Robust Digital Watermarking in Images

by

Yongqing Xin

A dissertation submitted to the Faculty of Graduate Studies
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Department of Electrical and Computer Engineering
The University of Manitoba, Winnipeg, Canada

Robust Digital Watermarking in Images

by

Yongqing Xin

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

of

Doctor of Philosophy

To Qiong, Wenbo, Becky, and Cindy.

Abstract

One of the fundamental and difficult issues in digital watermarking is the robustness of a watermark, in particular the robustness of a multibit watermark. We first formulate a framework for the robustness design of a watermark, which tackles the problem from three different stages of a watermark communication process, i.e., watermark formation, watermark embedding, and watermark extraction. Based on a zero-bit pseudonoise sequence watermark, a multibit watermark can be designed via feature division multiple access, code division multiple access, M-ary modulation, or a combination of them. Compared with a watermark based on feature division or code division techniques, an M-ary modulation based watermark is advantageous because it is capable of carrying more information with the same watermark energy (host distortion).

Conventionally, M-ary modulation has been limited to $M \leq 256$ due to the heavy computation associated with the correlation-based signal detection. However, with the proposed M-ary phase modulation, which is based on the circular shifts of a reference pseudonoise sequence, the amount of computation in watermark detection is drastically reduced. More interestingly, we also provide the design of an extended M-ary phase modulated watermark based on a set of windowed circular shifts of a pseudonoise sequence with length M, which overcomes the restriction on the value of M due to the length of the feature vector.

Watermark robustness against geometric attacks is especially difficult to achieve. We have proposed an invariant approach based on Zernike moments, drawing on their property of magnitude invariance to image rotation and flipping. Based on our finding that some of the Zernike moments computed in the conventional Cartesian way are inherently inaccurate, and hence not rotationally invariant, we design a multibit watermarking scheme by taking advantage of the accurate Zernike moments while avoiding those inaccurate ones. Simulation results show that such a Zernike moment-based watermark has robustness against geometric attacks including image rotation, flipping, scaling, moderate cropping, aspect ratio change, and a number of common image processing manipulations such as lossy compression, noise addition, and filtering.

Furthermore, we formulate a novel approach to accurate computation of Zernike moments in the polar coordinate system. The details of the algorithm are presented, including a polar pixel arrangement scheme, interpolation-based image resampling, and derivation of the corresponding formulas for Zernike moment computation. The effect of interpolation on the accuracy of Zernike moments is analyzed. Simulation results are given which display the advantages of the polar approach. The significantly improved accuracy of Zernike moments yields much better rotational invariance and a considerably lower level of image reconstruction error. These two factors are crucial in the successful design of the polar Zernike moment-based data hiding.

Acknowledgements

The completion of this thesis involved many factors, among which the following ones are crucial and must be mentioned.

As my academic supervisor, Dr. Pawlak has offered me a lot, from locating the financial support, determining the thesis topic, numerous constructive discussions squeezed in his always overcrowded schedule, all the way to the careful examination and correction of every paper and thesis draft, and so on. Many scenes over the years will remain in my memory forever: his unexpectedly humorous remarks, his impromptu deriving formulas skillfully, his amazingly accurate recall of old references in the middle of talks, his fervent passion for mountain skiing, the exotic wines and laughters in the year-end parties at his house, just to name a few.

Dr. Liao's priceless support and enlightening discussions are highly appreciated. He has been a role model in various aspects. As a respectable professor, he sets an example with his devotion to career and sense of responsibility. As an amiable friend, he shows continuous concern about my study and life, and offers timely help from time to time. His constant encouragement and valuable advice are indispensable. His integrity and frankness are worthy of admiration.

Sincere thanks go to Dr. Thavaneswaran and Dr. Thomas, for their work as members of my advisory committee, especially the examination of my candidacy report and the thesis manuscript. Their careful and detailed corrections and constructive suggestions have considerably improved the quality of this thesis.

It is very kind of Dr. Andrzej Tarczynski to agree to be the external examiner. His detailed comments are insightful. I would like to thank him for all the efforts he has made for the review of this thesis.

I would also like to express my gratitude to all my friends. In particular, Jing Yang offered a great deal of help, especially during my first year in Winnipeg. Without his weekly rides to Superstore, I couldn't have survived both hunger and frostbite!

Special thanks are extended for the financial support I received over the years, which included TRLabs Fellowship, University of Manitoba Graduate Fellowship, Dr. Liao's NSERC grant (172793-04), and Dr. Pawlak's NSERC grant. Without these kind and generous fundings, my PhD program would not have been possible.

Finally I give my heartfelt thanks to my wife Qiong and my son Wenbo, for their unfailing love and patience. Qiong undertakes most of the family responsibilities, putting her own dreams aside. Wenbo was let down repeatedly because I couldn't play games with him. It is no exaggeration that the completion of this thesis is due largely to their understanding and efforts. Thereby I dedicate this thesis to them.

Contents

List of Figures

List of Tables

List of Acronyms

AWGN	additive white Gaussian noise
BER	bit error rate
CDMA	code division multiple access
DCT	discrete cosine transform
DFT	discrete Fourier transform
dB	decibel
DM	dither modulation
DWT	discrete wavelet transform
FDMA	frequency division multiple access
FFT	fast Fourier transform
GGD	generalized Gaussian distribution
IDCT	inverse discrete cosine transform
IFFT	inverse discrete Fourier transform
i.i.d.	independent identically-distributed
JPEG	joint photographic experts group
pdf	probability density function
ML	maximum likelihood
MSE	mean squared error
PNS	pseudo-noise sequence
PSNR	peek signal-to-noise ratio
PZM	pseudo-Zernike moment
QIM	quantization index modulation
ZM	Zernike moment

List of Symbols

a	watermark strength factor
A_{pq}	Zernike/pseudo-Zernike moment with order p and repetition q, as defined for an analog image function by (4.22)
$\check{A}_{pq}$	Zernike/pseudo-Zernike moment with order p and repetition q of a digital image, as approximated by the formula (4.23) in Cartesian coordinates
$\hat{A}_{pq}$	Zernike moment with order p and repetition q of a digital image, as computed by the approximation formula (5.8) in polar coordinates
$\mathcal{C}()$	the (linear) correlation of
$\mathbb{D}$	the unit disk:$\{(x,y) : x^2 + y^2 \leq 1\}$
$E\{\}$	the mean of
$\mathcal{F}()$	the forward FFT of
$\mathcal{F}^{-1}()$	the inverse FFT of
$h()$	1-D interpolation kernel function
I_i	the ith affine transformation invariant, $i = 1, ..., 4$
K	secret key in a watermarking system
L	length of the feature vector
$\mathbf{m}$	array of message symbols
$\hat{\mathbf{m}}$	array of estimated message symbols
M	value of M in M-ary modulation
n	number of bits to be embedded by a watermark
N	number of pixels in a row or a column of an image
$\mathcal{N}(a, \sigma^2)$	normal distribution with mean a and variance σ^2
$P()$	the probability of
P_e	the probability of error
$Q()$	Q-function
$\mathcal{Q}()$	quantization function
$\mathcal{R}()$	rounding function
R_{pq}	Zernike/pseudo-Zernike radial polynomial with order p and repetition q
V_{pq}	Zernike/pseudo-Zernike function with order p and repetition q
$Var\{\}$	the variance of
$\mathbf{W}$	watermark signal vector
$\mathbf{W}_\mathrm{r}$	reference PNS vector
$\mathbf{W}_\mathrm{m}$	PNS vector modulated by a message
$\mathbf{X}$	Feature vector extracted from the cover signal
$\tilde{\mathbf{X}}$	watermarked feature vector
$\mathbf{X}'$	possibly attacked watermarked feature vector
Δ	quantization step size
λ	Cartesian pixel width when an image is mapped onto the square $[-1, 1]^2$
ψ_i	the ith orthogonal transformation invariant, $i = 1, ..., 7$

