uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Dong Zheng

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

RST invariance of image watermarking algorithms and the framework of mathematical analysis

TITRE DE LA THÈSE / TITLE OF THESIS

Jiying Zhao

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Éric Dubois

Rafik Goubran

Abdulmotaleb El Saddik

Ze-Nian Li

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

RST invariance of image watermarking algorithms and the framework of mathematical analysis

by

Dong Zheng

A thesis submitted to
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Doctorate of Philosophy
in
Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Information Technology and Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-50758-2
Our file Notre référence
ISBN: 978-0-494-50758-2

Abstract

In recent years, watermarking algorithms robust to the geometrical distortions have been the focus of research. Most of the proposed geometrical-transform-invariant algorithms are RST (Rotation, Scaling and Translation) invariant due to the fact that changing the image size or its orientation, even by slight amount, could dramatically deteriorate the performance of the watermark detection.

Most of the existing RST invariant watermarking algorithms can be classified into several categories: RST invariant domain, salient feature, template, image decomposition and stochastic analysis based algorithms. An in-depth theoretical analysis of these algorithms is given in this thesis. With the detailed experimental results, the advantages and disadvantages of each algorithm are presented. This provides a solid basis for the further research in this field. Moreover, the clarification of the current algorithms' limitation can lead to new ideas of designing better algorithms.

Based on the detailed analysis of the existing RST invariant watermarking algorithms, a novel feature-based RST invariant watermarking algorithm is proposed in this thesis. And, a framework is established to mathematically guide the watermark embedding process and analyze the performance of the watermarking algorithm like watermark embedding strength. Since it is difficult to model the entire image using a single mathematical model, the cover image is segmented into several homogeneous regions using the maximum a posteriority probability (MAP) segmentation. Each segmented-region of the image is modelled using a generalized Gaussian distribution model. Then the image can be approximated using a Gaussian mixture distribution model. And some rotation-invariant features are extracted from the cover image using the SIFT

(Scale Invariant Feature) detection algorithm. Image normalization is used to achieve scaling and translation invariance. Then, the user-defined disk regions centered at the well-selected feature points will be used for watermark embedding and extraction. In the watermark embedding process, the watermark is approximated as additive white Gaussian noise. And NVF (Noise Visibility Function) is used to adaptively adjust the watermark embedding strength. With the establishments of the stochastic models for the cover image and the watermark, it is easy to clarify the relation between the fidelity of the watermarked image and the embedding capacity in a more accurate mathematical way instead of the currently used empirical way. In the watermark extraction process, the linear correlation is used to detect the existence of the watermark. The experimental results demonstrate the proposed scheme is robust to RST transformation, noise pollution and JPEG compression.

The established mathematical model for images provides a good analysis tool for watermarking algorithms, and can be further exploited and refined to give a better understanding of the various aspects of watermarking algorithms.

Contents

List of Tables

List of Figures

List of Acronyms

ACF	Auto-Correlation Function
AWGN	Additive White Gaussian Noise
CHF	Circular Harmonic Function
CIT	Circular Integration Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DoG	Difference of Gaussian
DWT	Discrete Wavelet Transform
EM	Expectation Maximization
FFT	Fast Fourier Transform
FMT	Fourier Mellion Transfrom
GGD	Generalized Gaussian Distribution
HOS	Higher Order Spectra
HVS	Human Visual System
IDFT	Inverse Discrete Fourier Transform
i.i.d	Independent Identical Distribution
ILPM	Inverse Log Polar Mapping
JPEG	Joint Photographic Experts Group
LoG	Laplacian of Gaussian
LPM	Log Polar Mapping
LSB	Least Significant Bit

LTI	Linear Time Invariant
MAP	Maximum A posteriority Probability
ML	Maximum Likelihood
NVF	Noise Visibility Function
OTCHF	Optimal Tradeoff Circular Harmonic Function
PN	Pseudo Noise
PSNR	Peak Signal to Noise Ratio
PSR	Peak-to-Sidelobe Ratio
RBA	Random Bending Attack
RHF	Radial Harmonic Filter
RIT	Radial Integration Transform
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RST	Rotation, Scaling, and Translation
SD	Synchronous Detector
SIFT	Scale Invariant Feature
SPOMF	Symmetrical Phase-Only Matching Filter
TMD	Template Matching Detector

Acknowledgement

I would like to thank my supervisor, Professor Jiying Zhao, for bringing the problem of digital watermarking to me, and for his patient guidance and feedbacks during every step of my work. And I also thank Yan Liu and Huiyan Qi for their help in my research work.

I also would like to thank Sha and April for their positive supports.

And I deeply appreciate the encouragement from my parents.

Chapter 1

Introduction

1.1 Digital watermarking

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

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

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

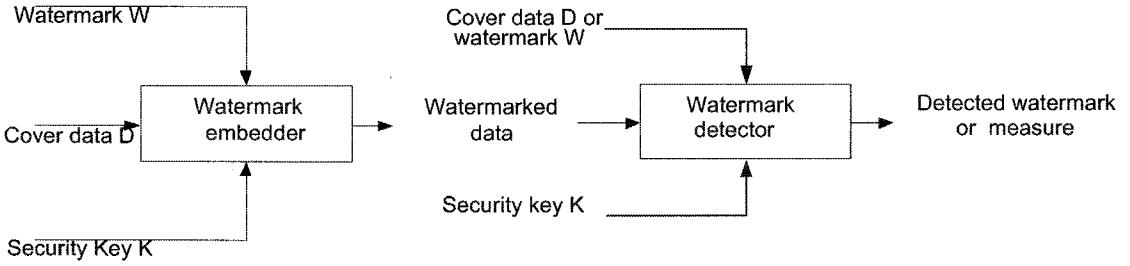


Figure 1.1: A typical watermarking system.

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

The inputs for the watermark detector are the watermarked data, the security key and, depending on the method, the original data and/or the original watermark. As discussed in [1], the watermark detector includes two functionalities: First, watermark extraction extracts the possible watermark vector from the watermarked data, then the second functionality is to justify whether the extracted watermark contains the original watermark or not. This usually involves comparing the extracted watermark with the original watermark and the result could be some kind of confidence measurement indicating how likely the original watermark is present in the watermarked data. For

some watermarking algorithms, the extracted watermark can be further decoded to get the embedded message for various purposes such as copyright protection.

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

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

The watermark is considered to be robust if it is embedded in a way such that the watermark can survive even if the watermarked data D' goes through severe distortions. The watermark extraction procedure is depicted as follows:

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

where $d(.)$ is the extraction function. D and W are the optional inputs for the extraction function.

Watermark detection can be thought as watermark extraction when the watermark carries only one bit information that indicates if the original watermark is present in the work or not.

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

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

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

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

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

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

1.2 Host signals

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

1. Digital image watermarking. Most of the research about digital watermarking are on image watermarking. This might be because there are so many images available on World Wide Web free of charge and without any copyright protection.

2. Digital video watermarking. A video consists of a sequence of still images, therefore all the watermarking methods applied on image can be applied on video. However, video watermarking has other problems.

   For example, [2] pointed out that it is dangerous to use the same watermarking key for a whole video. If the same key is used for all the frames or shots in a video sequence, it would make the watermarking algorithm vulnerable to the collusion attack. If a unique key is used for each frame or shot of the video sequence, it would make the key management and key distribution very difficult. A video watermark should be able to resist different types of attacks such as frame averaging, frame dropping, and frame swapping.

3. Digital audio watermarking. In the case of audio signals, the term watermarking can be defined as the way of transmission of additional data along with audio

signals in a robust and inaudible manner. Audio watermarking is based on the psycho-acoustical approach of perceptual audio coding techniques. It exploits the properties of the human ear by embedding one or more key-dependent watermark signals below the masking threshold.

4. 3D virtual objects watermarking. The most important component for watermark embedding in both VRML (Virtual Reality Modelling Language) and MPEG-4 is the 3D polygonal mesh. The shape of a 3D polygonal mesh is defined by two components, vertex coordinate and vertex topology. Vertex coordinates combined with vertex topology define more complex geometrical primitives such as lines and polygons. These components are the most important targets for embedding in 3D mesh polygonal meshes.

5. Others such as hologram, text, software, and database watermarking.

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

1.3 Digital image watermarking

Digital image watermarking embeds data (called the watermark) into the host images in an imperceptible or perceptible way. Digital image watermarking can be used in a number of applications with different requirements including copyright protection, content authentication and content description. As a great volume of image data is stored in digital format, it has become easier to modify or forge image information. Digital image watermarking can work as an effective solution to the problem of the copyright infringement since the embedded watermark can be used as a proof of the ownership. One of the most important requirements is that the embedded watermark should be

robust against certain malicious or unintentional attacks based on the design requirement. The attempt to remove or destroy the watermark will dramatically degrade the host image quality. This is often referred to as "robust watermarking". Watermarking can also be used to address the problem of tampering. For example, if an image is to be used as evidence, the image must be proved to be credible. By "credible", we mean that the image source is authentic and the information content has not been

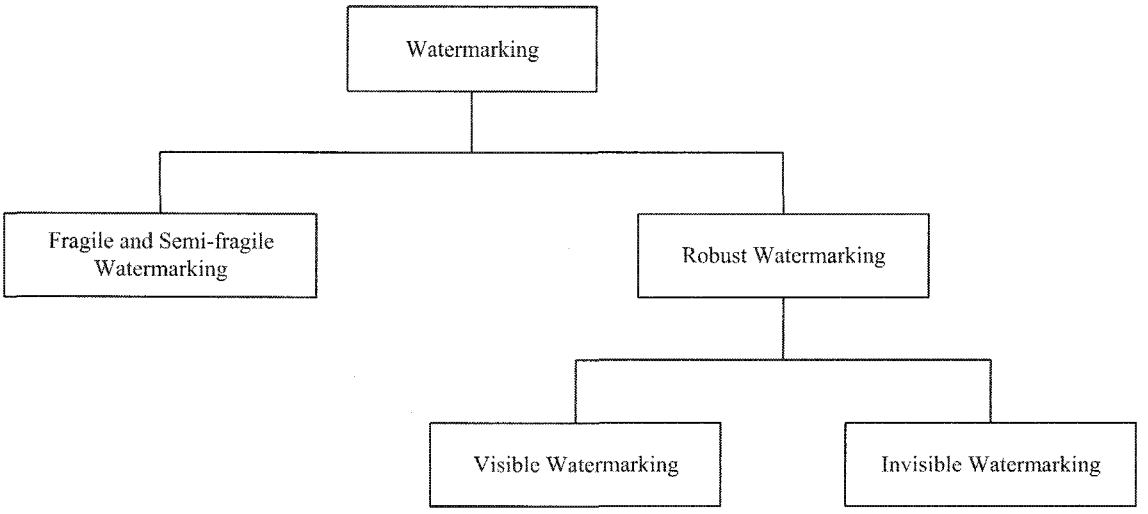


Figure 1.2: A classification of digital image watermarking.

altered in transit to its destination. Digital image watermarking serving this kind of purpose is referred to as "fragile watermarking" or "semi-fragile watermarking", which can indicate whether tampering to the original image data had occurred. Or it can give more information about the attacks and degradation of the host image [3]. There are other digital image watermarking applications such as content description, in which the embedded watermark bears the description information of the host images.

Fig. 1.2 is the classification of digital image watermarking.

1.4 A typical digital image watermarking system

Fig. 1.3 presents a simple block diagram of a typical digital image watermarking system.

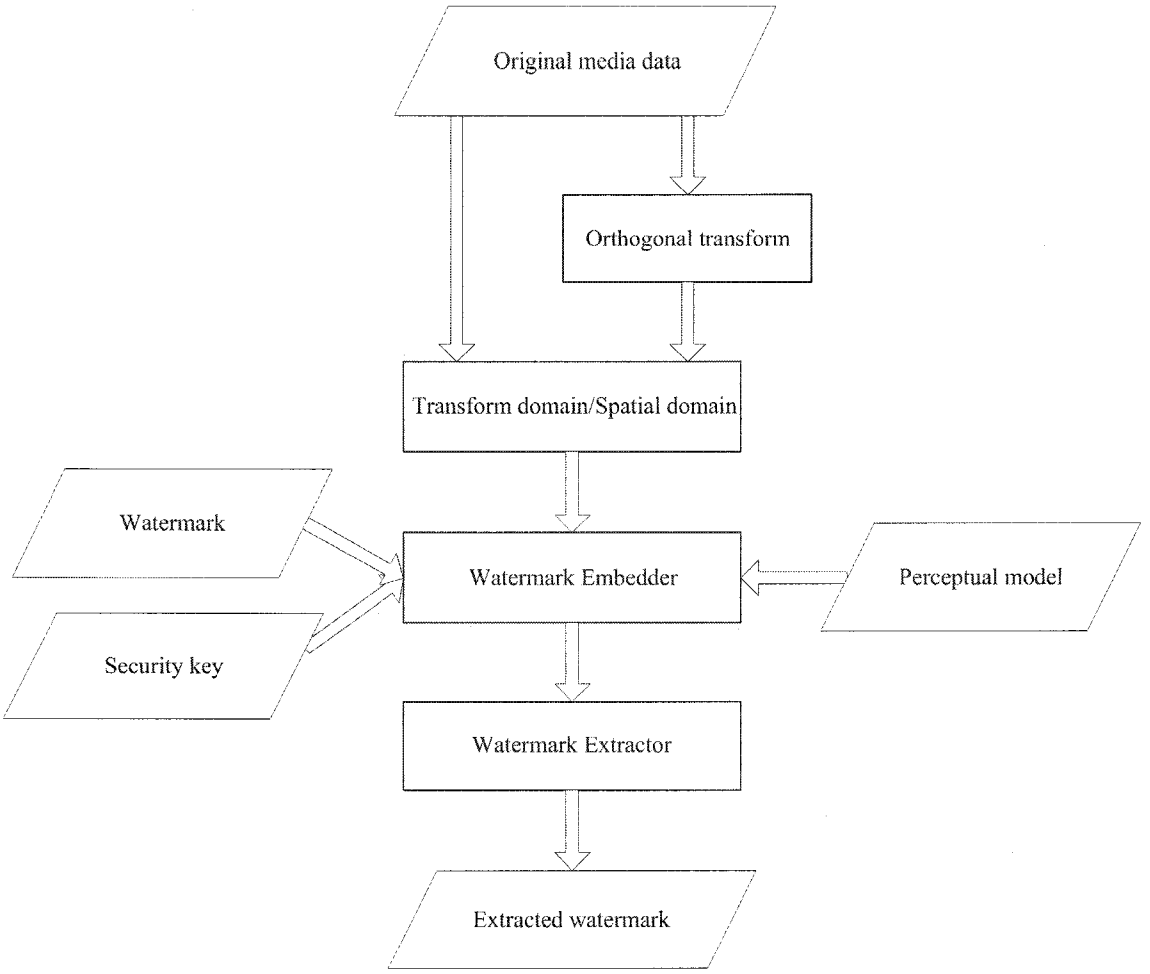


Figure 1.3: The block diagram of digital image watermarking.

Orthogonal transform can be Discrete Fourier Transform, Discrete Cosine Transform or Discrete Wavelet Transform. The perceptual model is used to select those regions suitable for watermark embedding, which is very important for invisible watermarking.

1.5 Performance evaluation of a watermarking system

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

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

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

3. Capacity. Capacity means the maximum amount of information the embedded watermark can carry and those information can be detected reliably for the purpose of copyright protection and authentication.

As mentioned by [4], "the smaller is the number of bits of core information or payload contained in a watermark, the greater the chance of it being communicated without error." One way to implement the spread spectrum based watermarking scheme is to generate the watermark in the form of pseudo random sequence; the watermark detection is executed by computing the correlation between the extracted mark and the original watermark. This approach is very robust however the capacity is low (only

1 bit). A good watermarking algorithm should achieve a good trade-off among these requirements, refer to Fig. 1.4.

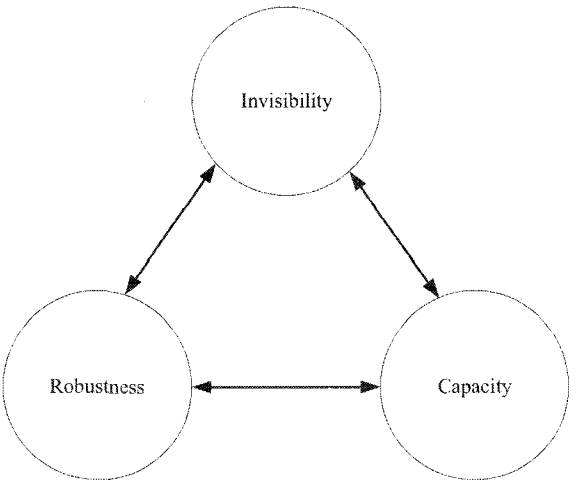


Figure 1.4: The requirements for a robust digital image watermarking system.

1.6 Applications of digital watermarking

Digital watermarking systems are developed based on the applications. Among others, the following applications of watermarking are more common:

1. Copyright protection. One of the main applications of watermarking is copyright protection. The idea is to embed information about the copyright owner into the data to prevent parties from claiming to be the rightful owners of the data. The watermarks used for that purpose are supposed to be very robust against various attacks intended to remove the watermark.

2. Content authentication. To be able to authenticate the content, any change to or tampering with the content should be detected. This can be achieved through

the use of "fragile/semi-fragile watermark" which has low robustness to the modifications of the host image. The semi-fragile watermarking can also serve the purpose of quality measurement. The extracted watermark can not only tell the possible tampering with the host image, but also give more information about the degradation of the host image, such as PSNR of the degraded host image. This can be very useful for broadcasting or network transmission, since sometimes the original reference is not available at the receiver side. The degradation of the transmitted media can be further used to evaluate the quality of service (QoS) of the transmission or the congestion of the network.

3. Copy and usage control. Different payment entitles the users to have different privilege (play/copy control) on the object. It is desirable in some systems to have a copy and usage control mechanism to prevent illegal copy of the content or limit the number of times of copying. A watermark can be used for such purpose.

4. Content description. The watermark can contain some descriptive information of the host image such as labelling and captioning. For this kind of application, the capacity of the watermark should be relatively large and there is usually no strict requirement for the robustness.

1.7 Benchmarking tools for watermarking

Large amounts of watermarking algorithms have been proposed in these years, and automated test and evaluation tools for watermarking algorithms are needed. Stirmark [5] is the first benchmarking tool developed at the University of Cambridge for digital watermarking technologies. Given a watermarked image, Stirmark can generate a num-

ber of image modifications which can be used to verify the watermarking algorithm. Version 4 of Stirmark has evolved into a fully automated test benchmarking tool. The attacks include: cropping, flip, rotation, sharpening, Gaussian filtering, line removal and JPEG compression. Checkmark [6] developed at the university of Geneva is a watermark benchmark suite in Matlab script. It also includes classes of attacks to test the performance of the watermarking algorithm and can be easily modified according to the requirement of the end-user. Stirmark is very popular and has been used widely to evaluate the performance of the image watermarking algorithms.

1.8 Techniques for digital image watermarking

According to the domain in which the watermark information is embedded in the image, digital image watermarking techniques can be classified as spatial domain and transform domain techniques. It cannot be exhaustive, however we list several image watermarking algorithms that are important and typical in this research area.

One of the first used techniques for image watermarking appeared in 1993. [7] presented two techniques to hide data in the spatial domain of images. These methods were based on the pixel value's Least Significant Bit (LSB) modifications. The algorithm proposed by [8] is known as image downgrading. Given two images of same size, one acts as cover and the other the watermark image. The most significant bits of the watermark image are taken and embedded in the least significant bits of the cover image. Extracting the least significant bits of the watermarked image can give a rough estimation of the watermark image. [9] proposed an algorithm based on the pixel region classification. Pixels are classified into homogeneous luminance zones. Then the pixels have their gray levels changed following a rule that takes into the account where the

pixel is inserted, and the value of the bit to be embedded.

Spread spectrum and transform domain watermarking was introduced by [10]. Cox's approach uses spread spectrum communication techniques to embed a single bit in the image. [11] defines spread spectrum communications as: "Spread spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery."

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

Most of the spatial domain based watermarking algorithms are easy to implement, but do not provide much resistance against attacks. The transform/spectral domain based watermarking has proved to be a better choice for robust image watermarking. Watermark has been embedded using the Discrete Cosine Transform (DCT) [12][13], Discrete Fourier Transform[14][4] and Discrete Wavelet Transform[15][16][17].

[13] proposed a DCT based watermarking algorithm. The image is first divided into 8×8 pixel blocks. After DCT transform and quantization, the mid-frequency range DCT coefficients are selected based on a Gaussian network classifier. The mid-frequency range DCT coefficients are then used for embedding. Those coefficients are modified using a linear DCT constraints. It is claimed that the algorithm is resistant to JPEG compression.

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

$$V' = V + \alpha \times W \tag{1.3}$$

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

$$sim = \frac{W' \times W^T}{\sqrt{(W' \times W'^T)(W \times W^T)}} \tag{1.4}$$

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

Kundur and Hatzinakos [16] proposed a wavelet based watermarking algorithm. The multiresolution data fusion is used for embedding where the image and the watermark are both transformed into the discrete wavelet domain. The watermark is embedded into each wavelet decomposition level of the host image. During detection, the watermark is an average of the estimates from each resolution level of wavelet decomposition. This algorithm is robust against JPEG compression, additive noise and filtering operations.

Contrary to the LSB approach, the key to making a watermark robust is that it should be embedded in the perceptually significant components of the image [10][15][18].

A good watermark is one which takes into account the behavior of human visual system. For the spread spectrum based watermarking algorithm, a scaling factor can be used to control the amount of energy a watermark has. The watermark energy should be strong enough to withstand possible attacks and distortions. Meanwhile a large watermark energy will affect the visual quality of the watermarked image. A perceptual model is needed to adjust the value of the scaling factor based on the visual property of the host image to achieve the optimal trade-off between robustness and invisibility.

The human visual system (HVS) shows variable sensitivities based on the properties of images. These properties include frequency, luminance sensitivity, color and contrast masking. A large smooth area of the image corresponds to low frequency component, while the heavily textured area corresponds to high frequency. In practice, an empirical perceptual model [4] that the watermark is embedded into the middle frequency component is widely used. The reason is that the watermark embedded in the high frequency is easily removed by attacks such as low pass filtering and JPEG compression, while embedding the watermark in the low frequency will affect the visual quality of the host image dramatically. It was also shown that the human eyes are less sensitive to the bright area of the image. So the watermark can be embedded with different strengths according to a luminance function to achieve the trade-off between robustness and invisibility [19]. This perceptual model can be used in the spatial domain watermarking algorithm. More complicated perceptual models take luminance sensitivity and texture masking into account [20][21]. The watermark is embedded according to luminance and texture masking (such as edge detection). A set of empirically adjusted parameters is required.

As discussed in [22], for the color channel model, it is found that the human eyes are less sensitive to the changes of high frequency components along the yellow-blue

axis. So for the color image watermarking, we can take advantage of the property of the human eye to embed the watermark.

1.9 RST invariant digital image watermarking

In order for a watermark to be useful, it must be robust against a variety of possible attacks by pirates. These include robustness against compression like JPEG, geometrical distortions such as scaling and aspect ratio changes, rotation, cropping, row and column removal, and other attacks such as noise pollution, filtering, cryptographic, statistical attacks, as well as insertion of other watermarks. Although, many methods perform well against compression, they lack robustness to geometric transformations [23].

Geometrical distortion including rotation, scaling, translation, cropping/shearing, projective transformation can be global or local. Global geometrical distortion affects all the pixels of an image in the same manner, while local geometrical distortion affects different portion of an image in different manners. As mentioned in [1], "robustness to geometrical transforms remains one of the most difficult outstanding areas of watermarking research". The geometrical distortion will cause synchronization error which can dramatically deteriorate the performance of the watermark detection. The global geometrical transform is uniquely determined by a set of parameters such as rotation degree, scaling ratio and translation parameters, while the local geometrical transform involves with a set of transforms applied to different sub-regions of the image with different parameters. Since the parameters needed to describe the local geometrical transform are normally much more than those needed for global geometrical transformation, re-synchronization from local geometrical distortion is much more difficult than re-synchronization from a global one. Random bending attacks are defined as a set of

local distortions which change each individual pixel location based on a random set of parameters. In Stirmark benchmark software, random bending attacks are introduced as local distortions to test the watermarking algorithms, and they prove to be very challenging. In this thesis, the main focus of the research is on the topic of rotation, scaling and translation invariant image watermarking. RST transform is one of the most common and challenging attacks that the watermarking scheme needs to handle. This is also the critical first step to lead to a watermarking scheme that can handle more complicated global or local geometrical transform.

RST (Rotation, Scaling and Translation) can affect the performance of watermarking algorithms regarding to the detection of the existing watermark from a host image. In [24], ROC (Receiver Operating Characteristic) curves are used to evaluate the performance of watermark detection when the RST transform is applied to the watermarked image. It was shown that the ROC (Receiver Operating Characteristic) curves for ESD (Exhaustive Search Detector) are much worse when the RST transform is present, which means the geometrical distortions will degrade the performance of the algorithm regarding to the detection of the existing watermark from the host image if there is not an effective scheme that can deal with geometrical distortions. The performance of TMD (Template Matching Detector) and SD (Synchronous Detector) are better because they provide such schemes that can deal with geometrical distortions. In the paper [25], a theoretical analysis of the watermarking algorithm proposed by [19] is given. The bit error probability of the watermark is increased because of the geometrical transform and the interpolation used. It was further discussed in papers [24][25] and [26], the local geometrical transform such as random bending attacks in Stirmark not only increase the search space and computation significantly for Exhaustive Search Detector, but also pose a serious problem for the template based watermarking algorithm.

Based on the template-based approaches, it is quite straightforward to come up with the idea that if we can identify some kind of pattern that the cover image bears with inherently, we can use this pattern as the reference template. Because it has to be recognizable, normally we use salient features of the cover image as the desired pattern. Therefore, we can identify the pattern even when the cover image is severely distorted. Meanwhile, image normalization has been widely used in pattern recognition and image registration [27]. It also helps researchers achieve scaling invariance in watermarking schemes.

Some of the feature-based watermarking schemes have been proposed in literature. In [28], the geometric invariant watermarking scheme is based on moments and image normalization. Geometric moments were used to geometrically normalize the image before watermark embedding at the encoder and before watermark extraction at the decoder. And in [29], the authors extracted features of the cover image and used the disk regions centered at the feature points for watermark embedding and extraction. Meanwhile, image normalization was applied in the scheme to make those disk regions invariant to rotation and scaling. It was stated that the extracted feature points can survive a variety of attacks and can be used as reference points for both watermark embedding and watermark extraction.

In [29], although the image normalization was used to grant the rotation and scaling invariance to the disk regions, it was clearly shown in the experimental results that the performance of the proposed watermarking scheme is not good against rotation. Through experiments, we find out that the feature points cannot be located accurately, if the watermarked image goes through distortions and geometrical transforms.

Also some other RST invariant image watermarking algorithms are proposed such as the Radon transform based algorithm [30], the image normalization based algorithm

[31] and the image decomposition (pseudo Zernike moment) based algorithm [32]. All these algorithms tried to realize the RST invariance using different approaches, which is also the main focus of the research work presented in this thesis.

1.10 Scope and structure of the thesis

To have a clear understanding of different RST invariant image watermarking algorithms, an in-depth review and evaluation of the currently proposed RST invariant image watermarking algorithms is given in this thesis. The working principle of each algorithm is illustrated, the performance of the algorithm is presented with detailed theoretical analysis and experiential result. The advantage and disadvantage of each algorithm is discussed with detailed reasoning and analysis. The fundamental theories and techniques related to the image watermarking are introduced in Chapter 2. Chapter 3 and Chapter 4 contain all the in-depth review and evaluation.

Based on the analysis in Chapter 4, a critical question is raised: how to analyze and guide the watermarking process mathematically? The basis of this analysis is to establish a mathematical model for images, which is very important for the analysis and guidance of various aspects of the watermarking processes such as watermark embedding strength, probability of error and the relationship between robustness and fidelity. The algorithms discussed in Chapter 4 either use the empirical parameters or use simplified Gaussian model. The result of analysis is not accurate. To solve this problem in this thesis, the mixture Generalized Gaussian distribution is introduced and MAP (Maximum A posteriority Probability) image segmentation is used to segment the image into homogeneous regions. Each region is approximated with the Generalized Gaussian distribution with calculated parameters. This mathematical model can give

an accurate approximation of the statistical information of the image content, on which the frame of analysis is constructed.

To show the effectiveness of the established model and frame of analysis, a novel RST invariant image watermarking algorithm is proposed and the watermarking process is analyzed and guided mathematically using the proposed model and analysis framework. The watermarking scheme is based on the rotation invariant feature and image normalization [33]. MAP Image Segmentation is used to segment the cover image into several homogeneous regions. For each region, one feature point is extracted using Gaussian scale model. These points are robust against rotation, scaling and noise. Using the method addressed in [34], the orientation of the feature points are calculated. For each disk region centered at the feature point, the region is first rotated to align with the orientation of the feature point. Then the image normalization is applied to transform the disk region to its compact size, which is scaling invariant. In this way, the disk region for watermark embedding and extraction is rotation and scaling invariant.

The details of the modelling and the proposed scheme and implementation are discussed in Chapter 5 and Chapter 6. In chapter 7, the experimental results have been presented in detail. The robustness of the algorithm against rotation, scaling, JPEG compression and noise have been shown in experiments. With the help of the mathematical model, the fidelity of watermarked image is maintained. Also the theoretical derivation of the probability of error has been given with the comparison of the simulation results.

In Chapter 8, we conclude the thesis and give some suggestions and ideas for future research work.

1.11 Contributions of the research

The research involved in this thesis is mainly focused on the topic of RST invariant digital image watermarking algorithms. The contributions include the following work:

1. To have a clear understanding of the existing algorithms and develop a new RST watermarking scheme, a detailed literature review and evaluation on the existing RST watermarking algorithms have been given. A 91-page journal paper was published on "ACM computing surveys" based on this work.

   [1]. **Dong Zheng**, Yan Liu, Jiying Zhao, and Abdulmotaleb El Saddik, "A Survey of RST Invariant Image Watermarking Algorithms", *ACM Computing Surveys*, Vol. 39, No. 2, Article 5, pp. 1-91, June 2007.

   [2]. **Dong Zheng**, Yan Liu, and Jiying Zhao, "A survey of RST Invariant Image Watermarking Algorithms", *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)* 2006, Ottawa, Ontario, Canada, pp.2055-2058, May 7-10, 2006.

2. Based on the survey work, the advantages and disadvantages of the existing algorithms are summarized. A mathematical model for analyzing and guiding the watermarking processes is established. With the proposed framework of analysis, a new feature-based RST watermarking algorithm is proposed. The watermark embedding positions are accurately located using stochastic models and the Noise Visibility Function is used to guide the watermark embedding. One conference paper has been published and one journal paper has been submitted to IEEE Transactions on Image Processing.

[1]. **Dong Zheng** and Jiying Zhao, "A rotation invariant feature and image normalization based image watermarking algorithm", submitted to IEEE Transactions on Image Processing.

[2]. **Dong Zheng** and Jiying Zhao, "A rotation invariant feature and image normalization based image watermarking algorithm", *2007 IEEE International Conference on Multimedia & Expo (ICME2007)*, Beijing, China, pp. 2098-2101, July 2-5, 2007.

3. The Fourier Mellin Transform (FMT) has the property of rotation and scaling invariance. Once the Discrete Fourier Transform (DFT) and the FMT are applied to the image, the image will be transformed to the RST invariant domain. The FMT provides good RST invariance though major difficulties are encountered in its implementation. To solve this problem, the FMT-based RST Invariant watermarking algorithm and some variations have been proposed in the following several papers. The watermark is embedded in the LPM domain to simplify RST transformations to shifts. Phase correlation is used to rectify the watermark position to avoid exhaustive search. The related work are also covered in ACM survey paper with in-depth analysis and experimental comparison with other RST invariant algorithms.

   [1]. **Dong Zheng**, Yan Liu, and Jiying Zhao, "RST invariant digital image watermarking based on a new phase-only filtering method", *Elsevier Journal: Signal Processing*, Vol.85, No.12, pp.2354-2370, December 2005.

   [2]. **Dong Zheng**, Jiying Zhao, and Abdulmotaleb El Saddik, "RST Invariant Digital Image Watermarking Based on Log-Polar Mapping and Phase Correla-

tion", *IEEE Transactions on Circuits and Systems for Video Technology*, Special Issue on Authentication, Copyright Protection and Information Hiding, Vol. 13, Issue 8, pp. 753-765, August 2003.

[3]. Yan Liu, **Dong Zheng**, and Jiying Zhao, "An image rectification scheme and its applications in RST invariant digital image watermarking", *Springer Journal: Multimedia Tools and Applications*, Vol. 34, No. 1, pp. 57-84, July 2007.

[4]. Yan Liu, **Dong Zheng**, and Jiying Zhao, "A Rectification Scheme for RST Invariant Image Watermarking", *IEICE Transactions on Fundamentals of Electronics*, Communications and Computer Sciences, Special Section on Cryptography and Information Security, Vol. $E88-A$, No.1, pp. 314-318, January 2005, LETTER.

[5]. Yan Liu, Xiangsheng Wu, **Dong Zheng**, Jiying Zhao, and Jianping Yao, "Phase Information in RST Invariant Image Watermarking", *the Proceedings of the CSEE (Chinese Society of Electrical Engineering)*, Vol.25, No.10, pp. 89-96, 2005.

[6]. **Dong Zheng**, Yan Liu, and Jiying Zhao, "RST Invariant Digital Image Watermarking Based on a New Phase-Only Filtering Method", *Proceedings of 7th International Conference on Signal Processing (ICSP04, IEEE, CIE, IEE)*, Beijing, China, pp.25-28, Aug 31-Sep 4, 2004.

[7]. **Dong Zheng** and Jiying Zhao, "RST Invariant Digital Image Watermarking based on Resynchronization", *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)* 2004, Niagara Falls, Ontario, Canada, pp.1281-1284, May 2-5, 2004.

[8]. **Dong Zheng** and Jiying Zhao, "Apply phase information in RST im-

age watermarking", *IEEE International Conference on Consumer Electronics (ICCE2003)*, Los Angeles, California, USA, pp. 218-219, June 17-19, 2003.

[9]. **Dong Zheng** and Jiying Zhao, "LPM-Based RST Invariant Digital Image Watermarking", *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)* 2003, Montreal, Canada, pp. 1951-1954, May 4-7, 2003.

[10]. **Dong Zheng** and Jiying Zhao, "RST Invariant Digital Image Watermarking: Importance of Phase Information", *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)* 2003, Montreal, Canada, pp. 785-788, May 4-7, 2003.

4. Some other contributions: The human visual system and its effect on watermarking algorithm is discussed in the following paper. The result can be used to achieve the trade off between fidelity and robustness of the watermarking algorithm.

   [1]. Huiyan Qi, **Dong Zheng**, and Jiying Zhao, "Human Visual System Based Adaptive Digital Image Watermarking", *Elsevier Journal: Signal Processing*, Vol. 88, No. 1, pp. 174-188, January 2008.

Chapter 2

Fundamental theories and techniques

This chapter introduces the fundamental theories and techniques used in the existing RST invariant image watermarking algorithms.

2.1 Rotation, scaling, and translation transform

1. 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) by an angle ϕ about the origin clockwise:

$$\begin{cases} x' &= x\cos\phi + y\sin\phi \\ y' &= -x\sin\phi + y\cos\phi \end{cases} \tag{2.1}$$

2. A scaling transformation alters the size of an image. We obtain the transformation equations by multiplying the coordinate values (x, y) by scaling factors a and b

to produce the transformed coordinates (x', y'):

$$\begin{cases} x' &= x \cdot a \\ y' &= y \cdot b \end{cases} \tag{2.2}$$

Scaling factor a scales images in the x direction, and b scales in the y direction. While a and b are assigned the same value, a uniform scaling is produced that maintains relative image proportions.

3. A translation (or shift) is applied to an image by repositioning it along a straight-line path from one coordinate location to another [35]. 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').

$$\begin{cases} x' &= x + x_0 \\ y' &= y + y_0 \end{cases} \tag{2.3}$$

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

2.2 Fourier transform

The 2 dimensional FT (Fourier transform) of image $f(x, y)is$:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux+vy)} dxdy \tag{2.4}$$

and the inverse Fourier transform is:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{i2\pi(ux+vy)}dudv \tag{2.5}$$

In practice, images are always in finite size and are obtained by sampling. The DFT (Discrete Fourier Transform) are widely used and the DFT of an image array $f(x,y)$ of size $M\times N$ and the corresponding IDFT (Inverse DFT) are defined as follows [36]:

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)} \tag{2.6}$$

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)} \tag{2.7}$$

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

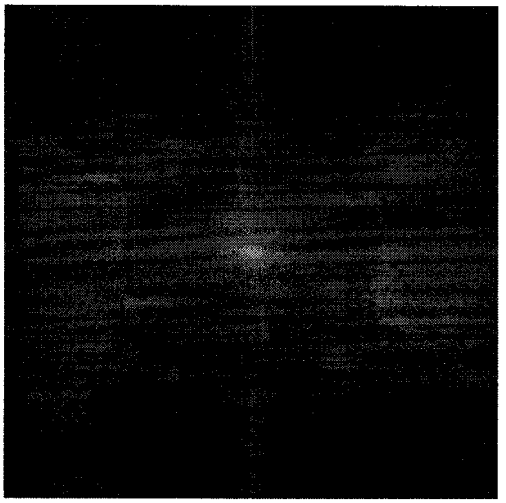
$$|F(u,v)| = \sqrt{R^2(u,v)+I^2(u,v)} \tag{2.8}$$

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

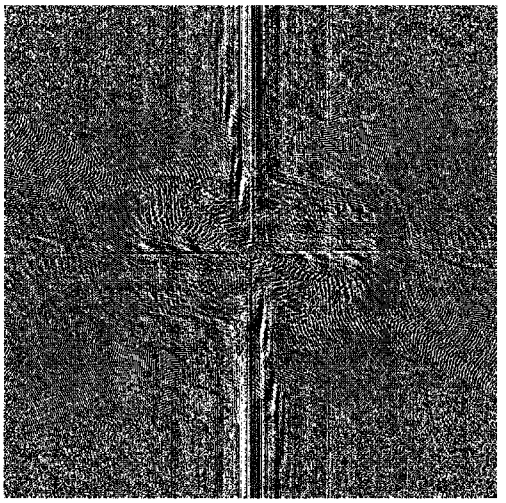
where $R(u,v)$ and $I(u,v)$ are the real and imaginary parts of $F(u,v)$, respectively.

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 amplitude does [37]. The FT and inverse FT processes of image *Barbara* are shown in Fig. 2.1. Fig. 2.1 (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. 2.1

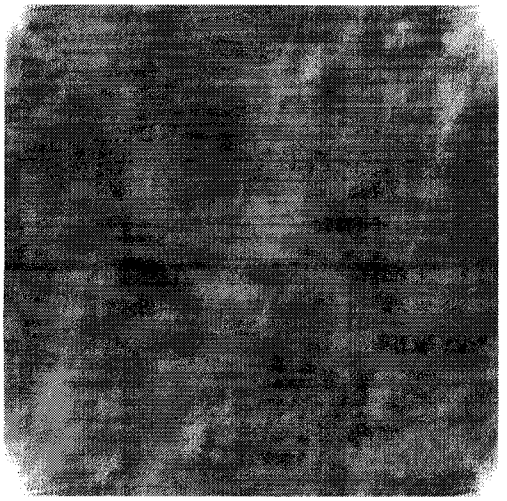
(a) *Barbara*.



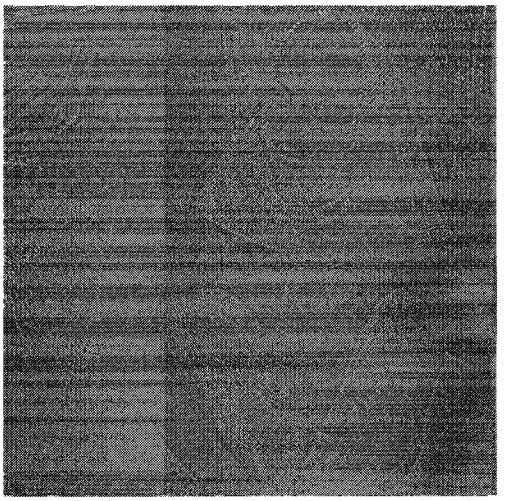
(b) The amplitude spectrum of image *Barbara*.



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



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



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

Figure 2.1: 2D FT and 2D inverse FT of image *Barbara*.

(d).

In the following, we present some basic properties of the 2D FT related to geometric transformations in the spatial domain.

1. Suppose $f_1(x, y)$ is achieved by rotating the image $f_0(x, y)$ by a degree of ϕ in the spatial domain:

$$f_1(x, y) = f_0((x\cos\phi + y\sin\phi), (-x\sin\phi + y\cos\phi)) \tag{2.10}$$

Suppose that $F_1(u, v)$ and $F_0(u, v)$ are respectively the Fourier transform of $f_1(x, y)$ and $f_0(x, y)$. They are related by :

$$F_1(u, v) = F_0((u\cos\phi + v\sin\phi), (-u\sin\phi + v\cos\phi)) \tag{2.11}$$

As shown in Fig. 2.2 (c) and (d), Eq. (2.11) indicates that rotating $f(x, y)$ by an angle of ϕ rotates $F(u, v)$ by the same angle.

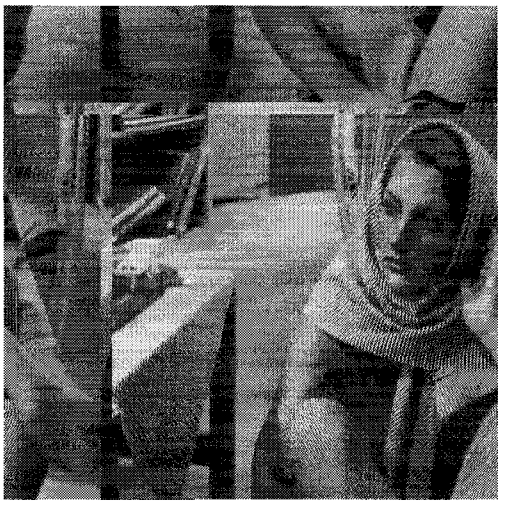
2. Scaling in the spatial domain can cause a reciprocal scaling in the frequency domain.

$$\mathfrak{F}[f(ax, by)] = \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right) \tag{2.12}$$

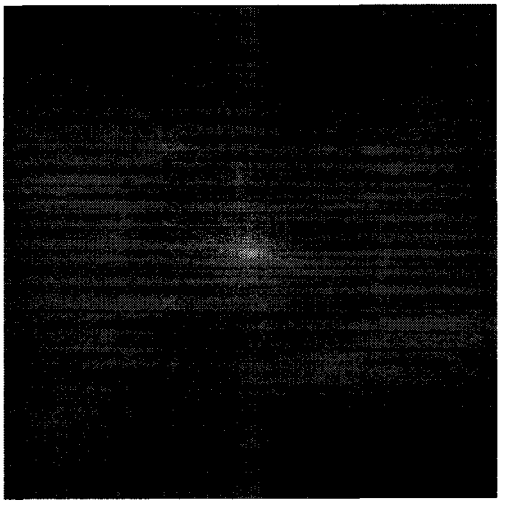
where a and b are respectively the scaling factor along x axis and y axis.

This property is shown in Fig. 2.2 (e) and (f).

3. Assuming that $F(u, v)$ is the Fourier transform of the image $f(x, y)$, then a translation (shift) in the spatial domain of $f(x, y)$ will cause a linear shift in the phase

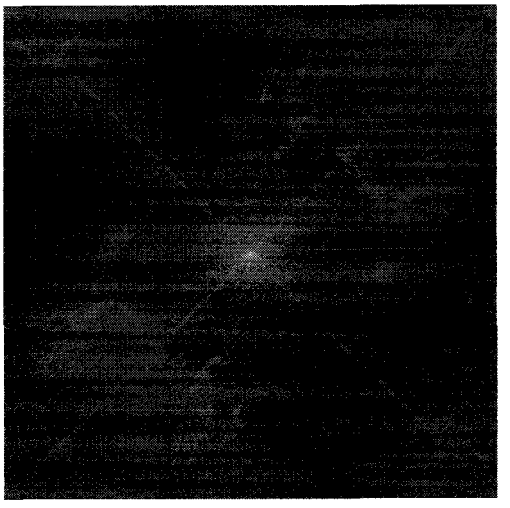


(a) *Barbara* circularly translated by 100 pixels.



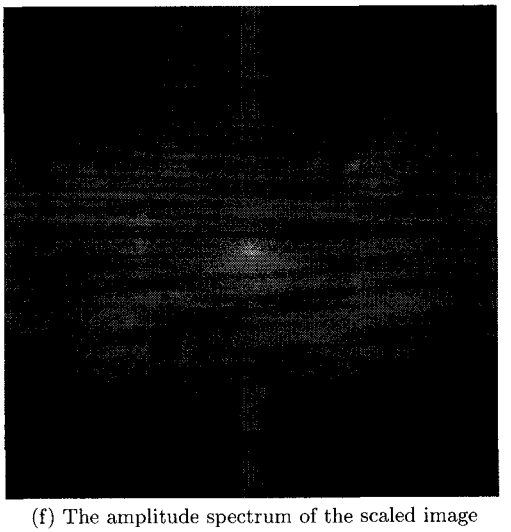
(b) The amplitude spectrum of the translated image *Barbara*.

(c) *Barbara* rotated by 45°.



(d) The amplitude spectrum of the rotated image *Barbara*.

(e) *Barbara* scaled by 1.2.



(f) The amplitude spectrum of the scaled image *Barbara*.

Figure 2.2: Properties of Fourier transform.

of $F(u,v)$ and will not change the magnitude spectrum, as shown in Fig. 2.2 (a) and (b) and Eq.(2.13).

$$\begin{aligned} \mathfrak{F}[f(x+x_0, y+y_0)] &= \mathfrak{F}[f(x,y)]e^{j2\pi(x_0u/M+y_0v/N)} \\ &= F(u,v)e^{j2\pi(x_0u/M+y_0v/N)} \end{aligned} \tag{2.13}$$

where $\mathfrak{F}[\cdot]$ denotes the Fourier transform.

We can consider rotation, uniform scaling, and translation altogether. Suppose that the RST parameters are ϕ, c and (x_0, y_0) respectively, and that the Fourier transform of $f_1(x,y)$ and $f_0(x,y)$ are respectively $F_1(u,v)$ and $F_0(u,v)$, their magnitudes are related by [4][38]:

$$|F_1(u,v)| = |c|^{-2}|F_0(c^{-1}(u\cos\phi + v\sin\phi), c^{-1}(-u\sin\phi + v\cos\phi))| \tag{2.14}$$

Eq. (2.14) is independent of the translational parameters (x_0, y_0), which is the translation property of the Fourier transform [39].

2.3 Log-polar mapping and inverse log-polar mapping

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

$$\begin{cases} \rho = \ln(\sqrt{x^2+y^2}) \\ \theta = \tan^{-1}(\frac{v}{u}) \end{cases} \tag{2.15}$$

(a) Sampling points on the Cartesian plane.

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

Figure 2.3: Log polar mapping.

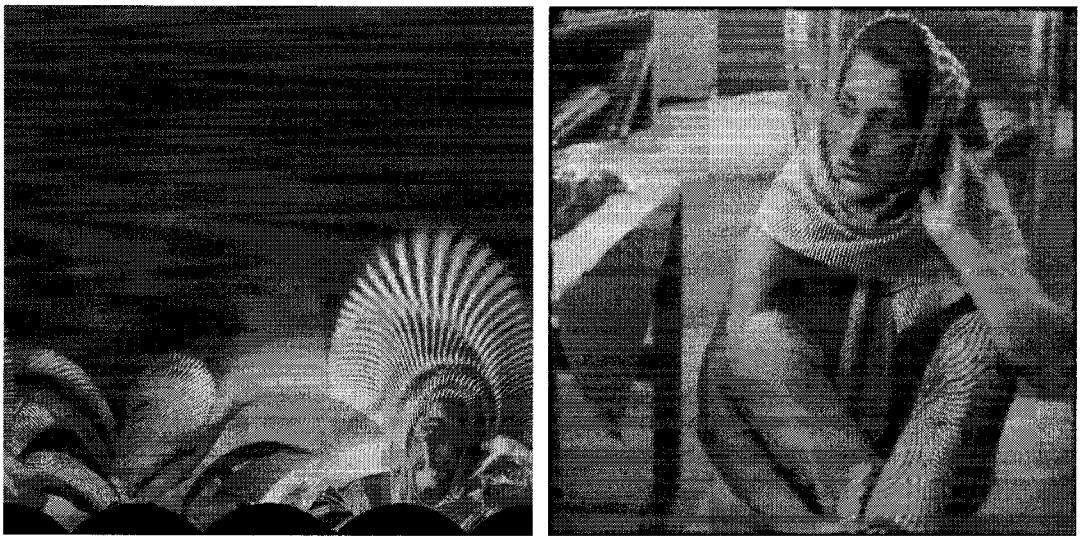
where $\rho \in \Re^2$ and $0 \leq \theta < 2\pi$.

As shown in Fig. 2.3, the log-polar mapping is just like a sampling process. The log-polar sampling points on the Cartesian plane in Fig. 2.3 (a) are used to construct the transformed image on the log-polar plane in Fig. 2.3 (b). The points on the circles which take the center of the image as the origin are mapped to form the Cartesian plane of the LPM transformed image.

The inverse log-polar mapping is:

$$\begin{cases} x &= e^{\rho}\cos\theta \\ y &= e^{\rho}\sin\theta \end{cases} \tag{2.16}$$

The log-polar mapping of image *Barbara* and the inverse log-polar mapping are shown in Fig. 2.4 (a) and Fig. 2.4 (b). The bilinear interpolation is used in the computation of the LPM and ILPM.



(a) LPM of image *Barbara*. (b) The result of inverse LPM.

Figure 2.4: LPM and inverse LPM of the *Barbara* image.

If we apply the log-polar mapping to the Fourier magnitude of an image, we can rewrite Eq. (2.14) by using log-polar coordinates.

The magnitude of the Fourier spectrum can be written as [38][4]:

$$|F_1(u,v)| = |c|^{-2}|F_0(c^{-1}e^{\rho}\cos(\theta-\phi), c^{-1}e^{\rho}\sin(\theta-\phi))| \tag{2.17}$$

or

$$|F_1(\rho,\theta)| = |c|^{-2}|F_0(\rho-\ln c, \theta-\phi)| \tag{2.18}$$

Eq. (2.18) demonstrates that the amplitude of the log-polar spectrum is scaled by $|c|^{-2}$, that image scaling results in a translational shift of $\ln c$ along the log-radius ρ axis, that image rotation results in a cyclical shift of ϕ along the angle θ axis, and that image translation has no effects in the LPM domain.

A log-polar sampled image is the one whose samples are centered on points mapping to integral ring number R and wedge number W, $R \in \{0, \ldots, n_r - 1\}$, $W \in \{0, \ldots, n_w - 1\}$. The separation between sample points is proportional to the distance from the sampling center. Log-polar sampled images are often displayed on orthogonal (R, W) axes, which is also called (ρ, θ) axes in the thesis. As shown in Fig. 2.3 (a), the ring number R represents the circle index and the wedge number W represents the sample point index on each circle.

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

$$\begin{cases} r &= \sqrt{(x - x_c)^2 + (y - y_c)^2} \\ \theta &= tan^{-1} \frac{y - y_c}{x - x_c} \end{cases} \tag{2.19}$$

$$\begin{cases} R &= \frac{(n_r - 1) \ln(r / r_{min})}{\ln(r_{max} / r_{min})} \\ W &= \frac{n_w \theta}{2\pi} \end{cases} \tag{2.20}$$

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

$$\rho = \ln r \tag{2.21}$$

The LPM can be explained by the following equation:

$$P = \mathfrak{L}(C) \tag{2.22}$$

where P and C are respectively the points in the LPM magnitude spectrum and the

points in Cartesian magnitude spectrum, while $\mathfrak{L}(\cdot)$ is the LPM computation operator.

2.4 Fourier-Mellin transform

The Fourier-Mellin transform is a log-polar mapping (LPM) followed by a Fourier transform, while the inverse Fourier-Mellin transform is an inverse log-polar mapping (ILPM) followed by an inverse Fourier transform. According to the translation property of the Fourier transform, after applying the Fourier transform to both sides of the Eq. (2.18), the result I_1 and I_0 is related by

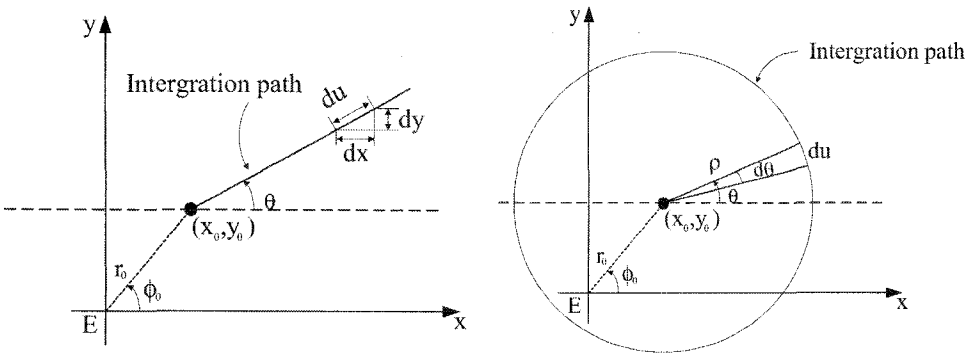
$$I_1(\omega_\rho, \omega_\theta) = |c|^{-2} e^{-j(\omega_\rho \cdot \ln c + \omega_\theta \cdot \phi)} I_0(\omega_\rho, \omega_\theta) \tag{2.23}$$

The Fourier magnitude of the two LPM mappings is related by

$$|I_1(\omega_\rho, \omega_\theta)| = |c|^{-2} |I_0(\omega_\rho, \omega_\theta)| \tag{2.24}$$

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

Eq. (2.24) is equivalent to computing the Fourier-Mellin transform [4]. Eq. (2.24) demonstrates that the amplitude of Fourier-Mellin spectrum is scaled by $|c|^{-2}$ caused by scaling transform, and is invariant to rotation and translation. $|c|^{-2}$ will cause no problem at all if we use normalized correlation to detect watermarks, so the Fourier-Mellin transform is truly invariant to RST.



(a) Radial Integration Transform (RIT). (b) Circular Integration Transform (CIT).

Figure 2.5: Radon transform.

2.5 Radon transform

The Radon transform represents an image as a collection of projections along various directions [41]. It is used in areas ranging from seismology to computer vision. The two one-dimensional generalized Radon transforms [42], as shown in Fig. 2.5 (a) and Fig. 2.5 (b), were introduced. As shown in Fig. 2.5 (a), the Radial Integration Transform (RIT) of a function $f(x, y)$ is defined as the integral of $f(x, y)$ along a straight line that begins from the origin $f(x_0, y_0)$ and has angle θ with respect to the horizontal axis. The RIT is defined as follows:

$$R_f(\theta) = \int_0^{+\infty} f(x_0 + ucos\theta, y_0 + usin\theta)du \tag{2.25}$$

If the image is rotated by a certain degree, the RIT of the rotated image will be circularly shifted.

The Circular Integration Transform (CIT) of a function $f(x, y)$ is defined as the integral of $f(x, y)$ along a circle curve with center $f(x_0, y_0)$ and radius ρ (see Fig. 2.5

(b)). The CIT is given by the following equation:

$$C_f(\rho) = \int_0^{2\pi} f(x_0 + \rho cos\theta, y_0 + \rho sin\theta)\rho d\theta \tag{2.26}$$

If the image is scaled uniformly by c, the CIT of the scaled image is scaled by c.

So the RIT is independent of scaling, and the rotation of the image only results in a shift of RIT. Similarly, the CIT is independent of rotation, and the CIT is scaled when the image is scaled. Using these properties, we can detect the rotation and scaling that the image has undergone by applying the CIT and RIT to the image and observing the changes of the CIT and RIT.

2.6 Circular harmonic functions

The circular harmonic expansion is another method used for rotation, scaling, and translation invariant pattern recognition [43][44][45]. The circular harmonic function (CHF) is useful in representing the rotational property of an image, which can be expressed in polar coordinates with period of 2π in angle and thus can be expressed in terms of a Fourier series expansion in angle [46]. 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 scaling and translation.

Let $f(x, y)$ denotes the reference image in Cartesian coordinates, we can transform $f(x, y)$ into polar coordinates $f_p(r, \theta)$. Because $f_p(r, \theta)$ is periodic in θ with period of

2π, we can use a Fourier series expansion in θ as follows [43]:

$$f_p(r,\theta) = \sum_k f_k(r) e^{jk\theta} \tag{2.27}$$

$$f_k(r) = \frac{1}{2\pi} \int_0^{2\pi} f_p(r,\theta) e^{-jk\theta} d\theta \tag{2.28}$$

where $f_k(r)$ is the k-th circular harmonic function (CHF) of $f_p(r,\theta)$.

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

$$h(r,\theta) = \sum_k h_k(r) e^{jk\theta} \tag{2.29}$$

where

$$h_k(r) = \frac{1}{2\pi} \int_0^{2\pi} h(r,\theta) e^{-jk\theta} d\theta \tag{2.30}$$

Then, the correlation function between $f(x,y)$ and $h(x,y)$ is shown in Eq. (2.31).

$$\begin{aligned} c &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) h^*(x,y) dx dy \\ &= \int_0^{2\pi} d\theta \int_0^{\infty} r dr f_p(r,\theta) h^*(r,\theta) \\ &= \int_0^{\infty} r dr \int_0^{2\pi} \left[\sum_k f_k(r) e^{jk\theta} \cdot \sum_l h_l^*(r) e^{-jl\theta} \right] d\theta \end{aligned} \tag{2.31}$$

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:

$$c = \sum_{k=-\infty}^{\infty} C_k \tag{2.32}$$

where

$$C_k = 2\pi \int_0^{\infty} f_k(r) h_k^*(r) r dr \tag{2.33}$$

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

$$c(\phi) = \sum_{k=-\infty}^{\infty} C_k e^{jk\phi} \tag{2.34}$$

If we only use a single circular harmonic

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

then, the correlation in Eq. (2.31) is expressed as

$$c_s(\phi) = C_k e^{jk\phi} \tag{2.35}$$

The output's center intensity is a constant as shown in the Eq. (2.35). It is invariant to the rotation of the image. The same strategy can be used for RHF.

2.7 Moments

For a 2-D continuous function $f(x, y)$, the moment of order $(p + q)$ is defined as [36]:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{2.36}$$

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

The central moments are defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \overline{x})^p (y - \overline{y})^q f(x, y) dx dy \tag{2.37}$$

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 Eq. (2.37) becomes

$$\mu_{pq} = \sum_x \sum_y (x - \overline{x})^p (y - \overline{y})^q f(x, y) \tag{2.38}$$

A set of seven invariant moments can be derived from the second and third moments [36][47].

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

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{2.39}$$

where

$$\gamma = \frac{p + q}{2} + 1 \tag{2.40}$$

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

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.41}$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \tag{2.42}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.43}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.44}$$

$$\begin{aligned}\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{2.45}$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{2.46}$$

$$\begin{aligned}\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{2.47}$$

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

2.8 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

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

$$r_{lc}(f, g) = \frac{1}{M \times N} \sum_x \sum_y f(x, y) g(x, y) \tag{2.48}$$

where, $M \times N$ is the size of the images. If f is the reference watermark and g is the watermark extracted from the image, r_{lc} reflects the similarity between the two watermarks. One of the problems with the linear correlation is that the detection values are highly dependent on the magnitudes of the watermark pattern extracted from the image. Therefore, for many extraction methods, the watermark will not be robust against attacks, such as the brightness change of images [1].

2. Normalized correlation

   The problem of the linear correlation can be solved by normalizing the extracted watermark and the reference watermark to unit magnitude before computing the inner product between them.

$$r_{nc}(f, g) = \sum_x \sum_y \tilde{f}(x, y)\tilde{g}(x, y) \tag{2.49}$$

where

$$\begin{cases} \tilde{f}(x, y) & = & \frac{f(x,y)}{\sqrt{\sum_x \sum_y f(x,y)^2}} \\ \tilde{g}(x, y) & = & \frac{g(x,y)}{\sqrt{\sum_x \sum_y g(x,y)^2}} \end{cases} \tag{2.50}$$

We refer to Eq. (2.49) as the normalized correlation. However, the 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 [1].

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.

$$r_{cc}(f,g) = r_{nc}(\hat{f}(x,y), \hat{g}(x,y)) \tag{2.51}$$

where

$$\begin{cases} \hat{f}(x,y) &= f(x,y) - \overline{f}(x,y) \\ \hat{g}(x,y) &= g(x,y) - \overline{g}(x,y) \end{cases} \tag{2.52}$$

where, $\overline{f}$ and $\overline{g}$ are the means of f and g, respectively. Because the mean of an image has been subtracted, the correlation coefficient will be robust against changes in the DC term of a work [1].

2.9 Filters

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 normalized cross-correlation function [48]:

$$C(i,j) = \frac{\sum_x \sum_y f(x-i, y-j) g(x,y)}{\sqrt{\sum_x \sum_y (f(x-i, y-j)^2}} \tag{2.53}$$

If the template matches the image exactly, at a translation (i_0, j_0), the cross-correlation will have its peak at (i_0, j_0). The major disadvantage of the cross-correlation method is being time-consuming. According to the correlation theorem, the Fourier transform of the correlation of the two images is the product of the Fourier transform of the correlation of the one image and the complex conjugate of Fourier transform of

the other.

$$C = \mathfrak{F}^{-1}[F(w_\rho, w_\theta) \cdot G^*(w_\rho, w_\theta)] \tag{2.54}$$

where

$$\begin{cases} F(w_\rho, w_\theta) &= \mathfrak{F}(f(\rho, \theta)) \\ &= A_F(w_\rho, w_\theta) e^{-j\Phi_F(w_\rho, w_\theta)} \\ G(w_\rho, w_\theta) &= \mathfrak{F}(g(\rho, \theta)) \\ &= A_G(w_\rho, w_\theta) e^{-j\Phi_G(w_\rho, w_\theta)} \end{cases} \tag{2.55}$$

and $*$ is the complex conjugate. $G(w_\rho, w_\theta)$ is called a matching filter. Here $A(\bullet)$ and $\Phi(\bullet)$ represent the magnitude and the phase components respectively.

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

1. Classical matched filter

$$G(\omega_\rho, \omega_\theta) = A_G(\omega_\rho, \omega_\theta) e^{-j\Phi_G(\omega_\rho, \omega_\theta)} \tag{2.56}$$

2. Amplitude-only filter

$$G_A(\omega_\rho, \omega_\theta) = A_G(\omega_\rho, \omega_\theta) \tag{2.57}$$

3. Inverse filter

$$G_I(\omega_\rho, \omega_\theta) = \frac{e^{-j\Phi_G(\omega_\rho, \omega_\theta)}}{A_G(\omega_\rho, \omega_\theta)} \tag{2.58}$$

4. Phase-only filter

$$G_{\Phi}(\omega_{\rho}, \omega_{\theta}) = e^{-j\Phi_G(\omega_{\rho}, \omega_{\theta})} \tag{2.59}$$

5. Binary Phase-only filter

$$G_B(\omega_{\rho}, \omega_{\theta}) = e^{-j\Phi_B(\omega_{\rho}, \omega_{\theta})} \tag{2.60}$$

where

$$\Phi_B(\omega_{\rho}, \omega_{\theta}) = \begin{cases} 0^{\circ} & G_r \geq 0 \\ 180^{\circ} & G_r < 0 \end{cases} \tag{2.61}$$

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

All these different filters listed can be used for image registration. In order to fairly compare various filters, two different criteria can be used. They are noise robustness and sharpness of the correlation peak [49].

1. Peak sharpness

   In order to compare the sharpness of correlation peaks, the peak-to-correlation energy (PCE) is calculated [49]:

$$PCE = 10 \log_{10} \frac{|C_0|^2}{CPE} \tag{2.62}$$

with

$$CPE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |C(i,j)|^2}{M \times N} \tag{2.63}$$

where, $C(i,j)$ is the cross-correlation function between the template and the power spectrum of watermarked image in the LPM domain, and C_0 is the maximum value of $C(i,j)$. The template is a small part of the power spectrum of the original image in the LPM domain. The higher the value of PCE, the sharper the peak value of cross-correlation comparing to the sidelobe.

2. Noise robustness

   Another criterion is the noise robustness, which corresponds to the optimization of the signal-to-noise ratio (SNR) [49],

$$SNR = 10\log_{10}\frac{|C_0|^2}{MSE} \tag{2.64}$$

   with

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|C'(i,j)|^2}{M \times N} \tag{2.65}$$

   where $C'(i,j)$ 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.

