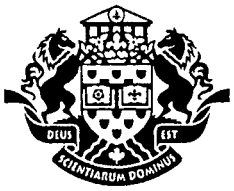
NOTE TO USERS



Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Yan, LIU

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M. A. Sc. (Electrical Engineering)

GRADE - DEGREE

Faculty og Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Rotation, scaling and translation invariant digital image watermarking

J. Zhao

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

M. Bouchard

R. Dansereau

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

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Rotation, scaling and translation invariant digital image watermarking

by

Yan Liu

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 2004

Your file Votre référence
ISBN: 0-494-01534-9
Our file Notre référence
ISBN: 0-494-01534-9

Abstract

Many digital image watermarking schemes have been proposed and are robust to common signal processing, such as compression and filtering. Geometric distortions, such as rotation, scaling, and translation (RST), are considered the most challenging attacks. This thesis is on RST invariant digital image watermarking.

In this thesis, we introduce the fundamental theories and techniques necessary for RST invariant digital image watermarking, and briefly review the existing RST invariant image watermarking schemes. Then, we propose an image rectification scheme for RST invariant digital image watermarking. Rotation and scaling transformations in the spatial domain result in cyclical shifts in the log-polar domain, which is the log-polar mapping (LPM) of the magnitude of the Fourier spectrum of the image. We utilize this property to detect the rotation and scaling parameters by using a matching template and our new phase-only filtering method in the LPM domain. We employ the same strategy in the spatial domain to detect the translation parameters. This rectification scheme can detect RST parameters very accurately.

Three criteria – signal-to-noise ratio (SNR), peak-to-correlation energy (PCE), and Horner efficiency – are used to compare the performance of different filters. It is clear that only the proposed phase-only filtering method gives satisfactory matching result in the LPM domain. We introduce three applications of this rectification scheme, in which a watermark is embedded in the spatial domain, the frequency domain and the log-polar domain, respectively. The experimental results demonstrate that our rectification scheme can be used for any image watermarking algorithms to provide robustness against rotation, scaling, and translation.

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-Mellin Transform
DWT	Discrete Wavelet Transform
LSB	Least Significant Bit
PN	Pseudo Noise
PSNR	Peak Signal to Noise Ratio
SNR	Signal to Noise Ratio
PCE	Peak to Correlation Energy
POF	Phase-Only Filter
BPOF	Binary Phase-Only Filter
HAS	Human Auditory System

This thesis is dedicated to Rongchun and Xiaolin.

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

Perfect copies can be made when image, video, audio, or other works are available in digital form. The Internet is an excellent distribution system for people to download pictures, music, and videos. However, on the other hand, the rightful ownership is at risk. The copyright protection for digital media suddenly becomes important.

The conventional technology to protect the owners' right is cryptography. The encrypted contents are delivered and a decryption key is provided to those who have purchased legitimate copies of the content. However, after the contents are decrypted, it is still hard to trace illegal reproductions. Digital watermarking is an alternative to cryptography. It can be designed to survive any decryption process.

Watermarking is derived from steganography. While cryptography is about protecting the content of a message, steganography is about concealing their existence [1]. Compared with steganography, digital watermarking needs to be robust against possible attacks.

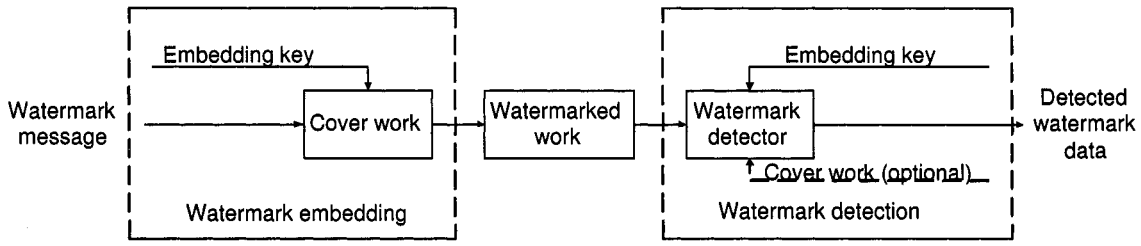


Figure 1.1: A typical watermarking system.

A typical watermarking system is shown in Figure 1.1. The watermark message could be a name, a number, text, an image, etc. It is embedded into the cover work by using the embedding key to get the watermarked work. This is the watermark embedding process. To detect the watermark data with the embedding key from the watermarked work is the watermark detection process. If the cover work is present in the detection process, it is an informed detection. Otherwise, it is a blind detection.

1.2 Applications of digital watermarking

The requirements that watermarking systems have to comply with are always based on applications. There is no "universal" watermarking method. We present some applications of watermarking in this section.

1. Copyright protection. It is the main application of watermarking. A watermark, such as the information about the source, copyright owner's name or signature, is embedded to the work in order to prevent other parties from claiming the copyright of the work. In this application, a very high level of robustness is required against various attacks intended to remove the watermark. The watermark should be unambiguous and still can be used to determine rightful ownership even if other parties embedded additional watermarks.

2. Content authentication. In this application, any changes or tampering with the content, like compression, noise pollution, should be detected. It can be achieved by so-called "fragile/semi-fragile watermark". Authentication requires the lowest level of robustness by definition.

3. Fingerprinting watermark. In this application, a digital watermark uniquely identifying the end user of a creation is embedded in the creation's carrier signal as the payload. The fingerprint watermarks can be embedded at the time of distribution to a specific customer. The robustness requirement is high against deliberate attacks.

4. Copy protection. A desirable distribution system does not allow unauthorized copying of the media. A copy status watermarks can be embedded at a copy protection system to prevent the copying of the content or limit the number of copying.

1.3 Properties of digital watermarking

The watermark system can be characterized by a number of defined properties. The relative importance of each property is dependent on watermark application and purpose. We will discuss about fidelity, blind or informed detection, robustness, false positive/false negative rate, and cost of digital watermarking in this section.

1.3.1 Fidelity

Theoretically, the watermark could be perceptible or imperceptible. However, in most applications, an imperceptible watermarking system is preferred. In other words, to

keep the good fidelity of watermarked work is required. The fidelity of a watermarking system refers to the perceptual similarity between the original and watermarked versions of the cover work [2].

It is extremely difficult to identify the acceptable fidelity because too little is known about human perception. We usually approximately use the objective and subjective measurement to define a threshold. For example, mean squared error (MSE, refer to Equ. (1.1)), signal-to-noise ratio (SNR, refer to Equ. (1.2)) and peak signal-to-noise ratio (PSNR, refer to Equ. (1.3), assume 8-bit pixel depth), are objective measurements. In these equations, c_1 and c_2 are N-dimension vectors in media space. Just noticeable difference (JND) is the subjective measurement. In the field of psychophysics, a JND is the amount that something must be changed for the difference to be noticeable.

$$D_{mse}(c_1, c_2) = \frac{1}{N}\sum_{i}^{N}(c_1[i] - c_2[i])^2 \tag{1.1}$$

$$D_{snr}(c_1, c_2) = 10\log_{10}\frac{\sum_i^N (c_1[i])^2}{\sum_i^N (c_2[i] - c_1[i])^2} \tag{1.2}$$

$$D_{psnr}(c_1, c_2) = 10\log_{10}\frac{255^2}{\frac{1}{N}\sum_i^N (c_1[i] - c_2[i])^2} \tag{1.3}$$

1.3.2 Blind or informed detection

Depending on the applications, the original data is available or unavailable to the watermark detection system. In some applications, such as a transaction tracking application, both the original and watermarked work are available during detection. However, in a copy control application, the original work is unavailable. We refer to a detector that requires accessing the original data as an informed detector and a detector that does not require the original data as a blind detector. The blind detection is normally more complicated than the informed detection. Therefore, most recent research focuses on the blind detection. So does our research in this thesis.

1.3.3 False positive and false negative probability

A false positive is the detection of a watermark in a work that does not actually contain one. A false negative is the detection of a non-watermark in a work that does actually contain one. The false positive or negative probability is the error probability of each detection. The required false positive probability depends on the application.

Depending on the method used for watermark detection, several methods are employed to estimate the false positive probability. For example, if the normalized correlation is used as the detection measure, the approximate Gaussian method and Fisher Z-statistic method can be used to estimate the false positive probability [3]. Then a threshold used for judging the presence of the watermark can be defined according to the false positive probability.

1.3.4 Robustness

Robustness is the ability to detect the watermark after common signal processing. The common signal processing for an image includes spatial filtering, lossy compression, printing and scanning and affine transformation. Geometric distortions (rotation, scaling, and translation), which are one type of affine transformations, are considered the most challenging for a detector to detect the watermark. We will tackle this difficult problem.

1.3.5 Cost

The estimation of the cost of deploying a watermark embedder and detector is complicated. In a real-time system, speed is important; while in other systems, speed is not a big issue. In a blind detection system, the cost refers to how much reference related to the original work need to be transferred to the detector. Other issues include whether the embedding and detection system need the special hardware or software, and so on.

1.4 Classification of digital watermarking

We can classify digital watermarking into different categories according to the host signal [4].

1. Image watermarking. Most research concentrates on the development of watermarking algorithms for images. Digital image watermarking algorithms can be extended to other media, for example video and audio.

2. Video watermarking. The increasing bandwidth of Internet connections allows the transmission of video streams with acceptable quality. A video sequence

consists of still images. In addition to all the attacks to image watermarking, video watermarking should resist more typical operations, such as color space conversion, frame rate conversion, real-time transmission, and compression.

3. Audio watermarking. Digital audio watermarking is a technology to hide information in an audio file without the information being audible to the listener, and without affecting in any way the audio quality of the original file.

4. 3D virtual objects watermarking. Compared to 2D objects, a 3D model may contain a wide range of data objects. For example, a VRML 2.0 file includes geometry of objects defined by polygons, lines, or predefined shapes (e.g., cylinders, spheres, or cones). These objects have attributes, such as shininess, per-surface or per-vertex colors, per-surface or per vertex normal vectors, per-vertex texture coordinates, and texture images. Watermarking three-dimensional data is becoming increasingly important.

5. Others such as holograph watermarking, text watermarking, and software watermarking.

In this thesis, most of the discussions are focused on digital image watermarking. Without considering visible watermarks, we can classify watermarks into fragile, semi-fragile, or robust watermarks. A fragile watermark is simply a mark to become undetectable even after the slightest alteration. Semi-fragile watermark is so designed that the integrity is proven if the content of the image has not been tampered with, despite some mild processing on the image. Robust watermarking is specially designed to withstand a wide range of "attacks", which basically are trying to remove the watermark, but without destroying the image. Fig. 1.2 illustrates the types of watermarks. Our algorithm in this thesis belongs to invisible robust watermarking.

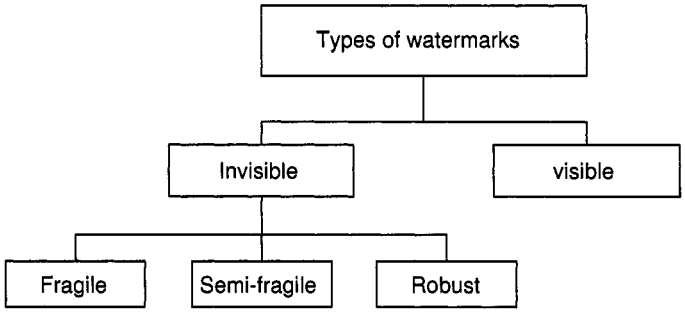


Figure 1.2: Types of watermarks.

The watermark can be added to the spatial domain or transformed domain. The well known transforms are FFT, DCT, and DWT. Other techniques are possible as well, for example using fractals.

1.5 Evaluation and benchmarking of watermarking systems

Robustness is the most important evaluation issue for a watermarking system. The robustness of a watermarking system depends on the following aspects [1]:

1. Amount of data payload. Data payload refers to the number of bits a watermark encodes within a Work. The more information is embedded to the Work, the lower the watermark robustness.

2. Watermark embedding strength. Embedding strength refers to the amount of change made to the host signal to represent the watermark data. The higher watermark embedding strength increases the robustness of a watermark system while increases the perceptibility of watermark as well. Therefore, there is a trade-off between the watermark embedding strength and watermark perceptibility.

3. Other issues depending on the applications.

To evaluate a watermarking system, benchmarking can be used as a fair system under comparable conditions. Several benchmarking tools have been developed by universities. Stirmark [5], which is the first benchmarking tool, can generate a number of image modifications to verify watermarking schemes. Stirmark programs include cropping, flipping, rotation, rotation-scaling, sharpening, shearing, Gaussian filtering, random bending, linear transformations, aspect ratio changes, scale changes, line removal, JPEG compression, etc. Checkmark [6] also provides an efficient and effective tool to evaluate and rate watermarking schemes. Checkmark provides wavelet compression, projective transformations, modelling of video distortions, warping, template removal attack, non-linear line removal, collage attack, down/up sampling, dithering, thresholding, etc. Certimark [7] is a new benchmarking system, which is a complete benchmarking suite for still image and video watermarking technologies.

1.6 Contributions of the thesis

This thesis presents the fundamental theories and techniques necessary for rotation, scaling and translation (RST) invariant watermarking, and gives a literature review on four different RST invariant image watermarking approaches. Based on the log-polar mapping and the new filtering method, we propose [8][9][10][11] an image rectification method that can be used by any RST invariant image watermarking algorithms. We utilize the properties of log-polar mapping to obtain the RST parameters by using the new filtering method. The main contribution of this thesis is the image rectification scheme based on the new filtering method, and the three RST invariant watermarking schemes, which are the applications of our rectification scheme.

We implement the rectification scheme and three watermarking schemes. Matching results show our filtering method is the only one that can be used in the LPM domain, and the rectification accuracy is very high. The experimental results of the three applications show that they are all robust against RST transformations, JPEG compression and other attacks while the fidelity of the watermarked image is preserved.

Publications generated from the research:

1. Yan Liu and Jiying Zhao, A new filtering method for RST invariant image watermarking, Proceeding of IEEE international workshop on haptic, audio and visual environments and their applications, pp. 101-106, Sep. 2003.

2. Yan Liu and Jiying Zhao, Rotation, scaling, and translation invariant image watermarking based on radon transform, Canadian conference on computer and robot vision, pp. 225-232, May, 2004.

3. Yan Liu and Jiying Zhao, A rectification scheme for RST invariant image watermarking, Canadian conference on electrical and computer engineering, pp. 527-530, May, 2004.

4. Dong Zheng, Yan Liu, and Jiying Zhao, RST invariant digital image watermarking based on a new filtering method, Proceedings of 7th International Conference on Signal Processing (ICSP'04, IEEE, CIE, IEE), Beijing, China, Aug 31-Sep 4, 2004.

5. Yan Liu, Dong Zheng, and Jiying Zhao, A rectification scheme for RST invariant image watermarking, accepted by IEICE Transactions on Fundamentals of

Electronics, Communications, and Computer Sciences, Special Section on Cryptography and Information Security, to be published in January 2005 (Vol. E88-A, No.1), LETTER.

6. Yan Liu, Xiangsheng Wu, Dong Zheng, Jiying Zhao, and Jianping Yao, Phase Information in RST Invariant Image Watermarking, accepted by the Proceedings of the CSEE (Chinese Society of Electrical Engineering).

1.7 Thesis structure

The rest of the thesis is organized as follows. In Chapter 2, we introduce the fundamental theories and techniques necessary for RST invariant watermarking systems. In Chapter 3, we briefly review the RST invariant image watermarking methods in four main categories. In Chapter 4, we propose our RST rectification algorithm. In Chapter 5, we describe three application schemes of our rectification algorithm. In Chapter 6, we evaluate our rectification scheme, by comparing its matching results and performance to other filters, and by giving detection accuracies for RST parameters, and give the experimental results for the three applications. Finally, in Chapter 7 we conclude the thesis and give some suggestions and ideas for future research work.

Chapter 2

Fundamental theories and techniques

Many RST invariant image watermarking algorithms are based on the fundamental characteristics of the transformation domain of an image. As shown in Fig. 2.1, the most current RST invariant watermarking systems embed watermarks into the following five main transformation domains to obtain the watermarked image: 1) Spatial domain, 2) Frequency domain after discrete Fourier transform (DFT), 3) Log-polar mapping (LPM) domain, 4) Fourier-Mellin transform (FMT) domain, and 5) 1-D vector after Radon transform. Other techniques, such as circular harmonic functions and moment, also can be used for RST invariant watermarking algorithm. In this chapter, we introduce different image transformations, techniques and basic detection theories, which are normally used for RST resilient image watermarking algorithms.

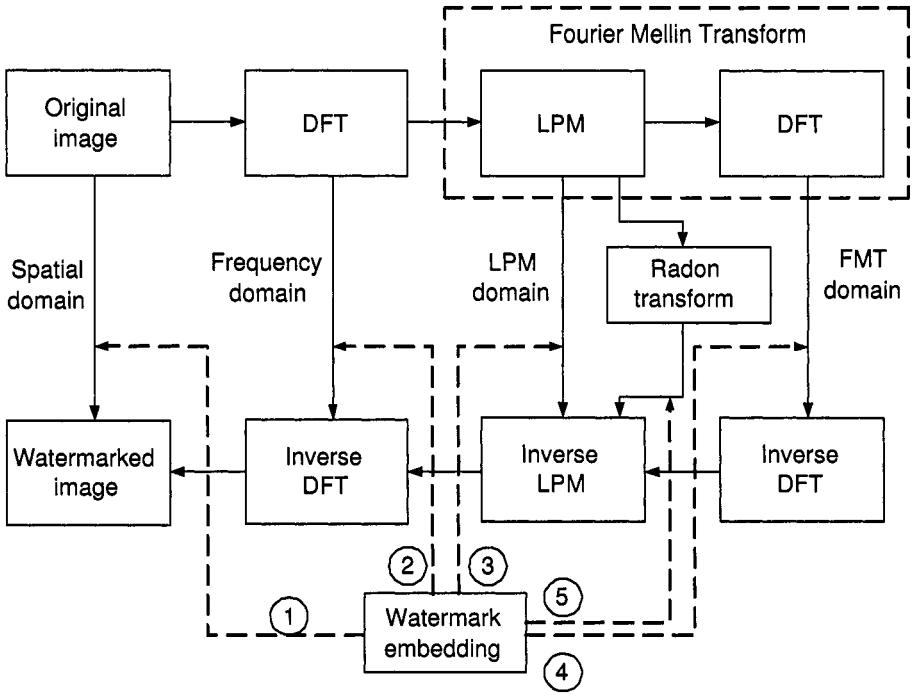


Figure 2.1: Common RST invariant watermark embedding systems.

2.1 Rotation, scaling, and translation transformations

First, we model Rotation, Scaling, and Translation (RST) transformations in this section.

2.1.1 Translation

A translation (or shift) is applied to an image by repositioning it along a straight-line path from one coordinate location to another [12]. We translate a two-dimensional point by adding translation distances, x_0 and y_0, to the original coordinate position (x, y) to move the point to a new position (x', y').

$$\begin{cases} x' &= x + x_0 \\ y' &= y + y_0 \end{cases} \tag{2.1}$$

The translation distance pair (x_0, y_0) is called a translation vector or shift vector.

2.1.2 Rotation

A two-dimensional rotation is applied to an image by repositioning it along a circular path in the xy plane. We obtain the transformation equations for rotating a point at (x, y) through an angle α about the origin counterclockwise:

$$\begin{cases} x' &= x\cos\alpha + y\sin\alpha \\ y' &= -x\sin\alpha + y\cos\alpha \end{cases} \tag{2.2}$$

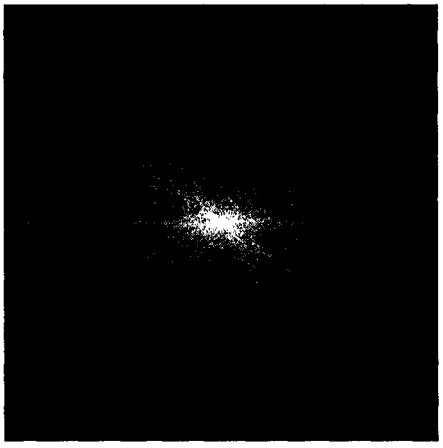
2.1.3 Scaling

A scaling transformation alters the size of an image. We obtain the transformation equations by multiplying the coordinate values (x, y) by scaling factors σ_x and σ_y to produce the transformed coordinates (x', y'):

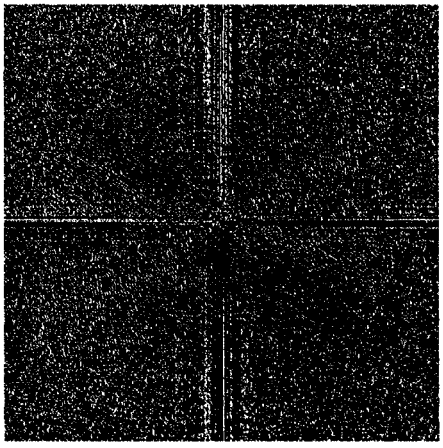
$$\begin{cases} x' &= x \cdot \sigma_x \\ y' &= y \cdot \sigma_y \end{cases} \tag{2.3}$$

Scaling factor σ_x scales images in the x direction, when σ_y scales in the y direction. While σ_x and σ_y are assigned the same value, a uniform scaling maintains the relative image proportions.