Chapter 1

Introduction

1.1 The Concept of Digital Watermarking

Digital watermarking is the process of embedding some hidden information in a digital medium, such as an image, an audio signal and a video signal, by modifying the medium slightly, and later extracting the embedded information from the modified medium for some purpose. The embedded signal is called a *watermark*, and the original medium is called the *host signal* or *cover signal*, while the modified medium called a *watermarked signal* or *stego signal*.

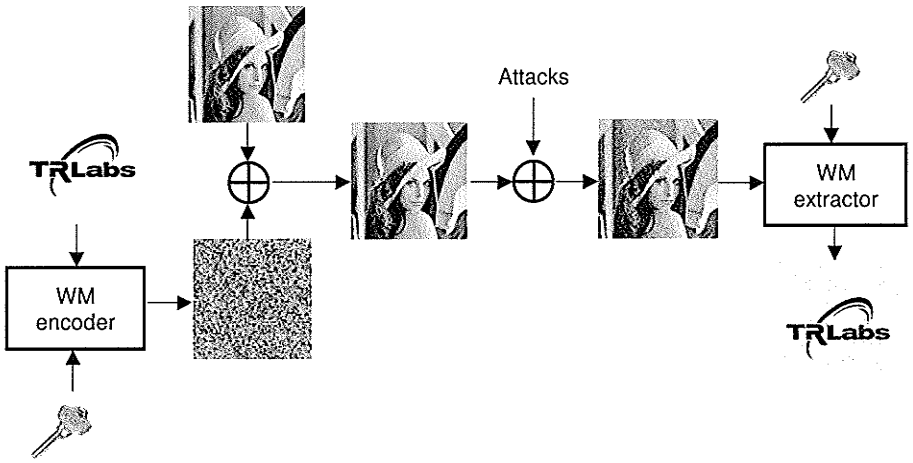


Figure 1.1: Illustration of the watermarking process

As an example, a typical watermarking process is shown in Fig. 1.1, where the cover signal is

an image and the message to be embedded is a logo. A secret key is often used in a watermarking system for security reasons. A watermarked signal often undergoes attacks, either intentional or unintentional, before it is used for watermark extraction. Because of the attacks, the extracted message is likely to contain some errors.

It is necessary to distinguish between *digital watermarking* and some other closely related terms, in particular, *information hiding* and *steganography*. Although there is not yet a universal agreement, *information hiding*, or *data hiding*, refers to the general technique by which information is embedded imperceptibly into a host signal, regardless of the purpose of the embedded information and the relationship between the the embedded signal and the host signal [63, 20]. Within the family of *information hiding* techniques, there are some purpose-oriented terms, including *digital watermarking* and *steganography*. *Steganography* [4, 51, 27] means covert communication, by which the host signal can be modified in any way, as long as the existence of the secret information does not arouse suspicion. In a steganographical scenario, the quality of the modified host signal is not the concern. In contrast, the basic purpose of *digital watermarking* is to protect the host signal in a certain way [20, 21, 22, 89, 30], and therefore the quality of the watermarked signal is as important as the communication of the embedded information, while the existence of a watermark does not necessarily have to be secret. Despite the subtle difference between *digital watermarking* and *data hiding*, we use them interchangeably in this thesis.

1.2 Applications of Digital Watermarking

Digital watermarking has attracted considerable attention in both academia and industry communities in the past decade, because it is regarded as a promising means to address problems related to digital media in a variety of ways.

- **Copyright protection** [21, 22, 89, 30]. This is the primary application of digital watermarking. The drastic development of new technologies, such as the advent of the Internet and

recordable compact disk, makes it extremely easy to disseminate and duplicate multimedia contents, which brings about the issue of copyright infringement. Under such a circumstance, the producer of a multimedia work can use a watermark as a proof to claim authorship or ownership.

- **Content authentication** [78, 79, 39, 46]. With the wide spread use of various multimedia editing software, it is a simple matter to modify the contents of a multimedia work. Without assurance of authenticity, one would not believe what he sees or hears. A watermark can be used as an effective means to verify the integrity of a work.

- **Fingerprinting** [71, 91, 74]. A watermark that uniquely identifies the buyer of a work can be embedded into the work before it is sold. Later, if an illegal copy of the work is found, the watermark extracted from it tells the source of the copy. This process is also referred to as traitor-tracing.

- **Device control** [8, 20]. The watermark embedded in a work can be used to control some devices. For example, in a DVD movie, a watermark can be embedded to indicate one of the three access modes, i.e., no copy, copy once and no restriction. A DVD recorder complying with the standard is controlled by the watermark, and operates accordingly.

- **Signal multiplexing**. Sometimes it is necessary to embed an auxiliary signal into a host signal for the purposes of annotation, captioning etc. For example, medical images can be watermarked with patients' identities, time stamps, and hospital information, which can be used either to avoid mix-up or for the purpose of classification or archiving.

These are just a few examples of applications of digital watermarking. As the research in this young field progresses, other application possibilities are being discovered.

1.3 Requirements of a Watermarking System

1.3.1 Typical Requirements

Depending on applications, the requirements of watermarking systems vary greatly. Amongst many possible requirements of a watermarking system, the most important ones include:

- **Watermark transparency.** In most situations, it is required that the existence of a watermark should not affect the quality of the host signal. In other words, a watermark should be imperceptible or transparent.

- **Watermark robustness.** This means the ability of a watermark to survive common signal processing it may undergo, such as noise addition and lossy compression. In the case of a zero-bit watermark [1], the rate of detection is used to measure the watermark robustness, given a fixed rate of false alarm. For a multibit watermark, bit rate error (BER) is often employed as a metric of watermark robustness.

- **Watermark security.** It refers to the ability of a watermark to survive intentional attacks, in particular, the ability to resist unauthorized detection, removal, and tampering [20]. It is generally accepted that the security of a watermark should depend on a secret key, rather than a secret algorithm of embedding or detection.

- **Payload amount.** This is the number of bits that a watermark conveys. Some applications can do with zero-bit watermarks, whereas others may require dozens or even hundreds of bits of data payload.

