark extraction scheme, phase correlation.

### 4.2.2 Phase correlation

This subsection gives the procedure of watermark extraction by using original image, which outperforms the exhaustive search scheme (refer to Fig. 4.3):

1. Apply DFT and LPM to both the original and the watermarked image, transform them into LPM domain.

2. Calculate displacement between the LPM of the original image and the LPM of the watermarked image, according to Equ. (3.21) and Equ. (3.22). For details,

refer to Section 4.3.5 and Section 5.1.

3. Rectify the original watermark position according to calculated displacement.

4. Retrieve the watermark data at the rectified location.

5. By using Equ. (4.3), calculate the normalized correlation coefficient between the original watermark data and the extracted watermark data. If the value of similarity is larger than the threshold, the watermark is successfully extracted, otherwise, the watermark does not exist or we fail to detect it.

## 4.3 Implementation strategies

In this section, we list several of the important problems we met when implementing our scheme proposed in Section 3, and our solutions to those problems.

### 4.3.1 LPM and ILPM

In log-polar mapping, pixels are indexed by ring number $R$ and wedge number $W$, related to ordinary $x$, $y$ image coordinates by the mapping [40]:

$$\begin{aligned} r &= [(x - x_c)^2 + (y - y_c)^2]^{1/2} \\ \theta &= tan^{-1} \frac{y - y_c}{x - x_c} \end{aligned} \tag{4.4}$$

$$R = \frac{(n_r - 1)\ln(r/r_{min})}{\ln(r_{max}/r_{min})} \tag{4.5}$$

$$W = \frac{n_w \theta}{2\pi}$$

where $(r, \theta)$ are polar coordinates, $(x_c, y_c)$ is the position of the centre of the log-polar sampling pattern, $n_r$ and $n_w$ are the numbers of rings and wedges respectively, and $r_{min}$ and $r_{max}$ are the radii of the smallest and largest rings of samples. We define log-polar radius $\rho$ as:

$$\rho = \ln r. \tag{4.6}$$

A log-polar sampled image is the one whose samples are centered on points mapping to integral $R$ and $W$, $R \in \{0, \ldots, n_r - 1\}$, $W \in \{0, \ldots, n_w - 1\}$. The separation between sample points is proportional to the distance from the sampling center.

Log-polar sampled images are often displayed on orthogonal $(R, W)$ axes, which is also called $(\rho, \theta)$ axes in this paper.

The LPM can be explained by the following equation:

$$P = L * C \tag{4.7}$$

where $P$ and $C$ are respectively the points in LPM magnitude spectrum and the points in Cartesian magnitude spectrum, while $L$ is the LPM computation operator.

In this scheme, the LPM and ILPM are the major causes of the image quality loss. Using bilinear interpolation, each point in log-polar magnitude spectrum is computed from a weighted average of four points in the Cartesian magnitude spectrum, shown in Equ. (4.8) and Fig. 4.4.

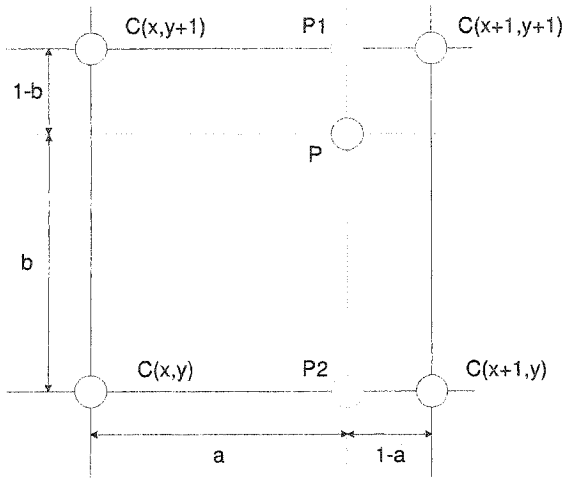


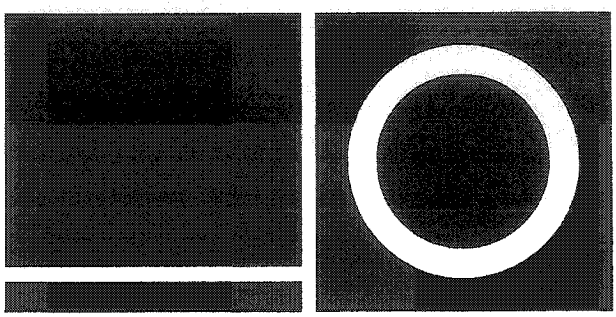

**Figure 4.4:** Bilinear interpolation.

$$\begin{aligned} P(\rho,\theta) &= C(x,y)\cdot(1-a)\cdot(1-b) \\ &+ C(x,y+1)\cdot(1-a)\cdot b \\ &+ C(x+1,y)\cdot a\cdot(1-b) \\ &+ C(x+1,y+1)\cdot a\cdot b \end{aligned} \tag{4.8}$$

where $C(x,y)$, $C(x,y+1)$, $C(x+1,y)$, and $C(x+1,y+1)$ are four points in Cartesian coordinate, $P(\rho,\theta)$ ($P$ in Fig. 4.4) is the corresponding point inside the square specified by the four points, and $a$ and $b$ are respectively the x-axis and y-axis coordinate difference between point $P$ and point $C(x,y)$.

If watermark data is embedded in log-polar magnitude spectrum, we need ILPM to get the corresponding position array in the Cartesian magnitude spectrum. We use an approximate ILPM instead to avoid computational imprecision.

Suppose that we want to insert watermark $M$ at position $P(\rho,\theta)$, so the value of



(a) LPM domain (b) Cartesian domain

**Figure 4.5:** Proposed watermark embedding position.

$P(\rho, \theta)$ should be added by $M$. From the value of $\rho$ and $\theta$, we can get the exact corresponding value of $x$ and $y$, then add $M$ to each of the four points points $C(x, y)$, $C(x, y+1)$, $C(x+1, y)$, and $C(x+1, y+1)$ in the Cartesian magnitude spectrum. Thus we can be sure that after log-polar mapping, the value of $P(\rho, \theta)$ is added exactly by $M$.

### 4.3.2 Watermark positions

Frequency domain watermarking is useful for taking advantage of perceptual criteria in the embedding process, for designing watermarking techniques which are robust to common compression techniques, and for direct watermark embedding of compressed bit streams [41]. For the digital watermarking scheme in frequency domain, to make the watermark robust to the attacks such as lossy compression and filter processing which may remove the high-frequency components in the magnitude spectrum, Cox et al. [42] proposed that the watermark should be embedded in the most significant frequency components. Thus the watermark can resist most attacks of lossy compression and filter processing. However, to make the watermark invisible and keep the fidelity of the image, we need to embed the watermark into the least significant frequency components.

To get the tradeoff, we choose the middle frequency components as the location to insert watermark data.

Meanwhile, because the log-polar mapping is just like a sampling process, the closer to the center, the higher the sampling rates. So if we insert the watermark data into the low frequency components, the change of the value of one point in the Cartesian magnitude spectrum will cause value changes of a lot of points in the log-polar magnitude spectrum because of the bilinear interpolation. That may cause imprecision in the extraction process.

So in our watermarking scheme, we use a simple and effective empirical perceptual model to embed watermark data into the middle frequency components. Experiments show the effectiveness of this approach.

For security issues, we can randomly choose to insert the watermark data into the points in the white region of LPM magnitude spectrum, as shown in Fig. 4.5 (a). Accordingly, the positions of points in Cartesian magnitude spectrum can be determined. Those points are located in the white region of Cartesian magnitude spectrum, as shown in Fig. 4.5 (b). The points are located in the middle frequency range, the tradeoff between the robustness and image fidelity can be achieved. The locations of these points can be randomly determined by a security key, without which the exact location of these points cannot be known. These points are randomly located in a region, therefore it is hard to retrieve the position information through observation, and it is hard to remove or attack the watermark maliciously.