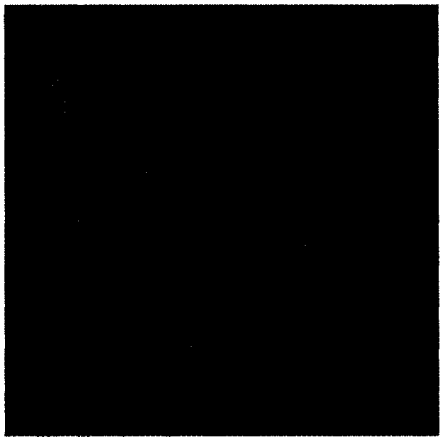
(a) *Lena*.



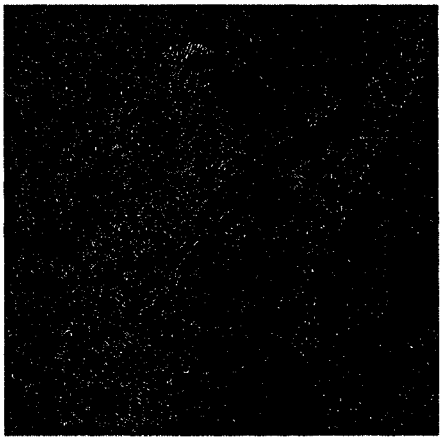
(b) The magnitude spectrum of image *Lena*.



(c) The phase spectrum of image *Lena*.



(d) The reconstructed image using only the amplitude spectrum.



(e) The reconstructed image using only the phase spectrum.

Figure 2.2: DFT and inverse DFT of image *Lena*.

2.2 Discrete Fourier transform

The DFT (Discrete Fourier Transform) of an image $f(x,y)$ of size $N_1 \times N_2$ and the corresponding IDFT (Inverse DFT) are defined as follows [13]:

$$F(u,v) = \frac{1}{N_1 N_2} \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f(x,y) e^{-j2\pi(ux/N_1+vy/N_2)} \tag{2.4}$$

$$f(x,y) = \sum_{u=0}^{N_1-1} \sum_{v=0}^{N_2-1} F(u,v) e^{j2\pi(ux/N_1+vy/N_2)} \tag{2.5}$$

The Fourier transform $F(u,v)$ is a complex function. Each function value has a real part $R(u,v)$ and an imaginary part $I(u,v)$, at each frequency (u,v) of the frequency spectrum:

$$F(u,v) = R(u,v) + jI(u,v) \tag{2.6}$$

where $j = \sqrt{-1}$. This can be expressed alternatively using the exponential form as:

$$F(u,v) = |F(u,v)| e^{j\phi(u,v)} \tag{2.7}$$

where $|F(u,v)|$ is the magnitude of the Fourier transform and $\phi(u,v)$ is the phase angle. The square of the magnitude is equal to the amount of energy or power at each frequency of the image and is defined as:

$$|F(u,v)|^2 = R^2(u,v) + I^2(u,v) \tag{2.8}$$

The phase angle describes the amount of phase shift as each frequency and is defined as:

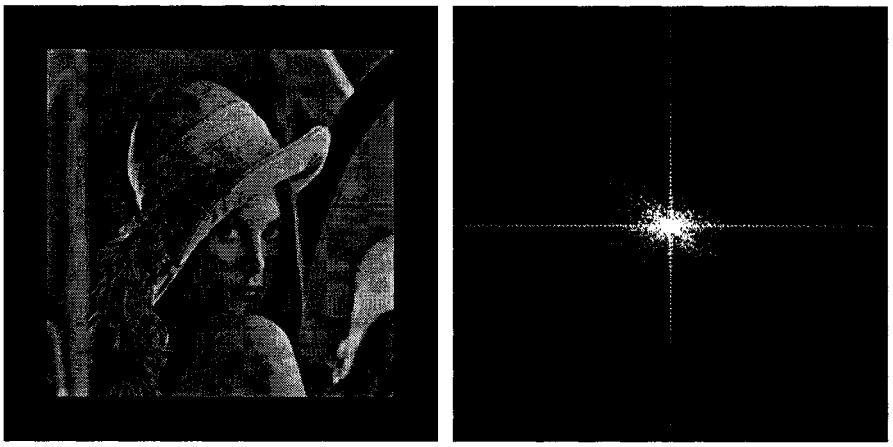
$$\phi(u,v) = tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right] \tag{2.9}$$

It is understood that the phase information is considerably more important than the amplitude information in preserving the visual intelligibility of the picture. Fourier synthesis of the structure from only the amplitude of the diffraction with zero phases does not reconstruct the correct atomic arrangement, whereas reconstruction from the phase data with unity magnitude does [14]. DFT and inverse DFT processing of image *lena* is shown in Fig. 2.2. Fig. 2.2 (e) clearly shows that the image reconstructed from only the phase information closely resemble the original image, while the reconstructed image from only the amplitude information does not, refer to Fig. 2.2 (d).

2.3 Properties of discrete Fourier transform

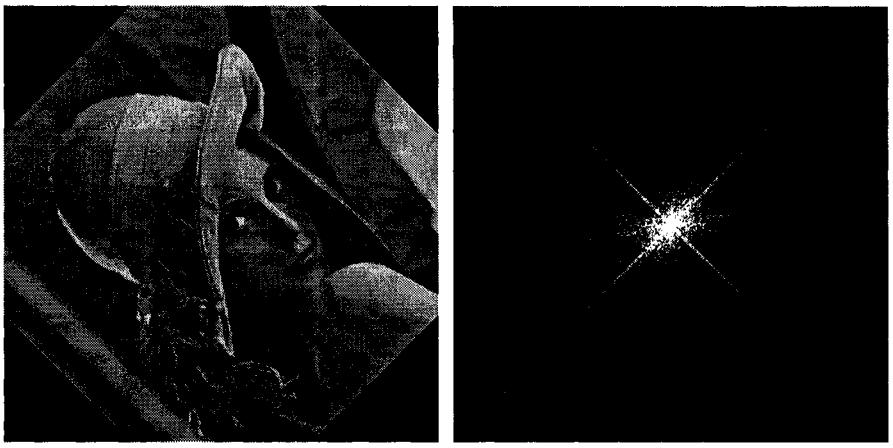
The relationship between an original image $i_0(x,y)$ and a rotated, scaled, and translated version of the image, $i_1(x,y)$, is shown as follows [15][16]:

$$i_1(x,y) = i_0(\sigma(x\cos\alpha + y\sin\alpha) - x_0, \sigma(-x\sin\alpha + y\cos\alpha) - y_0) \tag{2.10}$$



(a) *Lena* scaled by 80%.

(b) The magnitude spectrum of the scaled image *Lena*.



(c) *Lena* rotated by 45°.

(d) The magnitude spectrum of the rotated image *Lena*.

(e) *Lena* circularly translated by 100 pixels.

(f) The magnitude spectrum of the translated image *Lena*.

Figure 2.3: Properties of Fourier transform.

where the RST parameters are α, σ and (x_0, y_0), respectively.

The Fourier transform of $i_1(x, y)$ is $I_1(u, v)$, the magnitude of which is given by [16][17]:

$$|I_1(u,v)| = |\sigma|^{-2}|I_0(\sigma^{-1}(u\cos\alpha + v\sin\alpha), \sigma^{-1}(-u\sin\alpha + v\cos\alpha))| \quad (2.11)$$

As shown in Equ. (2.11), the magnitude spectrum is independent of the translational parameters (x_0, y_0), which is the translation property of the Fourier transform [18]. When an image is rotated in the spatial domain by angle α, its magnitude spectrum will be rotated by the same angle α. Scaling in the spatial domain can cause an inverse scaling in the frequency domain. Fig. 2.3 illustrates all these properties.

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

Figure 2.4: Log-polar mapping.

Figure 2.5: Inverse LPM of image *Lena*.

2.4 Log-polar mapping and inverse log-polar mapping

The log-polar mapping (LPM) is a conformal mapping from the points on the Cartesian plane (x, y) to points on the log-polar plane (ρ, θ):

$$\begin{cases} x &= e^{\rho}\cos\theta \\ y &= e^{\rho}\sin\theta \end{cases} \tag{2.12}$$

As shown in Fig. 2.4, a log-polar mapping is a sampling process. A circle of sampling points on the Cartesian plane are mapped to a straight line of sampling points on the log-polar mapping along the θ axis. The bilinear interpolation is used to compute the values of those sampling points. The reason is discussed in Section 2.12.

The inverse log-polar mapping (ILPM) is:

$$\begin{cases} \rho &= \ln(\sqrt{x^2+y^2}) \\ \theta &= \tan^{-1}(\frac{y}{x}) \end{cases} \tag{2.13}$$

However, ILPM deteriorates the image quality to an unacceptable level, shown in Fig. 2.5.

If we apply log-polar mapping to the Fourier magnitude of an image, we can rewrite Equ. (2.11) by using log-polar coordinates:

$$\begin{cases} u = e^{\rho}\cos\theta \\ v = e^{\rho}\sin\theta \end{cases} \tag{2.14}$$

where $\rho \in \Re^2$ and $0 \leq \theta < 2\pi$. Then the magnitude of the Fourier spectrum can be written as [15][16]:

$$|I_1(u,v)| = |\sigma|^{-2}|I_0(\sigma^{-1}e^{\rho}\cos(\theta-\alpha), \sigma^{-1}e^{\rho}\sin(\theta-\alpha))| \tag{2.15}$$

or

$$|I_1(\rho,\theta)| = |\sigma|^{-2}|I_0(\rho-\ln\sigma, \theta-\alpha)| \tag{2.16}$$

Equ. (2.16) demonstrates that the magnitude 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 ρ axis, that image rotation results in a cyclical shift of α along the angle θ axis, and that image translation has no effects in the LPM domain.

2.5 Fourier-Mellin transform

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

$$F_1(\omega_\rho, \omega_\theta) = |\sigma|^{-2} e^{-j(\omega_\rho \cdot \ln \sigma + \omega_\theta \cdot \alpha)} F_0(\omega_\rho, \omega_\theta) \tag{2.17}$$

The Fourier magnitude of the two LPM mappings is related by

$$|F_1(\omega_\rho, \omega_\theta)| = |\sigma|^{-2} |F_0(\omega_\rho, \omega_\theta)| \tag{2.18}$$

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. (2.18) is equivalent to computing the Fourier-Mellin transform [15]. Equ. (2.18) demonstrates that the magnitude of Fourier-Mellin spectrum is scaled by $|\sigma|^{-2}$ caused by scaling transform, and is invariant to rotation and translation. If normalized correlation is used to get rid of the problem of $|\sigma|^{-2}$, Fourier-Mellin transform is truly invariant to RST.

2.6 Radon transform

The Radon transform represents an image as a collection of projections along various directions [19]. It is used in areas ranging from seismology to computer vision. Projections can be computed along any angle q. In general, the Radon transform of $f(x, y)$ is the integral of f along a straight line parallel to the y' axis, which can be expressed as Equ. (2.19), shown in Fig. 2.6.

$$R_q(x') = \int_{-\infty}^{\infty} f(x' \cos q - y' \sin q, x' \sin q + y' \cos q) dy' \tag{2.19}$$

where

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos q & \sin q \\ -\sin q & \cos q \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.20}$$

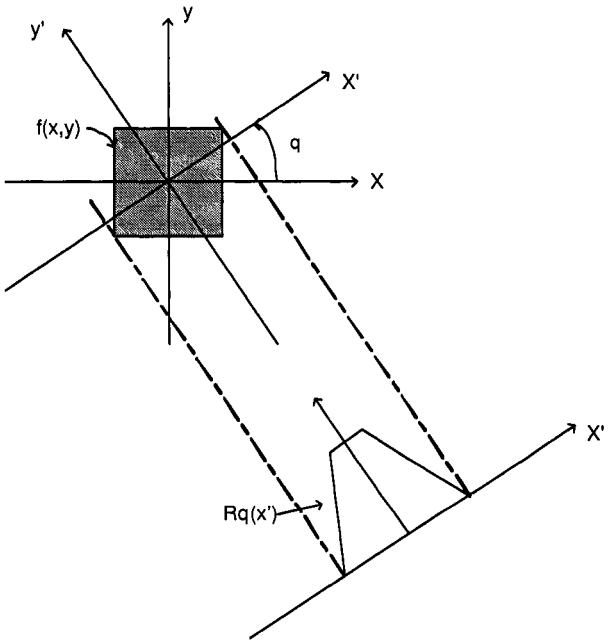


Figure 2.6: Radon transform.

In the most cases, we choose the angle q to be zero.

We define $g(\theta)$ to be a 1-D projection of $|I(\rho,\theta)|$ such that [16]:

$$g(\theta) = \sum_j |I(\rho_j, \theta)| \tag{2.21}$$

$g(\theta)$ is invariant to both translation and scaling. Rotations result in a circular shift of the values of $g(\theta)$ [16].

2.7 Circular harmonic functions

Circular harmonic expansion is another method used for shift, scale and rotation invariant pattern recognition [20][21][22]. The circular harmonic function (CHF) is useful in representing the rotational property of an image, which can be expressed in polar coordinates with period 2π in angle and thus can be expressed in terms of a Fourier series expansion in angle [23]. By taking the single harmonic, a circular harmonic filter is invariant to rotation. As with CHFs for rotation invariance, the radial harmonic filters (RHF) decomposes the object into a set of logarithmic radial harmonics. By taking the single harmonic, a radial harmonic filter function is invariant to the shift and scale.

Let $f(x,y)$ denote the reference image in Cartesian coordinates, we can transform $f(x,y)$ into polar coordinates $f(r,\theta)$. Because $f(r,\theta)$ is periodic in θ with period of 2π, we can use a Fourier series expansion in θ as follows [20]:

$$f(r,\theta) = \sum_k f_k(r) e^{jk\theta} \tag{2.22}$$

$$f_k(r) = \frac{1}{2\pi} \int_0^{2\pi} f(r,\theta) e^{-jk\theta} d\theta \tag{2.23}$$

where $f_k(r)$ is the k-th circular harmonic function (CHF) of $f(x, y)$.

Let $h(x, y)$ represent a correlation filter in Cartesian coordinates, a CHF decomposition is as follows:

$$h(r, \theta) = \sum_k h_k(r) e^{jk\theta} \tag{2.24}$$

where

$$h_k(r) = \frac{1}{2\pi} \int_0^{2\pi} h(r, \theta) e^{-jk\theta} d\theta \tag{2.25}$$

Then, the correlation function between $f(x, y)$ and $h(x, y)$ is shown in Equ. (2.26).

$$\begin{aligned} c &= \int\int f(x, y) h^*(x, y) dx dy \\ &= \int_0^{2\pi} d\theta \int_0^{\infty} r dr f(r, \theta) h^*(r, \theta) \\ &= \int_0^{\infty} r dr \int_0^{2\pi} \left[\sum_k f_k(r) e^{jk\theta} \cdot \sum_l h_l^*(r) e^{-jl\theta} \right] d\theta \end{aligned} \tag{2.26}$$

Because $\int_0^{2\pi} e^{j(k-l)\theta} d\theta$ is zero when $k \neq l$, the above equation can be expressed as:

$$c = \sum_{k=-\infty}^{\infty} C_k \tag{2.27}$$

where

$$C_k = 2\pi \int_0^\infty f_k(r) h_k^*(r) r dr \tag{2.28}$$

When the input image is rotated by the angle ϕ in the clockwise direction, the correlation function is given as follows:

$$c(\phi) = \sum_{k=-\infty}^{\infty} C_k e^{jk\phi} \tag{2.29}$$

If we only use a single circular harmonic

$$\begin{cases} f_s(r,\theta) = f_k(r) e^{jk\theta} \\ h_s(r,\theta) = h_k(r) e^{jk\theta} \end{cases} \tag{2.30}$$

then, the correlation in Equ. (2.26) is expressed as

$$c_s(\phi) = C_k e^{jk\phi} \tag{2.31}$$

The output's center intensity is a constant as shown in the Equ. (2.31). It is invariant to the rotation of the image. The same strategy can be used for RHF.

2.8 Moments

For a 2-D continuous function $f(x, y)$, the moment of order $(p + q)$ is defined as [13]:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{2.32}$$

for $p, q = 0, 1, 2, \ldots$.

The central moments are defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \overline{x})^p (y - \overline{y})^q f(x, y) dx dy \tag{2.33}$$

where $\overline{x} = \frac{m_{10}}{m_{00}}$ and $\overline{y} = \frac{m_{01}}{m_{00}}$

If $f(x, y)$ is a digital image, then Equ. (2.33) becomes

$$\mu_{pq} = \sum_x \sum_y (x - \overline{x})^p (y - \overline{y})^q f(x, y) \tag{2.34}$$

A set of seven invariant moments can be derived from the second and third moments [13][24].

The normalized central moments, denoted η_{pq}, are defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{2.35}$$

where

$$\gamma = \frac{p + q}{2} + 1 \tag{2.36}$$

for $p + q = 2, 3, \ldots$.

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.37}$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \tag{2.38}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.39}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.40}$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$

$$+ \quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \tag{2.41}$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{2.42}$$

$$\begin{aligned} \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{2.43}$$

This set of moments is invariant to rotation, scaling, and translation.

2.9 Similarity measurement

When the cross-correlation is used as similarity measurement, it can be computed as the different types of inner product of two images.

1. Linear correlation

   Linear correlation is the most basic. The linear correlation between two images f and g can be described as follows:

   $$r_{lc}(f, g) = \frac{1}{M \times N} \sum_x \sum_y f(x, y) g(x, y) \tag{2.44}$$

   where, $M \times N$ is the size of the images. If f is the reference mark pattern and g is the extracted mark pattern from the image, r_{lc} is called similarity. One of the problems with linear correlation is that the detection values are highly dependent on the magnitudes of the watermark pattern extracted from the images. Therefore, for many extraction methods, the watermark will not be robust against the attacks, such as, changing the brightness of images [2].

2. Normalized correlation

This problem can be solved by normalizing the extracted mark pattern and the reference mark pattern to unit magnitude before computing the inner product between them.

$$r_{nc}(f,g) = \sum_x \sum_y \widetilde{f}(x,y)\widetilde{g}(x,y) \tag{2.45}$$

where

$$\begin{cases} \widetilde{f}(x,y) &= \frac{f(x,y)}{\sqrt{\sum_x \sum_y f(x,y)^2}} \\ \widetilde{g}(x,y) &= \frac{g(x,y)}{\sqrt{\sum_x \sum_y g(x,y)^2}} \end{cases} \tag{2.46}$$

We refer to Equ. (2.45) as the normalized correlation. However, normalized correlation is not robust against changes in the DC term of a work, such as, the addition of a constant intensity to all pixels of an image [2].

3. Correlation coefficient

The third form of cross-correlation is the correlation coefficient, which computes the cross-correlation by subtracting the means of two images before the normalized correlation.

$$r_{cc}(f,g) = r_{nc}(\widetilde{f}(x,y), \widetilde{g}(x,y)) \tag{2.47}$$

where

$$\begin{cases} \widetilde{f}(x,y) &= f(x,y) - \overline{f}(x,y) \\ \widetilde{g}(x,y) &= g(x,y) - \overline{g}(x,y) \end{cases} \tag{2.48}$$

where, $\overline{f}$ and $\overline{g}$ are the mean of f and g, respectively. Because the mean of an

image has been subtracted, the correlation coefficient is robust against changes in the DC term of a work [2].

2.10 Phase correlation

The phase correlation is another efficient approach to rectify the watermark position to avoid exhaustive search [25]. Kuglin and Hines proposed the phase correlation based on the shift theorem of the Fourier transform [26]. Given an image f_0 and its shifted version f_1 with the displacement (x_0, y_0), i.e.,

$$f_1(x, y) = f_0(x - x_0, y - y_0) \tag{2.49}$$

Then, the relationship between their corresponding Fourier transforms F_0 and F_1 is as follows:

$$F_1(u, v) = e^{-j(ux_0+vy_0)} F_0(u, v) \tag{2.50}$$

If we compute the cross-power spectrum of the two images defined as

$$C = \frac{F_1(u, v)F_0^*(u, v)}{|F_1(u, v)F_0^*(u, v)|} = e^{j(ux_0+vy_0)} \tag{2.51}$$

where F^* is the complex conjugate of F. The shift 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,

$$D = \mathfrak{F}^{-1}(angle(C)) \tag{2.52}$$

where $\mathfrak{F}^{-1}$ is the inverse Fourier transform, and $angle(C)$ is the phase of C.

Based on the property of the Fourier transform, the Fourier transform of impulse function $\delta(x-d)$ is e^{-jwd}. Equ. (2.52) gives a two-dimensional δ function centered at the displacement. So D is a function which is an impulse, that is, it is approximately zero everywhere except at the displacement.