- **Oblivious extraction.** In many applications, the original cover signal is not available at the watermark detector or decoder. Under such circumstances, the watermark extraction should

[1] A zero-bit watermark is defined as a watermark containing no explicit message bit(s). The result of watermark extraction is simply a decision if a pattern is present or not.

be performed without access to the cover signal. This is called oblivious or blind watermark extraction.

1.3.2 Relationship Among the Requirements

The above requirements often interact and conflict with one another. In designing a watermarking system, one often has to take the following aspects into account.

- **Robustness vs. transparency.** For better robustness, a watermark is preferably inserted into perceptually significant features of the host signal and/or inserted with a big strength factor. However, both measures increase the obtrusiveness of the watermark.

- **Payload size vs. transparency.** Generally speaking, for a fixed level of watermark robustness, the energy of the watermark component per data unit is a fixed amount. The more data are to be embedded, the more is the total watermark energy, and consequently, the worse is the watermark transparency.

- **Robustness vs. payload size.** Similarly, for a fixed level of watermark transparency, increasing the amount of embedded data leads to less watermark energy per data unit, and hence lower watermark robustness.

- **Robustness vs. extraction mode.** In applications where nonblind watermark extraction is allowed, the interference of the host signal can be eliminated, and thus better watermark robustness can be achieved than in the case of blind watermark extraction. However, in many cases, the original host signal is not available.

Considering these restrictions, the fundamental task in designing a watermarking system is to achieve a good tradeoff among these conflicting requirements, depending on specific application scenarios.

1.4 Scope and Structure of the Thesis

As an emerging technology, digital watermarking has given rise to many research directions, among which we are particularly interested in the following aspects, which are be covered in this thesis.

- **Watermark robustness.** Despite the necessity of fragile or semi-fragile watermarks in some applications like data authentication [78, 79, 39, 46], we are concerned with robust watermarks, which find wider applications. In particular, watermarks robust to geometric distortions and common image processing manipulations are the focus of this thesis. Fragile watermarks are not considered.

- **Multibit watermarking.** Multibit watermarking has received less attention than zero-bit watermarking in literature, however it is necessary in many applications. As an extension of zero-bit and one-bit watermarking, it is more challenging technically.

- **Image watermarking.** Any form of digital media, including images, audio clips, video clips, and plain text, can be the object of digital watermarking. However, we are only concerned with digital image watermarking in this thesis. In spite of this, some of our techniques and algorithms proposed for image watermarking can be applied to audio and video watermarking as well.

- **Oblivious watermark extraction.** Many existing watermarking schemes assume the availability of the cover signal in watermark detection. Under this assumption, the cover signal can be subtracted from the stego signal before watermark extraction, posing no interference to the watermark signal. Nevertheless, in some applications, especially in scenarios of ownership dispute, the cover signal may not or should not be present in the process of watermark extraction [21, 22, 89, 30]. In all the algorithms presented in this proposal, we assume no access to the cover signal for watermark detection. In this case, the cover signal serves as a source of interference to the watermark signal.

This thesis is organized as follows. First, an investigation on techniques of robust watermarking is given in Chapter 2, which serves as a foundation for the watermarking systems to be set up in subsequent chapters. In Chapter 3, a series of watermarking systems are introduced progressively, from the most fundamental single-bit watermarking to some advanced multibit watermarking techniques such as efficient M-ary modulation based watermarking. In Chapter 4, a watermark based on image features on a circular domain, i.e., Zernike moments and pseudo-Zernike moments is proposed, which is robust to geometric transformations such as image rotation, image scaling and flipping. In Chapter 5 we further improve the geometrically invariant watermarking technique, by inventing an accurate way to compute Zernike moments in the polar coordinate system. Finally, in Chapter 6 a summary of the completed work, a list of contributions, and an outline of future research directions are presented.

Chapter 2

Robust Watermarking Techniques

As mentioned in Chapter 1, the robustness of a watermark is crucial in many applications. It has been and continues to be an important and difficult research problem in the area of digital watermarking. The difficulty lies in the fact that the attacks which could be applied to a watermarking system are virtually unpredictable. A careful design of a watermarking system may survive some of the known attacks, but fail under other attacks. It is almost certain that there does not exist a design of watermark with robustness to all possible attacks. A more practical approach is to make a watermark robust to application-specific attacks. In this chapter, we formulate some key techniques to tackle the watermark robustness from different perspectives. Particularly, we present three general aspects of watermark robustness: i) design of a robust watermark signal, ii) design of a robust watermark embedder, and iii) design of a robust watermark detector. We start by looking at the classification of existing watermarking techniques.

2.1 Types of Watermarking Techniques

A large number of digital watermarking algorithms have been proposed in the past decade. We can use Fig. 2.1 as a general model for various watermarking systems.

In Fig. 2.1, $\mathbf{m}$ is the message to be embedded in the cover signal $\mathbf{X}$, $\tilde{\mathbf{X}}$ is the watermarked

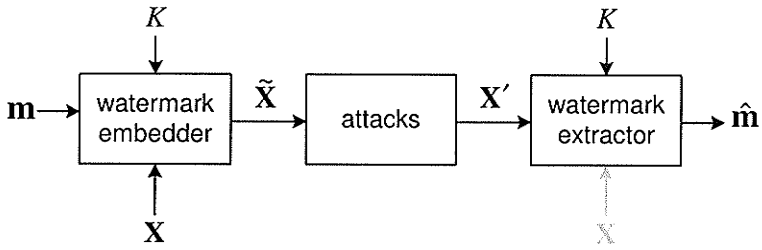


Figure 2.1: A generic model of digital watermarking

version of $\mathbf{X}$, $\mathbf{X}'$ is the attacked version of $\tilde{\mathbf{X}}$, K is a secret key shared by both the watermark embedder and the watermark detector/decoder, and $\hat{\mathbf{m}}$ is the estimate of the embedded message, which is the output from the detector/decoder. Non-blind detection requires that the cover signal $\mathbf{X}$ be present at the detector, while blind detection does not. It should be noted that due to the transparency condition, $\mathbf{X}$ and $\tilde{\mathbf{X}}$ should be very similar perceptually, and as well, the attacks are generally constrained to ensure that $\mathbf{X}'$ and $\tilde{\mathbf{X}}$ are similar.

There are different ways to categorize the existing watermarking techniques. For example, a popular method of categorization is conducted from the perspective of watermarking domain: the existing algorithms are either in direct signal sample space, or in transform domains. However in this work we are particularly interested in a communication perspective, namely, we are concerned with the form of a watermark signal and the method of its extraction. From this perspective, the watermarking algorithms fall into two types: coherent watermarking algorithms and noncoherent watermarking algorithms.