The number of the watermark points depends on the length of the PN sequence, which is 64 in our experiments. Refer to Section 4.1, the length of watermark sequence is doubled in order to be able to embed both positive and negative watermark data. Refer to Section 4.1 and Section 4.3.1, we embed at four points in DFT domain to achieve the

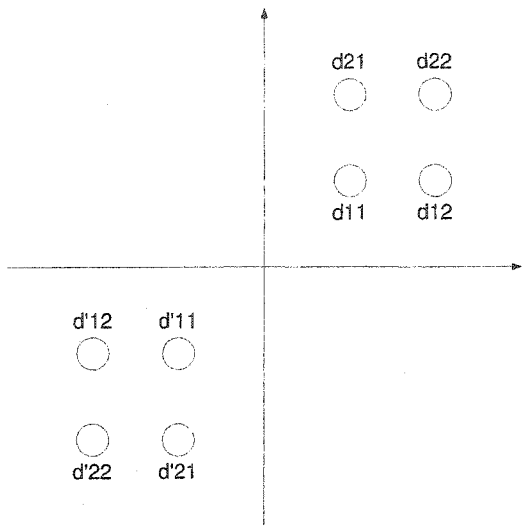
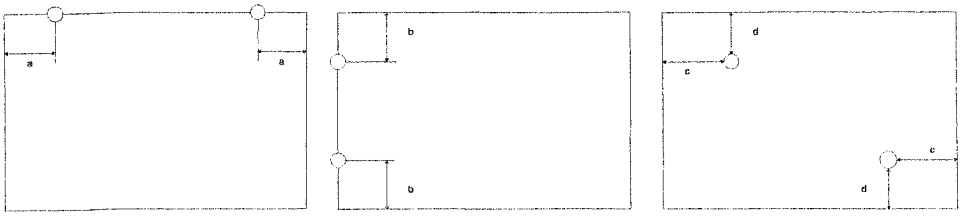



**Figure 4.6:** Symmetry of watermark embedding.

effect of embedding at one point in LPM domain. Refer to Section 4.3.3, the number of watermark points is doubled in order to maintain the frequency symmetry. Therefore, 1024 watermark points are used for embedding watermark in our experiments.

### 4.3.3 Symmetry of watermark embedding

We should maintain the frequency symmetry when we embed watermark in the Fourier frequency domain. Refer to Section 4.1 and Section 4.3.1, we embed at four points in DFT domain to achieve the effect of embedding at one point in LPM domain. Refer to Fig. 4.6, if we want to embed four points $d_{11}$, $d_{12}$, $d_{21}$, and $d_{22}$ in DFT domain, we need embed in the same strength at points $d'_{11}$, $d'_{12}$, $d'_{21}$, and $d'_{22}$. Here $d'_{11}$, $d'_{12}$, $d'_{21}$, and $d'_{22}$ are respectively symmetric about the origin with $d_{11}$, $d_{12}$, $d_{21}$, and $d_{22}$. Doubling the watermarking points will not necessarily reduce the image quality, because we can use weaker embedding strength and take advantage of the data redundancy. Keeping symmetry enhances the performance of watermark extraction on rotation transformations.




(a) Rotation (b) Scaling (c) Rotation and scaling

**Figure 4.7:** Phase correlation illustration.

### 4.3.4 Removal of low frequency components

From the watermark embedding process, we clearly know where the watermark data could be, so after the first DFT transform, we can discard the central part of the DFT amplitude spectrum. Such a procedure will not lose the watermark data and has advantages according to our experiments.

The amplitudes of the low frequency components are much larger than the high and middle frequency components. After the scaling and rotation operations, the changes in the low frequency components are overwhelming in the entire changes of the amplitude spectrum. And if there are transformations such as cropping and new boundary, which are usual when the scaling and rotation are applied to the image, such undesired changes in the low frequency components maybe even more obvious according to our observations. Removal of low frequency components helps to produce more accurate displacement calculation.

### 4.3.5 Displacement calculation and watermarking position rectification

Refer to Equ. (3.21) and Equ. (3.22), we calculate $C_{10}$ then $D_{10}$. The LPM of the watermarked image undergone rotational and/or scaling transformation and the LPM

of the original image are not simply translated (displaced) from each other, actually the former is a cyclically shifted version of the latter. Therefore, two peaks in the phase correlation spectrum are expected (refer to Fig. 4.7 and Fig. 5.1). The algorithm does not discriminate one peak from another, instead both peaks are declared as displacement. Fig. 4.7 illustrates the peaks in resulting phase correlation spectrum after the watermarked image has been undergone rotation, scaling, and rotation and scaling transformations. Theoretically the two peaks in Fig. 4.7 (a) are symmetric about the vertical line passing through the image center, the two peaks in Fig. 4.7 (b) are symmetric about the horizontal line passing through the image center, the two peaks in Fig. 4.7 (c) are symmetric about the image center. We can use this feature to rectify inaccuracy in the displacement calculation if any. It is very easy to detect the peaks if there is some displacement, since the peaks usually have very big value comparing the rest. If the watermarked image has not undergone rotation or scaling transformation, there is no displacement. In this situation, all the elements of array $D_{10}$ will be approaching zero, so that we cannot detect peaks in array $D_{10}$. However, when two images match the elements of array $C_{10}$ (cross-power spectrum) should be all approaching 1. By using this feature, we can judge that there is no displacement. We declare both the detected peaks and all their eight neighbors as displacement, to make our algorithms tolerate peak-detection inaccuracy. Using those eight neighbors as displacement could generate a slightly bigger normalized correlation for unwatermarked images, but the effect is neglectable. Experiments have demonstrated that the proposed method is very reliable.

### 4.3.6 Threshold selection

Since the normalized correlation is used as the detection measure, two approximate methods have been used to estimate the false positive probability. The threshold T can be set by first determining what false positive probability $P_{fpp}$ is required in practice. The first one is the approximate Gaussian method. Assuming the watermark sequence is zero mean, uncorrelated with the original image and n, the dimension of the watermark, is large enough; then under hypothesis that the watermark doesnt exist, the normalized correlations can be approximated as a Gaussian with standard deviation $1/\sqrt{n}$ and zero mean based on the central limit theorem.

$$P_{fpp} = \frac{1}{\sqrt{2\pi}\sigma_0}\int_T^{\infty} e^{\frac{-(x-m_0)^2}{2{\sigma_0}^2}} dx = Q(\frac{T-m_0}{\sigma_0}) \tag{4.9}$$

where mean $m_0 \approx 0$, standard deviation $\sigma_0 = 1/\sqrt{n}$.

While as mentioned in [43], as the threshold increases, we begin to dramatically overestimate the false positive probability.

We use the method proposed in [43] to give a more accurate estimation about the false positive probability.

$$P_{fpp} = \frac{I_{n-2}(T_a)}{2I_{n-2}(pi/2)} \tag{4.10}$$

$$I_d(\theta) = \int_0^{\theta} sin^d(u)du, T_a = cos^{-1}(T) \tag{4.11}$$

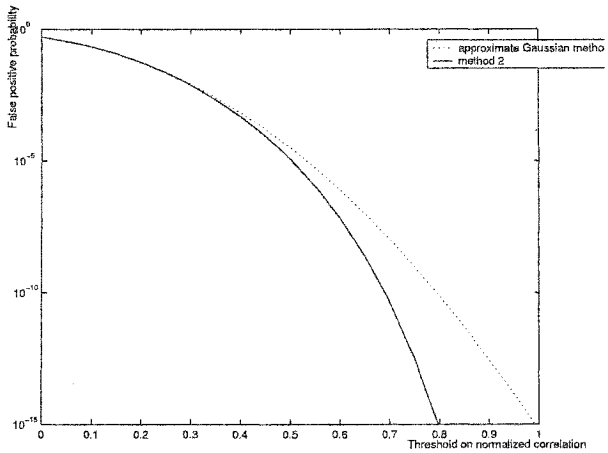



**Figure 4.8:** False positive probability for various threshold T, watermark length 64.

Based on the equation above, the relationship between T and $P_{fpp}$ is illustrated as following Fig. 4.8.