2.10 Phase correlation

The phase correlation is another efficient approach to rectifying the watermark position to avoid exhaustive search [50]. [51] proposed the phase correlation based on the property of the Fourier transform given by the shift theorem. Given an image f_0 and

its shift version f_1 with the displacement (x_0, y_0), i.e.,

$$f_1(x, y) = f_0(x - x_0, y - y_0) \tag{2.66}$$

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

$$F_1(u, v) = e^{-j(ux_0+vy_0)} F_0(u, v) \tag{2.67}$$

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

$$C = \frac{F_1(u, v)F_2^*(u, v)}{|F_1(u, v)F_2^*(u, v)|} = e^{j(ux_0+vy_0)} \tag{2.68}$$

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.

$$D = \mathfrak{F}^{-1}(angle(C)) \tag{2.69}$$

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 function $\delta(x - d)$ is e^{-jwd}. Eq. (2.69) 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 corner 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 [52]. 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. Therefore, the shape matrix M is formed for each pixel [30]:

$$M(i,j) = \begin{bmatrix} \sum_{m,n}(G'_x(m,n))^2 & \sum_{m,n} G'_x(m,n)G'_y(m,n) \\ \sum_{m,n} G'_x(m,n)G'_y(m,n) & \sum_{m,n}(G'_y(m,n))^2 \end{bmatrix} \tag{2.70}$$

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 [30]:

$$H(i,j) = det(M(i,j)) - k \cdot trace(M(i,j)) \tag{2.71}$$

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

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

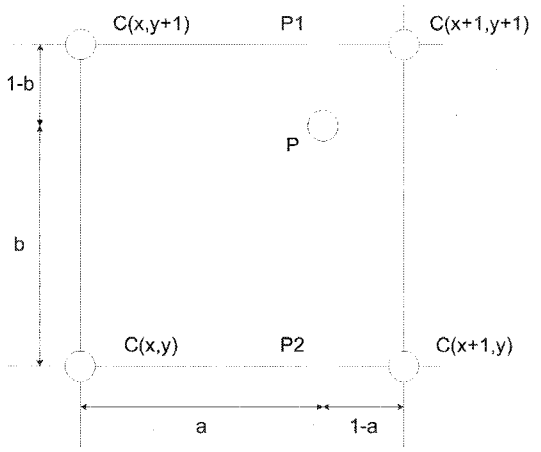


Figure 2.6: Bilinear interpolation.

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 most memory and computing time. *bilinear* is faster and less memory-intensive than *bicubic*, and produces the smoother surface than *nearest*. Therefore, most applications use the bilinear interpolation.

Here, we give an example about the bilinear interpolation during log-polar mapping. Each point in log-polar amplitude spectrum is computed from a weighted average of four points in the Cartesian amplitude spectrum, shown in Eq. (2.72) and Fig. 2.6.

$$\begin{aligned} P(\rho,\theta) &= C(x,y)\cdot(1-a)\cdot(1-b) \\ &+ C(x,y+1)\cdot(1-a)\cdot b \\ &+ C(x+1,y)\cdot a\cdot(1-b) \\ &+ C(x+1,y+1)\cdot a\cdot b \end{aligned} \tag{2.72}$$

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

Chapter 3

Literature review

As stated in Section 1, a digital image watermarking algorithm must be robust against a variety of possible attacks. In recent years, watermarking algorithms robust to geometrical distortions have been the focus of research. Most of the proposed geometrical-transform-invariant algorithms are actually only RST invariant due to the fact that changing the image size or its orientation, even by slight amount, could dramatically reduce the receiver's ability to retrieve the watermark. Meanwhile, the systematic analysis of the watermarking algorithm performance under geometrical distortion has begun to draw great attention. Most of these efforts confine to theoretically analyzing and quantifying the effect of the global affine transform to the performance of the watermarking algorithms. Though the local distortion are more and more regarded as a necessary benchmark test scenario, the theoretical analysis of its effect on the watermark detection performance remains untouched because of its complexity.

Based on the theories and techniques that are used, the proposed RST watermarking algorithms in literature can be generally categorized into seven groups [53]:

1. RST invariant domain based algorithms;

2. Radon transform based algorithms;

3. Template based algorithms;

4. Salient features based algorithms;

5. Image decomposition based algorithms;

6. Stochastic analysis based algorithms; and

7. Others.

It is worth noting that some watermarking algorithms may belong to two or more categories mentioned above. The classification of the watermarking algorithms is only for the convenience of discussion.

3.1 RST invariant domain based algorithms

This category includes the watermarking algorithms that embed watermark in domains that are invariant to geometric attacks. It is well known that the Fourier transform has the property of translation invariance. As shown in Section 2.4, the Fourier-Mellin transform (FMT) has the property of rotation and scaling invariance. Therefore, once the DFT and the FMT are applied to the image, the image will be transformed to the RST invariant domain. The watermark embedded into this domain can be RST invariant. In Section 3.1.1, we will introduce the Fourier-Mellin transform (FMT) based algorithms [4] [54]. Because of the implementation difficulties, some modified algorithms based on the FMT were proposed. Most of them used the log-polar mapping (LPM) instead of the FMT. The LPM domain is not a truly RST invariant domain. However, rotation and scaling in the spatial domain can only result in a translation in

the LPM domain, which simplifies the watermark detection dramatically. Furthermore, the translation in the LPM domain can be identified easily using the image registration related techniques or the 1-D projection. Then, the watermarked image can be re-synchronized to detect the watermark. Because these algorithms are derived from the FMT, we discuss them in this section. And, the LPM domain and 1-D projection can also be called semi-RST invariant domain.

Two watermarking algorithms [55][56] using phase correlation or phase-only matching filtering to re-synchronize the watermarked image are addressed in Sections 3.1.2 and 3.1.3. In Section 3.1.4, a watermark algorithm based on 1-D projection and LPM proposed by [38] is discussed.

3.1.1 Fourier-Mellin-transform-based algorithms

The authors of [4] first outlined the theory of integral transform invariants and showed the watermark can be resistant to RST, if it is embedded in Fourier-Mellin domain. The theoretical RST watermark embedding and extraction processes are shown in Fig. 3.1 and Fig. 3.2.

The theoretical FMT-based watermark embedding process can be summarized in the following steps:

1. Apply the DFT to the original image;

2. Apply the FMT to the DFT magnitude of the original image;

3. Embed the watermark in the resulting RST invariant domain;

4. Compute the inverse FMT by using the original phase;

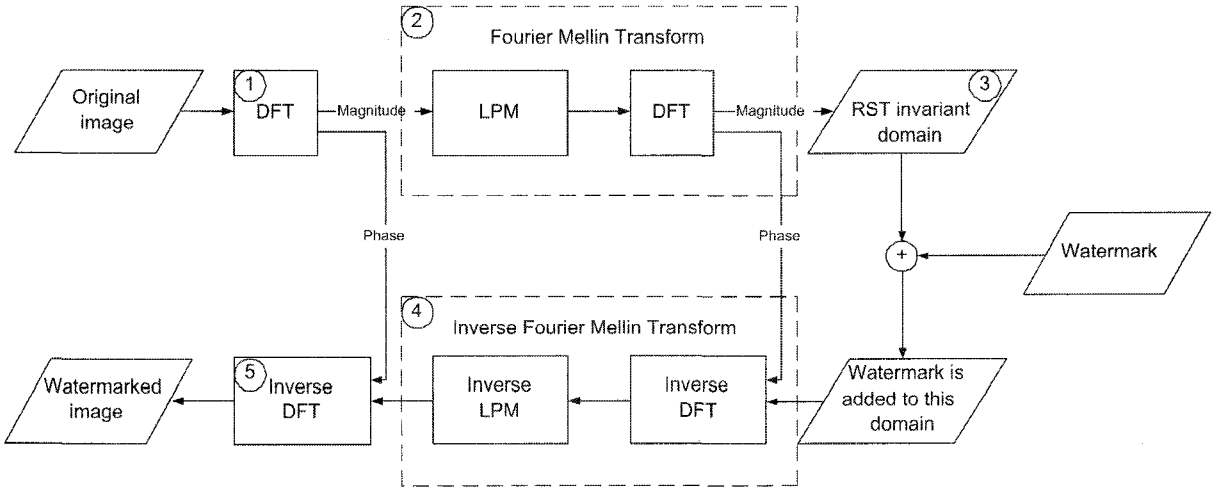


Figure 3.1: The FMT-baed watermark embedding scheme.

5. Compute the inverse DFT to obtain the watermarked image by using the original phase.

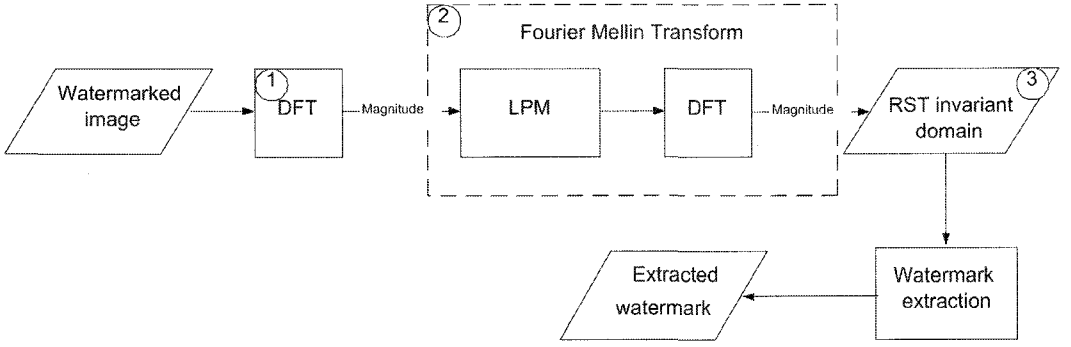


Figure 3.2: The FMT-based watermark extraction scheme.

Refer to Fig. 3.2, the theoretical FMT-based watermark extraction process can be carried out as follows:

1. Apply the DFT to the watermarked image;

2. Apply the FMT to the DFT magnitude of the watermarked image;

3. Extract the watermark from the RST invariant domain.

However, the authors of [4] noted very severe difficulties in implementation, which might hamper the further work in this area[38]. Interpolation is needed to accomplish the changing of coordinate system. Meanwhile, it is also experimentally found out that the LPM and ILPM will cause an unacceptable loss of image quality as shown in Fig. 2.4 (b), even when the bilinear interpolation is used.

Trying to solve the problems stated above, the authors of [4] proposed to embed watermark in the RST invariant domain independently of the original image, so that the original image can avoid sufferring from the quality degradation caused by LPM and ILPM. The proposed optimized Fourier-Mellin transform based watermarking schemes are shown in Fig. 3.3 and Fig. 3.4.

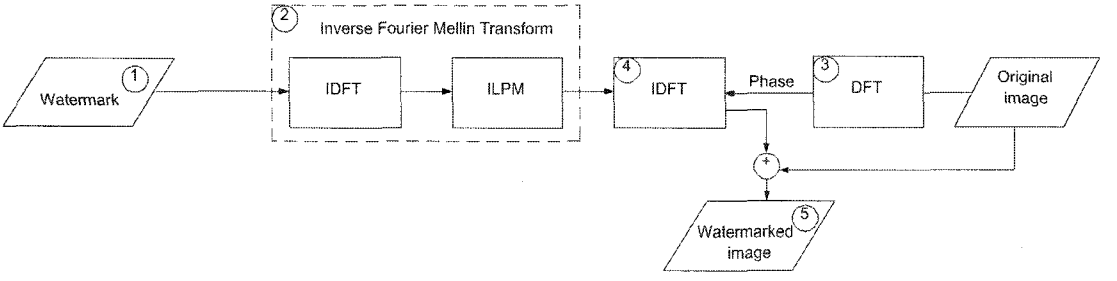


Figure 3.3: The optimized FMT-based watermark embedding process.

Refer to Fig. 3.3, the optimized watermark embedding scheme was carried out in the following steps [4]:

1. Generate a 2D spread spectrum watermark signal;

2. Apply the inverse FMT to the 2D watermark;

3. Apply the DFT to the original image to get the phase information;

4. Apply the inverse DFT to the result of Step (2) using the the phase of the original image from Step (3);

5. Add the result of Step (4) to the original image to achieve the watermarked image.

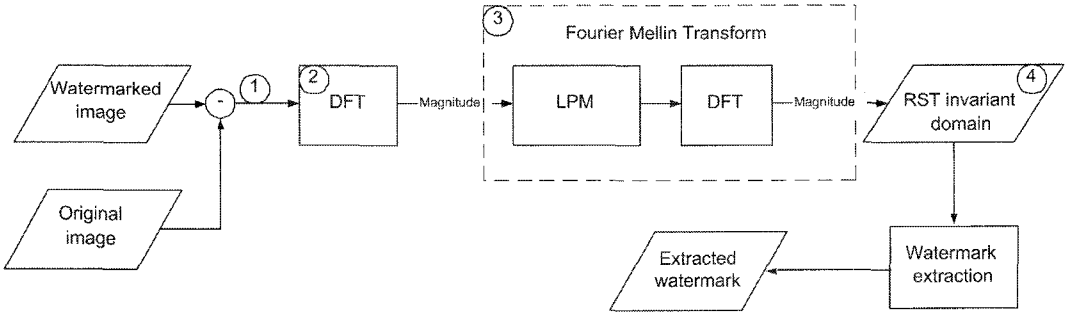


Figure 3.4: The optimized FMT-based watermark extraction process.

As shown in Fig. 3.4, the optimized watermark extraction of the works as follows [4]:

1. Compute the difference between the original image and the watermarked image;

2. Apply the DFT to the difference obtained in Step (1);

3. Apply the FMT to the magnitude spectrum of the DFT obtained in Step (2);

4. Extract the watermark in the RST invariant domain obtained from Step (3).

To avoid quality degradation caused by LPM and ILPM, another RST invariant watermarking scheme was proposed in [54]. This algorithm employed invariant centroid and reordered Fourier-Mellin transform. Unlike O'Ruanaidh's algorithm, the authors in [54] performed LPM on image in the spatial domain instead of in the frequency

domain. A calculation method of the invariant centroid at the origin of the LPM is proposed to guarantee the scaling and translation invariance of an image. The centroid $C = (C_x, C_y)$ of an image $f(x, y)$ is calculated as:

$$\begin{cases} C_x &= \sum_x \sum_y f'(x,y)x \\ C_y &= \sum_x \sum_y f'(x,y)y \end{cases} \tag{3.1}$$

where

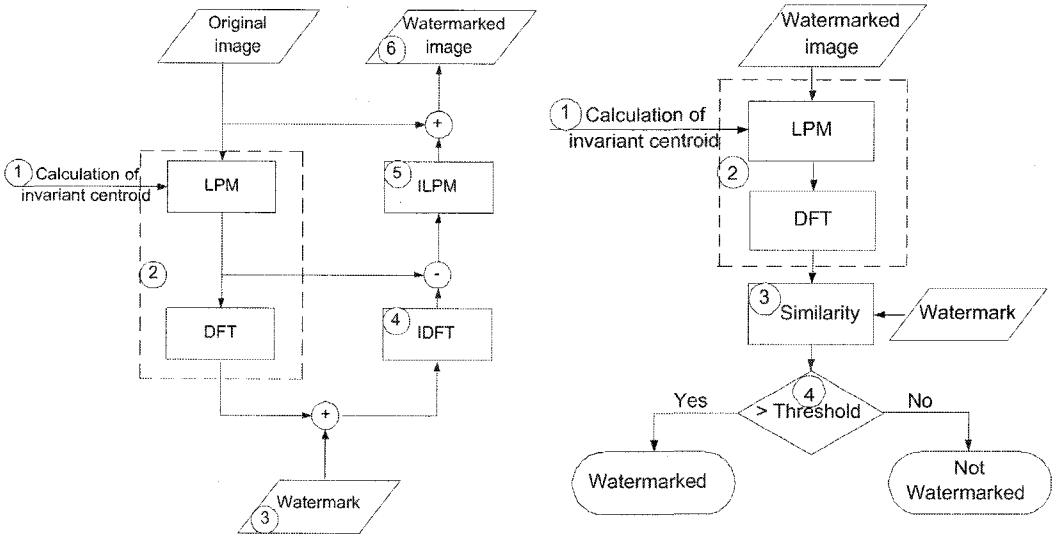
$$f'(x,y) = \frac{f(x,y)}{\sum_x \sum_y f(x,y)} \tag{3.2}$$

The invariant centroid is computed in following steps:

1. Apply low-pass filtering to the original image;
2. Obtain the initial centroid C_0 by using Eq. (3.1);
3. Calculate the centroid C_1 based on a circular region with radius r and center point C_0;
4. Calculate the centroid C_2 based on a circular region with radius r and center point C_1;
5. If C_2 equals C_1, then the invariant centroid is found; otherwise, repeat Step (4) until the centroid converges at the same point.

The rotation of an image in Cartesian coordinates results in a cyclic shift in the LPM domain. Therefore, the magnitude spectrum of the 2-D DFT performed on the LPM image is rotation invariant.

Refer to Fig. 3.5 (a), the watermark embedding includes the following steps:



(a) Embedding. (b) Detection.

Figure 3.5: Kim's watermarking algorithm.

1. Calculate the invariant centroid of the original image;

2. Compute the log-polar mapping and DFT of the original image by using the invariant centroid, obtained from Step (1), as the origin;

3. Generate a binary pseudo-random sequence as a watermark;

4. Compute the inverse DFT of the watermarked image after embedding the watermark in the frequency domain;

5. Compute the inverse LPM after removing the LPM magnitude of the original image to avoid destroying the quality of the original image by the ILPM;

6. Add the original image after Step (5) to obtain the watermarked image.

The watermark detection process is shown in Fig. 3.5 (b) and includes the following steps:

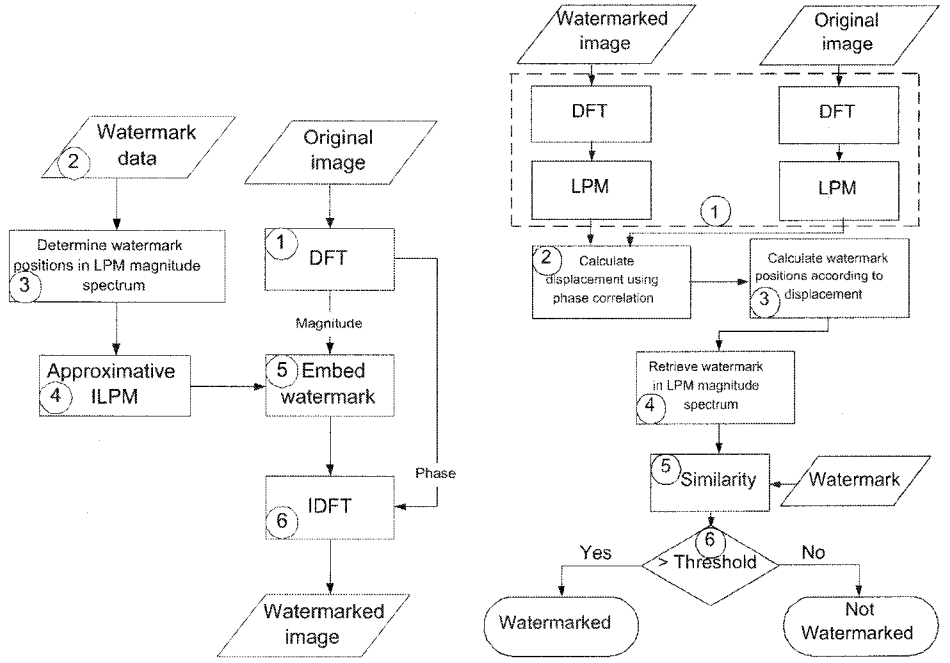
1. Calculate the invariant centroid of the watermarked image;
2. Compute the log-polar mapping and DFT of the watermarked image by using the invariant centroid, as the origin;
3. Calculate the similarity between the DFT magnitude of the watermarked image and the watermark data;
4. Compare the similarity with the predefined threshold to judge if the test image is watermarked.

This method avoids letting the original image go through the LPM and ILPM. The inverse LPM of the watermark signals are embedded to the original image to avoid image quality loss. However, the watermark data still needs to go through the inverse LPM, which will damage the watermark data.

3.1.2 Phase correlation and log-polar mapping based algorithm

A new FMT-based watermarking scheme was proposed in [55]. The watermark was embedded into LPM domain to simplify the RST transformations to shifts (refer to Eq. (2.18)). Middle frequency regions in the LPM magnitude spectrum were selected for watermark embedding. And, refer to Fig. 3.6(a), the approximate ILPM is employed to replace the ILPM in order to eliminate the imprecision caused by the ILPM. The watermark locations in the Cartesian coordinates of the DFT magnitude spectrum are approximated from the watermark locations in the LPM domain. Therefore actually watermarks are embedded in the Fourier magnitude spectrum of the original image, to achieve the effect of being embedded in the LPM domain.

The embedding process includes the following steps (refer to Fig. 3.6 (a)):



(a) Embedding. (b) Detection using phase correlation.

Figure 3.6: Zheng's watermarking algorithm.

1. Apply the DFT to the original image;
2. Use the Pseudo-random Noise (PN) generator to generate a watermark data sequence;
3. Select the middle frequency regions in the LPM magnitude spectrum for embedding the watermark data sequence;
4. Compute the approximative ILPM;
5. Embed the watermark into the DFT domain;
6. Apply the inverse DFT to get the watermarked image.

Since they do not apply the IDFT before the approximate ILPM, rotation and scaling operation in the spatial domain of the watermarked image will cause a translation of the watermark positions in the LPM domain, either a circular shift along the angle θ axis or the vertical shift along the log-radius ρ axis (refer to Eq. (2.18)).

If the original image is unavailable, an exhaustive search in the embedding area is used to handle the shift of watermark positions caused by rotation and scaling [57]. The detection steps are:

1. Compute the DFT and LPM of the watermarked image;
2. Compute the similarity between the watermark data and the data retrieved from the LPM magnitude of the watermarked image by exhaustive search;
3. Compare the highest similarity against the predefined threshold to judge if the image is watermarked.

However, the exhaustive search is time-consuming and produces large correlation coefficient for unwatermarked images. Therefore, [50] uses the phase correlation to rectify the watermark position to avoid exhaustive search when the original image is available. The watermark extraction process using phase correlation includes the following steps (refer to Fig. 3.6 (b)):

1. Compute the DFT and LPM of the watermarked image and the original image, respectively;
2. Calculate the displacement by using the phase correlation (refer to Section 2.10);
3. Calculate watermark positions according to the displacement;
4. Retrieve watermark at the calculated position in the LPM magnitude spectrum.

5. Compute the similarity between the watermark data and the retrieved watermark data;

6. Compare the similarity with the predefined threshold to judge if the test image is watermarked.

This algorithm is very reliable in displacement calculation and is invariant to translation, reasonable-range scaling and rotation of any angles. Furthermore, it is also very robust to JPEG compression and other attacks.

The drawback of this phase correlation based algorithm is that it needs the original image for watermark extraction.

3.1.3 Phase-only filtering and log-polar mapping based algorithm

The authors of [56] proposed an image rectification algorithm that can be used by any image watermarking algorithms to provide robustness against RST transformations. This algorithm utilized the properties of the LPM domain that rotation and scaling transformations in the spatial domain result in cyclically translational shifts in the LPM of the magnitude of the Fourier transform spectrum of an image.

A small block is cut from the LPM domain as matching Template 1. If embedding watermark in the spatial domain, another block is cut from the spatial domain as Template 2. At the receiver side, the cross-correlation is computed between Template 1 and the LPM magnitude of the RST transformed image to detect the rotation and scaling parameters. The same strategy is employed in the spatial domain to detect the translation parameters. The cost of the templates is low and the templates can also be compressed. Fig. 3.7 shows where the matching templates are cut.

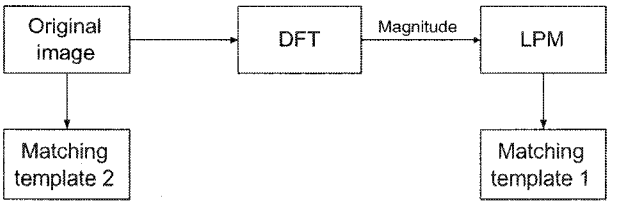


Figure 3.7: Matching templates.

As we mentioned in Section 2.9, there are five traditional filters used for the cross-correlation computation. Normally, the phase-only filter could give a very good result because the phase information is considerably more important than the amplitude information in preserving the visual intelligibility of the image [37]. However,according to the experimental results [58], all these five types of traditional filters fail to produce acceptable discrimination when rotation or scaling or both applied to the watermarked image.

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 [59] [60]. However, in most cases, the experimental targets are the real images, the power spectrums of which are not white. [60] 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. As a result, [56] proposed a new filtering method, named the phase-only filtering method, to transform the non-white spectrum in the LPM domain of an image to a mapping with a unity power spectrum, and then apply a phase-only filter to the resulting mapping [61]. The filtering process

is described as follows:

$$C = \Im^{-1}[F_\Phi(w_\rho, w_\theta) \cdot G^*_\Phi(w_\rho, w_\theta)] \tag{3.3}$$

where

$$F_\Phi(w_\rho, w_\theta) = e^{-j\Phi_F(w_\rho, w_\theta)} \tag{3.4}$$

The experimental results demonstrate that the phase-only filtering method is most robust against the noise, and has the highest peak energy (refer to Section 2.9) comparing to the sidelobe.

The RST parameters can be calculated, and the transformed watermarked image can be rectified in the following steps [56] (refer to Fig. 3.8):

1. Compute the DFT and LPM of the watermarked image;

2. Calculate the cross-correlation between Template 1 and the LPM spectrum of the watermarked image;

3. Based on the detected location of Template 1 and the original location of Template 1, the shift in the log-polar domain can be computed. Then based on the inverse log-polar mapping Eq. (2.19) and (2.20), the rotation and scaling parameters can be computed. Suppose the log-polar mapping is 512×512 and the detected shift is Δ_ρ along the ρ axis and is Δ_θ along the θ axis in the LPM domain, the rotation degree ϕ and the scaling ratio c in the spatial domain can be computed as:

$$\phi = \frac{\Delta\theta}{512} \times 2\pi \tag{3.5}$$

$$c = e^{(\Delta\rho \times \ln\sqrt{256^2+256^2})/(512-1)} \tag{3.6}$$

4. Rectify the watermarked image by using detected rotation and scaling parameters;

5. If the watermark is embedded in the spatial domain, calculate the cross-correlation between Template 2 and rectified watermarked image. The location of Template 2 can be detected by locating the highest peak in the cross-correlation plane;

6. The difference between the detected location of Template 2 and the original location of Template 2 is the translation parameters in the spatial domain;

7. Rectify the watermarked image obtained from Step (4) by using translation parameter to get the rectified watermarked image for watermark detection.

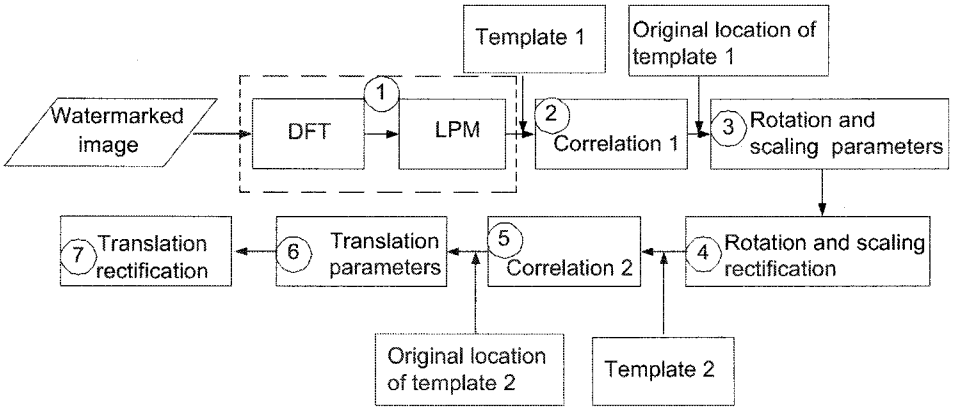


Figure 3.8: RST parameters detection and image rectification.

Three applications of the rectification algorithm show its effectiveness. In these three applications, the watermark is embedded in the spatial domain, the magnitude of the Fourier domain and the log-polar mapping of the Fourier transform of the image respectively.

In Application I, the watermark is embedded into the spatial domain of the image and the RST parameters are detected by using the two templates. Then the image is re-synchronized according to the detected RST parameters before the watermark detection. In Application II, the watermark is embedded into the magnitude of the Fourier transform domain of the image. Because the magnitude of the Fourier transform of the image is independent of the translational parameters, only the rotation and scaling parameters are needed for rectifying the image. In Application III, the watermark sequence is embedded into the log-polar domain of the image. The advantage of this application is that the watermark position is rectified directly in the log-polar domain of the image instead of the geometrical rectification of the image in spatial domain to avoid the imprecision by interpolation during log-polar mapping. The approximate inverse LPM instead of the real one is used to avoid the distortion.

3.1.4 One-dimensional projection and log-polar mapping based algorithm

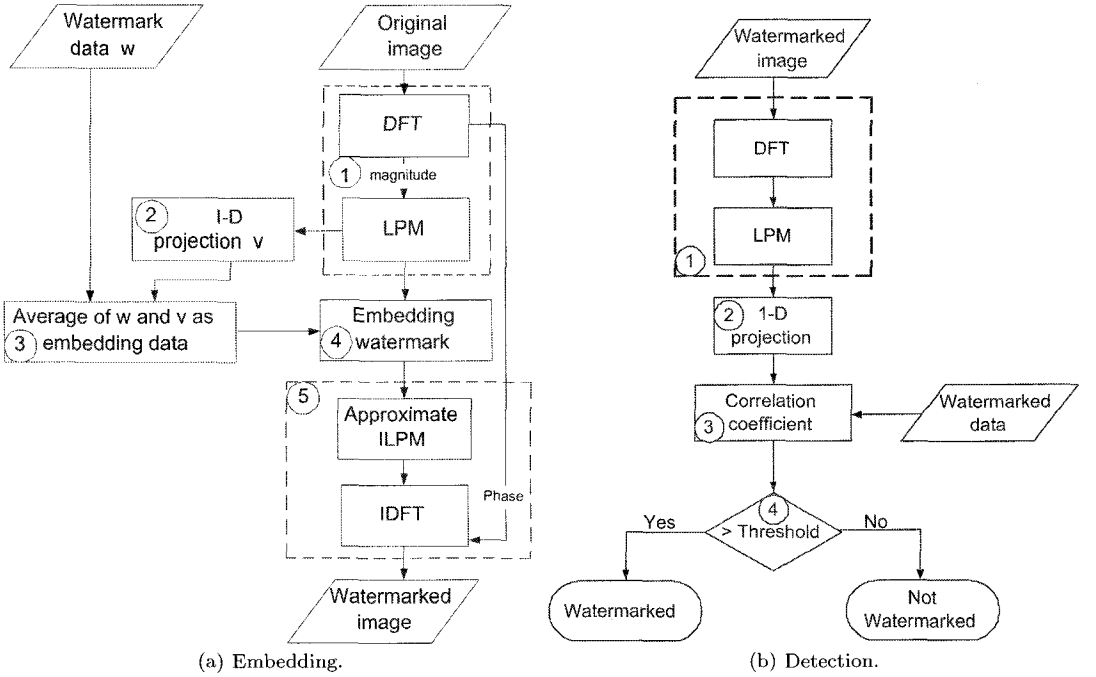
As shown in Section 2.3, the scaling and rotation in spatial domain only result in a translation in the LPM domain. The 1-D projection can be used to simplify the problem here. The image $f(x, y)$ can be transformed to LPM domain $|F(\rho, \theta)|$. Then, we can define $g(\theta)$ to be a 1-D projection of $|F(\rho, \theta)|$ [38]:

$$g(\theta) = \sum_{j} |F(\rho_j, \theta)| \tag{3.7}$$

The $g(\theta)$ is invariant to both translation and scaling. Rotations result in a circular shift of the values of $g(\theta)$ [38].

As mentioned in Section 2.5, the Radon transform is the 1-D projection along an

angle. So the $g(\theta)$ is actually a special case of the Radon transform of the LPM spectrum of the Fourier magnitude of an image. It has the property of being invariant to both translation and scaling of an image. Rotations result in the circular shift of the values of $g(\theta)$. The algorithm proposed by [38] is based on this principle. They embed the watermark into a 1-D signal by taking the Fourier transform of the image, re-sampling the Fourier magnitudes into the log-polar coordinates, and then summing a function of those magnitudes along the log-radius axis. The watermark embedding process includes the following steps (refer to Fig. 3.9 (a)):



(a) Embedding. (b) Detection.

Figure 3.9: Lin's watermarking algorithm.

1. Compute the DFT and LPM of the original image;
2. Calculate the 1-D projection of the LPM spectrum of the original image to get a vector v;

3. Compute the weighted average of the randomly generated watermark data w and the vector v as the embedding data;

4. Embed the data obtained from Step (3) to the LPM spectrum of the original image;

5. Compute the approximate inverse LPM and inverse DFT with the original phase to get the watermarked image.

The watermark detection process consists of the following steps (refer to Fig. 3.9 (b)):

1. Compute the DFT and LPM of the watermarked image;

2. Calculate the 1-D projection of the LPM spectrum of the watermarked image to get a vector v';

3. Compute the correlation coefficient between the watermark data and the vector v';

4. Compare the correlation coefficient with the predefined threshold to judge if the image is watermarked.

Rotation is the only problem they need to deal with. They handle rotation by an exhaustive search. However, exhaustive search is time-consuming and it may produce a higher false positive probability. A false positive error occurs when the detector incorrectly indicates that a watermark is present [1]. In order to solve this problem, a rectification method can be employed to battle rotations [62].

In the algorithm of [38], the LPM and ILPM are both applied to the image and the watermark data. Because the ILPM is non-invertible, instead, they perform an iterative

approximation of the ILPM. As shown in Fig. 3.10. $\mathbf{F}'$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.

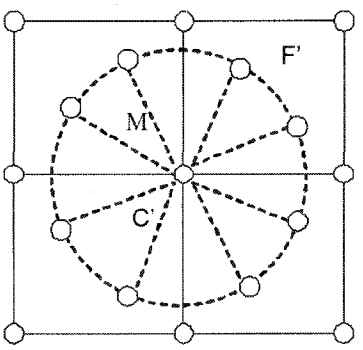


Figure 3.10: Approximate ILPM.

To get the value for $\mathbf{C}'$, they add the weighted difference between all the corresponding $\mathbf{F}'$ and $\mathbf{F}$ to $\mathbf{C}$ [38]. The relationship can be expressed by Eq. (3.8):

$$C'_j = C_j + \frac{M_{ij}(F'_i - F_i) + M_{kj}(F'_k - F_k)}{M_{ij} + M_{kj}} \tag{3.8}$$

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 cannot extract the exact inverse version $\mathbf{C}'$. The more times of iterations of this method, the closer to the desired inversion [38].

3.2 Radon transform based algorithms

As discussed in Section 2.5 and Section 3.1.4, the Radon transform is used to project an image in the LPM domain to a 1-D projection plane. The watermark embedded in this projection domain can be easily detected. Other algorithms try to exploit the geometrical transform invariance properties of the Radon transform, while embedding the watermark in the domains other than the projection domain to solve the problem. A Radon transform based digital image watermarking algorithm has been proposed in [42]. In the algorithm, the two generalized Radon transforms, CIT and RIT, were used to extract some characteristic values of the image, based on that the corresponding geometric transformation parameters can be calculated to re-synchronize the transformed watermarked image. The watermark is embedded in the spatial domain. Once the geometrically distorted watermarked image is transformed back to its original shape in terms of position, orientation and size, the detection of the watermark would be very straightforward.

Based on the properties of the RIT and CIT, the geometric transformation parameters can be calculated. As shown in Eq. (2.25) and Eq. (2.26), the computation origin $f(x_0, y_0)$ of the RIT and CIT is very important for the successful detection of the geometric transformation parameters. Corner points have been used for Radon transform computation.

The watermark embedding proposed in [42] is shown in Fig. 3.11 (a). It includes the follwing steps:

1. The corner detection algorithm is used to find the corner point.

2. The one most robust to possible attacks is used as the computation origin of the RIT and CIT. Applying the RIT and CIT to the image, the characteristic values

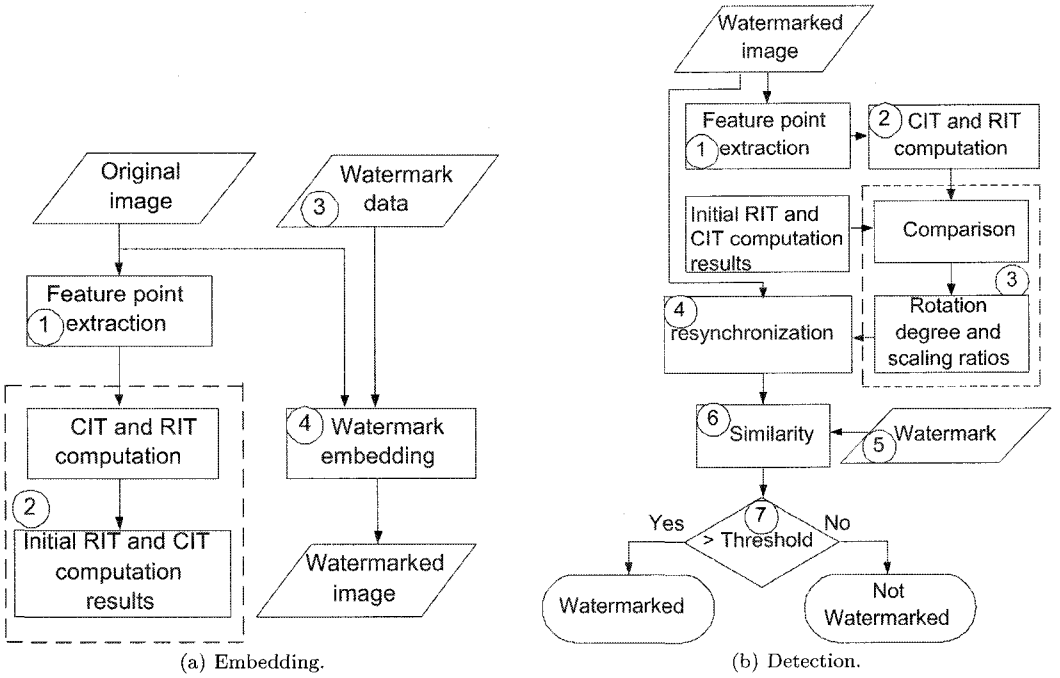


Figure 3.11: Simitopoulos's watermarking algorithm.

are computed (as shown in Fig. 4.7 and Fig. 4.8).

3. A random two-dimensional sequence of $+1$ and -1 is created using the PN generator. Each value of the sequence is spread in blocks with a size of $B \times B$. This block-based pattern will be used as the watermark.

4. The embedding strength of the watermark for each image pixel $X(i,j)$ is determined based on the local variance values in order to perform invisible watermark embedding. The watermark is embedded in the spatial domain representation of the image luminance.

$$X'(i,j) = X(i,j) + \alpha(i,j)W(i,j) \tag{3.9}$$

where $W(i,j)$ is the watermark bit and $\alpha(i,j)$ is the embedding strength. The analysis of watermark embedding algorithms shows that [63], coefficients with larger variance can be embedded with larger embedding strength.

$var(X(i,j))$ is the local variance value for each pixel. First a window centered at the pixel to be embedded is determined, then the local variance $var(X(i,j))$ for each pixel in this window area is calculated. Then the maximum variance is found and used for the normalization of the variances in the range $[0,1]$. Then the following equation is used to determine the embedding strength.

$$\alpha(i,j) = \frac{var(X(i,j))}{max(var(X(i,j)))}\alpha_{max} + \alpha_{min} \tag{3.10}$$

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 watermarking strength factor $\alpha(i,j)$.

The watermark detection process is shown in Fig. 3.11 (b). It was conducted as follows:

1. The corner detection algorithm is used to detect the corner point used for the RIT and CIT computation during the embedding process.

2. The RIT and CIT are applied to the possibly geometrically distorted watermarked image.

3. Based on the computed characteristic values, the geometric transformation parameters can be computed to transform the watermarked image to its original

shape. The scaling ratio of the CIT is the scaling ratio applied to the image. The CIT plot in Fig. 4.7 shows the change of the CIT computation after scaling. The Fig. 4.7 (a) is actually a scaled version of the Fig. 4.7 (b). Suppose the position of the highest peak in Fig. 4.7 (a) is CIT_a and the position of the highest peak in Fig. 4.7 (b) is CIT_b along the x axis, the scaling ratio can be computed as CIT_b/CIT_a. Also the RIT plot in Fig. 4.8 shows the change of the RIT computation after rotation. The Fig. 4.8 (a) is actually a shifted version of the Fig. 4.8 (b). Suppose the position of the highest peak in Fig. 4.8 (a) is RIT_a and the position of the highest peak in Fig. 4.8 (b) is RIT_b along the x axis, the rotation can be computed as $(RIT_b - RIT_a) \times 2\pi/N$. Here N is the number of sampling points of the RIT computation.

4. Re-synchronize the watermarked image by using detected geometric transformation parameters.

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

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

$$c = \frac{1}{N} \sum_i \sum_j (X'(i,j) - \overline{X'(i,j)}) W(i,j) \tag{3.11}$$

where N is the size of the image.

The local mean $\overline{X'(i,j)}$ of each pixel of the test image is calculated as:

$$\overline{X'(i,j)} = \frac{1}{M}\sum_{m}\sum_{n} X'(i,j) \tag{3.12}$$

The local mean is computed in a window area. 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 $W(i,j)$.

7. The correlation value is compared with the predefined threshold to give the judgement of the existence of the watermark.

3.3 Template based algorithms

Another strategy for detecting watermarks after geometric distortions 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 [19][64][65][66][67][23][68]. The researchers 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. The approaches are quite similar to each other. Therefore, in this section, we will only introduce the approach proposed by [23] as an example.

The authors in [23] use approximately 14 points along two lines that 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 [23]. Therefore, the transformations could be detected through the relationship

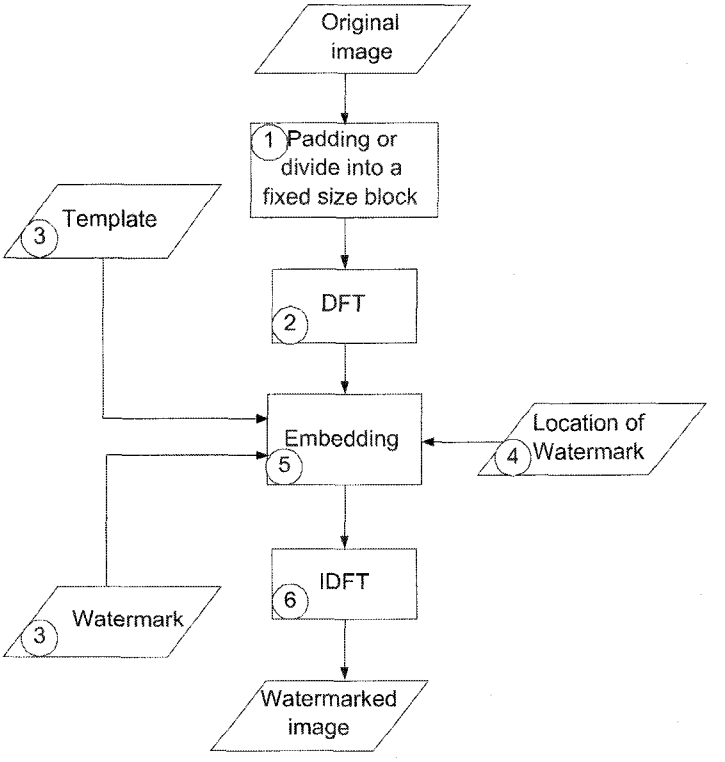


Figure 3.12: Watermark embedding procedures of Pereira's method.

of two lines. Once the template is detected, these transformations are inverted and the spread spectrum signal is retrieved.

Refer to Fig. 3.12, the embedding process includes the following steps:

1. Zero-padding the original image to a block of a fixed size if the original image is smaller or divide it into a block if the original image is bigger;

2. Compute the DFT of the above block;

3. Choose templates of approximately 14 points along two lines in the DFT domain;

4. Pseudo-randomly generate a sequence of points as the locations of watermarks;

5. Embed the watermark by changing the value of above locations and embed the template by adding to the Fourier domain of the original image;

6. Compute the inverse DFT to get the watermarked image.

The watermark extraction process includes two phases. The first one is the template detection, refer to Fig. 3.13, which includes the following steps:

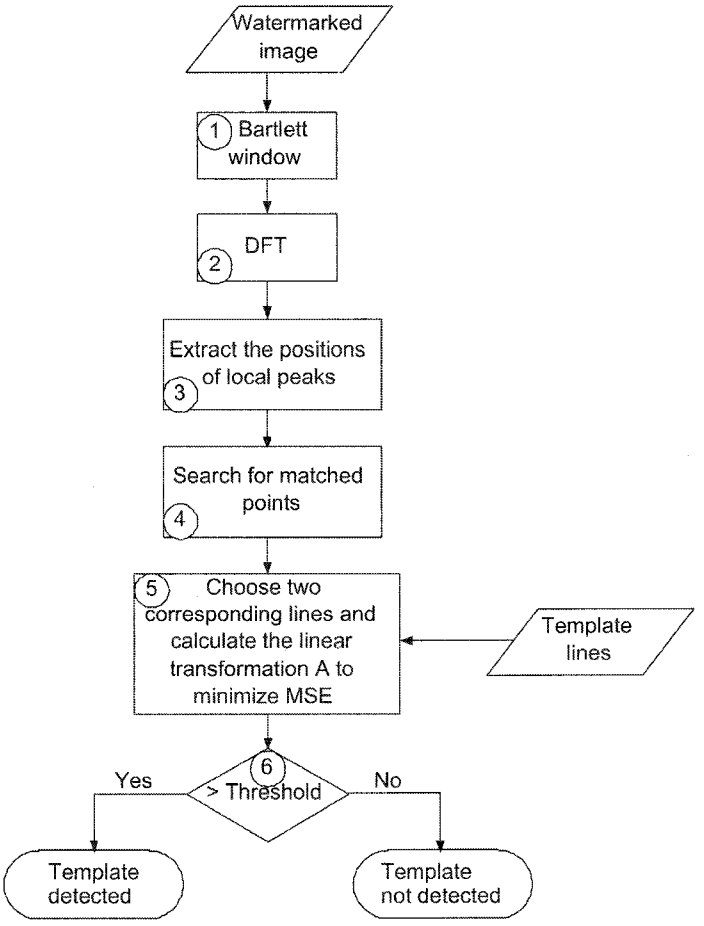


Figure 3.13: Template detection procedures of Pereira's method.

1. Apply a Bartlett window to the spatial domain of the watermarked image in order to eliminate artifacts of the implicit assumption of periodicity in the image during calculation of the DFT;

2. Compute the DFT of the above image;

3. Extract the positions of the local peaks (P_{x_i}, P_{y_i}) and map them to polar coordinates (r_{I_i}, θ_{I_i});

4. Sort the peaks by angle and divide them equally into N_b space bins by angle. For each bin, search for a K, so that at least N_m points match between the points r_{I_i} that are the radial coordinates of the points in bin i, and r_{T_i} that are the radial coordinates of the template along one of embedded template line by using equation: $|r_{I_i} - K_{r_{T_i}}| < threshold$. If at least N_m points match, then store these matched points;

5. Choose two sets of matched points corresponding to template line 1 and line 2, respectively, and calculate the linear transformation A to minimize MSE;

6. Compare the MSE with the predefined threshold to judge if the template is present.

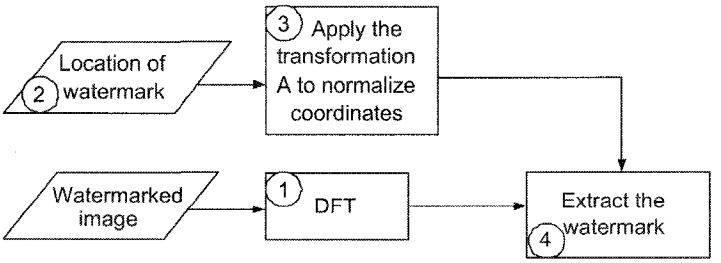
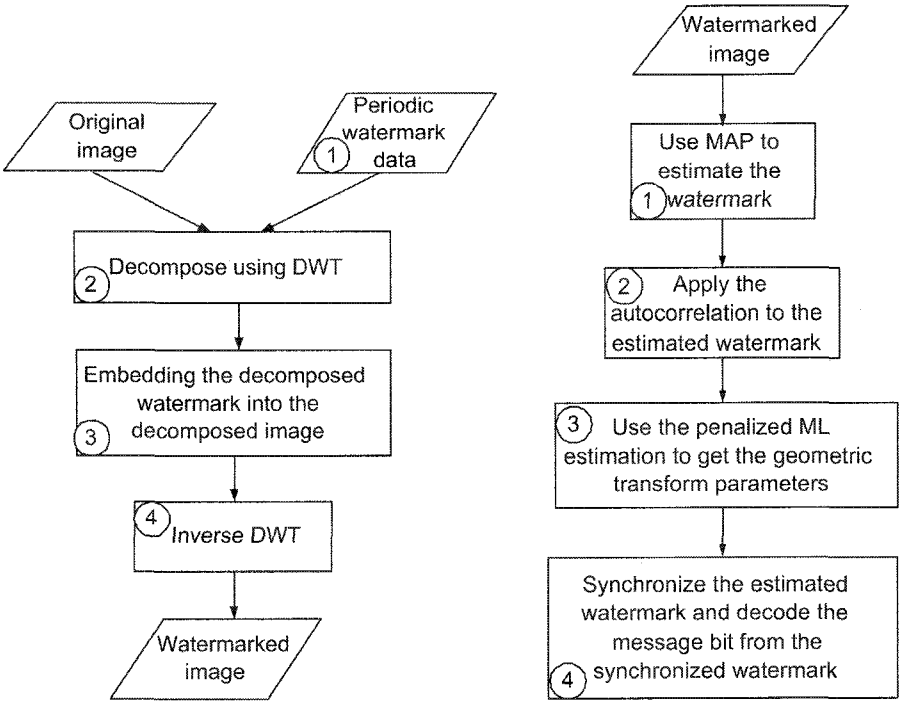


Figure 3.14: Watermark extraction procedure of Pereira's method.

After the template is detected, the watermark could be extracted. As shown in Fig. 3.14, the watermark extraction process includes the following steps:

1. Compute the DFT of the watermarked image;
2. Generate the position coordinates of the watermark data by the secret key used during embedding;
3. Apply the transformation matrix detected by the template detection process, to normalize the coordinates of the watermark position;
4. Extract the watermark from the transformed coordinates.

Because the traditional template based watermarking algorithms are easy to be attacked, some researchers came up with the new idea that the watermark bears with not only the copyright information but also the geometrical information about the original image. The watermark does not concentrate a strong energy into several points so that it is hard to be recognized by the attackers. [69] and [70] presented an efficient method for the watermark estimation and recovering from global or local geometrical distortions. The estimation of the affine transform parameters is formulated as a robust penalized Maximum Likelihood (ML) problem, which is suitable for the local level as well as for global distortions. The watermark is periodic with blocks. When no geometrical transform was applied, the message is decoded from the extracted watermark directly. If some geometrical transform was applied, based on the local ACF (autocorrelation function) or magnitude spectrums, or by exploiting the reference watermark information at the block level, the geometrical distortion can be determined, then the retrieved watermark can processed and re-synchronized and the message can



(a) Embedding. (b) Detection.

Figure 3.15: Voloshynovskiy's watermarking algorithm.

be decoded. In this way, the watermark acts as the roles of both the template and the copyright information bearer.

The proposed watermarking algorithm performs the watermark embedding in the following steps, as shown in Fig. 3.15 (a):

1. Encode the input message using error control coding. The resulting codeword is then mapped from $\{0, 1\}$ to $\{-1, 1\}$ using binary phase shift keying (BPSK), encrypted based on a key-dependent sequence, and followed by a spreading over a square block or segment of any shape with some density D based on the same secret key. The resulting block is upsampled by a factor of 2 and then flipped and copied once in each direction, which produces a symmetric macroblock. The

resulting macroblock is then replicated over the whole image size, which results in a symmetrical and periodical watermark.

2. Both the cover image and the watermark are first decomposed into a multi-resolution sub-band pyramid using a wavelet transform.

3. Add the decomposed cover image and the decomposed watermark together at each decomposition level based on the perceptual model (noise visibility function).

4. Apply the inverse wavelet transform to get the watermarked image.

The watermark detection and the estimation of the geometrical transforms are conducted in the following steps, as shown in Fig. 3.15 (b):

1. To estimate the watermark, a maximum a posteriori probability (MAP) estimate is used:

$$\widehat{w} = argmax_{w \in \Re^N} \{p_X(y|w) p_W(x)\} \tag{3.13}$$

where $p_X(\cdot)$ and $p_W(\cdot)$ are the probability density functions of the cover image and the watermark. y is the watermarked image. Assuming that the cover image and the watermark obey the conditionally i.i.d. Gaussian distribution, $x \in N(\overline{x}, R_x)$ and $w \in N(0, R_w)$. Here $\overline{x}$ is the mean value and R_x and R_w are the covariance matrix. The watermark $\widehat{w}$ can be determined as:

$$\widehat{w} = \frac{R_w}{R_w + R_x}(y' - \overline{y'}) \tag{3.14}$$

And $\widehat{R}_x = max(0, R_y - R_w)$ is the maximum likelihood estimate of the image covariance matrix if the original cover image is not available.

2. Once the $\widehat{w}$ is estimated, the autocorrelation of the $\widehat{w}$ is computed. Since the watermark is periodic, it results in a structure showing local maxima, or peaks, which is periodical too.

3. The rotation and scaling can be represented by a linear matrix A

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{3.15}$$

   The affine transform maps each point of Cartesian coordinates (x, y) to (x', y') by the linear matrix A.

   It is proposed to use an approach based on penalized ML estimation as a robust approach for the estimation of the affine transform A, which is suitable for the local level as well as for global distortions.

4. In the case when no geometrical transform was applied the message is decoded from the detected watermark directly. If some geometrical transform was applied, the geometrical distortion can be determined based on the estimation in the previous step. Then the retrieved watermark can be processed and resynchronized and the message is decoded.

3.4 Salient feature based algorithms

The template based watermarking algorithms are trying to add a recognizable template into the host image. And this template bears with some information about the geometrical structure of the host image. Given the assumption that the template can

always be retrieved, based on the distortion applied to the template, we can detect the geometrical transforms the image has undergone.

Based on the template-based approaches, it is quite straightforward to come up with the idea that if we can extract some kind of pattern that the cover image possesses so that we can use the pattern as the reference template. Because this pattern has to be recognizable, normally we use salient features of the cover image as the desired pattern. Therefore, we can identify the pattern even when the cover image is severely distorted. The location of the watermark is not linked with image coordinates, but with image features or semantics [71][72]. The problem of geometrical synchronization can be solved because the image features represent an invariant reference to geometrical transformations. The feature could be some feature points extracted through the corner or edge detection algorithms. The following two watermarking algorithms use the feature points as the reference coordinates of the watermark embedding location.

[71] proposed a geometrically invariant watermarking algorithm by using feature points and Delaunay tessellation.

The watermark embedding process shown in Fig. 3.16 (a), consists of the following steps:

1. Generate a random sequence with the shape of a right-angled isosceles triangle as a watermark;

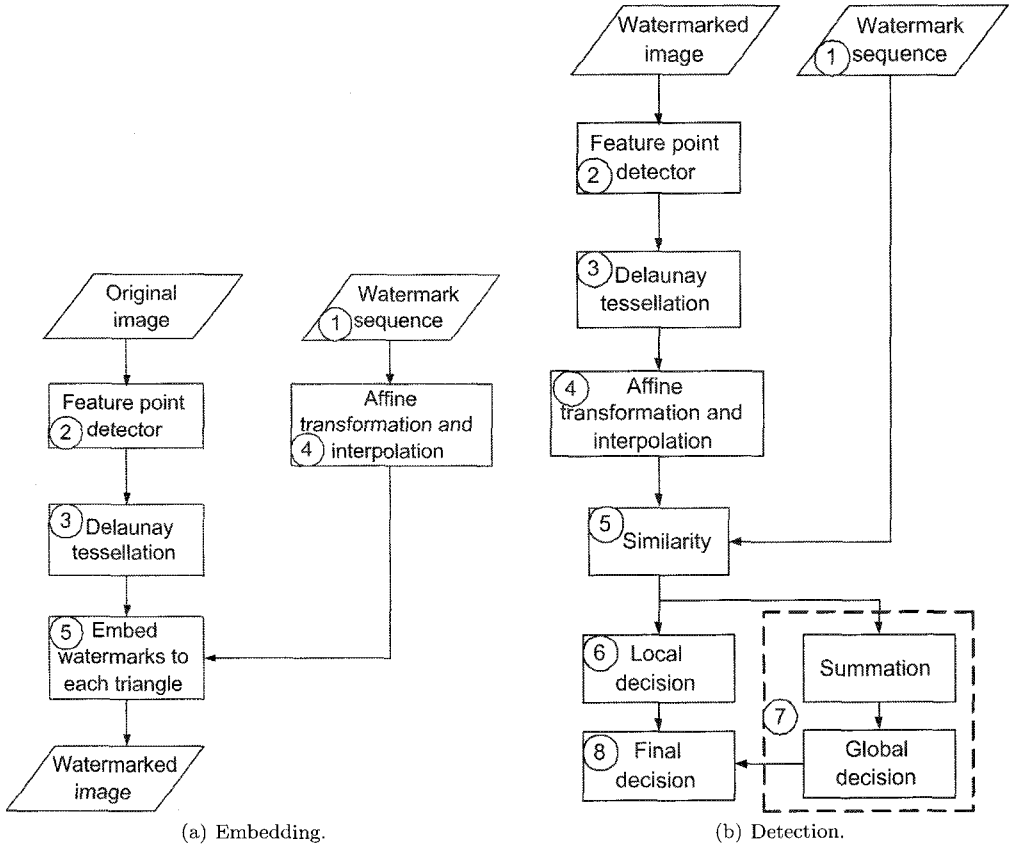
2. Detect robust feature points in the image by using the Harris corner detector, mentioned in Section 2.11;

3. 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;

4. Each watermark sequence is transformed to the same shape of each triangle using affine transform and interpolation;

5. The transformed watermark sequence is embedded to each triangle to obtain the watermarked image.

The watermark detection process shown in Fig. 3.16 (b), consists of the following steps:



(a) Embedding. (b) Detection.

Figure 3.16: Bas' salient feature based watermarking algorithm.

1. Generate a random sequence with the shape of a right-angled isosceles triangle as a watermark as in the embedding process;

2. Detect the robust feature points in the image by using the Harris corner detector;
3. Create a triangular tessellation of the image based on a set of feature points. The image is divided into a set of disjoint triangles by Delaunay tessellation;
4. Each triangle is warped into a right-angled isosceles triangle with the same size and shape with the watermark sequence;
5. The similarity for each triangle is calculated.
6. A local decision is made for each triangle. The mark is detected for a single triangle if the similarity is larger than a threshold.
7. The global decision is made by comparing the global similarity, which is the global sum of the local similarities that are above a threshold, to a threshold.
8. The final decision is made according to the local and global decision.

The important step of this algorithm is to choose Delaunay tessellation. The tessellation has two properties: 1) if a vertex disappears, only the connected triangles are modified; 2) if the vertex is moving inside the triangle area, the tessellation is not modified.

A feature extraction method called Mexican Hat wavelet scale interaction is used in the algorithm of [73]. The extracted feature points can survive a variety of attacks and be used as reference points for both watermark embedding and detection. The normalized image of an image (object) is nearly invariant with respect to rotations. As a result, the watermark detection task can be much simplified when it is applied to the normalized image.

The proposed watermarking algorithm performs the watermark embedding in the following steps, as shown in Fig. 3.17:

1. Extract the feature points using the Mexican Hat wavelet scale interaction.
2. Select the circular regions centered at these points as the watermark embedding regions.
3. Apply the image normalization to these circular regions. Select some points from the normalized circular regions, then transform the coordinates of these selected points back to the original image.
4. Based on the selected points, construct blocks with a size of 32×32 in those circular regions of the original image.
5. Compute 2-D DFT of these blocks.
6. Embed the watermark into the Fourier magnitude of these blocks using differential encoding.
7. Compute 2-D IDFT of these blocks.
8. Get the watermarked image by replacing those original circular regions with the watermarked ones.

Because the normalized version of an image (object) is nearly invariant with respect to rotations, the location regions for watermark embedding can always be retrieved regardless of rotation. The following are the steps for watermark detection, as shown in Fig. 3.18:

1. Extract the feature points using the Mexican Hat wavelet scale interaction.
2. Apply the image normalization to those circular regions.

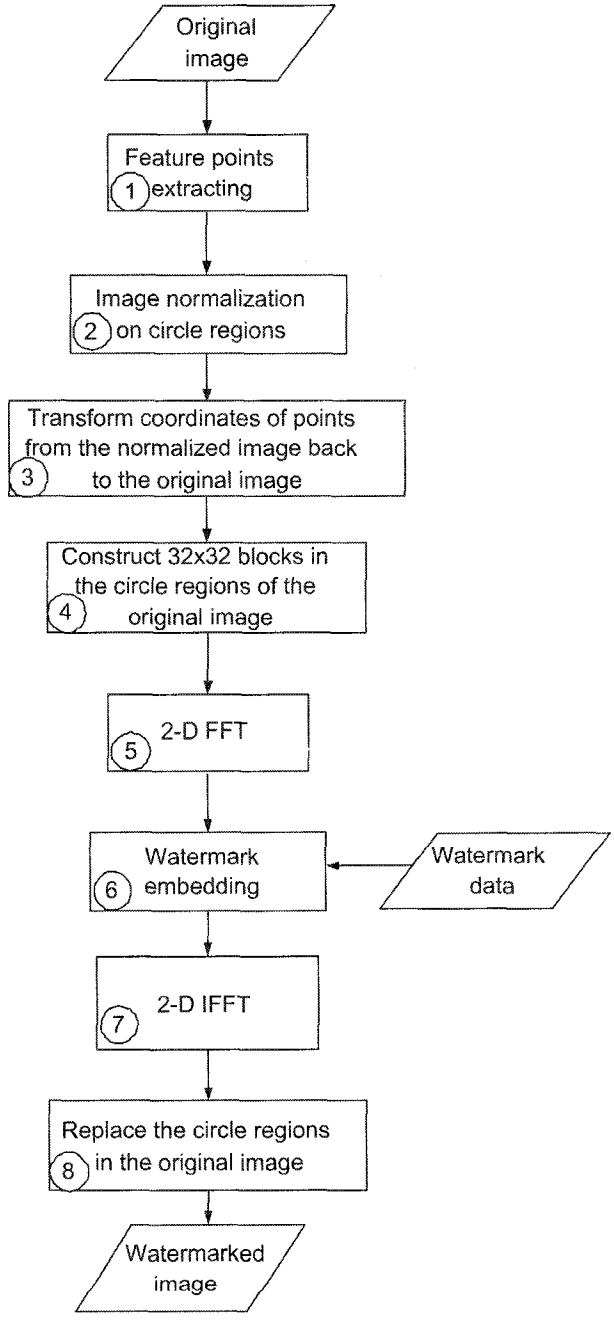


Figure 3.17: Watermark embedding procedures of Tang's method.

3. Since the normalized image regions are rotation invariant, the coordinates of these points can be determined and transformed back from the normalized image to the watermarked image.

4. Construct the blocks in the watermarked image, which are exactly those blocks used for watermark embedding.

5. Compute 2-D DFT of the blocks.

6. Use the differential decoding to extract the watermark in the Fourier magnitude of the those blocks constructed in Step (3) and calculate the similarity.

7. Compare to the predefined threshold to judge the presence of watermark.

Several other watermarking algorithms are feature-based. They are so-called the second-generation watermarking algorithms because the feature of an image is exploited for embedding watermark [74]. The authors of [75] have proposed an algorithm for recognizing geometrically distorted images and restoring their original appearances by using image feature points. [76] have developed a method based on image feature to identify the geometrical transformation. [77] introduced an RST synchronization algorithm based on the wavelet decomposition of an image. [78] have designed a content-based watermarking method that does not require the original image and uses self spanning pattern.

The limitations of the feature point detectors are that when the image is subjected to a geometric transform, the results of the feature point detection may be altered resulting in a false localization of the feature points and failure of watermark detection. Scaling and local distortion specially affect local operators significantly. So the

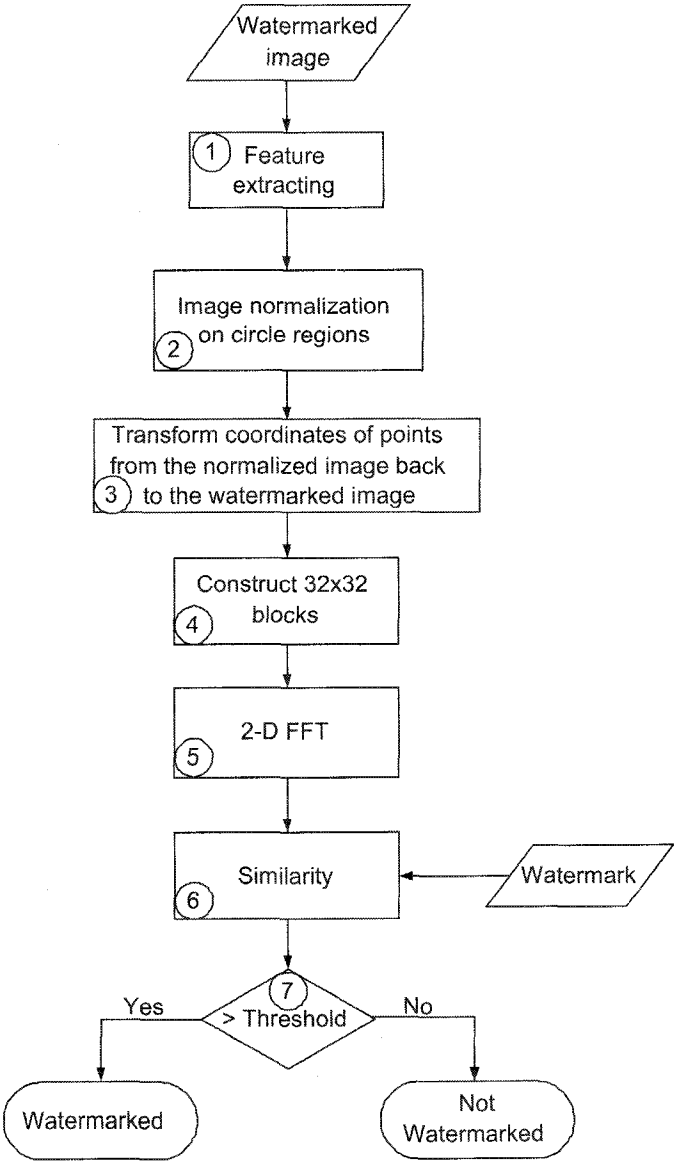


Figure 3.18: Watermark detection procedures of Tang's method.

segmentation-based feature point extraction is a good method that it uses a segmentation of the image for the determination of the feature points. For instance, the centroid of each region identified through segmentation may be selected as a feature point. The segmented regions are relatively invariant to various image manipulations. Also the centroid positions are more accurate reference of the positions of the watermark than the feature points extracted using those local feature point detectors once the image is subjected to some geometrical transforms. As an example, Gibbs Random Field (GRF) based image segmentation algorithm can be used to provide a segmentation of the image into spatially contiguous regions. The centroid of the regions can be selected as feature points.