2.1.1 Type I: Coherent Watermarking Systems

In this category a reference pattern $\mathbf{W}_\mathrm{r}$, serving as the carrier signal, is needed. For reasons of watermark security and robustness, the carrier signal normally takes the form of a pseudonoise sequence (PNS), which follows either Gaussian or uniform distribution. For the generation of a PNS, a secret key K is used as the seed of a gold-sequence or an m-sequence. The generated PNS $\mathbf{W}_\mathrm{r}$ is then modulated by $\mathbf{m}$, the message to be hidden in the cover signal, resulting in a modulated PNS $\mathbf{W}_\mathrm{m}$, which is to be mixed with the cover signal by modifying the cover feature vector $\mathbf{X}$ to

produce the watermarked signal $\widetilde{\mathbf{X}}$. The modification can be performed either additively,

$$\widetilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_{\mathrm{m}}, \tag{2.1}$$

or multiplicatively

$$\widetilde{\mathbf{X}} = \mathbf{X}(1 + a\mathbf{W}_{\mathrm{m}}), \tag{2.2}$$

where a is the watermark strength parameter, controlling the tradeoff of the watermark amplitude and the watermark robustness. The structure of Type I watermark embedders is shown in Fig. 2.2(a).

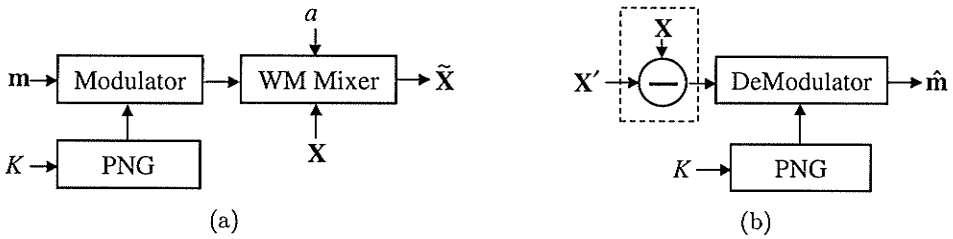


Figure 2.2: The architecture of a Type I (coherent) watermarking system. (a)Watermark embedder. (b) Watermark extractor. PNG stands for pseudonoise generator.

For the watermark extraction, the carrier signal $\mathbf{W}_{\mathrm{r}}$ is first regenerated using the same key K, and then used to demodulate the possibly distorted watermark signal $\mathbf{X}'$ to yield $\hat{\mathbf{m}}$, an estimate of the embedded message. Shown in Fig. 2.2(b), this process is just like a typical synchronous communication system. The demodulator often takes the form of a correlator followed by a comparator, on which we will elaborate in Chapter 3. It is worth noting that there is a subtracter marked by a dashed box in Fig. 2.2(b), which means it may or may not exist. In applications where the original unwatermarked cover signal $\mathbf{X}$ is available, this subtracter can help improve the performance of the watermark extractor. In this case of nonblind (nonoblivious) watermark extractor, the cover signal does not affect the extraction of the watermark. On the other hand, if $\mathbf{X}$ is not available, $\mathbf{X}'$ has to be used directly for blind (oblivious) watermark extraction. In this situation, the cover signal interferes in the extraction of the watermark, serving as a source of noise.

2.1.2 Type II: Noncoherent Watermarking Systems

In contrast to Type I watermarking systems, Type II watermarking systems do not require an explicit carrier signal. An outstanding example of this category is the quantization-based watermarking systems, where quantization index modulation (QIM) is applied in the watermark embedder [13]. The structure of a typical QIM-based embedder is shown in Fig. 2.3(a). Assume $m \in \{0, 1, ..., M-1\}$ is the message to embed into the feature vector $\mathbf{X}$. First M quantizers have to be designed: $\mathcal{Q}(\mathbf{X}, i), i = 0, ..., M-1$, which satisfy

$$\mathcal{Q}(\mathbf{X}; i) \approx \mathbf{X}, \ i = 0, ..., M-1 \tag{2.3}$$

and

$$\mathcal{Q}(\mathbf{X}; i) \neq \mathcal{Q}(\mathbf{X}; j), \ \forall \ \mathbf{X} \text{ if } i \neq j. \tag{2.4}$$

Each of these quantizers yields a unique partition of the feature space. The partitions of the feature space can be decided by a secret key K for security purpose, and the minimum distance between the partitions (and hence the watermark robustness) is controlled by the quantization step size Δ. Then the watermarked feature vector for message m is obtained as the one which has the least distance to $\mathbf{X}$ in partition m.

$$\widetilde{\mathbf{X}} = \mathcal{Q}(\mathbf{X}; m). \tag{2.5}$$

For the extraction of the embedded message from a possibly distorted vector $\mathbf{X}'(\approx \mathbf{X})$, $\mathbf{X}'$ has to be first quantized by the same M quantizers respectively

$$[\mathbf{X}']_i = \mathcal{Q}(\mathbf{X}'; i), \ i = 0, ..., M-1 \tag{2.6}$$

and then the distances between $\mathbf{X}'$ and its M quantized versions are compared. The index of the quantizer whose output has the minimum distance from $\mathbf{X}'$ is the estimate of the embedded message:

$$\hat{m} = \arg \min_{i \in \{0,...,M-1\}} \left\| [\mathbf{X}']_i - \mathbf{X}' \right\|, \tag{2.7}$$

where $\| \cdot \|$ denotes the norm operator. This process is illustrated in Fig. 2.3(b).

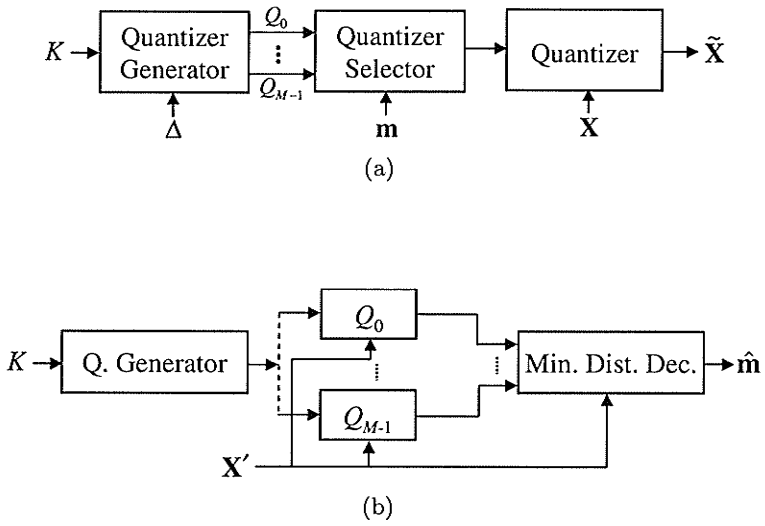


Figure 2.3: The architecture of a quantization-based (Type II, noncoherent) watermarking system. (a)Watermark embedder. (b) Watermark extractor.

Let's look at an example shown in Fig. 2.4, where the 2-D vector $\mathbf{X} = (X[1], X[2])$. The grid points marked with '0' and '1' indicate the possible outputs of the quantizers $\mathcal{Q}(\mathbf{X}, 0)$ and $\mathcal{Q}(\mathbf{X}, 1)$ respectively. Assume we have a vector point $\mathbf{X}$ and the message to embed is a '0', the watermarked vector $\mathbf{X}'$ is the nearest one among all the '0' grid points, shown in the figure as a black dot in the circle.