Based on the method 2, when the threshold is 0.4, 0.5, 0.6, The false positive probability is $4.8 \times 10^{-4}$, $1.1 \times 10^{-5}$, $6.4 \times 10^{-8}$ respectively. So we set the threshold to be a value between 0.5 and 0.6, by which the false positive probability smaller than $10^{-5}$ can be achieved to meet the requirement.

In order to select the optimum threshold T to minimize the total error probability $P_{error}$, we performed the stochastic analysis on the normalized correlation results using the approximate Gaussian method. The detection processes can be performed via the following statistical hypotheses:

$$H_0 : \ Watermark\ does\ not\ exist$$

$$H_1 : \ Watermark\ exists$$

We define $P_{error}$ as the total error probability.

$$P_{error} = P_{fpp}P(H_0) + P_{fnp}P(H_1) \tag{4.12}$$

$P_{fpp}$ is the false positive probability (detection of the watermark under $H_0$), and $P_{fnp}$ is the false negative probability(fail to detect the watermark under $H_1$), $P(H_0)$ and $P(H_1)$ are the a priori probability of $H_0$ and $H_1$. By assuming that $P(H_0)$ and $P(H_1)$ are equiprobable so that:

$$P_{error} = \frac{1}{2}(P_{fpp} + P_{fnp}) \tag{4.13}$$

Under hypothesis $H_0$, the normalized correlations can be approximated as $N(m_0, 1/n)$ [44]. The false positive probability (detecting watermark under $H_0$) is:

$$P_{fpp} = \frac{1}{\sqrt{2\pi}\sigma_0}\int_T^{\infty} e^{\frac{-(x-m_0)^2}{2{\sigma_0}^2}}dx = Q(\frac{T-m_0}{\sigma_0}) \tag{4.14}$$

where mean $m_0 \approx 0$, standard deviation $\sigma_0 = 1/\sqrt{n}$.

Under hypothesis $H_1$, the normalized correlations can be approximated as $N(m_1, 1/n)$. The false negative probability (fail to detect watermark under $H_1$) is:

$$P_{fnp} = \frac{1}{\sqrt{2\pi}\sigma_1}\int_{-\infty}^{T} e^{\frac{-(x-m_1)^2}{2{\sigma_1}^2}}dx = 1 - Q(\frac{T-m_1}{\sigma_1}) \tag{4.15}$$

where $m_1$ is mean, while standard deviation $\sigma_1 = 1/\sqrt{n}$.

In Equ. (4.14) and Equ. (4.15), $T$ is the threshold and $Q$ is defined as:

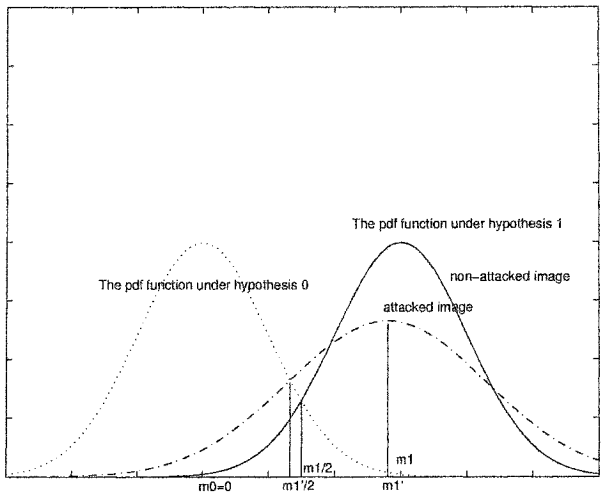



**Figure 4.9:** The pdf function of the correlation output with attacks considered.

$$Q(X) = \frac{1}{\sqrt{2\pi}} \int_X^{\infty} e^{\frac{-x^2}{2}} dx \tag{4.16}$$

Based on the MAP (maximum a posteriority probability) or the ML (maximum likelihood) criterion, the optimum threshold to minimize the total error probability should be equal to $(m_0 + m_1)/2$. Since $m_0 \approx 0$, threshold T is equal to $m_1/2$. When the image has been corrupted by attacks, both the mean value $m_1$ and the variance $\sigma_1^2$ may be altered. We can say that because of attacks, two Gaussian approximations are still valid, while the Gaussian approximation of correlation output under hypothesis 1 has now a larger variance and smaller mean as shown in Fig. 4.9. After attacks, the mean $m_1$ is smaller than 1 so that the threshold should be a little smaller than 0.5. Meanwhile, to achieve the requirement about the false positive probability, the threshold should be between 0.5 and 0.6. To achieve the balance, in the paper, we set the threshold to be 0.5, by which the accurate detection results have been achieved in

our experiments.

# Chapter 5

# Experimental results and evaluations

In this chapter, we illustrate and evaluate the performance of the proposed scheme against rotation, scaling, and translation transformations, JPEG compression and other attacks.

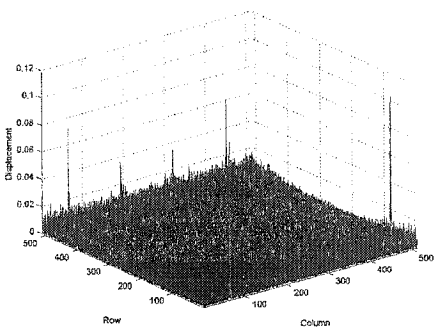
## 5.1 Phase correlation and displacement calculation

In this part, I will show how we can compute the displacement information based on the phase correlation.

### 5.1.1 Rotation with cropping

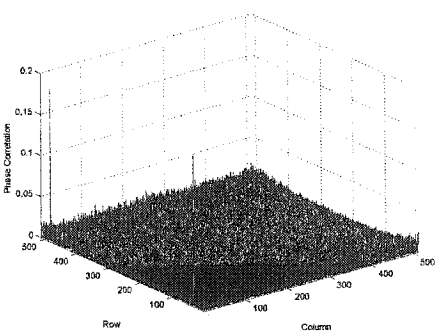
Fig. 5.1 (a) is transformed from the watermarked image (Fig. 5.2 (b)), by rotating the watermarked image by 45 ° counter-clockwise without scaling. Fig. 5.1 (b) is the amplitude of the corresponding phase correlation spectrum. Black pixels have been

(a) Rotated by 45 degree counter-clockwise without scaling. Four corners of the image have been cropped.

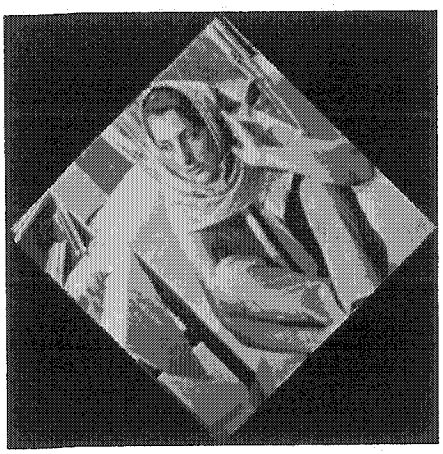



(b) Phase correlation between LPM of (a) and LPM of the original image. There are two peaks, one at $(1, 65)$ and another at $(1, 449)$.

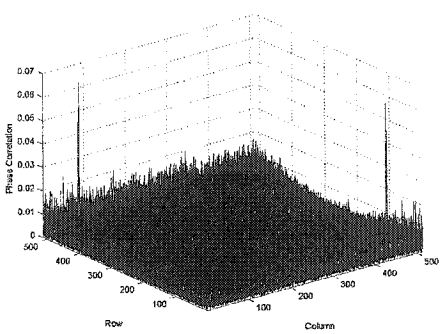
(c) Scaled by 0.7.