3.5 Image decomposition based algorithms

Another approach to an RST invariant watermarking algorithm is to decompose the image or watermark into components using a set of orthogonal or non-orthogonal base functions. These decomposed components have some RST invariance properties. As discussed in Section 2.6, the correlation between the decomposed components of the image (kth circular harmonic function) and the image will not be affected by rotation. The Pseudo-Zernike basis is a set of complete and orthogonal functions. The expansions of the image based on the Pseudo-Zernike [79] basis have the properties of RST invariance. the watermarking algorithms using these ideas are discussed in Section 3.5.1 and Section 3.5.2.

3.5.1 Match filtering based algorithm

[46] proposed a rotation-tolerant watermarking method by using the circular harmonic function correlation filter. They use this new filter instead of the conventional matched filter to achieve the rotation tolerant correlation peak in the watermark detection process. They design the filter by using the properties of the circular harmonic function (CHF), discussed in Section 2.6. The CHF filter h can be expressed by the filter design function f_0 as follows [46]:

$$h = f_0(W, c(\phi)) \tag{3.16}$$

with

$$c(\phi) = \begin{cases} 1, for\ |\phi| \leq \phi_t \\ 0, for\ |\phi| > \phi_t \end{cases} \tag{3.17}$$

where W is the watermark pattern and ϕ_t is the tolerance angle. $c(\phi)$ is the correlation function shown in Eq. (2.31). This filter also can be optimized by using Optimal Trade-off Circular Harmonic Function (OTCHF) [80], which get the trade-off performance using 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 [81][60] and embedding the watermark into the original image by simple addition (refer to Fig. 3.19). The watermark embedding steps are as follows:

1. Compute the perceptual mask of the original image;

2. Modulate the randomly generated watermark data with the perceptual masking;

3. Embed the modulated watermark into the original image to get the watermarked image;

4. Generate the OTCHF filter from the watermark data.

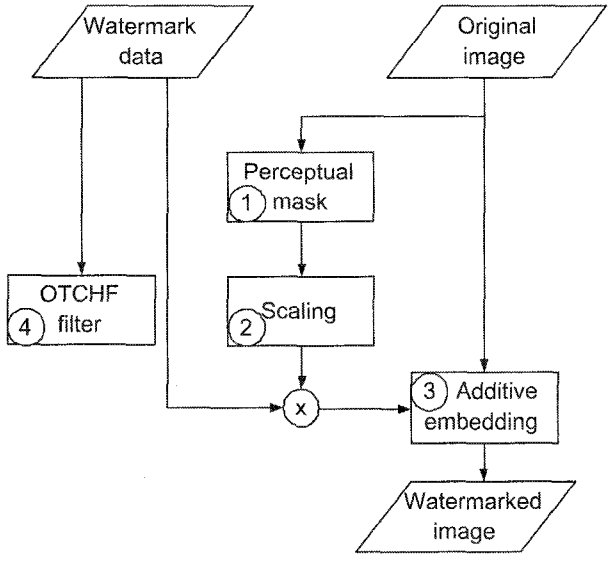


Figure 3.19: Watermark embedding procedure of circular harmonic based method.

At detector, they measure the correlation by computing the inner product between the test image and the OTCHF filter h, (refer to Fig. 3.20). The watermark detection steps include:

1. Decorrelate the host image from the watermark by using whitening filter;

2. Calculate the correlation between the decorrelated image and the OFCHF filter;

3. Compare the correlation coefficient with the predefined threshold to judge if the test image is watermarked.

The OTCHF filter can be designed off-line. Hence, it is suitable for any real-time watermark detector [46]. This method is only rotation tolerant watermark detection.

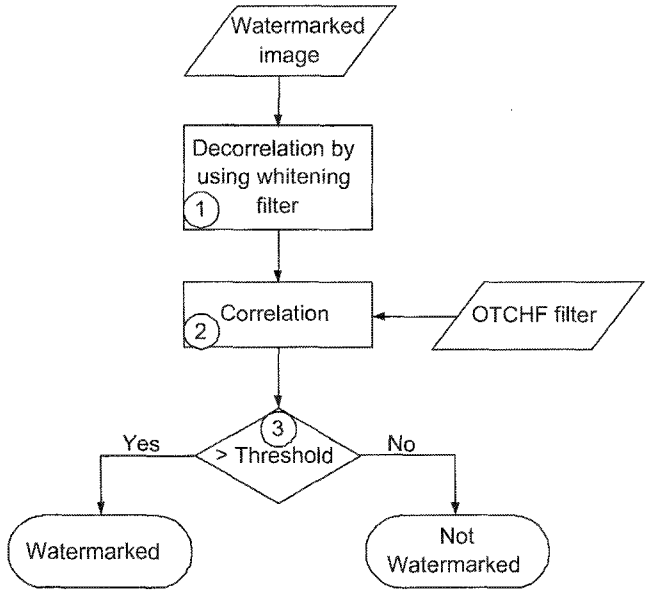


Figure 3.20: Watermark detection procedure of circular harmonic based method.

Experimental results show that this method is tolerant to rotation angles less than the 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 might produces higher false positive and false negative probabilities. The overlap of the histogram of peak-to-sidelobe ratio (PSR) [46] between the unwatermarked images and the watermark images before or after rotation shows this problem.

3.5.2 Pseudo-Zernike polynomial decomposition based algorithm

In the paper by [82], a geometrically robust image watermarking algorithm using Pseudo-Zernike moments is proposed. Some selected Pseudo-Zernike moments of an image are computed, and their magnitudes are quantized by dither modulation to embed an array of bits. In watermark detection, the embedded bits are estimated from

the invariant magnitudes of the Pseudo-Zernike moments using a minimum distance decoder. The Pseudo-Zernike basis is a set of complete and orthogonal functions defined in polar domain. The expansions of the image based on the Pseudo-Zernike basis have the properties of RST invariance, which are so-called Pseudo-Zernike moments. Thus the watermark can be embedded by modifying the magnitude of the Pseudo-Zernike moments.

The Pseudo-Zernike basis is a set of complete and orthogonal functions on the unit disk defined as follows:

$$V_{nm}(x,y) = R_{nm}(\rho)e^{jm\theta} \tag{3.18}$$

where $\rho = \sqrt{x^2+y^2}$, $\theta = tan^{-1}(x,y)$ and

$$R_{mn}(\rho) = \sum \frac{(-1)^2(2n+1-s)!\rho^{n-s}}{s!(n+|m|+1-s)!(n-|m-s!|)} \tag{3.19}$$

Then we can decompose the image into the Pseudo-Zernike moments:

$$A_{nm} = \frac{n+1}{\pi}\int\int f(x,y)V_{nm}(x,y)dxdy \tag{3.20}$$

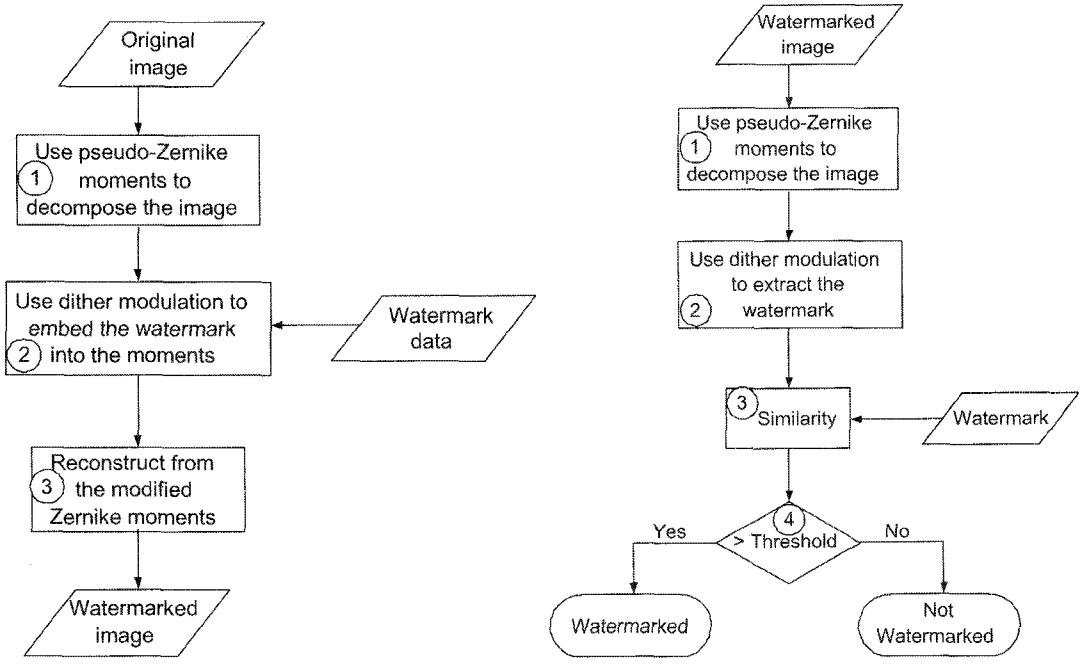
where $x^2+y^2 \leq 1$.

These Pseudo-Zernike moments are rotation and flipping invariant. So they can be used to implement RST invariant image watermarking.

The embedding steps of the Zernike moments based watermarking algorithm are as follows, as shown in Fig. 3.21 (a):

1. Decompose the image into the Pseudo-Zernike moments using the method mentioned above.

2. Embed the watermark bits using dither modulation to quantize the magnitude of



(a) Embedding. (b) Detection.

Figure 3.21: Zernike moments based watermarking algorithm.

each Pseudo-Zernike moment.

3. Reconstruct the image using the modified Pseudo-Zernike moments.

The detection steps of the Zernike moments based watermarking algorithm are quite straightforward, as shown in Fig. 3.21 (b);

1. Decompose the watermarked image into the Pseudo-Zernike moments using the method mentioned above.

2. Detect the watermark bits using dither modulation.

3. Compute the similarity between the extracted watermark data and the generated watermark.

4. Compare with the predefined threshold to judge the present of watermark.

3.6 Stochastic analysis based algorithms

The stochastic characteristics of the image are very important for image analysis. Local mean and local variance represents the image spatial distribution. The moments of an image can be used to implement the RST invariant watermarking algorithm.

3.6.1 Higher order spectra based algorithm

Higher order spectra are defined in terms of the higher-order moments or cumulants of a signal, and are used for identification of nonlinear, non-Gaussian random processes and deterministic signals [83]. It can be expressed as the products of Fourier coefficients at component frequencies and the conjugate of the Fourier coefficient at the sum frequency.

Bispectrum, $B(f_1, f_2)$, is the Fourier spectrum of the triple correlation of a signal, as expressed in Eq. (3.21). It is useful for shift and rotation invariant object recognition [84][83][85][86].

$$B(f_1, f_2) = X(f_1)X(f_2)X^*(f_1 + f_2) \tag{3.21}$$

where $X(f)$ is the DFT of the sequence $x(n)$ at the normalized frequency f. $B(f_1, f_2)$ is a triple product of Fourier coefficients. It is defined in the triangular region of computation $0 \leq f_2 \leq f_1 \leq f_1 + f_2 \leq 1$ by its symmetry properties.

[84] proposed an image watermarking method that is resilient to rotation, scaling and translation (RST) by using the higher order spectra (HOS) in particular bispectrum. The translation and scaling invariance is achieved by using the phases of the integrated

bispectra while rotation invariant is obtained by using the Radon transform of the image.

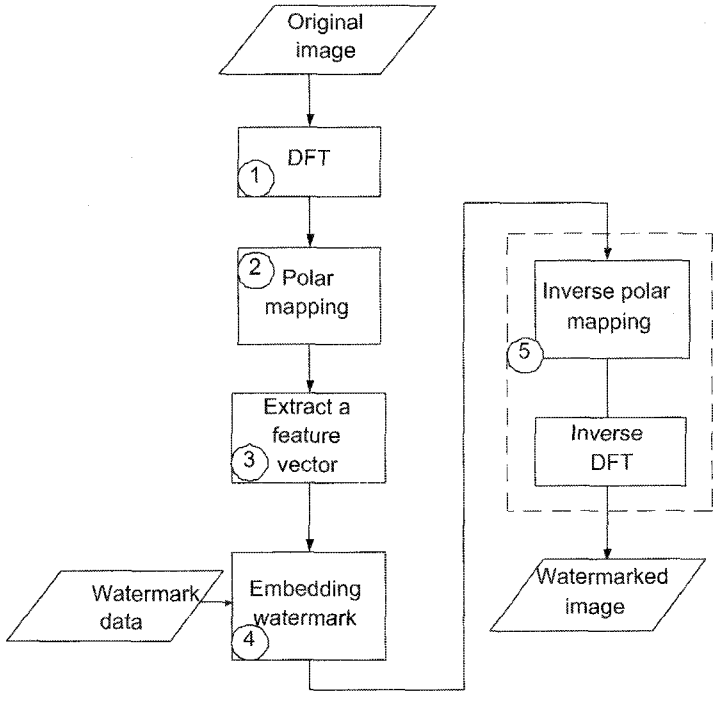


Figure 3.22: Watermark embedding procedure of high order spectra based method.

The watermark embedding consists of the following steps (refer to Fig. 3.22):

1. Compute the DFT of the original image;

2. Create the polar mapping of the Fourier magnitude;

3. Define a feature vector P as $P = (p(\theta_1), p(\theta_2), p(\theta_3), \ldots p(\theta_N))$. Each $p(\theta)$ is defined as the phase of the integrated bispectra of a 1-D Radon projection along the line of $f_1 = f_2$ as follows:

$$p(\theta) = \angle\left[\int_{f_1=0^+}^{0.5} B(f_1, f_2)df_1\right] = \angle\left[\int_{f_1=0^+}^{0.5} I_p^2(f_1, \theta)I_p^*(2f_1, \theta)df_1\right] \quad (3.22)$$

4. Embed the watermark by modifying k elements of the vector;

5. Compute the inverse polar-mapping and inverse DFT to get the watermarked image.

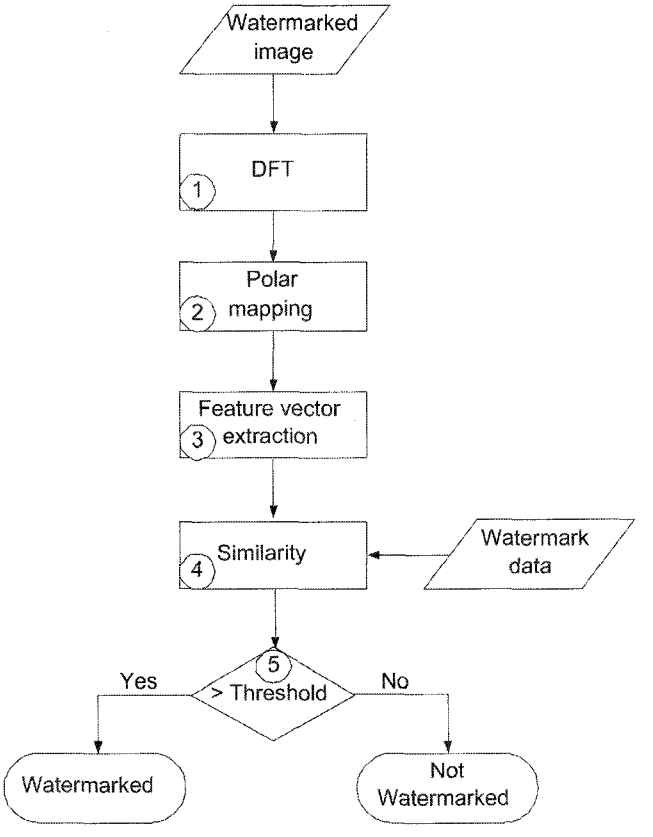


Figure 3.23: Watermark detection procedure of high order spectra based method.

However, interpolation and zero-padding will cause inversion errors. In order to avoid this problem, they use the extracted feature P^* from the watermarked image as the embedded watermark and adjust the embedding strength to make P^* lower than the maximum noise level r_1 (the maximum of the feature vector of an image and the image after RST distortion) and higher than the minimum noise level r_2 (the minimum

of the feature vectors of different images).

The detection process consists of the following steps (refer to Fig. 3.23):

1. Compute the DFT of the watermarked image;
2. Create the polar mapping of the Fourier magnitude;
3. Extract the feature vector by using same strategy as the embedding Step (3);
4. Calculate the correlation between the extracted feature vector and the watermark data;
5. Compare the correlation coefficient with the predefined threshold to judge if the test image is watermarked.

3.6.2 Image normalization based algorithm

The moments of objects have been widely used in pattern recognition, image registration [27], and image watermarking [87][28]. [28] proposed geometric invariance in image watermarking based on moments 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.

The idea is to geometrically transform the image into a standard form no matter how the image has undergone RST attacks. The translation invariant can be achieved by using the central moments of the image, which are origin independent (refer to Section 2.7). The scaling normalization transforms the image into its standard form by translating the origin of the image to its centroid $(\overline{x}, \overline{y})$. We change the coordinates

into $(\widehat{x}, \widehat{y})$ [28]:

$$\widehat{x} = \frac{x - \overline{x}}{a}, \ \widehat{y} = \frac{y - \overline{y}}{b} \tag{3.23}$$

with

$$a = \sqrt{\frac{\beta\gamma}{m_{0,0}}}, \ b = \sqrt{\frac{\beta}{\gamma m_{0,0}}} \tag{3.24}$$

In Eq. (3.23) and Eq. (3.24) 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 $\gamma = l_y/l_x$.

In order to realize the rotation normalization, two tensors t^1 and t^2 are defined as [28]:

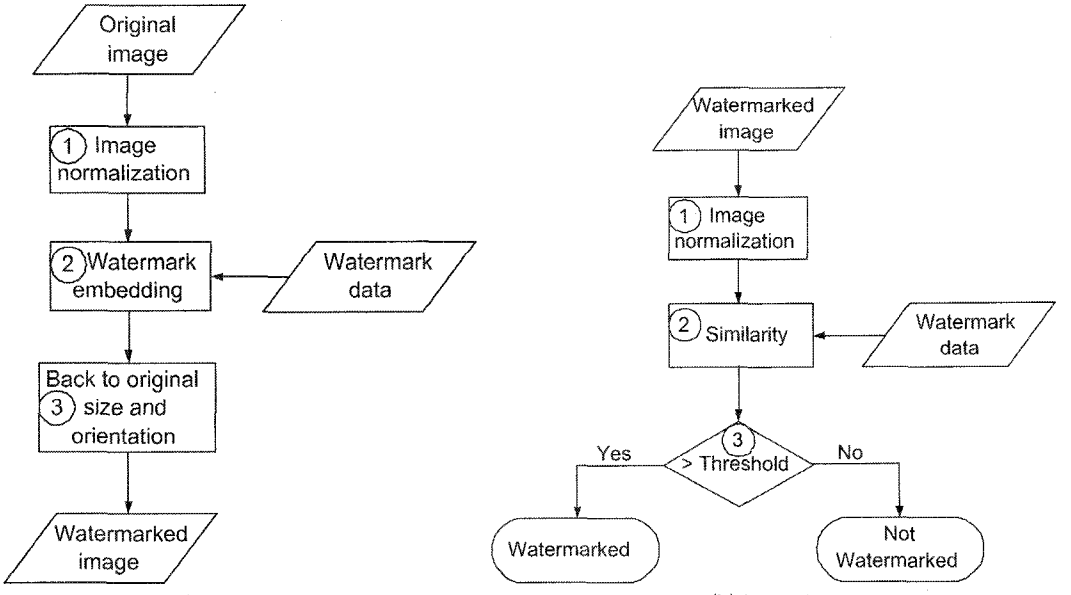
$$t^1 = \mu_{12} + \mu_{30}, \ t^2 = \mu_{03} + \mu_{21} \tag{3.25}$$

where μ is central moment, defined in Section 2.7. Then, the normalized angle ϕ is defined as

$$\phi = \arctan\left(-\frac{t^1}{t^2}\right) \tag{3.26}$$

After normalization, the distorted image and the original image have the same size, direction and orientation. The normalizing algorithm does not need the original image for implementing the normalization at decoder.

The watermark embedding process, as shown in Fig. 3.24 (a), includes the following



(a) Embedding. (b) Detection.

Figure 3.24: Moments based watermarking algorithm.

steps:

1. Normalize the original image according to the theory discussed above;
2. Embed the watermark into the normalized original image;
3. Transform the above image back to the original size and orientation to get the watermarked image.

The watermark detection process, as shown in Fig. 3.24 (b), includes the following steps:

1. Normalize the watermarked image;
2. Calculate the similarity between normalized image and the watermark data;
3. Compare with the predefined threshold to decide if the watermark is present.

[28] also proposed another approach to geometric transformation invariant watermarking. In that approach, the watermark is based on invariant parameters extracted from the geometric moments of the image. Also, other moments, like complex moments, are used for pattern recognition [88][89]. The similar watermarking algorithms have been proposed [32][90][91][92].

3.7 Others

There are some other RST invariant watermarking algorithms. In [93], it is proposed to use the 3-D eigenvectors of the image object to determine the orientation of the 2-D eigenvectors of the object, and then one can synchronize the distorted object based on the alignment of the 2-D eigenvectors. The image object segmentation is proceeded before watermark embedding and detection.

[94] proposed the mesh model based algorithm. A deformable mesh model is used to detect the geometrical distortions. Then the distorted image can be resynchronized for watermark detection.

In this section, we introduced various representative RST watermarking algorithms in detail. It covers most of known RST watermarking algorithms. The algorithms and embedding/detection implementation are discussed in detail. In the following section, we will give analysis and discussions about the performance and advantages/disadvantages of these algorithms.

Chapter 4

Analysis of RST invariant image watermarking algorithms

The most important and widely-used criteria to evaluate the performance of a robust digital image watermarking algorithm are robustness, capacity, and invisibility as mentioned in Section 1.5. So we will analyze the performance of the algorithms mentioned below based on these three criteria.

There are different ways to design watermarking algorithms to achieve the RST invariance property. We discussed the existing RST invariant watermarking algorithms based on the techniques they used. First, we would like to discuss the watermarking algorithms that utilize the RST invariant domains of the image. Naturally, if the watermark is embedded into these RST invariant domains, the processes of the watermark detection/extraction can be completed regardless of the geometrical transforms. One way to reach such a domain is to apply the Fourier-Mellin transform to the image. To detect the geometrical transforms, one possible way is to embed into the image a template that can be easily detected even the watermarked image is geometrically transformed.

Thus based on the characteristics of the template, the geometrically transformed image or watermark can be transformed back to its original position, size and shape. Also there are some salient features of the image such as corner points, regions, and edges that are the vital parts of the image. These features can be detected even when the image is geometrically transformed or degraded in quality, thus various algorithms utilizing these features have been proposed. After projecting the two dimensional image into one dimension, some RST invariant properties appear and can be exploited to implement RST invariant watermarking algorithms. One such example is the Radon transform. Match filtering methods or Pseudo-Zernike polynomial decomposition can decompose the image/watermark into components, some of which bear the properties of RST invariance. So RST invariant watermarking algorithms can be designed based on these theories. From the stochastic analysis of the image, the idea of moments has been introduced to design the watermarking algorithm. Moments and bispectra based watermarking algorithms are also discussed in this section.

The simulation results presented in this survey is first to illustrate the working principles of different watermarking algorithms, that is, how the RST invariant features are extracted and exploited to design an RST invariant watermarking algorithm. Based on this, the performance of each algorithm is presented and some comments combined with the theory-based discussion are given. To make a relatively meaningful discussion and comparison, the spread spectrum is used to generate the watermark for embedding and the correlation detection (normalized correlation or linear correlation) is used to detect the existence of the watermark. The embedding strength is adjusted so that the PSNR of the watermarked image is around 40dB. Also since the false positive probability and true negative probability are very important to evaluate the performance of the watermarking algorithm, the detection results are computed when various at-

tacks are applied to the watermarked image and unwatermarked image. Because this survey is mainly focused on the RST invariant watermarking algorithms, the performances of the watermarking algorithms are evaluated for geometrical attacks such as rotation and scaling. Also the performance of the watermarking algorithms for the JPEG compression is given since it is widely used in image processing. Both linear correlation and normalized correlation are used to detect the watermark. The values for linear correlation and normalized correlation are different. Please refer to Section 2.8 for the detailed explanation of linear correlation and normalized correlation. In the following, the experimental results are presented in figures, where the X axis represents the different test scenarios such as rotation by different degrees or scaling by different ratios, and the Y axis represents the correlation detection results. To give a comparison between different algorithms, several grades are used to comment the performance of the watermarking algorithms including "very good", "good", "average", and "poor". The judgement is made based on the experimental results and some subjective evaluation. The advantages and disadvantages of these watermarking algorithms will also be analyzed in this section.

4.1 RST invariant domain based algorithms

4.1.1 Fourier-Mellin transform based algorithms

In the paper by [4], the Fourier-Mellin transform was used to implement an RST invariant image watermarking algorithm for the first time. Also the author is one of the first who used the idea of the spread spectrum for watermark generation, embedding and detection. The Fourier transform has the property of the shift invariance. Because of the log-polar mapping, the rotation and scaling in the spatial domain result in the shift

in the LPM domain. So the Fourier transform followed by the Fourier-Mellin transform can transform the image to the RST invariant domain. The watermark embedded into this RST invariant region can resist the global affine transform. However, the LPM is not invertible in terms of implementation, the image having undergone the LPM and ILPM will be severely distorted. Thus the author proposed a modified watermarking algorithm in which only the watermark will go through the ILPM and the fidelity of the watermarked image can be preserved. The spread spectrum techniques make the watermarking algorithm robust against noise and compression, and the Fourier-Mellin transform grant the algorithm the RST invariance property. Though this watermarking algorithm is novel, there are some problems coming with it. First, the image is transformed to the RST invariant domain used to embed the watermark by applying the DFT and Fourier-Mellin transform. In the RST invariant domain, the spatial and frequency distribution information of the image is lost so that it is difficult to utilize human perceptual model such as texture/luminance masking function to adjust the embedding strength and select the suitable embedding locations. Also the LPM is actually a sampling process which oversamples at the center area and downsamples when the distance from the center becomes larger. This may cause serious problem for the watermark embedding according to our experiments. Second, the watermark will go through the ILPM that may cause a lot of distortions to the watermark. This could affect the performance of the detector. Also the spread spectrum-based watermarking can increase robustness dramatically, while the correlation detection makes the capacity very low. In spite of these problems, this watermarking algorithm is original and many modified watermarking algorithms based on it have been proposed later, which solved one or more problems mentioned above.

4.1.2 Phase correlation and log-polar mapping based algorithm

In this algorithm [55], the watermark is embedded into the LPM domain of the original image. The watermark is generated using spread spectrum and embedded into the middle frequencies to achieve the tradeoff between robustness and fidelity. The evaluations demonstrate that the algorithm is invariant to rotation and translation, invariant to scaling when the scale is in a reasonable range, and very robust to JPEG compression and other attacks. To solve the difficulties in the embedding processes caused by the LPM and ILPM, the embedding locations are first determined in the LPM domain of the image, then the corresponding locations in the DFT magnitude domain are computed using the ILPM. In this way, the watermark can be embedded directly into the DFT magnitude domain of the image and will not be distorted by the ILPM.

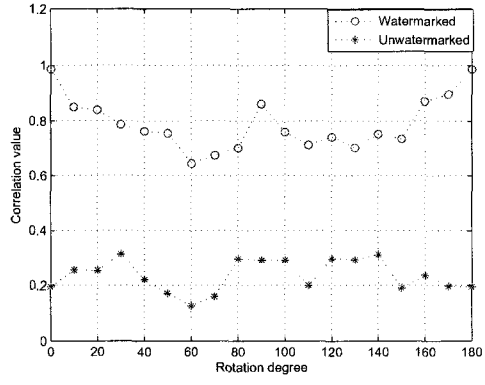
According to our experiments, the performance of this algorithm is as follows:

1. Rotation and scaling: *very good*

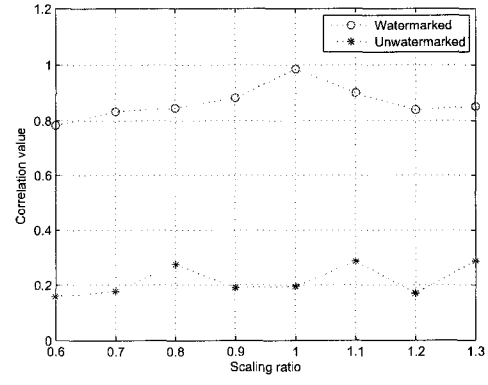
   The rotation and scaling in the spatial domain result in the shift in the LPM domain, the phase correlation between the LPM of the original image and the LPM of the watermarked image is used to calculate the shift of the watermark positions in LPM domain. The performance of this algorithm against the rotation and scaling are very good as shown in Fig. 4.1 (a) and Fig. 4.1 (b). The normalized correlation value differences between the unwatermarked images and the watermarked images are distinguished enough to make a clear justification of the existence of the watermark based on the predefined threshold, which is determined using stochastic analysis as discussed in [55].

2. JPEG compression and additive noise pollution: *good*

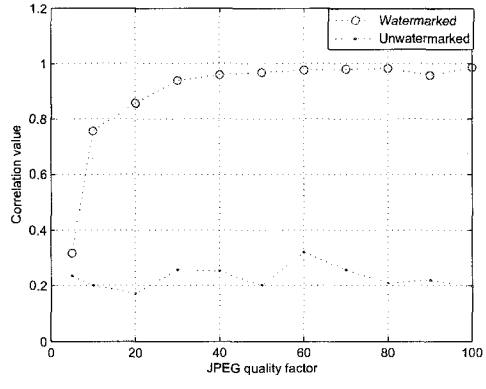
The watermark is embedded into the middle frequencies using spread spectrum to increase the robustness, which is a popular method to increase the robustness of the watermark against the JPEG compression and noise pollution. Experiments show that it is robust against JPEG compression and additive noise pollution. In Fig. 4.1 (c), it is shown that even with a quality factor down to 20%, the watermark can still survive. This is good enough for practical applications.



(a) Rotation.



(b) Scaling.



(c) JPEG compression.

Figure 4.1: The experimental results of the phase correlation and log-polar mapping based algorithm.

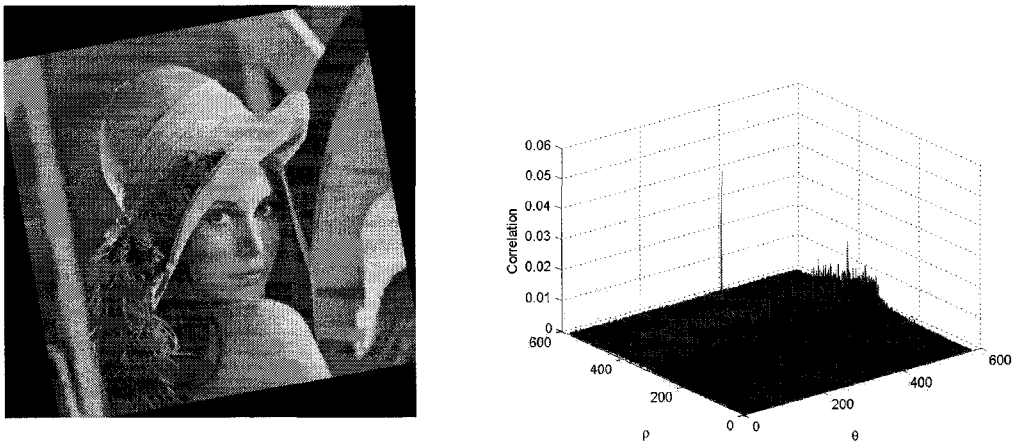
Although this watermarking algorithm shows strong robustness against various at-

tacks, the original image is needed for the watermark detection, which may poses a problem for some applications. Also both the watermark embedding/detection and the shift rectification are done in the LPM domain, which make the whole algorithm less flexible. These two procedures should be relatively independent for the purpose of optimization. Another problem is that, because of the characteristic of the LPM, the available region in the LPM domain for watermark embedding is limited. This may increase the chance of being attacked. The exhaustive search method is used to retrieve the watermark if the original image is unavailable, however it could increase the false positive probability and the computation cost.

4.1.3 Phase-only filtering and log-polar mapping based algorithm

This algorithm is proposed by [56]. Since the rotation and scaling in the spatial domain result in the shift in the LPM domain, the symmetrical phase-only matching filter (SPOMF) is used to compute the possible shift in the LPM domain. Then the scaling ratio and rotation degree can be computed to rectify the geometrically transformed image to its original shape, size and position. The SPOMF uses only the phase information of the matching template and the LPM of the watermarked image having undergone RST transformations to compute the shift parameters in the LPM domain. The matching template is an 8 bpp (bits per pixel) 64×64 data block cut from the LPM domain of the original image. For the phase only filter matching method, the energy is concentrated in a much narrower peak so that the filter has a better discrimination capability. The Fig. 4.2 (a) is the watermarked image *Lena* after scaling and rotation and the Fig. 4.2 (b) is the result of the phase-only filtering. By locating the peak in

the correlation plain, we can compute the possible shift in the LPM domain by which the scaling ratio and rotation degree can be retrieved.



(a) The rotated and scaled image.

(b) The result of Phase-only filtering.

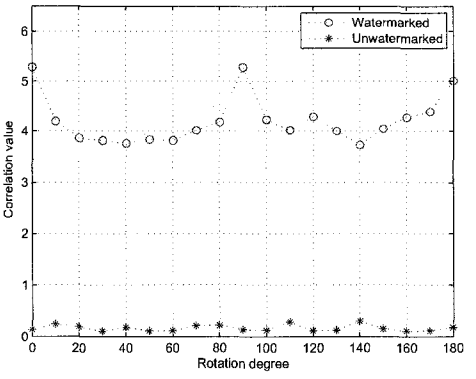
Figure 4.2: The rotated, scaled image and its phase-only filtering result.

According to our experiments, the performance of this algorithm is as follows:

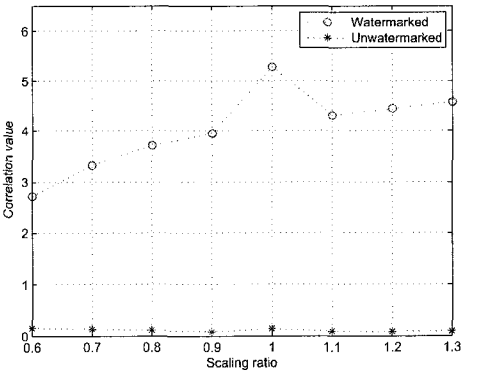
1. Rotation and scaling: *very good*

   The experimental results show that this method is feasible and it can detect the RST parameters. Based on this, the watermark embedding/detection algorithms are proposed such that the watermark is embedded into the spatial domain, the DFT magnitude domain and the LPM domain respectively. Once the RST parameters can be detected, the geometrically transformed image is re-synchronized to its original shape, size and position. All these three watermark detection algorithms can detect the existence of the watermark correctly. Fig. 4.3 (a) and Fig. 4.3 (b) show the performance of the SPOMF based watermarking algorithm. The watermark is embedded and detected in the spatial domain and the spread

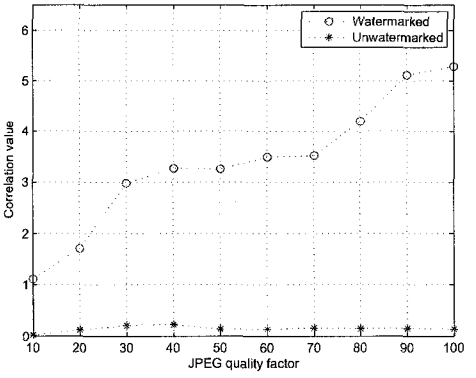
spectrum is used for enhancing security.



(a) Rotation.



(b) Scaling.



(c) JPEG compression.

Figure 4.3: The experimental results of the phase-only filtering and log-polar mapping based algorithm.

2. JPEG compression and additive noise pollution: *good*

The watermarking embedding/detection algorithm is based on the spread spectrum and the embedding strength is adjusted adaptively based on the local variance of the image. The SPOMF provides good detection accuracy for the RST transforms even under JPEG compression or noise pollution. So the whole algorithm shows good performance against the JPEG compression and noise pollution.

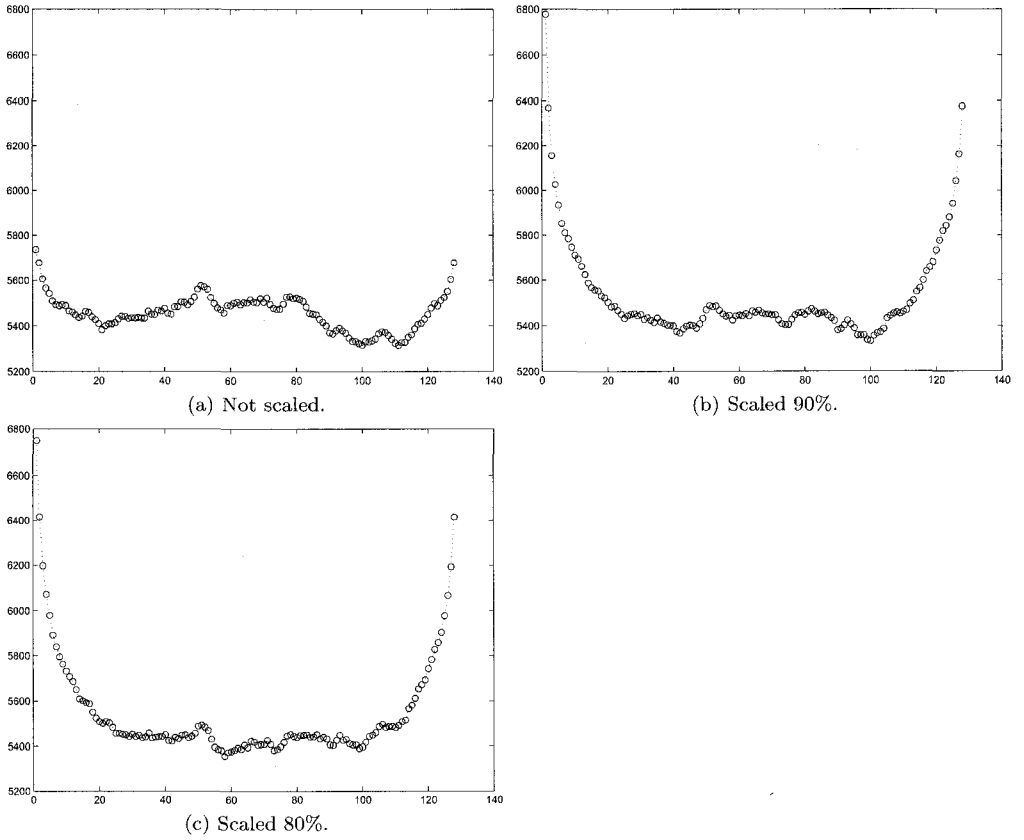
Fig. 4.3 (c) shows the result of JPEG compression.

This matching template is a small part of the LPM domain of the original image so that it does not have the same problem as the traditional template based watermarking algorithm. It cannot be detected by locating the local maxima. The only disadvantage of this method is that the template is derived from the image, which makes it not a completely blind detection. These watermark embedding/detection algorithms are independent of the image re-synchronization algorithm, so some algorithms incorporated with more advanced human perceptual models could be used to get a better performance for robustness and fidelity.

4.1.4 One-dimensional projection and log-polar mapping based algorithm

This method proposed by [38] is based on the Fourier-Mellin transform, while utilizes some invariance properties of the one-dimensional projection. The context-related watermark is embedded into a domain of 1-D projection of the LPM spectrum of an image. This domain is scaling and translation invariance according to its properties and rotation results in a circular shift in this domain. Fig. 4.4 shows the 1-D projection of the LPM spectrum of an image. The image is under scaling with 100%, 90% and 80%. The figures are almost kept the same, as mentioned in its properties. Fig. 4.5 shows the 1-D projection of the same image's LPM spectrum under rotation from 5° to 90°. The circular shift peak shows the other property of this domain that rotation results the circular shift in the LPM domain. In addition, the approximate inverse LPM is employed in order to avoid the implementation difficulties of the inverse LPM [4][38]. Because of the LPM, circles of the original image are converted to the rows of the transformed

image. Then 1-D projection along the columns of the transformed image is performed. This process is similar to the Radon transform to a certain degree.



(a) Not scaled.

(b) Scaled 90%.

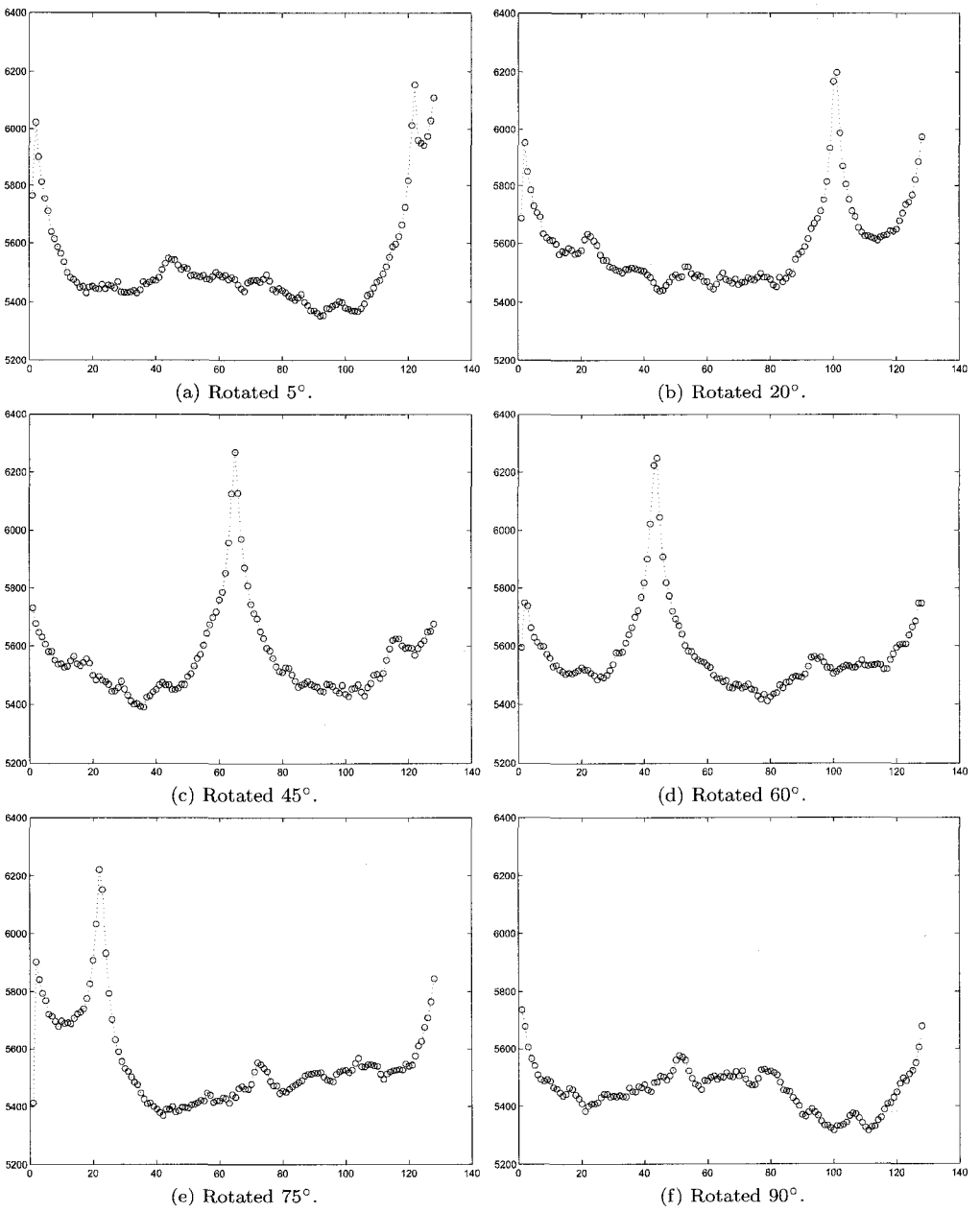
(c) Scaled 80%.

Figure 4.4: 1-D projection transform of *Lena* subjected to scaling with different factors.

According to our experiments, the performance of this algorithm is as follows:

1. Rotation and scaling: *good*

   This method has good performance again the rotation, scaling and translation. They calculate the correlation coefficient between the extracted data from the watermarked image and the watermark vector. According to our experimental results, shown in Fig. 4.6 (a) and Fig. 4.6 (b), the top line contains the correla-



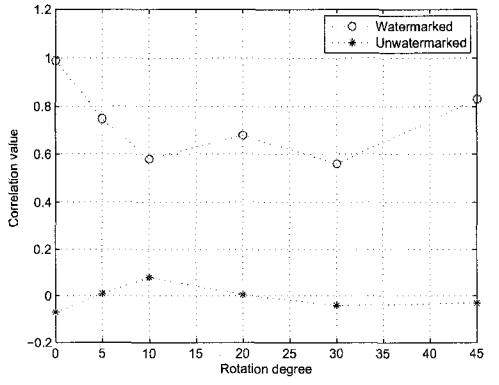
(a) Rotated 5°. (b) Rotated 20°.

(c) Rotated 45°. (d) Rotated 60°.

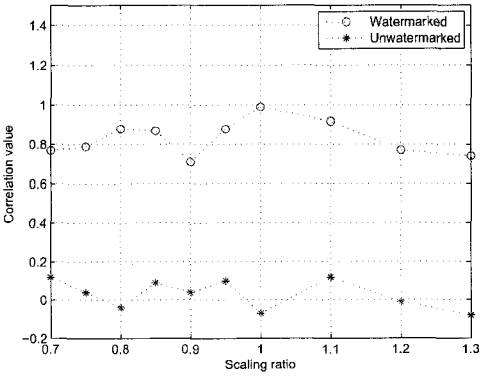
(e) Rotated 75°. (f) Rotated 90°.

Figure 4.5: 1-D projection of *Lena* having undergone different rotation angles.

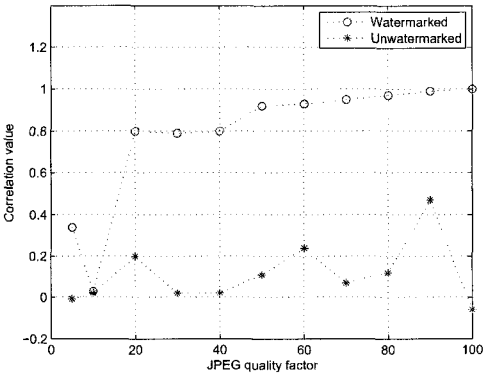
tions for the watermarked image having undergone different rotating angles and scale ratios and the bottom one shows the correlations for the unwatermarked image having undergone different scale ratios and rotating angles. They clearly distinguish the watermarked image and unwatermarked one.



(a) Rotation.



(b) Scaling.



(c) JPEG compression.

Figure 4.6: The experimental results of the one-dimensional projection and log-polar mapping based algorithm.

2. JPEG compression and additive noise pollution: *good*

We compressed the watermarked image *Lena* by different quality factors, the test results are shown in Fig. 4.6 (c). It has a good performance again JPEG com-

pression. We also added Gaussian noise with zero mean and different variances to the image *Lena*, the test results are showed in Table 4.1.

Table 4.1: Gaussian noise

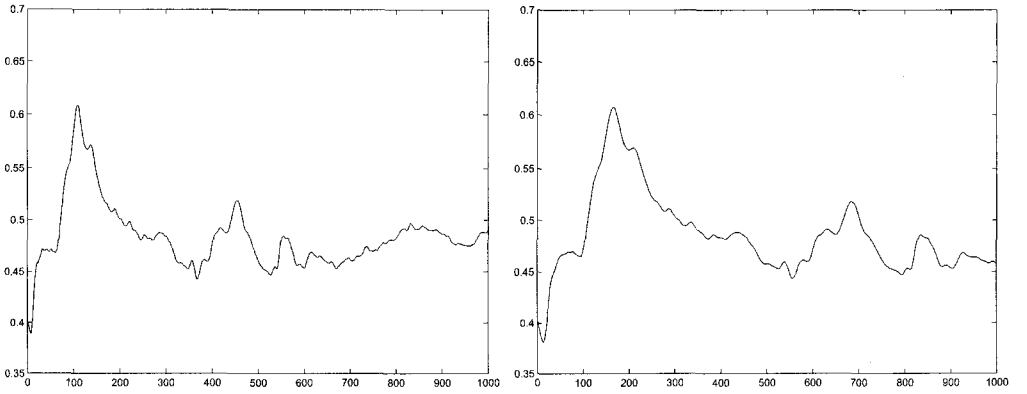
Attack	Watermarked image	Unwatermarked image
N(0,0.001)	0.91	-0.01
N(0,0.005)	0.69	-0.09
N(0,0.05)	0.42	-0.02

Embedding method of this algorithm is not a simple addition. A nonlinear embedding method is employed here. Watermarks are embedded by multiplication instead of addition. Because of the property of informed embedding method in this algorithm, only one embedding strength can give the best quality to the watermarked image. Higher or lower embedding strength will worsen the quality of watermarked image. According to our experimental results, when the embedding strength equals to 55, we got the highest PSNR. It is 40.3781 dB. In the paper by [38], exhaustive search is used for the rotation of image. The false positive probability is higher. This can be improved by rectifying the rotation angles, and then computing the correlation [58].

4.2 Radon transform based algorithms

The method proposed by [42] is based on the Radon transform. The Radon transform is used to retrieve the geometrical transform parameters such as rotation degree and scaling ratio. During detection, the watermarked image is first re-synchronized to its original shape, size and position, then the watermark can be detected. The watermark is embedded in the spatial domain of the image and the correlation is used to detect the existence of the watermark.

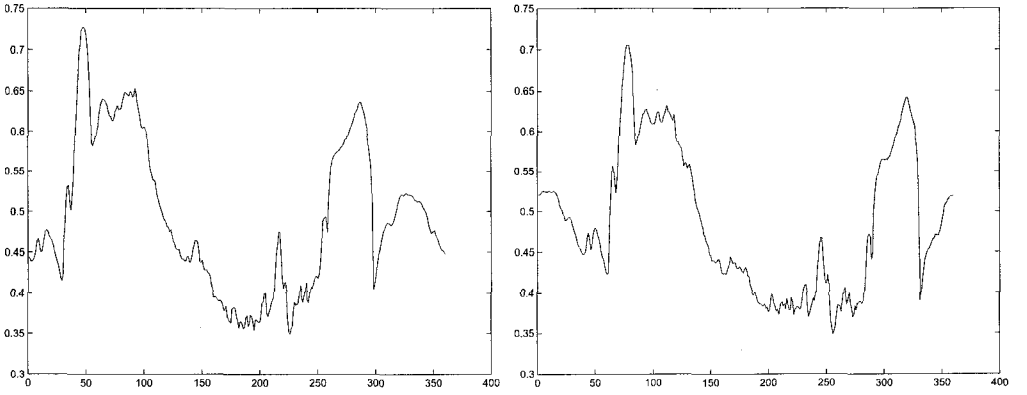
The general Radon transform includes Radial Integration Transform (RIT) and Circular Integration Transform (CIT). The CIT plot in Fig. 4.7 shows the change of



(a) CIT computation of original image *Lena*. (b) CIT computation of scaled image *Lena*.

Figure 4.7: CIT computation.

the CIT computation after scaling. The Fig. 4.7 (a) is actually a scaled version of the Fig. 4.7 (b). The scaling ratio of the CIT is the scaling ratio applied to the image *Lena*. The RIT plot in Fig. 4.8 shows the change of the RIT computation after rotation. The Fig. 4.8 (a) is actually a shifted version of the Fig. 4.8 (b). The shift is determined by the rotation degree.



(a) RIT computation of original image *Lena*. (b) RIT computation of rotated image *Lena*.

Figure 4.8: RIT computation.

From Fig. 4.7 and Fig. 4.8, the rotation degree and scaling ratio can be computed.

For the above example, the computed rotation degree is 31 degree, while the actual rotation degree is 30. The scaling ratio computed is 1.5229, and the actual scaling ratio is 1.5. So there is the inaccuracy caused by the CIT and RIT computation. The inaccuracy computed in the experiments is less than 0.05 for scaling ratio and 0.5 degree for the rotation.

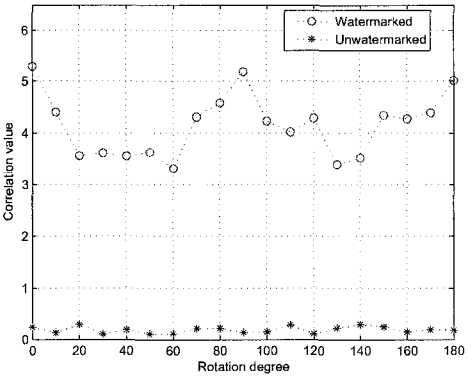
So in the experiments, the search is executed around the detected rotation degree [detected_degree − 0.5, detected_degree + 0.5] with a step of 0.1 degree and [detected_scaling_ratio − 0.05, detected_scaling_ratio + 0.05] with a step of 0.01. By doing this, the watermark detection are executed with all possible combinations of the rotation degree and scaling ratio. The largest correlation value is selected as the final detection result. This process compensates the inaccuracy of the RIT/CIT computation.

The performance of this algorithm heavily depends on the correctness of the retrieved geometrical parameters. The origin of the Radon transform computation (CIT/RIT) is vital to the success of the detection. This algorithm uses the corner detection algorithm to make sure the same corner point in the image can be detected during the embedding and detection processes. Also the Radon transform is computed using the data within the circle area whose origin is the corner point. From the statistics point of view, the more data is used for computing the CIT/RIT, the more accurate the result is. So to increase the accuracy of the Radon transform computation, the corner point used as the computation origin is normally located in the center area of the image, which makes it susceptible to attacks. The attacker could try to remove the corner point or add some corner points to make the detector unable to detect the desired corner point that is used as the computation origin during the embedding processes. Also some local geometrical distortions could result in the shift of the position of the corner point.

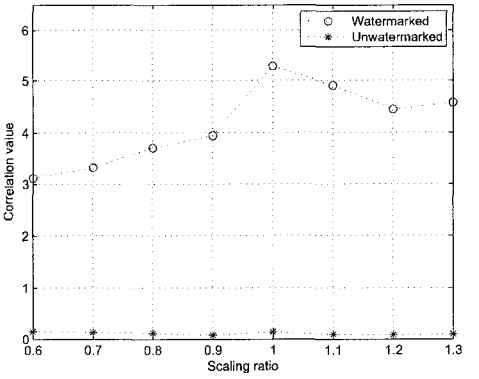
According to our experiments, the performance of this algorithm is as follows:

1. Rotation and scaling: *good*

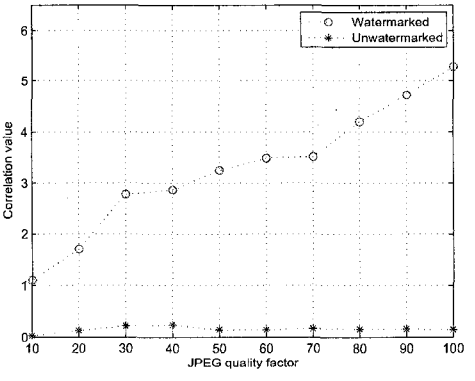
   As discussed above, once corner point used as the computation origin of the CIT/RIT can be successfully located, the performance of the watermarking algorithm will be very good. Fig. 4.9 (a) shows the performance of the watermarking algorithm under rotation and Fig. 4.9 (b) shows the performance under scaling. It is clearly shown that the correlation values computed using the watermarked



(a) Rotation.



(b) Scaling.



(c) JPEG compression.

Figure 4.9: The experimental results of the Radon transform based algorithm.

images and those using the unwatermarked images are distinguishable enough to give clear judgements of the existence of the watermark in the tested images. The threshold used for watermark detection can be easily determined and the false positive probability can be satisfactory.

2. JPEG compression and additive noise pollution: *good*

   Since the detection of the geometrical transform parameters and the watermark embedding and detection are relatively independent of each other, it makes this algorithm more flexible. To increase the robustness against the compression such as JPEG compression and noise pollution, the method similar to spread spectrum can be used to generate the watermark. The watermark is embedded with the embedding strength adaptively adjusted by the local variance and the correlation is used to detect the existence of the watermark. This watermark embedding and detection algorithm has been widely used for its robustness against various image processing techniques including JPEG compression, filtering operations and additive Gaussian noise. So this watermarking algorithm has a good performance against compression and noise pollution as shown in Fig. 4.9 (c).

The watermarking algorithms discussed in this section and Section 4.1.3 are similar except that the different approaches for the geometrical transform detection are used. From the presented experimental results, both algorithms have a similar and good performance. However because of the different approaches for the geometrical transform detection, both algorithms have its advantage and disadvantage. The Radon transform based algorithm features the blind detection which is very useful in practical applications. However its dependence on the feature point extraction makes itself susceptible to malicious attacks. Using more than one feature point and computing Radon trans-

form on every feature point can add some redundancy and increase the robustness. While this approach means less data for the Radon transform computation which will decrease the geometrical transform detection accuracy. So there is a trade-off to be taken into consideration for this algorithm. On the other hand, the algorithm discussed in Section 4.1.3 does not rely on the feature points of the image and use the small piece of the frequency information of the image to re-synchronized the geometrically distorted image based on the symmetrical phase only matching filter (SPOMF). However the watermark extraction will need the extra information which may pose some difficulty for the practical applications.

4.3 Template based algorithms

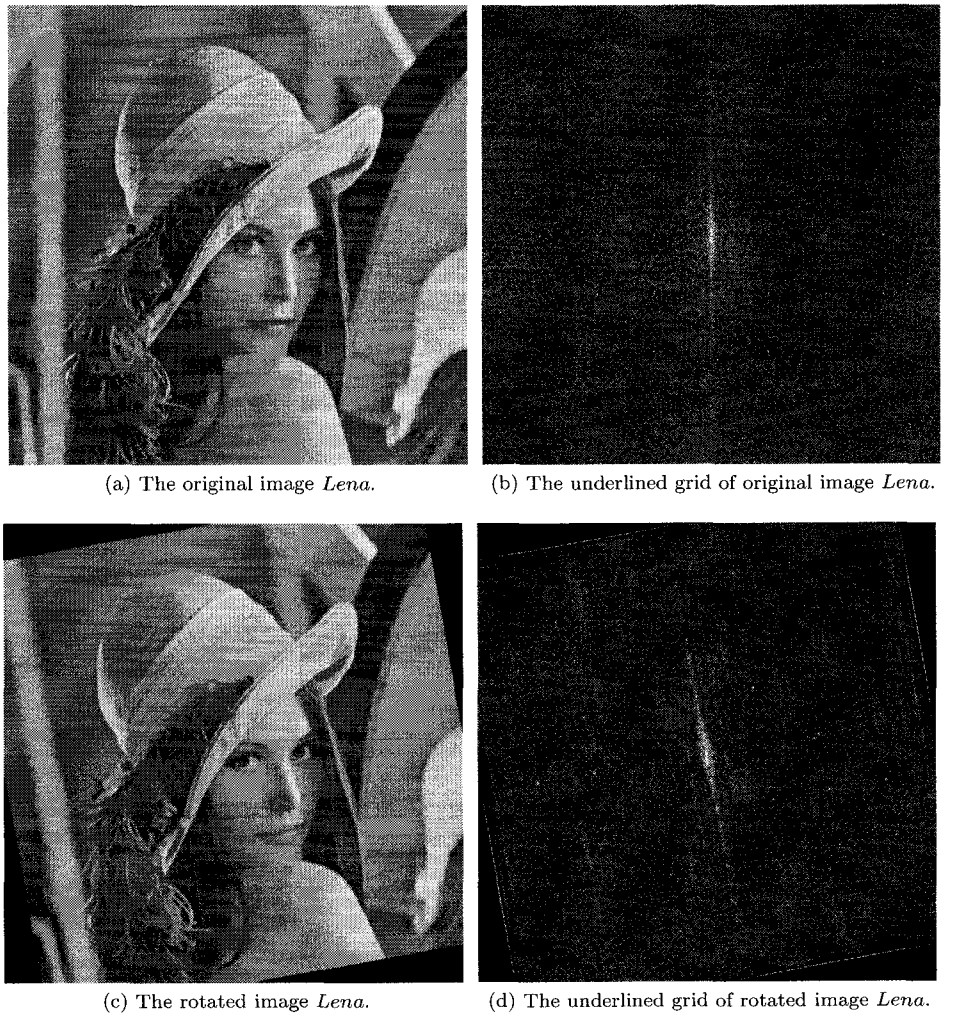
The first generation template matching based watermarking algorithms identify the geometrical transforms by retrieving artificially embedded references. In the algorithm proposed by [23], templates are located in a region corresponding to the middle frequencies of the image spectrum. Templates are generated by increasing the magnitude of selected coefficients and creating a local peak. The initial location of templates and the detected location of local maxima are matched to perform the identification of the affine transform. The algorithm is robust against JPEG compression with a quality factor as low as 75% . In all cases, the combinations of rotations, scalings and cropping were correctly recovered. The watermark can be successfully recovered when the scaling ratio is in the range of 0.75 to 2. This algorithm is vulnerable to the template attack which could locate the local maxima and remove the template [95]. Another problem is that the template consists of the local maxima, and it may degrade the fidelity of the watermarked image. In this algorithm, a message is represented as a binary se-

quence, and then the binary sequence is coded using BCH coding algorithm and the bipolar modulation is used to generate the bipolar watermark sequence. During embedding, each bit of the bipolar watermark sequence is embedded using the differential encoding algorithm. Every 2 points in the DFT magnitude domain are modified such that their differences in values indicate the embedded watermark bit. In this way, this watermarking algorithm has a larger embedding capacity than other spread spectrum based watermarking algorithms. However, for spread spectrum based watermarking algorithm, the watermark is embedded to spread over the host image such that the change to the image is very small while the robustness is good. So for this embedding mechanism, it may need a large embedding strength to get a strong robustness. Also the imprecision of the detection of the template could result in the failure of the detection since there is no error tolerance for the inaccurate position of the watermark detection. The watermark and template embedding algorithms do not take the human perceptual model into account. The embedding position and strength of the watermark and template should be carefully chosen.

Trying to solve the problem of being easily attacked for the traditional template based algorithms, [69] presented in their paper an efficient method for the watermark estimation and recovering from nonlinear or local geometrical distortions, such as the random bending attack (RBA) and restricted projective transforms. The distortions are modeled as a set of local affine transforms, the watermark being repeatedly allocated into small blocks in order to ensure its locality.

The template based watermarking algorithm embeds a periodic watermark into the host image. The watermark is retrieved using the Maximum Likelihood (ML) or Maximum A posterior Probability (MAP) algorithm. Then the ACF (autocorrelation function) is applied to the retrieved watermark to get the underlined grid of the correlation

peak.



(a) The original image *Lena*. (b) The underlined grid of original image *Lena*.

(c) The rotated image *Lena*. (d) The underlined grid of rotated image *Lena*.

Figure 4.10: The illustration of the template extraction.

Based on the geometric information of the underlined grid, the geometrical transform applied to the host image can be determined. The estimation of the affine transform parameters is formulated as a robust penalized Maximum Likelihood (ML) problem, which is suitable for the local level as well as for global distortions. In the case when no geometrical transform was applied, the message is decoded from the detected

watermark directly. If some geometrical transform was applied, based on the local ACF or magnitude spectrums, or by exploiting the reference watermark information at the block level, the geometrical distortion can be determined. Then the retrieved watermark can be processed and re-synchronized and the message is decoded.

From Fig. 4.10 (b) and Fig. 4.10 (d), we can clearly see the grid formed by applying ACF to the retrieved watermark. By comparing the geometric structure of the ACF of the retrieved template (Fig. 4.10 (b)) and the rotated one (Fig. 4.10 (d)), we can compute the rotation degree and scaling ratio.

Based on our implementation using correlation detector and test by the Checkmark, we get the evaluation of this watermarking algorithm. In the following, the performance of this template based watermarking algorithm is examined.