2.11 Harris detector

Feature points detectors find salient points in natural images. Normally, these points are located near corners and edges of the image. The Harris corner detector is one of the feature points detectors, developed for 3D reconstruction [27]. It extracts the corner points of an image. A Harris corner detector first calculates the horizonal and the vertical gradients of an image, G_x and G_y. Then, two gradient images are filtered by a low-pass filter to get G'_x and G'_y. Then, the shape matrix M is formed for each pixel [28]:

$$M(i,j) = \begin{bmatrix} \sum_{m,n}(G'_x(m,n))^2 & \sum_{m,n} G'_x(m,n)G'_y(m,n) \\ \sum_{m,n} G'_x(m,n)G'_y(m,n) & \sum_{m,n}(G'_y(m,n))^2 \end{bmatrix} \tag{2.53}$$

where (m,n) represents all pixel positions of a window area centered at the pixel (i,j). By using $M(i,j)$, the Harris corner detector's output for each image pixel is based on

the trace and determinant of M [28]:

$$H(i,j) = det(M(i,j)) - k \cdot trace(M(i,j)) \tag{2.54}$$

where k is an arbitrary constant. Feature points extraction is achieved by searching for the response $H(i,j)$ larger than a threshold η.

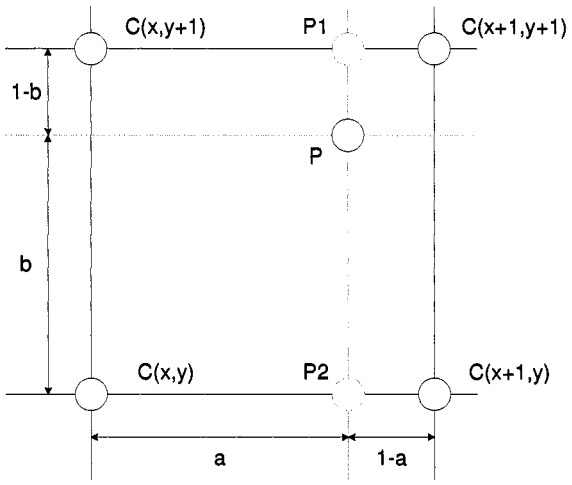


Figure 2.7: Bilinear interpolation.

2.12 Interpolation

Interpolation is a process for estimating values by taking an average of known values at neighboring points. It has important applications in areas such as signal and image processing. Many methods are used for interpolation, such as, 1) *nearest*, the value of an interpolated point is the value of the nearest point; 2) *bilinear*, the value of an interpolated point is a combination of the values of the four closest points; 3) *bicubic*, the value of an interpolated point is a combination of the values of the sixteen closest points. Among these three popular methods, *bicubic* is the most advanced one. It

produces the smoothest surface. But it needs the most memory and computing time. *Bilinear* is faster and less memory-intensive than *bicubic*, and produces a smoother surface than *nearest*. Therefore, most applications, including the algorithm of this thesis use the bilinear interpolation.

Here, we give an example about the bilinear interpolation during log-polar mapping. Each point in the log-polar magnitude spectrum is computed from a weighted average of four points in the Cartesian magnitude spectrum, shown in Equ. (2.55) and Fig. 2.7.

$$\begin{aligned} P(\rho,\theta) &= C(x,y)\cdot(1-a)\cdot(1-b) \\ &+ C(x,y+1)\cdot(1-a)\cdot b \\ &+ C(x+1,y)\cdot a\cdot(1-b) \\ &+ C(x+1,y+1)\cdot a\cdot b \end{aligned} \tag{2.55}$$

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. 2.7) 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 $C(x,y)$.

In this chapter, we introduced the fundamental theories and techniques used for RST invariant digital image watermarking algorithms. These include RST transformations, the DFT and its properties, the LPM and ILPM, Fourier-Mellin transform, Radon transform, circular harmonic functions, moments, similarity measurement, phase correlation, Harris detector, and interpolation. This chapter helps understand the rest of the thesis.

Chapter 3

Literature review

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 [29]. 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 a slight amount, could dramatically reduce the receiver's ability to retrieve the watermark [30]. Recently it has been clear that even very small geometric distortions can prevent the detection of a watermark, especially in the case the original unwatermarked image is unavailable to the detector [16].

In this chapter, we roughly divide the RST invariant image watermarking method into four main categories. 1) RST invariant domain based approaches 2) Template based approaches 3) Feature structure based approaches 4) Moment based approaches.

3.1 RST invariant domain based approaches

This category includes watermarking schemes that are based on the domain that is invariant to geometric attacks. These domains correspond to the properties of discrete Fourier transform (DFT), log-polar mapping (LPM), Fourier-Mellin transform (FMT), Radon transform, circular harmonic function (CHF), etc.

3.1.1 Fourier-Mellin transform based approaches

O'Ruanaidh et al. [15] first 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 Fourier-Mellin transform and an inverse DFT) after considering the original phase [15][30]. As we have discussed in Section 2.5, the Fourier-Mellin transform is a log-polar mapping (LPM) followed by a Fourier transform, while an inverse Fourier-Mellin transform is an inverse Fourier transform followed by an inverse log-polar mapping (ILPM). In the scheme, the embedded watermark may be extracted by transforming the watermarked image into a RST invariant domain. However, they noted very severe implementation difficulties which might have hampered further work in this area [16]. The change of coordinate system means that interpolation is needed. In our experiment, the LPM and ILPM will cause an unacceptable loss of image quality as shown in Fig. 2.5 even if bilinear interpolation is used.

O'Ruanaidh et al. [15] noticed this and suggested to embed watermark in the RST

invariant domain independent 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 a large magnitude component in the center (the low frequency areas), so the computation of the Fourier Mellin transform is problematic because the interpolation only performs well if the neighboring samples are of the same value level. So the scheme is difficult to implement according to our experiments. 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 the watermark and the robustness of the watermark.

3.1.2 Phase correlation and log-polar mapping based approaches

Theoretically the Fourier-Mellin domain is the best place to embed watermark, considering that it is invariant to RST. However, the LPM and ILPM will cause the loss of the image quality or the watermark data.

Zheng et al. [17] proposed an alternative to the Fourier-Mellin transform based digital image watermarking. They embed the watermark in the LPM domain to simplify the effects of RST transformations into simple shifts (refer to Equ. (2.16)). For watermark embedding, the approximate ILPM is employed to replace ILPM, in order to eliminate the imprecision caused by ILPM. They select a desired location in the

LPM magnitude spectrum for embedding the watermark data sequence. The watermarking locations in the Cartesian DFT magnitude spectrum is approximated from the watermark points in LPM domain. 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 they do not apply the IDFT before the approximate ILPM, rotation and scaling operations in the spatial domain of the watermarked image will cause translation of the watermark positions in the LPM domain, either a circular shift along the angle axis or a vertical shift along the log-radius axis (refer to Equ. (2.16)). Exhaustive search in the embedding area can be used to handle the shift of watermark positions caused by rotation and scaling [31]. However exhaustive search is time consuming and produces large correlation coefficients for unwatermarked images. Therefore, they use phase correlation to rectify the watermark position to avoid exhaustive search [25] if the original image is available. They calculate the displacement between the LPM of the original image and the LPM of the watermarked image, according to Equ. (2.51) and Equ. (2.52). They rectify the original watermark position according to calculated displacement. Then, the similarity measurement is conducted by using the normalized correlation, refer to Equ. (2.45). A decision may be made by comparing the value of similarity with a predefined threshold.

This scheme is very reliable in displacement calculations and is invariant to any angles of rotation and translation, invariant to scaling when the scale is in a reasonable range, and very robust to JPEG compression and other attacks. The main contribution of the work includes 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. However, in some applications, detection must be

preformed without access to the original work, i.e., a copy control application. This scheme is unfeasible when the original image is not available. This is the drawback of this scheme.

3.1.3 Radon transform based approaches

As mentioned in Section 2.6, $g(\theta)$, the Radon transform of the LPM spectrum of Fourier magnitude of an image, is invariant to both translation and scaling of the image. Rotations result in the circular shift of the values of $g(\theta)$. Lin et al.'s algorithm is based on this principle [16]. They embed the watermark into a one-dimensional (1-D) signal 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.

Rotation is the only problem that they need to deal with. They handle rotation by an exhaustive search. However, exhaustive search is time-consuming and it may produce higher false positive probability. A false positive error occurs when the detector incorrectly indicates that a watermark is present [2]. In order to solve this problem, a rectification method can be employed to battle rotation [9].

In Lin et al.'s approach, LPM and ILPM are both applied to the image and the watermark data. Because ILPM is non-invertible, they instead perform an iterative approximation of ILPM. As shown in Fig. 3.1. $\mathbf{F}'$s, the points on the circle, are the elements of the watermarked image in the log-polar domain. $\mathbf{C}'$s, the points on the square, are the elements of the watermarked image in the Cartesian domain.

To get the values for $\mathbf{C}'$, they add the weighted difference between all the corre-

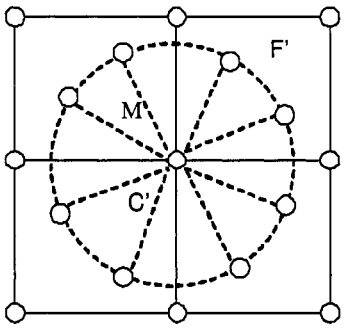


Figure 3.1: Approximate ILPM.

sponding $\mathbf{F}'$ and $\mathbf{F}$ to $\mathbf{C}$ [16]. The relationship can be expressed by Equ. (3.1):

$$C' = C + \frac{M_{ij}(F'_i - F_i) + M_{kj}(F'_k - F_k)}{M_{ij} + M_{kj}} \tag{3.1}$$

where, $\mathbf{F}$ and $\mathbf{C}$ represent the elements of the original image in the log-polar domain and in the Cartesian domain, respectively. $\mathbf{M}$ contains the weights for interpolation.

This method is a rough approximation of desired inversion. It can not extract the exact inverse version $\mathbf{C}'$. The more iterations of this method, the closer to the desired inversion [16].

Their experimental results show that the scheme is invariant to rotation of any angles, scaling up to 120%, translation, and JPEG compression at quality factors down to 70.

3.1.4 Circular harmonic function based approaches

Kim et al. [23] proposed a rotation-tolerant watermarking method by using a circular harmonic function correlation filter. They use this new filter instead of the conventional matched filter to achieve the rotation tolerant correlation peak at the watermark detection process. They design the filter by using the properties of circular harmonic

function (CHF), discussed in Section 2.7. The CHF filter h can be expressed by the filter design function f_0 as follows [23]:

$$h = f_0(W, c(\phi)) \tag{3.2}$$

with

$$c(\phi) = \begin{cases} 1, for\ |\phi| \leq \phi_t \\ 0, for\ |\phi| > \phi_t \end{cases} \tag{3.3}$$

where W is the watermark pattern and ϕ_t is the tolerance angle. $c(\phi)$ is the correlation function shown in Equ. (2.26). This filter also can be optimal by using Optimal Trade-off Circular Harmonic Function (OTCHF) [32], which gets the trade-off performance in three criteria: 1) sensitivity to additive noise, 2) sharp correlation peak with little energy in sidelobe, and 3) similarity measure of all correlation output.

They modulate and decorrelate the watermark pattern by using a whitening procedure [33][34] and embedding the watermark into the original image by simple addition. At the detector, they measure the correlation by computing the inner product between the test image and the OTCHF filter h. The OTCHF filter can be designed off-line. Hence, it is suitable for any real-time watermark detector [23].

This method is the only rotation tolerant watermark detection. Experimental results show that this method is tolerant to the rotation angles less than a predefined angle ϕ_t. However, when the image is rotated more than ϕ_t, there are some relatively high peaks. This is because of the correlation of the original image. It maybe produces the higher false positive and false negative probabilities. The overlap of the histogram of peak-to-sidelobe ratio (PSR) [23] between the unwatermarked images and the watermark

images before or after rotation shows this problem.

3.2 Template based approaches

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. [29][35] 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. They use approximately 14 points along two lines which go through the origin in the DFT domain at two random angles with radii varying between two random values. A linear transformation of an image will produce an inverse linear transformation in the DFT domain. Moreover, with a linear transformation, a line going through the origin will be transformed into a corresponding line going through the origin [29]. Therefore, the transformations could be detected through the relationship of two lines. Once the template is detected, these transformations are inverted and the spread spectrum signal is retrieved.

Using templates has its advantage that it is much more robust than the watermark itself since they concentrate a significant amount of energy into a few points in the FFT [29]. 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. To make the template invisible, they use a fixed structured template. These templates may be easy to remove [36] since they usually represent peaks in a transform domain. This type of template is applicable for all images. 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 invertible ability of any geometric distortions [16].

According the test results shown in their paper [29], the PSNR of the watermarked image is 38 dB. The algorithm is robust against scaling between 0.75 and 2, JPEG compression with quality factor down to a level of 75, crop up to 50% of the image, and shearing of 0.01% and 0.1%.

3.3 Feature structure based approaches

Feature based watermarking algorithms provide another solution to geometrical distortions. The location of the watermark is not linked with image coordinates, but with image semantics [37]. The problem of geometrical synchronization can be solved because the image content represents an invariant reference to geometrical transformations.

P. Bas et al. proposed a geometrically invariant watermarking algorithm by using feature points and Delaunay tessellation [37]. First, they detect robust feature points in an image by using Harris corner detector, mentioned in Section 2.11. Then, they create a triangular tessellation (also called Delaunay tessellation) of the image based on a set of the feature points. The image is divided into a set of disjoint triangles by Delaunay tessellation. A random sequence is generated as the watermark sequence. It is transformed to the same shape of each triangle using affine transform and interpolation. The transformed watermark sequence is embedded to each triangle to obtain the watermarked image. During the detection process, the tessellation is reconstructed. Each triangle is mapped to the shape of the original triangular watermark. Finally the correlation is computed to judge the existence of the watermark.

The important step of this algorithm is to choose Delaunay tessellation. The tes-

sellation has two properties: 1) if a vertex disappears, only the connected triangles are modified, 2) if the vertex is moved inside the triangle area, the tessellation is not modified.

This algorithm is robust to StirMark, shearing transformation, scaling operation, rotation, printing and scanning process, and JPEG compression. But it is only robust to slight rotations up to 10°.

Several other watermarking algorithms are feature-based. They are so-called the second-generation watermarking scheme because the feature of an image is exploited for embedding watermark [38]. Duric et al. have proposed a method for recognizing geometrically distorted images and restoring their original appearance by using image feature points [39]. Sun et al. have developed a method based on image feature to identify the geometrical transformation [40]. Alghoniemy et al. introduced a RST synchronization scheme based on the wavelet decomposition of an image [41]. Dittman et al. have designed a content-based watermarking method that does not require the original image and uses a self spanning pattern [42].

3.4 Moment based approaches

The moments of objects have been widely used in pattern recognition. Alghoniemy et al. proposed geometric invariance in image watermarking based on moment and image normalization. They use geometric moments to geometrically normalize the image before watermark embedding at the encoder and before watermark detection at the decoder [43]. Translation invariance can be achieved by using the central moments of the image which are origin independent, mentioned in Section 2.8. Scaling invariance can be obtained according to that the moments of the scaled image is the energy-scaled

version of the moments of the original image. Therefore, by normalizing the moments of an image, scaling invariance can be achieved. Scaling normalization transforms the image into its standard form by translating the origin of the image to its centroid $(\overline{x}, \overline{y})$. Changing the coordinates into $(\widehat{x}, \widehat{y})$ [43]

$$\widehat{x} = \frac{x - \overline{x}}{a}, \ \widehat{y} = \frac{y - \overline{y}}{b} \tag{3.4}$$

with

$$a = \sqrt{\frac{\beta\gamma}{m_{0,0}}}, \ b = \sqrt{\frac{\beta}{\gamma m_{0,0}}} \tag{3.5}$$

In Equ. (3.4) and Equ. (3.5) a and b are the factors to make the aspect ratio of an image to 1. a and b are defined by $al_x = bl_y$, where l_x and l_y are the height and the width of the image. β and $m_{0,0}$ are respectively the zero-order moment of $f((x/a),(y/b))$ and $f(x,y)$. γ is the aspect ratio of the image $f(x,y)$, defined as $\gamma = \frac{l_y}{l_x}$.

In order to realize the rotation normalization, two tensors t^1 and t^2 are defined as [43]

$$t^1 = \mu_{12} + \mu_{30}, \ t^2 = \mu_{03} + \mu_{21} \tag{3.6}$$

where μ is central moment, defined in Section 2.8. Then, the normalized angle θ is defined as

$$\theta = \arctan(-\frac{t^1}{t^2}) \tag{3.7}$$

After normalization, the distorted image and the original image have the same size,

direction and orientation. The normalizing scheme does not need the original image for implementing the normalization at decoder.

Alghoniemy et al. also proposed another approach to geometric transformation invariant watermarking [43]. In that approach, the watermark is based on invariant parameters extracted from the geometric moments of the image. Also, other moments, like complex moments and Zernike moments, are used for pattern recognition [44][45].

In this chapter, we reviewed the existing RST invariant digital image watermarking algorithms by dividing them into four main categories. Each approach has their own advantages and disadvantages. In the next chapter, we propose a new RST rectification algorithm that can be used by any RST invariant digital image watermarking schemes.

Chapter 4

Proposed RST rectification algorithm

This thesis presents an image rectification algorithm that can be used by any image watermarking algorithms to provide robustness against rotation, scaling and translation (RST) transformations. The scheme belongs to the first approach of RST invariant image watermarking introduced in Chapter 3: RST invariant domain based approach. As mentioned in Chapter 2, rotation and scaling transformations in the spatial domain results in cyclically translational shifts in the log-polar domain, which is the log-polar mapping (LPM) of the magnitude of the Fourier spectrum of the image. We propose our image rectification scheme based on this property of the log-polar domain.

4.1 Matching template

We cut a small block as a matching template from the LPM domain during the watermark embedding process. When extracting the watermark, we calculate the cross-

correlation between this matching template and the LPM spectrum of the Fourier transform of the watermarked image to get the translational shifts of the template in the LPM domain. Therefore, we can obtain the rotation and scaling parameters according to the relationship between the spatial domain and LPM domain (refer to Section 2.4). We employ the same strategy to an image in the spatial domain to detect the translation parameters. One or two templates are needed depending on the application.

Fig. 4.1 shows where we cut the matching templates. Template 1 is necessary for our rectification scheme to obtain the rotation and scaling parameters for any applications. Normally, we cut the template 1 from low and middle frequency of LPM domain of an image because the high frequencies will be significant affected by the compression and watermark embedding. Template 2 is optional, which is needed to find the translational parameter. Because the magnitude spectrum of an image is independent of the translation transformation according to the property of Fourier transform, applications that embed the watermark into the Fourier domain or LPM domain, do not need template 2.

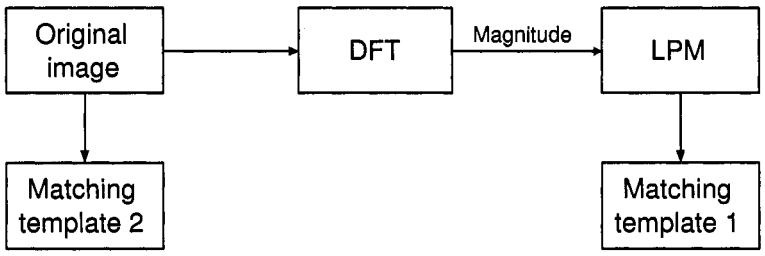


Figure 4.1: Matching templates.

4.2 Filters design

The key technique is to match the template and the watermarked image having undergone RST attacks. For a template g and an image f, where g is smaller than f, the template could be matched in the image by using the two dimensional cross-correlation function:

$$r(u,v) = \sum_x \sum_y f(x-u, y-v)g(x,y) \tag{4.1}$$

If the template matches the image exactly at a translation (i,j), the cross-correlation will have its peak at $r(i,j)$. The major disadvantage of the cross-correlation method is that it is computationally intensive. According to the correlation theorem, the Fourier transform of the correlation of the two images is the product of the Fourier transform of the one image and the complex conjugate of Fourier transform of the other.

$$r = \mathfrak{F}^{-1}[F(w_\rho, w_\theta) \cdot G^*(w_\rho, w_\theta)] \tag{4.2}$$

where

$$\begin{cases} F(w_\rho, w_\theta) &= \mathfrak{F}(f(\rho,\theta)) \\ &= A_F(w_\rho, w_\theta)e^{-j\Phi_F(w_\rho, w_\theta)} \\ G(w_\rho, w_\theta) &= \mathfrak{F}(g(\rho,\theta)) \\ &= A_G(w_\rho, w_\theta)e^{-j\Phi_G(w_\rho, w_\theta)} \end{cases} \tag{4.3}$$

and $\mathfrak{F}$ and $\mathfrak{F}^{-1}$ are DFT and inverse DFT, respectively, $*$ is the complex conjugate, and $G(w_\rho, w_\theta)$ is called a matching filter.