(d) Phase correlation between LPM of (c) and LPM of the original image. There are two peaks, one at $(32, 1)$ and another at $(482, 1)$.



(e) Rotated 45 degree counter-clockwise after being scaled by 0.7062.




(f) Phase correlation between LPM of (e) and LPM of the original image. There are two peaks, one at $(482, 65)$ and another at $(32, 449)$.

**Figure 5.1:** Phase correlation for watermarked images undergone different transformations.

padded to the outside of the three transformed images to maintain the shape and size of the entire resulting image.

For comparison purpose, the displacement calculated by using Equ. (4.4) and Equ. (4.5) is $(1, 65)$, where 1 and 65 are respectively the row number and column number of the displacement. Note the indexes start from 1. $(1, 65)$ here means that the log-polar radius $\rho$ in LPM of the watermarked image has not been changed, while the angle $\theta$ in LPM of the watermarked image has been cyclically shifted 65 columns right (rotated 45 ° counter-clockwise).

We applied DFT and LPM to both the original image (Fig. 5.2) and the rotated watermarked image (Fig. 5.1 (a)), then obtained the phase correlation spectrum between the LPM of the original and that of the watermarked, shown as Fig. 5.1 (b). There are two peaks in the phase correlation spectrum. The phase correlation of the first peak at $(1, 65)$ is $+0.0000 + 0.1166i$, and the phase correlation of the second peak at $(1, 449)$ is $+0.0000 - 0.1166i$. We use these two displacements to rectify the watermarking positions.

After the rectification, we calculate and get the correlation coefficient of 0.7078, which is the same as the result of exhaustive search.

### 5.1.2 Scaling without rotation

Fig. 5.1 (c) is transformed from the watermarked image (Fig. 5.2 (b)), by scaling the watermarked image by 0.7. Fig. 5.1 (d) is the amplitude of the corresponding phase correlation spectrum.

For comparison purpose, the displacement calculated by using Equ. (4.4) and Equ. (4.5) is $(482, 1)$, which means that the log-polar radius $\rho$ in LPM of the watermarked image has been cyclically shifted 482 columns down (scaled by 0.7), while the angle $\theta$

in LPM of the watermarked image has not been changed.

We applied DFT and LPM to both the original image (Fig. 5.2) and the rotated watermarked image (Fig. 5.1 (c)), then obtained the phase correlation spectrum between the LPM of the original and that of the watermarked, shown as Fig. 5.1 (d). There are two peaks in the phase correlation spectrum. The phase correlation of the first peak at $(32, 1)$ is $-0.0000 - 0.1863i$, and the phase correlation of the second peak at $(482, 1)$ is $-0.0000 + 0.1863i$. We use these two displacements to rectify the watermarking positions.

After the rectification, we calculate and get the correlation coefficient of 0.8335, which is the same as the result of exhaustive search.

### 5.1.3 Scaling and rotation

Fig. 5.1 (e) is transformed from the watermarked image (Fig. 5.2 (b)), by rotating the watermarked image by 45 ° counter-clockwise after scaling by 0.7062. Fig. 5.1 (f) is the amplitude of the corresponding phase correlation spectrum.

For comparison purpose, the displacement calculated by using Equ. (4.4) and Equ. (4.5) is $(482, 65)$, which means that the log-polar radius $\rho$ in LPM of the watermarked image has been cyclically shifted 482 columns down (scaled by 0.7), while the angle $\theta$ in LPM of the watermarked image has been cyclically shifted 65 columns right (rotated 45 ° counter-clockwise).

We applied DFT and LPM to both the original image (Fig. 5.2) and the rotated watermarked image (Fig. 5.1 (e)), then obtained the phase correlation spectrum between the LPM of the original and that of the watermarked, shown as Fig. 5.1 (f). There are two peaks in the phase correlation spectrum. The phase correlation of the first peak at $(482, 65)$ is $-0.0000 - 0.0648i$, and the phase correlation of the second

peak at $(32, 449)$ is $-0.0000 + 0.0648i$. We use these two displacements to rectify the watermarking positions.

After the rectification, we calculate and get the correlation coefficient of 0.8119, which is the same as the result of exhaustive search.

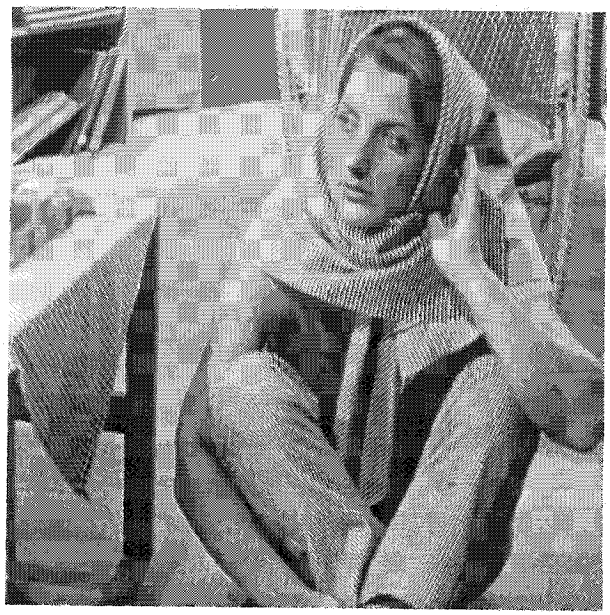
## 5.2 Experimental results

In this section, I will present the experimental results while I test the performance of the proposed watermarking scheme against various attacks. And I will give a comparison between the results of the phase correlation extraction method and the results of the exhaustive search extraction method.

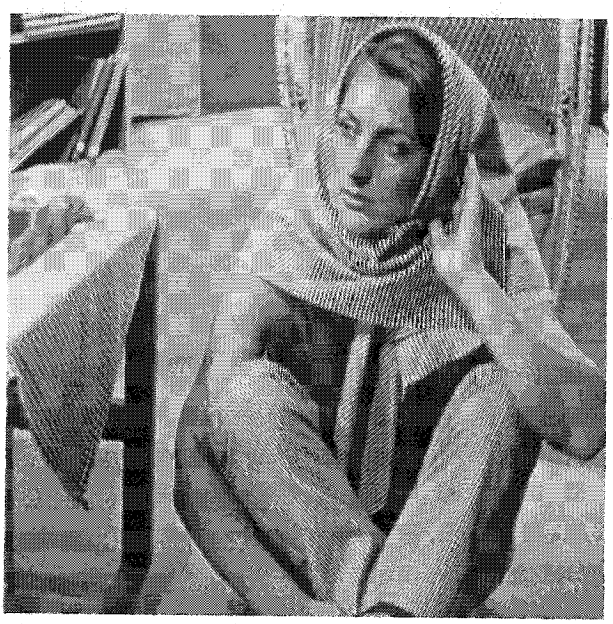
### 5.2.1 The original image and the watermarked image

We use image *Barbara* as the original test image, shown as Fig. 5.2 (a). The watermarked image is shown as Fig. 5.2 (b), which is obtained by embedding using such a strength that the watermark is just imperceptible. The PSNR of Fig. 5.2 (b) is $44.2070dB$. We experimented with various $\alpha$ values, and we found that $\alpha$ values in the range of 150 to 200 give the best objective and subjective qualities to the test images. In our experiments, we use $\alpha = 150$. All the following experiments will be conducted on this watermarked image.

Because we embed the watermark data into the middle frequency region, it will cause some distortion in the regions which contain a lot of middle frequency components. If we apply "zooming in" operation to the area within the black square in Fig. 5.3 for both the original image and the watermarked image, we can notice some distortion through the comparison between Fig. 5.4 (a) and Fig. 5.4 (b). However such distortion



(a) Barbara: the original test image



(b) The watermarked image, $PSNR = 44.2070dB$. All the following tests were based on this image.