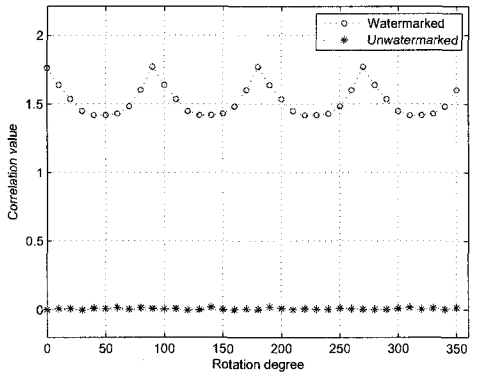
1. Rotation and scaling: *good*

   This algorithm has good performance against the rotation, scaling and translation as shown in Fig. 4.11 (a) and Fig. 4.11 (b). The geometrically transformed watermark can be resynchronized and either the correlation detection or the error control coding based detection can be executed to detect the watermark information. Based on our experiments, the results are satisfactory.

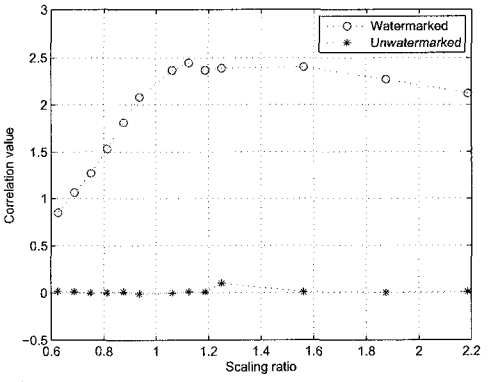
2. JPEG compression, additive noise pollution and filtering operation: *average*

   It has been mentioned by [96], that the attackers can use the available prior information about the watermark and the host image to perform the watermark removal and damage. The watermark is embedded into the host image and retrieved using the denoising algorithm such as MAP and wavelet shrinkage. Naturally, the denoising algorithms can also be used to remove the watermark or add the false watermark create the false copyright information. The idea of using lossy

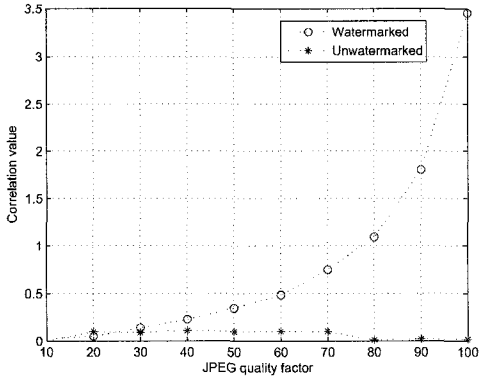
compression for denoising has been proposed in [97]. Based on our experiments as shown in Fig. 4.11 (c), the JPEG compression with quality factors under 50 will begin to affect the performance of the detection. With the quality factor decreasing further below 50, the detector will fail to detect the watermark. Similarly, the noise pollution and low pass/moving average filtering will make the detection of the watermark difficult as mentioned in the paper by [25].



(a) Rotation.



(b) Scaling.



(c) JPEG compression.

Figure 4.11: The experimental results of the template based algorithm.

3. Denoising and remodulation attack, copy attacks: *poor*

As mentioned previously, the watermark can be predicted using stochastic approach such as MAP and wavelet shrinkage. Then the watermark can be removed or manipulated. These two attacks can remove the embedded watermark to make the watermark unable to be detected. Also the predicted watermark can be maliciously added to another host image to create the false positive problems. It is called 'copy attack' in [98]. These two attacks can pose a serious problem for this watermarking algorithm. It is proposed by [99] that by exploiting the prior knowledge, specially designed watermarking algorithm can resist these two attacks.

Although we can detect the underlined grid of the correlation peak, the accuracy of the detected rotation degree and scaling ratio could be a problem for the successful detection of the watermark. In the experiments, there exist deviances between the original geometrical transform parameters and the detected ones. Some searches around the detected parameters such as scaling factor and rotation degree are necessary to give a good watermark detection result. While it is possible to detect the local geometric transform based on the change of underlined grid, it is difficult to give an accurate mathematical description like the affine transform. Thus to transform the image back to its original size/shape is difficult. In the paper by [25], it is pointed out that the small local geometrical distortions will seriously degrade the performance of the algorithms. The cropping under a reasonable range, small amount of rows/columns removing does not seriously affect the performance of this algorithm seriously.

4.4 Salient feature based algorithms

4.4.1 Salient feature and Delaunay tessellation based algorithms

The image features could be used to aid watermarking algorithms. In the work of [71], the feature points such as the salient corner points are extracted from the image using local feature detector. These points are used as the vertex and Delaunay tessellation is employed to divide the image into disjoint triangles. The watermarks are embedded into these triangles area. The geometrical distortions including the local nonlinear

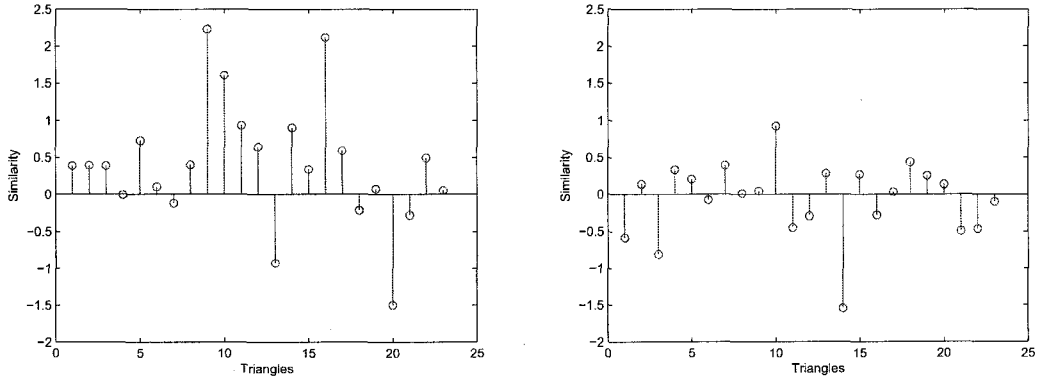
(a) Original image. (b) Watermarked image after being scaled 80%.

Figure 4.12: Delaunay tessellation with enhanced Harris corners detectors.

geometrical attacks will not change the alignments and relative positions of these points. The Delaunay tessellation can define those triangle areas for watermark embedding. For example, if a vertex disappears, the tessellation is only modified on connected triangles. Each vertex is associated with a stability area in which the tessellation is not modified when the vertex is moving inside this area. In this way, even some part of the image is

cropped or some feature points are moved because of the local nonlinear distortion or the noise introduced by various image processing processes, the triangle area used for watermark embedding should be located.

In the experiments, this salient feature based RST invariant digital image watermarking algorithm embeds and detects the watermark in each triangle of Delaunay tessellation structures [71][100]. Enhanced Harris corner detector is used to extract the robust feature points in this method [71][52]. The Delaunay tessellation is kept almost the same structures between the original image and the distorted image. Fig. 4.12 shows the Delaunay tessellations for the original image and watermarked image after scaling 80%. The watermark judgment is based on the local and global similarity to a threshold, in which the local similarity works on the similarity detection of each triangles and the global similarity is the sum of the local similarities. The analysis of the experimental results is as follows:



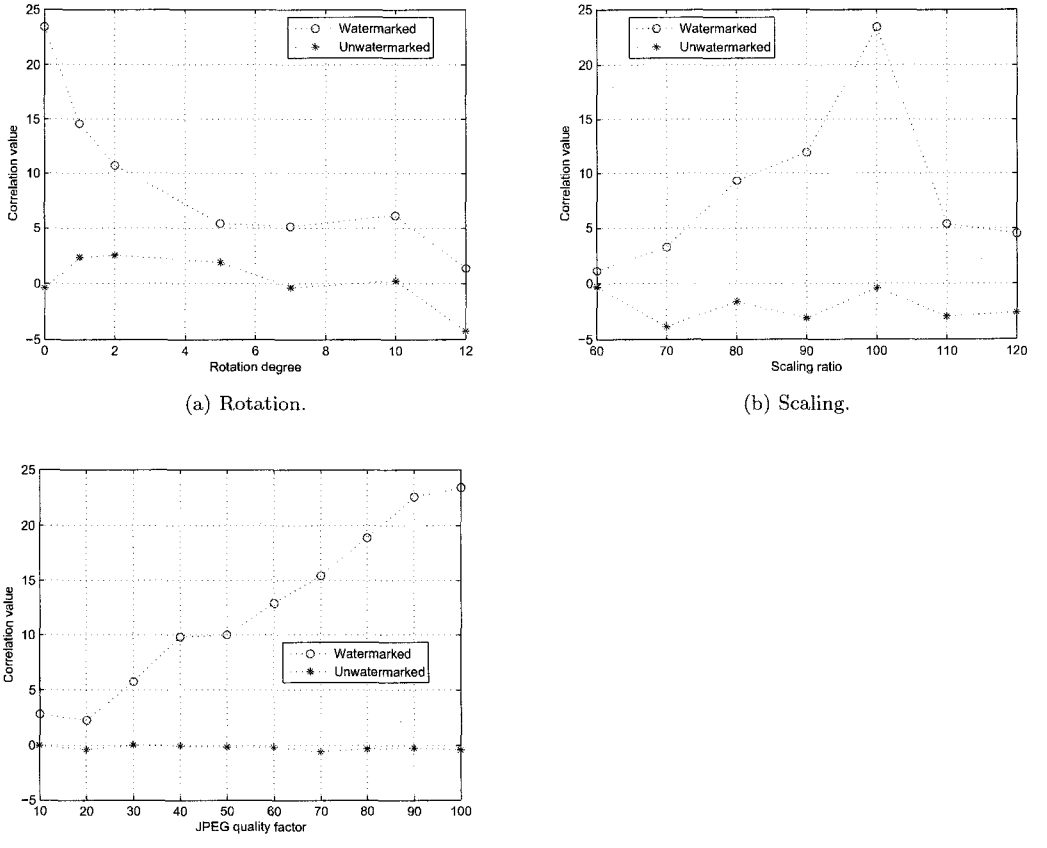
(a) Watermarked image, score = 6.65.

(b) Unwatermarked image score = −0.51.

Figure 4.13: Detection results for watermarked image and unwatermarked image after scaling 80%.

1. Rotation and scaling: *average*

This algorithm has good performance against scaling, translation and slight rotation. Fig. 4.13 shows an example of similarity detection after we perform an 80% scaling operation. The local similarity for each triangle is represented by a vertical bar. The global similarity results are shown under the figures, represented by "scores". In Fig. 4.13, the score for watermarked image is 6.65 and the score for unwatermarked image is -0.51. It proves that this method successfully detect



(a) Rotation.

(b) Scaling.

(c) JPEG compression.

Figure 4.14: The experimental results of the salient feature and Delaunay tessellation based algorithm.

if there is a watermark pattern in the test image. The method can successfully

detect the watermark for the watermarked image having undergone scaling from 60% to 120%, shown in Fig. 4.14 (a), slightly rotation up to ± 12°, shown in Fig. 4.14 (b).

2. JPEG compression, additive noise pollution and filtering operation: *good*

   The performance for JPEG compression is good. The watermarking algorithm works well even under JPEG compression quality factor 10% as shown in Fig. 4.14 (c).

There are several problems to this algorithm. First, because of the triangle formation for embedding, each embedding area is relatively small so that the capacity of the watermark embedding is limited. For the correlation detector, this could result in a large variance for the correlation result, and the performance of the detector such as the false positive probability could be degraded. When the image undergoes a geometric operation that alters the pixels, the results of the feature point detection may also be altered and consequently the watermark detection will fail. In the experiments presented in their paper, the algorithm shows robustness against the scaling operation down to 80%, a rotation of up to 10 degrees and JPEG compression with a quality factor of down to 50%. The result may not be good enough for practical applications. Because of the analysis mentioned above, the feature based watermarking algorithm has some inherited advantages and disadvantages.

4.4.2 Salient feature and image normalization based algorithms

[73] use a feature extraction method called Mexican Hat wavelet scale interaction. The extracted feature points can survive a variety of attacks and be used as reference

points for both watermark embedding and detection. The normalized image (object) is nearly invariant with respect to rotation and scaling. As a result, the watermark detection task can be much simplified when it is applied to the normalized image. Since the loss of information caused by the cropping will cause the inaccuracy of the image normalization as discussed in Section 4.6.2, the image normalization is applied to non-overlapped image regions separately in this algorithm. The regions are centered at the extracted feature points. These points are salient feature points and the small circle regions centered these points are unlikely to be affected by the cropping which normally occurs around the brim area of the image. Also the multiple circle regions can work as redundancy to increase the possibility of the successful detection of the watermark. After image normalization, the geometrically transformed circular regions will have the same direction, size and orientation as the original circular regions. So the regions for watermark embedding can be located after rotation and the watermark can be detected. While this strategy can solve the problem of cropping, it introduces other problems. The origins of those circular areas are located using some feature points. This increases the possibility of the inaccuracy since the interpolation used in rotation and scaling will cause the shift of the feature points. It is worth noting that this algorithm does not modify the moments of the image directly, it instead computes the Cartesian coordinates of those pixels to be modified based on their coordinates in the normalized domain. So the watermark is embedded into the spatial domain (or frequency domain after some orthogonal transform). In this way, the distortion caused by those unorthogonal (or deviation from orthogonality caused by computation in digital domain) transforms or coordination changes can be avoided during embedding.

The performance of this algorithm totally depends on the correct detection of feature points and the image normalization. When the watermarked image has undergone

some geometric distortion, the precision of the image normalization will be affected. In this algorithm, the small circular areas mean less statistical information for normalization computation, which make the image normalization less error-tolerant. All these disadvantages lead to the mediocre performance of the whole algorithm. The experimental results reported in the paper by [73] confirm its mediocre performance against RST as discussed above, which is mostly caused by the inaccuracy of the normalization. The performance of the image normalization based watermarking algorithm has been evaluated and discussed in Section 4.6.2. Due to its similarity to other algorithms, we have not implemented this algorithm. The following discussion about the performance of this watermarking algorithm is given based on the experimental results presented in [73], as listed in Table 4.2, Table 4.3 and Table 4.4.

1. Rotation and scaling: *average*

   The performance against rotation and scaling is mediocre, and is dependent on the successful extraction of those feature points and small size of those disk regions for watermark embedding. Since geometrical transforms introduce a lot of distortion caused by interpolation, the extracted feature points could be shifted in a small range. Since the embedding disk is only 32×32, the shift of the extracted points could be a serious problem. In the experiments, as long as the size/ratio of the image is changed, the performance of the algorithm drops quickly. [28] mentioned in their paper that "It should be noted that the normalized image suffers from smoothing effect which is a direct result of the interpolation that occurs in scaling and rotation correction. This is the price that is paid to achieve a normalized image."

   Table 4.2 shows the performace of the algorithm against rotation, scaling, and

cropping. Tables 4.2, 4.3, and 4.4 use the notation "x/y", where y means the total number of disk regions used for watermark embedding, while x means the number of disks out of y from which watermark was detected. As shown in Table 4.2, the performance of the algorithm against geometric transform is not very good. The disks are required to be non-overlapping. Thus, the feature points should be sparsely distributed. How to select those suitable feature points could pose a problem for the implementation of the algorithm.

Table 4.2: Rotation and scaling

Image	*Lena*	*Baboon*	*Pepper*
Rotation (1°) & Cropping & Scale	0/8	4/11	2/4
Rotation (2°) & Cropping	0/8	1/11	1/4
Rotation (5°) & Cropping	0/8	0/11	0/4

2. JPEG compression and additive noise pollution: *average*

Since the Mexican Haar wavelet is used to extract the feature points, most of the feature points are located around areas with low frequency since Mexican Haar wavelet is similar to the low/median pass filter. The performance of feature extraction against compression and noise is good. However the performance of the image normalization against the JPEG compression and noise is mediocre as shown in Section 4.6.2. Although the multiple disk approach can add some redundancy to increase the robustness, the small size of each disk leads to more inaccuracy of the image normalization under distortion. So the performance of this algorithm against compression and noise is only average. Table 4.3 shows the performance of the algorithm against JPEG compression and additive noise. In the table, σ means the standard derivation of the noise, while q means the quality factor.

Table 4.3: JPEG compression

Image	*Lena*	*Baboon*	*Pepper*
Additive noise ($\sigma = 0.1$)	5/8	6/11	4/4
Additive noise ($\sigma = 0.15$)	4/8	4/11	2/4
JPEG ($q = 70$)	7/8	11/11	3/4
JPEG ($q = 50$)	5/8	11/11	1/4

3. Local geometric transform: *good*

   Image normalization is rotation, scaling and translation invariant, and it is not robust against the local geometrical distortion. However, multiple disk regions centered at extracted feature points are used for watermark embedding and detection. Considering the local geometrical transform only distorts a small area of the image, the multiple-regions embedding and detection algorithm can make sure that the watermark always is detected successfully in the regions unaffected by the local geometrical distortion.

4. Median filtering, Gaussian filtering and sharpening: *average*

   Since the performance of the algorithm dependents on the extraction of the feature points, even slight shift of the feature points could cause the inaccuracy of the image normalization computation. The median filtering and sharpening make the performance of the algorithm deteriorate more than low pass filtering does as shown in Table 4.4.

Table 4.4: Filtering operations

Image	*Lena*	*Baboon*	*Pepper*
Gaussian filtering 3×3	5/8	8/11	2/4
Median filtering 2×2	2/8	6/11	0/4
Median filtering 3×3	1/8	1/11	1/4
Sharpening 3×3	4/8	2/11	4/4

4.5 Image decomposition based algorithms

4.5.1 Matched filter based algorithm

The conventional matched filter based correlator has the inherent advantage of shift invariance, but suffers from high sensitivity to geometrical transforms such as rotation and scaling. The circular harmonic expansion approach can be used for the rotation-invariant pattern. It is based on decomposing the reference pattern into a series of circular harmonic components. The circular harmonic decomposition from the polar frequencies of an image is invariant with any rotation angles [101]. A rotation invariant watermarking algorithm is proposed based on this theory [46]. A correlation filter is first designed, in which the polar-transformed version is a CHF decomposition of the polar frequencies of a watermark pattern. The cross-correlation between the watermark pattern and the watermarked image will have a peak when the rotation angles of the target image are lower than the predefined one. Meanwhile, there will not be a clear peak when the rotation angles are larger than the predefined ones.

The cross-correlation C between the correlation filter and the watermarked image can be represented by [101]:

$$c = \sum_{k=-\infty}^{\infty} C_k \tag{4.1}$$

with

$$C_k = 2\pi \int_0^{\infty} F_k(\rho) H_k^*(\rho) \rho d\rho \tag{4.2}$$

Thus, the correlation c is the sum of C_k (also called CHF weights), and k represents

the circular harmonic order. An individual C_k can be obtained from the knowledge of the k-th CHF of the polar frequencies of the watermarked image and the k-th CHF of the polar frequencies of the filter function. c shows the center value of the cross-correlation. The equation for $c(\theta)$ below gives the distribution of correlations with different rotation angles.

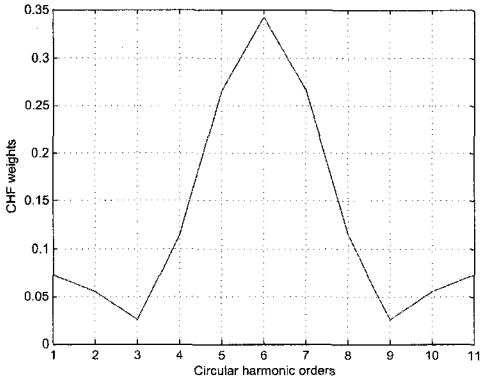
$$c(\theta) = \sum_{k=-\infty}^{\infty} C_k e^{jk\theta} \tag{4.3}$$

Ideally, the correlation output $c(\theta)$ should be 1 when the rotation angle is less than the predefined angle and should be 0 for the larger angles than the predefined one. However, the number of circular harmonic orders, k, decides the discrimination capability [101]. In order to give an example, we predefine the angle to be 45°. Fig. 4.15 shows the correlation output with different circular harmonic orders under the predefined angle 45°. When lower circular harmonic orders are applied, shown in Fig. 4.15 (a) and (d) with orders as 10, lower discrimination capability is obtained. On the other hand, when larger circular harmonic orders are used, shown in Fig. 4.15 (c) and (f) with a order of 500, a big overshoot will affect the results. As a result, we can choose the circular harmonic order as 100, which get the better discrimination, shown in Fig. 4.15 (b) and (e).

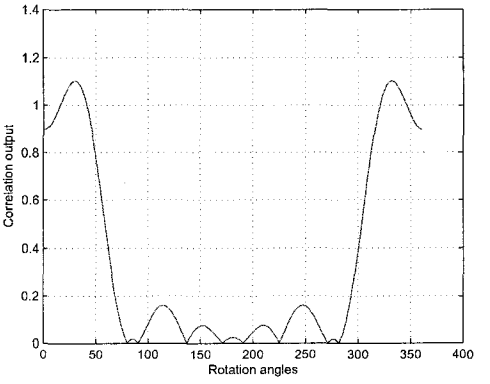
According to our experiments, the performance of this algorithm is as follows:

1. Rotation: *good*

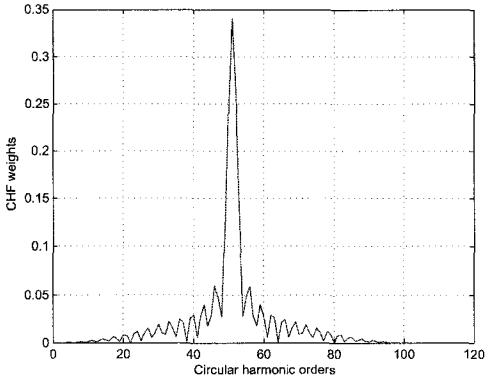
   This algorithm works well for rotation invariance. We define a predefined angle first, then, calculate the cross-correlation c according to Eq. (2.31) in Section 2.6. Experimental results show that the method has good discrimination to clarify the angles comparing with the threshold angles. Fig. 4.16 (a) and Fig. 4.16



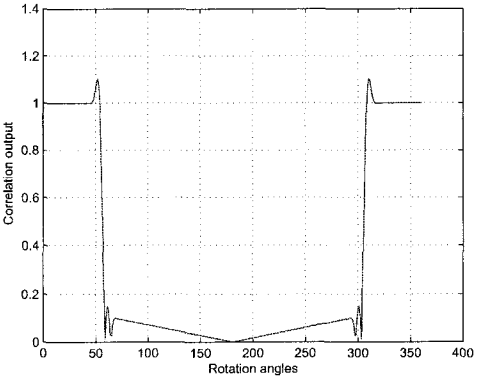
(a) C_k with circular harmonic order = 10.



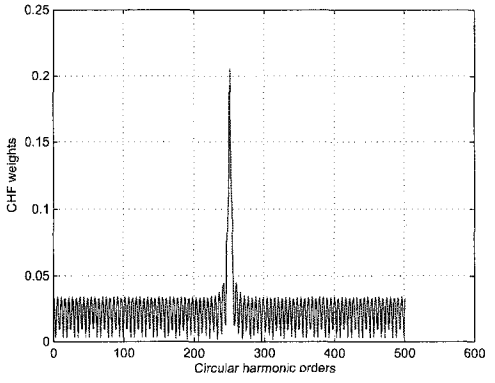
(d) c with circular harmonic order = 10.



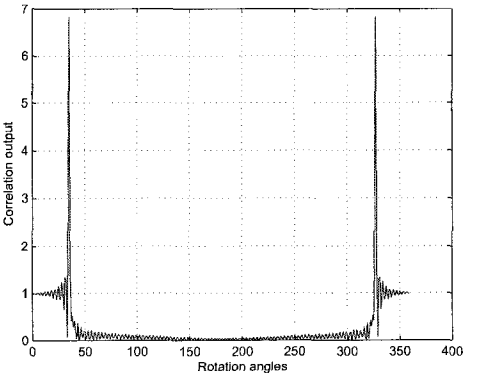
(b) C_k with circular harmonic order = 100.



(e) c with circular harmonic order = 100.



(c) C_k with circular harmonic order = 500.



(f) c with circular harmonic order = 500.

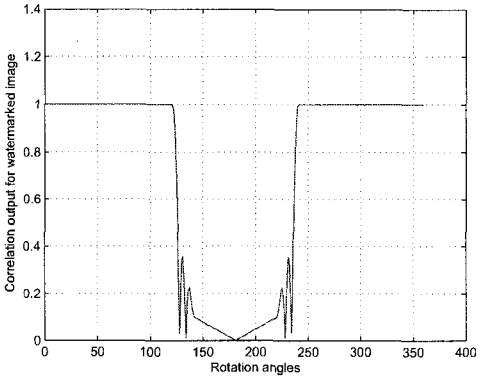
Figure 4.15: CHF weights (C_k) and correlation output (c) with different circular harmonic orders under a predefined angle 45°.

(b) show the correlation output of watermarked image and unwatermarked image with a predefined angle of 120°. Because the symmetry between rotation angles $0° - 180°$ and $180° - 360°$, we only consider the rotation between $0° - 180°$. This algorithm almost tolerates all rotation angles except for 180° because such a filter cannot be designed. Fig. 4.16 (c) and Fig. 4.16 (d) show the correlation output for rotation toleration for watermarked image with predefined angles of 170° and 179°, respectively.

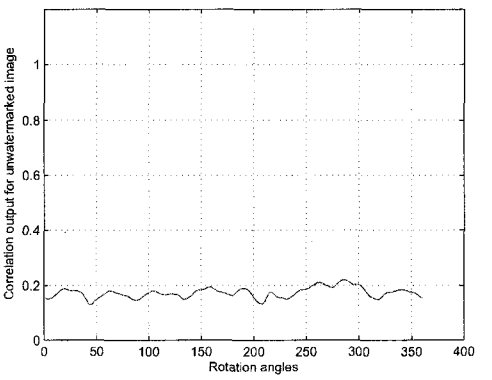
2. JPEG compression and additive noise pollution: *good*

   The algorithm is robust to JPEG compression with a quality factor down to 10%. Fig. 4.16 (e) and Fig. 4.16 (f) show the correlation output of watermarked image and unwatermarked image with the predefined angle of 45°, respectively. We try to add Gaussian noise to the watermarked and unwatermarked image with zero mean and difference variance. The experimental results show that this algorithm is robust to it.

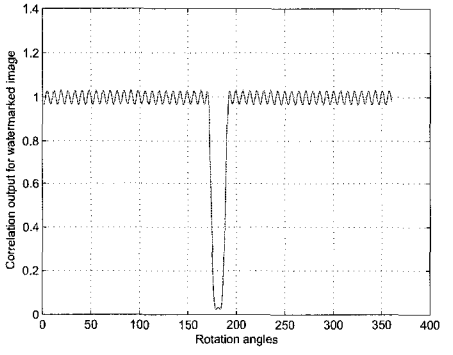
It is worth further discussing that there are several problems with the circular harmonic filter (CHF). First, by taking only a single harmonic from the expansion as a matched filter, one achieves shift and rotation-invariance. The performance of CHF strongly depends on the choice of expansion center. If the center is chosen at random, the maximum correlation does not usually coincide with the center of expansion, which means that the sidelobes will be higher than the correlation peak. The watermarking algorithm proposed is only shift and rotation invariant. There is also some tolerance range for the rotation degree that limits the used of this watermarking algorithm in practice. In pattern recognition, the radical harmonic filter (RHF) is widely used. The radial harmonic filter, similar to the CHF, can handle shift and scaling-invariant water-



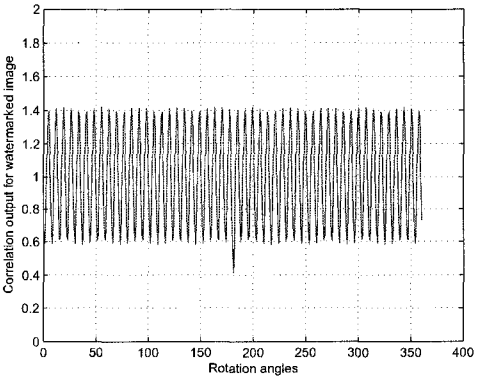
(a) Rotation of watermarked image with a predefined angle of 120°.



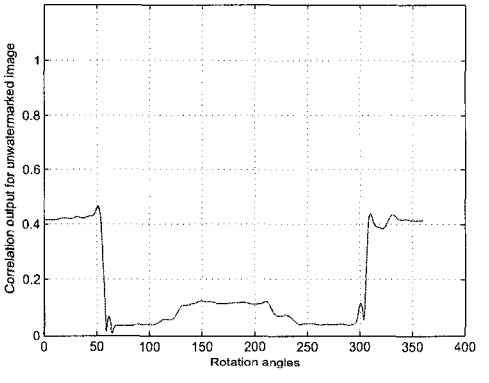
(b) Rotation of unwatermarked image with a predefined angle of 120°.



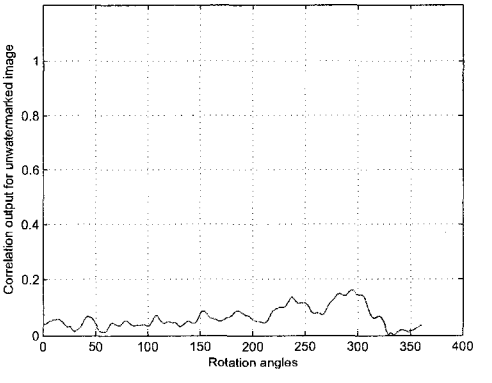
(c)
Rotation of watermarked image with a predefined angle of 170°.



(d) Rotation of watermarked image with a predefined angle of 179°.



(e) JPEG compressed watermarked image under a quality factor of 10% with a predefined angle of 45°.



(f) JPEG compressed unwatermarked image under a quality factor of 10% with a predefined angle of 45°.

Figure 4.16: Experimental results for circular harmonic based method.

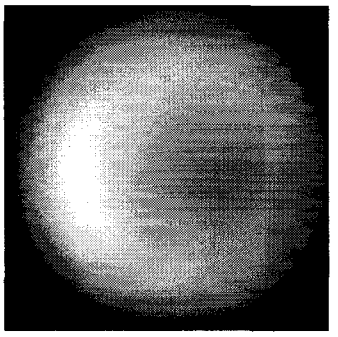
marking. The reference pattern in polar coordinates can be decomposed into a series of radical harmonic components. The RHF is obtained by selecting one radial harmonic from the expansion. The correlation between the RHF and the scaled input with scale factor has a high correlation peak. It should be noted that the RHF is different from the rotation-invariant CHF. For the CHF the peak intensity keeps constant for all rotated input patterns, so CHF is therefore fully rotation-invariant. For the RHF, however, the peak intensity is proportional to scaling ratio, so it is not fully scaling-invariant. To differentiate from the full invariance, it is called quasi-invariance. The scaling-invariance can be improved using phased-only radial harmonic filter. Though the CHF and RHF can provide rotation and scaling invariance respectively, unfortunately the CHF and RHF will not work properly under the situation when both rotation and scaling are applied to the image. Some hybrid filtering methods have been proposed to combine the CHF and RHF to achieve the rotation and scaling invariance, however such kind of filtering methods need a lot of training and the discrimination is not very good. So the use of the CHF to implement a watermarking algorithm proposed in this paper is not RST invariant. Trying to introduce the concepts in the pattern recognition into digital image watermarking is innovative, however there are still many theoretical problems to solve.

4.5.2 Pseudo-Zernike polynomial decomposition based algorithm

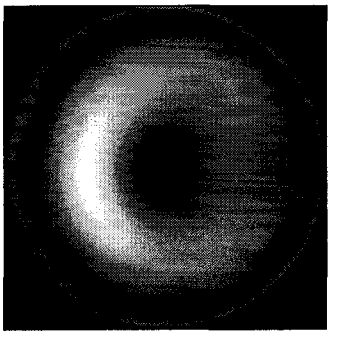
Zernike moments have been widely used in pattern recognition to retrieve the geometrical invariant features. Recently, it was used to design the watermarking algorithm as in [32]. Similar to the Fourier-Mellin transform, the approximation error of Pseudo-

Zernike polynomial integration and interpolation can cause degradation of the RST invariance property and the image quality after the Pseudo-Zernike decomposition and reconstruction. It is pointed out in [92] that "Though watermark signal can be inserted into the Zernike moments of cover image, we have found that this approach causes severe implementation difficulty. After modifying the Zernike moments of the input image, we have to reconstruct watermarked image using image from the modulated moments. However, the reconstruction procedure is computationally expensive and there is severe fidelity loss during the process.". In experiments for the algorithm in [32], it is found that it is very difficult to achieve the trade-off between fidelity and robustness. So an alternative algorithm has been proposed in [92], which gives better performance. Same as image normalization and Fourier-Mellin transform, the watermark cannot be directly embedded into the transformed image since the transformation processes and inverse transformation processes of Fourier-Mellin transform, image normalization and Zernike moments will bring too much distortion to the image, which makes the quality/fidelity of the watermarked image unacceptable. All the watermarking algorithms based on these theories use some alternative methods that embed the watermark into the untransformed domain while the embedding parameters such as embedding locations are determined by the Fourier-Mellin transform, image normalization or Zernike moments. It is worth noting that the Zernike moments itself is only rotation invariant. By combining it with image normalization, it can deal with the scaling and translation.

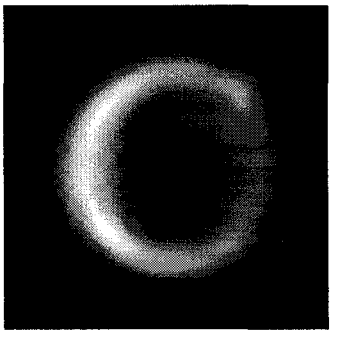
The authors of [82] proposed a geometrically robust image watermarking algorithm using Pseudo-Zernike moments. Some selected Pseudo-Zernike moments of an image are computed, and their magnitudes are quantized by dither modulation to embed an array of bits. In watermark extraction, the embedded bits are estimated from the invariant magnitudes of the Pseudo-Zernike moments using a minimum distance



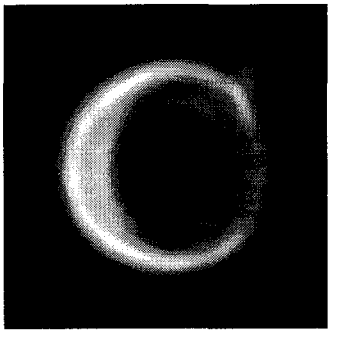
(a) Reconstructed image using up to order 5 Zernike moments.



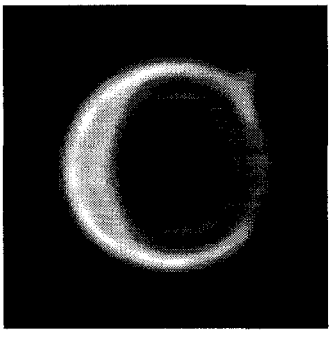
(b) Reconstructed image using up to order 10 Zernike moments.



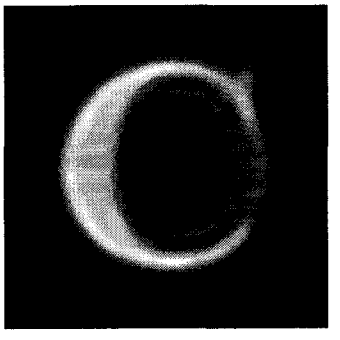
(c) Reconstructed image using up to order 20 Zernike moments.



(d) Reconstructed image using up to order 30 Zernike moments.



(e) Reconstructed image using up to order 36 Zernike moments.



(f) Reconstructed image using up to order 40 Zernike moments.

Figure 4.17: The examples of image reconstruction using Zernike moments.

decoder. The Pseudo-Zernike basis is a set of complete and orthogonal functions defined in polar domain. The expansions of the image based on the Pseudo-Zernike basis have the properties of RST invariance, which are so-called Pseudo-Zernike moments. Thus the watermark can be embedded by modifying the magnitude of the Pseudo-Zernike moments.

Table 4.5 shows the Zernike moments vectors A_{20}, A_{22}, and A_{31} (refer to Section 3.5.2 for the definition) computed under different rotation degrees, which shows the rotation invariance. Also the order of the Zernike moments computation is very important for the image/watermark reconstruction. The order of the Zernike moments has to be high enough to give a good reconstruction. From the Fig. 4.17, it can be shown that the order of 36 is optimal for the image construction and is used in all our experiments.

Table 4.5: Zernike moments vectors

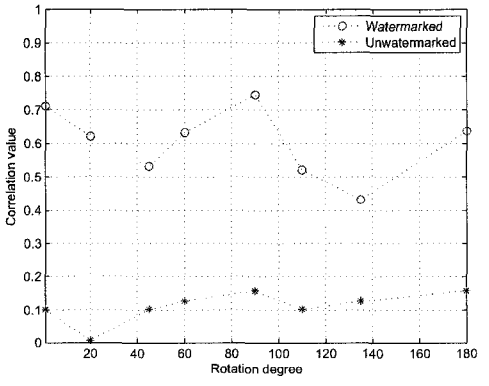
Rotation degree	A_{20}	A_{22}	A_{31}
0	339.11	31.21	152.22
45	339.27	31.45	151.98
55	339.78	31.73	152.31
135	338.21	31.11	152.44
180	339.53	31.33	153.01
310	339.29	31.97	152.39

According to our experiments, the performance of this algorithm is as follows:

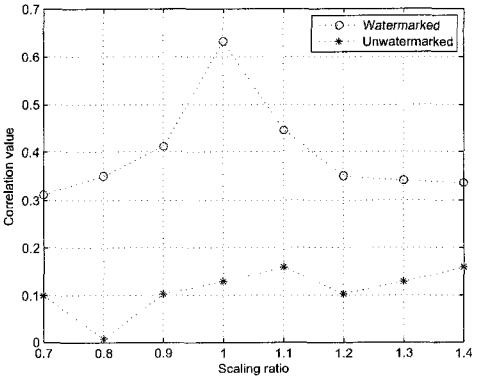
1. Rotation and scaling: *good*

   Zernike moment is rotation invariant and the scaling invariance is achieved through image normalization. The image is normalized first, and then the Zernike moments are computed. The watermark is embedded by modifying the selected Zernike moments. So the performance against scaling is worse than its performance against rotation because of the normalization. The result is shown in Fig.

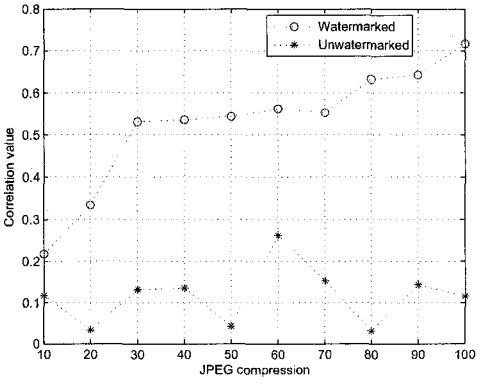
4.18 (a) and (b).



(a) Rotation.



(b) Scaling.



(c) JPEG compression.

Figure 4.18: The experimental results of the Zernike moments based algorithm.

2. JPEG compresssion, noise pollution and filtering: *good*

It has been shown that Zernike moments are superior to other moments in terms of insensitivity to image noise and image content, and it performances well against JPEG compression, Gaussian noise pollution and filtering such as Gaussian filtering and median filtering. So the Zernike moment based watermarking algorithm shows good performance against JPEG compression as shown in Fig. 4.18 (c).

In this watermarking algorithm, the watermark is embedded by quantizing the magnitude of the Pseudo-Zernike moments vector. The embedding process is controlled by the quantization step. So the change of the luminance of an image may pose a problem to the detector. Also how the Pseudo-Zernike moments reflect the visual characteristics of the original image is not clear. For example, the discrete Fourier transform can reflect the frequency distribution of the image and the discrete wavelet transform can reflect the spatial and frequency characteristics. All of these can be utilized to select the embedding locations and adjust the embedding strength. Thus the quantization step is merely an experimental value that may vary with different images. For the case of image scaling, it is assumed that if the computation region is made to cover the same contents of the image such that all the Pseudo-Zernike moments should remain unchanged. Theoretically, this can be achieved by image normalization. However, due to the desynchronization and imprecision caused by the image normalization, the scaling could cause a serious problem here. As shown in Section 4.6.2, the performance of the image normalization based algorithms is not very good. Combining the Pseudo-Zernike moments and image normalization together to design a watermarking algorithm can not handle the geometrical distortion including rotation and scaling simultaneously very well. Also to handle the translation of the image, one may need to use the translation invariant property of Fourier transform or use the feature point of the image to synchronization. These all may pose a negative effect on the performance of the Pseudo-Zernike moments based watermarking algorithms.

4.6 Stochastic analysis based algorithms

An image watermarking algorithm that is resilient to rotation, scaling, and translation (RST) is proposed by [84]. The higher order spectra (HOS), in particular, the bispectrum feature vector of an image is used. The bispectrum is the Fourier spectrum of the triple correlation of a signal. The phases of the integrated bispectra are invariant to translation and scaling. Rotation invariance is achieved using the Radon transform of the image. An image is decomposed into the 1-D projections and a feature vector is constructed from them. A watermark is embedded by modifying the vector. The distance between the feature vector extracted from the test image and the watermark at detector is measured. Results of experimental studies show that this method is robust to geometric attacks, JPEG compression, and Gaussian noise.

4.6.1 Higher order spectra based algorithm

This algorithm is similar to the algorithm proposed by [38]. The image is first polar mapped to the polar domain. Instead of adding all the elements of each column to form the one dimensional projection vector, the phase of the bispectrum integration of each column is computed. The phase of the bispectrum integration is trnslation, scaling and DC level invariant as mentioned in the paper by [86]. So hypothetically, the rotation of the image will cause the circular shift of the bispectrum phase vector and the scaling and DC level change applied to the image will not change the bispectrum phase vector. Fig. 4.19 is the bispectrum phase vector of the original image *Lena*. Fig. 4.20 shows the circular shift of the bispectrum phase vector when the image is rotated. Fig. 4.21 shows that the bispectrum phase vector will barely be affected by the scaling applied to the image.

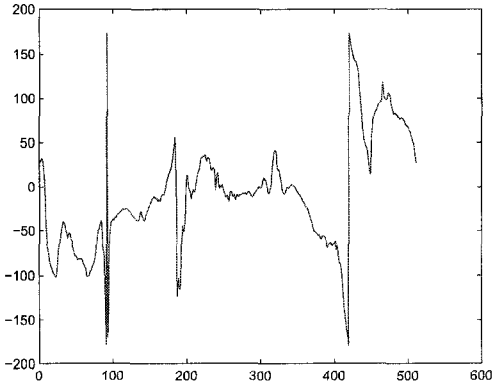
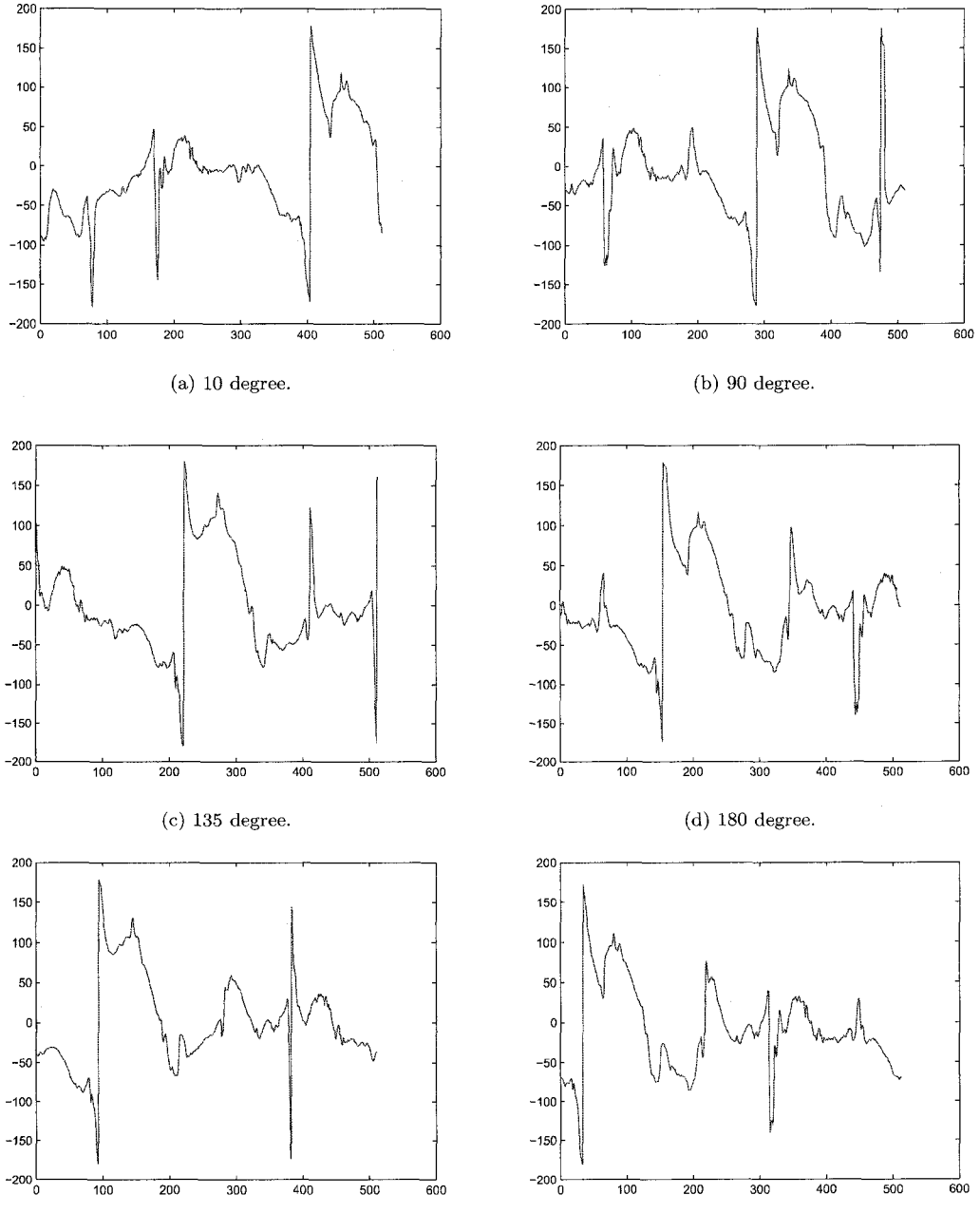


Figure 4.19: The bispectrum phase vector of the original image *Lena*.

The RST invariant watermarking algorithm can be implemented by modifying the bispectrum phase vector. However there are serious implementation difficulties. First the polar mapping is needed to extract the bispectrum phase vector and the watermark is embedded by modifying the bispectrum phase vector. After watermark embedding, the inverse polar mapping is applied to get the watermarked image. As mentioned in the paper, this will greatly affect the fidelity of the watermarked image. This is the problem most RST invariant watermarking algorithms face and try to solve. Those algorithms either use approximate mapping methods so that the image does not go through the mapping and inverse mapping processes [55] [73] or use the iterative processes to adjust the watermark embedding to minimize the distortion caused by the inverse mapping process [38]. However in this paper, it is not clearly shown how to solve the image fidelity loss caused by the polar and inverse polar mapping. From our simulation, the watermarked image shows sever quality loss. Also it is mentioned in the paper the PSNR of the watermarked image can only be maintained higher than 36 dB which is not good enough.



(a) 10 degree.

(b) 90 degree.

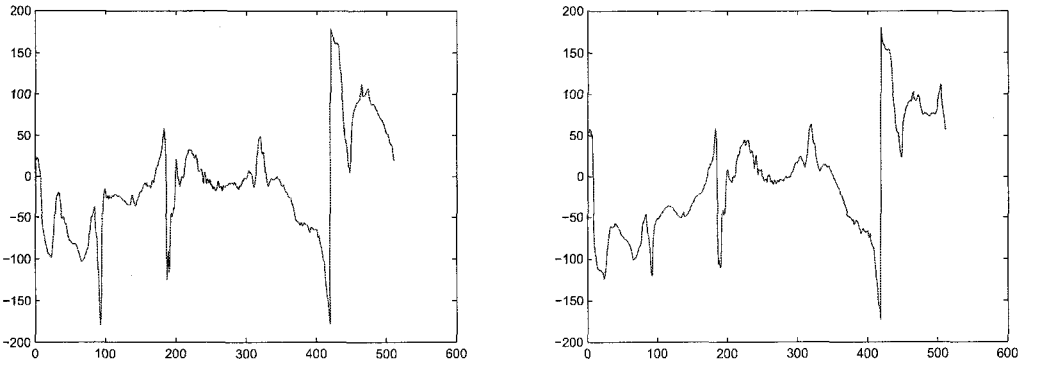
(c) 135 degree.

(d) 180 degree.

(e) 225 degree.

(f) 270 degree.

Figure 4.20: The bispectrum phase vector of the rotated image *Lena*.



(a) Scaled by 0.9.

(b) Scaled by 1.2.

Figure 4.21: The bispectrum phase vector of the scaled image *Lena*.

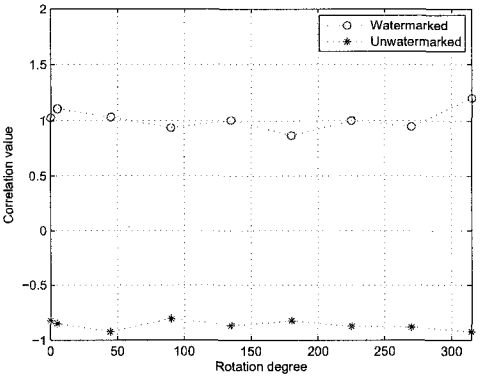
Also the modified bispectrum phase vector is used as the watermark. The iterative process is used to make sure the RMSE (root mean squared error) between the modified bispectrum phase vector and the original bispectrum phase vector is larger than the RMSE computed between the unwatermarked original image and the unwatermarked while geometrically transformed original image, and smaller than the RMSE computed between the original image and other images. Once this condition is satisfied, the modified bispectrum phase vector is accepted as the watermark. However this algorithm is questionable to maintain the acceptable false positive probability and true negative probability. Also this iterative process will affect the fidelity of the watermarked image. It is really hard to achieve a desired result while so many conditions are needed to meet during the adjustment.

In our simulation and test, it is really difficult to keep the fidelity of the watermarked image while the robustness of the watermark is still acceptable. In order to finish the test, we increase the robustness by sacrificing the fidelity of the watermarked image. After the embedding, the PSNR is watermarked image is around 31 dB.

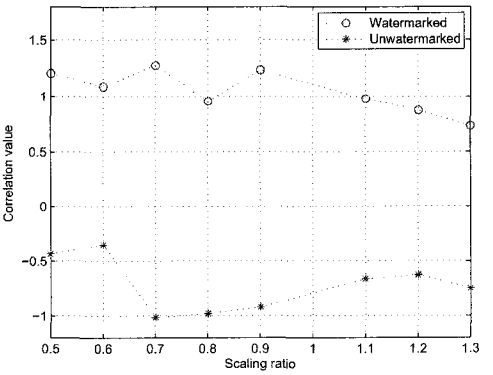
According to our experiments, the performance of this algorithm is as follows:

1. Rotation and scaling: *good*

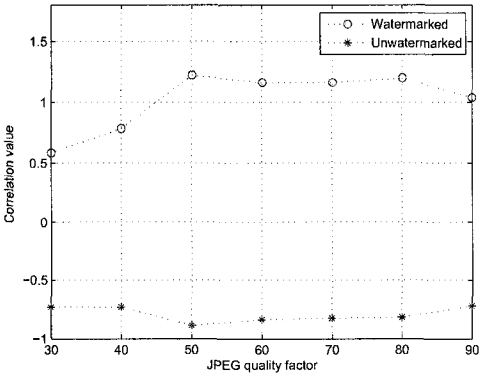
   As shown in Fig. 4.22 (a) and Fig. 4.22 (b), this algorithm has a good performance against rotation and scaling.



(a) Rotation.



(b) Scaling.



(c) JPEG compression.

Figure 4.22: The experimental results of the higher order spectra (bispectrum) based algorithm.

2. JPEG compression: *good*

   The phase information acts as an important role in determining image structure. Unlike power spectrum, the bispectrum preserves the phase information of the

Fourier transform of an image.

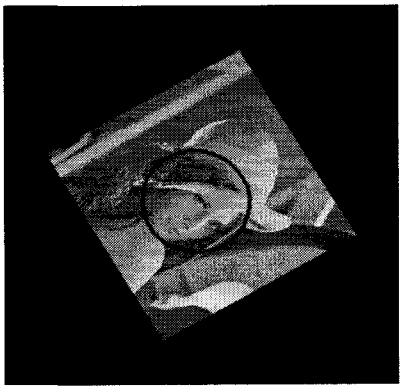
As discussed in [102], the phase of the image carries more essential information of the image than the magnitude. As the JPEG compression is to remove the spatial redundancy so that most of the essential information should be preserved after the JPEG compression. This is also shown in [61], the phase only filter shows much better performance than the amplitude only filter under the noise pollution and compression. So this algorithm also shows a good performance against the JPEG compression because of the bispectrum.

4.6.2 Image normalization based algorithm

[28] proposed a geometric invariant watermarking algorithm based on moments and image normalization. The image normalization can make the geometrical transformed image and the original image the same in size, direction and orientation. Thus the geometrically transformed image can be re-synchronized for watermark detection.

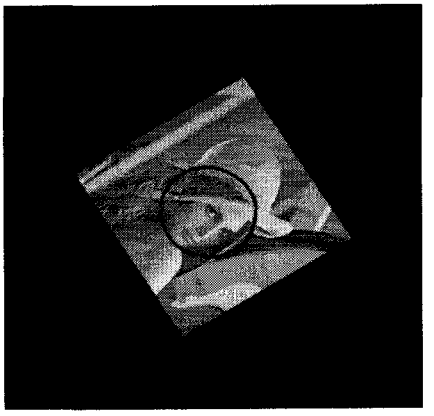
Image normalization is to exploit the statistics parameters of the image to get the rotation, scaling and invariance property. One drawback of the image normalization is that it will cause severe image quality loss since the image normalization consists of rotation and scaling operation, and normally the rotation and scaling involve interpolation. For example, after image *Lena* goes through image normalization and inverse operation, lots of the details of the original image are lost due to the interpolation. Due to this reason, the watermark is not directly embedded into the normalized domain. Instead, the watermark is embedded into the original image. First the desired embedding location in the normalized domain is determined, and then the corresponding location in the original image is computed through the inverse normalization operation. Finally

(a) The original image *Lena*.



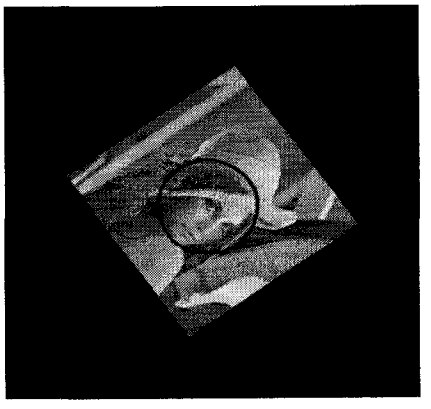
(b) The normalized image *Lena*.

(c) The rotated image *Lena*.



(d) The normalized image *Lena*.

(e) The rotated image *Lena*.



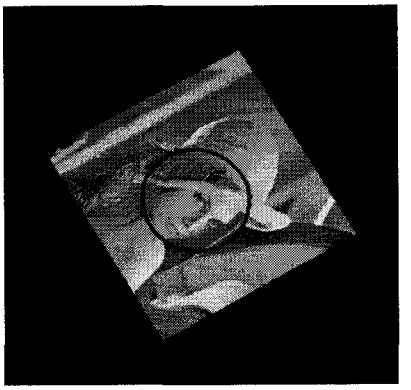
(f) The normalized image *Lena*.

Figure 4.23: The examples of image normalization under rotation.

the watermark is embedded into the original image.

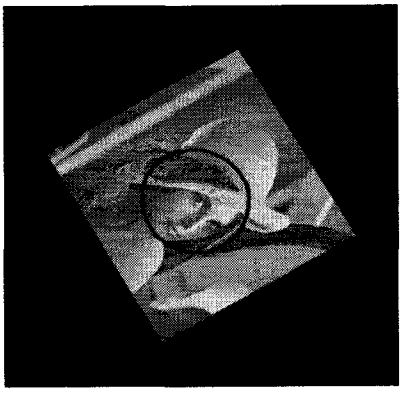
We analyzed the performance of the image normalization. The image normalization cannot resist cropping, because it needs the complete information of the image to compute the normalize vectors to normalize the image. Fig. 4.23 displays the example of the image normalization. The images having undergone different geometrical transforms can be normalized to the same size and direction.

(a) The enlarged image *Lena*, scaled up by 1.2.



(b) The normalized image *Lena*.

(c) The shrinked image *Lena*, scaled down by 0.8.



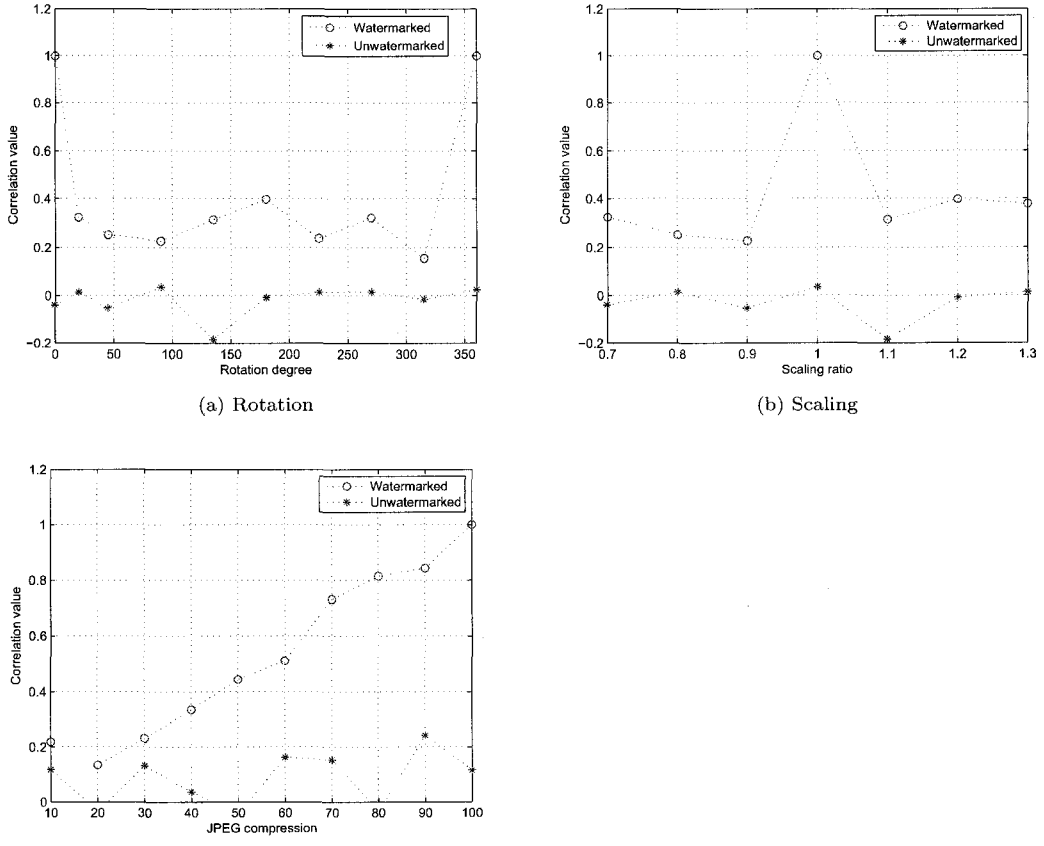
(d) The normalized image *Lena*.

Figure 4.24: The examples of image normalization under scaling.

In our experiments, we make sure that the rotation and scaling do not involve cropping. The performance of this algorithm is as follows:

1. Rotation and scaling: *average*

   The normalization operation will rotate and scale the images, which will cause the severe distortion to watermark. Also when the watermarked image has under-



(a) Rotation

(b) Scaling

(c) JPEG compression.

Figure 4.25: The experimental results of the image normalization based algorithm.

gone some geometric distortion, the precision of the image normalization will be affected. The distinction between the detection values with watermark or without

watermark is not clear enough to give a low false positive probability. As shown in Fig. 4.25 (a) and (b), the performance of the watermarking algorithm under rotation and scaling is not satisfactory.

2. JPEG compression and additive noise pollution: *average*

   This algorithm has a mediocre performance against JPEG compression as shown in Fig. 4.25 (c) and additive noise pollution. Due to the information loss caused by the compression or noise pollution, the accuracy of the normalization will be affected. Consequently, the performance of the whole algorithm will be degraded.

As we have discussed, the information loss of the image caused by cropping can make quite a difference to the computer normalization vectors. Thus, the cropped images can hardly be normalized to their original size and shape. Once the synchronization between the cropped images and the uncropped images can not be achieved through the image normalization, the successful detection of the watermark is almost impossible. To solve this problem, certain algorithms have been proposed. Detailed analysis can be seen in Section 4.4.

The performance of the image normalization based algorithms is worse than expected in the experiments. From the observation, the major reason is due to the failure of resynchronization. This problem should be solvable if some auxiliary algorithms can be proposed to increase the accuracy of the image normalization. The image normalization processes is also susceptible to other interferences such as noise, interpolation used in geometrical transforms.

4.6.3 Summary

The Fourier-Mellin transform is to transform the image into the RST invariant domain.

One problem with this method is that it is difficult to implement. The log-polar and inverse log-polar mapping process uses interpolation that causes a degradation and fidelity loss. So some methods [55] [56] [38] use the log-polar mapping instead of the Fourier-Mellin transform. The LPM can convert the rotation and scaling in the spatial domain to the translation in the LPM domain, which is easy to deal with. One dimensional projection [38] or image registration related techniques [55] [56] have been used to solve the translation in the LPM domain. They all share some similarity in theory and have their own advantages and problems.

Using a template to identify geometrical transforms is a straightforward idea. However the template-based watermarking schemes insert the template into the image by manually increasing the energy of some points or regions. This makes the template recognizable and removable by image processing such as compression and geometrical transforms, while also makes it easy to be detected by attackers. Newly proposed template-based watermarking schemes generate the information bearing template, embed and extract the watermark based on the stochastic models and analysis, which makes it mathematically convincible.

Using the salient features of an image such as corner points, centroid of the homogeneous regions can serve as the same reference purpose as the template to locate the watermark embedding and extraction region. Since these salient features are part of the image, they are better than the template.

Some researches are focusing on the exploitation of the geometrical transform invariance property of the image contents. One method is the Radon transform. The one-dimensional projection, such as Radon transform, can be used to exploit some geometrical transform invariance property. The watermarking scheme proposed by [42] utilizes both the salient features and the Radon transform, which works quite well.

Other content based schemes decompose the watermark and image into polynomial components. Some of these components are RST invariant, we can either embed the watermark into these components or use the component as the matching filter. Stochastic analysis is widely used in the error probability analysis. Now the stochastic analysis can be used to get the RST invariant content of the image such as moments, image normalization, and bispectrum.

For the methods of the watermark generation, spread spectrum is widely used because of the security and robustness it can grant to the watermarking schemes. Also various coding schemes, such as differential coding and lattice coding, are used to increase the embedding capacity. Error control coding are used to increase the error tolerance and correction ability. The proper combination of the watermark generation method and geometrical invariance schemes are yielding satisfactory results. How to achieve better trade-off between the robustness and fidelity is an important issue to address for almost every robust watermarking scheme. Various empirical human perceptual models such as the middle frequency embedding method or strong embedding for large local variance are widely used. Recently advanced mathematical models such as noise visibility functions have been proposed to use the stochastic model to analyze the statistics property of the image.

All the proposed RST invariant image watermarking schemes have their own advantages and disadvantages. And the overview of all these existing algorithms can provide good knowledge and background to design the new algorithm.

Chapter 5

The proposed RST invariant watermarking scheme

Several topics about the stochastic models and analysis of images are introduced in following sections. This knowledge is very important for our proposed RST invariant watermarking algorithm. First, the Mixture Generalized Gaussian model is used to approximate the image mathematically. Maximum a Posteriori probability (MAP) is used as the part of segmentation algorithm to segment the image into homogeneous regions, which make the mixture Generalized Gaussian model of the image feasible. Also the Expectation Maximization (EM) is used to give an estimation of parameters of the mixture Generalized Gaussian model, which is very efficient and important for both the MAP segmentation and Mixture Generalized Gaussian approximation.

5.1 Stochastic theories used in the research

5.1.1 Markov Random Field

Markov random field is a powerful tool used in statistical image analysis. The neighboring system is defined such that: S is a finite index set, the collection $N = \{N\{i\} : i \in S\}$ of sets is called the neighboring system, if i does not belong to $N\{i\}$ and $i \in N\{j\}$ if and only if $j \in N\{i\}$. The sites $j \in N\{i\}$ are called the neighbors of i. A subset C of S is a clique if any two different elements of C are neighbors.

A random field is defined as the Markov random field on S with respect to N if and only if $P(x) > 0$ and $P(x_i|x_{S-\{i\}}) = P(x_i|x_{N\{i\}})$.

The Markov random field can be describes by the Gibbs distribution:

$$P(x) = Z^{-1}exp(-H(x)) \tag{5.1}$$

$$Z = \sum_{x \in X} exp(-H(x)) \tag{5.2}$$

where denominator Z is called the partition function and H is the energy function in the form of:

$$H(x) = \sum_{A \subset S} U_A(x) \tag{5.3}$$

which is a sum of clique potentials $U_A(x)$ over all possible cliques.