As a subclass of QIM, dither modulation (DM) is more practical and useful [13]. It is defined as

$$\mathcal{Q}_{\mathrm{DM}}(\mathbf{X}; m) = \mathcal{Q}(\mathbf{X} + d(m)) - d(m) \tag{2.8}$$

where $\mathcal{Q}(\cdot)$ is a base quantizer, and $d(m)$ is a dither factor corresponding to m from the dither vector $\mathbf{d} = (d[0], ..., d[M-1])$. Dither modulation has the property that the quantization cells and reconstruction points of any given quantizer are shifted versions of the quantization cells and reconstruction points of any other quantizer. Due to this special structure of quantizers, dither modulation has the advantage of easy implementation. As well, with the help of distortion compensation [13], it has the capability to reach the information capacity.

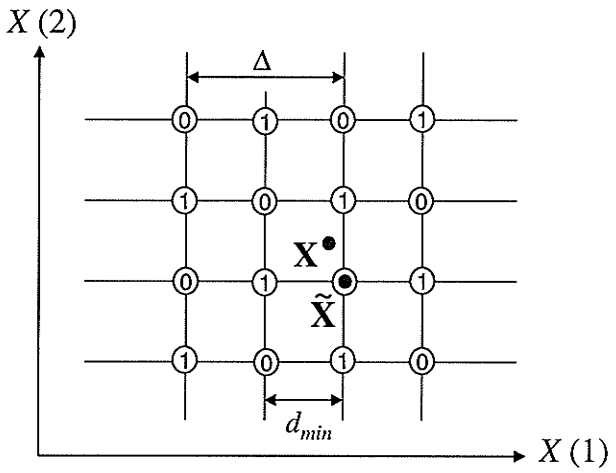


Figure 2.4: A 2-D signal partitioning for data hiding

2.2 Designing a Robust Watermark Signal

The watermark signal, $\mathbf{W} = \widetilde{\mathbf{X}} - \mathbf{X}$, which is the difference between a watermarked signal and cover signal, is the first and foremost factor to consider in the design of a robust watermarking system. In this section, we look at the ways to improve the watermark robustness through proper design of the watermark signal.

2.2.1 Choosing a Robust Watermark Signal Form

A watermark can take different forms. As introduced above, a watermark can be a modulated pseudo-noise sequence as in Type I techniques, or a quantization noise sequence as in Type II techniques. Generally speaking, when the watermark robustness is of prime concern, a PNS-based watermark signal is preferable due to the following facts:

- A PNS watermark is resistant to amplitude scaling. Multimedia data often undergo some slight amplitude change, such as the brightness change of images and videos and the loudness change of audios. If a watermark detector uses correlation coefficient as its sufficient statistics, or uses a comparison-based algorithm as shown in Chapter 3, the embedded watermark is invariant to amplitude changes. However, the quantization-based watermark is extremely

sensitive to the amplitude change of features. For example, in the case of binary QIM, when the amplitude change $\delta > \Delta/4$ (Δ is the quantization step size), the decoded data are likely to be erroneous.

- A PNS watermark is insensitive to signal cropping and filtering. Because a PNS watermark spreads its energy over many features, it can still be detected even if some of the features are lost due to signal cropping or filtering. On the contrary, for a quantization-based watermark, if some features are lost, the information embedded in these features cannot be recovered.

It should be noted that the robustness advantage of PNS-based watermarks is obtained at some price. First, it is usually difficult for a spread spectrum based system to embed a large data payload. Second, the extraction of hidden data in a PNS watermark is sensitive to synchronization. In the case of signal cropping, rotation and scaling etc., some measure has to be taken to address the synchronization issue. Moreover, a PNS watermarking system is usually more computationally expensive than a QIM watermarking system.

2.2.2 Using Side Information

For a watermarking system with oblivious detector, the existence of the cover signal whose power is dominantly larger than that of the watermark serves as a source of noise, interfering with the extraction of hidden data. Therefore, even if no attacks are applied to the watermarked signal, there is a chance that detection errors occur. However, this noise, different from other unpredictable noise, is completely known to the watermark embedder. This side information, as it is termed in communication theory, can be employed either to improve the watermark robustness or to increase the data payload. According to the dirty-paper theory [17], the existence of the known noise does not affect the information capacity of a channel. However in practice, it is not possible to implement the so called Costa codes which entails an infinitely large codebook. In spite of this, the side information can still assist us in the design of a robust watermark. The basic idea follows.

To embed a message $m \in \{0, ...M-1\}$, M groups of pseudonoise sequences are generated, with each group containing η PNSs. Assume that:

- The group corresponding to message m is $\{\mathbf{W}_1, ..., \mathbf{W}_\eta\}$.
- The watermark detector is a correlator $\mathcal{C}(\cdot, \cdot)$.

Any reference PNS in the group can represent the message m uniquely, but we only pick the one $\mathbf{W}_i$, which has the largest correlation with the known feature vector $\mathbf{X}$, i.e.:

$$i = \arg \max_{j \in \{1,...,\eta\}} \mathcal{C}(\mathbf{X}, \mathbf{W}_j), \tag{2.9}$$

and mix this PNS with the cover signal $\widetilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_i$, where a is the watermark strength factor.

In this way, we can improve the watermark robustness based on the knowledge of the cover signal. This is the concept of informed coding [20].

2.2.3 Gaining Watermark Robustness via Tradeoffs

As mentioned in Chapter 1, tradeoffs exist between watermark robustness and other conflicting aspects of a watermarking system, including watermark visibility, payload size etc. Let us now look at a few common approaches to watermark robustness by means of tradeoffs.

Repetitive watermarks are very effective in combating signal cropping and filtering. When the watermarking features are selected directly in the signal sample space, embedding the same watermark signal in different parts does prevent the loss of hidden data due to signal cropping. Similarly if the discrete cosine transform (DCT) or discrete Fourier transform (DFT) coefficients are selected as the watermarking features, a repetitive watermark is resistant to filtering which is likely to remove some of the watermarked features.

Error control codes (ECCs) are often applied in practical watermarking systems. Before data transmission (embedding), the data are coded with some redundancy so that when attacks take place, the redundant data can help to correct the possible errors. There are a variety of ECC

techniques that are useful in data hiding, such as Bose-Chaudhuri-Hocquenghem (BCH) codes and convolutional codes [7]. With an ECC, the BER of the extracted data can be improved significantly. Clearly the improved watermark robustness brought by ECCs are at the cost of data payload.

By fixing the amount of data payload, we can also raise the watermark robustness by increasing the watermark strength. However, this approach is limited by the requirement of watermark transparency. In image watermarking, peek signal-to-noise ratio (PSNR) is often used to measure the level of watermark transparency, which is defined by

$$\mathrm{PSNR}(f, \tilde{f}) = 10 \log_{10} \frac{f_{\max}^2}{\sigma_e^2}, \tag{2.10}$$

where f is the original image and $\tilde{f}$ is the watermarked version, both with dimensions $N_1 \times N_2$, $f_{\max}$ is the maximum intensity of image f, and

$$\sigma_e^2 = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} [\tilde{f}(i,j) - f(i,j)]^2 \tag{2.11}$$

is the mean square error. Roughly speaking, watermark transparency requires that PSNR $\geq$ 40dB.