**Figure 5.2:** The original image and watermarked image.

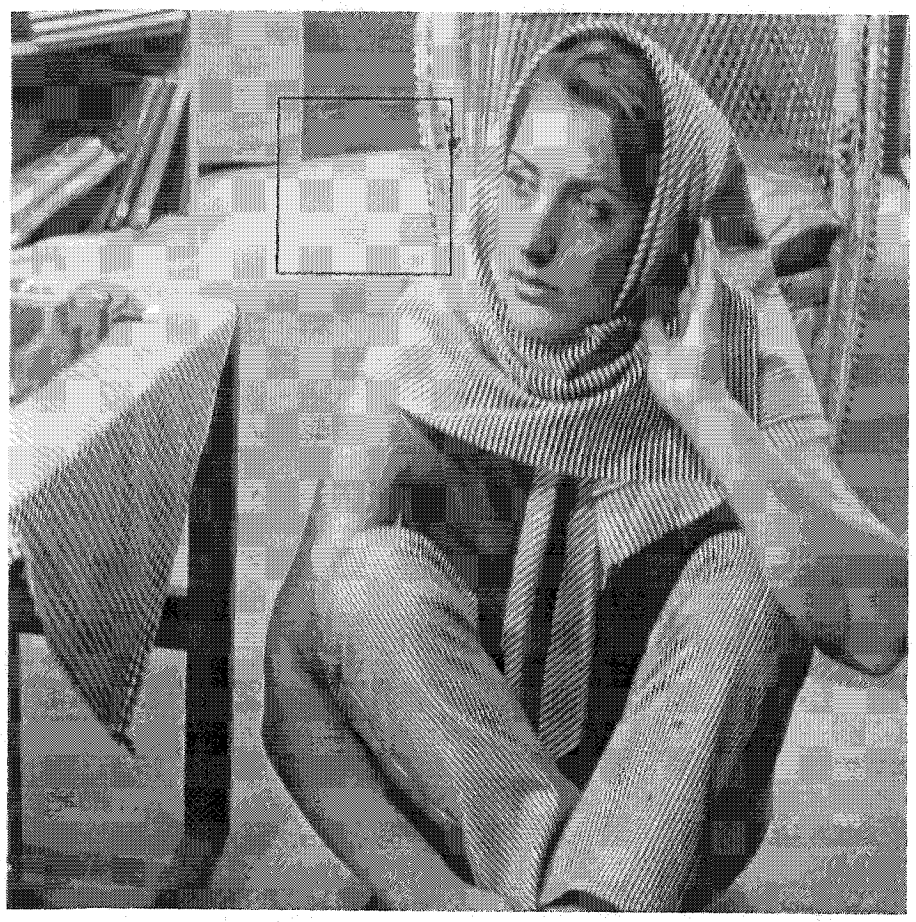
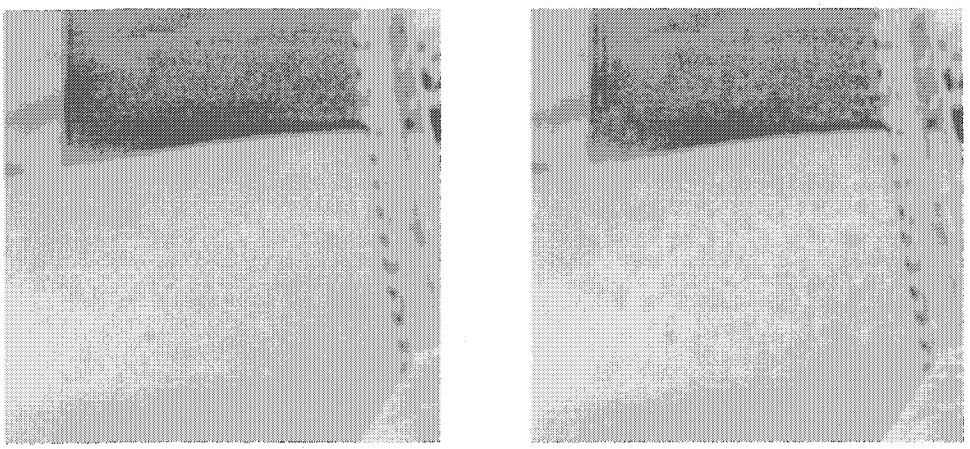


**Figure 5.3:** The original Barbara image.

is not noticable without the reference of the orginal image and applying the zooming in operation. Through both the objective and subjective measurement of the watermarked image quality, the fidelity of the image can be preserved after the watermark embedding process. The goal of the invisiblity is achieved.

We also have done experiments on other test images, and the similar results have been achieved. Test results will show that the scheme can meet the requirements of both imperceptibility and robustness.



(a) The zoomed-in (5x) area of the original image. (b) The zoomed-in (5x) area of the watermarked image.

**Figure 5.4:** The comparison for distortion checking.

### 5.2.2 Rotation with cropping

Refer to Fig. 5.1 (a) for the illustration of the experiment. We rotated the watermarked image (Fig. 5.2 (b)) counterclockwise by different angles listed in Table 5.1. The four corners of the watermarked image have been cropped off due to the rotation, and black pixels have been padded in order to maintain the image size and shape.

In all the tables of this section, $Correlation_1$ is the normalized correlation coefficient between the original watermark sequence and the watermark sequence detected from the watermarked image. Since we can find the correct watermark data with either the phase correlation extraction method or the exhaustive search extraction method if the watermark is embedded in the test images, $Correlation_1$ is the same for these two extraction methods.

$Correlation_2$ is the normalized correlation coefficient between the original watermark sequence and 'the watermark sequence' detected from the corresponding unwatermarked image that undergone the same transformations as the watermarked image

when the phase correlation method is used to detect the watermark. $Correlation_3$ is the normalized correlation coefficient when the exhaustive search method is used. We can find that the $Correlation_3$ are larger than $Correlation_2$. This difference is due to the exhaustive search which always choose the maximum correlation value after the searching procedure. While using the phase correlation, we can rectify the shifted watermark to the original position. So we can avoid the possible deviation introduced by the searching procedure.

In Table 5.1, the *Angle* is the angle we rotated the watermarked image before extracting watermark, We only listed the results for rotation angles up to 180.5 °, since

$$Correlation(180\,^\circ + \alpha) = Correlation(\alpha). \tag{5.1}$$

From the table, we can see that the correlation coefficients for the watermarked image are all greater than 0.6451, and the correlation coefficients $Correlation_2$ for the unwatermarked image are all smaller than 0.3145. The correlation coefficients $Correlation_3$ are smaller than 0.4. The algorithm can handle fractional degree rotation. When rotational degree becomes bigger, larger areas need to be cropped and more black pixels need to be padded, therefore $Correlation_1$ tends to be smaller. The $Correlation_2$ values are random noise under about 0.3 and the $Correlation_3$ values are random noise under about 0.4. We can see from Table 5.1, that the majority of the $Correlation_1$ values are bigger than 0.7, the majority of the $Correlation_2$ values are smaller than 0.3 and the $Correlation_3$ values are all smaller than 0.4. In section 4.3.6, we set the threshold to 0.5. So it is clear that the threshold 0.5 can give the judgement about whether the watermark exists or not without any mistake.

We have done the similar tests to other standard test images, and we obtained the

similar results. Therefore we can conclude that the scheme is robust to rotation.