One of the example is Gauss-Markov field. It is a multivariate Gaussian distribution:

$$P_X(x) = \frac{1}{\sqrt{det(2\pi\Sigma)}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)) \tag{5.4}$$

where the μ is the expectation and the Σ is the covariance matrix. Each variable only

correlates with a few neighbors represented in the quadratic energy function, which means the Σ^{-1} is the sparse matrix.

The energy function can be defined as:

$$H(x) = \frac{1}{2}x^T Ax - x^T b \tag{5.5}$$

where A is a sparse symmetric positive matrix and $b \in \Re^n$.

The Markov random field can approximate the spatial property of the image since it reflects the local statistical characteristic of the image. However because of its complexity, it is difficult to get a closed mathematical equation result when use it as the model to analysis the watermarking process. So the finite mixture model is used instead as discussed in the following sections.

5.1.2 Gaussian distribution model

A random variable X is said to be Gaussian or normal if its density function has the form:

$$p_G(x) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-(x-\mu_g)^2/2\sigma_g^2} \tag{5.6}$$

where μ_g is the mean of the distribution, and σ_g is the standard deviation of the distribution.

5.1.3 Generalized Gaussian distribution model

A random variable X has generalized Gaussian distribution, if its density function has the form [103]:

$$p_{GG}(x) = \frac{1}{2\Gamma(1+\frac{1}{\alpha}) \cdot \sigma_{gg}\sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}} \cdot exp(-|\frac{(x-\mu_{gg})}{\sigma_{gg}\sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}}|^{\alpha}) \tag{5.7}$$

where α is the shape parameter which ranges from 0.3 to 2. σ_{gg} and μ_{gg} are the standard deviation and mean of the generalized Gaussian, respectively. And the gamma function is defined as:

$$\Gamma(x) = \int_0^{\infty} t^{x-1}e^{-t}dt \tag{5.8}$$

where $x > 0$.

By setting $a = \Gamma(1+1/\alpha)$ and $b = \sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}$, Equ. (5.7) becomes:

$$p_{GG}(x) = \frac{1}{2ab \cdot \sigma_{gg}} \cdot exp(-|\frac{(x-\mu_{gg})}{\sigma_{gg}b}|^{\alpha}) \tag{5.9}$$

When $\alpha = 2$, recall the property of gamma function $\Gamma(x+1) = x\Gamma(x)$, we have:

$$a = \Gamma(1+1/2) = \frac{1}{2}\sqrt{\pi}$$

$$b = \sqrt{\frac{\Gamma(1/2)}{\Gamma(3/2)}} = \sqrt{\frac{\sqrt{\pi}}{\frac{1}{2}\sqrt{\pi}}} = \sqrt{2}$$

Equ. (5.7) becomes:

$$p_{GG}(x) = \frac{1}{\sqrt{2\pi}\sigma_{gg}} exp(-\frac{(x-\mu_{gg})^2}{2\sigma_{gg}^2}) \tag{5.10}$$

which is a Gaussian distribution.

When $\alpha = 1$,

$$a = \Gamma(1+1) = 1$$

$$b = \sqrt{\frac{\Gamma(1)}{\Gamma(3)}} = \frac{1}{\sqrt{2}}$$

Therefore, Equ. (5.7) becomes:

$$p_{GG}(x) = \frac{1}{\sqrt{2}\sigma_{gg}} exp(-\frac{\sqrt{2}|x-\mu_{gg}|}{\sigma_{gg}}) \tag{5.11}$$

which is a Laplace distribution.

When $\alpha = 0.5$, the Generalized Gaussian becomes a Gamma distribution.

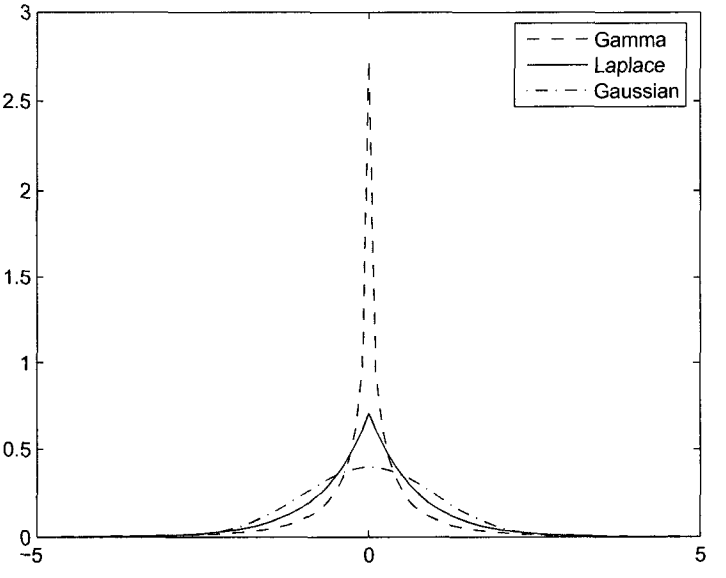


Figure 5.1: The generalized Gaussian distribution.

5.1.4 Parametric mixture probability density model

The parametric mixture density model is widely used in various research fields, such as image segmentation and pattern recognition, because of its mathematical flexibilities. A parametric mixture probabilistic model is normally defined with unknown parameters Θ as the following:

$$p(x|\Theta) = \sum_{i=1}^{N} m_i p_i(x|\theta_i) \tag{5.12}$$

where $\Theta = (m_1, m_2, \cdots, m_N, \theta_1, \theta_2, \cdots, \theta_N)$. $p_i(x|\theta_i)$ is the multivariate distribution model parameterized by θ_i. m_i is weighting parameter of $p_i(x|\theta_i)$ and it satisfies that $\Sigma_{i=1}^{N} m_i = 1$.

Equ. (5.12) can also be interpreted as that N probability density models are mixed together in weights of m_i.

In image modelling, an image cannot be simply described using a single mathematical model. But an object or a part of an object can normally be modelled using a Gaussian distribution model because the pixels within it have the similar characteristics. Therefore, an image which consists of various objects can be modelled using a mixture probability density model. In this case, $p_i(x|\theta_i)$ is Gaussian distribution density function. And the unknown parameter θ can be $\{\mu, \sigma^2\}$.

5.1.5 Maximum likelihood and maximum a posteriori probability

Maximum likelihood estimation is one of the most popular statistical method that can be used to calculate the best way of fitting a mathematical model to some data. It is

mathematically defined as:

$$p(X|\Theta) = \prod_{i=1}^{N} p(x_i|\Theta) = \mathcal{L}(\Theta|X) \tag{5.13}$$

where $p(X|\Theta)$ is a density function with a set of unknown parameter Θ. The random variable $X = \{x_1, x_2, \cdots, x_N\}$ is a sequence of i.i.d. distributed observations or samples in size of N. The function $\mathcal{L}(\Theta|X)$ is called the likelihood function and is a function of parameter Θ where the data X is fixed.

To accomplish the maximum likelihood estimation, it is desired to find the $\hat{\Theta}$ that maximizes $\mathcal{L}(\Theta|X)$, which can be interpreted as:

$$\hat{\Theta} = \arg\max_{\Theta} \mathcal{L}(\Theta|X) \tag{5.14}$$

To make the analysis easier, the log maximum likelihood $\ln(\mathcal{L}(\Theta|X))$ is normally maximized instead of $\mathcal{L}(\Theta|X)$.

Also the non-linear MAP can give more accurate estimation than ML (maximum likelihood) estimation. Here it is assumed that the prior distribution $f(\Theta)$ is available. The MAP is used as following:

$$\hat{\Theta} = \arg\max_{\Theta} \prod_{i=1}^{N} p(x_i|\Theta) f(\Theta) \tag{5.15}$$

To give a clearer demonstration, the use of ML and MAP for the image denoising is presented as following:

The degradation model can be expressed as:

$$y = x + n \tag{5.16}$$

where y is the noisy signal, x is the original signal and n is the noise. This is very similar to the watermark embedding if we treat the watermark as the noise.

For the denoising process, the goal is to estimate the original signal x from the noisy signal y. From the MMSE (minimum mean squared error) estimation, we get:

$$\widehat{x}_{MMSE} = \overline{x} + C_{xy}C_{y^{-1}}(y - \overline{y}) \tag{5.17}$$

To simplify the estimation, let us assume both x and n are Gaussian processes with respective mean $\overline{x}$ and 0, auto-covariance C_x and C_n. C_{xy} is the cross-covariance of x and y.

So that:

$$\begin{aligned} \widehat{x}_{MMSE} &= \overline{x} + C_{xy}C_{y^{-1}}(y - \overline{y}) \\ &= \overline{x} + C_{xy}C_{y^{-1}}(y - \overline{x}) \\ &= \overline{x} + \tfrac{C_x}{C_x+C_n}(y - \overline{x}) \\ &= \tfrac{C_n}{C_x+C_n}\overline{x} + \tfrac{C_x}{C_x+C_n}y \end{aligned} \tag{5.18}$$

To further simplify the analysis, it is well known that after application of wavelet transforms, the wavelet coefficients can be approximated as i.i.d Gaussian with zero mean and unit variance. Let's apply the orthonormal wavelet transform to:

$$y = x + n \tag{5.19}$$

we get:

$$Y(k) = X(k) + N(k) \tag{5.20}$$

Based on the result above we can get:

$$\widehat{X(k)}_{MMSE} = \frac{C'_{X(k)}}{C'_{X(k)} + C'_{N(k)}} Y(k) \tag{5.21}$$

So for the image denoising scheme, we can apply the discrete wavelet transform to the noisy image y, then we get:

$$\widehat{X(k)}_{MMSE} = \frac{\sigma^2_{X(k)}}{\sigma^2_{X(k)} + \sigma^2_{N(k)}} Y(k) \tag{5.22}$$

here $Y(k)$ is the observed noisy image and the $\sigma^2_{X(k)}$ is the unknown prior knowledge, the variance of the original image. However, we can estimate it from the $Y(k)$ since it contains the most part of the information of the $X(k)$.

Based on the principle of ML (Maximum Likelihood) estimation, we use a square window $W(k)$ centered at the pixel $Y(k)$. Assuming the correlation between $Y(k)$ and its neighboring pixels in $W(k)$ are very high; they all have roughly the same variance σ^2_k. Then we can use ML estimation:

$$\hat{\sigma}^2_{X(k)} = \arg\max_{\sigma^2} \prod_{j \in W(k)} P(Y(j)|\sigma^2) \tag{5.23}$$

Based on the analysis above, $P(Y(j)|\sigma^2)$ is a Gaussian distribution with zero mean and variance $\sigma'^2 = \sigma^2 + \sigma^2_{N(k)}$. Then we can get:

$$\hat{\sigma}^2_{X(k)} = \arg\max_{\sigma^2} \sum_{j=1}^{n} ln\left(\frac{1}{\sqrt{2\pi}\sigma'} e^{-\frac{Y^2(j)}{2\sigma'^2}}\right) \tag{5.24}$$

here $ln(\bullet)$ is used to simplify the computation.

$$\sum_{j=1}^{n} ln\left(\frac{1}{\sqrt{2\pi}\sigma'}e^{-\frac{Y(j)^2}{2\sigma'^2}}\right) = -nln(\sqrt{2\pi}) - \frac{n}{2}ln(\sigma'^2) - \sum_{j=1}^{n}\frac{Y^2(j)}{2\sigma'^2} \tag{5.25}$$

After applying the derivative to it, we can get the result:

$$\hat{\sigma}^2_{X(k)} = max\left(0, \frac{1}{n}\sum_{i=1}^{n} Y^2(j) - \sigma_n^2\right) \tag{5.26}$$

The non-linear MAP can give a more accurate estimation result, given some prior knowledge. Here assuming the prior distribution pdf function is available, we can use MAP as following:

$$\hat{\sigma}^2_{X(k)} = \arg\max_{\sigma^2}\left[\prod_{j\in W(k)} P(Y(j)|\sigma^2)\right] f(\sigma^2) \tag{5.27}$$

There exists different stochastic models for the prior distribution of σ^2. Here we give an approximation to the prior distribution using the exponential distribution:

$$f(\sigma^2) = \lambda e^{-\lambda\sigma^2} \tag{5.28}$$

$$\hat{\sigma}^2_{X(k)} = \arg\max_{\sigma^2}\sum_{j=1}^{n} ln\left\{\left(\frac{1}{\sqrt{2\pi}\sigma'}e^{-\frac{Y^2(j)}{2\sigma'^2}}\right)\lambda e^{(-\lambda\sigma^2)}\right\} \tag{5.29}$$

After applying the derivative to it, we can get the result:

$$\hat{\sigma}^2_{X(k)} = max\left(0, \frac{n}{4\lambda}\left(-1 + \sqrt{1 + \frac{8\lambda}{n^2}\sum_{j=1}^{n} Y^2(j)}\right) - \sigma_n^2\right) \tag{5.30}$$

5.1.6 Expectation maximization

Expectation maximization is an efficient iterative method to find unknown parameters for maximum likelihood mixture density function, even when the given data are incomplete or having missing values [104].

Assume we have a Gaussian mixture density function as the following:

$$p(X|\mu,\Lambda) = \sum_{j=1}^{M} \alpha_j f_j(X|\mu_j,\Lambda_j) \tag{5.31}$$

where $p(X|\mu,\Lambda)$ is the Gaussian mixture density function for random variables $X = x_1, x_2, \cdots, x_N$ with unknown parameters μ and Λ. μ is the mean of the distribution. Λ is the covariance matrix of X which is defined as: $\Lambda_X = \mathcal{E}\{[X-\mu_X]^{*T}[X-\mu_X]\}$. As mentioned in Section 5.1.4, $\sum_{j=1}^{M} \alpha_j = 1$. And $f_i(x|\mu_j,\Lambda_j)$ is a single Gaussian distribution function parameterized by μ_j and Λ_j.

For a d-dimensional Gaussian distribution with mean μ and covariance Λ, we have:

$$f_j(x|\mu_j,\Lambda_j) = \frac{1}{(2\pi)^{d/2}|\Lambda_j|^{1/2}} e^{-\frac{1}{2}(x-\mu_j)^T\Lambda_j^{-1}(x-\mu_j)} \tag{5.32}$$

The log-likelihood expression for Equ. (5.31) is:

$$\begin{aligned} ln(\mathcal{L}(\mu,\Lambda,\alpha|X)) &= \ln\prod_{i=1}^{N} p(x_i|\mu,\Lambda) \\ &= \ln\prod_{i=1}^{N}\sum_{j=1}^{M} \alpha_j f_j(x_i|\mu_j,\Lambda_j) \\ &= \sum_{i=1}^{N} \ln\sum_{j=1}^{M} \alpha_j f_j(x_i|\mu_j,\Lambda_j) \end{aligned} \tag{5.33}$$

Maximum likelihood estimation is to find:

$$\begin{aligned}\{\hat{\mu}, \hat{\Lambda}, \hat{\alpha}\} &= \arg\max_{\{\mu,\Lambda,\alpha\}} (\ln \mathcal{L}(\mu, \Lambda, \alpha|X)) \\ &= \arg\max_{\{\mu,\Lambda,\alpha\}} (\ln \sum_{i=1}^{N} p(x_i|\mu, \Lambda)) \\ &= \arg\max_{\{\mu,\Lambda,\alpha\}} (\sum_{i=1}^{N} \ln p(x_i|\mu, \Lambda)) \end{aligned} \tag{5.34}$$

To find the maximum likelihood estimation of α, by applying the derivation to $\sum_i \ln p(x_i|\mu, \Lambda)$ over α_ℓ, we get:

$$\begin{aligned}\frac{\partial(\sum_{i=1}^{N} \ln p(x_i|\mu, \Lambda))}{\partial \alpha_\ell} &= \sum_{i=1}^{N} \frac{1}{p(x_i|\mu, \Lambda)} \frac{\partial p(x_i|\mu, \Lambda)}{\partial \alpha_\ell} \\ &= \sum_{i=1}^{N} \frac{1}{p(x_i|\mu, \Lambda)} \frac{\partial \sum_{j=1}^{M} \alpha_j f_j(x_i|\mu_j, \Lambda_j)}{\partial \alpha_\ell} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{1}{p(x_i|\mu_j, \Lambda_j)} \frac{\partial(\alpha_j f_j(x_i|\mu_j, \Lambda_j))}{\partial \alpha_\ell} \end{aligned} \tag{5.35}$$

where $\ell \in \{1, ..., M\}$.

Consider the probability that an observation x_i comes from the distribution $f_j(x|\mu_j, \Lambda_j)$:

$$p(j|x_i, \mu_j, \Lambda_j) = \frac{p(j, x_i, \mu_j, \Lambda_j)}{p(x_i|\mu_j, \Lambda_j)} = \frac{\alpha_j f_j(x_i|\mu_j, \Lambda_j)}{p(x_i|\mu_j, \Lambda_j)} \tag{5.36}$$

on the other hand, Equ. (5.36) can be re-written as:

$$\frac{1}{p(x_i|\mu_j, \Lambda_j)} = \frac{p(j|x_i, \mu_j, \Lambda_j)}{\alpha_j f_j(x_i|\mu_j, \Lambda_j)} \tag{5.37}$$

substituting Equ. (5.37) into Equ. (5.35), we have:

$$\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{p(j|x_i,\mu_j,\Lambda_j)}{\alpha_j f_j(x_i|\mu_j,\Lambda_j)}\cdot\frac{\partial(\alpha_j f_j(x_i|\mu_j,\Lambda_j))}{\partial\mu_j}$$
$$=\sum_{i=1}^{N}\sum_{j=1}^{M}p(j|x_i,\mu_j,\Lambda_j)\cdot\frac{\partial(\ln(\alpha_j f_j(x_i|\mu_j,\Lambda_j)))}{\partial\alpha_\ell} \tag{5.38}$$

therefore,

$$\begin{aligned}\sum_{i=1}^{N}\ln p(x_i|\mu,\Lambda) &= \sum_{i=1}^{N}\sum_{j=1}^{M}p(j|x_i,\mu_j,\Lambda_j)\cdot\ln(\alpha_j f_j(x_i|\mu_j,\Lambda_j))\\ &= \sum_{i=1}^{N}\sum_{j=1}^{M}\ln(p_j(x_i|\mu_j,\Lambda_j))p(j|x_i,\mu_j,\Lambda_j)+\sum_{i=1}^{N}\sum_{j=1}^{M}\ln(\alpha_j)p(j|x_i,\mu_j,\Lambda_j)\end{aligned} \tag{5.39}$$

Using Lagrange multiplier λ for the condition $\alpha_1+\alpha_2+\alpha_3+\cdots+\alpha_N=1$, we can achieve:

$$\sum_{i=1}^{N}\ln p(x_i|\mu,\Lambda)+\lambda(\sum_{i=1}^{M}\alpha_i-1)=0 \tag{5.40}$$

applying derivation over α_ℓ to it and setting it equal to 0, we have:

$$\sum_{i=1}^{N}\frac{p(\ell|x_i,\mu,\Lambda)}{\alpha_\ell}+\lambda=0 \tag{5.41}$$

summing both sizes over ℓ, we get that $\lambda=-N$ resulting in:

$$\hat{\alpha}=\alpha_\ell=\frac{1}{N}\sum_{j=1}^{N}p(\ell|x_i,\mu,\Lambda) \tag{5.42}$$

To achieve $\hat{\mu}$, we need to calculate $\frac{\partial\sum_{i=1}^{N}\ln p(x_i|\mu,\Lambda)}{\partial\mu_\ell}$

Recall Equ. (5.32), taking the log of it and ignore the constant terms which will disappear after taking derivatives and substitute into the right side of Equ. (5.39), we have:

$$\sum_{i=1}^{N}\sum_{j=1}^{M}\ln(p_j(x_i|\mu_j,\Lambda_j))p(j|x_i,\mu_j,\Lambda_j)$$
$$=\sum_{i=1}^{N}\sum_{j=1}^{M}(-\frac{1}{2}ln|\Lambda_j|-\frac{1}{2}(x_i-\mu_j)^T\Lambda^{-1}(x_i-\mu_j))p(j|x_i,\mu_j,\Lambda_j) \quad (5.43)$$

applying derivation over μ_ℓ to it, and setting it equal to 0, we get:

$$\sum_{i=1}^{N}\Lambda^{-1}(x_i-\mu_\ell)p(\ell|x_i,\mu,\Lambda)=0 \quad (5.44)$$

therefore, $\hat{\mu}$ can be easily solved as:

$$\hat{\mu}=\mu_\ell=\frac{\sum_{i=1}^{N}x_ip(\ell|x_i,\mu,\Lambda)}{\sum_{i=1}^{N}p(\ell|x_i,\mu,\Lambda)} \quad (5.45)$$

Equ. (5.43) can be re-written as:

$$\sum_{i=1}^{N}\sum_{j=1}^{M}\ln(p_j(x_i|\mu_j,\Lambda_j))p(j|x_i,\mu_j,\Lambda_j)=$$
$$\sum_{j=1}^{M}\{\frac{1}{2}ln|\Lambda_j^{-1}|\sum_{i=1}^{N}p(j|x_i,\mu,\Lambda)-\frac{1}{2}\sum_{i=1}^{N}p(j|x_i,\mu,\Lambda)Tri(\Lambda_j^{-1}(x_i-\mu_j)(x_i-\mu_j)^T)\} \quad (5.46)$$

where $Tri(\Lambda^{-1}(x_i-\mu_j)(x_i-\mu_j)^T)$ denotes the sum of the diagonal elements of matrix $\Lambda^{-1}(x_i-\mu_j)(x_i-\mu_j)^T$.

Set $N_{i,\ell}=(x_i-\mu_\ell)(x_i-\mu_\ell)^T$. Therefore, to find $\hat{\Lambda}$, taking derivative with respect

to Λ_ℓ^{-1} and set it equal to 0, and we will achieve:

$$\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)(2\Lambda_\ell - diag(\Lambda_\ell)) - \frac{1}{2}\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)(2N_{i,\ell} - diag(N_{i,\ell})) \\
&= \frac{1}{2}\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)(2(\Lambda_\ell - N_{i,\ell}) - diag(\Lambda_\ell - N_{i,\ell})) \\
&= \sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)(\Lambda_\ell - N_{i,\ell}) - \frac{1}{2}diag(\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)(\Lambda_\ell - N_{i,\ell})) \\
&= 0
\end{aligned} \tag{5.47}$$

Equ. (5.47) indicates that:

$$\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)(\Lambda_\ell - N_{i,\ell}) = 0 \tag{5.48}$$

then, we can solve $\hat{\Lambda}$ as:

$$\hat{\Lambda} = \Sigma_\ell = \frac{\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)N_{i,\ell}}{\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)} = \frac{\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)(x_i-\mu_\ell)(x_i-\mu_\ell)^T}{\sum_{i=1}^{N} p(\ell|x_i,\mu,\Lambda)} \tag{5.49}$$

In summary, the Maximum likelihood estimates of the new parameters in terms of the old parameters are shown in Equ. (5.50), (5.51) and (5.52).

$$\hat{\alpha} = \frac{1}{N}\sum_{j=1}^{N} p(\ell|x_i,\mu^{old},\Lambda^{old}) \tag{5.50}$$

$$\hat{\mu} = \frac{\sum_{i=1}^{N} x_i p(\ell|x_i,\mu^{old},\Lambda^{old})}{\sum_{i=1}^{N} p(\ell|x_i,\mu^{old},\Lambda^{old})} \tag{5.51}$$

$$\hat{\Lambda} = \frac{\sum_{i=1}^{N} p(\ell|x_i,\mu^{old},\Lambda^{old})(x_i-\hat{\mu})(x_i-\hat{\mu})^T}{\sum_{i=1}^{N} p(\ell|x_i,\mu^{old},\Lambda^{old})} \tag{5.52}$$

5.2 The proposed feature-based RST invariant image watermarking scheme

For the RST image watermarking algorithms discussed in Chapter 3 and 4, one of the major problems is that the lack of an accurate mathematical model to analyze and guide the watermarking processes. Among those algorithms, the most commonly used model is the Gaussian distribution model. Although the Gaussian model greatly simplified the analysis, it can not represent the complicated statistic characteristic of various natural images accurately. To solve this problem, Generalized Gaussian distribution is introduced in this thesis with the great advantage that it can approximate the image or its transformed coefficients accurately. However, because of the variety of image characteristics, different regions of image may show different stochastic characteristics. A natural way of thinking is to segment the image into different regions. Each region is represented by its own distribution. Based on the above discussion, a Gaussian Mixture distribution model and MAP image segmentation are used in our proposed watermarking scheme. In MAP image segmentation, the prior distribution is approximated as a Bayesian distribution and the conditional posterior distribution can be estimated using Expectation maximization which is detailed in Section 5.1.6 After the image segmentation, the image is segmented into several homogeneous regions and each region is an object or part of an object. The mean and variance of the distribution for each region are derived as well. Then the image can be modelled approximately as the Gaussian mixture distribution. This can guide the watermarking processes.

After the establishment of the image model, the next step is to grant the watermarking algorithm the RST invariance property. Since the image is segmented into homogenous regions, we choose to embed the watermark region-wise. Image normaliza-

tion is widely used in pattern recognition for handling rotation, scaling and translation transforms. However, as discussed in Chapter 3, the high order moment computation needed for the rotation invariance is susceptible to noise and distortion caused by interpolation. Therefore, in our scheme, image normalization is only used for the translation and scaling invariance. To achieve the rotation invariance, the local features are used. Because the normal extracted features are sensitive to noise and interpolation, we use the Scale Invariant Feature Transform (SIFT) algorithm to detect and extract the distinctive local features which should be tolerant to image noise, uniform scaling, rotation, changes in illumination and minor changes in viewing direction. Moreover, using local features also helps us to define the watermark embedding regions. In the SIFT algorithm, the Laplacian of Gaussian (LoG) and Gaussian scale model make the extracted features points robust to additive noise and interpolation. Meanwhile, the SIFT detection will be performed on the segmented image, which is actually a low-pass filtered image so that the extracted feature points can be robust against the rotation. Based on the extracted feature points and the segmented image regions, the watermark embedding regions are defined and its associated feature points are used to identify these regions during the watermark detection/extraction process. For each watermark embedding region, the orientation of its associated feature point will be calculated. The watermark embedding region will always aligned with this orientation. Combining this with image normalization, the RST invariance property can be achieved.

Several aspects of watermark embedding can be mathematically analyzed based on the established Gaussian mixture model. The first one is the adaptive watermark embedding strength which achieves better trade-off between robustness and fidelity. In our scheme, the Noise Visibility Function (NVF) is introduced to guide the watermark embedding. Also analysis about the probability of error, such as false positive probability

can be derived mathematically.

The algorithm proposed in the thesis utilized all the knowledge mentioned above and establish an advanced mathematical model for watermarking process analysis. Most of the current watermarking algorithms only use a much simplified, inaccurate model with some empirical embedding control parameters. The experimental results show that the proposed watermarking scheme works very well and is effective regarding the robustness against RST transform. Moreover, the in-depth theoretical analysis is presented.

5.2.1 Watermark embedding and watermark detection

The proposed image watermarking algorithm is shown in Fig. 5.2. The following are the steps for the watermark embedding:

1. The image is segmented into homogeneous regions using the MAP image segmentation.

2. The geometrical invariant features are extracted using the Gaussian scale model.

3. Based on the extracted feature points and the segmented regions, the watermark embedding circular regions centered at the feature points are determined. Those feature points are used as the reference points.

4. The orientations of the reference points are calculated, the image normalization and orientation alignment will make the watermark embedding regions RST invariant.

5. Based on the NVF, the watermark is adaptively embedded into those circular regions.

The watermark detection includes the following steps:

1. Using the same steps in embedding process, the watermark embedding regions are located, which are RST invariant regions.

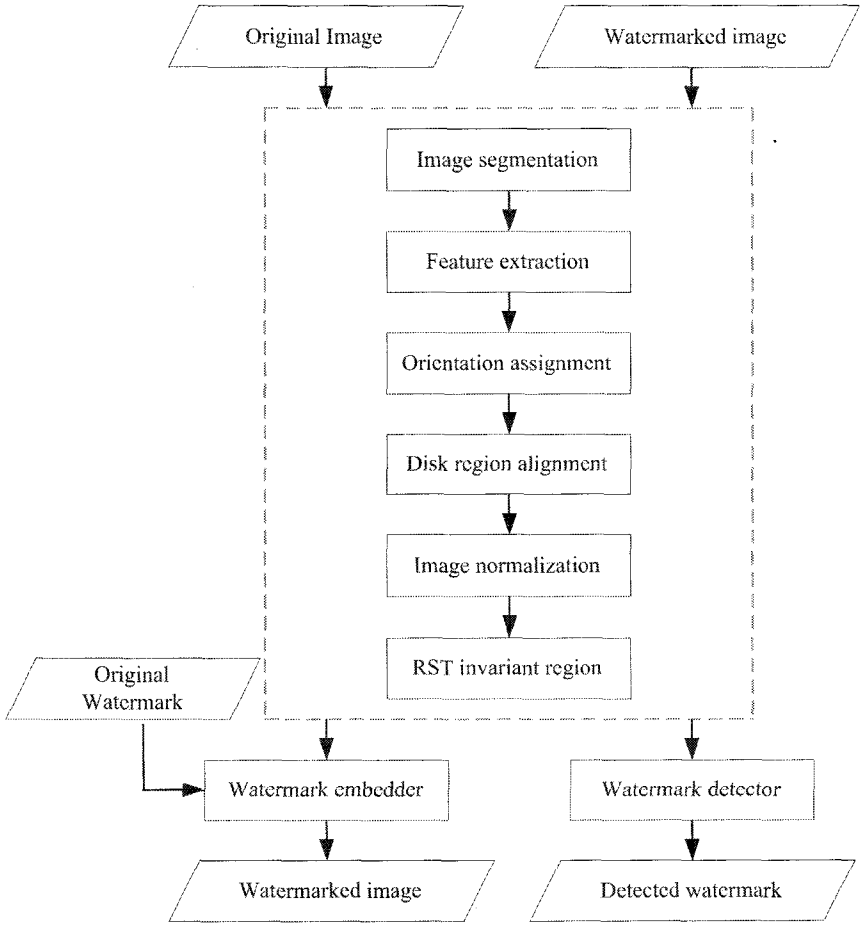


Figure 5.2: The watermark embedding and extraction scheme

2. During the watermark detection, the linear correlation, addressed in Section 2.8, is used to detect the existence of the watermark in the segmented regions. Once the value of the linear correlation is larger than a threshold, it is claimed that the

watermark is detected in that region. Multiple regions can also work together as a redundancy to increase the robustness of the watermark embedding.

In the following, the Gaussian scale model used to extract the feature points and the NVF (noise visibility function) are introduced. The implementation details are presented in Chapter 6.

5.2.2 Gaussian scale model

In our scheme, the desired distinctive feature points are located using the Gaussian scale model in the scale space. Similar to other feature detectors, the Gaussian scale model also works with a filter. Under a variety of reasonable assumptions, it is shown that the isotropic Gaussian filter is the only possible scale-space kernel to generate the desired scale space [34].

The scale space is generated by convolving the 2-D Gaussian filter, $G(x, y, \sigma)$, shown in Equ. (5.53) with the source signal.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{5.53}$$

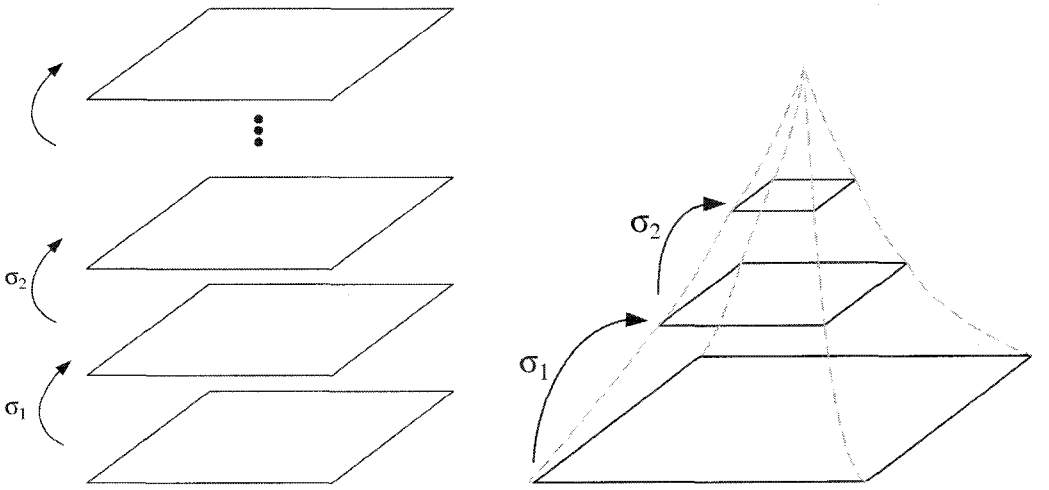
Therefore, the scale-space, $L(x, y, \sigma)$, of an image will be:

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y) \tag{5.54}$$

where $I(x, y)$ is the image, and $\otimes$ indicates the convolution operation. Fig. 5.3 shows the representations of the scale space.

The scale space can be constructed by successive smoothing versions of a high resolution images, as Fig. 5.3 (a), or constructed pyramidly by combined filtering and

sampling, as shown in Fig. 5.3 (b), or constructed by the combination of Fig. 5.3 (a) and (b).



(a) Successive filtering. (b) Scaling.

Figure 5.3: Representations of the scale space.

For the purpose of feature detection, recall that the Laplacian operator is defined as the second derivatives simultaneously along both x and y axis:

$$\nabla^2(\cdot) = \frac{\partial^2(\cdot)}{\partial x^2} + \frac{\partial^2(\cdot)}{\partial y^2} \tag{5.55}$$

Unfortunately, the above filter is highly sensitive to noise since a derivative operation in the spatial domain corresponds to an amplification of noise in the frequency domain. To solve this problem, the LoG (Laplacian of Gaussian) is used instead by combining a Gaussian filter and a Laplacian filter, which is defined as follows:

$$LoG(x, y) = \nabla^2 G(x, y) \tag{5.56}$$

It is shown in [105] that the normalization of the Laplacian filter with the factor of

σ^2 is required for the true scale invariance. And it is further found out in [106] that the extrema of $\sigma^2\nabla^2 G$ are the most stable image features comparing to the features produced by other feature detectors, which is also desired in the Gaussian scale model. To simplify the calculation, the difference-of-Gaussian (DoG), $DoG(x, y, \sigma)$, is used to approximate LoG. The DoG can be computed from the difference of the two adjacent scales separated by a constant factor, k.

$$DoG(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma) \tag{5.57}$$

As derived in [34],

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2\nabla^2 G(x, y) \tag{5.58}$$

which indicates that the DoG achieves a good approximation of LoG. The $(k-1)$ is a constant factor over all the scales, therefore it does not affect the feature locations.

Thus, the DoG of the image will be derived as:

$$\begin{aligned} D(x, y, \sigma) &= DoG(x, y, \sigma) \otimes I(x, y) \\ &= (G(x, y, k\sigma) \otimes I(x, y)) - (G(x, y, \sigma) \otimes I(x, y)) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{5.59}$$

Therefore, more than one filtered image will be needed to achieve a DoG in the scale space domain. As proposed in [34], the image is transformed to a group of n filtered images with different Gaussian scales, thus, $n-1$ DoGs will be accomplished. After that, the image will be upsampled or downsampled and transformed to another group of n filtered images in the scale space. Through one group of filtered image, the

variance σ is doubled and $k = 2^{1/(n-3)}$. In the thesis, we choose to produce 5 blurred images in one group and that indicates the scale separation, k, between two adjacent blurred images is $1/4$. Therefore, if the variance for the first image is σ, the variance of the 5^{th} image will be 2σ. And all the details are shown in Fig. 5.4.

In Fig. 5.4, the left column has all the blurred images in different scales, $(s-1)k\sigma$, $1 \leq s \leq 5$. And the corresponding *DoG* in the right column is generated by subtracting the two adjacent blurred images. Therefore, the extrema of *DoG* can be detected from the blurred images in the right column as the feature candidates.

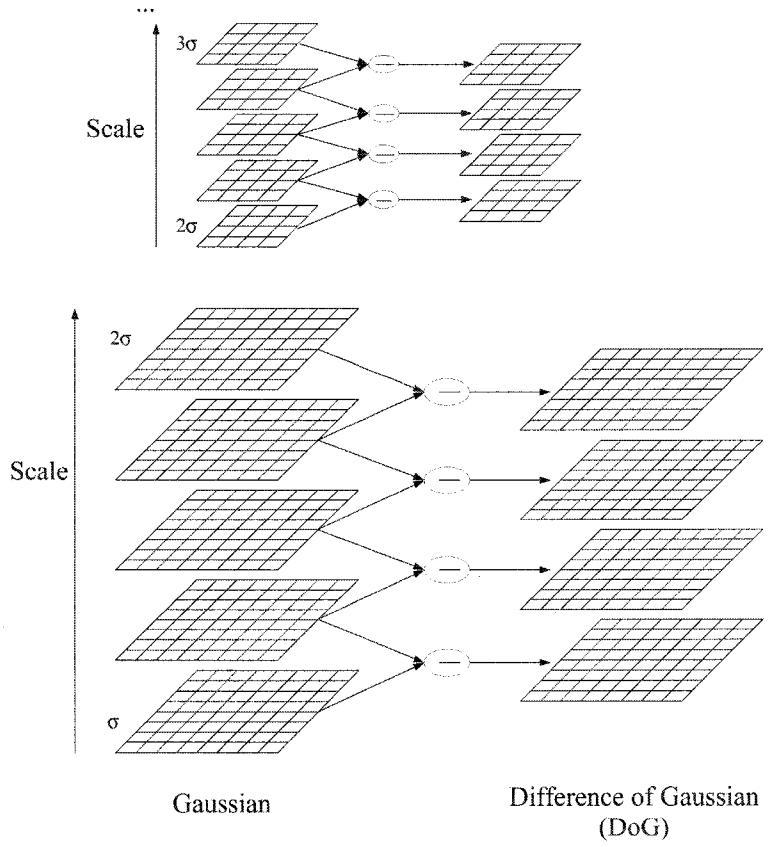


Figure 5.4: The difference of Gaussian in the scale spacc.

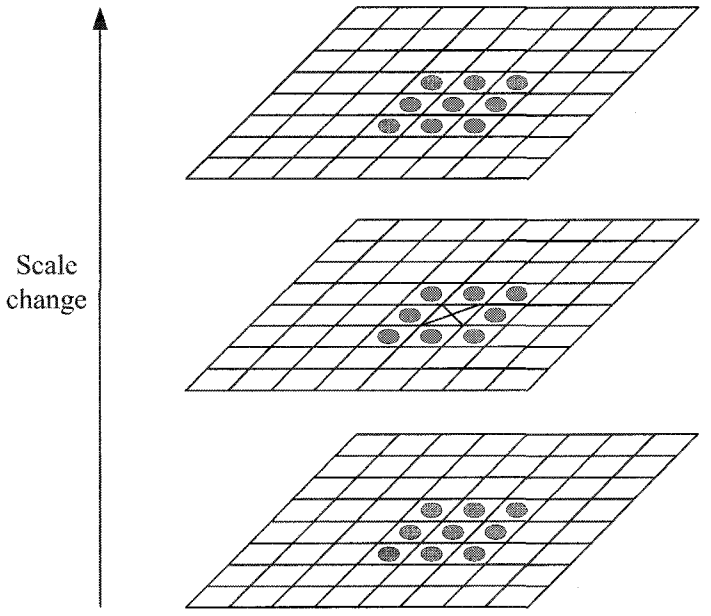


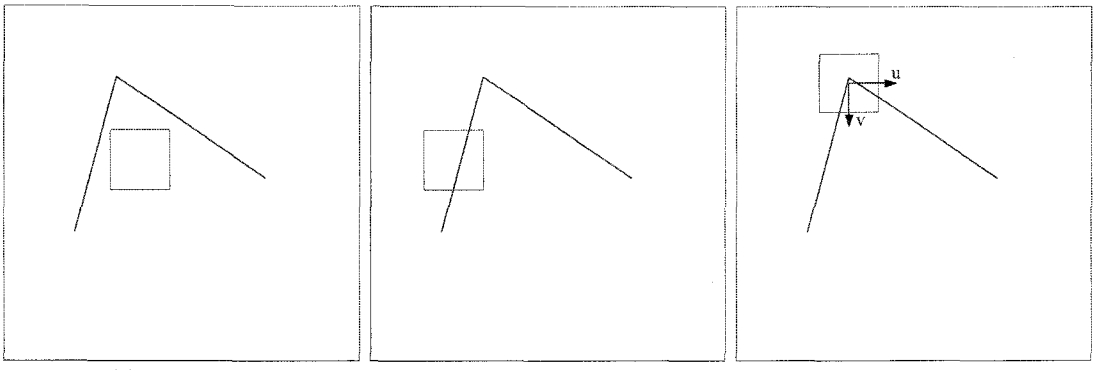
Figure 5.5: Feature detection.

To accurately detect the stable features, we choose to detect the points corresponding to the same scene points consistently over different views. Moreover, there should be enough information in the neighborhood of the target points so that the corresponding points in the other scale images can be automatically matched.

As shown in Fig. 5.5, for each feature point, indicated as $\times$ in the figure, it will be compared with its 8 neighbors in the current blurred image and will be compared with its 9 neighbors in the scale above and below. The feature point which is larger than all its $8 + 2 \times 9$ neighbors or smaller than all of them will be selected.

Moreover, to make these points stable and robust to noise and local distortions, the strong edge responses, such as edges and corners, should be removed.

As shown in Fig. 5.6, the edges are the locations where the intensity variation in a certain direction is high, while in the orthogonal direction is low. Corners are the local



(a) Flat region. (b) Edge region. (c) Corner.

Figure 5.6: Responses in the scale space.

image features characterized by locations where the variations of intensity in both x and y directions are high. The flat regions indicate that there is no intensity change or very small intensity change in all directions.

To calculate the change of intensity for the shift $[u, v]$ as shown in Fig. 5.6 (c), the change of the intensity is defined as:

$$E(u,v) = \sum_{x,y \epsilon W} [I(x+u, y+u) - I(x,y)]^2 \tag{5.60}$$

Based on the Taylor expansion:

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{f''(x)(\Delta x)^2}{2!} + \cdots \tag{5.61}$$

The $I(x+u, y+v) - I(x,y)$ can be approximated as (truncated after the first order derivative):

$$[I(x+u, y+v) - I(x,y)] \approx I(x,y) + [I_x(x,y) I_y(x,y)] \begin{bmatrix} u \\ v \end{bmatrix} \tag{5.62}$$

so that,

$$E(u,v) = \sum_{x,y \epsilon W} \left([I_x(x,y) I_y(x,y)] \begin{bmatrix} u \\ v \end{bmatrix} \right)^2 \tag{5.63}$$

where

$$\begin{aligned} \left([I_x(x,y) I_y(x,y)] \begin{bmatrix} u \\ v \end{bmatrix} \right)^2 &= \left([I_x(x,y) I_y(x,y)] \begin{bmatrix} u \\ v \end{bmatrix} \right)^T \left([I_x(x,y) I_y(x,y)] \begin{bmatrix} u \\ v \end{bmatrix} \right) \\ &= [\ u \quad v\ ] \begin{bmatrix} I_{xx} & I_{yy} \\ I_{xy} & I_{yy} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned} \tag{5.64}$$

so the matrix:

$$M = \begin{bmatrix} I_{xx} & I_{yy} \\ I_{xy} & I_{yy} \end{bmatrix} \tag{5.65}$$

The two eigenvalues of the matrix M can be computed as follows by assuming that the large eigenvalue is α and the small one is β:

$$Tr(M) = I_{xx} + I_{yy} = \alpha + \beta \tag{5.66}$$

$$|M| = I_{xx} Iyy - (I_{xy})^2 = \alpha\beta \tag{5.67}$$

Suppose that $\alpha = \gamma\beta$, so that,

$$\frac{Tr(M)^2}{|M|} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(\gamma+1)^2}{\gamma} \tag{5.68}$$

For those points with strong edge response, α will be much larger than β which the change of intensity is very large along the direction perpendicular to the edge, while

the change of intensity is not that much along the direction of the edge as shown in Fig. 5.6 (b).

Based on this, for a point to be selected as the feature point, its $\frac{Tr(M)^2}{|M|}$ should be smaller than a pre-defined threshold.

5.2.3 Noise visibility function

As mentioned above, the Noise Visibility Function (NVF) is used to control the watermark embedding strength. The noise visibility function characterizes the local features of the image and is one way to do the texture masking in spatial domain [107]. It is straightforward to use noise visibility function to guide the watermark embedding process, if we treat the watermark as noise. To do so, normally the watermark can be approximated as i.i.d Gaussian process with unit variance, $N(0,1)$.

For each subregion after segmentation, the local variance $\sigma_{x(i,j)}$ for every pixel $x(i,j)$ can be used to compute the NVF. β is an empirical factor.

$$NVF(x(i,j)) = \frac{1}{1+\beta\sigma^2_{x(i,j)}} \tag{5.69}$$

For the non-stationary Gaussian model, $\beta = 1$ in the above equation. As for the Generalized Gaussian model, The noise visibility function is:

$$NVF(x(i,j)) = \frac{w(i,j)}{w(i,j)+\sigma_s^2} \tag{5.70}$$

where $w(i,j)$ is a weighting factor and $w(i,j) = \alpha_s(\eta(\alpha_s))^{\alpha_s}\frac{1}{||r(i,j)||^{2-\alpha_s}}$ with $r(i,j) = \frac{x(i,j)-\mu_s}{\sigma_s}$. α_s is the shape parameter, for most real images, $0.3 \leq \alpha_s \leq 2$. The use of $w(i,j)$ can get more accurate result than using the parameter β.

Based on the previous work, the image can be approximated as generalized Gaussian mixture distribution. For the region with the highest variance, the shaping parameter is set to be 1 so that it is approximated as the Laplacian distribution, the shaping parameter for the region with lowest variance will be 0.5, all the other regions will be approximated as Gaussian distribution with shaping parameter to be 2.

It is clear that the noise visibility function is related to the local variance. From the definition of the local variance, it reflects the image texture distribution. If the area in the image is very flat, or if the variation of the pixel value is very small, the local variance of these areas will be close to zero because the individual pixel value is almost equal to the average value of the pixels within these areas. Therefore, the noise visibility function is close to one. On the contrary, if the area in the image with high activity, or if the variation of the pixel value is very high, the local variance approaches a very large number. This causes the noise visibility function to approach zero.

With the noise visibility function, the watermark embedding equation is given in Equ. (5.71). In the equation, x is the original image and α is the pre-defined embedding strength and NVF is the noise visibility function and is used to control the embedding strength. After watermark embedding, we get the watermarked image y.

$$y = x + (1 - NVF) \cdot \alpha \cdot w \tag{5.71}$$

After analyzing the relationship between the local variance and the noise visibility function, it is easy to understand how the noise visibility function adjusts the embedding strength of the watermark. When there are highly textured regions in the image, their noise visibility functions are close to zero, and the watermark is embedded into these regions with the maximum embedding strength. On the other hand, for the flat regions

in the image, such as the sky area, their noise visibility functions approach to one, and as a result of the embedding function, no watermark will be embedded into these regions. In this way, the spatial watermark is adaptive to the image features. However, in the flat region, it still has the capability to hide watermark based on the darkness and brightness. This is used to optimize the proposed watermarking scheme.

5.3 The *pdf* of the watermarked region

As addressed in the previous sections, the cover image is segmented into a number of homogeneous regions. And each segmented region is described using the generalized Gaussian with a specific shape parameter α.

The watermark is approximated as the i.i.d. white Gaussian noise, which can be expressed as the following equation:

$$p_G(x) = \frac{1}{\sqrt{2\pi}\sigma_g} \cdot exp(-(\frac{x-\mu_g}{\sqrt{2}\sigma_g})^2) \tag{5.72}$$

Therefore, after embedding the watermark, w, into one circular region of the cover image, x, without considering additive noise at the current moment, the watermarked region can be described as the follows:

$$y = x + \rho \cdot w \tag{5.73}$$

where, ρ is the watermark embedding strength. Assume the mean and standard deviation of the watermark are μ_w and σ_w, the mean and standard deviation of $\rho \cdot w$ becomes

$\rho\mu_w$ and $\rho\sigma_w$, respectively. In this case, the distribution of ρw becomes:

$$p_G(\rho w) = \frac{1}{\sqrt{2\pi}\rho\sigma_w} \cdot exp(-(\frac{x-\rho\mu_w}{\sqrt{2}\rho\sigma_w})^2) \tag{5.74}$$

For a further discussion, the *pdf* of the watermarked region, y, can be achieved by convolving the *pdf* of the segmented region of the cover image and the *pdf* of the embedded watermark. Then, we have:

$$\begin{aligned} p_y(y) &= p_{GG}(x) \otimes p_G(\rho w) && (5.75) \\ &= \int_{-\infty}^{\infty} p_{GG}(\tau) \cdot p_G(y-\tau) d\tau \\ &= \int_{-\infty}^{\infty} \frac{1}{2ab\sigma_{gg}} exp(-|\frac{\tau-\mu_{gg}}{\sigma_{gg}b}|^{\alpha}) \cdot \frac{1}{\sqrt{2\pi}\rho\sigma_w} exp(-\frac{(y-\tau-\rho\mu_w)^2}{2\rho^2\sigma_w^2}) d\tau && (5.76) \end{aligned}$$

Setting $z = \tau - \mu_{gg}$, we have $\tau = z + \mu_{gg}$ and $d\tau = dz$. The above equation becomes:

$$p_y(y) = \int_{-\infty}^{\infty} \frac{1}{2ab\sigma_{gg}} \cdot \frac{1}{\sqrt{2\pi}\rho\sigma_w} \cdot exp(-|\frac{z}{\sigma_{gg}b}|^{\alpha}) \cdot exp(-\frac{(y-z-\mu_{gg}-\rho\mu_w)^2}{2\rho^2\sigma_w^2}) dz \tag{5.77}$$

Set $s = y - \mu_{gg} - \rho\mu_w$:

$$\begin{aligned} p_y(y) &= \int_{-\infty}^{\infty} \frac{1}{2\sqrt{2\pi}ab\rho\sigma_{gg}\sigma_w} \cdot exp(-|\frac{z}{\sigma_{gg}b}|^{\alpha}) \cdot exp(-\frac{(s-z)^2}{2\rho^2\sigma_w^2}) dz \\ &= \int_{-\infty}^{\infty} \frac{1}{2\sqrt{2\pi}ab\rho\sigma_{gg}\sigma_w} \cdot exp(-|\frac{z}{\sigma_{gg}b}|^{\alpha}) \cdot exp(-\frac{(z-s)^2}{2\rho^2\sigma_w^2}) dz \\ &= \int_{-\infty}^{\infty} \frac{1}{2\sqrt{2\pi}ab\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot exp(-\frac{|z|^{\alpha}}{|\sigma_{gg}b|^{\alpha}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2}) dz \end{aligned} \tag{5.78}$$

In Equ. (5.78), the component $|z|^{\alpha}$ has a variable power α ranging from 0.3 to 2, which results that the closed form of the equation is unachievable. To illustrate the

possible forms of the *pdf* of y, we derive Equ. (5.78) in two most frequently used cases: $\alpha = 1$ and $\alpha = 2$.

As derived in Section 5.2, when $\alpha = 2$, Equ. (5.7) becomes Gaussian distribution, which indicates the circular region is Gaussian distributed. Recall the integral $\int_{-\infty}^{\infty} e^{-ku^2} du = \sqrt{\pi/k}, k > 0$, the *pdf* of y will be:

$$\begin{aligned}
p_y(y) &= \frac{1}{2\pi\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot \int_{-\infty}^{\infty} exp(-(\frac{z^2}{2\rho^2\sigma_w^2} - \frac{2sz}{2\rho^2\sigma_w^2} + \frac{z^2}{2\sigma_{gg}^2}))dz \\
&= \frac{1}{2\pi\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot \int_{-\infty}^{\infty} exp[-((\frac{1}{2\rho^2\sigma_w^2} + \frac{1}{2\sigma_{gg}^2})z^2 - 2(\frac{s}{2\rho^2\sigma_w^2})z)]dz \\
&= \frac{1}{2\pi\rho\sigma_{gg}\sigma_w} \cdot e^{ms} \cdot \int_{-\infty}^{\infty} e^{-(nz^2+2mz)} dz \\
&= \frac{1}{2\pi\rho\sigma_{gg}\sigma_w} \cdot e^{ms} \cdot \int_{-\infty}^{\infty} e^{-n(z+m/n)^2} \cdot e^{m^2/n} dz \\
&= \frac{1}{2\pi\rho\sigma_{gg}\sigma_w} \cdot e^{ms+m^2/n} \cdot \int_{-\infty}^{\infty} e^{-n(z+m/n)^2} d(z+m/n) \\
&= \frac{1}{2\pi\rho\sigma_{gg}\sigma_w} \cdot e^{ms+m^2/n} \cdot \sqrt{\pi/n} \\
&= \frac{\sqrt{\pi/n}}{2\pi\rho\sigma_{gg}\sigma_w} \cdot e^{ms+m^2/n}
\end{aligned} \tag{5.79}$$

where $m = -\frac{s}{2\rho^2\sigma_w^2}$ and $n = \frac{1}{2\rho^2\sigma_w^2} + \frac{1}{2\sigma_{gg}^2}$

$$\begin{aligned}
ms + \frac{m^2}{n} &= -\frac{s}{2\rho^2\sigma_w^2} + \frac{s}{4\rho^4\sigma_w^4} \cdot \frac{4\rho^2\sigma_w^2\sigma_{gg}^2}{2\rho^2\sigma_w^2 + 2\sigma_{gg}^2} \\
&= -\frac{s^2}{2\rho^2\sigma_w^2 + 2\sigma_{gg}^2}
\end{aligned}$$

$$\begin{aligned}\frac{\sqrt{\pi/n}}{2\pi\rho\sigma_{gg}\sigma_w} &= \sqrt{\frac{\pi 4\rho^2\sigma_w^2\sigma_{gg}^2}{2\rho^2\sigma_w^2+2\sigma_{gg}^2}}\cdot\frac{1}{2\pi\rho\sigma_{gg}\sigma_w} \\ &= \frac{1}{\sqrt{2\pi(\rho^2\sigma_w^2+\sigma_{gg}^2)}}\end{aligned}$$

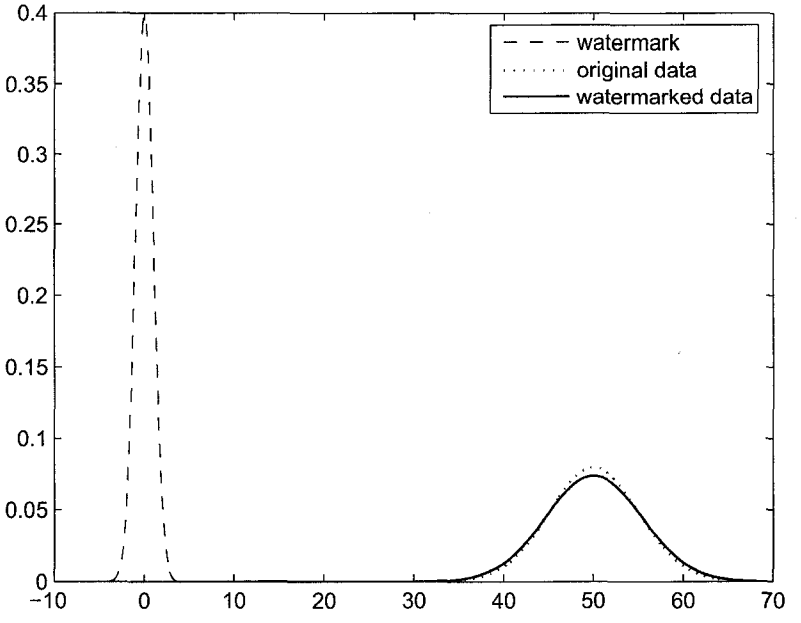


Figure 5.7: The illustration of distributions of the watermark and the image data before and after watermark embedding. The watermark is Gaussian distributed with $\mu_g = 0$, $\sigma_g = 1$. The original image data is Gaussian distributed with $\mu_{gg} = 50$, $\sigma_{gg} = 5$.

Therefore, the *pdf* of the watermarked region becomes:

$$p_y(y) = \frac{1}{\sqrt{2\pi(\rho^2\sigma_w^2+\sigma_{gg}^2)}}\cdot exp\{-\frac{(y-(\rho\mu_w+\mu_{gg}))^2}{2(\rho^2\sigma_w^2+\sigma_{gg}^2)}\} \tag{5.80}$$

When $\alpha = 1$, Equ. (5.7) becomes a Laplace distribution. Based on Equ. (5.78), we

have:

$$\begin{aligned} p_y(y) &= \frac{1}{2\sqrt{\pi}\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot \int_{-\infty}^{\infty} exp(-\frac{\sqrt{2}|z|}{\sigma_{gg}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2})dz \\ &= \frac{1}{2\sqrt{\pi}\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot [\int_{0}^{\infty} exp(-\frac{\sqrt{2}|z|}{\sigma_{gg}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2})dz \\ &\quad + \int_{-\infty}^{0} exp(-\frac{\sqrt{2}|z|}{\sigma_{gg}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2})dz] \end{aligned} \tag{5.81}$$

Set $x = -z$, then $dz = -dx$. When $z \in (-\infty, 0]$, $-x \in (\infty, 0]$, $dz = -dx$. So,

$$\begin{aligned} &\int_{-\infty}^{0} exp(-\frac{\sqrt{2}|z|}{\sigma_{gg}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2})dz \\ =& \int_{-\infty}^{0} exp(-\frac{\sqrt{2}(-z)}{\sigma_{gg}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2})dz \\ =& -\int_{\infty}^{0} exp(-\frac{\sqrt{2}x}{\sigma_{gg}}) \cdot exp(-\frac{(x^2+2sx)}{2\rho^2\sigma_w^2})dx \\ =& \int_{0}^{\infty} exp(-\frac{\sqrt{2}x}{\sigma_{gg}}) \cdot exp(-\frac{(x^2+2sx)}{2\rho^2\sigma_w^2})dx \end{aligned} \tag{5.82}$$

Substituting Equ. (5.82) in Equ. (5.81), we have:

$$\begin{aligned} p_y(y) &= \frac{1}{2\sqrt{\pi}\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot [\int_{0}^{\infty} exp(-\frac{\sqrt{2}z}{\sigma_{gg}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2})dz \\ &\quad + \int_{0}^{\infty} exp(-\frac{\sqrt{2}z}{\sigma_{gg}}) \cdot exp(-\frac{(z^2+2sz)}{2\rho^2\sigma_w^2})dz] \end{aligned} \tag{5.83}$$

$$\begin{aligned}
& \frac{1}{2\sqrt{\pi}\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot \int_0^\infty exp(-\frac{\sqrt{2}z}{\sigma_{gg}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2})dz \\
= & \frac{1}{2\sqrt{\pi}\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot \int_0^\infty exp[-(\frac{1}{2\rho^2\sigma_w^2}z^2 + 2(\frac{1}{\sqrt{2}\sigma_{gg}} - \frac{s}{2\rho^2\sigma_w^2})z)]dz \\
= & \frac{1}{2\sqrt{\pi}\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot exp(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w})^2 \\
& \cdot \int_0^\infty exp(-(\frac{z}{\sqrt{2}\rho\sigma_w} + (\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w}))^2)dz
\end{aligned} \tag{5.84}$$

Set $t = \frac{z}{\sqrt{2}\rho\sigma_w}$, Since $z \in [0,\infty)$, $t \in [0,\infty)$. $dt = \frac{1}{\sqrt{2}\rho\sigma_w}dz$

$$\begin{aligned}
& \int_0^\infty exp(-(\frac{z}{\sqrt{2}\rho\sigma_w} + (\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w}))^2)dz \\
= & \sqrt{2}\rho\sigma_w \int_0^\infty exp(-(t + (\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w}))^2)dt
\end{aligned} \tag{5.85}$$

Set $t_1 = t + (\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w})$. $dt_1 = dt$. $t \in [0,\infty)$, $t_1 \in [\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w}, \infty)$. Thus, Equ. (5.85) becomes:

$$\sqrt{2}\rho\sigma_w \int_{(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w})}^\infty exp(-t_1^2)dt_1 = \sqrt{2}\rho\sigma_w \cdot \frac{\sqrt{\pi}}{2} \cdot erfc(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w}) \tag{5.86}$$

Substituting Equ. (5.86) into Equ. (5.84), we have:

$$\begin{aligned}
& \frac{1}{2\sqrt{\pi}\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot \int_0^\infty exp(-\frac{\sqrt{2}z}{\sigma_{gg}}) \cdot exp(-\frac{(z^2-2sz)}{2\rho^2\sigma_w^2})dz \\
= & \frac{1}{2\sqrt{2}\sigma_{gg}} exp(-\frac{s^2}{2\rho^2\sigma_w^2}) exp((\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w})^2) erfc(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w})
\end{aligned} \tag{5.87}$$

where the complementary error function is defined as:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^0 e^{-t^2} dt \tag{5.88}$$

Similarly,

$$\begin{aligned} & \frac{1}{2\sqrt{\pi}\rho\sigma_{gg}\sigma_w} \cdot exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \cdot \int_0^\infty exp(-\frac{\sqrt{2}z}{\sigma_{gg}}) \cdot exp(-\frac{(z^2+2sz)}{2\rho^2\sigma_w^2}) dz \\ = & \frac{1}{2\sqrt{2}\sigma_{gg}} exp(-\frac{s^2}{2\rho^2\sigma_w^2}) exp((\frac{\rho\sigma_w}{\sigma_{gg}} + \frac{s}{\sqrt{2}\rho\sigma_w})^2) erfc(\frac{\rho\sigma_w}{\sigma_{gg}} + \frac{s}{\sqrt{2}\rho\sigma_w}) \end{aligned} \tag{5.89}$$

Therefore, the *pdf* of the watermarked region can be written as:

$$\begin{aligned} p_y(y) = & \frac{1}{2\sqrt{2}\sigma_{gg}} exp(-\frac{s^2}{2\rho^2\sigma_w^2}) \{ exp[(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w})^2] \cdot erfc(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{s}{\sqrt{2}\rho\sigma_w}) \\ & + exp[(\frac{\rho\sigma_w}{\sigma_{gg}} + \frac{s}{\sqrt{2}\rho\sigma_w})^2] \cdot erfc(\frac{\rho\sigma_w}{\sigma_{gg}} + \frac{s}{\sqrt{2}\rho\sigma_w}) \} \\ = & \frac{1}{2\sqrt{2}\sigma_{gg}} exp(-\frac{(y-\mu_{gg}-\rho\mu_w)^2}{2\rho^2\sigma_w^2}) \cdot \\ & \{ exp[(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{y-\mu_{gg}-\rho\mu_w}{\sqrt{2}\rho\sigma_w})^2] \cdot erfc(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{y-\mu_{gg}-\rho\mu_w}{\sqrt{2}\rho\sigma_w}) \\ & + exp[(\frac{\rho\sigma_w}{\sigma_{gg}} + \frac{y-\mu_{gg}-\rho\mu_w}{\sqrt{2}\rho\sigma_w})^2] \cdot erfc(\frac{\rho\sigma_w}{\sigma_{gg}} + \frac{y-\mu_{gg}-\rho\mu_w}{\sqrt{2}\rho\sigma_w}) \} \\ = & \frac{1}{2\sqrt{2}\sigma_{gg}} exp(-\frac{(y-\mu_{gg}-\rho\mu_w)^2}{2\rho^2\sigma_w^2}) \cdot \\ & \{ erfcx(\frac{\rho\sigma_w}{\sigma_{gg}} - \frac{y-\mu_{gg}-\rho\mu_w}{\sqrt{2}\rho\sigma_w}) + erfcx(\frac{\rho\sigma_w}{\sigma_{gg}} + \frac{y-\mu_{gg}-\rho\mu_w}{\sqrt{2}\rho\sigma_w}) \} \end{aligned} \tag{5.90}$$

where, $erfcx(x)$ is defined as:

$$erfcx(x) = e^{x^2} \cdot erfc(x) \tag{5.91}$$

and is the scaled complementary error function.

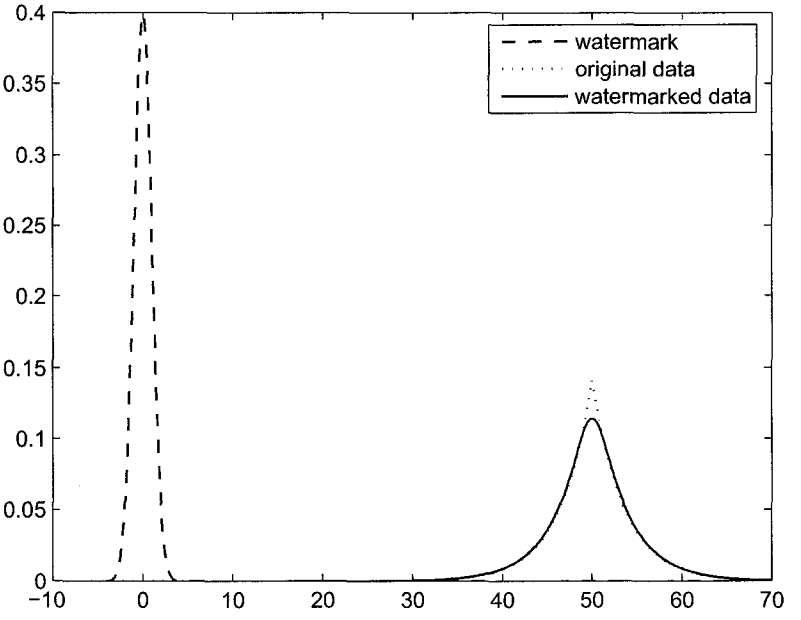


Figure 5.8: The illustration of distributions of the watermark and the image data before and after watermark embedding. The watermark is Gaussian distributed with $\mu_g = 0$, $\sigma_g = 1$. The original image data is Laplace distributed with $\mu_{gg} = 50$, $\sigma_{gg} = 5$.