2.3 Robust Watermark Embedder

Having fixed a watermark signal, we are now concerned with how to insert it into the cover signal. A careful design of the watermark embedder can help improve the watermark robustness considerably while maintaining the acceptable watermark invisibility. In this respect, there are several techniques of which we can take advantage.

2.3.1 Using Robust Host Features

A natural question is what features of the cover signal are suitable for watermark embedding. There is probably no universal answer to this question, because different features have different levels of robustness to a certain attack. For example, low-frequency DFT coefficients of an image are robust to lowpass filtering, but are vulnerable to image rotation and resizing etc. We list below a few kinds of features and their corresponding behaviors in combating common attacks.

- **Block-DCT coefficients.** For image watermarking, block-DCT coefficients are a useful feature, because a very common type of processing for watermarked images is JPEG lossy compression, which is right based on block-DCT. Therefore one can design an image watermark with a specific level of robustness to JPEG compression [46]. A watermark residing in the upper-left corner block-DCT coefficients can also survive lowpass filtering and contrast-change manipulations, but it is rather weak under geometric distortions.

- **Global DCT and DFT coefficients.** Low-frequency global-DCT and global-DFT coefficients (magnitudes) have good performance in the watermark robustness to JPEG lossy compression and lowpass filtering etc. However, like block-DCT coefficients, they have no robustness against geometric distortions.

- **DWT coefficients.** Discrete wavelet transform (DWT) coefficients are another important class of features for data hiding due to two facts. On the one hand, as a joint time-frequency transform, DWT coefficients are more efficient in representing perceptually important signal features, and thus potentially more robust to distortions. On the other hand, the latest image compression standard JPEG-2000 is based on DWT. Therefore it is possible to design a watermark with a specific level of robustness to JPEG-2000. A good watermark design with DWT coefficients can have better robustness to lowpass filtering than with DCT or DFT coefficients [87]. It is possible to design a DWT-based watermark with robustness to image scaling, but in general DWT coefficients are still vulnerable to image rotation etc.

- **Fourier-Mellin coefficients.** Fourier-Mellin coefficients have been proposed to deal with the watermark vulnerability to geometric attacks since they are invariants under rotation-scaling-translation (RST) operations [56, 47]. In theory watermarks residing in Fourier-Mellin coefficients are RST-resistant, but in practice there are some implementation problems [56], because the log-polar transforms lead to severe image degeneration.

- **Circularly orthogonal moments.** Zernike moments (ZMs) and pseudo-Zernike moments

(PZMs) are based on Zernike/pseudo-Zernike functions which are complete and orthogonal on the unit disk. One of the outstanding properties of ZMs/PZMs is that their magnitudes are invariant to image rotation and flipping. Therefore it is possible to design a watermark with good geometric robustness, as well as robustness to additive noise, lowpass filtering and lossy compression etc. [80, 81, 82, 86, 84, 83]. We present the details of watermark systems based on ZMs/PZMs in Chapters 4 and 5.

These are some commonly used image features for data hiding. Because there do not exist a single kind of features which are robust to all possible attacks, the choice of features should depend on application scenarios.

2.3.2 Adaptive Embedding

In Type I techniques, the most commonly used watermark embedding method is the linear additive formula $\widetilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_{\mathrm{m}}$. In such an embedder, all the watermark features are treated equally, and the watermark energy is spread over them evenly. The advantage of such an embedder is the ease of implementation. However, from the perspective of watermark robustness, it is not the optimal one. In fact, different host features have different capabilities in carrying a watermark due to their varying perceptual roles and magnitudes. Taking this into account, one can design more robust watermarks while meeting other system requirements.

Multiplicative Embedding

It is believed that host features with large amplitudes play more important perceptual roles, and are less likely to be lost or distorted significantly. Considering this, larger host features should bear more watermark energy. In particular, if the watermark energy distributed to the features is proportional to their amplitudes, we arrive at the multiplicative embedding:

$$\widetilde{\mathbf{X}} = \mathbf{X} + \mathbf{W}, \quad \mathbf{W} = a\mathbf{W}_{\mathrm{m}}\mathbf{X}, \tag{2.12}$$

where a is, again, the watermark strength factor used to control the tradeoff between the watermark visibility and the watermark robustness.

There are a number of algorithms proposed with a multiplicative embedder. In [19], the watermark is embedded multiplicatively into a certain number of the largest DCT coefficients of an image. In [15, 5], the strategies for optimum extraction of multiplicative watermarks are proposed.

Perceptually Weighted Embedding

Except that they are in direct sample space, the perceptual roles of different host features usually vary, and so do their capabilities of accommodating changes without causing perceivable signal distortion. Based on this fact, a watermark should be embedded into host features adaptively for the sake of watermark robustness. In algorithms of perceptually weighted watermark embedding, there are two fundamental issues, namely, how to model the features perceptually, and how to apply the model in watermark embedding.

The perceptual modeling of signal features is a complex issue. It depends on a number of factors, such as the signal type and the feature domain. For example, an image signal has a different perceptual model from an audio signal, and the DCT coefficients of an image have to be modelled differently from its DWT coefficients. In the cases of images and videos, we have the following human visual properties that could be employed in modelling the features [34].

- **Frequency sensitivity.** Human eyes are more sensitive to low-frequency image components than high-frequency ones.
- **Luminance sensitivity.** Human eyes are more sensitive to the noise in areas with low luminance than the noise in areas with high luminance.
- **Contrast masking.** The perceptual response to a certain feature is affected by its neighboring features.

There are several ways to utilize these perceptual properties to enhance the watermark ro-

bustness. One of them is perceptually weighted watermark embedding. Based on the frequency sensitivity property, one can obtain the just-noticeable-difference (JND) threshold for a certain signal feature like a DCT coefficient [75]. Assume $\mathbf{J}$ is the corresponding JND vector for the host feature vector $\mathbf{X}$, then the perceptually weighted watermark embedding can be implemented in the following way [64, 77]:

$$\widetilde{X}[i] = \begin{cases} X[i] + J[i]W[i], & \text{if } |X[i]| > J[i] \\ X[i], & \text{otherwise.} \end{cases} \tag{2.13}$$

Other ways to use the perceptual properties include the global perceptual distance-based embedding and optimally scaled embedding etc., as suggested in [20].

2.4 Robust Watermark Detector

For nonoblivious watermark detection/decoding, the impact of the cover signal on the watermark signal does not pose a problem, because the former can be subtracted from the latter before watermark detection. But for oblivious watermark detection, the cover signal is a source of noise. In nature, oblivious watermark extraction is a process of detecting/estimating weak signals from strong noise. In this section, we list some methods of optimal watermark detection from a statistical communications perspective. In particular, the detection statistic is considered in some typical cases.

2.4.1 Optimal Detection of an Additive Watermark

An additive watermark signal $\mathbf{W}$ is usually independent of the host vector $\mathbf{X}$. It is a scaled version of a PNS modulated by a message, i.e., $\mathbf{W} = a\mathbf{W}_m$, and the watermarked feature vector is $\widetilde{\mathbf{X}} = \mathbf{X} + \mathbf{W}$.