The fast Fourier transform based methods are fast and efficient. The following defines five types of traditional filters:

1. Classical matched filter, which is the Fourier transform of the template g, with full phase and amplitude.

$$G(\omega_\rho, \omega_\theta) = A_G(\omega_\rho, \omega_\theta) e^{-j\Phi_G(\omega_\rho, \omega_\theta)} \tag{4.4}$$

2. Amplitude-only filter, which is defined as the amplitude spectrum of the Fourier transform of the template g.

$$G_A(\omega_\rho, \omega_\theta) = A_G(\omega_\rho, \omega_\theta) \tag{4.5}$$

3. Inverse filter, which is the division between the phase spectrum and the amplitude spectrum of the template g.

$$G_I(\omega_\rho, \omega_\theta) = \frac{e^{-j\Phi_G(\omega_\rho, \omega_\theta)}}{A_G(\omega_\rho, \omega_\theta)} \tag{4.6}$$

4. Phase-only filter, which contains only the full phase spectrum of the template g.

$$G_\Phi(\omega_\rho, \omega_\theta) = e^{-j\Phi_G(\omega_\rho, \omega_\theta)} \tag{4.7}$$

5. Binary phase-only filter, which contains 1 or -1 according to the sign of the real part of the Fourier transform of template g.

$$G_{BPOF}(\omega_\rho, \omega_\theta) = e^{-j\Phi_{BPOF}(\omega_\rho, \omega_\theta)} \tag{4.8}$$

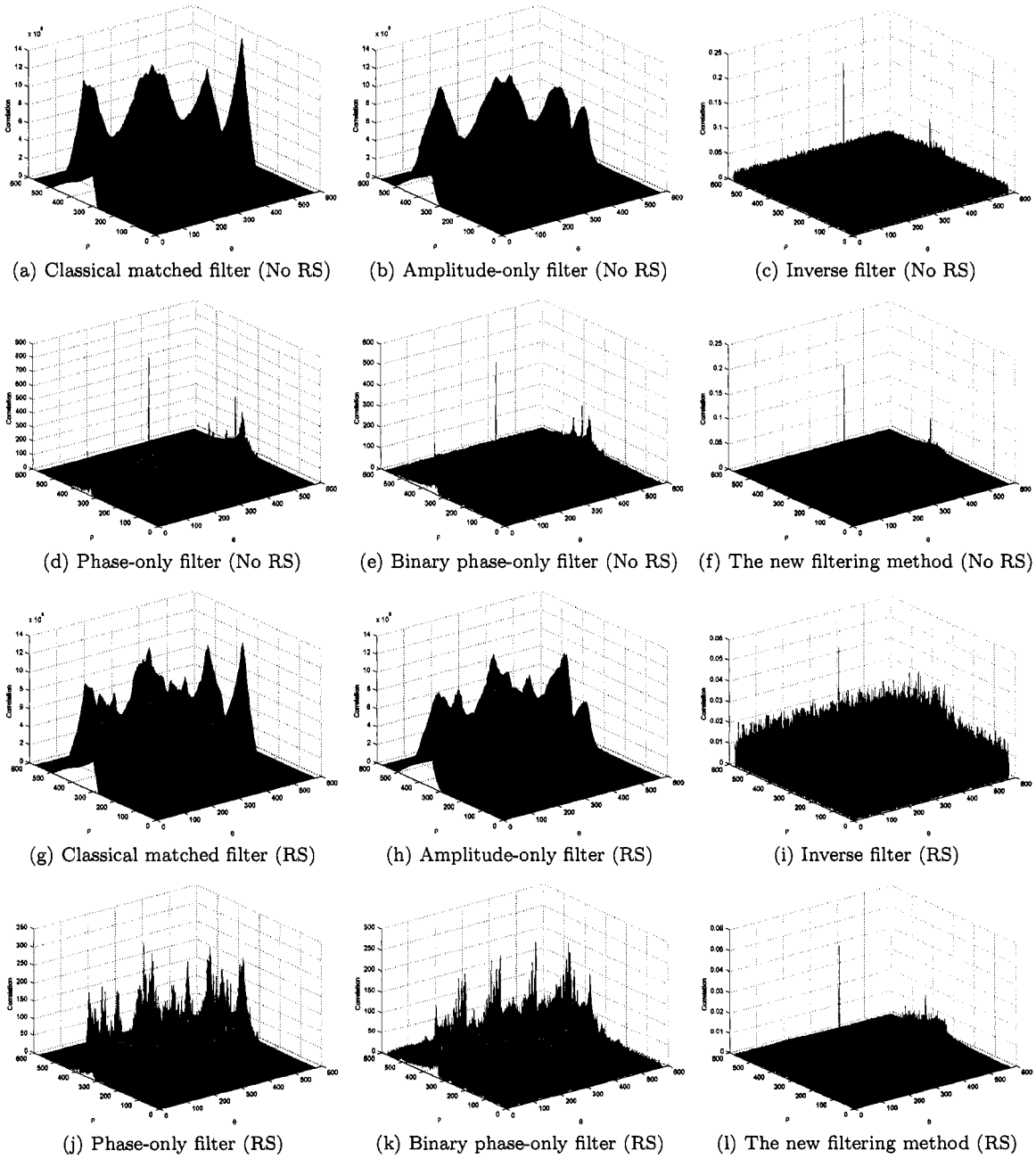
where

$$\Phi_{BPOF}(\omega_\rho, \omega_\theta) = \begin{cases} 0° & G_r \geq 0 \\ 180° & G_r < 0 \end{cases} \tag{4.9}$$

and G_r stands for the real part of the Fourier transform $G(\omega_\rho, \omega_\theta)$.

Phase information is considerably more important than the amplitude information in preserving the visual intelligibility of an image. A phase-only filter (POF) is obtained by setting the magnitude to unity for all the frequencies. The advantages of the POF are: 1) all the energy gets through the filter plane without magnitude function's attenuation; 2) the energy is concentrated in a much narrower peak with better discrimination capability; 3) the POF retained only the phase information requires less space to store the data [46]. Normally, the phase-only filter yields much sharper correlation peaks and better discrimination [21]. In the recent review papers, phase information was applied in optical signal processing [14] [46], image registration [47], broadcast applications [48], digital image watermarking [17].

Fig. 4.2, which is the cross-correlation in 3-dimensions between image *Lena* (512×512) and a template (64×64) by using different filters, illustrates the matching results. In Fig. 4.2, Fig. 4.3 and Table 6.1, "No RS" means that there is no rotation or scaling transformation applied to the watermarked image, while "RS" means that the watermarked image is rotated clockwise 10° and scaled 90%. The classical matched filter and the amplitude-only filter have unacceptable discrimination (refer to Fig. 4.2 (a), (b), (g) and (h)). The inverse filter, the phase-only filter and the binary phase-only filter perform reasonably well when there is no rotation or scaling applied to the watermarked image. The sharp peaks in Fig. 4.2 (c), (d) and (e) clearly indicate



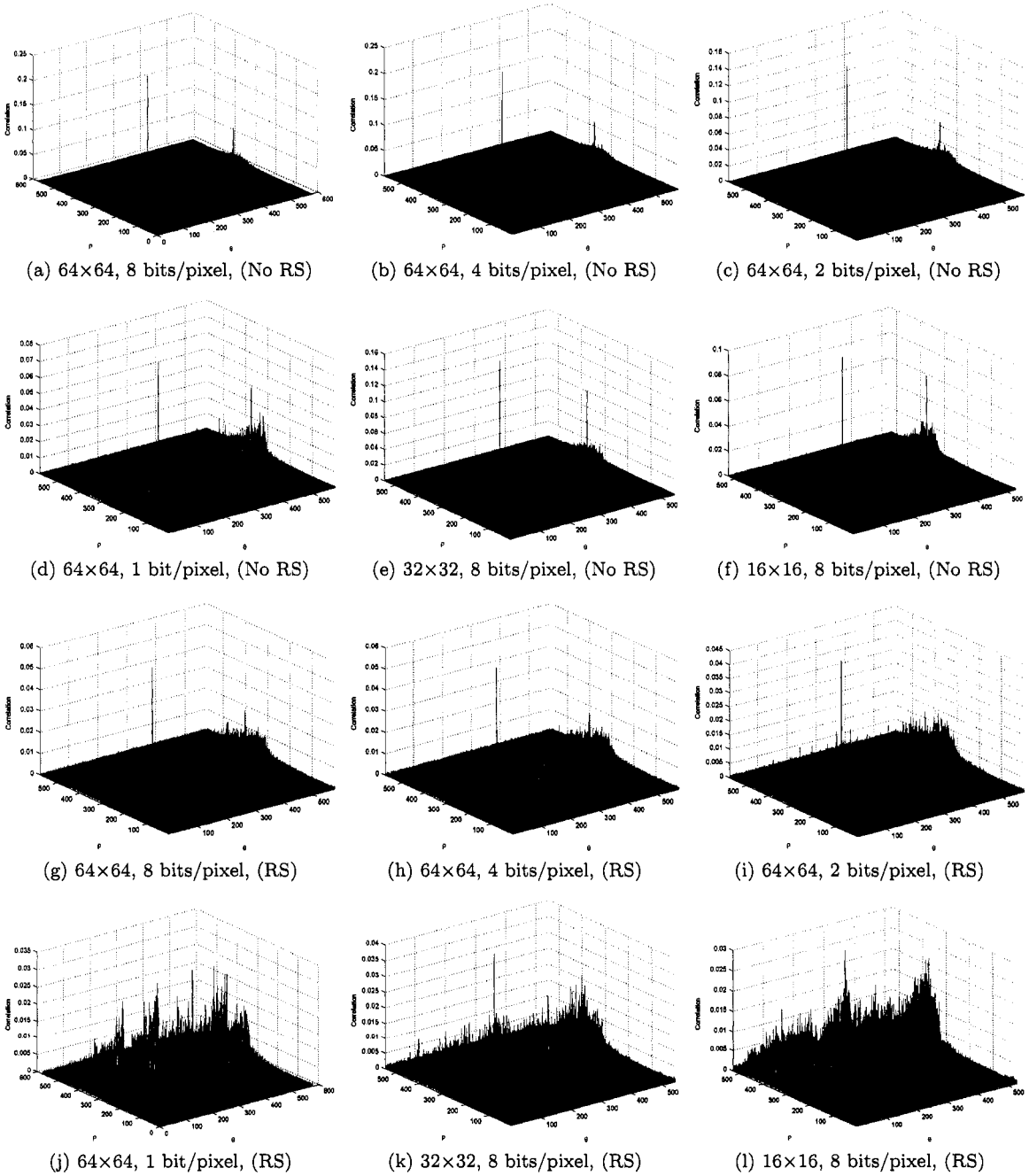
(a) Classical matched filter (No RS) (b) Amplitude-only filter (No RS) (c) Inverse filter (No RS)

(d) Phase-only filter (No RS) (e) Binary phase-only filter (No RS) (f) The new filtering method (No RS)

(g) Classical matched filter (RS) (h) Amplitude-only filter (RS) (i) Inverse filter (RS)

(j) Phase-only filter (RS) (k) Binary phase-only filter (RS) (l) The new filtering method (RS)

Figure 4.2: Matching results of different filters.



(a) 64×64, 8 bits/pixel, (No RS) (b) 64×64, 4 bits/pixel, (No RS) (c) 64×64, 2 bits/pixel, (No RS)

(d) 64×64, 1 bit/pixel, (No RS) (e) 32×32, 8 bits/pixel, (No RS) (f) 16×16, 8 bits/pixel, (No RS)

(g) 64×64, 8 bits/pixel, (RS) (h) 64×64, 4 bits/pixel, (RS) (i) 64×64, 2 bits/pixel, (RS)

(j) 64×64, 1 bit/pixel, (RS) (k) 32×32, 8 bits/pixel, (RS) (l) 16×16, 8 bits/pixel, (RS)

Figure 4.3: Matching results of the new filtering method with different parameters.

the position of the template in the LPM of the watermarked image. However, if the watermarked image has undergone rotation or scaling transform, these five filters fail to produce acceptable discrimination (refer to Fig. 4.2 (i), (j), and (k)).

Since all types of traditional filters fail to produce acceptable discrimination while rotation or scaling or both applied to the watermarked image. Therefore, we have to devise a new filtering method to deal with this problem.

From detection theory, correlation detectors are optimum in the case of a Linear Time Invariant (LTI), frequency non-dispersive, Additive White Gaussian Noise (AWGN) channel [49]. However, in most cases, we use real images as our experimental targets, the power spectrums of which are not white. It is still possible to achieve optimum detection in the case of non-white Gaussian noise, by applying a so-called whitening filter at the input of the correlation receiver. This filter transforms the non-white input signal of the receiver to a signal with a constant power spectrum [50]. As a result, we propose a new filtering method, named phase-only filtering method, to transform the non-white spectrum in the LPM domain of the image to a mapping with a unity power spectrum, and then apply phase-only filter to the resulting mapping. The filtering process is described as follows:

$$r = \mathfrak{F}^{-1}[F_{\Phi}(w_{\rho}, w_{\theta}) \cdot G^{*}_{\Phi}(w_{\rho}, w_{\theta})] \tag{4.10}$$

where

$$F_{\Phi}(w_{\rho}, w_{\theta}) = e^{-j\Phi_F(w_{\rho}, w_{\theta})} \tag{4.11}$$

and $\mathfrak{F}^{-1}$ is the inverse DFT. The new filtering method gives extremely sharp peaks no matter whether rotation and/or scaling are applied to the watermarked image or not

(refer to Fig. 4.2 (f) and (l)). It clearly indicates the position of the template when other filters could not.

The new filtering method performs reasonably well even for a smaller template and a fewer number of bits per pixel. From Fig. 4.3, we recommend that the minimum size for the matching template be 32×32, and that the minimum bits per pixel be 2 bits/pixel. Considering that the size of most images to be watermarked is much larger than 64×64, and that the images are most likely colored, we can say even a grey-scale matching template of 64×64 is not a big cost. The performance of 64×64 template is more than enough, and employing a larger template is not necessary.

4.3 Three filter comparison criteria

All these different filters can be used for image registration. In order to fairly compare various filters, we emphasize three different criteria: noise robustness, sharpness of the correlation peak, and Horner efficiency [51].

1. Noise robustness

   The first criterion is the noise robustness, which corresponds to the optimization of the signal-to-noise ratio (SNR) [51],

$$SNR = 10 \log_{10} \frac{|C_0|^2}{MSE} \tag{4.12}$$

   with

$$MSE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |C'_{ij}|^2}{M \times N} \tag{4.13}$$

where C_0 is the peak value of the cross-correlation between the template and the power spectrum of watermarked image in LPM domain, C'_{ij} is the cross-correlation between the template and the noise, and $M \times N$ are the size of the image. The higher the value of SNR, the more robust the filter to noise.

2. Peak sharpness

   In order to compare the sharpness of correlation peaks, the peak-to-correlation energy (PCE) is calculated [51]:

$$PCE = 10 \log_{10} \frac{|C_0|^2}{CPE} \tag{4.14}$$

   with

$$CPE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |C_{ij}|^2}{M \times N} \tag{4.15}$$

   where, C_{ij} is the cross-correlation function between the template and the power spectrum of watermarked image in LPM domain. C_0 is the maximum value of C_{ij}. We use decibel (dB) as the unit of PCE. The higher the value of PCE, the sharper the peak value of the cross-correlation compared to the sidelobe.

3. Horner efficiency

   In optical correlators, it is important to consider the efficiency of the correlation function. The maximum value of the optical filter must be less than one [51]. This characteristic is represented by Horner efficiency η_H. Mathematically, it is given as [14]:

$$\eta_H = \frac{\int\int |f(x,y) \star g^*(x,y)|^2 dxdy}{\int\int |f(x,y)|^2 dxdy} \tag{4.16}$$

where $f(x,y)$ is the input function, $g(x,y)$ is the filter function, $*$ stands for the complex conjugate, $\star$ stands for the complex cross-correlation because $f(x,y)$ and $g(x,y)$ are not same in our method.

4.4 Rectification algorithm

Depending on the domain we embed the watermark into, one or two templates are needed to calculate the RST parameters by using the new filtering method. If we embed the watermark in the Fourier transform domain or LPM domain, we need only one template from the LPM domain because the magnitude of Fourier transform is independent of the translation; if we embed the watermark in the spatial domain, we need one template from the LPM domain and one template from the spatial domain of the original image, respectively. These two templates are uncorrelated with the watermark data, and they can be used for the same image watermarked with different data.

4.4.1 Calculation of rotation and scaling parameters

First, we employ the template in the LPM domain to obtain the rotation and scaling parameters. The peak in the correlation spectrum is the position of the matching template in the LPM spectrum of the watermarked image having undergone attacks. Suppose the coordinate of the peak is (ρ_1, θ_1), and we know the original position of the

template is (ρ_0, θ_0), so the translations are:

$$\begin{cases} \Delta_\rho &= \rho_1 - \rho_0 \\ \Delta_\theta &= \theta_1 - \theta_0 \end{cases} \tag{4.17}$$

where Δ_ρ and Δ_θ are the number of pixels shifted along the ρ axis, and the θ axis respectively. They correspond to the rotation and scaling parameters, respectively, in the spatial domain. Therefore, the rotation and scaling parameters can be calculated by Equ. (4.18).

$$\begin{cases} \alpha' &= \frac{360^0 \cdot \Delta_\theta}{N'} \\ \sigma' &= e^{\frac{\ln(r_{max}) \cdot \Delta_\rho}{M'}} \end{cases} \tag{4.18}$$

with

$$r_{max} = \sqrt{(M/2)^2 + (N/2)^2} \tag{4.19}$$

where, M' and N' are, respectively, the number of pixels along the ρ axis and θ axis in the LPM domain ($M' \times N'$ is the size of the LPM domain), and $M \times N$ is the size of the magnitude spectrum of the Fourier transform. $M' \times N'$ may be different from $M \times N$ because the sampling points can be increased or decreased during the log-polar mapping.

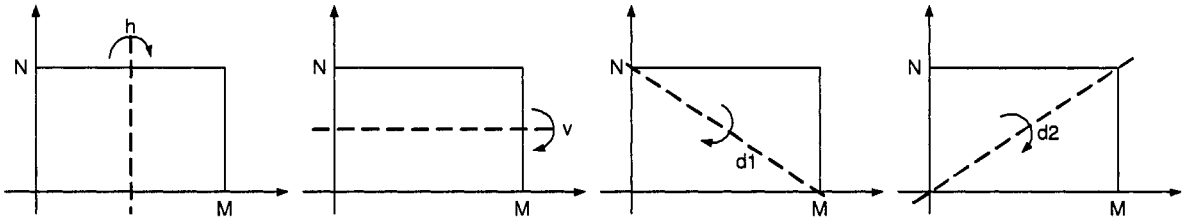
α' and σ' are, respectively, the rotation angle and scaling ratio in the spatial domain calculated from Δ_θ and Δ_ρ in the LPM domain. Therefore, we can invert the scaling and rotation transforms to synchronize the watermarked image geometrically.

4.4.2 Calculation of translation parameters

Because the magnitude of the Fourier transform of an image is independent of the translational parameters, we have to calculate the translational parameters in the spatial domain instead of the Fourier transform domain. We need to cut another block in the spatial domain as a template and compute the cross-correlation between this template and the watermarked image having undergone RST attacks by using the phase-only filtering method. Suppose the coordinate of the peak is (x_1, y_1), and we know the original position of the template is (x_0, y_0), so the translational parameters are:

$$\begin{cases} \Delta_x = x_1 - x_0 \\ \Delta_y = y_1 - y_0 \end{cases} \tag{4.20}$$

It is very easy to invert the translation transform applied to the watermarked image.



(a) Horizontal flipping (F_H) (b) Vertical flipping (F_V) (c) Diagonal flipping 1 (F_{D1}) (d) Diagonal flipping 2 (F_{D2})

Figure 4.4: Flipping.

4.4.3 Detection of image flipping

Our rectification method works well even when an image is horizontally, vertically or diagonally flipped (refer to Fig. 4.4). The four possible flipped versions of an image

can be described as follows:

$$\begin{cases} f^{FH}(x,y) &= f(M+1-x,y) \\ f^{FV}(x,y) &= f(x,N+1-y) \\ f^{FD_1}(x,y) &= f(y,x) \\ f^{FD_2}(x,y) &= f(N+1-y,M+1-x) \end{cases} \tag{4.21}$$

where f^{FH}, f^{FV} , f^{FD_1} and f^{FD_2} are, respectively, the horizontally, vertically and diagonally flipped versions of the original image f, while M and N are, respectively, the width and height of the image.