**Table 5.1:** Experimental results, rotation with cropping

| *Angle* | $Correlation_1$ | $Correlation_2$ | $Correlation_3$ |
|---|---|---|---|
| 0° | 0.9860 | 0.1952 | 0.3516 |
| 1° | 0.8613 | 0.2360 | 0.3402 |
| 1.5° | 0.8251 | 0.2928 | 0.3727 |
| 5° | 0.9288 | 0.2223 | 0.3894 |
| 5.5° | 0.8269 | 0.2782 | 0.3660 |
| 10° | 0.8508 | 0.2569 | 0.3385 |
| 10.5° | 0.8476 | 0.2624 | 0.3872 |
| 40° | 0.7627 | 0.2223 | 0.3835 |
| 50° | 0.7555 | 0.1715 | 0.3355 |
| 60° | 0.6451 | 0.1260 | 0.2999 |
| 90° | 0.8626 | 0.2919 | 0.3922 |
| 120° | 0.7416 | 0.2980 | 0.3772 |
| 130° | 0.7027 | 0.2932 | 0.3556 |
| 150° | 0.7349 | 0.1914 | 0.3472 |
| 160° | 0.8724 | 0.2386 | 0.3672 |
| 180° | 0.9860 | 0.1952 | 0.3516 |
| 180.5° | 0.8219 | 0.1976 | 0.3057 |

### 5.2.3 Scaling without rotation

Refer to Fig. 5.1 (c) for illustration. We scaled the watermarked image (Fig. 5.2 (b)) by using the scales listed in the column *Scale* of the Table 5.2. From the table, we can see that all the $Correlation_1$ are greater than 0.7834, all the $Correlation_2$ are less than 0.2882 and all the $Correlation_3$ are less than 0.39. And the predefined threshold 0.5 can give the correct judgement. Therefore we can conclude that the scheme is robust to scaling when the scale factor is in a reasonable range.

### 5.2.4 Scaling and rotation

There was no cropping in this test, because the image was shrunk accordingly before being rotated. Refer to Fig. 5.1 (e) for illustration, black pixels have been padded in order to maintain the image size and shape. The test results are shown in Table

**Table 5.2:** Experimental results, scaling without rotation

| *Scale* | $Correlation_1$ | $Correlation_2$ | $Correlation_3$ |
|---|---|---|---|
| 0.6 | 0.7834 | 0.1596 | 0.3821 |
| 0.7 | 0.8335 | 0.1779 | 0.3436 |
| 0.8 | 0.8469 | 0.2752 | 0.3246 |
| 0.9 | 0.8839 | 0.1913 | 0.3479 |
| 1.0 | 0.9860 | 0.1952 | 0.3516 |
| 1.1 | 0.9013 | 0.2873 | 0.3836 |
| 1.2 | 0.8401 | 0.1704 | 0.3764 |
| 1.3 | 0.8511 | 0.2882 | 0.3882 |

**Table 5.3:** Experimental results, scaling and rotation

| *Angle/Scale* | $Correlation_1$ | $Correlation_2$ | $Correlation_3$ |
|---|---|---|---|
| 0°/1.0000 | 0.9860 | 0.1952 | 0.3516 |
| 1°/0.9827 | 0.7956 | 0.2701 | 0.3522 |
| 1.5°/0.9679 | 0.8054 | 0.1735 | 0.3162 |
| 5°/0.9192 | 0.8332 | 0.3258 | 0.3588 |
| 5.5°/0.9127 | 0.8268 | 0.2758 | 0.3174 |
| 10°/0.8634 | 0.8541 | 0.2478 | 0.3647 |
| 10.5°/0.8576 | 0.8185 | 0.2730 | 0.3328 |
| 30°/0.7304 | 0.7966 | 0.2790 | 0.3621 |
| 40°/0.7107 | 0.7120 | 0.1750 | 0.3019 |
| 50°/0.7107 | 0.7625 | 0.2574 | 0.3460 |
| 60°/0.7304 | 0.7593 | 0.2363 | 0.3756 |
| 90°/1.0000 | 0.8626 | 0.3031 | 0.3922 |
| 120°/0.7304 | 0.7331 | 0.1725 | 0.3680 |
| 130°/0.7107 | 0.7200 | 0.2357 | 0.3786 |
| 150°/0.7304 | 0.7499 | 0.2224 | 0.3398 |
| 160°/0.7793 | 0.7724 | 0.3099 | 0.3213 |
| 180°/1.0000 | 0.9860 | 0.1952 | 0.3516 |
| 180.5°/0.9903 | 0.9148 | 0.2908 | 0.3827 |

5.3, where the *Angle* is the angle we rotated counterclockwise the watermarked image before extracting watermark, *Scale* is the scaling factor used for scaling before rotation in order to keep the whole image inside the original frame.

From the table, we can see that the correlation coefficients $Correlation_1$ for the watermarked image are all greater than 0.7120, the correlation coefficients $Correlation_2$ for the unwatermarked image are all smaller than 0.3258 and the correlation coefficients $Correlation_3$ are all smaller than 0.4. Same as before, the threshold 0.5 works quite well in this experiment. We have done the similar tests to other standard test images,

and we obtained the similar results. Therefore we can conclude that the scheme is robust to rotation and scaling.

### 5.2.5 JPEG compression

We compressed the watermarked image *Barbara* (Fig. 5.2 (b)) by different quality factors, the test results are shown in Table 5.4. From the table, we can see that the results are very good for both extraction methods until the quality factor equal to 5%. We think that the robustness to JPEG compression is due to our embedding watermarks in mid-range frequencies.

**Table 5.4:** Experimental results, JPEG compression

| *Quality* | $Correlation_1$ | $Correlation_2$ | $Correlation_3$ |
|---|---|---|---|
| No compression | 0.9860 | 0.1952 | 0.3318 |
| 90 % | 0.9582 | 0.2216 | 0.3479 |
| 80 % | 0.9823 | 0.2102 | 0.3340 |
| 70 % | 0.9804 | 0.2580 | 0.3552 |
| 60 % | 0.9766 | 0.3218 | 0.3294 |
| 50 % | 0.9681 | 0.2023 | 0.3325 |
| 40 % | 0.9605 | 0.2542 | 0.3534 |
| 30 % | 0.9394 | 0.2586 | 0.3957 |
| 20 % | 0.8579 | 0.1725 | 0.3396 |
| 10 % | 0.7561 | 0.2004 | 0.4131 |
| 5 % | 0.3189 | 0.2360 | 0.3451 |

### 5.2.6 Performance on different images

We applied our phase correlation scheme to 100 test images downloaded from the Internet. The image set was chosen to contain a variety of natural images. The images were compressed by JPEG using quality factor equal to 50%, scaled by 0.7793, and then rotated counterclockwise by 20°. The test results are shown in Fig 5.5. The horizontal axis shows different images, while the vertical axis shows the corresponding correlation. The upper curve represents the correlation between the original watermark

sequence and the watermark sequence detected from the watermarked image, and the lower curve represents the correlation between the original watermark sequence and 'the watermark sequence' detected from the corresponding unwatermarked image. The scheme could still detect correctly from all the watermarked images without false positive from unwatermarked images, under the quite severe condition. Experiments show that the exhaustive search extraction method can achieve the similar results. In both cases, two extraction methods can tell whether the watermark exists or not correctly for images containing various characteristics.

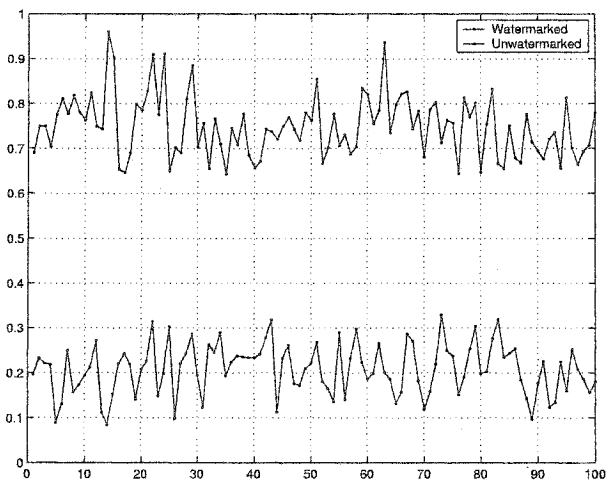



**Figure 5.5:** Watermark detection results for 100 test images.

We have tested our scheme against different transformations and JPEG compression, and the excellent performance of the scheme has been demonstrated.