Gaussian Distributed Features

If the feature vector $\mathbf{X}$ can be modeled as an i.i.d. Gaussian distributed sequence, then it can be shown [35, 72] by the theory of hypothesis testing that the optimal detection statistic is the correlation:

$$C(\widetilde{\mathbf{X}}, \mathbf{W}) = \frac{1}{L} \sum_{i=1}^{L} \widetilde{X}[i] W[i], \tag{2.14}$$

where L is the number of elements in vector $\mathbf{W}$. Comparison of this detection statistic to a threshold gives an optimal decision regarding the presence of $\mathbf{W}$ in $\widetilde{\mathbf{X}}$. The advantage of the correlator is that it can be implemented easily, and therefore it has been used widely in watermarking applications.

Generalized-Gaussian Distributed Features

The correlation (2.14) as a detection statistic works optimally for Gaussian distributed features. However, the most commonly used host features, such as DCT and DWT coefficients, are never Gaussian distributed in a strict sense. Therefore the watermark detection through a correlator functions suboptimally at best. For this reason, optimal watermark detection from non-Gaussian features has been investigated [89, 29, 14, 55]. Basically, two aspects are involved in this issue: the statistical modeling of host features and its corresponding optimal watermark detector.

Several kinds of transform domain features, such as DCT and DWT coefficients, are better modeled by generalized Gaussian distribution (GGD) with the following probability density function (PDF):

$$f_{\mathbf{x}}(x) = A e^{-|\beta(x-m)|^c}, \tag{2.15}$$

where $\beta = \frac{1}{\sigma}\sqrt{\frac{\Gamma(3/c)}{\Gamma(1/c)}}$, $A = \frac{c\beta}{2\Gamma(1/c)}$, m is the mean, σ is the standard deviation, c is the shape parameter, and $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$ is the Gamma-function. It is worth noting that when $c = 2$ and 1, GGD reduces to Gaussian and Laplacian distributions respectively, while for $c = 0$ and $c \to \infty$, GGD becomes uniform and impulse distributions respectively.

For a possibly watermarked test feature vector $\widetilde{\mathbf{X}}$, the objective of watermark detection is to

decide which of the following two hypotheses is true:

$$\begin{cases} H_0 : \widetilde{\mathbf{X}} = \mathbf{X} \\ H_1 : \widetilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_m \end{cases} \text{ or equivalently } \begin{cases} H_0 : a = 0 \\ H_1 : a > 0 \end{cases}. \tag{2.16}$$

When a is not known to the detector, there does not exist a uniformly most powerful (UMP) test [35]. Therefore a locally optimal detection (LOD) statistic has to be sought, which can be derived as the following [14]

$$R_{\text{lod}}(\widetilde{\mathbf{X}}) = \frac{1}{L}\sum_{i=1}^{L} \text{sgn}(\widetilde{X}[i]) \left|\widetilde{X}[i]\right|^{c-1} W_m[i] \tag{2.17}$$

for GGD host features, where $\text{sgn}(\cdot)$ is the signum function.

Another asymptotically optimum detection statistic can be obtained as the Rao test [35], which is in the case of GGD additive noise [55]:

$$R_{\text{rao}}(\widetilde{\mathbf{X}}) = \frac{\left[\sum_{i=1}^{L} \text{sgn}(\widetilde{X}[i]) \left|\widetilde{X}[i]\right|^{c-1} W_m[i]\right]^2}{\sum_{i=1}^{L} \left|\widetilde{X}[i]\right|^{2(c-1)}}. \tag{2.18}$$

In order to use either (2.17) or (2.18), one must have the GGD shape parameter c of the feature vector $\mathbf{X}$, which is not available in an oblivious watermark detector. Due to the fact that the watermarked signal is only slightly different from the original signal, we can estimate the GGD parameters of the original features, including the variance σ^2 and the shape factor c, from the watermarked version. The variance can readily be estimated by

$$\hat{\sigma}^2 = \frac{1}{L}\sum_{i=1}^{L} \widetilde{X}^2[i] - \frac{1}{L^2}\left(\sum_{i=1}^{L} \widetilde{X}[i]\right)^2, \tag{2.19}$$

while the shape parameter can be estimated numerically by using the equation [50]

$$\frac{E^2[|\widetilde{\mathbf{X}}|]}{\hat{\sigma}^2} = \frac{\Gamma^2(2/\hat{c})}{\Gamma(1/\hat{c})\Gamma(3/\hat{c})}. \tag{2.20}$$

2.4.2 Optimal Detection of a Non-additive Watermark

By non-additive watermark, we mean that the watermark signal $\mathbf{W}$ is dependent on the host feature $\mathbf{X}$ in some way, i.e., $\mathbf{W} = \mathbf{W}(\mathbf{X}, \mathbf{W}_m)$. In particular, the watermarked signal is

$$\widetilde{X}[i] = X[i](1 + a[i]W_m[i]). \tag{2.21}$$

Examples include the multiplicative watermarks as expressed by (2.12), and more generally, the perceptually adaptive watermarks as expressed by (2.13). The optimal detection of non-additive watermarks can also be addressed by the Neyman-Pearson criterion. The particular solutions depend on the statistics of the host features.

Generalized-Gaussian Distributed Features

As stated previously, the transform domain coefficients of the host signal, such as those in DWT and DCT domains, can be statistically modelled by GGD, i.e., $p_{X[i]}(x) = A[i]e^{-|\beta[i](x-m)|^{c[i]}}$. If the watermark strength $a[i]$ is known to the detector, there exists a uniformly most powerful (UMP) detector [15]:

$$L(\widetilde{\mathbf{X}}) = \sum_{i=1}^{L} \left| a[i]\widetilde{X}[i] \right|^{c[i]} \left(1 - (1 + a[i])^{-c[i]W_m[i]}\right) \underset{\text{w absent}}{\overset{\text{w present}}{\gtrless}} T, \tag{2.22}$$

where T is a pre-determined threshold meeting the requirement of the false alarm rate.

However, when the watermark strength $a[i]$ is not known to the detector, there does not exist a UMP watermark detector. In this case, a locally most powerful (LMP) detector can be obtained, which is [15]

$$\sum_{i=1}^{L} c[i] \left| a[i]\widetilde{X}[i] \right|^{c[i]} W_m[i] \underset{\text{w absent}}{\overset{\text{w present}}{\gtrless}} T', \tag{2.23}$$

where T' is a pre-determined threshold meeting the requirement of the false alarm rate.

Weibull Distributed Features

Different from DCT or DWT coefficients, DFT coefficients cannot be statistically modelled by GGD. A good model for the magnitudes of DFT coefficients is Weibull distribution, which has the

following pdf:

$$f_{\mathbf{x}}(x) = \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} e^{-\left(\frac{x}{\alpha}\right)^{\beta}}, \tag{2.24}$$

where the parameters $\alpha > 0, \beta > 0$ determine the shape, the mean, and the variance of the distribution [57].