Assume that F^{FH}, F^{FV}, F^{FD_1}, F^{FD_2} and F are, respectively the Fourier transform of f^{FH}, f^{FV}, f^{FD_1}, f^{FD_2} and f. The following equations give their relationships in the Fourier transform domain.

$$\begin{cases} F^{FH}(u,v) &= F(-u,v)\cdot e^{j2\pi u(M+1)} \\ F^{FV}(u,v) &= F(u,-v)\cdot e^{j2\pi v(N+1)} \\ F^{FD_1}(u,v) &= F(v,u) \\ F^{FD_2}(u,v) &= F(-v,-u)\cdot e^{j2\pi v(N+1)}\cdot e^{j2\pi u(M+1)} \end{cases} \tag{4.22}$$

If we insert

$$\begin{cases} u &= e^{\rho}\cos\theta \\ v &= e^{\rho}\sin\theta \end{cases} \tag{4.23}$$

into Equ. (4.22), we obtain a set of equations about the magnitude of the Fourier transform of the different images in the log-polar coordinates.

$$\begin{cases} |F^{FH}(u,v)| &= |F(-u,v)| \\ &= |F(-e^{\rho}cos\theta, e^{\rho}sin\theta)| \\ &= |F(e^{\rho}cos(180^{\circ}-\theta), e^{\rho}sin(180^{\circ}-\theta))| \\ |F^{FV}(u,v)| &= |F(u,-v)| \\ &= |F(e^{\rho}cos\theta, -e^{\rho}sin\theta)| \\ &= |F(e^{\rho}cos(-\theta), e^{\rho}sin(-\theta))| \\ |F^{FD_1}(u,v)| &= |F(v,u)| \\ &= |F(e^{\rho}sin\theta, e^{\rho}cos\theta)| \\ &= |F(e^{\rho}cos(90^{\circ}-\theta), e^{\rho}sin(90^{\circ}-\theta))| \\ |F^{FD_2}(u,v)| &= |F(-v,-u)| \\ &= |F(-e^{\rho}sin\theta, -e^{\rho}cos\theta)| \\ &= |F(e^{\rho}cos(270^{\circ}-\theta), e^{\rho}sin(270^{\circ}-\theta))| \end{cases} \tag{4.24}$$

Assume that L^{FH}, L^{FV}, L^{FD_1}, L^{FD_2} and L are, respectively, the LPM of the magnitude of the Fourier transforms F^{FH}, F^{FV}, F^{FD_1}, F^{FD_2} and F. The following equations give the relationships in the LPM domain.

$$\begin{cases} L^{FH}(\rho,\theta) &= L(\rho, 180^{\circ}-\theta) \\ L^{FV}(\rho,\theta) &= L(\rho, -\theta) \\ L^{FD_1}(\rho,\theta) &= L(\rho, 90^{\circ}-\theta) \\ L^{FD_2}(\rho,\theta) &= L(\rho, 270^{\circ}-\theta) \end{cases} \tag{4.25}$$

If we represent the above equations in pixels, we can obtain:

$$\begin{cases} L^{FH}(\rho,\theta) &= L(\rho, N' - \theta + 1) \\ L^{FV}(\rho,\theta) &= L(\rho, N' - \theta + 1) \\ L^{FD_1}(\rho,\theta) &= L(\rho, \frac{1}{4}N' - \theta + 1) \\ L^{FD_2}(\rho,\theta) &= L(\rho, \frac{3}{4}N' - \theta + 1) \end{cases} \tag{4.26}$$

where $M' \times N'$ is the size of the LPM domain.

In Equ. (4.26), L^{FH} and L^{FV} are exactly the same because of the symmetry property of the DFT domain and they are the flipped version of L along θ axis; while L^{FD_1} and L^{FD_2} are the shifted version of flipped L, with 90° shift along θ axis. Actually, any flipping can be obtained by one particular flipping followed by a rotation. After flipping and shifting back, our template and phase-only filtering method still work well. If the template cut from the LPM does not match the LPM of the Fourier transform domain of the watermarked image having undergone attacks, we try to flip the lop-polar magnitude of the watermarked image horizontally and match the template again to see whether the image is horizontally, vertically or diagonally flipped.

4.4.4 Cost of the rectification algorithm

In our implementation, the size of the template is 64×64 with 8 bpp (Section 4.2 gives the reason). So the file size of the template data is 4096 bytes. In practice, it can be compressed to minimize its data file size without losing computation accuracy in the phase-only filtering method. In this thesis, we simply use the JPEG compression method to compress the template to 1 KB. All of the following experiments are carried out under this compression ratio.

In this chapter, we proposed a new image rectification algorithm that can be used by any RST invariant digital image watermarking schemes. We devised a new filtering method that performs well in the LPM domain for template matching and also proposed the algorithm to detect the RST parameters. In the next chapter, we will introduce three application schemes of the proposed rectification algorithm.

Chapter 5

Applications of the proposed rectification algorithm

In this chapter, we describe three watermarking schemes of our rectification scheme and introduce the watermarking scheme for each of them. In Scheme I, II, and III, the watermark is embedded in the spatial domain, the magnitude of the Fourier transform domain, and the log-polar mapping of the Fourier transform of the image, respectively. In all these schemes, we use *Lena* as our test image, the size of which is 512×512.

5.1 Scheme I: embedding watermark in spatial domain

In this scheme, we embed the watermark into the spatial domain of the image. We detect RST parameters by using two templates and re-synchronize the image before the watermark extraction.

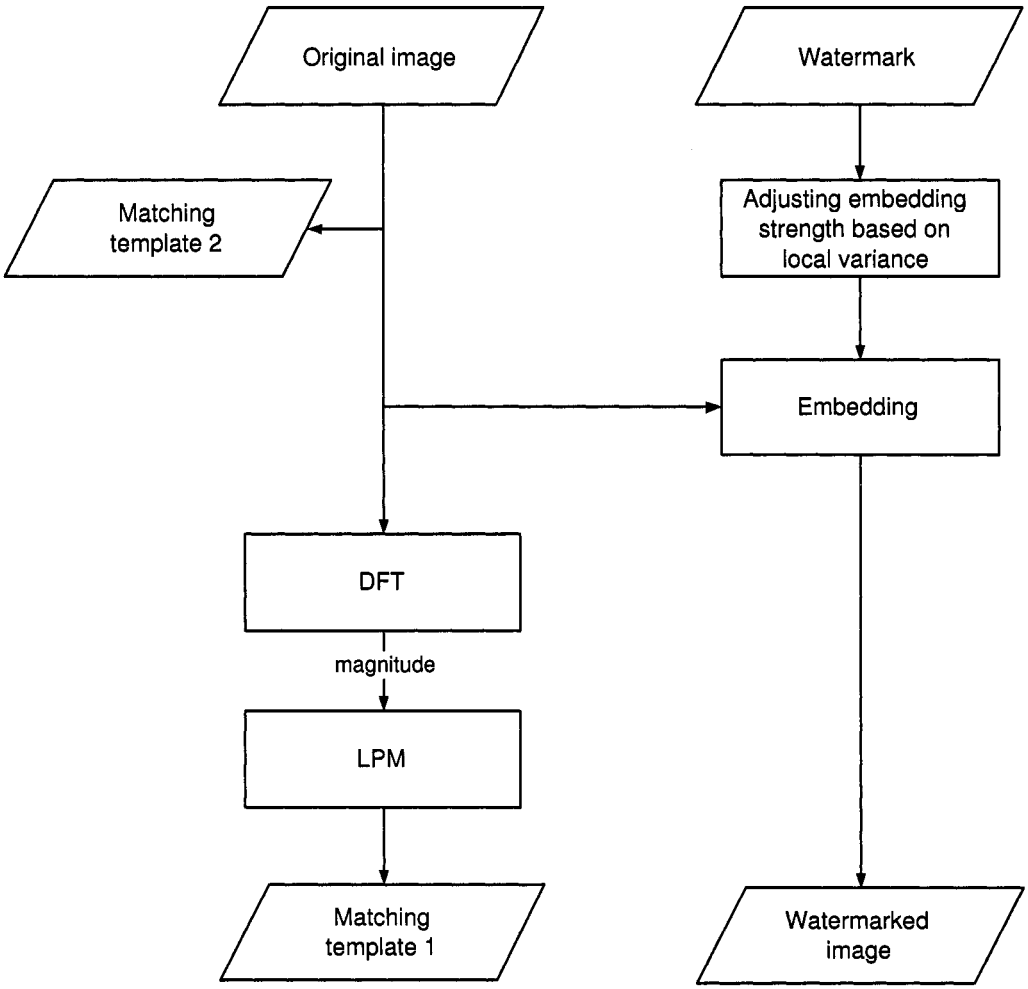


Figure 5.1: Scheme I: watermark embedding.

5.1.1 Watermark embedding

The watermark embedding process comprises the following steps, refer to Fig. 5.1:

1. Create a random two-dimensional sequence of $+1$ and -1 using a PN generator. Each value of the sequence is spread in blocks of size $B \times B$. This block-based watermark will be used as the reference watermark (also called watermarking key in this thesis).

2. Cut a matching template 1 in the log-polar mapping for calculating the rotation and scaling parameters, and cut a matching template 2 in the spatial domain for calculating translation parameters.

3. Determine the watermark embedding strength for each image pixel $X(i,j)$ based on the local variance values in order to perform invisible watermark embedding. The analysis of watermark embedding schemes shows that coefficients with larger variance can be embedded with a larger embedding strength [52]. So the embedding scheme proposed by Simitopoulos et al. [28] is used in this paper. Suppose $var(i,j)$ is the local variance value for each pixel, the following steps are used to calculate the embedding strength:

   (a) Determine a window centered at the pixel to be embedded;

   (b) Calculate the local variance $var(i,j)$ in this window area centered at each image pixel;

   (c) Find the maximum variance and use it for the normalization of the variances to the range $[0,1]$;

   (d) Use the following equation to determine the embedding strength:

$$\alpha(i,j) = \frac{var(X(i,j))}{max(var(X(i,j)))}\alpha_{max} + \alpha_{min} \tag{5.1}$$

   where α_{min} is the strength of the watermark that will be embedded in the case of a pixel whose neighborhood has zero variance (smooth area) and α_{max} is the maximum strength that the variance computation will contribute to the final watermark strength factor $\alpha(i,j)$.

4. Embed the watermark in the spatial domain representation of the image, as shown

by the following equation:

$$X'(i,j) = X(i,j) + \alpha(i,j) \cdot W(i,j) \tag{5.2}$$

where $W(i,j)$ is the watermark bit and $\alpha(i,j)$ is the embedding strength.

5.1.2 Watermark detection

The watermark detection is conducted in the following steps, shown in Fig. 5.2:

1. The phase-only filtering method described in Chapter 4 is used, and the rotation, scaling, and translation parameters are calculated by using Equ. (4.17), Equ. (4.18) and Equ. (4.20). The watermarked image is then geometrically re-synchronized using these three parameters.

2. The two-dimensional watermark reference is generated as in the embedding process.

3. The correlation value c between the watermarked image after re-synchronization and the watermark pattern is calculated by using:

$$c = \frac{1}{N} \sum_i \sum_j (X'(i,j) - \overline{X'}(i,j)) W(i,j) \tag{5.3}$$

where N is the size in pixels of the image. The local mean $\overline{X'}(i,j)$ of the $X'(i,j)$ of each pixel of the test image is calculated by using:

$$\overline{X'}(i,j) = \frac{1}{M} \sum_m \sum_n X'(i,j) \tag{5.4}$$

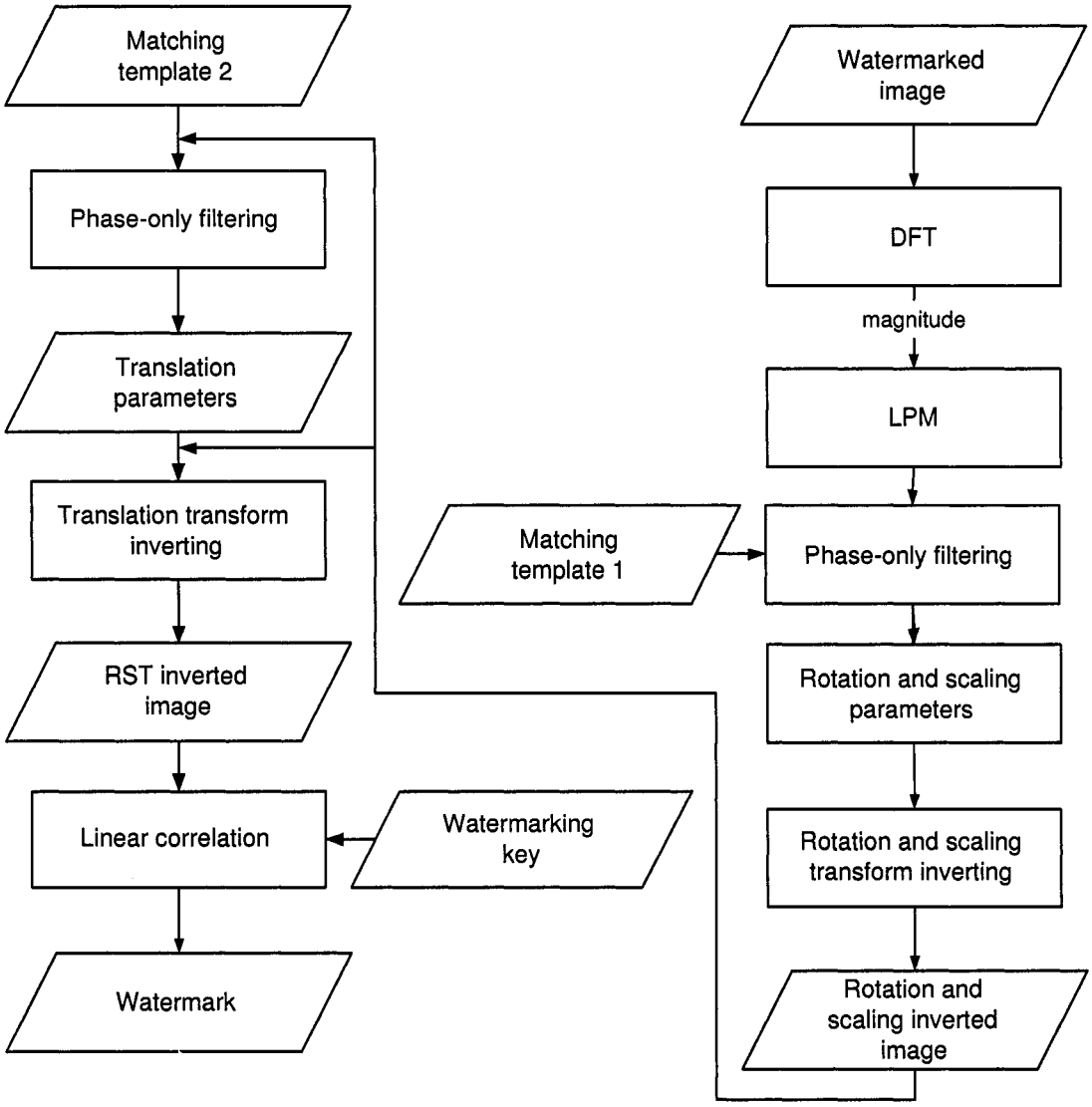


Figure 5.2: Scheme I: watermark detection.

where M is the size of the window within which the local mean is computed. This area contains all test image pixels belonging to this window centered at $X'(i,j)$, while excluding those pixels belonging to the block which contains the $W(i,j)$. This is done in order to exclude from the local mean $\overline{X'}(i,j)$ the pixel values $X'(i,j)$ that correlate with $W(i,j)$.

4. If the correlation value calculated by using Equ. (5.3) is greater than a predefined threshold, the image is declared to be watermarked. Otherwise, it is not.

5.2 Scheme II: embedding watermark in Fourier domain

In this scheme, we embed the watermark into the magnitude of the Fourier transform domain of an image. According to Equ. (2.11), the magnitude of the Fourier transform of the image is independent of the translational parameters, which is the translation property of the Fourier transform [18]. Therefore, we only need to calculate the rotation and scaling parameters and rectify the image according to these two parameters. We do not care about image translation. Therefore, only one template is needed.

5.2.1 Watermark embedding

The proposed watermarking scheme embeds the watermark in the following steps, refer to Fig. 5.3 :

1. Create a random two-dimensional sequence matrix using a pseudo-random number generator. The size of the matrix is $M \times N$, the same as the image size.

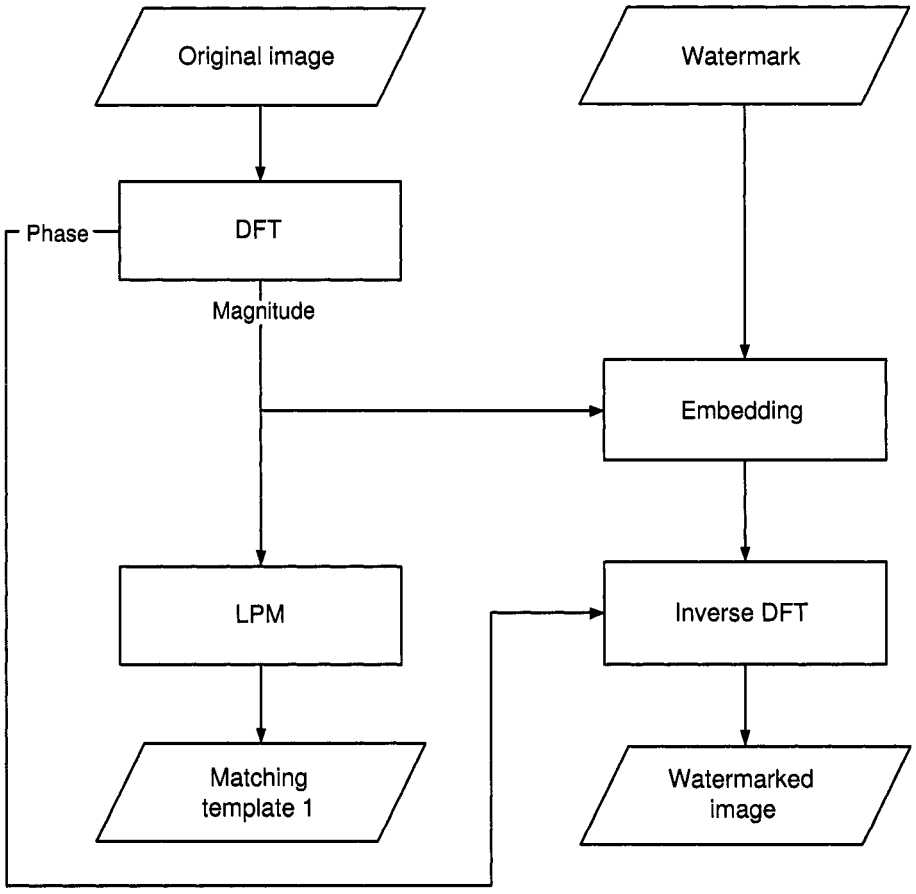


Figure 5.3: Scheme II: watermark embedding.

2. Compute DFT and LPM of the original image.

3. Cut a matching template from the log-polar mapping of the image for detection.

4. Set to zero the elements of this matrix if their distance to the center element is smaller than r_1 or larger than r_2, to make the watermark pattern a ring shaped pattern. The reason for this is that we do not want to embed watermark into the low frequency or the high frequency areas of an image. Embedding the watermark into the low frequency of an image could affect the image quality significantly, while embedding watermark into the high frequency of an image makes it easy to

remove the watermark by normal signal processing, such as, JPEG compression.

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

$$E' = E + \alpha * W \tag{5.5}$$

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 α is watermarking strength used to achieve the tradeoff between the robustness and the visibility of the watermark.

6. Obtain the watermarked image by computing inverse DFT of E' with the original phase.

5.2.2 Watermark detection

The watermark detection procedure comprises the following steps, refer to Fig. 5.4:

1. Get the LPM of the Fourier transform of the watermarked image having undergone geometrical attacks.

2. Match the template in the LPM domain of the watermarked image, by using the phase-only filtering method described in Chapter 4, to calculate the rotation and scaling parameters according to Equ. (4.17) and Equ. (4.18). The watermarked image is then geometrically re-synchronized by using these two parameters.