### 5.2.7 Random watermark test

We have conducted random-watermark false positive tests in order to demonstrate the ability of our watermark extraction algorithm. Here we present the experiment of the phase correlation extraction method. As shown in Fig. 5.6, where the x-axis represents

1000 randomly generated PN sequences and the y-axis shows the resulting detection values. The test was on Fig. 5.2 (a), and the embedding strength is the same as the one used to generate Fig. 5.2 (b). Here, the large difference between the value obtained from the PN sequence originally embedded (shown at location 20 on the x-axis) and the other PN sequences is presented as a demonstration of the ability of the watermarking system to distinguish different PN sequences. The similar result is achieved for the exhaustive search extraction method.

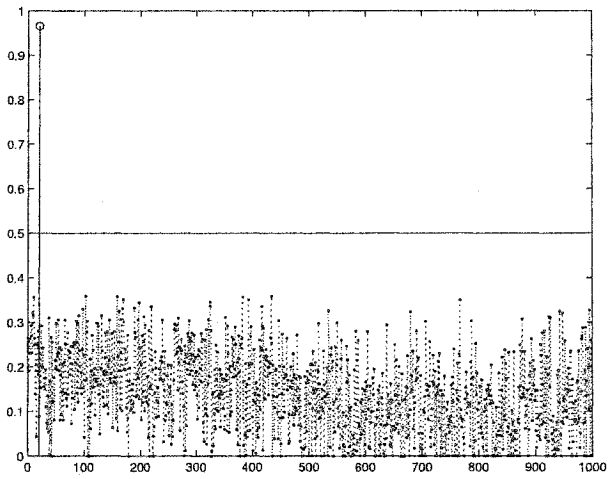



**Figure 5.6:** Watermark detection results for 1000 PN sequence including the one originally embedded.

### 5.2.8 Performance on different watermarks

We have test our watermarking system on different watermarks in order to demonstrate the good results are not random, as shown in Fig. 5.7, where the x-axis represents 1000 randomly generated and embedded PN sequences and the y-axis shows the resulting detection values. The test was on Fig. 5.2 (a), and the embedding strength is the same as the one used to generate Fig. 5.2 (b). Here, all the detection values are bigger

than 0.9. So the watermarking system performs well on any PN sequence. For both extraction methods, we get the same results.

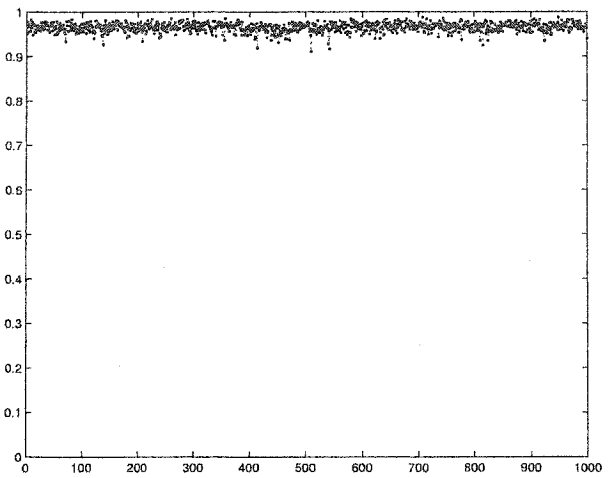



**Figure 5.7:** Watermark detection results for 1000 watermarks.

### 5.2.9 Miscellaneous attacks

The watermark scheme was tested with many kinds of attacks including noise pollution, noise removing operation, filter operation, and pixel removing attacks. StirMark was used for generating the attacks. The experiments show the robustness of the proposed watermarking scheme. Both extraction schemes work well in all the following tests. Due to the embedding mechanism, the robustness of the embedded watermark is not compromised by the ordinary image processing techniques presented in Table 5.5. For pixel removal in the table, we randomly removed 10 columns or 10 rows for all 5 test cases.

**Table 5.5:** Experimental results, miscellaneous attacks

| *Attack* | $Correlation_1$ | $Correlation_2$ | $Correlation_3$ |
|---|---|---|---|
| Gaussian white noise: N(0, 0.001) | 0.9399 | 0.2177 | 0.3145 |
| Gaussian white noise: N(0, 0.005) | 0.8748 | 0.2513 | 0.3798 |
| Gaussian white noise: N(0, 0.01) | 0.6760 | 0.1512 | 0.3538 |
| Gaussian filter: filter size [7 7], standard deviation 0.5 | 0.9659 | 0.1963 | 0.3417 |
| Gaussian filter: filter size [7 7], standard deviation 1 | 0.9507 | 0.1967 | 0.3356 |
| Gaussian filter: filter size [3 3], standard deviation 1 | 0.9605 | 0.1786 | 0.3522 |
| 1. Gaussian noise pollution: N(0, 0.001) 2. Wiener Filter to remove noise | 0.9101 | 0.2513 | 0.3657 |
| 1. Salt & pepper pollution: noise density 0.001 2. Median filter to remove noise | 0.8887 | 0.2352 | 0.3317 |
| 1. Salt & pepper pollution: noise density 0.01 2. Median filter to remove noise | 0.8846 | 0.2292 | 0.3722 |
| 1. Salt & pepper pollution: noise density 0.1 2. Median filter to remove noise | 0.8005 | 0.3347 | 0.3507 |
| Pixel removal: Case1 | 0.9580 | 0.2733 | 0.3823 |
| Pixel removal: Case2 | 0.9540 | 0.2471 | 0.3475 |
| Pixel removal: Case3 | 0.9647 | 0.2129 | 0.3535 |
| Pixel removal: Case4 | 0.9676 | 0.2015 | 0.3666 |
| Pixel removal: Case5 | 0.9661 | 0.2592 | 0.3878 |

## 5.3 Summary

In this chapter, we first present the experimental results to show the displacement information can be computed using the phase correlation method. So in the following tests, the possible shifted watermark data can rectified to its original position and the watermark data can be retrieved to do the cross correlation computation without the need of exhaustive search.

Then we test our watermarking scheme with all kinds of attacks. Our watermarking scheme shows the robustness to all the possible RST (rotation, scaling and translation) attacks. Because of using of the spread spectrum technique and the perceptual model, the watermarking scheme shows that it is robust against JPEG, noise and filtering operations and can preserve the fidelity of the image after the embedding process.

# Chapter 6

# Conclusions and future works

Fourier-Mellin transform is an excellent theory for RST invariant watermarking scheme, however it is difficult to implement due to the impracticality of implementing LPM and ILPM. In this paper, we proposed a LPM and phase correlation based digital watermarking scheme that is invariant to RST transformations. We calculate the displacement by detecting peaks in phase correlation spectrum and then rectify the shifts of angle $\theta$ and log-radius $\rho$ in LPM magnitude spectrum, and consequently to make the scheme invariant to RST transformations. Also the exhaustive search extraction is proposed as a method of extracting the watermark data while the original image is not available.

Spread spectrum technique and perceptual model are used to enhance the security of this algorithm and achieve the optimum balance between invisibility and robustness. Based on the properties of the cross correlation, we perform the stochastic analysis to determine the optimum threshold for the judgement of the existence of the watermark.

The test results demonstrated that the scheme is very reliable in displacement calculation and invariant to rotation and translation, invariant to scaling when the scale

is in a reasonable range, and very robust to JPEG compression. In all the experiments, the pre-defined threshold works well to give a correct judgement about the existence of the watermark. The main contributions of the work include the idea of using phase correlation spectrum in digital image watermarking, and the simple and feasible implementation of RST invariant watermarking scheme in LPM domain.

As for our future work, we will improve our watermark extraction algorithm to better deal with cropping and black pixel padding caused by rotation and scaling. More advanced perceptual model will be introduced to improve the performance of the watermarking algorithm. For example, the stochastic model based NVF (Noise Visibility Function) can be used to analysis the image properties, determining the regions suitable for watermark embedding. It finally can lead to a perceptual model that can be used to determine the optimal locations and strength for watermark embedding more accurately.