5.4 The error probability

As defined in [1], a false positive error occurs when a watermark is detected present in an unwatermarked image. And the false positive probability, P_{fpp}, is the likelihood of such an error occurance.

On the contrary, a false negative error occurs when a watermark is detected absent in a watermarked image. And the false negative probability, P_{fnp}, is the chance that such an error occurs.

To calculate P_{fpp} and P_{fnp}, normally two hypothesis will be made, i.e.:

H_0: watermark is absent. H_1: watermark is present.

Associated with the two hypothesis, the measurement space $\mathfrak{K}$ will be used to determine the false positive probability and false negative probability. For this theoretical analysis, in $\mathfrak{K}$, each element, k, is calculated between the watermarked image and other available information such as the original watermark or the original image. With a hypothesis testing rule, the measurement space is partitioned into two disjoint sub-spaces $\mathfrak{K}_0$ and $\mathfrak{K}_1$, where $\mathfrak{K}_0 = \mathfrak{K}_1{}^C$. If $k \in \mathfrak{K}_0$, H_0 is true. Otherwise, H_1 is true.

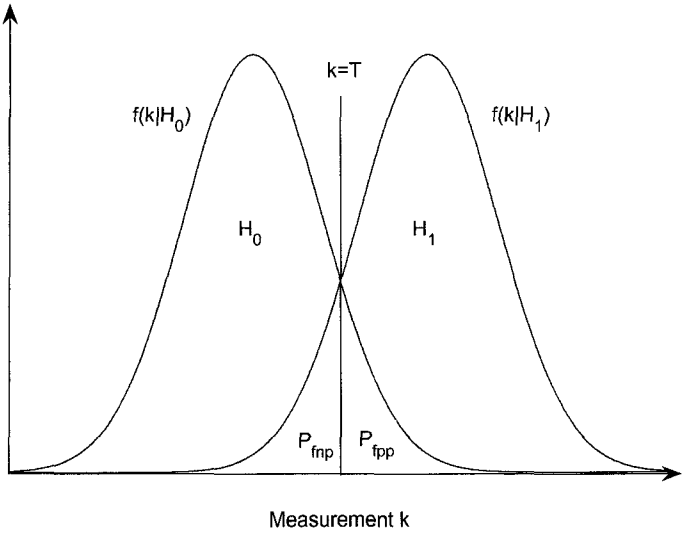


Figure 5.9: False positive probability and false negative probability.

As illustrated in Fig. 5.9, the false positive probability is calculate as:

$$P_{fpp} = \int_{T}^{\infty} f(k|H_0)dk \tag{5.92}$$

where, $f(k|H_0)$ is the *pdf* of measurement k in $\mathfrak{K}_0$. T is the threshold to determine which hypothesis is true.

Similarly, the false negative probability can be calculated using the following equation:

$$P_{fnp} = \int_{-\infty}^{T} f(k|H_1)dk \tag{5.93}$$

where, $f(k|H_1)$ is the *pdf* of measurement k in $\mathfrak{K}_1$.

In this thesis, we embed one watermark bit in one well selected circular region centered at the extracted feature point. By denoting the circular region as S, the watermark embedding equation for a single region can be rewritten as:

$$Y^S = X^S + \rho \cdot w \tag{5.94}$$

where ρ is the embedding strength. X^S means the original data in S, which have a *pdf* of $f_{X^S}(x)$. As mentioned in Chapter 5, $f_{X^S}(x)$ is the i.i.d. Generalized Gaussian. w is the watermark which has a Gaussian distribution. Y^S means the watermarked pixels in S, which have a *pdf* of $f_{Y^S}(y)$. And the *pdf* of $f_{Y^S}(y)$ has been developed in two special cases in Section.

Thus, for a single circular region, the two hypothesis declared above can be rewritten as:

H_0^S: $Y^S \sim f_{X^S}(y)$.

H_1^S: $Y^S \sim f_{Y^S}(y)$.

In the above declaration, hypothesis H_0^S indicates that Y^S has a *pdf* the same as the original data in S, X^S, which means that the watermark is absent. On the contrary, H_1^S indicates that the watermark is present.

Recall the Neyman-Pearson theorem [108], the hypothesis testing rule for a circular region can be formed as:

$$\Re_0{}^S(\tau) = \Re_1{}^S(\tau)^C = \{y : \ln \frac{f_{X^S}(y)}{f_{Y^S}(y)} \geq \tau\} \tag{5.95}$$

where, $\Lambda_S = \ln \frac{f_{X^S}(y)}{f_{Y^S}(y)}$ is also known as the log-likelihood ratio.

Therefore, the false positive probability and false negative probability for a single circular region can be rewritten, respectively, as:

$$P_{fpp}^S = \int_{\tau}^{\infty} f(\Lambda_S | H_0^S) d\Lambda_S \tag{5.96}$$

$$P_{fnp}^S = \int_{\infty}^{\tau} f(\Lambda_S | H_1^S) d\Lambda_S \tag{5.97}$$

where $f(\Lambda_S|H_0^S)$ and $f(\Lambda_S|H_1^S)$ are the *pdf* of Λ_S under hypothesis H_0^S and H_1^S, respectively.

As mentioned in [108], we have a bound:

$$P_{fnp}^S \log_b \frac{P_{fnp}^S}{1 - P_{fpp}^S} + (1 - P_{fnp}^S) \log_b \frac{1 - P_{fnp}^S}{P_{fpp}^S} \leq D(f_{Y^S}(y) \parallel f_{X^S}(y)) \tag{5.98}$$

where $D(f_{Y^S}(y) \parallel f_{X^S}(y))$ is known as the Kullback-Leibler divergence and is defined as:

$$D(f_{Y^S}(y) \parallel f_{X^S}(y)) = \int_{-\infty}^{\infty} f_{Y^S}(y) \cdot \log_b \frac{f_{Y^S}(y)}{f_{X^S}(y)} dy \tag{5.99}$$

Similarly,

$$D(f_{X^S}(y) \parallel f_{Y^S}(y)) = \int_{-\infty}^{\infty} f_{X^S}(y) \cdot \log_b \frac{f_{X^S}(y)}{f_{Y^S}(y)} dy \tag{5.100}$$

The Kullback-Leibler divergence is also known as the discrimination function or the relative entropy. $D(f_{Y^S}(y) \parallel f_{X^S}(y))$ and $D(f_{X^S}(y) \parallel f_{Y^S}(y))$ are used to evaluate the difficulty to discriminate between the hypothesis H_0^S and H_1^S. It is found out that the relative entropy, $D(f_{Y^S}(y) \parallel f_{X^S}(y))$, approximately linearly increases with the number of circular embedding regions [109]. The larger the relative entropy, the smaller the error probability of watermark detection.

In the above equation, we use b to denote the logarithm base in the relative entropy, because b can be arbitrary. As mentioned in [108], it will be easier to use natural logs to develop theories. While, it will be convenient to use base-2 logs for numerical examples which will make the relative entropy in unit of bits.

Equ. (5.98) can be further developed as:

$$P_{fnp}^S[\log_2 P_{fnp}^S - \log_2 1 - P_{fpp}^S] + (1 - P_{fnp}^S)[\log_2 1 - P_{fnp}^S - \log_2 P_{fpp}^S] \leq D(f_{Y^S}(y) \parallel f_{X^S}(y)) \tag{5.101}$$

$$D(f_{Y^S}(y) \parallel f_{X^S}(y)) - P_{fnp}^S \log_2 P_{fnp}^S - (1 - P_{fnp}^S) \log_2 1 - P_{fnp}^S + P_{fnp}^S \log_2 1 - P_{fpp}^S \geq -(1 - P_{fnp}^S) \log_2 P_{fpp}^S \tag{5.102}$$

Then we have:

$$-(1 - P_{fnp}^S) \log_2 P_{fpp}^S \leq D(f_{Y^S}(y) \parallel f_{X^S}(y)) + 1 \tag{5.103}$$

Therefore,

$$P_{fpp}^{S} \geq 2^{-D(f_{Y^S}(y)\|f_{X^S}(y))/(1-P_{fnp}^{S})} \tag{5.104}$$

The total probability error of a single circular region can be calculated as:

$$P_{e}^{S} = \pi_0 P_{ffp}^{S} + \pi_1 P_{fnp}^{S} \tag{5.105}$$

where, π_0 and π_1 are the *a priori* probabilities of the two hypothesis H_0^S and H_1^S. These two probabilities are defined as $\pi_0 = \frac{1}{2}$ and $\pi_1 = \frac{1}{2}$ in this thesis.

Therefore, P_e^S can be evaluated based on J-divergence [108]:

$$P_{e}^{S} > \pi_0 \pi_1 e^{-J/2} \tag{5.106}$$

where, the J-divergence, $J = D(f_{Y^S}(y) \parallel f_{X^S}(y)) + D(f_{X^S}(y) \parallel f_{Y^S}(y))$, is symmetric and nonnegative.

Chapter 6

Implementation of the proposed RST invariant watermarking scheme

6.1 Image modelling

The accurate image modelling is critical for the analysis of the watermarking processes. As mentioned in Section 1.8, the watermarked image can be expressed by the following equation:

$$Y = X + \alpha \times W \tag{6.1}$$

where, Y is the watermarked image, X is the original image, W is the watermark and α is the embedding strength. As addressed in Section 5.1, the image X is normally modelled as the mixture Gaussian distribution, because it is very difficult to establish

an accurate and general mathematical model for a natural image with content diversity.

Another approach is to model the image in the transform domain such as DWT and DCT domain, In [110], the Gaussian and Laplacian model is used to model the DWT coefficient of the image. In [111], the generalized Gaussian model is used to model DCT coefficients. Also the wavelet shrinkage is to use the Gaussian model to approximate the wavelet coefficients for denoising and image restoration.

So as mentioned in [111], for different frequency components of the image, the generalized Gaussian distribution can better approximate those components with different shape parameters. The low frequency components can be better approximated when the shape parameter is set around 0.5. And for middle and high frequency components, the shaping parameter should between 1 and 2.

In the proposed watermarking algorithm in this thesis, the image is to be modelled as the mixture Generalized Gaussian distribution in the spatial domain. In the following section, the Generalized Gaussian distribution modelling of the image can be established after image segmentation.

6.2 MAP image segmentation

The image segmentation algorithm mentioned in [112] is used. Assume the observed image is y and the segmentation is x, for each subregion of the segmentation, the conditional distribution of y_s given x_s is a Gaussian distribution with mean μ_{y_s} and variance σ_{y_s}.

$$p_{y_s|x_s}(y_s|x_s) \sim N(\mu_{y_s}, \sigma_{y_s}) \tag{6.2}$$

here the Gaussian mixture distribution $p_{y|x}(y|x)$ is used to model the observed image y. The density is:

$$p_{y|x}(y|x) = \sum_{s \epsilon S} m_s p_{y_s|x_s}(y_s|x_s) \tag{6.3}$$

m_s is the mixture weighting factor and $\sum_{s \epsilon S} m_s = 1$.

To get the parameters of the Gaussian mixture distribution, the EM (Expectation-Maximization) algorithm is used: first the mean vector μ_{y_s} and covariance σ_{y_s} for each Gaussian distribution is set to the initial value. The covariance can be set to be identity matrix and mean is calculated by finding the mean of the different regions of the image. For example, the image can be divided evenly into a certain number of sub regions and the mean values can be calculated from these subregions as the initial values. Normally the image will be segmented into 5 to 8 regions which is enough for watermark embedding. Then the probability of the pixel y_j falling into one of the Gaussian distribution s can be calculated ($p_{y_s|x_s}(y_s|x_s)$ is simplified as $p_s(y)$):

$$P(s|y_j) = \frac{m_s p_s(y_j)}{\sum_{s \epsilon S} m_s p_s(y_j)} \tag{6.4}$$

Based on EM iteration, the parameters can be updated as Equ. (5.50), (5.51) and (5.52).

The EM iteration will continue until $log \prod_{k=1}^{N} p(y_k)$ is increased within 1% for one iteration.

$$\log p_{y|x}(y|x) = \sum_{s \epsilon S} \log m_s p_{y_s|x_s}(y_s|x_s) \tag{6.5}$$

The $p_x(x)$ is the prior information of the segmentation and is approximated as Gibbs distribution.

$$p_x(x) = \frac{1}{z} \exp\{-\beta \sum_{\{i,j\} \epsilon C} b_{i,j} \delta(x_i \neq x_j)\} \tag{6.6}$$

Based one Maximum A posterior probability, the segmentation $\hat{x}$ can be estimated using Equ. (6.7). Equ. (6.8) is the cost function which the MAP estimator will minimize during the estimation of $\hat{x}$.

$$\hat{x} = \arg \max_{x \epsilon \{0,1\}^N} \{\log p_{y|x}(y|x) + \log p(x)\} \tag{6.7}$$

$$c(x) = \sum_{s \epsilon S} -\log m_s p_{y_s|x_s}(y_s|x_s) + \beta \sum_{\{i,j\} \epsilon C} b_{i,j} \delta(x_i \neq x_j) \tag{6.8}$$

To solve the above equation, the iterative conditional modes (ICM) can be used to get the segmentation result. Also mean μ_{x_s} and variance σ_{x_s} can be calculated to adjust the watermark embedding strength. The image segmentation is shown in Fig. 6.1 and Fig. 6.2.

Figure 6.1: The original image.

The Fig. 6.2 shows the image segmentation contains 5 classes of regions, each

Figure 6.2: The segmentation region.

region corresponds to one generalized Gaussian distribution, whose mean and variance are estimated using the EM algorithm. Now the image can be approximated as the mixture Generalized Gaussian distribution.

6.3 Feature points detection

As addressed in Section 5.2.2, Gaussian scale model uses the Difference of Gaussian to approximate the filtering effect of the Laplacian of Gaussian second order derivative. Once the segmentation information is retrieved, Gaussian scale model [34] will be used to locate the geometrical-transform-invariant feature points which will be used as the reference for watermark embedding and extraction. The features with strong edge responses will be removed due to their sensitivity to noise.

Suppose $f(x, y)$ is the image, and $g(x, y)$ is the blur image filtered by the Gaussian filter defined in Equ. (6.9).

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-\frac{x^2 + y^2}{2\sigma^2}] \tag{6.9}$$

$$g_1(x, y) = G_\sigma * f(x, y) \tag{6.10}$$

$$g_2(x, y) = G_{k\sigma} * f(x, y) \tag{6.11}$$

Basically, $g_1(x, y)$ and $g_2(x, y)$ are different from each other in scale by a factor of k. Then the DoG filtered image can be computed as follows:

$$DoG = g_1(x, y) - g_2(x, y) = G_{\sigma_1} * f(x, y) - G_{\sigma_2} * f(x, y) \tag{6.12}$$

As shown in Fig. 6.3, the Gaussian scale model works in the following steps:

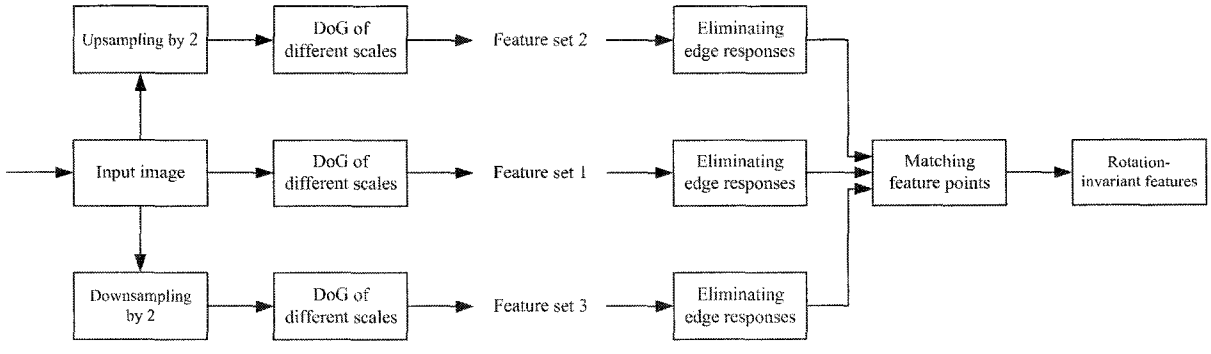


Figure 6.3: The Gaussian scale model.

1. Transform the input image to the scale space with a group of 5 different scales. Every two scales are separated by a constant factor, $k = 2^{1/(5-3)} = 1/4$. Therefore, we will achieve 5 blurred images. The input image can be the original image or the watermarked image.

2. Calculate *DoG*s of the current group of blurred images by subtracting every two adjacent blurred images and we will get 4 *DoG*s.

3. Locate local extrema in every DoG in the group using the feature detection method addressed in Section 5.2.2.

4. Eliminate the feature points with strong edge responses.

5. The feature points that are more robust to distortions of the current group of blurred images are obtained.

6. Upsample the input image and repeat the procedure from step 1 to 5.

7. Downsample the input image and repeat the procedure from step 1 to 5.

8. Select the scale invariant feature points by matching the feature points obtained in steps 5, 6 and 7.

9. Finally, we achieve the desired rotation invariant feature points.

The DoGs calculation steps 1, 2, 6, 7 are summarized in Fig. 6.4. The original image, *Lena*, which is 512×512 in size, is used in the calculation.

The four images in the second row are the *DoG*s of the original image with scales changing from low to high. The four images in the first row are the *DoG*s of the upsampled image. The images in the third and fourth row are the *DoG*s calculated on the image downsampled by 2 and 4, respectively, from the original image.

By locating the local extreme of DoGs of different scales and by removing the edges, the features are selected as shown in Fig. 6.5.

With the result of image segmentation, for each sub-region in the image, we select one feature point as the reference point. There are several rules to guide the selection:

Figure 6.4: The difference of Gaussian filtered images calculated based on the original image, upsampled image and downsampled image.

Figure 6.5: The features of the original image.

Figure 6.6: The SIFT feature used to guide watermark embedding.

1. Some pre-possessing such as Gaussian filtering is applied to the image to make sure the feature points are very robust.

2. In the implementation, the feature point extraction is actually done on the segmented image as shown in Fig. 6.2, because it is low-pass filtered and all the high frequency components have been removed.

The extracted feature points are shown in Fig. 6.6. Comparing Fig. 6.6 with Fig. 6.5, it shows that much fewer feature points can be extracted on Fig. 6.6, which are more robust to geometrical transform and noise.

As shown in Fig. 6.7, the circle regions centered at the feature points are used for watermark embedding. The rule to select the the feature points is: for the detected feature points, the circle regions with the same radius centered at the feature points are

defined. The regions with most of the pixels fitted into one Gaussian distribution are selected. This criteria is called homogeneity. If within the distance of radius, multiple regions have homogeneity difference within 10%, the centroid of those feature points are used as the composite feature point. As shown in Fig. 6.7, 5 circular regions is selected as the watermark embedding regions.

Figure 6.7: The embedding region selection.

6.4 Orientation assignment and region alignment

Once the reference feature points are located, the orientation will be assigned to each point. For the orientation assignment, we use the method proposed in [34]. First, a window centered at those feature points is defined. The gradients of all the pixels in this windows are computed using the first order derivative. Then the histogram of the

gradient are calculated and the peak of the histogram is assigned as the orientation of the feature point. In this way, the orientation would not be affected by noise, small local distortion or some displacement of the feature point position.

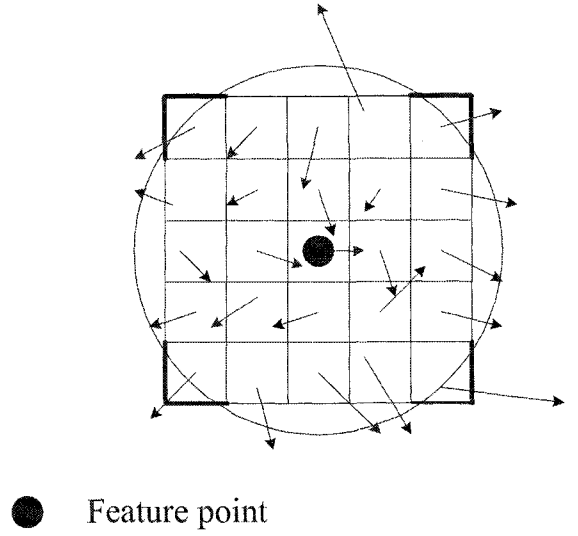


Figure 6.8: The orientation computation.

The gradient of pixel (x_0, y_0) in the image I is computed as following:

$$\bigtriangledown I(x_0, y_0) = [\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}]|_{(x_0,y_0)} \tag{6.13}$$

The magnitude of this gradient is given by $\sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}$ and its orientation is given by by $tan^{-1}(\frac{\partial I}{\partial y}/\frac{\partial I}{\partial x})$.

To generate the histogram, two weighting factors are taken into consideration: the magnitude of the gradient and a Gaussian 2D filter centered at the feature points. In this way, the gradient with larger magnitude contributes more to the histogram and the point closer to the center feature point contributes more. The example of the orientation assignment is shown in Fig. 6.8 and Fig. 6.9.

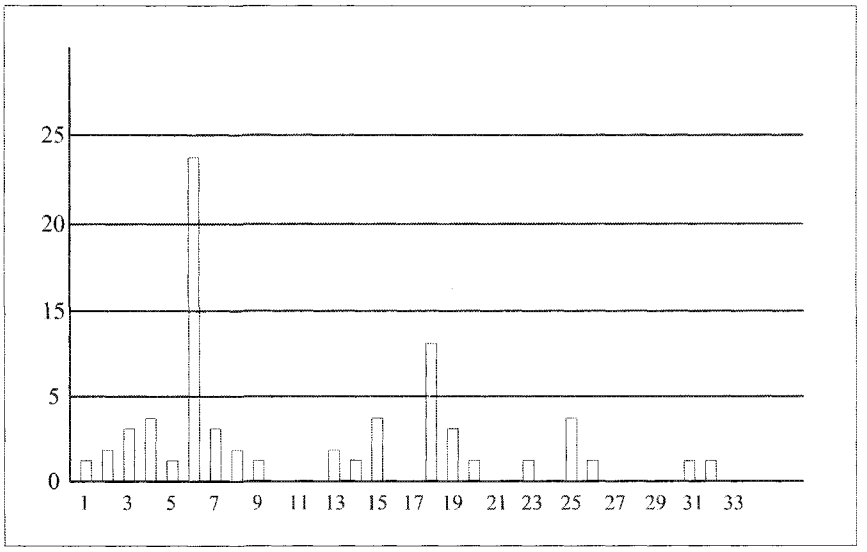


Figure 6.9: The histogram of orientation calculation.

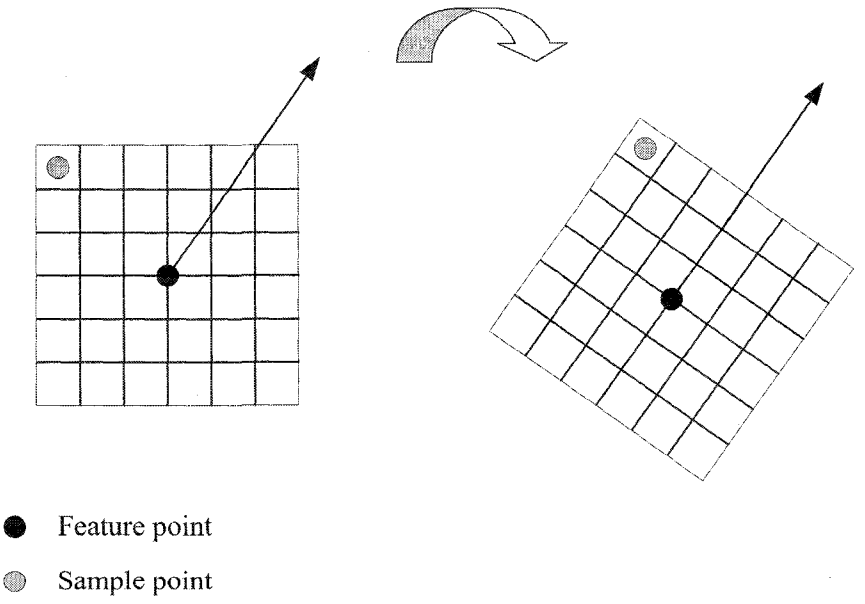


Figure 6.10: The local feature description.

As shown in Fig. 6.10, the orientation of the feature is calculated. The watermark embedding/extraction regions will align with this orientation.

6.5 Image normalization

The circular region centered at those feature point are used for the watermark embedding and extraction. First, those regions are rotated to align with the orientation of the feature points.

Scaling normalization transforms the image into its standard form by translating the origin of the image to its centroid $(\overline{x}, \overline{y})$ [113]. Changing the coordinates into $(\widehat{x}, \widehat{y})$ [28]

$$\widehat{x} = \frac{x - \overline{x}}{a}, \ \widehat{y} = \frac{y - \overline{y}}{b} \tag{6.14}$$

with

$$a = \sqrt{\frac{\beta\gamma}{m_{0,0}}}, \ b = \sqrt{\frac{\beta}{\gamma m_{0,0}}} \tag{6.15}$$

In Equ. (6.14) and Equ. (6.15), 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 $\gamma = l_y/l_x$.

Using the scaling normalization, we can transform those aligned disk regions to its compact size. In the end, those disk regions are scaling invariant and ready for watermark embedding.

Based on the above analysis, the rotation and scaling invariant regions can be located

in the image for watermark embedding. Since each region is homogeneous area and its mean vector and covariance matrix has been calculated during image segmentation, these information can guide the watermark embedding process.

6.6 Watermark embedding

For each watermarking embedding regions defined by above procedures, the watermark embedding consists of following steps:

1. Use the pseudorandom number (PN) generator to generate a watermark data sequence, which is spread spectrum consisting of both positive and negative values.

2. The watermark data is adaptively embedded into the disk embedding regions selected in Section 6.4 using the following equation:

$$f_s^{'}(x,y) = \frac{1}{\mathcal{S}_s}\{f_s(x,y) + [1 - NVF(f_s(x,y))] \cdot \alpha \cdot w\} \tag{6.16}$$

where, $f_s(x,y)$ is one of the segmented region of the original image. $f_s^{'}(x,y)$ is the watermarked result. w is the watermark data. α is the watermark embedding strength which is pre-set as 2. $\mathcal{S}_s$ is the size of region.

$NVF(f_s(x,y))$ is the noise visibility function and is defined with the local mean μ_s and local variance σ_s as follows:

$$NVF(f_s(i,j)) = \frac{q(i,j)}{q(i,j) + \sigma_s^2} \tag{6.17}$$

$q(i,j)$ is a weighting factor and $q(i,j) = \beta(\eta(\beta))^{\beta}\frac{1}{||r(i,j)||^{2-\beta}}$ with $r(i,j) = \frac{f_Z(i,j)-\mu_s}{\sigma_s}$. β is the shaping factor of the generalized Gaussian distribution for the region and

$\eta(\beta) = \sqrt{\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}}$.

3. If the PSNR of the watermarked image is not less than 40 dB, the watermark embedding is succeeded. Else, the embedding strength, α will be reduced by 0.1 iteratively until either of the following criterion reached:

   a) α is not bigger than 1.5.

   b) the PSNR of the watermarked image is not less than 40 dB.

6.7 Watermark detection

To detect watermark, the linear correlation in Equ. (6.18) is used.

1. Use pseudo random number generator to generate the same watermark as the one used for watermark embedding.

2. As mentioned in Sec 5.2.1, locate the watermark RST invariant embedding/extraction regions using the Gaussian scale model.

3. Calculate the linear correlation of the watermark and the watermarked image using Equ. (6.18). The watermark is detected when the result is larger than the watermark embedding strength used for watermark embedding.

$$z_{lc} = \frac{1}{\mathcal{S}_s} \sum_{x,y \epsilon s} w \cdot f'_s(x,y) \tag{6.18}$$

where, z_{lc} is the linear correlation. s is one of the segmented regions. $\mathcal{S}_s$ is the area of region. w is the watermark generated using the first step above. $f'_s(x,y)$ is the watermarked image.

6.8 The evaluation of error probability

As developed in Section 5.4, the lower bound of error probability can be calculated the following equation:

$$P_e^S > \pi_0 \pi_1 e^{-J/2} \tag{6.19}$$

where, the J-divergence, $J = D(f_{Y^S}(y) \parallel f_{X^S}(y)) + D(f_{X^S}(y) \parallel f_{Y^S}(y))$, is symmetric and nonnegative.

Unfortunately, these relative entropies are unable to calculate numerically due to the complexity of the pdf functions of Y^S and X^S. An analytical solution for $D(f_{Y^S}(y) \parallel f_{X^S}(y))$ is given in [109] as follows:

$$D(f_{Y^S}(y) \parallel f_{X^S}(y)) = \frac{N_S}{2}[\frac{D_1}{\sigma_{X^S}^2} - \ln(1 + \frac{D_1}{\sigma_{X^S}^2})] \tag{6.20}$$

and

$$D(f_{X^S}(y) \parallel f_{Y^S}(y)) = \frac{N_S}{2}[\ln(1 + \frac{D_1}{\sigma_{X^S}^2}) - \frac{D_1}{\sigma_{X^S}^2 + D_1}] \tag{6.21}$$

where N_S is the total number of pixels in the region of S. In this thesis, we set the radius, R, of S equal to 21. Therefore, $N_S = \pi R^2 = 441\pi$. D_1 is the mean squared error per sample, which is defined as $D_1 = \frac{1}{N_S} E\|Y^S - X^S\|^2$.

Substituting Equ. (6.20) and Equ. (6.21) into Equ. (6.19), the lower bound of the error probability can be numerically solved.

$$\begin{aligned} P_e^S &> \frac{1}{4} exp\{-\frac{N_S}{4}[\frac{D_1}{\sigma_{X^S}^2} - \ln(1 + \frac{D_1}{\sigma_{X^S}^2}) + \ln(1 + \frac{D_1}{\sigma_{X^S}^2}) - \frac{D_1}{\sigma_{X^S}^2 + D_1}]\} \\ &> \frac{1}{4} exp\{-\frac{N_S}{4}[\frac{D_1}{\sigma_{X^S}^2} - \frac{D_1}{\sigma_{X^S}^2 + D_1}]\} \end{aligned} \tag{6.22}$$

The definition of PSNR is:

$$PSNR = 20\log_{10}\left(\frac{V_{MAX}}{\sqrt{D_1}}\right) \tag{6.23}$$

where, V_{MAX} is the maximum pixel value in regions s. Then, we can rewrite D_1 in terms of PSNR:

$$D_1 = \left(\frac{V_{MAX}}{10^{\frac{PSNR}{20}}}\right)^2 \tag{6.24}$$

Since we tried to control the PSNR of the watermarked image to be not lower than 40 dB, the lower bound of the error probability can be rewritten as:

$$P_e^s > \frac{1}{4}exp\left\{-\frac{N_s}{4}\left[\left(\frac{V_{MAX}}{10^{\frac{PSNR}{20}}\sigma_{x_s}}\right)^2 - \frac{V_{MAX}^2}{10^{\frac{PSNR}{10}}\sigma_{x_s}^2 + V_{MAX}^2}\right]\right\} \tag{6.25}$$

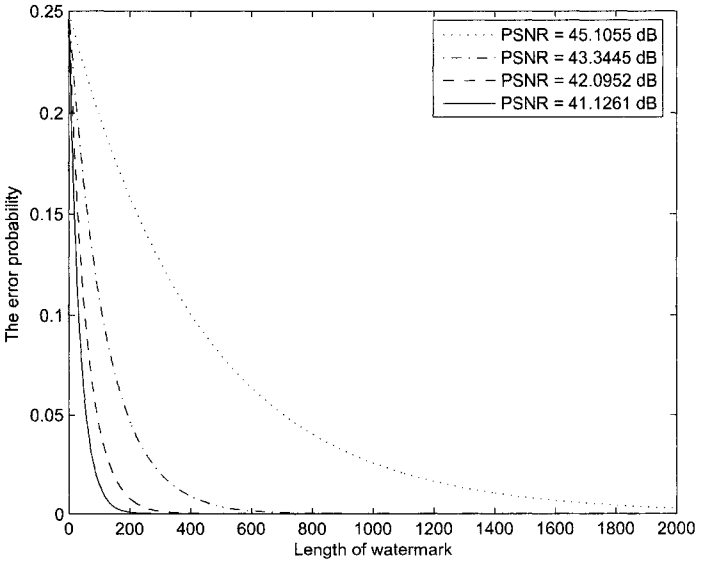


Figure 6.11: The lower bound of the error probability with respect to the length of watermark.

In one circular region, with fixed V_{MAX} and σ_{x_s}, the lower bound can be illustrated in Fig. 6.11.

From Fig. 6.11, it is shown that the larger the power of the watermark, the smaller the probability of error. This is consistent with the analysis. Also because the spread spectrum is used for watermark embedding, the length of the watermark in pseudo number sequence has the effect on the robustness of the watermarking scheme. The bigger the length, the more robust the watermark. Since the size of each watermark embedding region is 441π, the probability of error is very small given the PSNR after embedding is around 40 dB, this got confirmed in the following section.

This section gives a clear analysis between fidelity and robustness of the watermarking scheme.

Chapter 7

Experimental results

In this chapter, the experimental results will be presented to show the effectiveness of the proposed algorithm and the mathematical modelling. In Section 7.1, it is shown how the mathematical modelling and image segmentation guide the embedding region selection to enhance the watermark detectability. The histograms of the pixels in different segmented regions are also listed in Section 7.3 to show the validity of the mixture Generalized Gaussian modelling. All the 100 test images used in the experiments are listed in Section 7.4 which include a large variety of image contents with different characteristics. In Section 7.6, the robustness of the proposed watermarking scheme has been demonstrated. The watermarking scheme is very robustness to attacks including rotation, scaling, JPEG compression and noise pollution and performances well enough to achieve the goal. In Section 7.7, the performance comparison between the proposed algorithm and the algorithm in [73] is presented to shown the better performance of the proposed algorithm. In Section 7.8, the experimental results are presented to verify the theoretical derivation of probability of error in Section 6.8. The experimental results are consistent with the theoretical derivation.

7.1 The advantage of the proposed scheme

Linear correlation is used in the proposed watermarking scheme as the method of watermark detection. The analysis of various aspects of the watermarking processes including robustness and the probability of error is based on linear correlation. However, the linear correlation is affected by the characteristic of different image contents. In this section, we will show how the proposed scheme and modelling improve the performance of linear correlation based watermark detection step by step.

The basic concept of linear correlation based watermark detection is to calculate the linear correlation between the watermarked image and the original watermark and compare it with the threshold to decide the existence of the watermark. As defined in [1], the linear correlation is,

$$Z_{lc} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} f'(i,j) \cdot w(i,j) \tag{7.1}$$

where, Z_{lc} is the linear correlation calculated between the watermarked image and the original watermark. $M \times N$ is the length of the watermark or the total pixels in the watermarked region. w is the watermark. f' is the watermarked image. The watermarking process is represented as following:

$$f' = f + \alpha \cdot w \tag{7.2}$$

The embedding strength, α, is set to 2 and f is the original image.

In [1], the watermark is generated from Normal distribution $N(0,1)$. The pixels in the original image, $f(x,y)$, are assumed to be i.i.d. Gaussian distribution. Therefore, all the pixels in the original image are assumed to have the same stochastic characteristics.

And the original image and the watermark are assumed to be uncorrelated. So, the linear correlation can be calculated as,

$$\begin{aligned} f' \cdot w &= f \cdot w + \alpha \cdot w \cdot w \\ &= 0 + \alpha \\ &= \alpha \end{aligned} \tag{7.3}$$

where, '$\cdot$' means linear correlation.

Therefore, the linear correlation calculated between the watermarked image and the original watermark is,

$$Z_{lc} = \alpha \tag{7.4}$$

The result in Equ. (7.4) is achieved based on the assumption that the pixels in the entire image are under the Gaussian distribution. Theoretically, the result of linear correlation between the watermarked image and the watermark should be close to α and the the result of linear correlation between the non-watermarked image and the watermark should be close to 0 if there is no distortion or attack involved. Then the threshold used to determine the existence of watermark can be clearly defined. However due to the complexity of natural image contents and characteristics, the result of linear correlation fluctuates a lot and make the correct detection of the existence of watermark impossible without extra measure being taken. The watermarking scheme proposed in this thesis solves this problem and the effectiveness of the proposed scheme is shown in the following experiments.

7.1.1 Experiment I

This experiment shows the ideal case: the host image is generated under the Gaussian distribution with mean μ and standard deviation σ as shown in Fig. 7.1.

Figure 7.1: The 42×42 image generated under the Gaussian distribution.

Two 42×42 watermarks are generated under the Normal distribution $N(0, 1)$ as shown in Fig. 7.2.

(a) Watermark pattern 1

(b) Watermark pattern 2

Figure 7.2: Two 42×42 watermarks.

The two watermarks are embedded into the image in Fig. 7.1 respectively. Using Equ. (7.1), the linear correlations between the watermarked images and the corresponding watermark patterns are calculated to be 2.0129 and 2.3413 respectively. Also the linear correlation between the unwatermarked images and the watermark patterns

are 0.0213 and 0.0179 respectively. The results of linear correlation are consistent with the theoretical analysis mentioned above. The linear correlations between the watermarked/unwatermarked images and 100 different watermark patterns randomly generated under the Normal distribution $N(0,1)$ are calculated. The result is shown in Fig. 7.3. It is clearly shown in the figure that the 100 linear correlation values of the watermarked images stay close to the expected value, α, which equals to 2. The variance of the linear correlations is 0.0072. Also the 100 linear correlation values of the unwatermarked images are around 0 and the variance is 0.0037. Therefore, the existence of the watermark can be accurately detected.

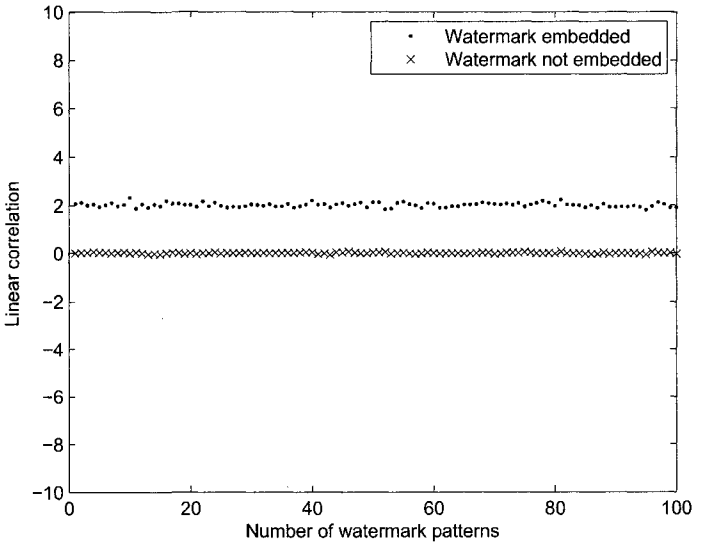


Figure 7.3: The 100 linear correlations calculated between the watermarked/unwatermarked image and 100 different watermark patterns.

7.1.2 Experiment II

This experiment shows the real case: we verify how actually the stochastic characteristic of natural images affects the ability of linear correlation to detect the existence of the watermark. The experiment has the same procedure as Experiment I except the image is a part of natural image instead of the image generated under the Gaussian distribution.

Figure 7.4: The 42×42 square region on *Barbara*.

The 42×42 square region randomly cut from *Barbara* is used as the image for watermark embedding and detection as shown in Fig. 7.4. The two watermark patterns in Fig. 7.2 are embedded into the square region in Fig. 7.4 respectively. The linear correlation between the watermark in Fig. 7.2 (a) and the corresponding watermarked image is calculated to be 2.5631 and the linear correlation between the watermark in 7.2 (b) and the corresponding watermarked image is calculated to be 0.2584. This makes it dif-

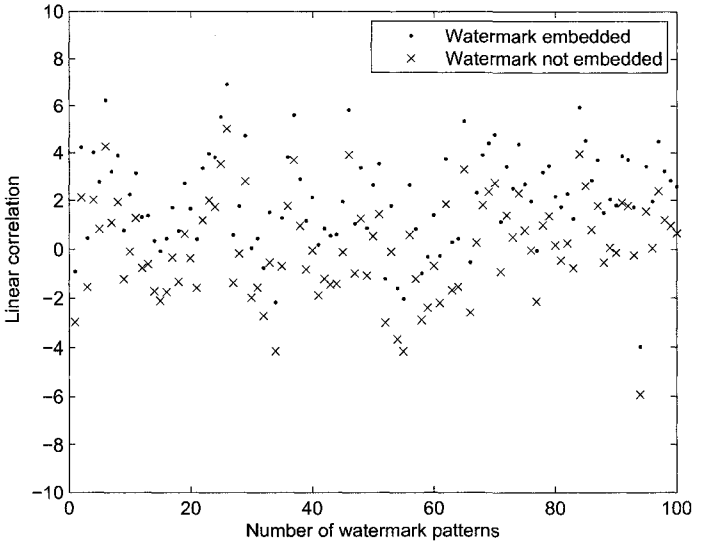


Figure 7.5: The linear correlations calculated between the watermarked image and the same 100 random watermarks used in Experiment I. The false positive probability is 0.44. And the false negative probability is 0.22.

ficult to discriminate the non-watermarked image and the watermarked image. Also the 100 linear correlations are calculated between the watermarked/unwatermarked images and the same 100 watermark patterns are shown in Fig. 7.5. The linear correlations of both the watermarked images and the unwatermarked images range from -6.1297 to 6.7336 with variance of 4.0267. By setting the threshold to 0.5, the false negative probability error is 0.22 and the false positive probability is 0.44. The experiment shows that the linear correlation is sensitive to the watermark patterns and image contents, which makes it difficult to analyze the watermarking system using linear correlation as the watermark detector. The fluctuation of the lincar correlation values is caused by the correlation between the watermark and the natural image content. To solve this problem, the whitening filter is used to decorrelate the watermark and the image.

The whitening filter mention in [1] is shown in Fig. 7.6.

By applying the whitening filter on the watermarked image and the watermark before calculating the linear correlation, the results are improved as shown in Fig. 7.7.

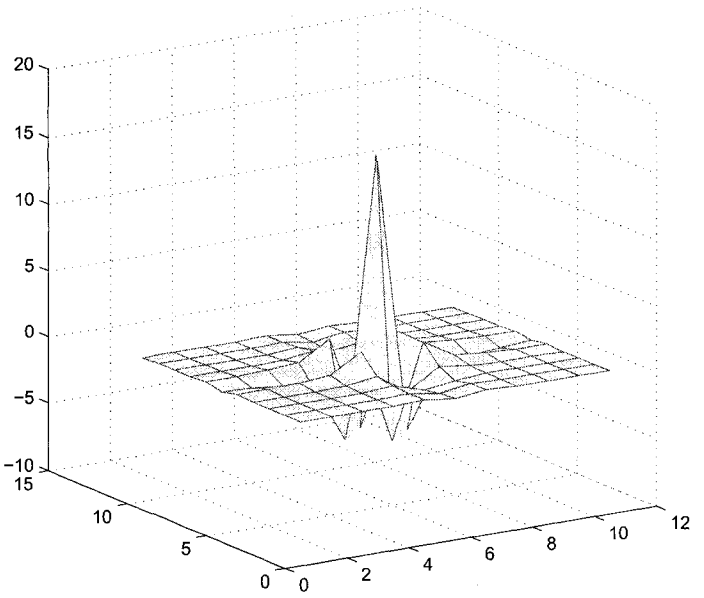


Figure 7.6: The 3-D plot of whitening filter.

The linear correlations for the watermarked images range from 1.0174 to 3.3187. We can see that the linear correlations are much more converged to the expected value, but the variance is still 0.2385. By setting the threshold to 0.5, the false positive probability is 0.05 and the false negative probability is 0 which is much improved from the previous result. However, the variance is still relatively large, the probability of error may be greatly increased when the watermarked image is distorted. Therefore, to further reduce the variance of the linear correlation is critical for the success of watermark detection, which in turn leads to the better performance of robustness against attacks.

To further reduce the variance of the linear correlation, it is important that the wa-

termark embedding region should be a homogeneous region. To model the cover image

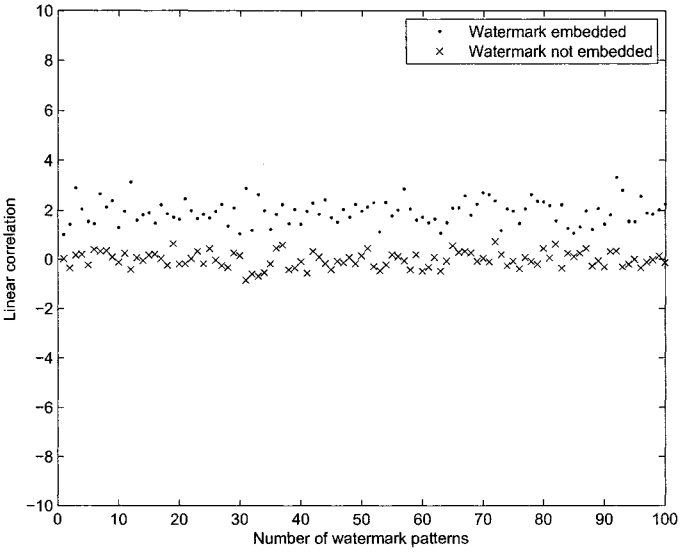


Figure 7.7: The 100 linear correlations calculated with 100 different watermarks. The false positive probability is 0.05. The false negative probability is 0.

using the mixture Generalized Gaussian distribution, the MAP image segmentation is used to segment the image into several homogeneous regions. Each region can be represented using a Generalized Gaussian distribution with parameters estimated using EM (Expectation Maximization) algorithm. This segmentation provides a good method to locate the suitable watermark embedding regions. Also the SIFT (Scale Invariant Feature) feature extraction algorithm locates the salient feature points which work as the reference points and are used to define the circular regions for watermark embedding and extraction. Major part of each embedding or extraction region belongs to one segmented region. Therefore the characteristics within each embedding or extraction region is uniform. The experimental result proves the effectiveness of this approach.

7.1.3 Experiment III

In this experiment, a 42×42 square region of *Barbara* is selected for watermark embedding and detection as shown in Fig. 7.8. The '+' denotes a selected SIFT feature point. This region is selected with the help of MAP image segmentation and SIFT feature detection as mentioned in Section 6.2 and Section 6.3. The decorrelation filter is applied to the watermarked image and watermark before calculating the linear correlation.

The linear correlation is calculated between the watermarked image and the two watermark patterns in Fig. 7.2, respectively. For the watermark pattern in Fig. 7.2 (a), the linear correlation is 2.0131 and for the watermark pattern in Fig. 7.2 (b), the linear correlation is 2.2839. The linear correlations calculated between the water-

Figure 7.8: The 42×42 square cut from *Barbara*.

marked/unwatermarked images and the same 100 random watermark patterns used in Experiment I are shown in Fig. 7.9. The linear correlations of the watermarked images range from 1.6701 to 2.2763 with a variance of 0.0138 which is much closer to the ideal variance, 0.0072, that that of Fig. 7.7. The mean of the linear correlations is 1.9318 and is very close to 2. Also the linear correlations of the unwatermarked images are close to 0. By setting the threshold to 0.5, both the false positive probability and the false negative probability are 0. This experiment shows that the proposed algorithm is more effective in selecting the suitable watermark embedding and detection regions.

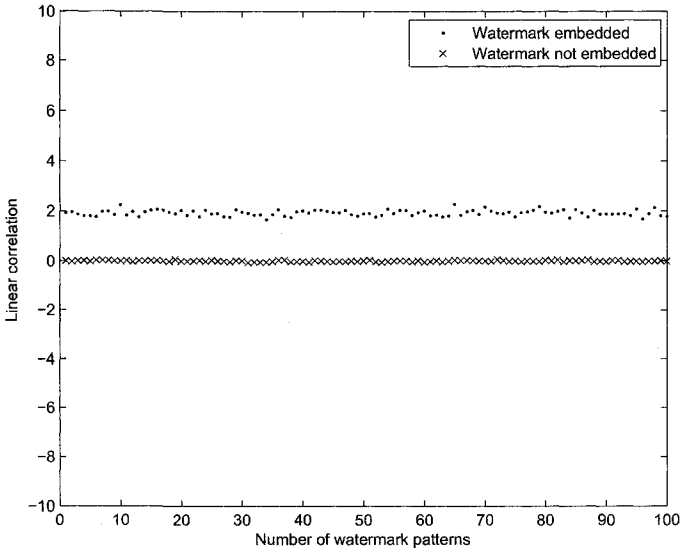
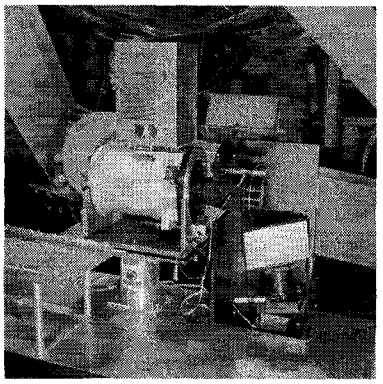


Figure 7.9: The 100 linear correlations calculated with 100 different watermarks. Both the false positive probability and false negative probability are 0.

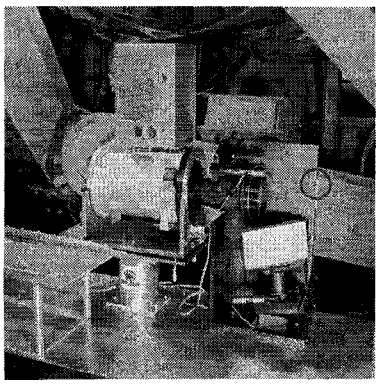
7.1.4 Experiment IV

To further illustrate the effectiveness of the proposed scheme, the following comparison experiments are done with one watermark and 3 different images under rotation.

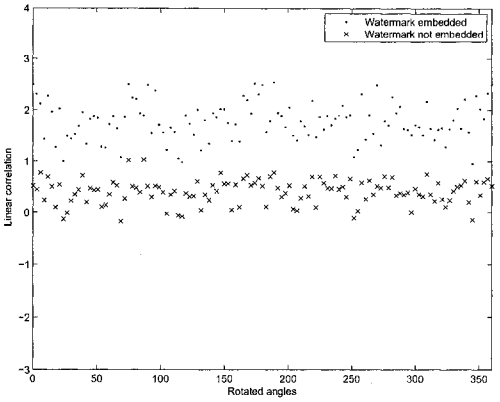
As shown in Fig. 7.10 (a), Fig. 7.11 (a) and Fig. 7.12 (a), the red circles are the randomly selected circular regions from the 3 images. All of the regions are modelled



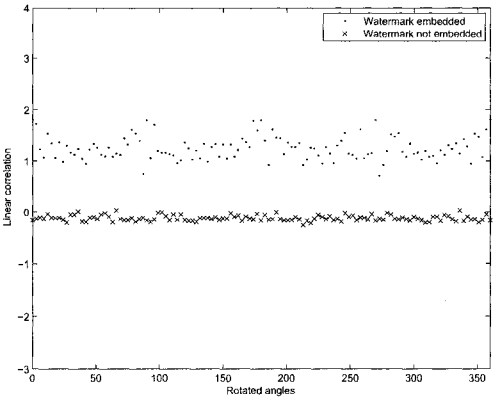
(a) A randomly selected circular region.



(b) The circular region centered at the selected SIFT feature point.



(c) The linear correlation calculated on the randomly selected region shown in (a).

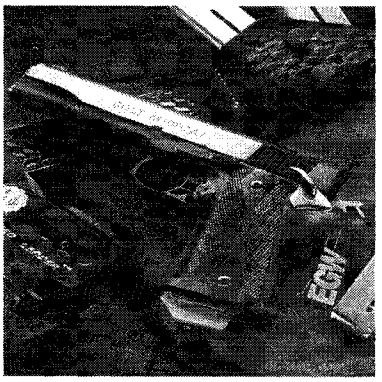


(d) The linear correlation calculated on the feature centered region shown in (b).

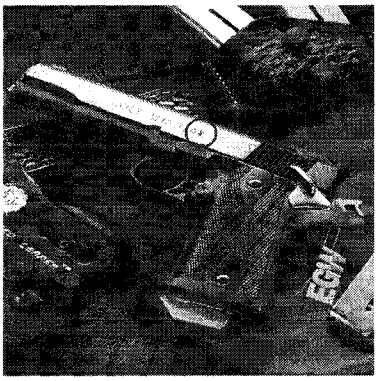
Figure 7.10: Comparison between the linear correlation calculated on random selected region and SIFT feature centered region on image 16.

using the Gaussian distribution. One watermark is embedded into the 3 circular regions with the embedding strength of 2 respectively. After watermark embedding, the watermarked image is rotated with angles varying from 0^o to 360^o with a step of 3^o. For each red region, we detected the watermark under each rotation distortion by compar-

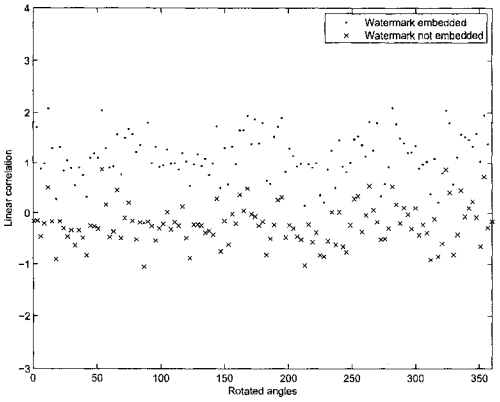
ing the linear correlation values calculated between the watermarked/unwatermarked region and the original watermark with the threshold of 0.5. And the results are shown in Fig. 7.10 (c), 7.11 (c) and Fig. 7.12 (c). The '.' denoted curve is the linear correlations calculated when watermark is embedded. And the 'x' denoted curve is the linear correlations calculated when watermark is not embedded.



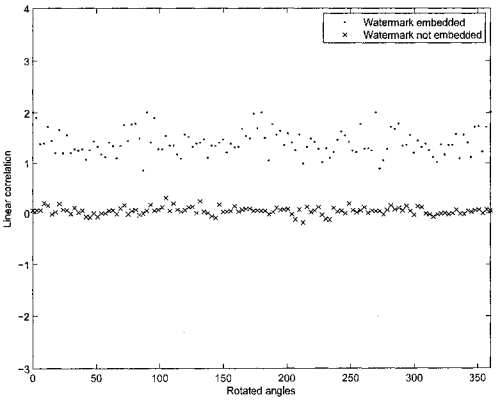
(a) A randomly selected circular region.



(b) The circular region centered at the selected SIFT feature point.



(c) The linear correlation calculated on the randomly selected region shown in (a).



(d) The linear correlation calculated on the feature centered region shown in (b).

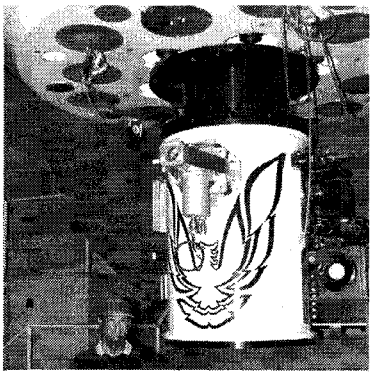
Figure 7.11: Comparison between the linear correlation calculated on random selected region and SIFT feature centered region on image 21.

The false positive probability Fig. 7.10 (c) is 0.4215. The false negative probability

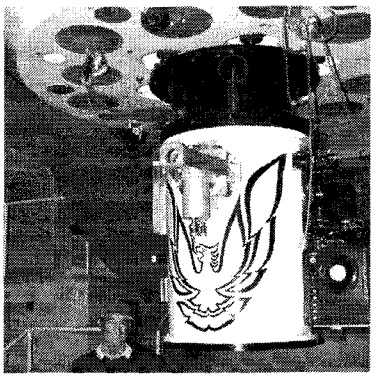
is 0.

The false positive probability Fig. 7.11 (c) is 0.0496. The false negative probability is 0.0074.

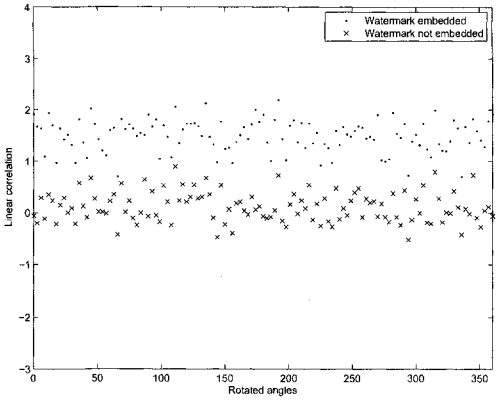
The false positive probability Fig. 7.12 (c) is 0.124. The false negative probability is 0.



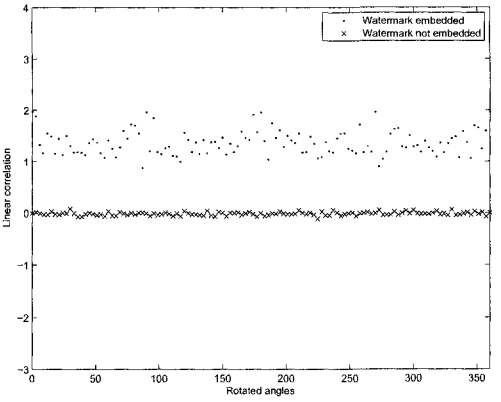
(a) A randomly selected circular region.



(b) The circular region centered at the selected SIFT feature point.



(c) The linear correlation calculated on randomly selected regions shown in (a).



(d) The linear correlation calculated on the feature centered region shown in (b).

Figure 7.12: Comparison between the linear correlation calculated on random selected region and SIFT feature centered region on image 36.

The red circles in Fig. 7.10 (b), 7.11 (b) and Fig. 7.12 (b) are selected from

the segmented regions and are centered at the feature points. These regions can be accurately modelled using the Generalized Gaussian with specific parameters. The '+' denoted points are the feature points. The linear correlation calculated on the feature centered regions are shown in Fig. 7.10 (d), 7.11 (d) and Fig. 7.12 (d). Both the false positive and false negative probabilities are 0 for the 3 red circular regions.

By comparing the two sets of results shown in Fig. 7.10 (c), 7.11 (c) 7.12 (c) with the results in Fig. 7.10 (d), 7.11 (d) Fig. 7.12 (d), it can be seen that the homogeneous regions are more suitable for watermark embedding.

The four experiments in this section experimentally proved the effectiveness of the proposed scheme. The MAP segmentation of images is not only useful for mathematically modelling the image into the mixture Generalized Gaussian distribution, but also provides the homogeneous watermark embedding regions which make the linear correlation results more converged and less fluctuated. This improves the detector ability to determine the existence of watermark by having a clearer distinction between the linear correlation values. This also enhances the robustness of the watermarking scheme under various distortion as presented in the following sections.

7.2 The radius R of the circular region for the watermark embedding and detection

The linear correlation is used to detect watermark from the circular regions. The embedding strength, $\alpha = 2$, is used to achieve the balance between the watermark robustness and fidelity. The initial value of the radius, R, of the circular regions is set to 10. To find an appropriate value of R for watermark detection, the linear correlation is evaluated by varying R according to the flowchart shown in Fig. 7.13.

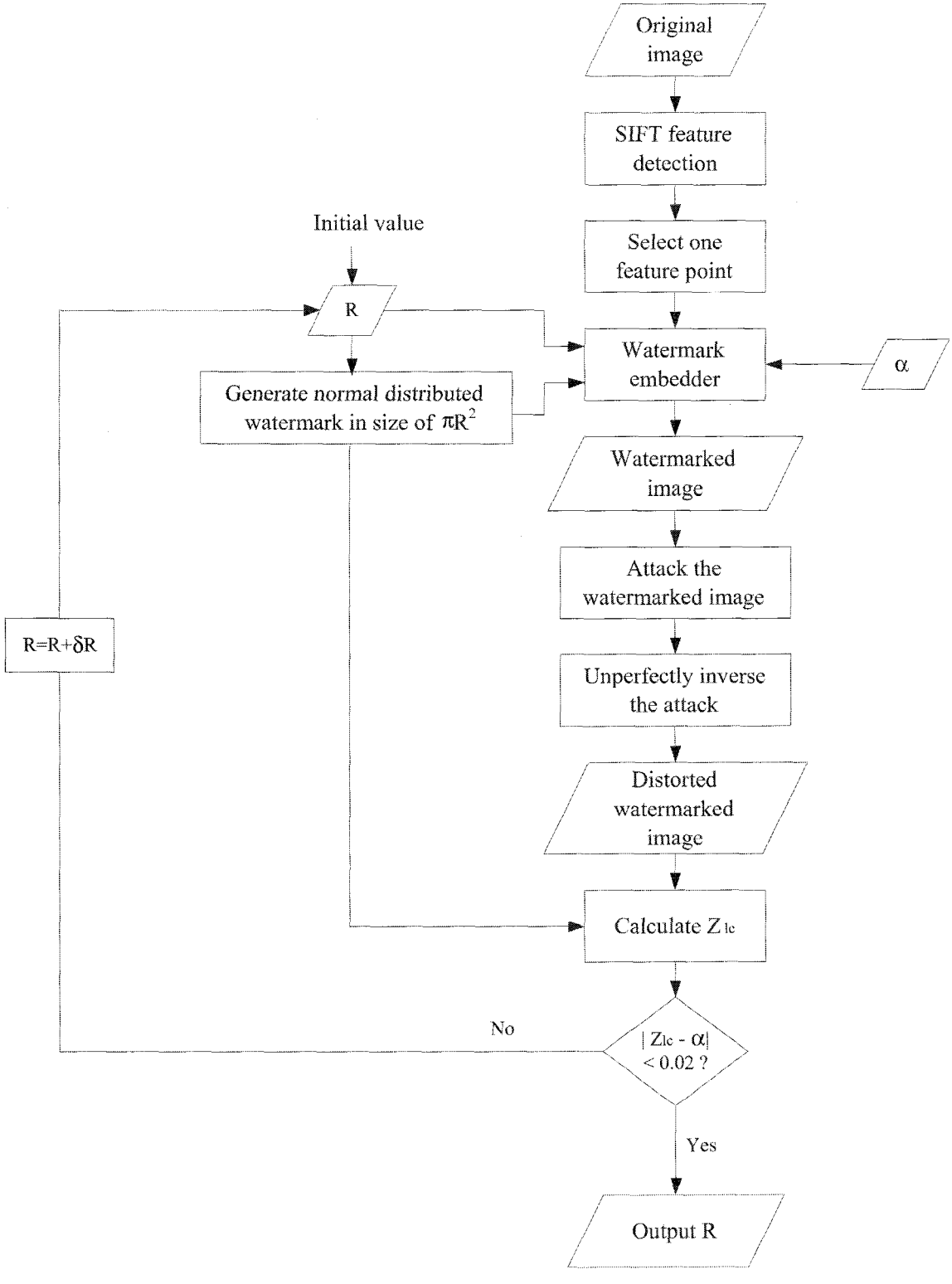


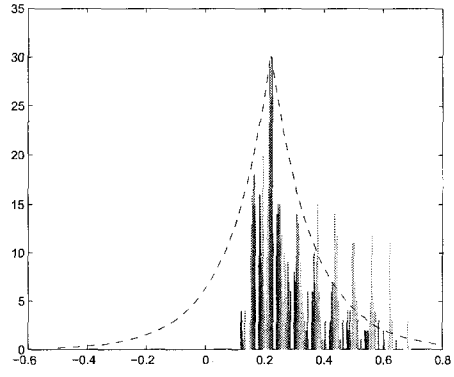
Figure 7.13: The set up of R for watermark embedding/detection.

With the initial value of R, the watermark is generated in size of πR^2 under normal distribution and the watermark is embedded into one selected circular region with the embedding strength of 2 to get the watermarked image. The rotation is applied to the watermarked image. After the watermarked image is rotated with an angle of θ, it is inversely rotated with an angle of $-\theta + \Delta\theta$. In the experiment, the $\Delta\theta$ is randomly generated under normal distribution. Then the linear correlation is calculated between the distorted watermarked image and the original watermark. The R is defined such that the criteria of $|Z_{lc} - \alpha| < 0.02$ is met and the R is the closest to its initial value. The criteria makes that the linear correlation calculated with R is robust to distortion. Also R should be selected such that there is enough space to embed more watermark bits into multiple circular regions. According to this process, R is set to be 21, which is suitable for all the test images used in the thesis.

7.3 Histograms of three selected circular regions

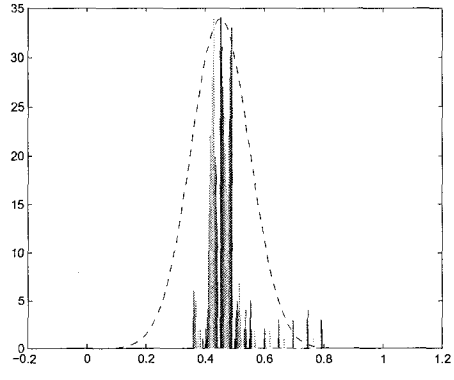
The mixture Generalized Gaussian distribution is used to model the image. The MAP image segmentation is used to segment the image into different homogeneous regions. Each region is approximated using a Generalized Gaussian distribution with specific parameters. Fig. 7.14 (b), (d), (f) show the histograms of the pixel values and the approximated distributions of the circular regions shown in Fig. 7.14 (a), (c), (e), respectively. The radius of the circular regions is 21. It is shown that the distribution of the pixels can be approximated as the generalized Gaussian distribution.