3. Generate the two-dimensional watermark pattern as in the embedding procedure.

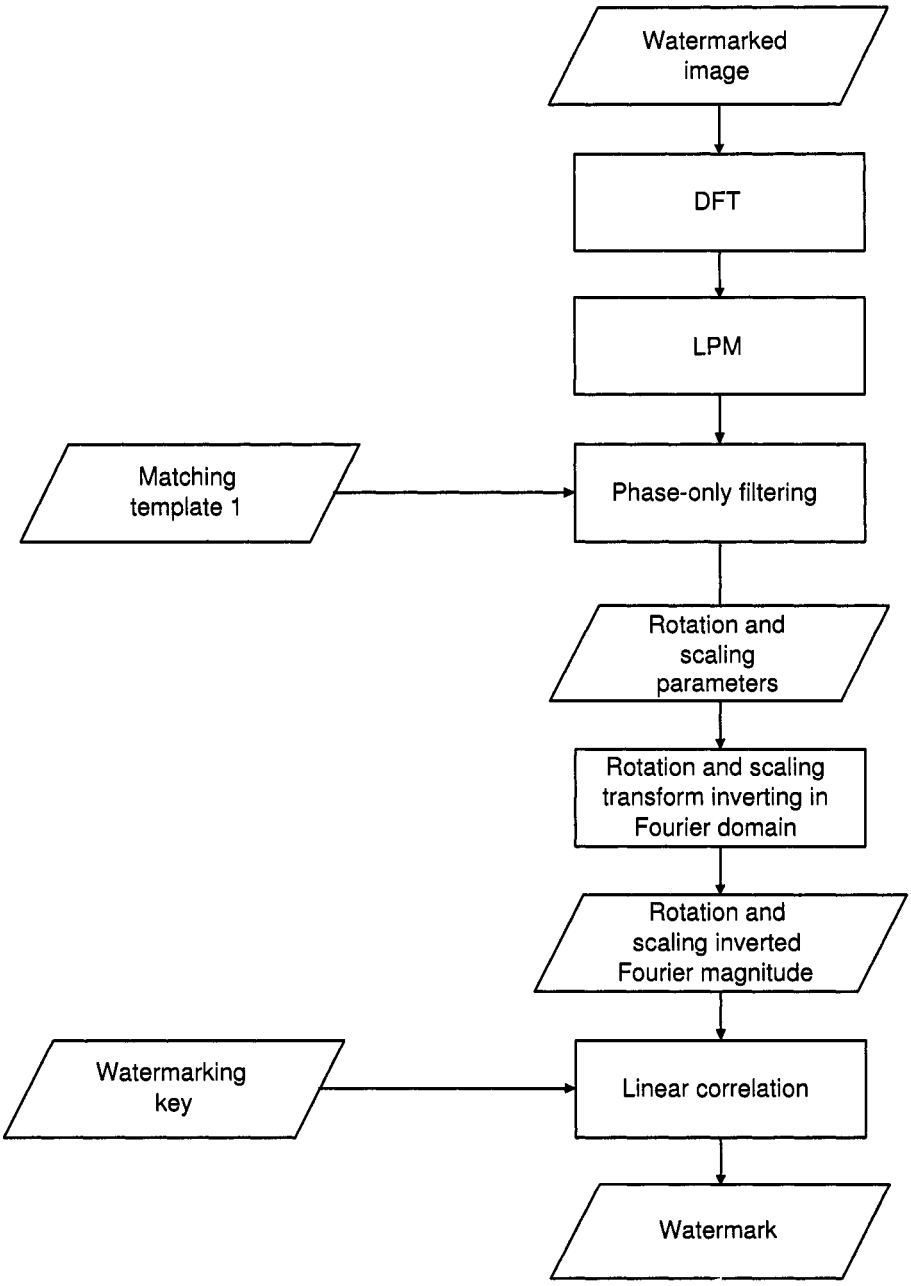


Figure 5.4: Scheme II: watermark detection.

4. The correlation value *cor* between the watermarked image after re-synchronization and the watermark pattern is calculated by using:

$$cor = \frac{1}{MN}\sum_i\sum_j (X(i,j) - \overline{X}(i,j))W(i,j) \tag{5.6}$$

with

$$\overline{X}(i,j) = \frac{1}{MN}\sum_i\sum_j X(i,j) \tag{5.7}$$

where $M \times N$ is the size of the image, $X(i,j)$ is the watermarked image, $\overline{X}(i,j)$ is the mean of the $X(i,j)$, and $W(i,j)$ is the watermark pattern.

5. If the correlation value *cor* is greater than a predefined threshold, the watermark exists. Otherwise, it does not.

5.3 Scheme III: embedding watermark in LPM domain

In this scheme, we embed the watermark sequence into the log-polar mapping of an image. The advantage of this scheme is that we rectify the watermark position directly in the LPM domain of the image instead of the geometrical rectification of the image in the spatial domain to avoid the imprecision introduced by the interpolation during the log-polar mapping.

5.3.1 Watermark embedding

The procedure of embedding a watermark consists of the following steps (refer to Fig. 5.5):

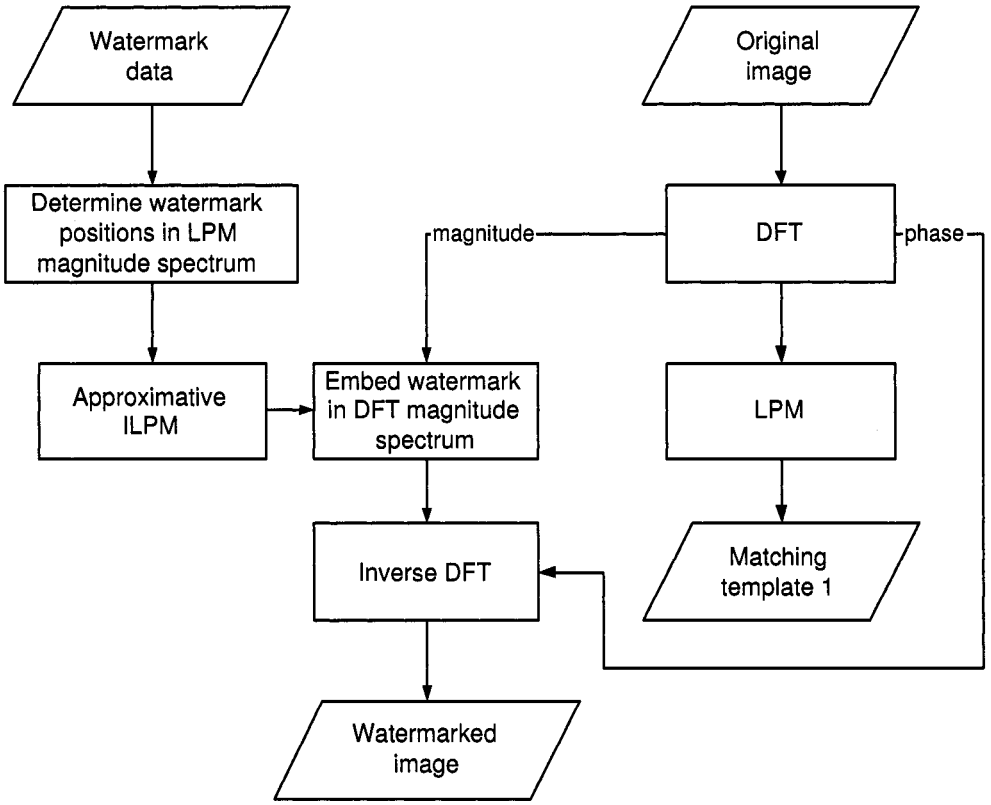


Figure 5.5: Scheme III: watermark embedding.

1. First, use a PN 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 the DFT is positive, so we need to extend the watermark data sequence to its double length, i.e. encode positive numbers x as $(x, 0)$ and negative numbers x as $(0, x)$.

3. Save a block portion of LPM spectrum as the matching template g, which will be available to the watermark extraction process.

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

5. If we embed the 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 approximate ILPM and embed the 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 4.

   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 [25].

6. Most algorithms use a simple embedding equation, refer to Equ. (5.8), mentioned in embedding process of Application II item 5.

   According to our experiments, by carefully selecting the watermarking positions and watermarking factor β, the difference between E and $\beta * W$ can be small enough. So we can use Equ. (5.8) to replace the values of the embedding points by $\beta * W$.

$$E' = \beta * W \tag{5.8}$$

   This embedding process will not change the magnitude 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 magnitude changes by using the correlation coefficient, which is in the range of -1 to 1, independent of scale changes in the magnitude (refer to Equ. (5.12)).

7. 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.

To get the tradeoff between invisibility and robustness, we choose the middle frequency components as the location to insert watermark data. Because the log-polar mapping is just like a sampling process, the closer to the center, the higher the sampling rates. 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. Therefore, in our watermarking scheme, we insert the watermark data into the middle frequency components. Experiments show the effectiveness of this approach.

5.3.2 Watermark detection

To the watermark extraction process, available are the watermarked image that may or may not suffered from attacks, the watermarking key, the original watermark positions, and the matching template g. The procedure of extracting the watermark consists of following steps (refer to Fig. 5.6):

1. Apply DFT and LPM to the watermarked image, and transform it into LPM domain.

2. Calculate the correlation between the LPM domain of the watermarked image and the matching template g, by using the new filtering method, according to Equ. (4.10). For details, refer to Chapter 4.

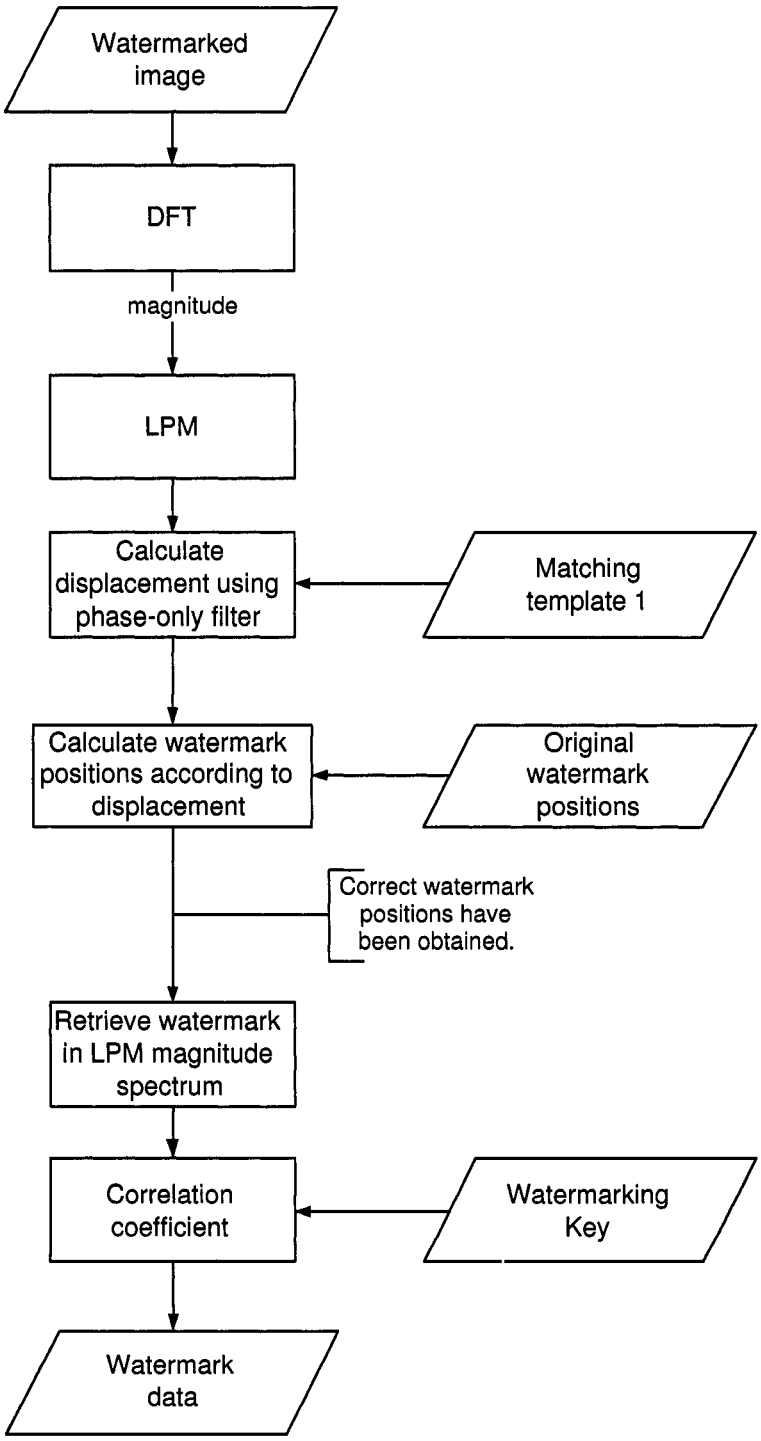


Figure 5.6: Scheme III: watermark detection.

3. Find the peak in the calculated correlation, and find the translations according to Equ. (4.17). Suppose the coordinates of the original watermark position for w_i is $(\rho_{w_i}, \theta_{w_i})$, then we can calculate the coordinates of the new watermark position in the watermarked image having undergone attacks by using the following equations:

$$\rho_{w_i}{}' = \rho_{w_i} + \Delta_\rho \tag{5.9}$$

$$\theta_{w_i}{}' = \theta_{w_i} + \Delta_\theta \tag{5.10}$$

4. Retrieve the watermark data V at the rectified location by using the following equation:

$$V = \frac{E''}{\beta} \tag{5.11}$$

where E'' is the DFT magnitude spectrum of the watermarked image that undergone RST transformations and other attacks, while β is defined in Equ. (5.8). The value change of V caused by a scaling transformation is proportional to the scale factor, and the normalized correlation calculated by Equ. (5.12) is independent of the change.

5. By using Equ. (5.12), 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.

$$sim = \frac{W \times V^T}{\sqrt{(W \times W^T)(V \times V^T)}} \tag{5.12}$$

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

In this chapter, we introduced three RST invariant image watermarking schemes as the applications of our rectification algorithm. In order to show that our rectification algorithm can be used by any RST invariant watermarking schemes, we embed the watermark into three different domains: the spatial domain, Fourier domain, and LPM domain, respectively. In the next chapter, we will evaluate the proposed rectification scheme and give the experimental results for the three application schemes.

Chapter 6

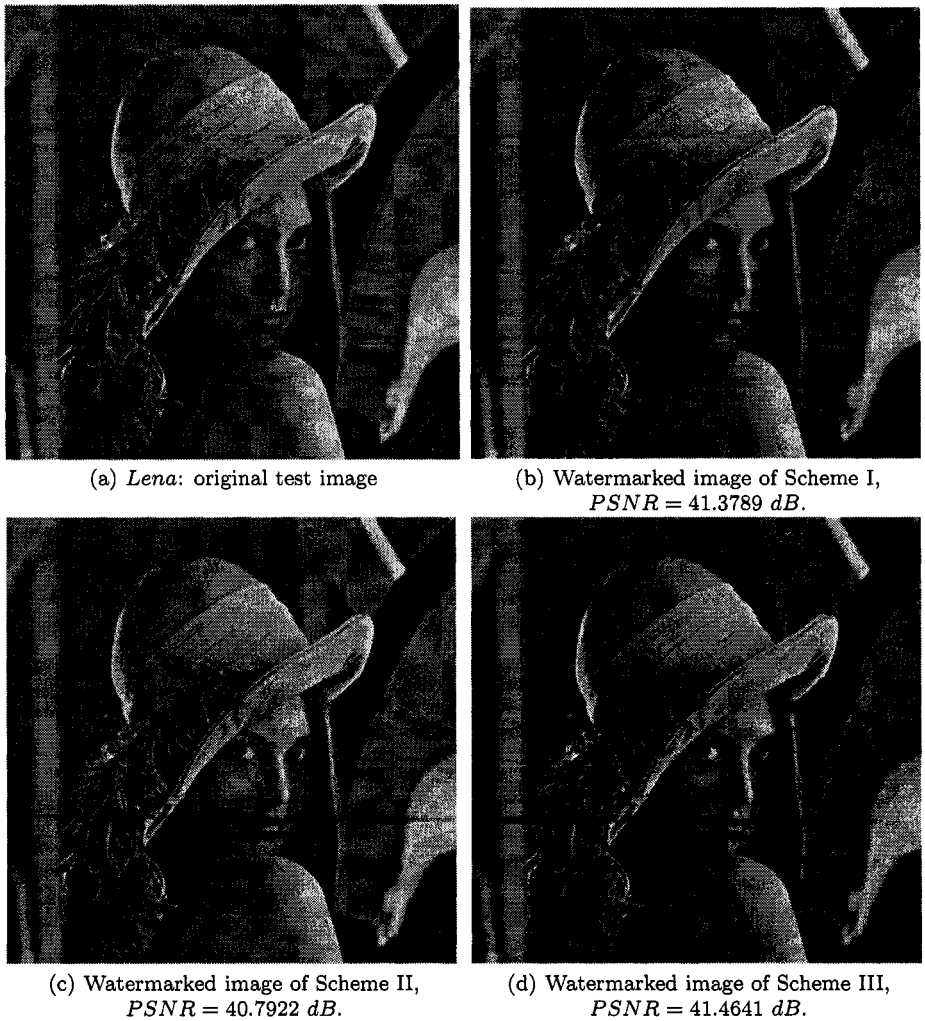
Experimental results and evaluation

In this chapter, we give the experimental results and evaluation. First, we evaluate our rectification scheme, by comparing performance to other filters, and by giving detection accuracies for RST parameters. Second, we illustrate the performance of the proposed scheme. Finally, we implement the three schemes discussed in Chapter 5 and give the experimental results under different attacks. In our experiments, unless otherwise stated, we use the image *Lena* (Fig. 6.1 (a)) as our test image, the size of which is 512×512.

6.1 Performance of filters

Table 6.1 shows the performance of the different filters measured by the three criteria, where, "New", "POF", "BPOF", "Classical", "AOF", and "Inverse", respectively, stand for our new filtering method, the phase-only filter, the binary phase-only filter, the classical filter, the amplitude-only filter and the inverse filter. In our case, the new filtering method and the phase-only filter have the highest SNR no matter whether ro-

tation or scaling transformation or both are applied to the watermarked image or not, while the other filters are less robust to noise. The new filtering method has the highest PCE among all the filters. Meanwhile, the classical matched filter and the amplitude-only filter have poor SNR and PCE. The new filtering method, the phase-only filter and the binary phase-only filter have 100% efficiency, while the classical matched filter and the ampitude-only filter have efficiencies of only 58%. The Horner efficiency of the inverse filter is even as low as 10^{-5}.



(a) *Lena*: original test image

(b) Watermarked image of Scheme I, $PSNR = 41.3789\ dB$.

(c) Watermarked image of Scheme II, $PSNR = 40.7922\ dB$.

(d) Watermarked image of Scheme III, $PSNR = 41.4641\ dB$.

Figure 6.1: The original image and watermarked images.

Table 6.1: The performance of the filters measured by the three criteria

Filters	SNR (dB)		PCE (dB)		Horner efficiency (%)	
	No RS	RS	No RS	RS	No RS	RS
New	44.04	37.86	44.09	31.34	100	100
POF	44.04	37.86	34.09	25.39	100	100
BPOF	40.09	36.32	30.20	23.46	100	100
Classical	29.32	32.47	16.17	16.11	58.09	51.66
AOF	28.39	31.07	15.22	15.39	58.09	51.66
Inverse	32.54	22.45	29.23	20.02	2.65×10^{-5}	1.04×10^{-5}

6.2 Detection accuracy

In this section, we describe the detection accuracy of this rectification method. The following detection accuracy is computed according to scheme II. However, the detection methods for all three schemes are the same. The different watermarking domains do not influence the detection accuracy significantly. The maximum difference among different schemes could be one pixel in terms of detected shift for RST parameters. Therefore, the detection accuracy for rotation, scaling and translation is almost the same for the three schemes.

6.2.1 Rotation degree

To increase the computation precision, we set the sampling numbers along the θ axis to be 1024, which means the size of the power spectrum of an image in the LPM domain is 512×1024. In our case, a shift of one pixel along the θ axis corresponds to the rotation angle 0.3516° in the spatial domain according to Equ. (4.18). The imprecision of the rotation angles is the absolute value of the real rotation angles α subtracting the calculated rotation angle α', shown in the right column of Table 6.2. According to our experimental results, the imprecision is almost always below 0.5°, shown in Table 6.2.

Table 6.2: Detection accuracy for rotation

α	Detected shift in pixels	α'	$\lvert\alpha' - \alpha\rvert$
0.5°	2	0.7031°	0.2031°
1°	3	1.0547°	0.0547°
1.5°	4	1.4063°	0.0938°
2°	6	2.1094°	0.1094°
5°	14	4.9219°	0.0781°
5.5°	16	5.6250°	0.1250°
10°	29	10.1953°	0.1953°
10.5°	30	10.5469°	0.0469°
20°	57	20.0391°	0.0391°
30°	86	30.2344°	0.2344°
40°	115	40.4297°	0.4297°
45°	129	45.3516°	0.3516°
50°	143	50.2734°	0.2734°
60°	172	60.4688°	0.4688°
70°	200	70.3125°	0.3125°
80°	228	80.1563°	0.1563°
90°	256	90°	0°
90.5°	258	90.7031°	0.2031°
100°	285	100.1953°	0.1953°
110°	313	110.0391°	0.0391°
120°	341	119.8828°	0.1172°
130°	369	129.7266°	0.2734°
140°	398	139.9219°	0.0781°
150°	425	149.4141°	0.5859°
160°	454	159.6094°	0.3906°
170°	483	169.8047°	0.1953°
180°	512	180°	0°
180.5°	513	180.3516°	0.1484°

6.2.2 Scaling factor

According to Equ. (4.18), the shift of one pixel along the ρ axis corresponds to the scaling ratio 1.0116 in the spatial domain. The imprecision of the scaling ratio is the absolute value of the real scaling ratio σ subtracting the calculated scaling ratio σ', shown in the right column of Table 6.3. According to our experimental results, the imprecision of the scaling ratio is always below 0.01, shown in Table 6.3.

6.2.3 Translation

In the spatial domain, the detected translation parameters are very accurate when shifting the image by integer pixels. If shifting the image by non-integer pixels, the

Table 6.3: Detection accuracy for scaling

σ	Detected shift in pixels	σ'	$\lvert\sigma' - \sigma\rvert$
0.6	-45	0.5958	0.0042
0.7	-31	0.7000	0
0.8	-20	0.7944	0.0056
0.9	-9	0.9016	0.0016
1.0	0	1.0	0
1.1	8	1.0964	0.0036
1.2	16	1.2022	0.0022
1.3	23	1.3033	0.003

imprecision of the translation factor is at most 0.5 according to our experiments.