# Bibliography

[1] http://www.cl.cam.ac.uk/ fapp2/watermarking/stirmark/

[2] http://www.checkmark.com/

[3] http://www.certimark.org/

[4] J. O'Ruanaidh and T. Pun, Rotation, scale, and translation invariant digital image watermarking, Signal Processing, Vol.66, No.3, pp. 303-317, 1998.

[5] D. Zheng and J. Zhao, LPM-Based RST invariant digital image watermarking, The IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2003, 2003.

[6] D. Zheng and J. Zhao, RST invariant digital image watermarking: Importance of phase information, The IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2003, 2003.

[7] D. Zheng, J. Zhao and A.E. Saddik, RST invariant digital image watermarking based on log-polar mapping and phase correlation, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on

Authentication, Copyright Protection and Information Hiding, To appear in September 2003 issue.

[8] D. Zheng and J. Zhao, A Novel RST Invariant Digital Image Watermarking Scheme, Multispectral image processing and pattern recognition (MIPPR) conference 2003, Beijing, China, 2003.

[9] D. Zheng and J. Zhao, Apply phase information in RST image watermarking, IEEE International Conference on Consumer Electronics (ICCE2003), Los Angeles, USA, 2003.

[10] A. Tirkel, G.Rankin, R. Van Schyndel, W. Ho, N. Mee And C. Osborne, Electronic watermark, Proc. DICTA, pp. 666–672, 1993.

[11] C. Kurah and J.McHughes, A cautionary note on image downgrading, Proc. of IEEE Computer Security Applications Conference, Vol. 2, pp. 153–159, 1992.

[12] O. Bruyndonckx, J. J. Quisquater and B. Macq, Spatial method for copyright labeling of digital images, Proc. IEEE workshop Nonlinear Signal and Image Processing, pp. 456–459, 1995.

[13] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Transactions on Image Processing, Vol. 6, No. 12, pp. 1673–1687, 1997.

[14] R.L. Pickholtz, D.L. Schilling, and L.B. Milstein, Theory of spread spectrum communications - a tutorial, IEEE Transactions on Communications, Vol. 30, No. 5, pp. 855–884, May 1982.

[15] E. Koch and J. Zhao, Towards robust and hidden image copyright labeling Proc. IEEE Workshop on Nonlinear Signal and Image Processing, pp. 452–455, 1995.

[16] A. Bors and I.Pitas, Image watermarking using DCT domian constraints, Proc. Int. Conf. Image Processing, pp. 231–234, 1996.

[17] J. O'Ruanaidh, W.J. Dowling and F.M. Boland, Phase watermarking of digital images, Proc. of IEEE Int. Conf. on Image Processing, pp. 239–242, 1996.

[18] J. O'Ruanaidh, W.J. Dowling and F.M. Boland, Watermarking digital images for copyright protection, IEEE Proc. on Vision, Image and Signal Processing, pp. 250–256, 1995.

[19] D. Kundur and D. Hatzinakos, A robust digital image watermarking method using wavelet based fusion, Proc. of the Int. Conf. on Image Processing, Vol. 1, pp. 544–547, 1997.

[20] D. Kundur and D. Hatzinakos, Digital watermarking for telltale tamper proofing and authentication, Proc. of the IEEE, Vol. 87, No. 7, pp. 1167–1179, 1999.

[21] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, Secure spread spectrum watermarking for images, audio and video, Proc. of the IEEE Int. Conf. on Image Processing ICIP-96, pp. 243–246, 1996.

[22] M. Kutter, Watermarking resistant to translation, rotation, and scaling, Proc. of SPIE, Vol. 3628, pp. 423-431, 1998.

[23] M. Kankanhalli, R. Ramakrishman, Content based watermarking of images, ACM Multimedia, 1998.

[24] F. Bartolini, M. Barni, V. Cappellini, A. Piva, Mask building for perceptually hiding frequency embedded watermarks, Proc. of 5th IEEE International Conference on Image Processing ICIP'98, Vol. I, pp. 450-454, Chicago, Illinois, USA, October 4-7, 1998.

[25] S. Pereira and T. Pun, Robust template matching for affine resistant image watermarks, IEEE Transactions on Image Processing, Vol.9, No.6, pp. 1123-1129, 2000.

[26] M. Alghoniemy and A. Tewfik, Progressive quantized projection watermarking scheme, Proc. of ACM Multimedia 99, Vol.1, pp. 295-298, 1999.

[27] C. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y Lui, Rotation, scale, and translation resilient watermarking for images, IEEE Transactions on Image Processing, Vol.10, No.5, pp. 767-782, 2001.

[28] P. Bas, N. V. Boulgouris, F. D. Koravos, J.-M. Chassery, M. G. Strintzis, and B. Macq, Robust watermarking of video objects for MPEG-4 Applications, Proc. SPIE, Applications of Digital Image Processing XXIV, Vol. 4472, pp. 85–94, December 2001.

[29] P. Toft, The Radon Transform: theory and implementation, Ph.D. thesis, Technical University of Denmark, 1996.

[30] D. Simitopoulos, D. Koutsonanos, and M. G. Strintzis, Image watermark-

ing resistant to geometric attacks using Generalized Radon Transformations, Proc. DSP 2002, Vol. 1, pp. 85–88, July 2002.

[31] Srdjan Stankovic, Igor Djurovic, and Ioannis Pitas, Watermarking in the space/spatial-frequency domain using two-dimensional RadonCWigner Distribution, IEEE Transactions on Image Processing, Vol. 10, No. 4, April 2001.

[32] Hyung Shin Kim, Yunju Baek, and Heung Kyu, Lee Rotation, scale and translation invariant image watermarking using higher order spectra, Optical Engineering, Vol. 42, No. 2, February 2003.

[33] J. Behar, M. Porat, and Y. Zeevi, Image reconstruction from localized phase, IEEE Transactions on Signal Processing, Vol. 40, No.4, pp. 736-743, 1992.

[34] S. Urieli, M. Porat, and N. Cohen, Image characteristics and representation by phase: from symmetric to geometric structure, Proc. of International Conference on Image Processing 1996, Vol. 1, pp. 705-708.

[35] A. Oppenheim and J. Lim, Importance of phase in signals, Proc. of IEEE, Vol. 69, No.5, pp. 529-541, 1981.

[36] L. G. Brwon, A survey of image registration techniques, ACM Computing Surveys, Vol. 24, No. 4 pp. 325-376, 1992.

[37] B. S. Reddy and B. N. Chatterji, A FFT-based technique for translation, rotation, scale-invariant image registration, IEEE Transactions on Image Processing, Vol. 5, No. 8 pp. 1266-1271, 1996.

[38] R. Gonzalez and R. Woods, Digital image processing, Prentice-Hall, 2002, ISBN: 0-201-18075-8.

[39] R. Bracewell, The Fourier Transform and its applications, McGraw-Hill, 2000, ISBN: 0-07-303938-1.

[40] D. Young, Straight lines and circles in the log-polar image, BMVC2000: Proceedings of the 11th British Machine Vision Conference, pp. 426-435, 2000.

[41] C. Podilchuk and E. Delp, Digital watermarking: algorithms and applications, IEEE Signal Processing Magazine, pp. 33-46, 2001.

[42] I. Cox, M. Miller, and J. Bloom, Digital watermarking, Morgan Kaufmann Publishers, 2002, ISBN: 1-55860-714-5.

[43] M. Miller and J. Bloom, Computing the probability of false watermark detection, Proc. of the Third International Workshop on Information Hiding, Dresden, Germany, 1999.

[44] W. Zeng and B. Liu, A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images, IEEE Transactions on Image Processing, Vol. 8, No.11, pp. 1534-1548, 1999.