(a) Circular region centered at the selected feature point. The area inside the circle is the circular region. The + in the middle of the circle denotes the SIFT feature point.



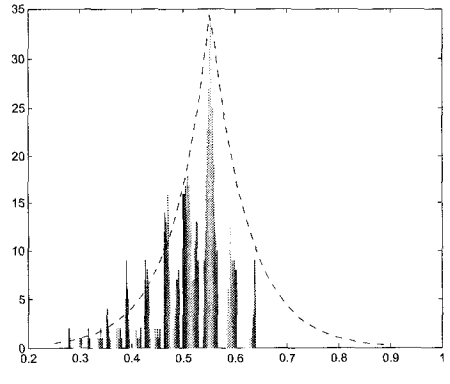
(b) Histogram of the circular region. The dashed curve is the approximated Laplace distribution with mean 0.22, variance 0.2.

(c) Circular region centered at the selected feature point. The area inside the circle is the circular region. The + in the middle of the circle denotes the SIFT feature point.



(d) Histogram of the circular region. The dashed curve is the approximated Gaussian distribution with mean 0.45, variance 0.1.

(e) Circular region centered at the selected feature point. The area inside the circle is the circular region. The + in the middle of the circle denotes the SIFT feature point.



(f) Histogram of the circular region. The dashed curve is the approximated Laplace distribution with mean 0.552, variance 0.1.

Figure 7.14: Histogram of three selected circular regions. The radius of the circular region is 21.

7.4 The 100 original images used in the experiments

The validity of the image watermarking scheme to a variety of natural images is critical. For all the tests listed in Section 7.6, 100 images are used. These 100 images contain a variety of portraits, landscapes, peoples, natural scenes which should be able to cover most of used scenarios for the image watermarking scheme. These 100 images are shown in Fig. 7.15.


CARD CAPTOR SAKURA

Figure 7.15: The 100 original image used in the experiments.

7.5 The noise visibility function

The NVF (noise visibility function) is used to adjust the embedding strength according to the characteristic of the images.

$$y = x + (1 - NVF) \cdot \alpha' \cdot w \tag{7.5}$$

The $(1 - NVF) \cdot \alpha'$ will be calculated for each pixel according to NVF and for each watermark embedding region. The average of the $(1 - NVF) \cdot \alpha'$ from all the pixels is the watermark embedding strength used in the experiments denoted by α, which should be around 2. In the following experiments in Section 7.6 and Section 7.7, the embedding strength is an average value instead of a fixed value for all the pixels involved in the watermarking processes.

The advantage of NVF is that it changes the watermark embedding strength based on the local characteristic of the pixels. The general principle of NVF is that in the region with larger variance, the watermark can be embedded with larger strength and in the region with smaller variance, the watermark should be embedded with smaller strength.

In the following table, $(1 - NVF) \cdot \alpha'$ means the embedding strength is adjusted by calculating the noise visibility function for each pixel in the watermark embedding strength. And α is the average value of the the $(1 - NVF) \cdot \alpha'$ from all the pixels involved in the watermark embedding. The fixed α which means the watermark is embedded using a fixed strength for all the pixels, the fixed strength is the average value of $(1 - NVF) \cdot \alpha'$. For example, for the image 1, when using the $(1 - NVF) \cdot \alpha'$ to do the watermark embedding, the PSNR and weighted PSNR of the watermarked image is 41.9794 dB and 51.4267 dB. The average of all the $(1 - NVF) \cdot \alpha'$ is 2.2466.

If we use this average value as the fixed embedding strength, the PSNR and weighted PSNR of the watermarked image is 42.0698 dB and 47.6803 dB. It can be seen the PSNR is roughly the same since the average embedding strength in both tests are the same. However the weighted PSNR shows that the watermark embedding using NVF to adjust the embedding strength has better quality in terms of wPSNR. All the tests in Tab. 7.1 show the similar results which prove the effectiveness of the NVF based watermark embedding strength adjusting mechanism to improve the fidelity.

Table 7.1: The comparison of the fidelity by using NVF adjusted embedding strength and the fixed embedding strength

	fixed α		$(1 - NVF) \cdot \alpha'$		
Image	PSNR (dB)	WPSNR (dB)	PSNR (dB)	WPSNR (dB)	Value of α
Img 1	42.0698	47.6803	41.9794	51.4267	2.2466
Img 2	43.9494	44.5077	47.5903	49.8012	1.8516
Img 3	41.0289	49.3541	41.1757	50.2136	2.5916
Img 4	47.7531	49.3941	47.7766	52.4080	1.7950
Img 5	44.1644	47.4913	44.2062	51.1759	1.7856
Img 6	44.8380	47.9676	45.3291	48.9260	1.9715
Img 7	44.4356	47.5960	44.5729	52.8821	1.9508
Img 8	44.7485	45.3958	46.3494	48.3558	1.6888
Img 9	42.1379	46.3744	41.9603	48.1948	2.2809
Img 10	41.5651	45.0344	41.9227	46.8617	2.4364

7.6 Robustness of the proposed watermarking scheme

The performance of our proposed algorithm can be demonstrated using the following four sets of experiments. The algorithm is tested on 100 different images as shown in Section 7.4. For 100 images, the same watermark is embedded into the circular region of

the images respectively. The watermark sequence is generated randomly under normal distribution $N(0,1)$. The radius of circular regions is selected to be 21 using the rule addressed in Section 7.2. The quality of the watermarked region in terms of PSNR is around 42 dB.

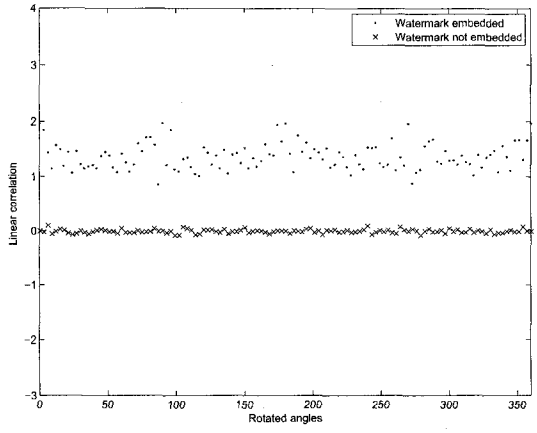
To show the effectiveness of our algorithm, we determine the existence of the watermark in the watermarked region by comparing the linear correlation with the threshold. Rotation, scaling, JPEG compression and Gaussian noise pollution are included in the testing. The experimental results for these four types of attacks are listed in Section 7.6.1, 7.6.2, 7.6.3 and Section 7.6.4. By comparing the value of linear correlation with the threshold,

$$\begin{cases} z_{lc} > 0.5 & Watermark\ is\ present \\ z_{lc} \leq 0.5 & Watermark\ is\ absent \end{cases} \tag{7.6}$$

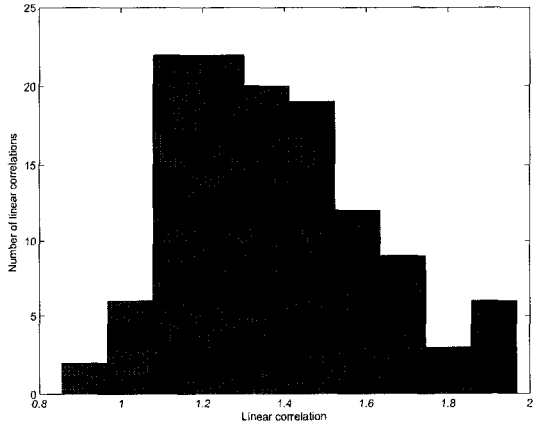
the existence of the watermark is determined. If the linear correlation is smaller than the threshold and the image is watermarked, the false negative probability error occurs. If the linear correlation is larger than the threshold and the image is unwatermarked, the false positive probability error occurs. The probability of error includes the two parts, false negative probability and false positive probability. The more robust the watermarking scheme is, the smaller probability of error will be. For 100 images, the test results show the robustness of the watermarking scheme since the probabilities of error are 0 for all the following tests. For the purpose of demonstration, the results of the first 20 test images are shown in the following subsections. In all the following figures, the '.' represents the value of the linear correlation of the watermarked image and the 'x' represents the value of the linear correlation of the unwatermarked image.

7.6.1 Rotation

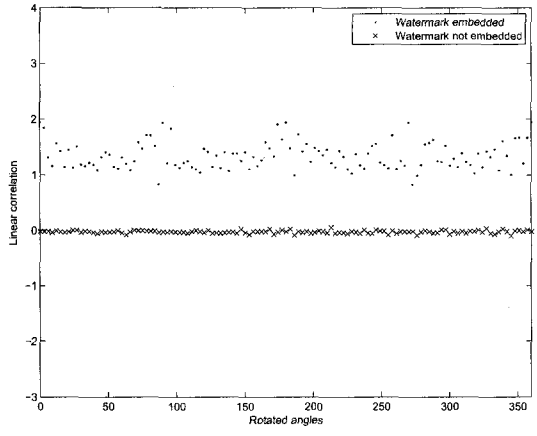
Under rotation, the algorithm is tested with rotation angles varying from 0^o to 360^o with the step of 3^o. Therefore, for each image, we tested the algorithm for 121 times. And the results are shown in Fig. 7.16. The odd numbered figures, such as $R1, R3, R5, \cdots, R39$ are the linear correlations calculated on the watermarked regions of image 1 to 20, respectively. And the even numbered figures, such as $R2, R4, R6, \cdots, R40$ are the corresponding histograms of the linear correlations calculated on image 1 to 20. From Fig. 7.16, it can be seen that the watermark detector can determine the existence of watermark correctly since the linear correlation values are all larger than 0.5 for the watermarked images and the linear correlation values are all smaller than 0.5 for the unwatermarked image. The histograms in Fig. 7.16 give a clear illustration of the distribution of the linear correlation values for the watermarked images. The results of the rest 80 images have quite similar results as the ones in Fig. 7.16. Therefore, only results for the first 20 images are shown here. Both the false positive probabilities and the false negative probabilities are 0, which means that the watermark can be correctly detected under the rotation. The robustness against rotation of the proposed scheme is very good.



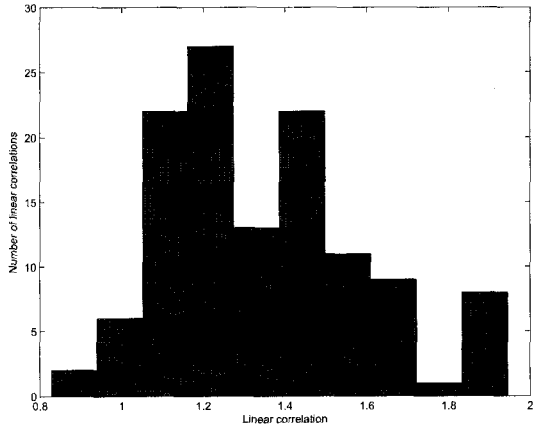
(R1)



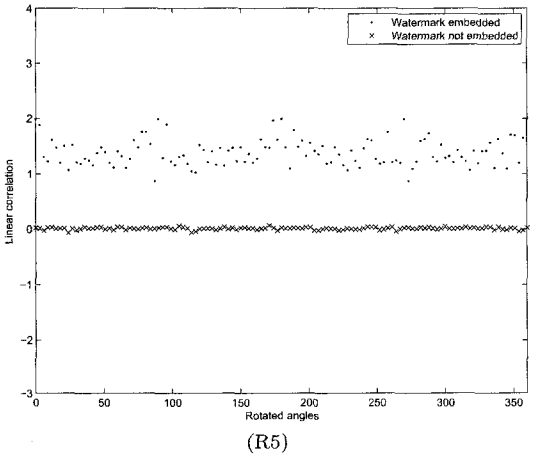
(R2)



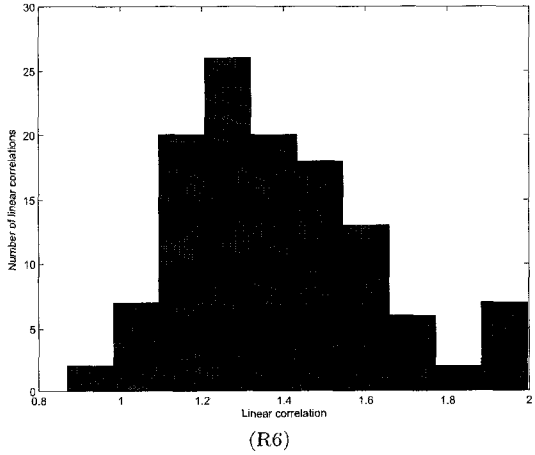
(R3)



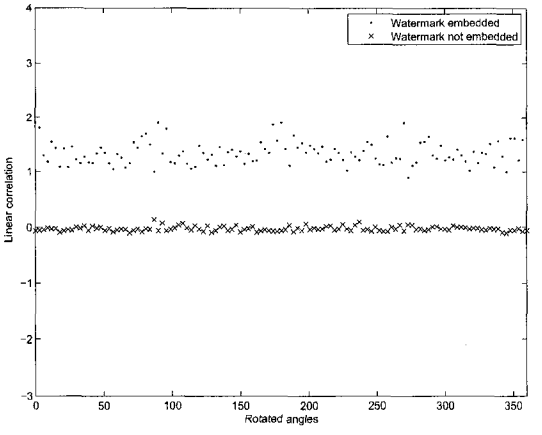
(R4)



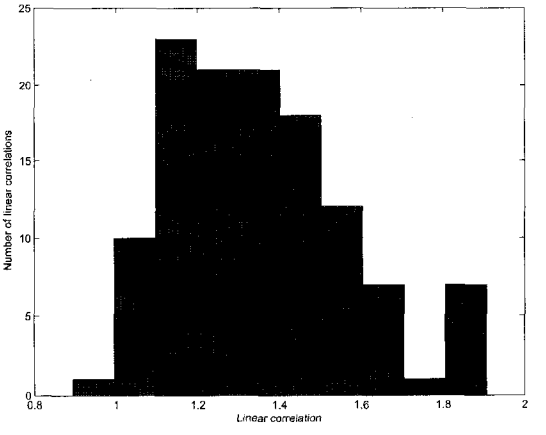
(R5)



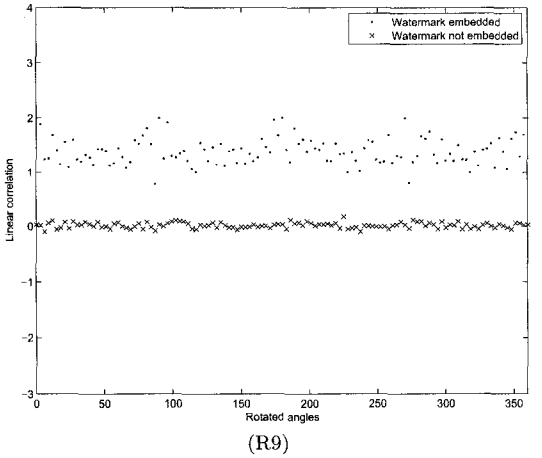
(R6)



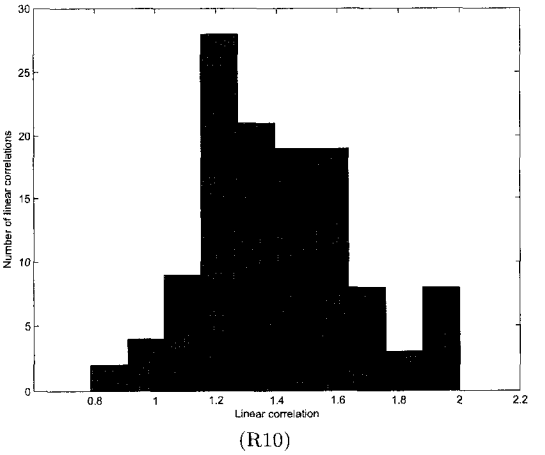
(R7)



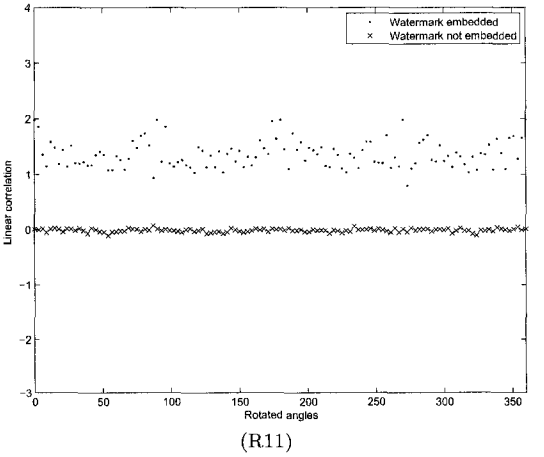
(R8)



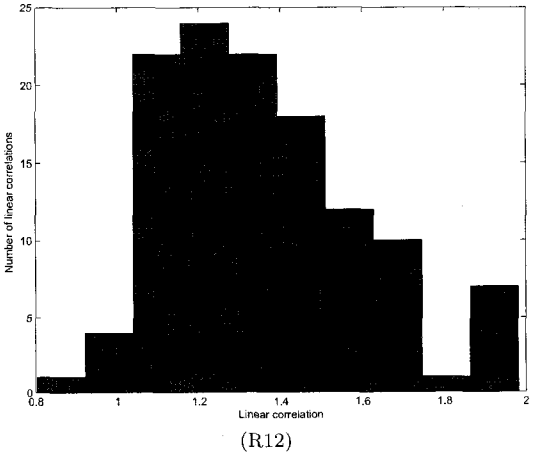
(R9)



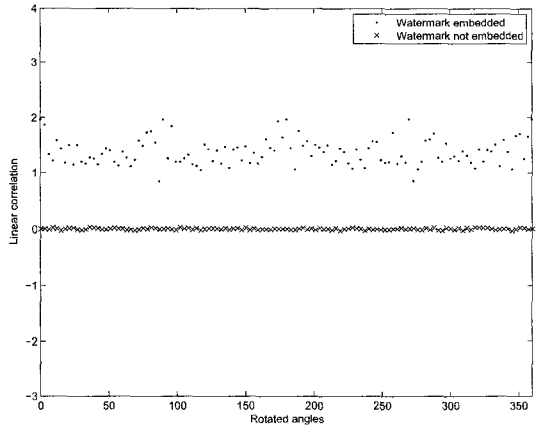
(R10)



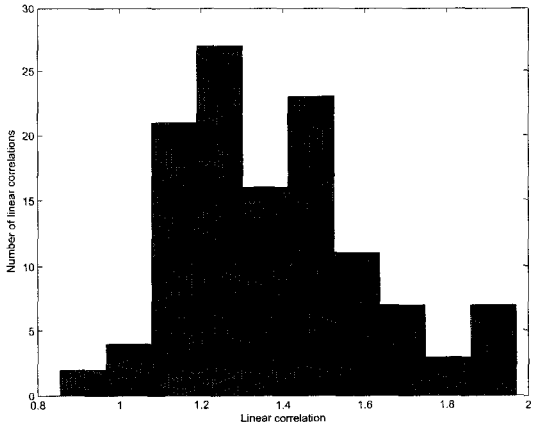
(R11)



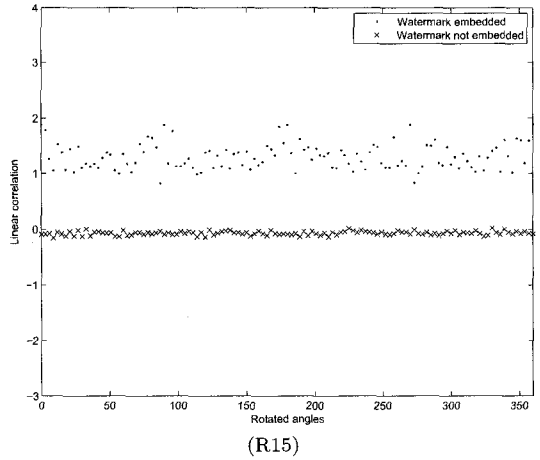
(R12)



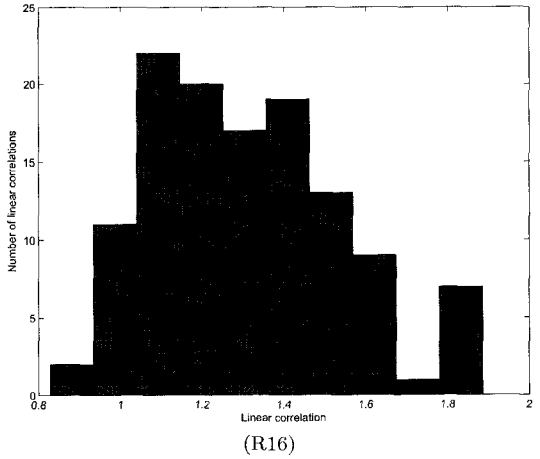
(R13)



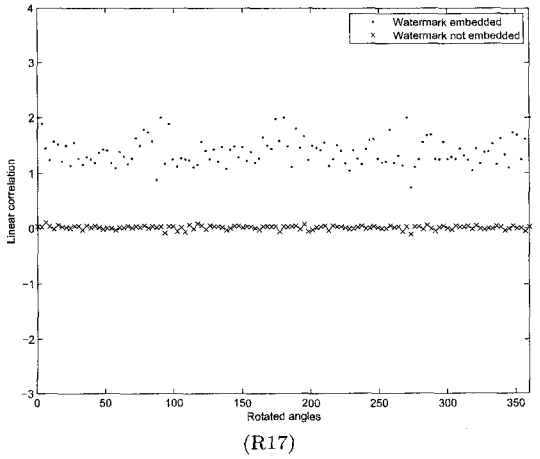
(R14)



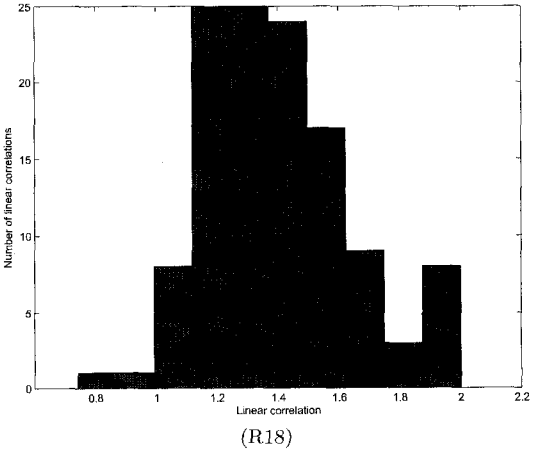
(R15)



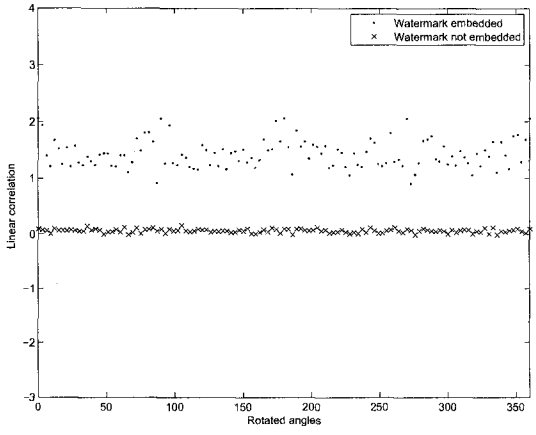
(R16)



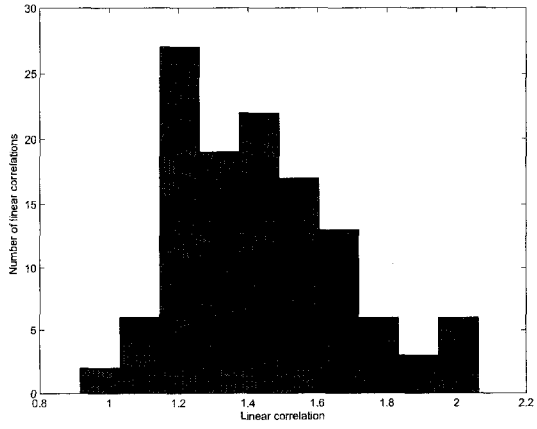
(R17)



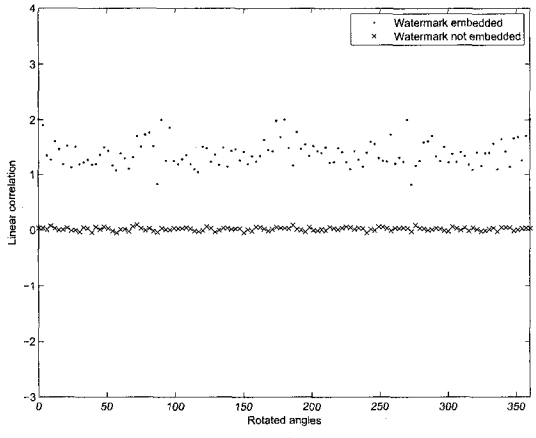
(R18)



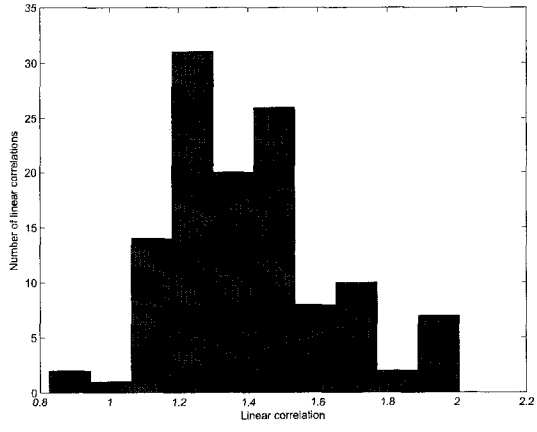
(R19)



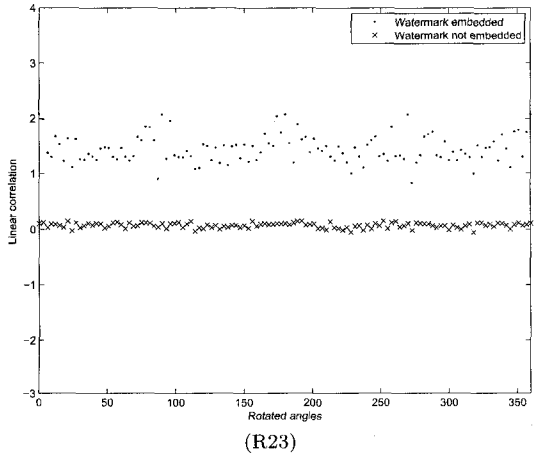
(R20)



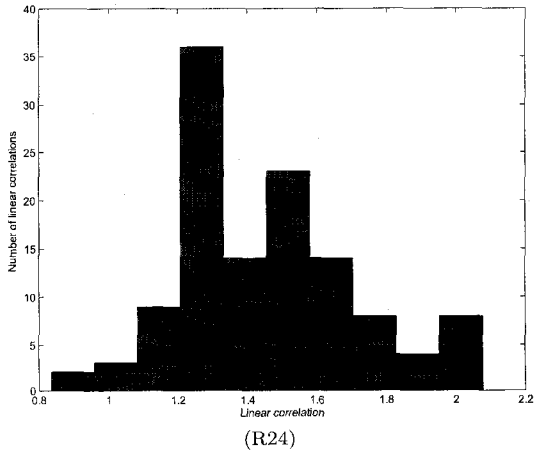
(R21)



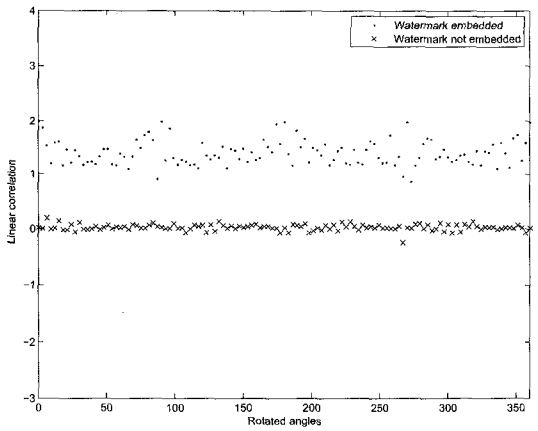
(R22)



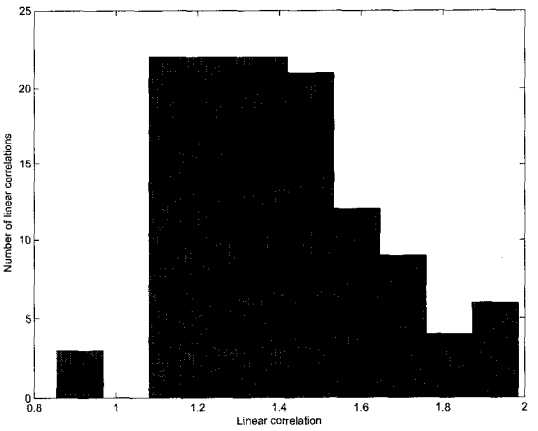
(R23)



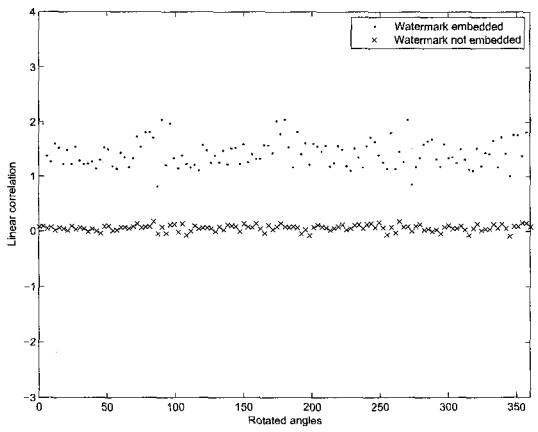
(R24)



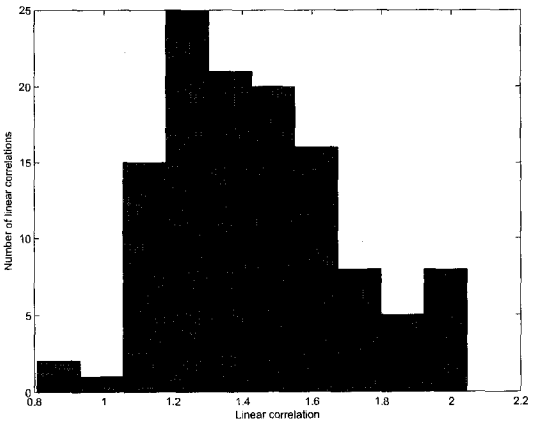
(R25)



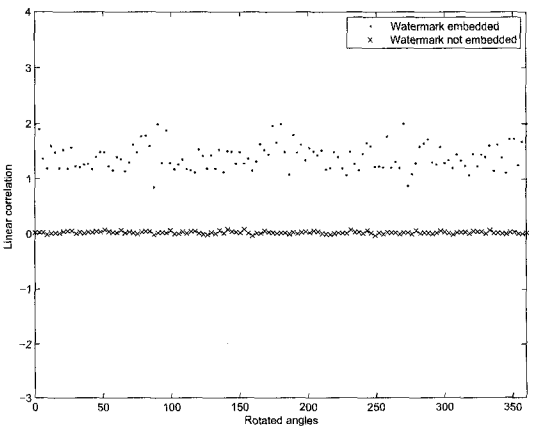
(R26)



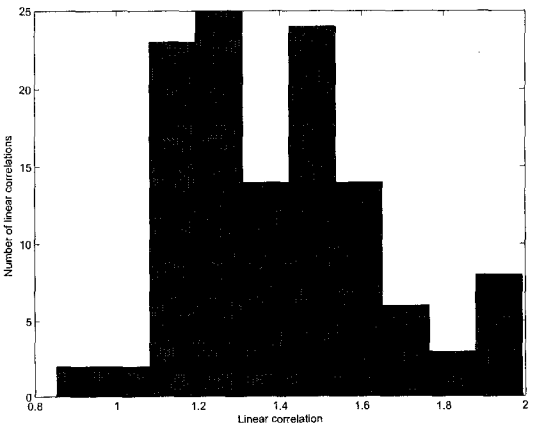
(R27)



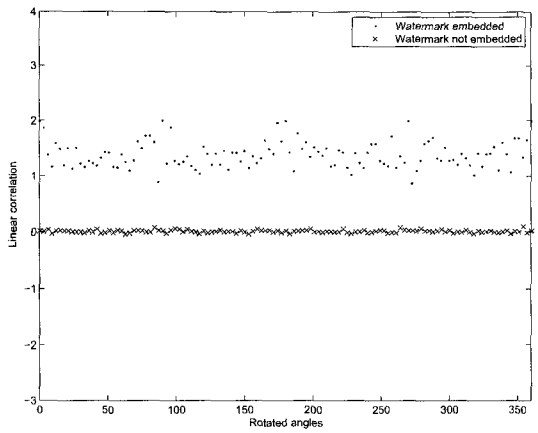
(R28)



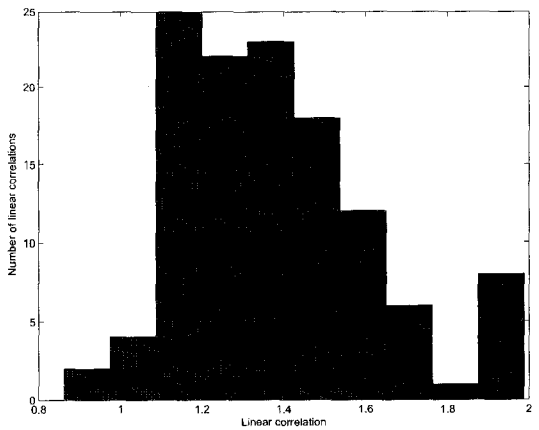
(R29)



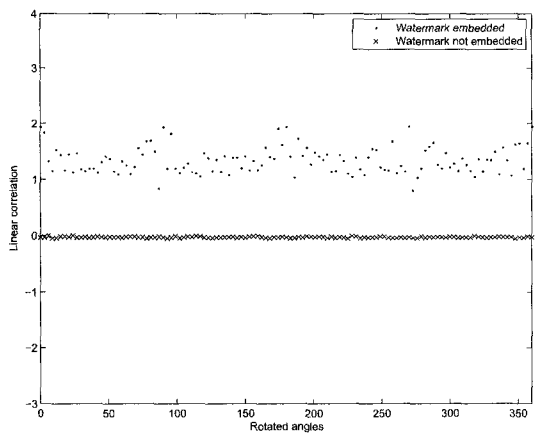
(R30)



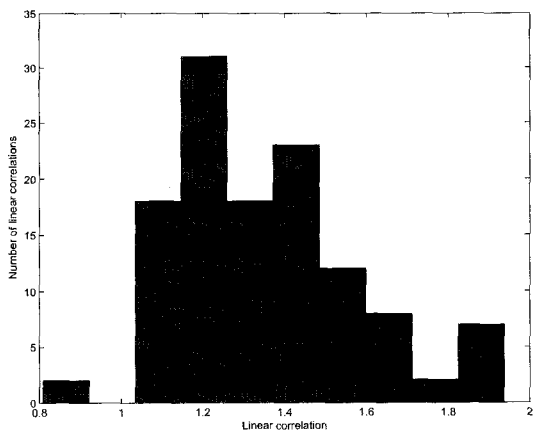
(R31)



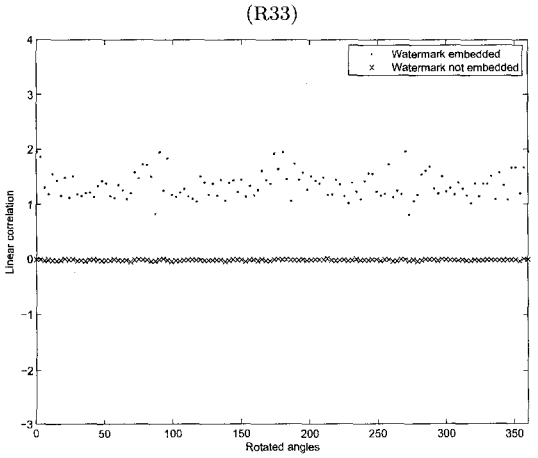
(R32)



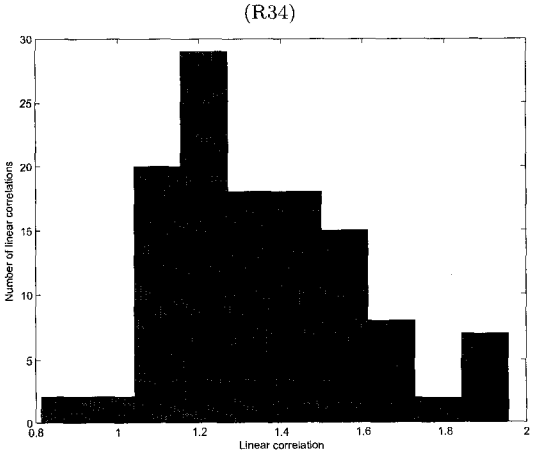
(R33)



(R34)



(R35)



(R36)

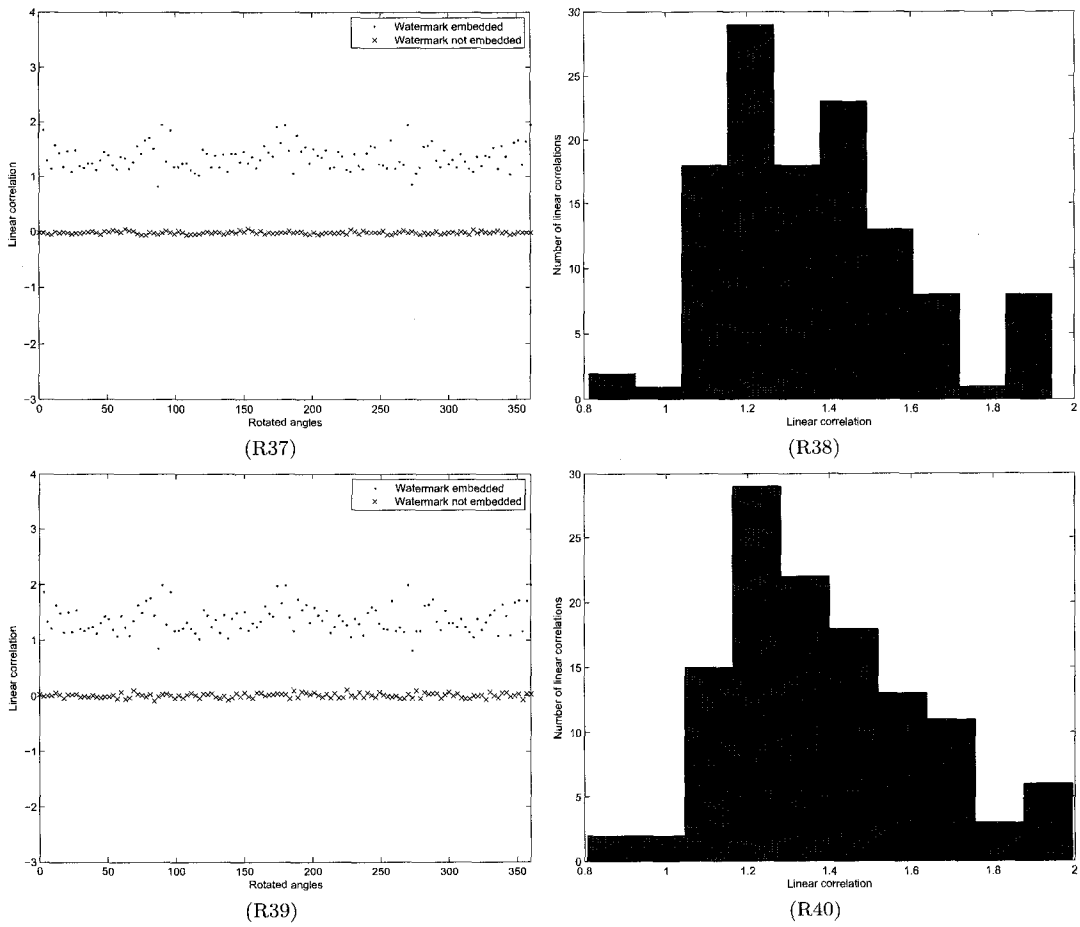
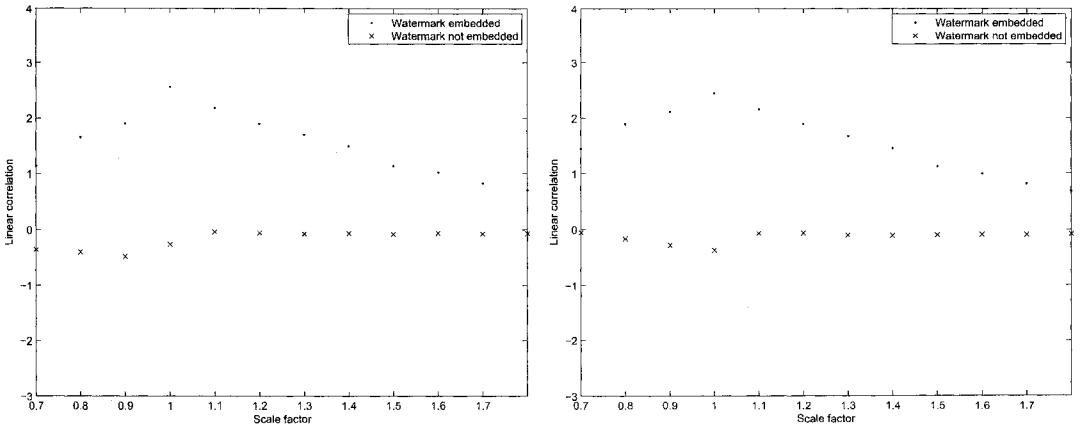


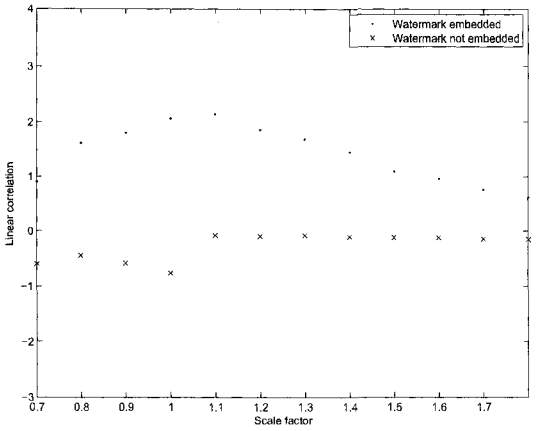
Figure 7.16: Results under rotation for the first 20 images shown in Fig. 7.15.

7.6.2 Scaling

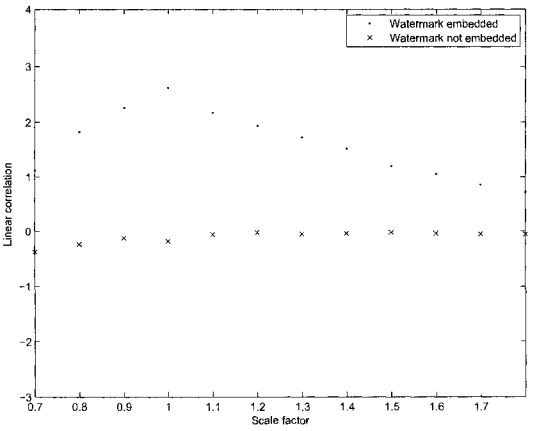
We tested our algorithm under scaling distortion with scale factor varying from 0.7 to 1.8 with a step of 0.1. The results are shown in Fig. 7.17. The 20 sub-figures $S1, S2, S3, \cdots, S20$ in Fig. 7.17 are the experimental results for image 1 to 20 respectively. For the 100 image test cases, the false positive probability error and the false negative probability error are all 0 which means all the tests of scaling are successful. So the proposed watermarking scheme also shows good performance of the robustness against scaling. Also it is shown that linear correlation values become smaller as the images shrink or enlarge with a larger ratio. Although this will deteriorate the linear correlation based detection, the proposed scheme still handles all the test cases of scaling very well and shows good performance of robustness against scaling.



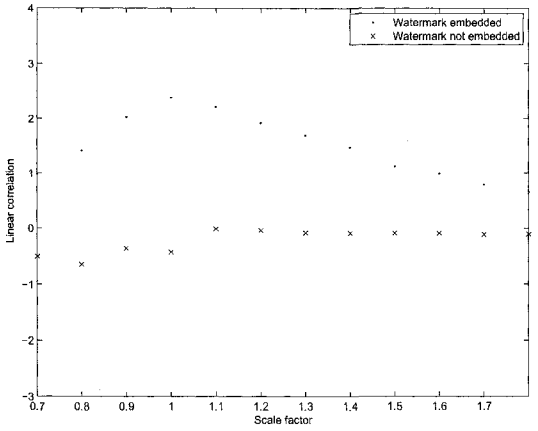
(S1) (S2)



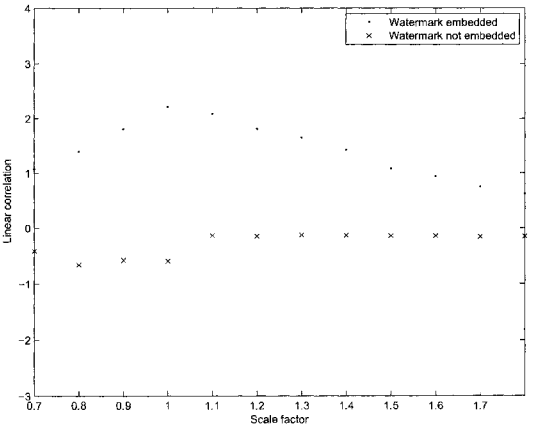
(S3)



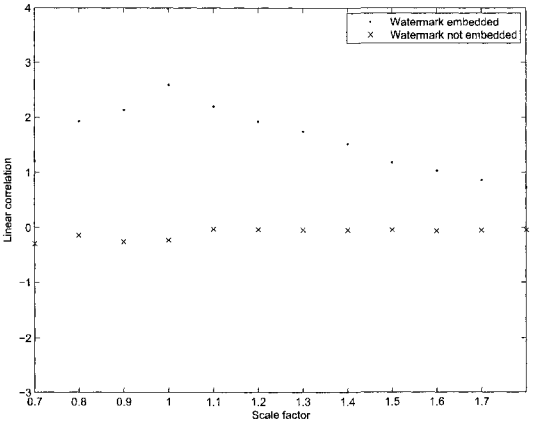
(S4)



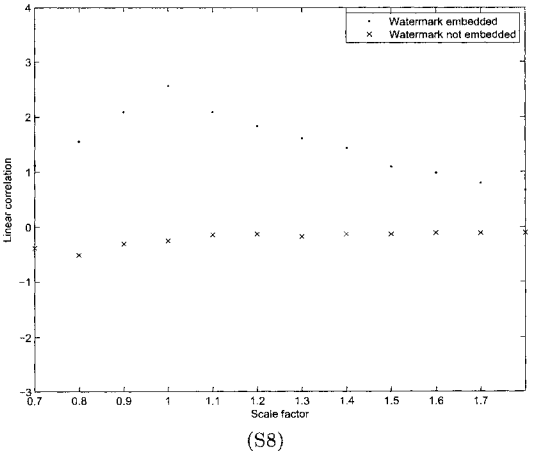
(S5)



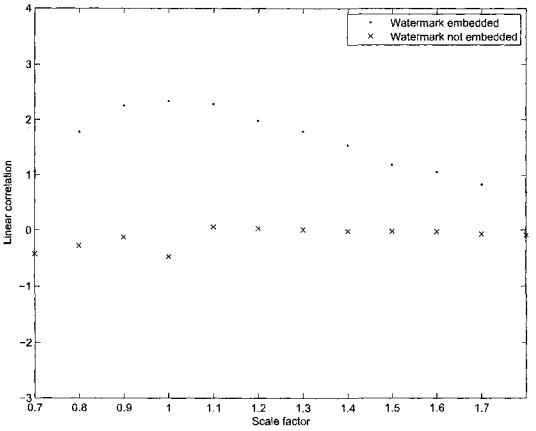
(S6)



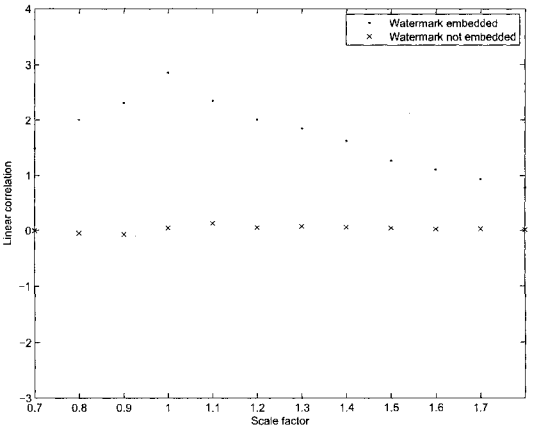
(S7)



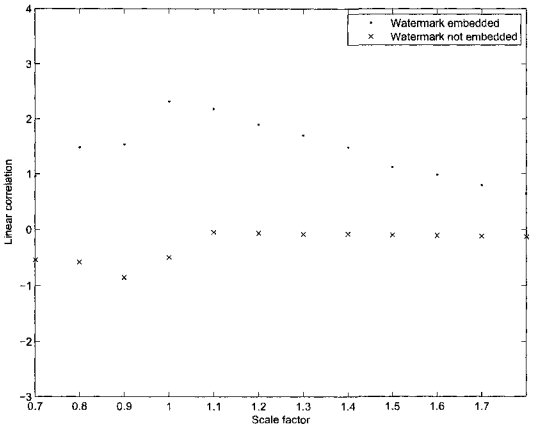
(S8)



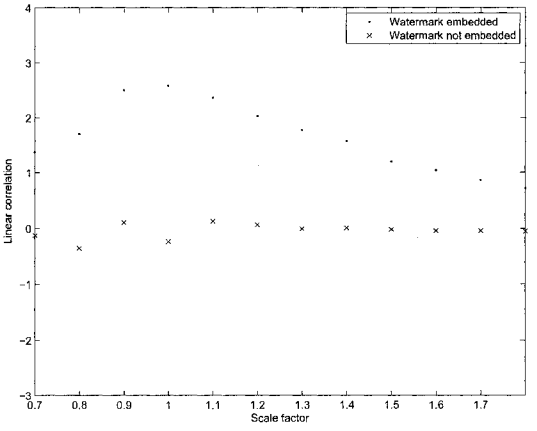
(S9)



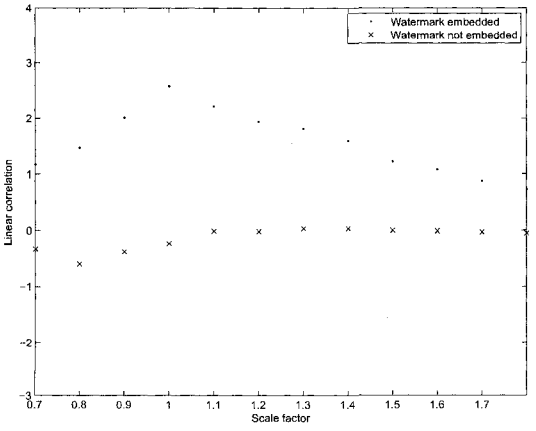
(S10)



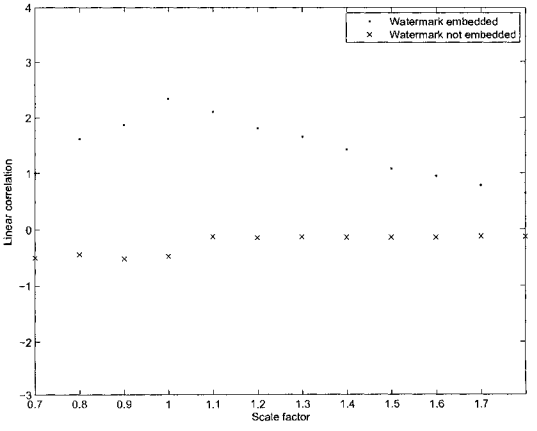
(S11)



(S12)



(S13)



(S14)

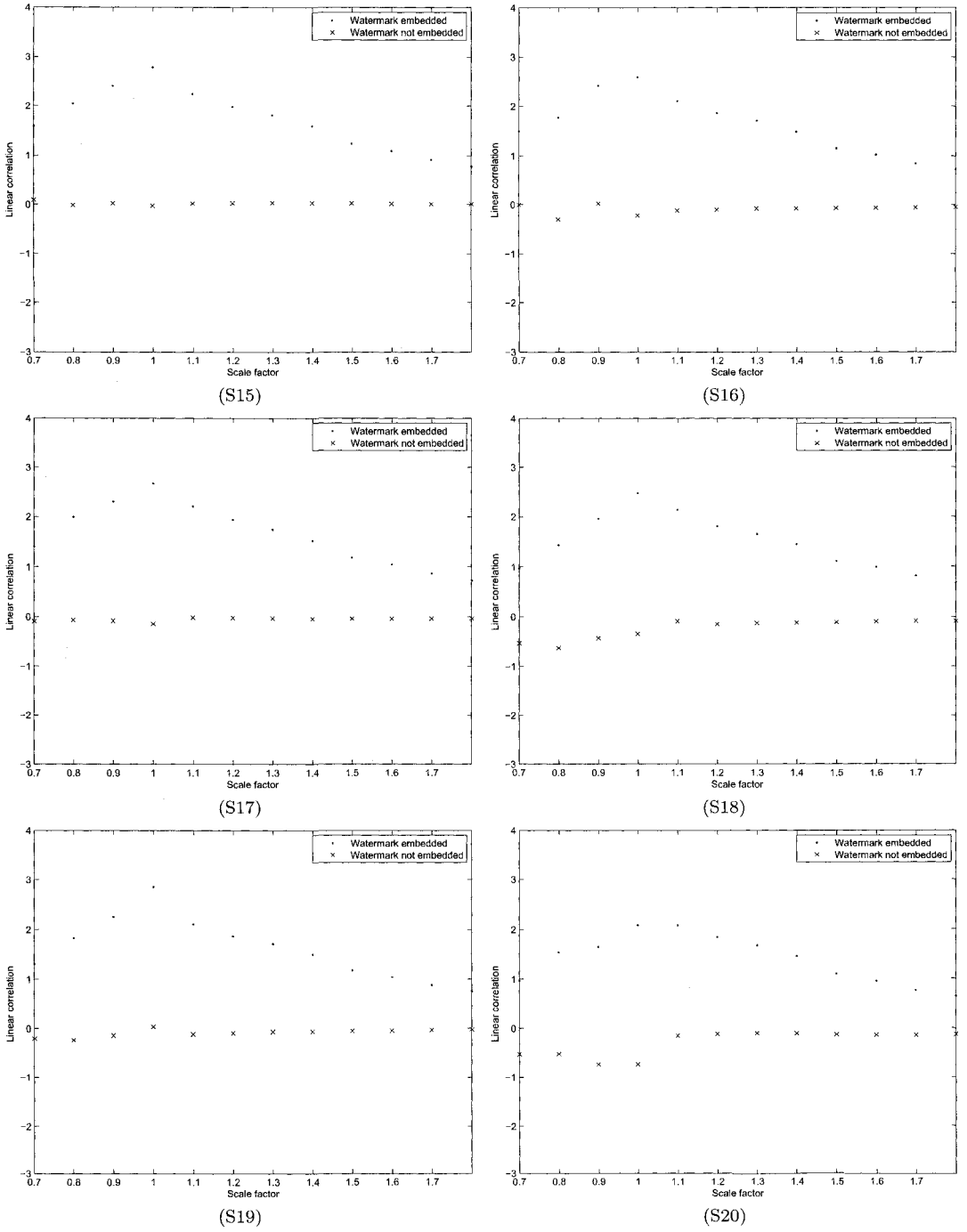
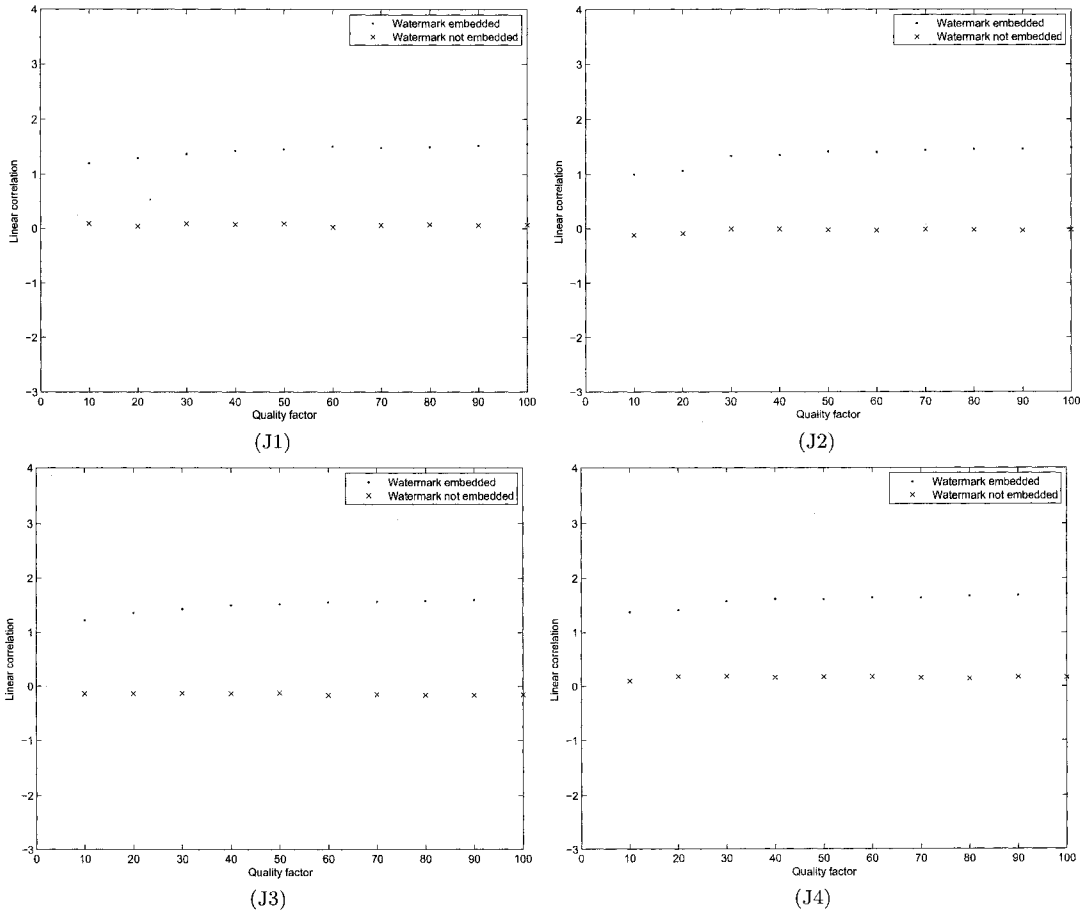


Figure 7.17: Results under scaling for the first 20 image used in the thesis.

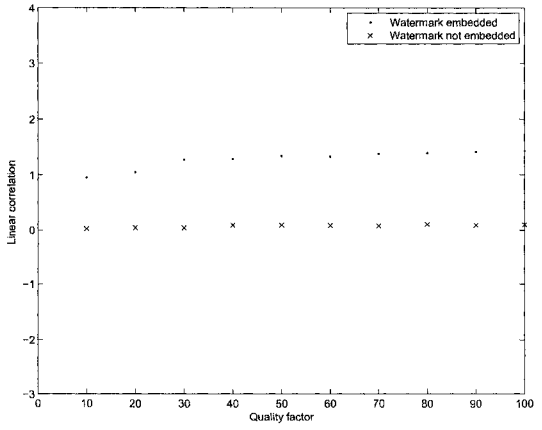
7.6.3 JPEG compression

We also tested our algorithm under JPEG compression with quality factor varying from 100 to 10 with a step of -10. The results for the first 20 images $J1, J2, J3, \cdots, J20$ are shown in Fig. 7.18. When the quality factor becomes smaller, more compression is introduced which leads to smaller value of linear correlation. However the linear correlation based watermark detector can still detect the existence of the watermark correctly as shown in Fig. 7.18. All the JPEG compression tests for 100 images are successful and the probability of error is 0.