In RST invariant image watermarking, the detection imprecision of the RST parameters can be tolerated by using a search in a small neighborhood of the detected parameters.

6.3 Illustrations

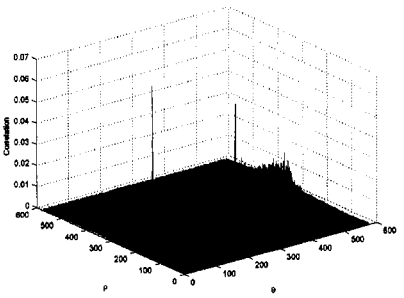
In this section, we illustrate our proposed scheme against rotation, scaling and translation transformations. We use the watermarked image *Lena* (Fig. 6.1 (d)) in scheme III as an example, the $PSNR$ of which is 41.4641 dB.

6.3.1 Rotation with cropping

Fig. 6.2 shows the correlation for watermarked image *Lena* under different transformations. Fig. 6.2 (a) is transformed from the watermarked image by rotating the watermarked image by 45 ° counter-clockwise without scaling. 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.

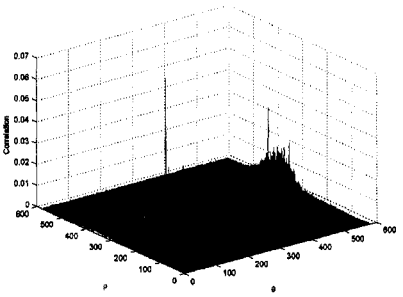
For comparison purposes, the translation caused by the rotation is calculated as $(-65, 0)$ by using the log-polar mapping equations, where -65 and 0 are, respectively,

(a) Rotated by 45 degree counter-clockwise without scaling.



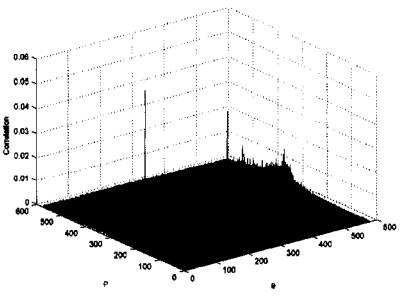
(b) Correlation between LPM of (a) and the matching template. The peak is at $(185, 371)$.

(c) Scaled by 0.7.



(d) Correlation between LPM of (c) and the matching template. The peak is at $(250, 401)$.

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



(f) Correlation between LPM of (e) and the matching template. The peak is at $(186, 401)$.

Figure 6.2: Correlation for watermarked images having undergone different transformations.

the number of row (θ) and number of column (ρ) of the translation. $(-65, 0)$ here means that the log-polar radius ρ in LPM of the watermarked image has not been changed, while the angle θ in LPM of the watermarked image has been cyclically shifted 65 columns left (rotated 45 ° counter-clockwise).

We applied DFT and LPM to the rotated watermarked image (Fig. 6.2 (a)), then obtained the correlation spectrum between the matching template g and the LPM of the watermarked, shown as Fig. 6.2 (b). There is one peak in the correlation spectrum. The peak is detected at $(185, 371)$. The original embedding position is $(250, 370)$. So the translation caused by the rotation is calculated as $(-65, 1)$. We use the translation to calculate the watermarking positions.

6.3.2 Scaling without rotation

Fig. 6.2 (c) is transformed from the watermarked image, by scaling the watermarked image by 0.7. The translation caused by the transformation is calculated as $(0, 30)$ by using the log-polar mapping equations, which means that the log-polar radius ρ in LPM of the watermarked image has been cyclically shifted 30 columns up (scaled by 0.7), while the angle θ in LPM of the watermarked image has not been changed.

We applied DFT and LPM to the rotated watermarked image (Fig. 6.2 (c)), then obtained the correlation spectrum between the matching template g and the LPM of the watermarked, shown as Fig. 6.2 (d). There is one peak in the correlation spectrum. The peak is detected at $(250, 401)$. Then we calculate the translation as $(0, 31)$, which then is used to calculate the watermarking positions.

6.3.3 Scaling and rotation

Fig. 6.2 (e) is transformed from the watermarked image, by rotating the watermarked image by 45° counter-clockwise after scaling by 0.7062. The translation caused by the transformation is calculated as $(-65, 30)$ by using the log-polar mapping equations, which means that the log-polar radius ρ in LPM of the watermarked image has been cyclically shifted 30 columns up (scaled by 0.7062), while the angle θ in LPM of the watermarked image has been cyclically shifted 65 columns left (rotated 45° counter-clockwise).

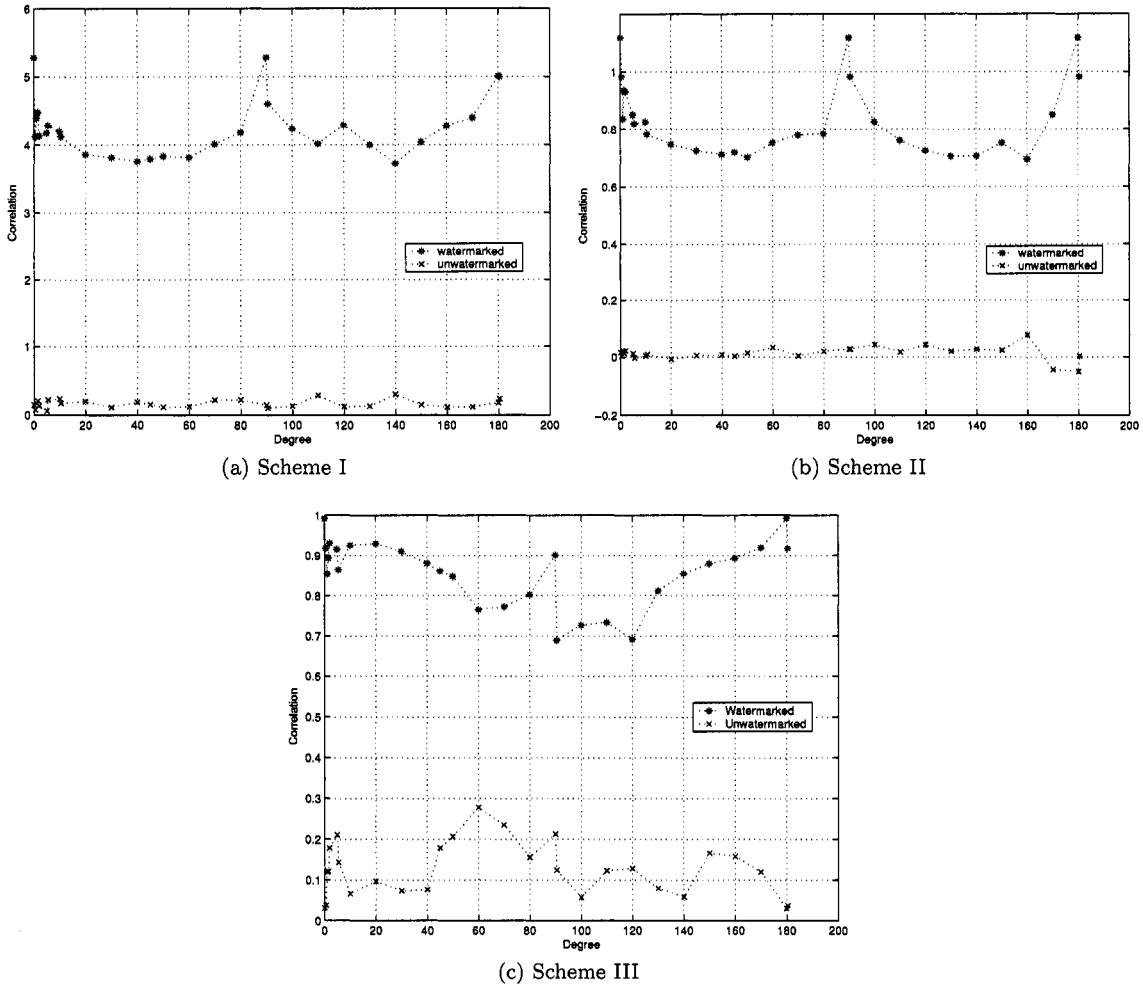
We applied DFT and LPM to the rotated watermarked image (Fig. 6.2 (e)), then obtained the correlation spectrum between the matching template g and the LPM of the watermarked, shown as Fig. 6.2 (f). There is one peak in the correlation spectrum. The peak is detected at $(186, 401)$. We calculate the translation as $(-64, 31)$, which then is used to calculate the watermarking positions.

6.4 Experimental results for the three application schemes

In this section, we show the experimental results for the three schemes under different attacks. For these three schemes, we embed watermark in such a strength that the distortion is just imperceptible. The PSNR of the watermarked images are 41.3789 dB, 40.7922 dB and 41.4641 dB for these three schemes, respectively, shown in Fig. 6.1.

All the following experiments will be conducted on these watermarked images. We also have done experiments on other test images, and similar results have been achieved. Test results will show that the scheme can meet the requirements of both impercepti-

bility and robustness.



(a) Scheme I

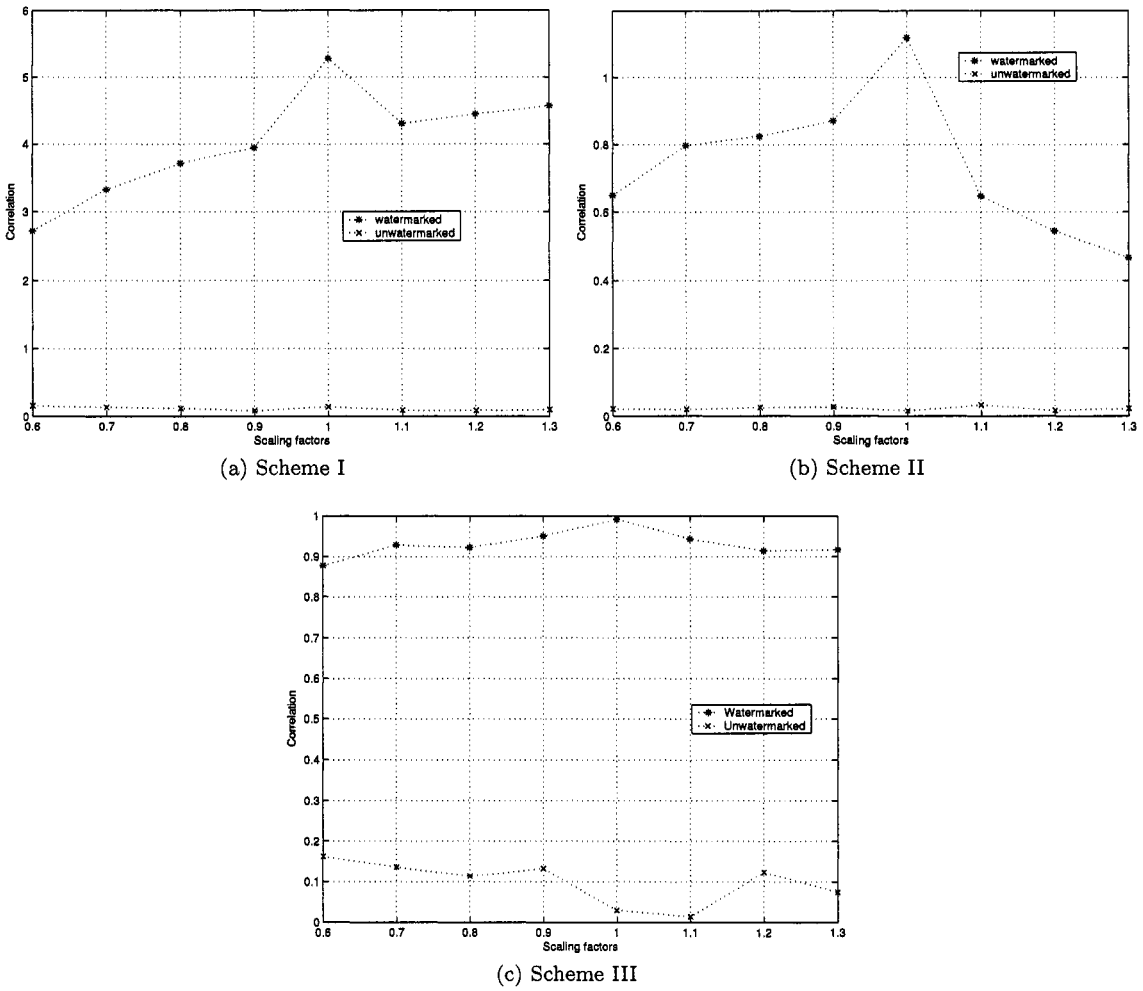
(b) Scheme II

(c) Scheme III

Figure 6.3: Rotation with cropping.

6.4.1 Rotation with cropping

We rotate the watermarked images by different angles. The test results in Fig. 6.3 show the correlation of three schemes under different rotation angles. In Fig. 6.3 - 6.7, the curve "Watermarked" is the cross-correlation between the original watermark data and the watermark data detected from the watermarked image under different attacks,



(a) Scheme I

(b) Scheme II

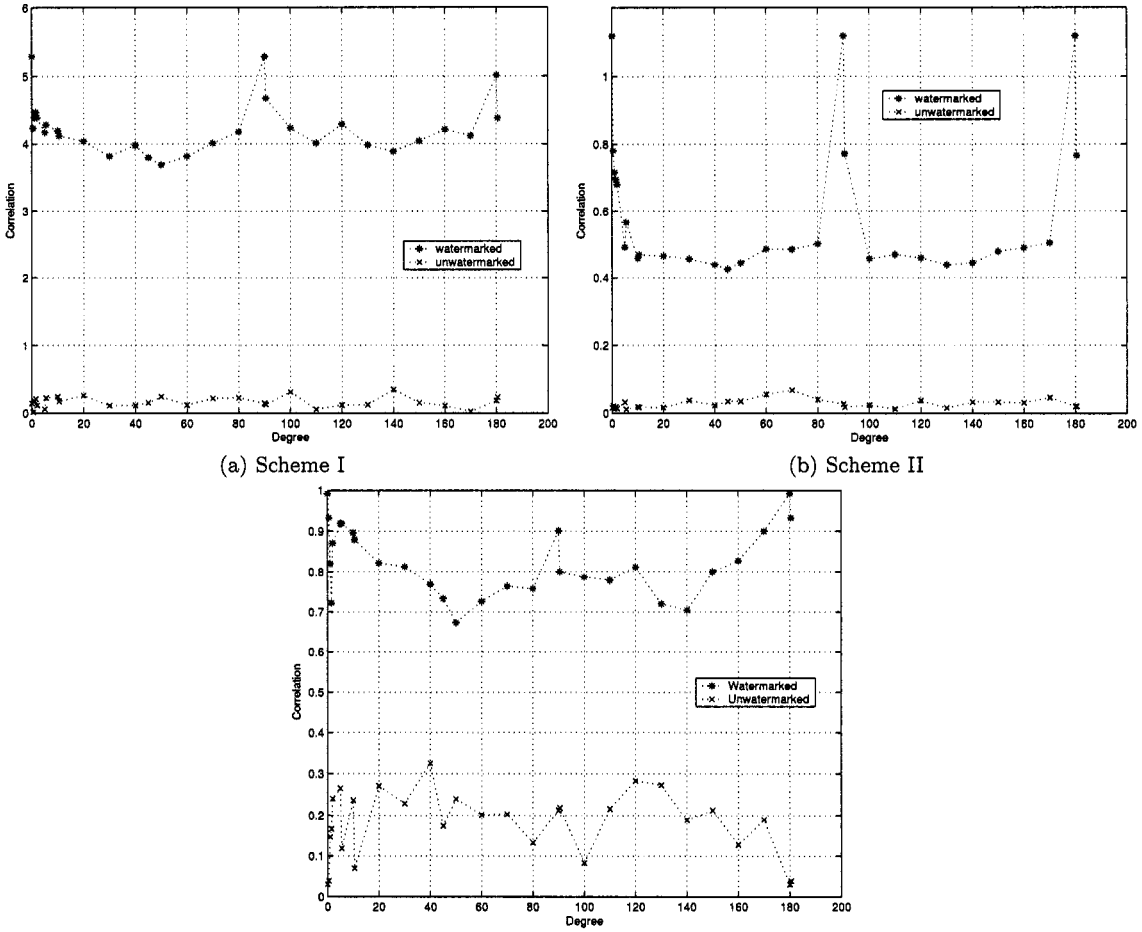
(c) Scheme III

Figure 6.4: Scaling without rotation.

and the curve "Unwatermarked" is the cross-correlation between the original watermark data and "the watermark data" detected from the corresponding unwatermarked image that undergone the same transformations as the watermarked image.

In Fig. 6.3, the horizontal axis contains angles we rotated the watermarked image before detecting the watermark. We only showed the results for rotation angles up to 180.5°, since

$$Correlation(180° + \alpha) = Correlation(\alpha) \tag{6.1}$$



(a) Scheme I

(b) Scheme II

(c) Scheme III

Figure 6.5: Scaling and rotation.

The correlation for the watermarked image and the unwatermarked image can successfully separated according to Fig. 6.3. The algorithms can handle fractional degree rotation. When the rotational degree becomes bigger, larger areas need to be cropped and more black pixels need to be padded, therefore the correlation for watermarked image tends to be smaller. When an image is rotated by 90° or 180°, no information is lost and no black pixels are needed. Therefore, a peak value is got for these rotation angles. Another reason for this is that the detection accuracy is perfect with these rotation angles, as shown in Table 6.2.

We have done similar tests to other standard test images, and we obtained similar results. Therefore, we can conclude that the scheme schemes are robust to rotation with any angles.

6.4.2 Scaling without rotation

We scaled the image by using the scaling factors shown on the horizontal axis of Fig. 6.4. The clearly separated curves, which represent the correlation between the original watermark data and detected one from watermarked and unwatermarked image, respectively, illustrate that the three schemes are invariant to scaling without rotation.

6.4.3 Scaling and rotation

Fig. 6.5 shows the test results of scaling and rotation for the three schemes. In this test, there is no cropping to the image because the image is shrunk accordingly before being rotated. Black pixels are padded to maintain the image size and shape. The horizontal axis contains the rotation angles and the vertical axis represents the correlation values. The two curves on the figures are separated very well. The reason for the peaks at rotation angles of 90° and 180° has been explained in Section 6.4.1. It illustrates that all the three schemes are robust to the scaling and rotation.

6.4.4 Scaling and translation

In this experiment, we scale the watermarked image by 70%, and then translate it by 50 pixels in different directions. Black pixels are padded to maintain the image size. The test results are shown in Table 6.4. In Table 6.4, 6.6 and 6.5, *Cor.*1 is the cross-correlation between the original watermark data and the watermark data detected from

Table 6.4: Scaling by 0.7 and translation

Translation	Scheme I		Scheme II		Scheme III	
	Cor.1	Cor.2	Cor.1	Cor.2	Cor.1	Cor.2
No translation	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352
up	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352
down	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352
left	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352
right	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352
right & up	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352
right & down	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352
left & up	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352
left & down	3.3253	0.1363	0.7177	0.0202	0.9282	0.1352

the watermarked image, and $Cor.2$ is the cross-correlation between the original watermark data and the watermark data detected from the corresponding unwatermarked image that undergone the same transformations as the watermarked image. We can see that the correlation results for the three schemes are constant. The schemes are invariant to translation.

Table 6.5: Gaussian noise pollution

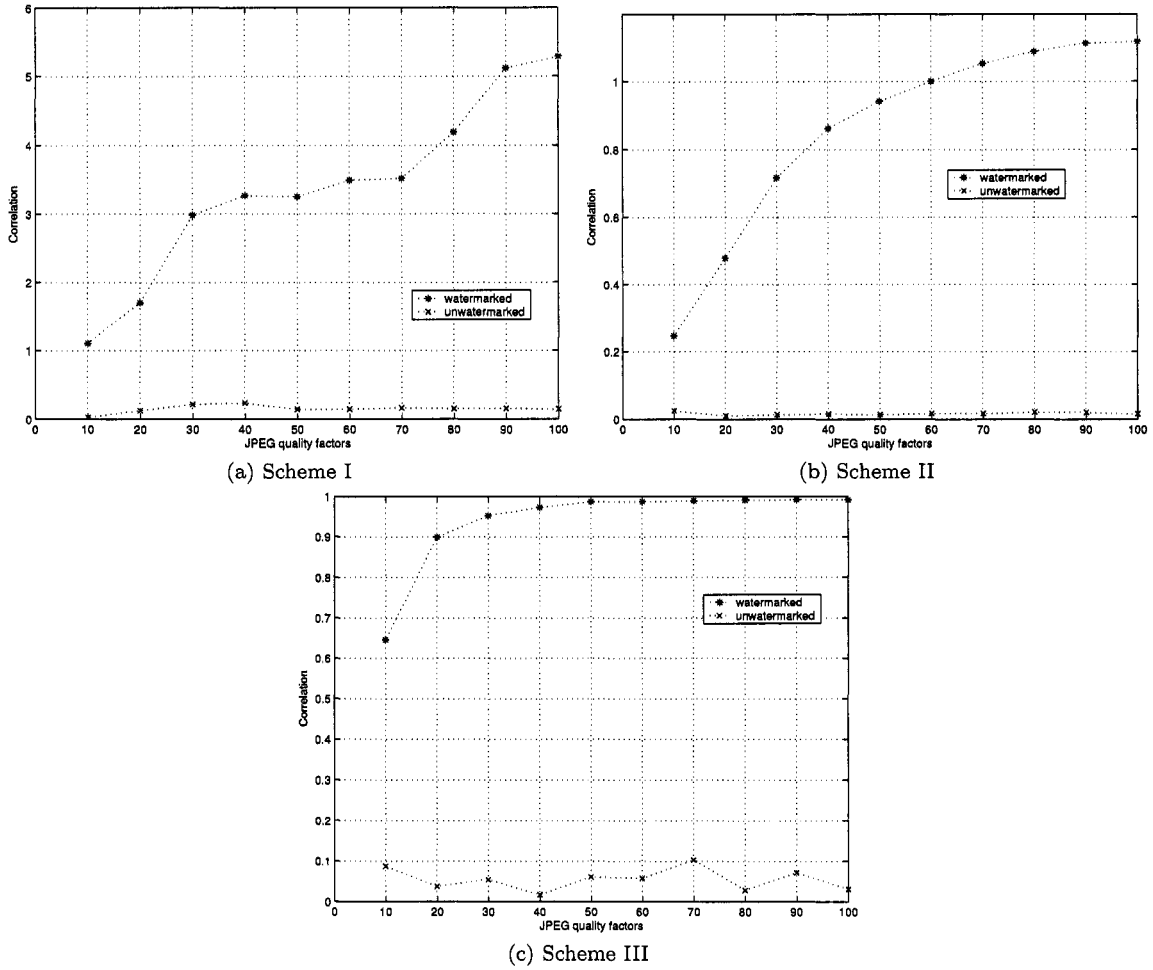
Variance	Scheme I		Scheme II		Scheme III	
	Cor.1	Cor.2	Cor.1	Cor.2	Cor.1	Cor.2
0.001	5.7378	0.1255	0.9747	0.0242	0.9844	0.1491
0.01	5.6323	0.1763	0.6286	0.0428	0.8804	0.1533

Table 6.6: Salt-and-pepper noise pollution

Noise density	Scheme I		Scheme II		Scheme III	
	Cor.1	Cor.2	Cor.1	Cor.2	Cor.1	Cor.2
0.01	5.2223	0.1179	0.8431	0.0122	0.9744	0
0.1	5.1579	0.2503	0.3634	0.0463	0.9692	0.0638

6.4.5 JPEG compression

We compress the watermarked image by different quality factors, the test results for the three schemes are shown in Fig. 6.6. The results are very good for all of them until



(a) Scheme I

(b) Scheme II

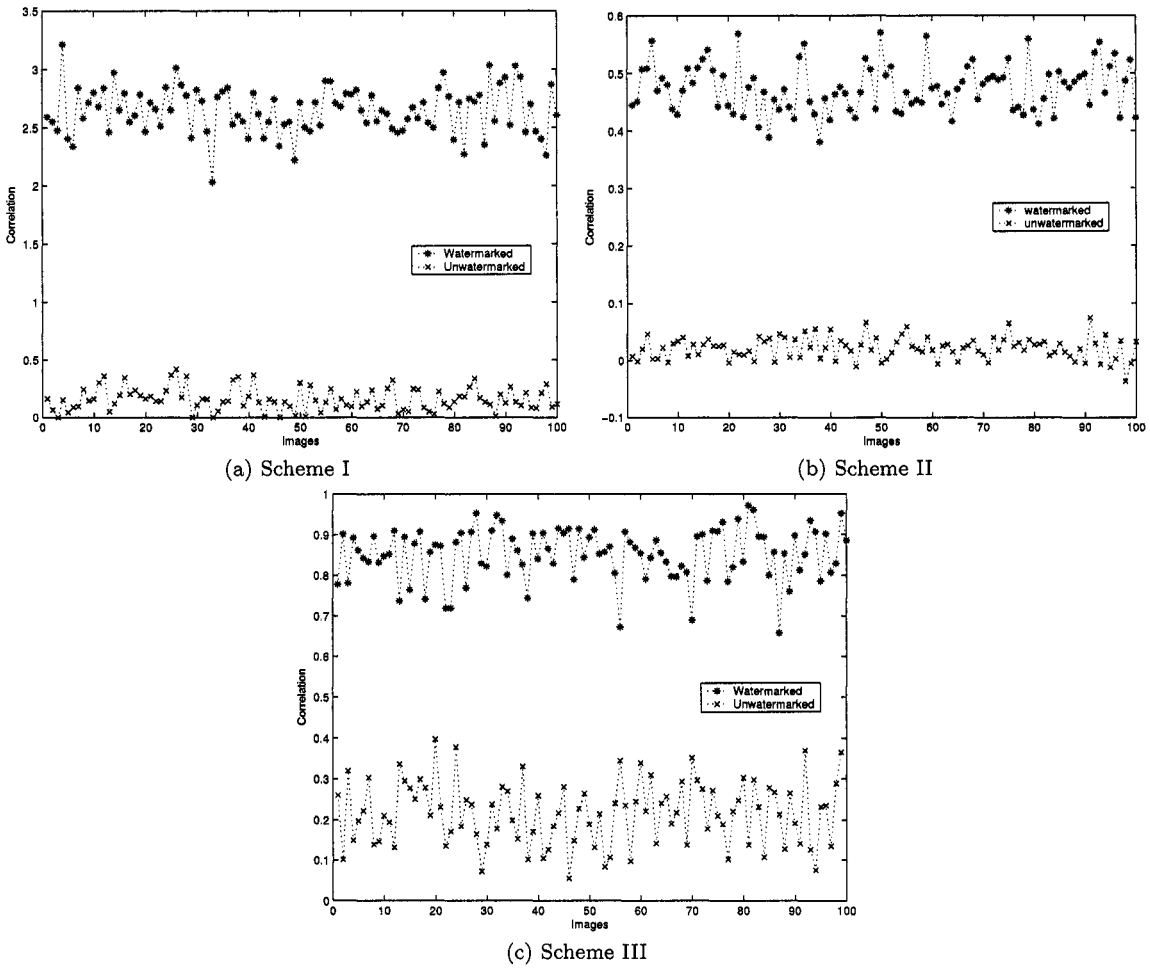
(c) Scheme III

Figure 6.6: JPEG compression.

the quality factor reaches 10%.

6.4.6 Noise addition

The watermarking schemes are tested with Gaussian noise pollution and salt-and-pepper noise pollution. The test results are satisfactory, as shown in Table 6.5 and Table 6.6. The experiments show the robustness of the proposed watermarking schemes.



(a) Scheme I

(b) Scheme II

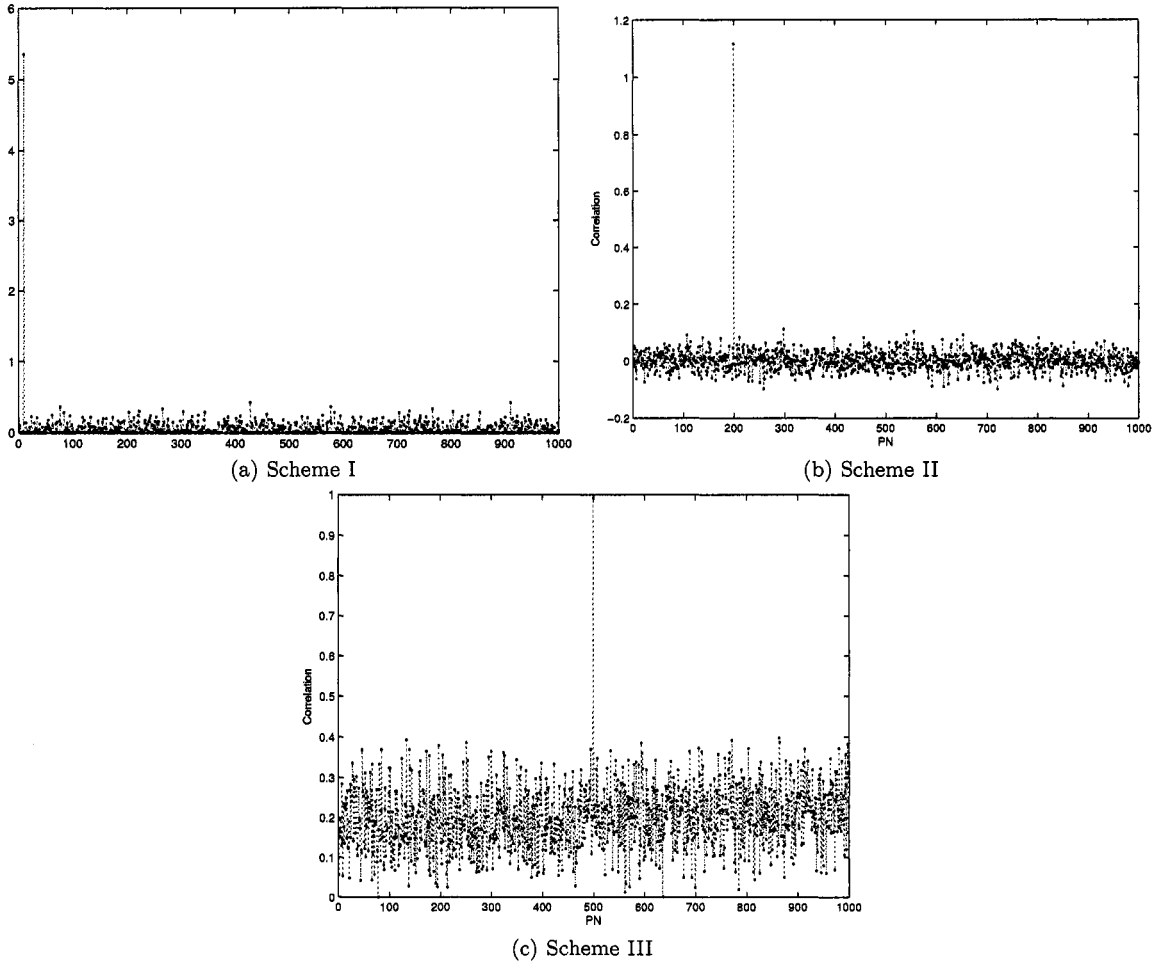
(c) Scheme III

Figure 6.7: Watermark detection results for 100 test images.

6.4.7 Performance on different images

We applied our watermarking schemes to 100 test images downloaded from the internet. The image set is chosen to contain a variety of natural images. The images are 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. 6.7 (a), (b), and (c) for these three schemes, respectively. The horizontal axis shows different images, while the vertical axis shows the corresponding correlation.

The three schemes could still detect correctly from all the watermarked images without false positive from unwatermarked images despite the quite severe conditions.



(a) Scheme I

(b) Scheme II

(c) Scheme III

Figure 6.8: Watermark detection results for 1000 PN sequence including the one originally embedded.

6.4.8 Random watermark test

We have conducted random-watermark false positive tests in order to demonstrate the ability of our watermark extraction algorithms. We randomly generate 1000 PN sequences during the watermark detection process. The experimental results show that

only the original embedded one could obtain the acceptable correlation value, refer to Fig. 6.8.

In this chapter, we evaluated the proposed rectification algorithm. The proposed new filtering method always has the best performance according to the three comparison criteria. The test results show that our rectification algorithm can detect the RST parameters very accurately. The three RST invariant watermarking schemes based on our rectification algorithm can successfully detect the RST parameters and are very robust against RST attacks.

Chapter 7

Conclusions and future work

This thesis focuses on RST invariant image watermarking. We introduced the fundamental theories and techniques that are necessary for RST invariant image watermarking, reviewed the current algorithms and divided them into four main categories. Then, we proposed a rectification scheme for rotation, scaling and translation invariant image watermarking. This scheme belongs to RST invariant domain based approach. It is based on log-polar mapping and a new filtering method to get the geometric transformation parameters. Geometrically transformed images can be re-synchronized. Matching results and performance comparisons of the different filters showed that the new filtering method is the only one that can produce acceptable discrimination. Detection accuracy tests confirmed that the algorithm is quite accurate to estimate the rotation, scaling, and translation parameters.

This RST rectification scheme can be used in any image watermarking algorithms that provide robustness to RST transformations. We also introduced three schemes of the rectification scheme, in which the watermark is embedded in the spatial domain, in the magnitude of the Fourier transform domain, and in the log-polar mapping of the

Fourier transform of an image respectively. Experimental results demonstrated that all these three schemes are invariant to rotation, scaling and translation, and very robust to JPEG compression and noise pollution attacks.

As for our future work, we will improve our rectification algorithm to further lower the cost and increase the detection accuracy. We will try to use the feature of an image itself as the template for our rectification algorithm to avoid the cost. More advanced method will be introduced to deal with the local distortion, shearing distortion and other affine transformations.

Bibliography

[1] S. Katzenbeisser and F. Petitcolas, *Information hiding techniques for steganography and digital watermarking.* ISBN: 1-58053-035-4, Boston : Artech House, 2000.

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

[3] M. Miller and J. Bloom, "Computing the probability of false watermark detection," in *Proc. of The Third International Workshop on Information Hiding*, pp. 146–158, 1999.

[4] M. Arnold, M. Schmucker, and S. D. Wolthusen, *Techniques and applications of digital watermarking and content protection.* ISBN 1-58053-111-3, Boston : Artech House, 2003.

[5] Stirmark, "http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark."

[6] Checkmark, "http://www.checkmark.com/."

[7] Certmark, "http://www.certimark.org/."

[8] Y. Liu and J. Zhao, "A rectification scheme for rst invariant image watermarking," in *Proc. of Canadian Conference on Electrical and Computer Engineering*, pp. 527–530, 2004.

[9] Y. Liu and J. Zhao, "Rotation, scaling, and translation invariant image watermarking based on radon transform," in *Proc. of Canadian Conference on Computer and Robot Vision*, pp. 225–232, 2004.

[10] Y. Liu and J. Zhao, "A new filtering method for rst invariant image watermarking," in *Proc. of IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications*, pp. 101–106, 2003.

[11] D. Zheng, Y. Liu, and J. Zhao, "Rst invariant digital image watermarking based on a new phase-only filtering method," in *Proc. of 7th International Conference on Signal Processing*, 2004.

[12] D. Hearn and M. Baker, *Computer graphics.* ISBN:0-13-530924-7, Prentice Hall, 1997.

[13] R. Gonzalez and R. Woods, *Digital image processing.* ISBN: 0-20-118075-8, Prentice Hall, 2002.

[14] J. Horner and P. Gianino, "Phase-only matched filter," *Applied Optics*, vol. 23, no. 6, pp. 812–816, 1984.

[15] J. O'Ruanaidh and T. Pun, "Rotation, scale, and translation invariant digital image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 303–317, 1998.

[16] 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.

[17] 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*, vol. 13, pp. 753–765, 2003.

[18] R. Bracewell, *The Fourier transform and its applications.* ISBN: 0073039381, Boston : McGraw Hill, 2000.

[19] H. Kim, Y. Baek, H. Lee, and Y. Suh, "Watermark using radon transform and bispectrum invariants," in *IH 2002, LNCS 2578*, pp. 145–159, 2003.

[20] Y. Hsu and H. Arsenault, "Optical pattern recognition using circular harmonic expansion," *Applied Optics*, vol. 21, pp. 4012–4015, 1982.

[21] J. Yao and G. Lebreton, "Scale-invariant correlation with truncated phase-only radial harmonic fitlers," *Optics Communications*, vol. 145, pp. 213–219, 1998.

[22] J. Yao and L. Chin, "Power-adjusted fractional power radial harmonic filters for shift- and scale-invariant pattern recognition with improved noise robustness and discrimination," *Optics Communications*, vol. 162, pp. 26–30, 1998.

[23] H. Kim and B. Kumar, "Rotation-tolerant watermark detection using circular harmonic function correlation filter," *Digital Watermarking, LNCS 2939*, pp. 263–276, 2004.

[24] M. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. IT-8, no. 8, pp. 1409–1420, 1962.

[25] D. Zheng and J. Zhao, "Rst invariant digital image watermarking: importance of phase information," in *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 785–788, 2003.

[26] C. Kuglin and D. Hines, "The phase correlation image alignment method," in *Proc. of IEEE 1975 International conference on cybernetrics and society (Sept.)*, pp. 163–165, 1975.

[27] C. Harris and M. Stephen, "A combined corner and edge detector," in *Proc. of 4th Alvey Vision Conference*, pp. 147–151, 1988.

[28] D. Simitopoulos, D. Koutsonanos, and M. G. Strintzis, "Image watermarking resistant to geometric attacks using generalized radon transformations," in *Proc. of DSP 2002*, vol. 1, pp. 85–88, 2002.

[29] 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.

[30] M. Alghoniemy and A. Tewfik, "Progressive quantized projection watermarking scheme," in *Proc. of ACM Multimedia 99*, vol. 1, pp. 295–298, 1999.

[31] D. Zheng and J. Zhao, "Lpm-based rst invariant digital image watermarking," in *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1951–1954, 2003.

[32] B. Kumar, A. Mahalanobis, and A. Takessian, "Optimal tradeoff circular harmonic function correlation filter methods providing controlled in-plane rotation response," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1025–1034, 2000.

[33] M. Maes, T. Kalker, J. Linnartz, J. Talstra, G. Depovere, and J. Haitsma, "Digital watermarking for dvd video copy protection," *IEEE Signal Processing Magazine*, pp. 47–57, 2000.

[34] G. Depovere, T. Kalker, and J. Linnartz, "Improved watermark detection using filtering before correlation," in *Proc. of IEEE International Conference on Image Processing*, vol. 1, pp. 430–434, 1998.

[35] S. Pereira, J. O'Ruanaidh, and F. Deguillaume, "Template based recovery of fourier-based watermarks using log-polar and log-log maps," in *Proc. of IEEE International Conference on Multimedia Computing and System*, vol. 1, pp. 870–874, 1999.

[36] S. Pereira, S. Voloshynovskiy, M. Madueo, S. Marchand-Maillet, and T. Pun, "Second generation benchmarking and application oriented evaluation," in *Proc. of Information Hiding Workshop III*, 2001.

[37] P. Bas, J. Chassery, and B. Marq, "Geometrically invariant watermarking using feature points," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1014–1028, 2002.

[38] M. Kutter, S. Bhattacharjee, and T. Ebrahimi, "Towards second generation watermarking schemes," in *Proc. of IEEE-ICIP'99*, vol. I, pp. 320–323, 1999.

[39] Z. Duric, N. Johnson, and S. Jajodia, "Recovering watermarks from images," in *Proc. of Informaion & Software Engineering Technical Report ISE-TR-99-04*, 1999.

[40] Q. Sun, J. Wu, and R. Deng, "Recovering modified watermarked image with reference to original image," in *Proc. SPIE*, pp. 415–424, 1999.

[41] M. Alghoniemy and A. Tewfik, "Geometric distortion correction in image watermarking," in *Proc. SPIE*, pp. 82–89, 2000.

[42] J. Dittmann, T. Fiebig, and R. Steinmetz, "New approach for transformation-invariant image and video watermarking in the spatial domain: self-spanning patterns (ssp)," in *Proc. SPIE*, pp. 176–186, 2000.

[43] M. Alghoniemy and A. H. Tewfik, "Geometric invariance in image watermarking," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 145–153, 2004.

[44] Y. Abu-Mostafa and D. Psaltis, "Recognitive aspects of moment invariants," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 698–706, 1984.

[45] A. Khotanzad and Y. Hong, "Invariant image recognition by zernike moments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 489–497, 1990.

[46] D. L. Flannery and J. L. Horner, "Fourier optical signal processors," *Proc. of the IEEE*, vol. 77, no. 10, 1989.

[47] B. Reddy and B. Chatterji, "An fft-based technique for translation, rotation and scale-invariant image registration," *IEEE Transactions on Image Processing*, vol. 5, no. 8, 1996.

[48] L. Hill and T. Vlachos, "On the estimation of global motion using phase correlation for broadcast applications," *IEE Proc. of Image Processing and Its Application*, vol. 465, no. 2, pp. 721–725, 1999.

[49] J. Linnartz, T. Kaler, G. Depovere, and R. Beuker, "A reliability model for the detection of electronic watermarks in digital images," in *Proc. of IEEE Fifth Symposium on Communication and Vehicular Technology*, pp. 202–209, 1997.

[50] G. Depovere, T. Kalker, and J. Linnartz, "Improved watermark detection using

filtering before correlation," in *Proc. of IEEE International Conference on Image Processing*, pp. 430–434, 1998.

[51] P. Refregier, "Optimal trade-off filters for noise robustness, sharpness of the correlation peak, and horner efficiency," *Optics Letters*, vol. 16, no. 11, pp. 829–831, 1991.

[52] P. Moulin, "A mathematical approach to watermarking and data hiding," in *ICASSP Tutorial*, 2002.