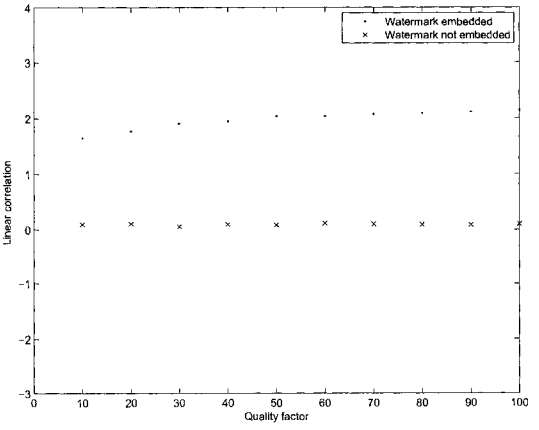


(J1) (J2)

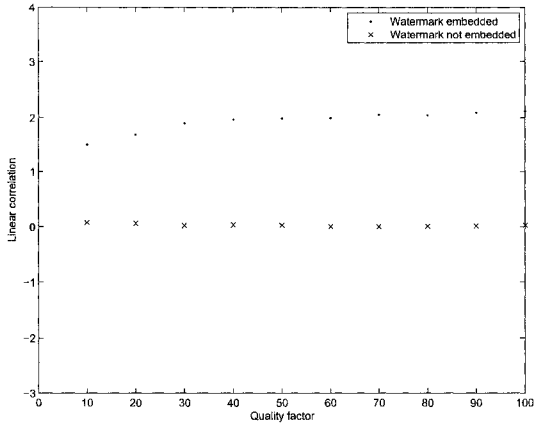
(J3) (J4)



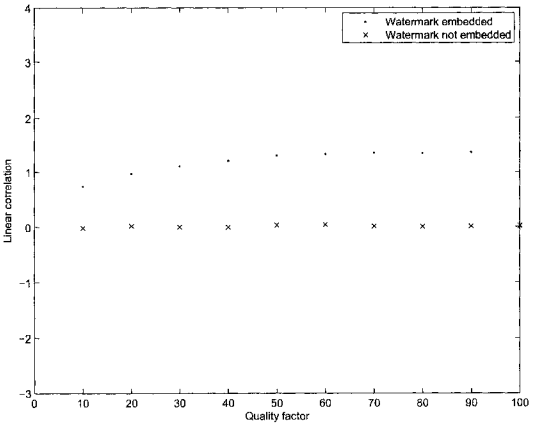
(J5)



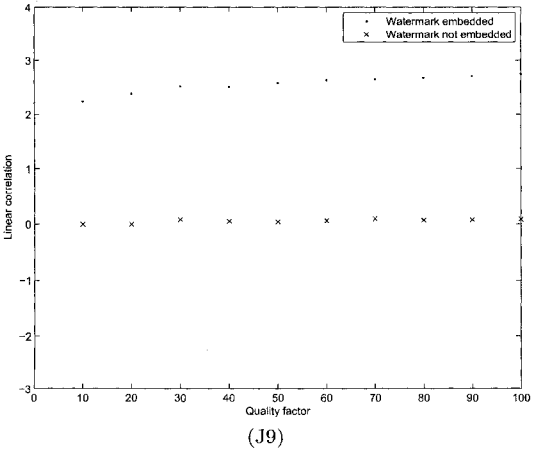
(J6)



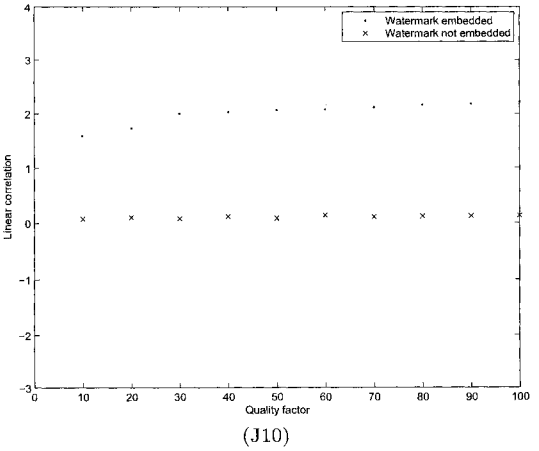
(J7)



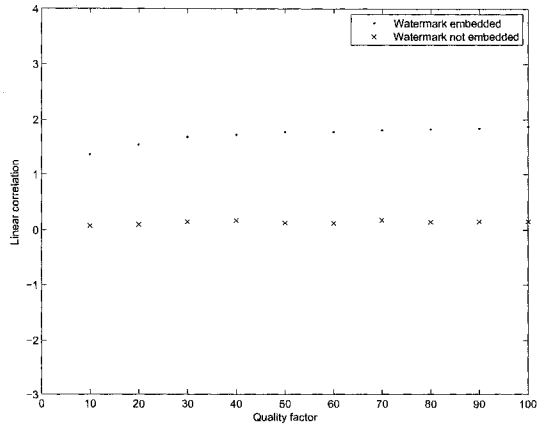
(J8)



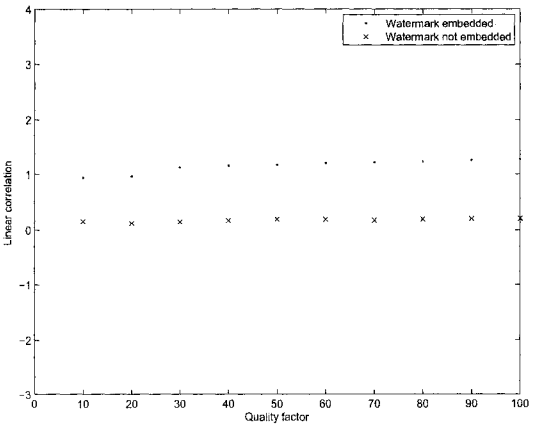
(J9)



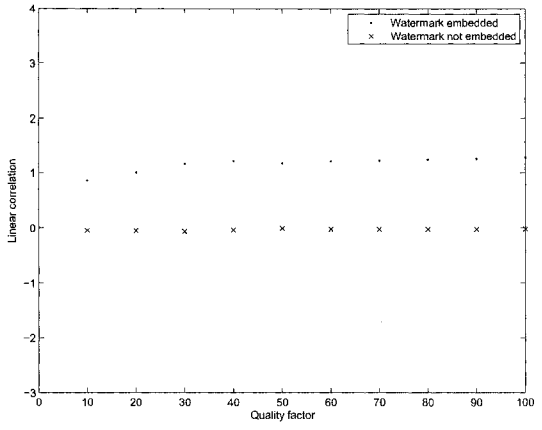
(J10)



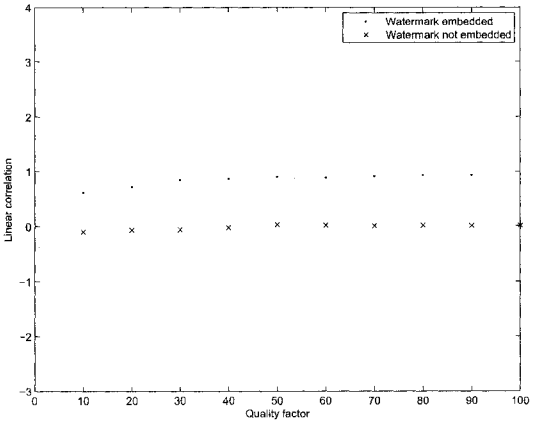
(J11)



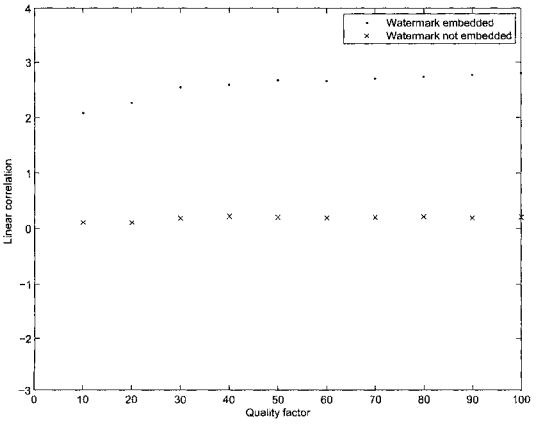
(J12)



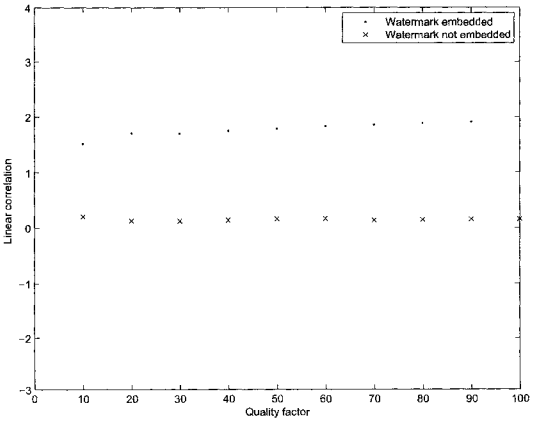
(J13)



(J14)



(J15)



(J16)

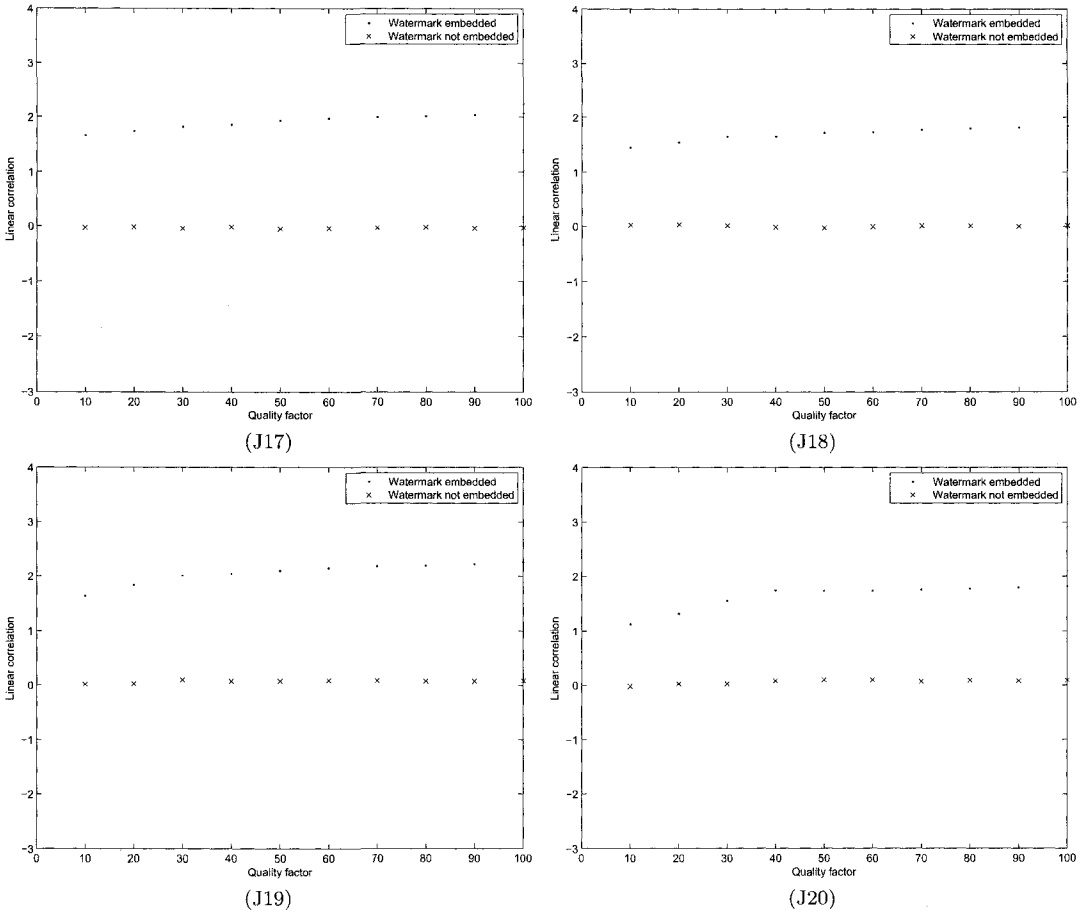
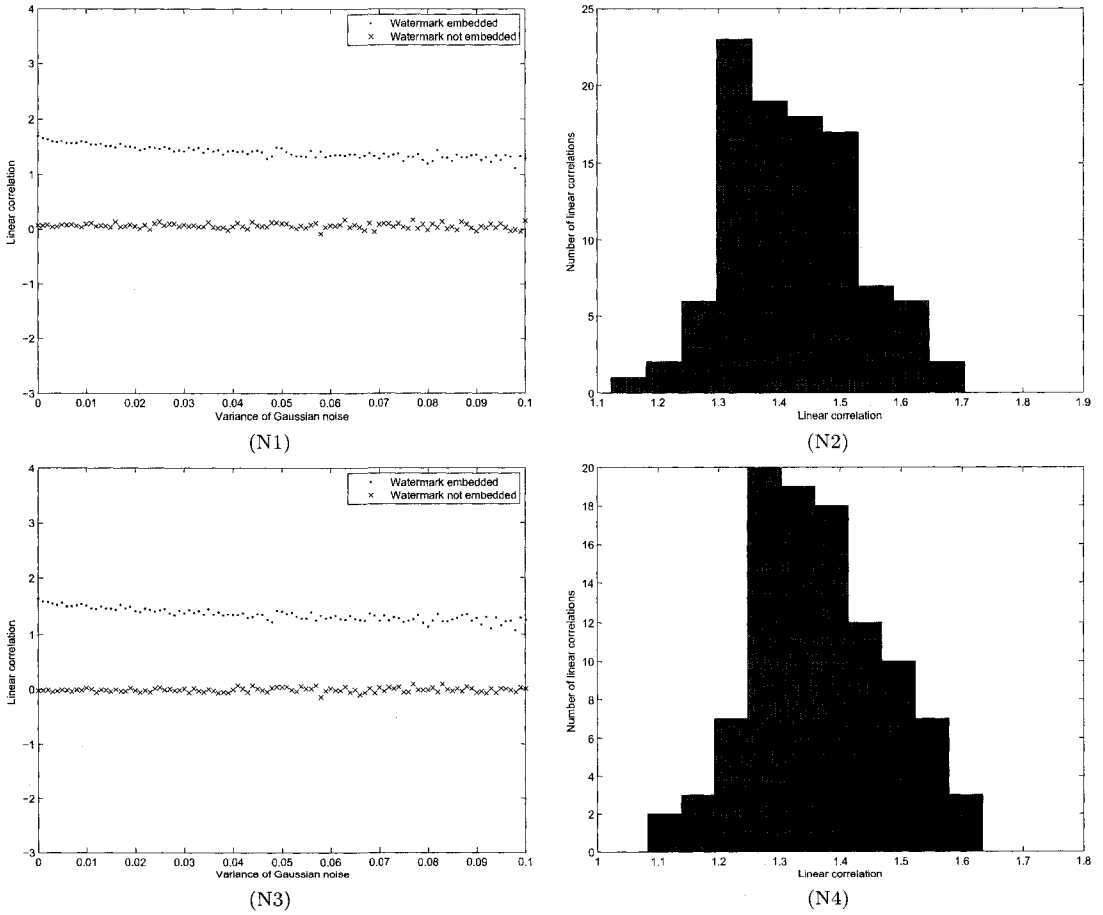


Figure 7.18: Results under JPEG compression for the first 20 images.

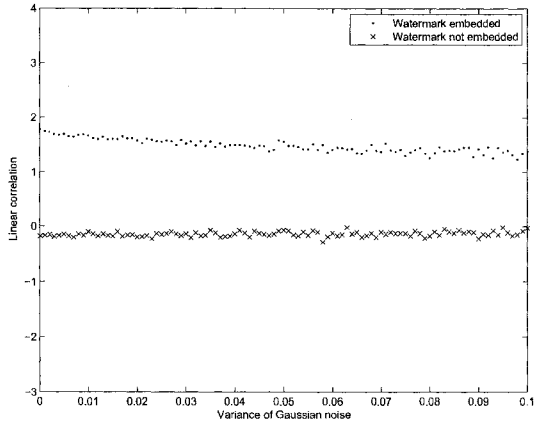
7.6.4 Gaussian noise pollution

We also tested our algorithm under Gaussian noise pollution. The variance of the noise varies from 0.001 to 0.1 with a step of 0.001. So, for each image, the algorithm is tested 100 times. And the results are shown in Fig. 7.19. The odd numbered figures are the linear correlations calculated on image 1 to 20, respectively. And the even numbered figures are the corresponding histograms of the linear correlations. All the noise tests for 100 images are successful as the probability of error is 0.

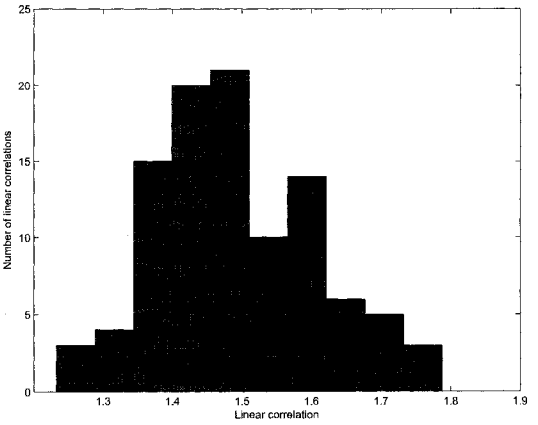


(N1) (N2)

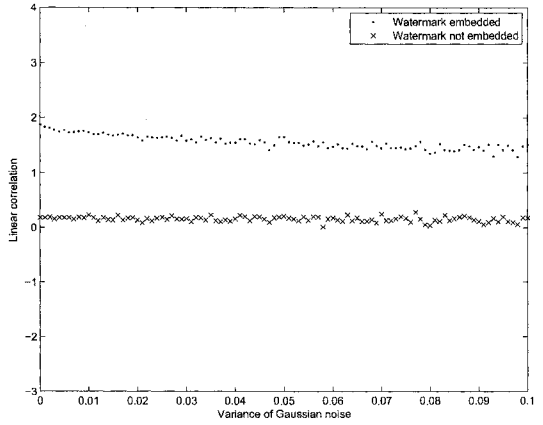
(N3) (N4)



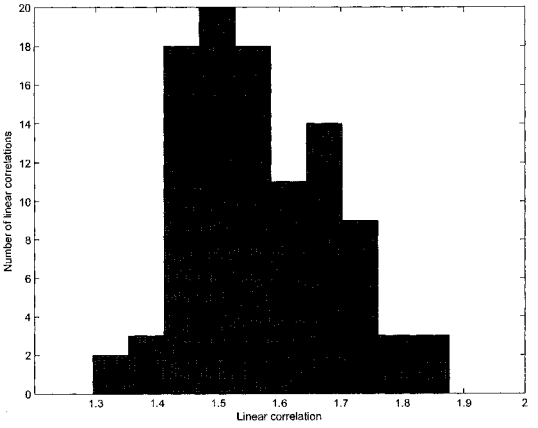
(N5)



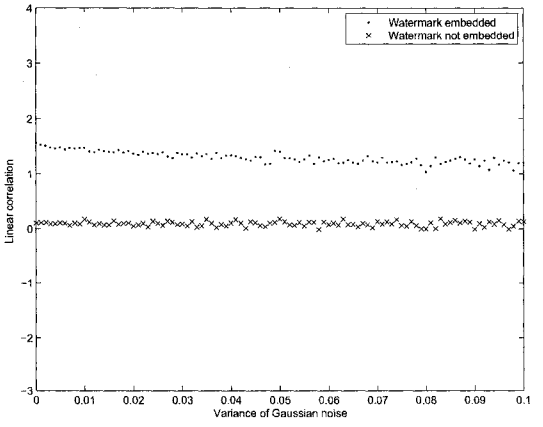
(N6)



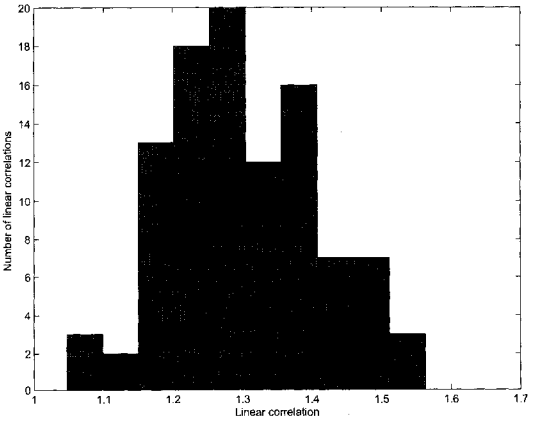
(N7)



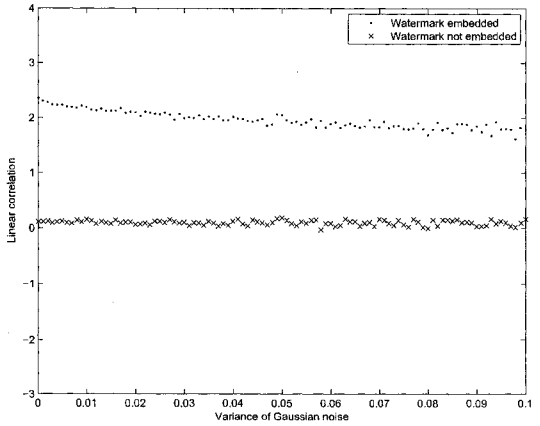
(N8)



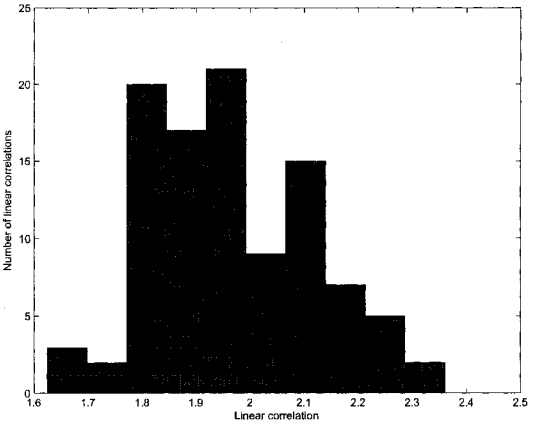
(N9)



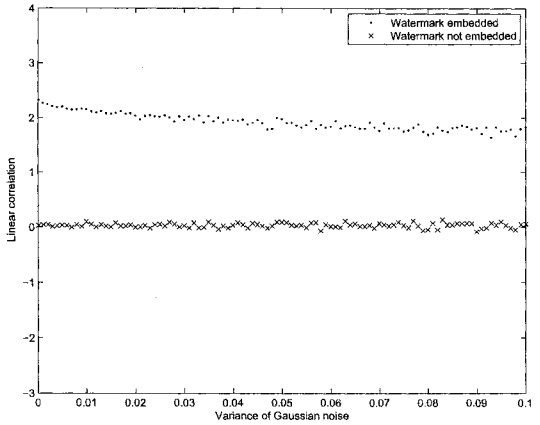
(N10)



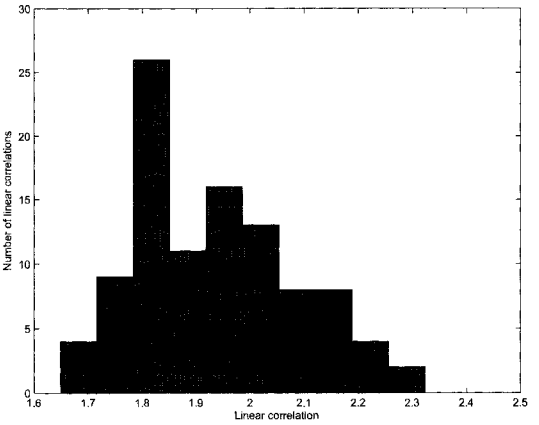
(N11)



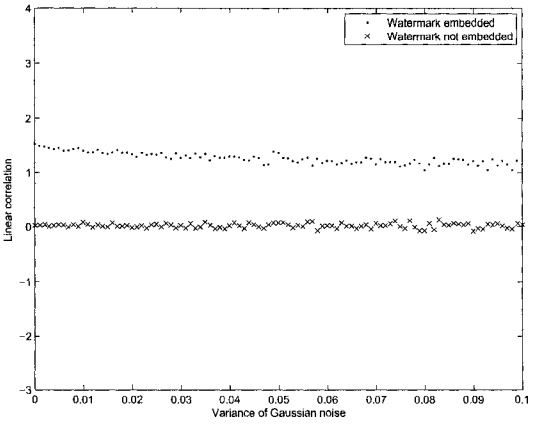
(N12)



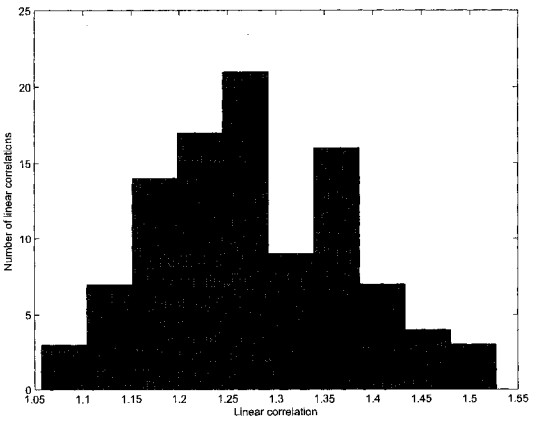
(N13)



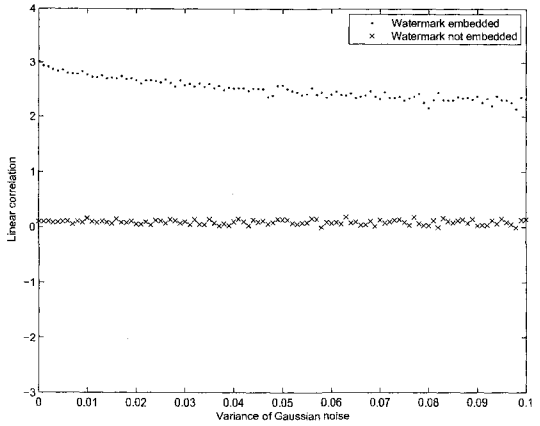
(N14)



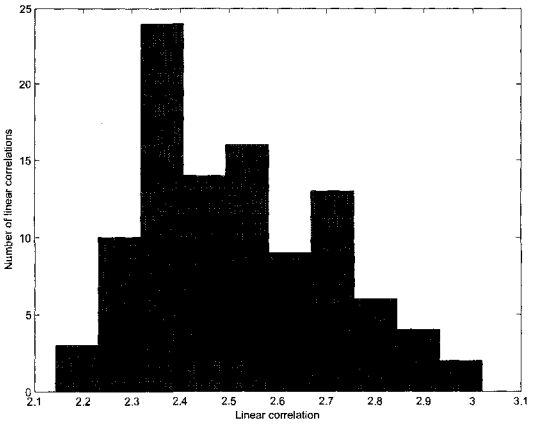
(N15)



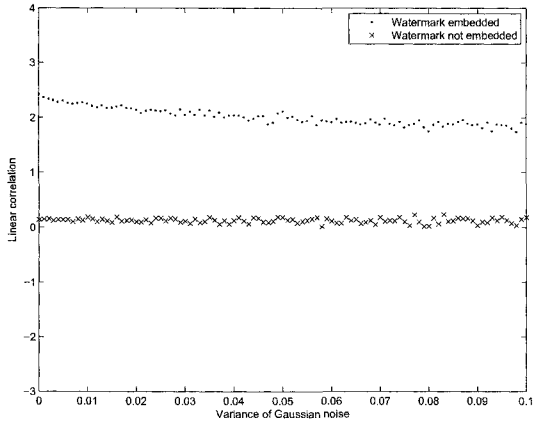
(N16)



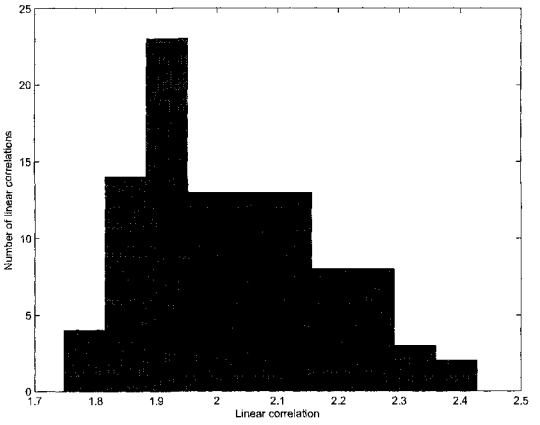
(N17)



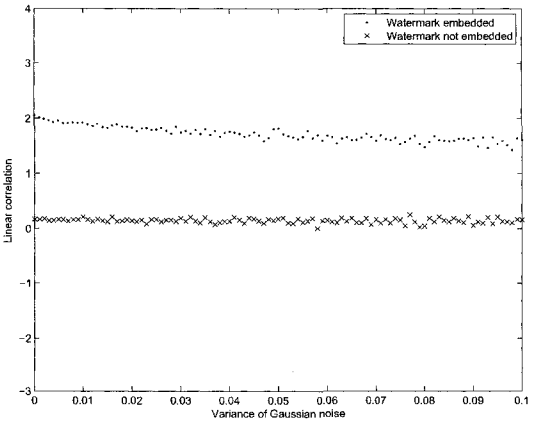
(N18)



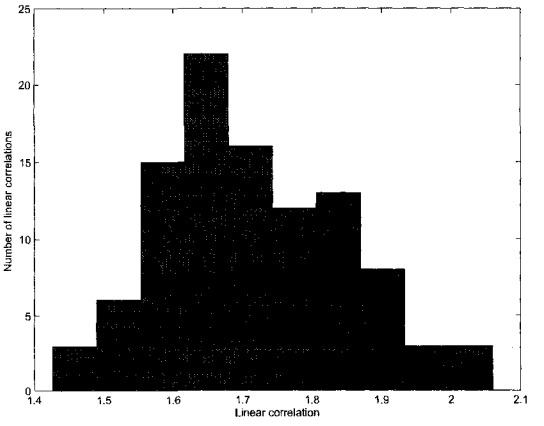
(N19)



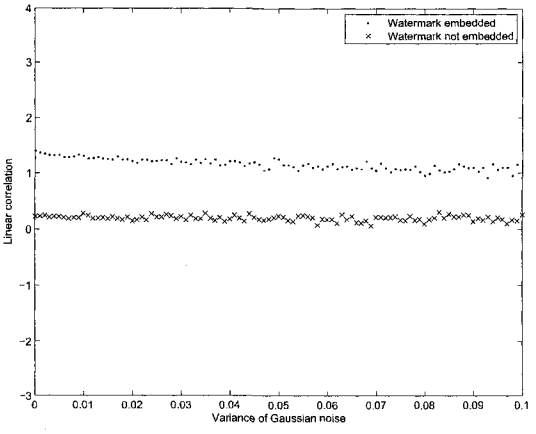
(N20)



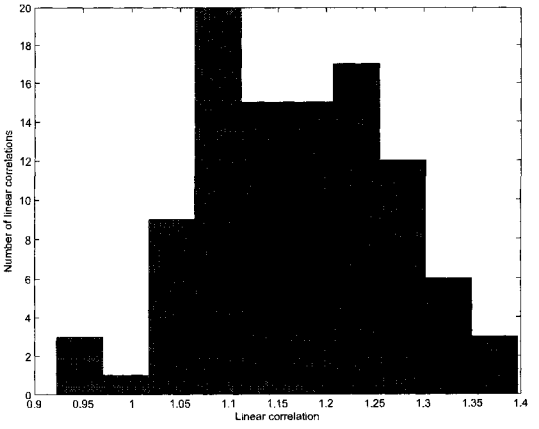
(N21)



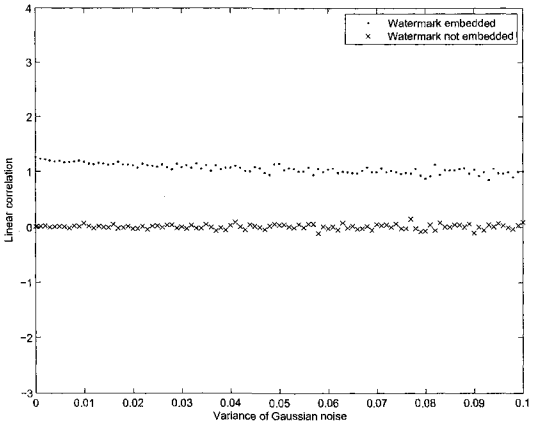
(N22)



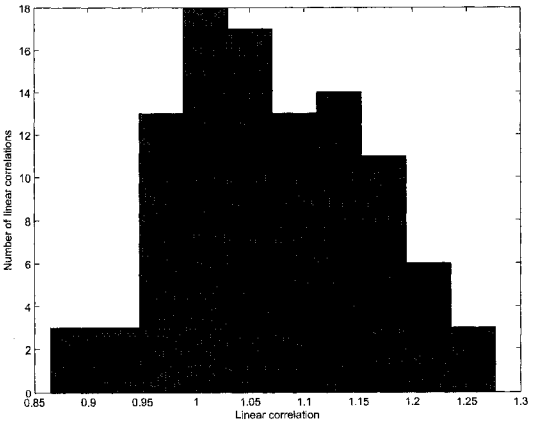
(N23)



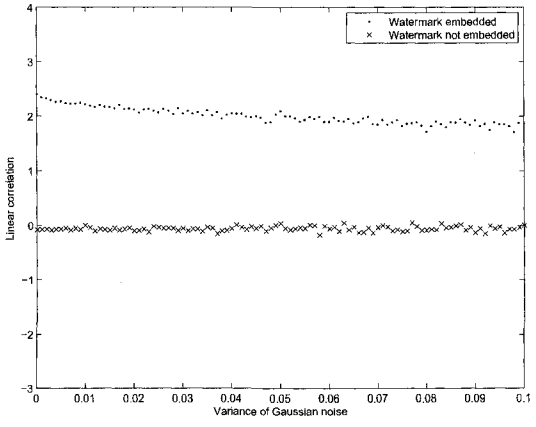
(N24)



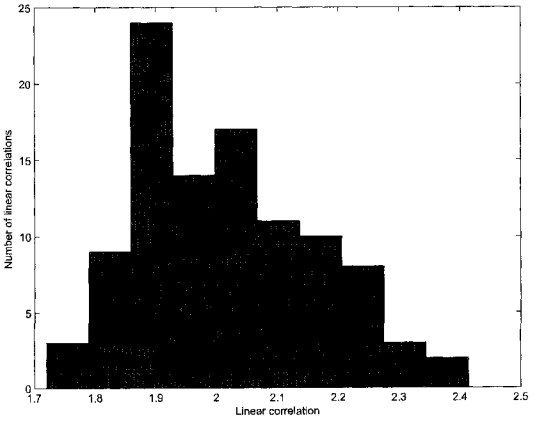
(N25)



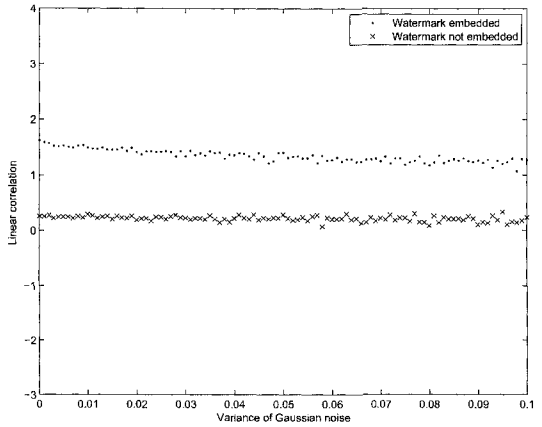
(N26)



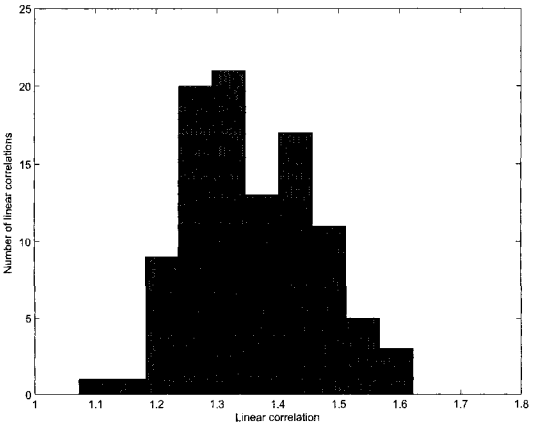
(N27)



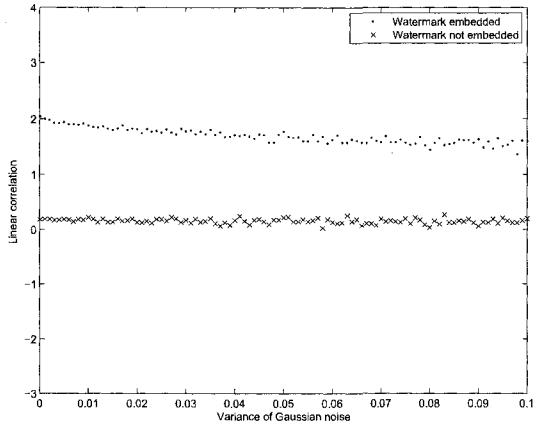
(N28)



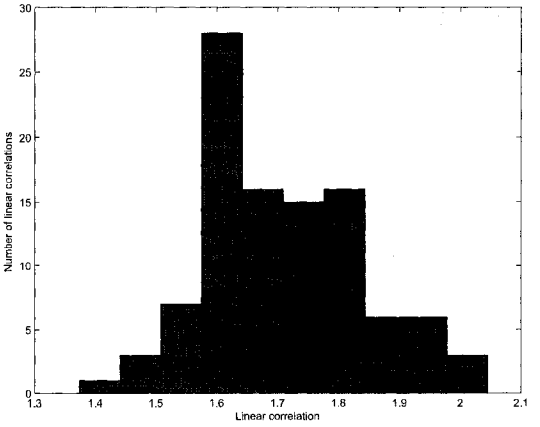
(N29)



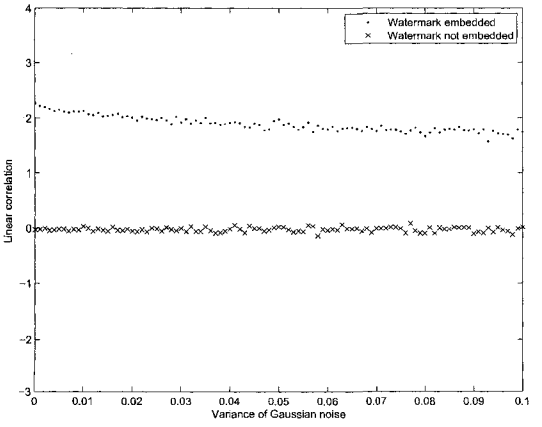
(N30)



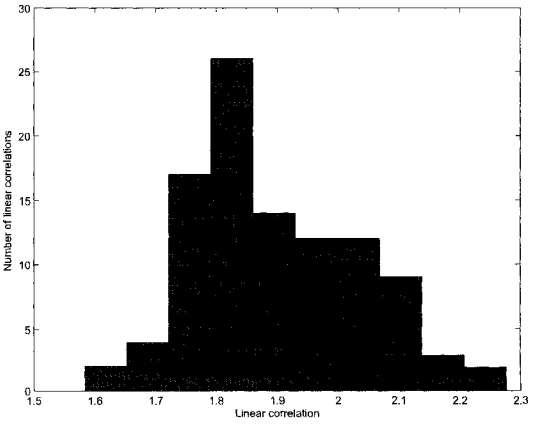
(N31)



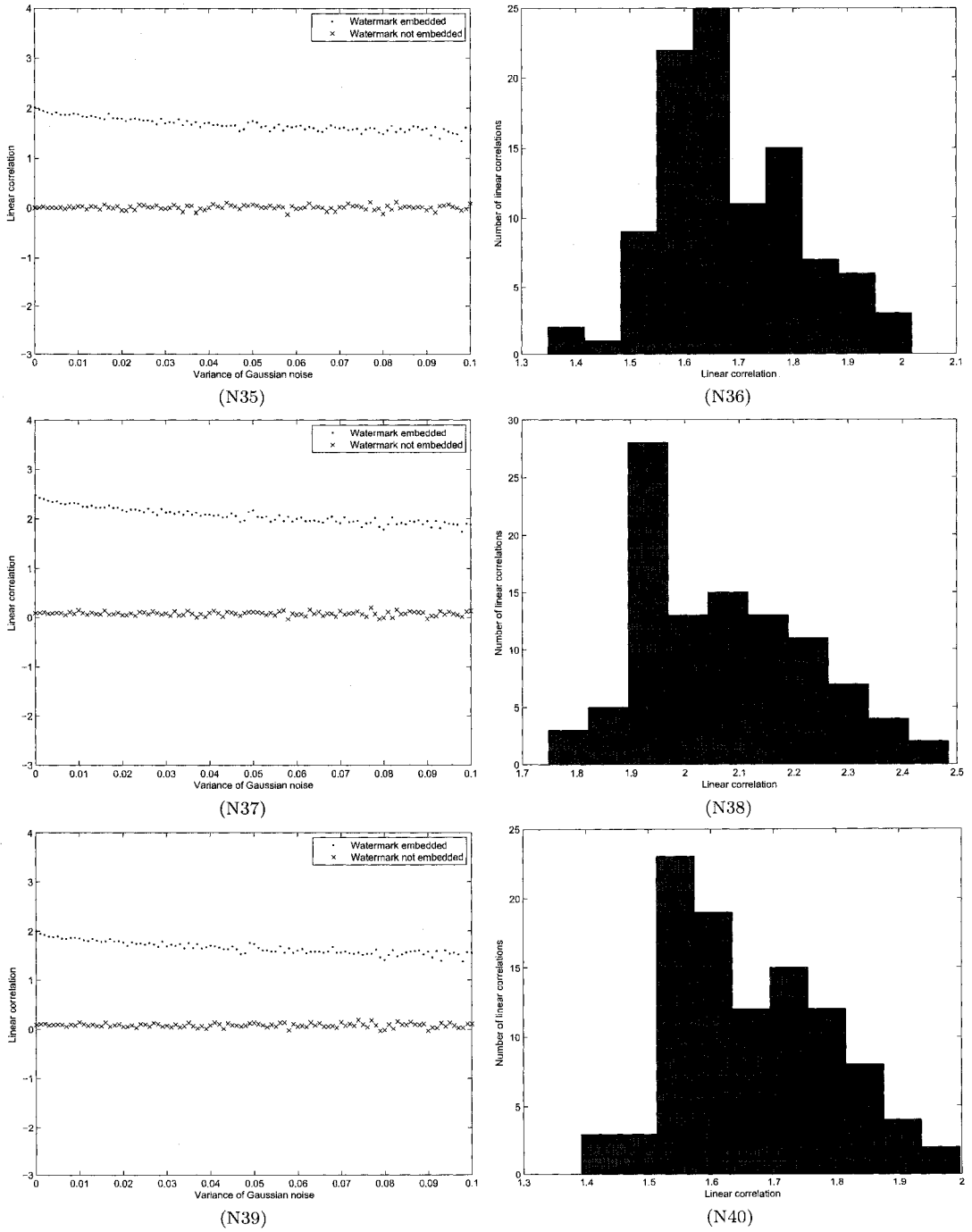
(N32)



(N33)



(N34)



(N35) (N36)

(N37) (N38)

(N39) (N40)

Figure 7.19: Results under noise pollution for the first 20 images.

From the above experimental results, the effectiveness of the algorithm is clearly shown. The proposed watermarking scheme is robust against rotation, scaling, JPEG compression and noise pollution.

7.7 The performance comparison of the watermarking algorithm

As a comparison with the algorithm proposed in [73], the following experiments have been done. The experimental results of [73] is presented in Section 4.4.2.

Tab. 7.2, 7.3, 7.4 and 7.5 show the results, in which the first number shows how many regions have the watermark successfully detected and the second number shows how many regions have the watermark actually embedded. As in Tab. 7.2 and Tab. 7.3, the performance of the scheme against geometric transform is very good. The performance against rotation and scaling is much better than that in [73].

Table 7.2: Rotation

Image	Image1	Image2	Image3
Rotation 10^o & Cropping	8/8	8/8	8/8
Rotation 20^o & Cropping	8/8	8/8	8/8
Rotation 45^o & Cropping	7/8	8/8	7/8

Table 7.3: Scaling

Image	Image1	Image2	Image3
Scaling 0.8	8/8	8/8	8/8
Scaling 1.1	8/8	8/8	8/8
Scaling 1.2	7/8	8/8	7/8

Since those reference feature points are extracted from the segmented image, the reference feature points are robust against noise and filtering operation. Also the spread spectrum is used for watermark embedding. The watermark detection results against noise is very good. Also the performance against JPEG compression is very good. The results are shown in Tab. 7.5.

Table 7.4: Noise pollution

Image	Image1	Image2	Image3
Additive noise (scale = 0.1)	8/8	8/8	8/8
Additive noise (scale = 0.25)	4/8	5/8	6/8

Table 7.5: JPEG compression

Image	Image1	Image2	Image3
JPEG (Quality factor = 70)	8/8	8/8	8/8
JPEG (Quality factor = 30)	4/8	3/8	5/8

7.8 Experimental results for probability of error

In Section 6.8, the theoretical analysis between fidelity (PSNR) and robustness (probability) is given. The relationship between the theoretical results and the experimental results are shown in this section. The experiment setting is: for 100 test images , each image will go through the watermark embedding and detection with the radius R changing from 8 to 25 with step of 1. Also the embedding strength will be adjusted to meet the requirement PSNR after embedding and the threshold is set to be 0.5. The watermark detection results will be calculated and the probability of error will be calculated including false positive probability and false negative probability. The results

are shown in Fig. 7.20 and Fig. 7.21.

It is shown that the experimental results are consistent with the theoretical derivation. In Fig. 7.20 the watermark is embedded with a larger embedding strength than

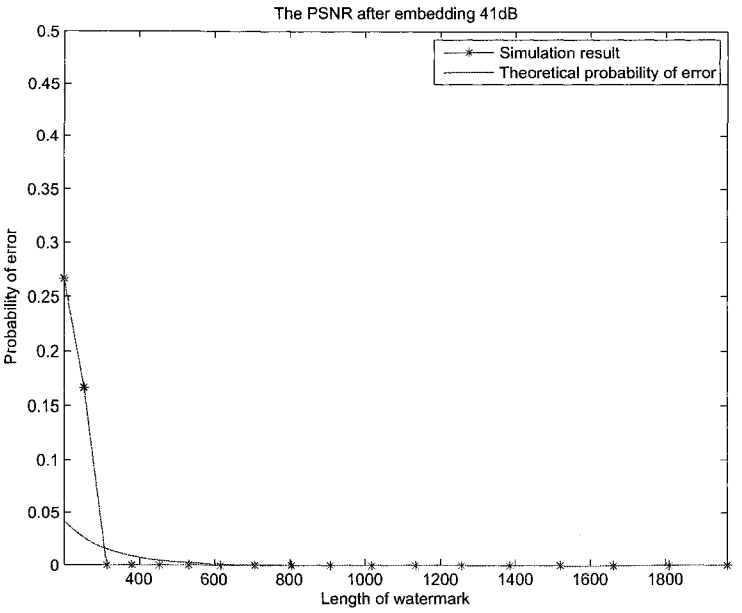


Figure 7.20: The experimental result and theoretical derivation comparison for 41dB.

that of Fig. 7.21, this causes the decrease of PNSR from around 45dB to around 41dB and the increase of the robustness of the watermarking scheme. Also the larger the watermark length, the image and the watermark can be more accurately approximated to the mathematical models from the stochastic perspective, which leads to better performance of the linear correlation detector. When the length is smaller than 200, the probability of error from experiments are larger than the theoretical derivation since the length of watermark and the size of image are too small to meet the stochastic assumption for the linear correlation detector. However by observing the two figures, the

cutoff length of watermark for the probability of errors are rough the same for both the theoretical derivation and the experimental results. Also the change of the probability with regarding to PSNR and the length of watermark are consistent with each other for both the theoretical derivation and the experimental results.

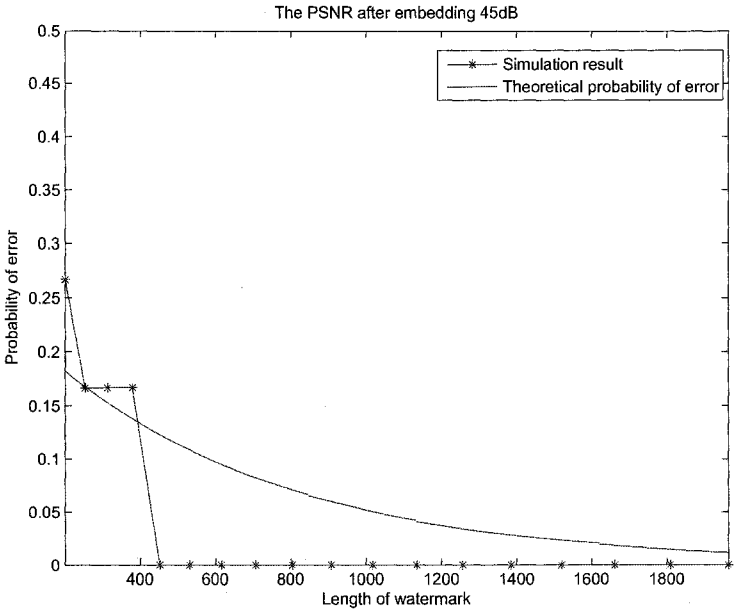


Figure 7.21: The experimental result and theoretical derivation comparison for 45 dB.

In conclusion, the proposed watermarking scheme shows good performance in terms of the robustness against rotation, scaling, JPEG compression and noise pollution. Also the mathematical modelling provides to be very useful to guide the watermark embedding process and analyze the probability of error.

Chapter 8

Conclusions and future work

In this thesis, detailed theoretical analysis is given to the existing RST invariant watermarking algorithms. Combining with experimental results, the advantages and disadvantages of each algorithm are presented in depth. This gives a good review and evaluation to current RST watermarking algorithms. Also, it provides a solid basis for further research in this field. A brief summarization is given as following:

The Fourier-Mellin transform is to transform the image into the RST invariant domain. One problem with this method is that it is difficult to implement. The log-polar mapping (LPM) and inverse log-polar mapping (ILPM) processes use interpolation that causes a degradation and fidelity loss. Therefore, some methods [55] [56] [38] use the log-polar mapping instead of the Fourier-Mellin transform. The LPM can convert the rotation and scaling in the spatial domain to the translation in the LPM domain, which is easy to deal with. One dimensional projection [38] or image registration related techniques [55] [56] have been used to solve the translation in the LPM domain. They all share some similarity in theory and have their own advantages and problems.

Using a template to identify geometrical transforms is a straightforward idea. How-

ever, the template-based watermarking algorithm inserts the template into the image by manually increasing the energy of some points or regions. This makes the template recognizable and removable by the image processes such as compression and geometrical transforms, and also makes it easy to be detected by attackers. Newly proposed template-based watermarking algorithm generates the information bearing template, embeds and detects the watermark based on the stochastic models and analysis, which makes it mathematically convincible.

Using the salient features of an image such as corner points, centroid of the homogeneous regions can serve as the same reference purpose as the template to locate the watermark embedding and detection region. Since these salient features are part of the image, they are better than the template.

Some researches are focusing on the exploitation of the geometrical transform invariance property of the image contents. One method is the Radon transform. The one-dimensional projection, such as Radon transform, can be used to exploit some geometrical transform invariance property. The watermarking algorithm proposed by [42] utilizes both the salient features and the Radon transform, which works quite well. Other content based algorithms decompose the watermark and image into polynomial components. Some of these components are RST invariant, we can either embed the watermark into these components or use the component as the matching filter. Stochastic analysis is widely used in the error probability analysis. Now the stochastic analysis can be used to get the RST invariant content of the image such as moments, image normalization, and bispectrum.

In the RST invariant watermarking algorithms mentioned above, the lack of mathematical model for image makes it difficult to analyze the watermarking processes. So, in this thesis, the mixture Gaussian model is used to model the image and the MAP im-

age segmentation is used to segment the image into homogeneous regions. Each region can be represented as a generalized Gaussian distribution with parameters estimated using EM algorithm. The SIFT (Scale Invariant Feature) feature extraction algorithm locates the salient feature points which work as the reference points and are used to define the circular regions for watermark embedding and extraction. Major part of each embedding or extraction region belongs to one segmented region. Thus, the characteristic within each embedding or extraction region is uniform. The image modeling provides a better guidance to adjust adaptively the watermark embedding strength together with NVF (noise visibility function). Meanwhile, the image normalization and SIFT (Scale Invariant Feature) feature extraction is used to achieve the RST invariance. The spread spectrum and linear correlation are used for watermark embedding and extraction. The experimental results show that the proposed algorithm performs well against RST transform and other attacks such as noise pollution and compression. Also the mixture Gaussian model provides an accurate mathematical model.

For the further work, several possible approaches could be taken. The proposed algorithm works in spatial domain, however the image modelling could be well exploited in transform domain. The image in frequency domain can be separated into different frequency spectrums and the mixture Gaussian distribution can also apply. Then a lot of watermarking algorithms in frequency domain can benefit from the established image model. This will also be very helpful for video watermarking since the video encoding normally happened in transform domain. Also the mathematical model can be further improved and refined, for example, the non-stationary Gaussian distribution and Markov random field can be applied and the unsupervised image segmentation and clustering techniques can be used to segment images. Currently the noise visibility function is used to guide the watermark embedding based on the parameters of

the mixture Gaussian distribution. The capacity of the watermarking algorithm can be improved if some other perceptual models can be introduced. Contrast sensitive function and texture sensitive function can be used here as well. The spread spectrum is used for embedding, which provided good performance of robustness. Other coding technique and modulation can also be used, especially after the image is segmented into different regions, which can be treated as the parallel transmission channels structure. By exploiting this property, the capacity and error correction ability of watermarking system should be improved. In all, this is an exciting area, which could lead to many innovative ideas and applications.

Bibliography

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

[2] J. Zhao, R. Hayasaka, R. Muranoi, M. Ito, and Y. Matsushita. A video copyright protection system based on content ID. *IEICE Transactions on Information and Systems*, E83-D(12):2131–2141, 2000.

[3] D. Zheng, J. Zhao, W.J. Tam, and F. Speranza. Image quality measurement by using digital watermarking. In *Proc. of IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications*, pages 65–70, 2003.

[4] J. O'Ruanaidh and T. Pun. Rotation, scale, and translation invariant digital image watermarking. *Signal Processing*, 66(3):303–317, 1998.

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

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

[7] A. Tirkel, R. V. Schyndel G.Rankin, N. Mee W. Ho, and C. Osborne. Electronic watermark. In *Proc. of DICTA*, pages 666–672, 2003.

[8] C. Kurah and J. McHughes. A cautionary note on image downgrading. In *Proc.*

of IEEE Computer Security Applications Conference, volume 2, pages 153–159, 1992.

[9] O. Bruyndonckx, J. J. Quisquater, and B. Macq. Spatial method for copyright labeling of digital images. In *Proc. of IEEE Workshop Nonlinear Signal and Image Processing*, pages 456–459, 1995.

[10] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[11] R.L. Pickholtz, D.L. Schilling, and L.B. Milstein. Theory of spread spectrum communications - a tutorial. *IEEE Transactions on Communications*, 30(5):855–884, 1982.

[12] E. Koch and J. Zhao. Towards robust and hidden image copyright labeling. In *Proc. of IEEE Workshop on Nonlinear Signal and Image Processing*, pages 452–455, 1995.

[13] A. Bors and I. Pitas. Image watermarking using DCT domian constraints. In *Proc. of IEEE International Conference on Image Processing*, volume 2, pages 231–234, 1996.

[14] J. O'Ruanaidh, W.J. Dowling, and F.M. Boland. Phase watermarking of digital images. In *Proc. of IEEE International Conference on Image Processing*, pages 239–242, 1996.

[15] J. O'Ruanaidh, W.J. Dowling, and F.M. Boland. Watermarking digital images

for copyright protection. In *Proc. of IEEE International Conference on Vision, Image and Signal Processing*, pages 250–256, 1995.

[16] D. Kundur and D. Hatzinakos. A robust digital image watermarking method using wavelet based fusion. In *Proc. of IEEE International Conference on Image Processing*, volume 1, pages 544–547, 1997.

[17] D. Kundur and D. Hatzinakos. Digital watermarking for telltale tamper proofing and authentication. *Proc. of the IEEE*, 87(7):1167–1180, 1999.

[18] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *Proc. of IEEE International Conference on Image Processing ICIP-96*, pages 243–246, 1996.

[19] M. Kutter. Watermarking resisting to translation, rotation and scaling. In *Proc. of the SPIE: Multimedia Systems and Applications*, volume 3528, pages 423 – 431, November, 1998.

[20] M. Kankanhalli, R. Ramakrishman, and Rajmohan. Content based watermarking of images. *Proc. of ACM Multimedia*, pages 61–70, 1998.

[21] F. Bartolini, M. Barni, V. Cappellini, and A. Piva. Mask building for perceptually hiding frequency embedded watermarks. In *Proc. of the 5th IEEE International Conference on Image Processing ICIP'98*, volume 1, pages 450–454, 1998.

[22] A. Reed and B. Hannigan. Adaptive color watermarking. In *Proc. of the SPIE: Security and Watermarking of Multimedia Contents*, volume 4675, pages 222 – 229, April, 2002.

[23] S. Pereira and T. Pun. Robust template matching for affine resistant image watermarks. *IEEE Transactions on Image Processing*, 9(6):1123–1129, 2000.

[24] M. Barni. Effectiveness of exhaustive search and template matching against watermark desynchronization. *IEEE Signal Processing Letters*, 12(2), February, 2005.

[25] M. Alvarez-Rodriguez and F. Perez-Gonzalez. Analysis of pilot-based synchronization algorithms for watermarking of still images. *Signal Processing: Image Communication*, 17(8):611 – 633, September, 2002.

[26] N. Merhav. An information-theoretic view of watermark embedding-detection and geometric attacks. *WaCha 05, Barcelona*, 2005.

[27] X. Dai and S. Khorram. A feature-based image registration algorithm using improved chain-code representation combined with invariant moments. *IEEE Transactions on Geoscience and Remote Sensing*, 37(5):2351–2362, 1999.

[28] M. Alghoniemy and A. H. Tewfik. Geometric invariance in image watermarking. *IEEE Transactions on Image Processing*, 13(2):145–153, 2004.

[29] C. W. Tang and H.M. Hang. A feture-based robust digital image watermarking scheme. *IEEE Tranactions on Signal Processing*, 51(4), April 2003.

[30] D. Simitopoulos, D. E. Koutsonanos, and M. G. Strintzis. Image watermarking resistant to geometric attacks using Generalized Radon transformations. In *Proc. of DSP 2002*, volume 1, pages 85–88, 2002.

[31] P. Dong and N. P. Galatsanos. Affine transform resistant watermarking based on image normalization. In *Proc. of IEEE International Conference Image Processing*, pages 489–492, 2002.

[32] M. Farzam and S. Shirani. A robust multimedia watermarking technique using Zernike transform. In *Proc. of IEEE international Workshop Multimedia Signal Processing*, pages 529 – 534, 2001.

[33] D. Zheng and J. Zhao. A rotation invariant feature and image normalization based image watermarking algorithm. In *IEEE International Conference on Multimedia & Expo (ICME 2007)*, pages 2098–2101, July 2-5, 2007.

[34] D.G.Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

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

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

[37] J. Horner and P. Gianino. Phase-only matched filter. *Applied Optics*, 23(6):812–816, 1984.

[38] 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*, 10(5), 2001.

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

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

[41] H. Kim, Y. Baek, H.K. Lee, and Y.H. Suh. Watermark using Radon transform and bispectrum invariants. In *IH 2002, LNCS 2578*, pages 145–159, 2003.

[42] D. Simitopoulos, D. E. Koutsonanos, and M. G. Strintzis. Robust image watermarking based on Generalized Radon transformations. *IEEE Transcations on Circuits and System for Video technology*, 13(8):732–745, 2003.

[43] Y. N. Hsu and H. H. Arsenault. Optical pattern recognition using circular harmonic expansion. *Applied Optics*, 21(22):4012–4015, 1982.

[44] J. P. Yao and G. Lebreton. Scale-invariant correlation with truncated phase-only radial harmonic fitlers. *Optics Communication*, 145(1):213–219, 1998.

[45] J. P. 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 Communication*, 162(1):26–30, 1998.

[46] H. Kim and B.V.K. Vijaya Kumar. Rotation-tolerant watermark detection using circular harmonic function correlation filter. *Digital Watermarking, LNCS 2939*, pages 263–276, 2004.

[47] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8(8):1409–1420, 1962.

[48] L. G. Brwon. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[49] P. Refregier. Optimal trade-off filters for noise robustness, sharpness of the correlation peak, and horner efficiency. *Optics Letters*, 16(11):829–831, 1991.

[50] D. Zheng and J. Zhao. RST invariant digital image watermarking: importance of phase information. In *Proc. of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 785–788, 2003.

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

[52] C. Harris and M. Stephens. A combined edge and corner detector. In *Proc. of the 4th Alvey Vision Conference*, pages 189–192, 1988.

[53] D. Zheng, Y. Liu, J. Zhao, and A. E. Saddik. A survey of rst invariant image watermarking algorithms. *ACM Computing Surveys*, 39(2):1–91, 2007.

[54] B. Kim, J. Choi, and K. Park. RST-resistant image watermarking using invariant centroid and reordered Fourier-Mellin transform. In *Proc. of IWDW' 2003, LNCS 2939*, pages 370–381, 2004.

[55] D. Zheng, J. Zhao, and A. El. Saddik. RST invariant digital image watermarking based on log-polar mapping and phase correlation. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:753–765, 2003.

[56] Y. Liu, D. Zheng, and J. Zhao. A rectification scheme for RST invariant image watermarking. *IEICE Transactions on Fundamentals, Special Section on Cryptography and Information Security*, E88-A(1):314–318, 2005.

[57] D. Zheng and J. Zhao. LPM-based RST invariant digital image watermarking. In *Proc. of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1951–1954, 2003.

[58] 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*, pages 101–106, 2003.

[59] 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*, pages 202–209, 1997.

[60] G. Depovere, T. Kalker, and J. Linnartz. Improved watermark detection using filtering before correlation. In *Proc. of IEEE International Conference on Image Processing*, volume 1, pages 430–434, 1998.

[61] Y. Liu and J. Zhao. A rectification scheme for RST invariant image watermarking. In *Proc. of Canadian Conference on Electrical and Computer Engineering*, pages 527–530, 2004.

[62] 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*, pages 225–232, 2004.

[63] P. Moulin. A mathematical approach to watermarking and data hiding. In *ICASSP Tutorial*, 2002.

[64] A. Herrigel, J. J. K. O'Ruanaidh, H. Petersen, S. Pereira, and T. Pun. Secure copyright protection techniques for digital images. *Proc. of International Workshop on Information Hiding*, 1525(2):169–190, 1998.

[65] F. Deguillaume, G. Csurka, J. J. K. O'Ruanaidh, and T. Pun. Robust 3D DFT

video watermarking. In *Proc. of IS&T/SPIE Electronic Imaging 99*, volume 3657, pages 113–124, 1999.

[66] S. Pereira, J.J.K. 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*, volume 1, pages 870–874, 1999.

[67] G. Csurka, F. Deguillaume, J. J. K. O'Ruanaidh, and T. Pun. A bayesian approach to affine transformation resistant image and video watermarking. In *Proc. of Int. Workshop on Information Hiding*, volume 2, pages 315–330, 1999.

[68] R. Caldelli, M. Barni, and A. Piva. Geometric-invariant robust watermarking through constellation matching in the frequency domain. In *Proc. of IEEE International Conference on Image Processing*, volume 2, pages 65–68, 2000.

[69] S. Voloshynovskiy, F. Deguillaume, and T. Pun. Multibit digital watermarking robust against local nonlinear geometrical distortions. In *IEEE International Conference on Image Processing, ICIP 2001*, pages 999–1002, 2001.

[70] S. Voloshynovskiy, F. Deguillaume, and T. Pun. Content adaptive watermarking based on a stochastic multiresolution image modeling. In *EUSIPCO2000, X Europian Signal Processing Conference*, September 4-8, 2000.

[71] P. Bas, J.M. Chassery, and B. Macq. Geometrically invariant watermarking using feature points. *IEEE Transactions on Image Processing*, 11(9):1014–1028, 2002.

[72] Bassem Abdel-Aziz, Jiying Zhao, and Jean-Yves Chouinard. On the limits of sec-

ond generation watermarks. In *Proc. of IEEE Canadian Conference on Computer and Robot Vision (CRV2004)*, pages 209–216, 2004.

[73] C. W. Tang and H. M. Hang. A feature-based robust digital image watermarking scheme. *IEEE Transactions on Signal Processing*, 51(4):950–959, April 2003.

[74] M. Kutter, S.K. Bhattacharjee, and T. Ebrahimi. Towards second generation watermarking schemes. In *Proc of IEEE International Conference on Image Processing '99*, volume I, pages 320–323, 1999.

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

[76] Q. Sun, J. Wu, and R. Deng. Recovering modified watermarked image with reference to original image. In *Proc. of SPIE*, volume 3657, pages 415–424, 1999.

[77] Masoud Alghoniemy and Ahmed H. Tewfik. Geometric distortion correction in image watermarking. In *Proc. of SPIE: Security and Watermarking of Multimedia Contents*, volume 3971, pages 82–89, 2000.

[78] 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. of SPIE*, pages 176–186, 2000.

[79] C.H. Teh and R. T. Chin. On image analysis by the method of moments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.

[80] B. V. 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*, 9(6):1025–1034, 2000.

[81] M. Maes, T. Kalker, J.P. Linnartz, J. Talstra, G. Depovere, and J. Haitsma. Digital watermarking for DVD video copy protection. *IEEE Signal Processing Magazine*, 17(5):47–57, 2000.

[82] Y. Xin, S. Liao, and M. Pawlak. Geometrically robust image watermarking via Pseudo-Zernike momens. In *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, volume 2, pages 939–942, 2004.

[83] V. Chandran, B. Carswell, B.Boashash, and Stephen L. Elgar. Pattern recognition using invariants defined from higher order spectra: 2-d image inputs. *IEEE Transactions on Image Processing*, 6(5):703–712, 1997.

[84] H. Kim, Y. Baek, and H.K. Lee. Rotation, scale and translation invariant image watermark using higher order spectra. *Optica Engineering*, 42(2):340–349, 2003.

[85] B. Sadler. Shift and rotation invariant object recognition using the bispectrum. In *Proc. of Workshop on Higher Order Spectral Analysis*, pages 106–111, 1989.

[86] V. Chandran and S. L. Elgar. Pattern recognition using invariants defined from higher order spectra-one-dimensional inputs. *IEEE Transactions on Signal Processing*, 41(1):205–213, 1993.

[87] M. Alghoniemy and A. H. Tewfik. Image watermarking by moment invariants. In *Proc. of IEEE International Conference on Image Processing*, volume 2, pages 73–76, 2000.

[88] Y. Abu-Mostafa and D. Psaltis. Recognitive aspects of moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):698–706, 1984.

[89] A. Khotanzad and Y. Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[90] M. Pawlak and Yongqing Xin. Robust image watermarking: an invariant domain approach. In *Proc. of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, volume 2, pages 885–888, 2002.

[91] P. Dong and N. P. Galatsanos. Affine transformation resistant watermarking based on image normalization. In *Proc. of IEEE International Conference on Image Processing*, volume 3, pages 489–492, 2002.

[92] H. S. Kim and H. K. Lee. Invariant image watermark using Zernike moments. *IEEE Transactions on Circuits and System for Video technology*, 13(8):766–775, 2003.

[93] C. T. Hsu, Y. S. Tsai, and C. Y. Li. Video object watermarking by quadratic region decomposition and tessellation. *Images and Recognition*, 9(1), March 2003.

[94] P. Dong, J. G. Brankov, N. Galatsanos, and Y. Yang. Geometric robust watermarking based on a new mesh model correction approach. In *Proc. of IEEE International Conference on Image Processing*, volume 3, pages 493–496, 2002.

[95] A. Herrigel, S. Voloshynovskiy, and Y. Rytsar. The watermark template attack. In *Proc. of the SPIE Security and Watermarking of Multimedia Contents III*, volume 4314, pages 394–405, 2001.

[96] S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgartner, and T. Pun. Generalized watermark attack based on watermark estimation and perceptual remodu-

lation. In *SPIE 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content*, volume 3971, pages 358–370, 2000.

[97] B. K. Natarajan. Filtering random noise from deterministic signals via data compression. *IEEE Transactions on Signal Processing*, 43(11):2595 – 2605, 1995.

[98] M. Kutter, S. Voloshynovskiy, and A. Herrigel. The watermark copy attack. In *Electronic Imaging 2000, Security and Watermarking of Multimedia Content II*, volume 3971, pages 371–380, 2000.

[99] C. S. Lu, H. Y. Liao, and M. Kutter. Denoising and copy attacks resilient watermarking by exploiting prior knowledge at detector. *IEEE Transactions on Image Processing*, 11(3):280 – 292, 2002.

[100] P. Bas, J.M. Chassery, and B. Macq. Robust watermarking based on the warping pre-defined triangular patterns. In *Proc. of SPIE 2000*, pages 99–109, January 2000.

[101] B.V. 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*, 9(6):1025–1034, 2000.

[102] A. Oppenheim and J. Lim. Importance of phase in signals. *Proc. of IEEE*, 69(5):529–541, 1981.

[103] K. Waheed and F.M.Salam. Blind source recovery using an adaptive generalized Gaussian score function. In *The 2002 45th Midwest Symposium on Circuits and Systems, MWSCAS-2002.*, volume 2, pages 418–421, Aug 2002.

[104] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[105] T. Lindeberg. Scale theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.

[106] K. Mikolajczyko and C. Schmid. An affine invariant interest point detector. In *In European Conference on Computer Vision (ECCV)*, pages 128–142, 2002.

[107] S. Voloshynovskiy, A. Herrigel, N. Baumgartner, and T. Pun. A stochastic approach to content adaptive digital image watermarking. In *Proc. of the Third International Workshop on Information Hiding*, volume 1768, pages 211 – 236, 1999.

[108] R. E. Blahut. *Principles and practice of information theory.* ISBN: 0-201-10709-0. Addison-Wesley Longman Publishing Co., Inc., 1987.

[109] Y. Wang and P. Moulin. Steganalysis of block-structured stegotext. In *SPIE: Security and Watermarking of Multimedia Contents*, volume 5306, pages 477–488, 2004.

[110] L. Sendur and I.W. Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Transactions on Signal Processing*, 50(11):2744–2756, November.

[111] J. R. Hernandez, F. Perez-Gonzalez, and M. Amado. DCT-domain image watermarking and generalized Gaussian models. In *In Proc. of the COST No. 254*

Int. Workshop on Intelligent Communications and Multimedia Terminals, pages 23–26, Nov. 1998.

[112] C. A. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Trans. on Image Processing*, 3(2):162–177, 1994.

[113] F.K. Mohamed and R. Abbes. RST robust watermarking schema based on image normalization and DCT decomposition. *Malaysian Journal on Computer Science*, 20(1):77–90, 2007.