Assume $X[i] \sim \text{Weibull}(\alpha[i], \beta[i])$, then the optimal watermark detector for the embedder (2.21) is the following test [15]:

$$\sum_{i=1}^{L} \beta[i]\left(\frac{\widetilde{X}[i]}{\alpha[i]}\right)^{\beta[i]} W_m[i] \underset{\text{w absent}}{\overset{\text{w present}}{\gtrless}} T'', \tag{2.25}$$

where T'' is a pre-determined threshold meeting the requirement of the false alarm rate.

Chapter 3

A Robust Multibit Watermark Based on Spread Spectrum Technique

Spread spectrum techniques have been widely used in digital watermarking due to its distinguishing characteristics such as excellent security and robustness performance [19, 20]. In a typical spread spectrum based watermarking system, a feature vector extracted from the host signal, such as a vector of DCT or DWT coefficients, is slightly modified by a pseudonoise sequence (PNS), either additively or multiplicatively. By calculating the correlation between the pseudonoise sequence and the feature vector extracted from a test object, and then comparing it to a threshold, one can detect the presence of the pseudonoise sequence. This is the concept of a zero-bit watermarking system.

In this chapter, we consider the problem of multibit watermarking based on spread spectrum techniques. We start by introducing the simplest 1-bit watermark, from which we proceed to n-bit watermarks. There are a number of approaches one can take to design an n-bit watermark based on a 1-bit watermark. The most straightforward methods are through channel multiplexing techniques, which are borrowed from communication theory. One such technique is to divide the feature vector into n subvectors, each of which accommodates a 1-bit watermark. Time division multiple access

and frequency division multiple access fall into this type of techniques. Another technique is to use the same feature vector many times, each time a separate message symbol is embedded as a layer of noise (from the perspective of the host signal), which is the concept of code division multiple access. We show that these two multiplexing approaches have the same performance in terms of decoding error rate when the watermark visibility and the payload amount are fixed. Subsequently, a more effective multibit watermarking technique, known as M-ary modulation, is employed to design an n-bit watermark. The M-ary modulation technique offers much better error performance than the multiplexing techniques. One relevant issue of concern is the complexity of an M-ary watermarking system. We present an efficient implementation scheme of M-ary modulation, i.e., M-ary phase modulation, which only requires reasonably low cost of computation, even if M is as large as 2^{20}. Simulation results show the advantage of the watermark based on M-ary phase modulation.

3.1 A One-bit Watermark

Before we discuss the design of a multibit watermark, let us look at the 1-bit watermark, which is the basis of a multibit watermark. Assume $\mathbf{X} = (X[1], ..., X[L])$ is a vector of signal features selected for watermarking, which can be original signal samples, or coefficients of some transform, such as DCT, DFT, and DWT, and the message to embed is a binary digit $m \in \{0, 1\}$. There are two alternatives we can take to embed the bit into the host signal.

3.1.1 One-bit Watermarking via Two PNSs

For the embedding of the message bit m, we first generate two i.i.d. pseudonoise sequences $\mathbf{W}_0 = (W_0[1], ..., W_0[L])$ and $\mathbf{W}_1 = (W_1[1], ..., W_1[L])$ with a key K, where $W_j[i] \sim \mathcal{N}(0, 1), j = 0, 1; i = 1, ..., L$. The basic idea is that we use $\mathbf{W}_0$ and $\mathbf{W}_1$ to represent '0' and '1' respectively. $\mathbf{W}_\mathrm{m}$, the PNS used to modify the host signal, is either $\mathbf{W}_0$ or $\mathbf{W}_1$, depending on the bit value to be

embedded:

$$\mathbf{W}_{\mathrm{m}} = \begin{cases} \mathbf{W}_0, & \text{if } m = 0; \\ \mathbf{W}_1, & \text{if } m = 1. \end{cases} \tag{3.1}$$

Then the watermarked signal is obtained as a mixture of $\mathbf{X}$ and $\mathbf{W}_{\mathrm{m}}$, formed either additively

$$\widetilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_{\mathrm{m}} \tag{3.2}$$

where a is a constant watermark strength factor, or multiplicatively

$$\widetilde{\mathbf{X}} = \mathbf{X}(1 + \mathbf{a}\mathbf{W}_{\mathrm{m}}) \tag{3.3}$$

where $\mathbf{a} = (a[1], ..., a[L])$ is a vector of watermark strength.

For watermark extraction from $\widetilde{\mathbf{X}}$, $\mathbf{W}_0$ and $\mathbf{W}_1$ are re-generated with the same key K. Afterwards a certain detector $\mathcal{S}()$ is invoked for the calculation of the detection statistics between $\widetilde{\mathbf{X}}$ and both $\mathbf{W}_0$ and $\mathbf{W}_1$, respectively. The embedded bit is estimated based on the following decision rule:

$$\hat{m} = \begin{cases} 0, & \text{if } \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_0) > \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_1) \text{ and } \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_0) > T_s; \\ 1, & \text{if } \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_1) > \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_0) \text{ and } \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_1) > T_s; \\ \text{None}, & \text{if } \max\{\mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_0), \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_1)\} < T_s. \end{cases} \tag{3.4}$$

where T_s is a pre-determined threshold for a required false alarm rate. For the watermark detector $\mathcal{S}()$, one can employ one of the optimal methods introduced in Chapter 2. If the watermark is embedded additively, (2.14) is used for a Gaussian host feature vector $\mathbf{X}$, and (2.17) or (2.18) is used for a GGD host feature vector $\mathbf{X}$. If the watermark is embedded non-additively, (2.22) or (2.23) is used for a GGD host feature vector, while (2.25) is used for a Weibull host feature vector, such as a vector of DFT coefficients.

Let us look at the error performance of the watermark detector under the simplistic assumption that the host features are i.i.d. Gaussian random variables, i.e., $X[i] \sim \mathcal{N}(0, \sigma_{\mathrm{x}}^2), i = 1, ..., L$. In this case, the optimal watermark detector is a correlator (matched filter) as expressed by (2.14).

It can be shown (see Appendix A.2 for reference) that the distribution of the correlation between the watermarked signal $\widetilde{\mathbf{X}}$ and $\mathbf{W}_k, k = 0, 1$ is described by

$$C(\widetilde{\mathbf{X}}, \mathbf{W}_k) \sim \begin{cases} \mathcal{N}(0, \frac{\sigma_{\mathbf{X}}^2 + a^2}{L}) & \text{if } k \neq m \\ \mathcal{N}(a, \frac{\sigma_{\mathbf{X}}^2 + 2a^2}{L}) & \text{if } k = m \end{cases}, \tag{3.5}$$

which is illustrated in Fig. 3.1. If the message m is equally likely to be '0' and '1', and $\sigma_{\mathbf{X}} \gg a$, the error rate can be obtained based on (3.5):

$$P_e = \int_{-\infty}^{\infty} \phi(x) Q(\frac{x}{\sigma_c}) dx, \tag{3.6}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{x^2}{2}} dx$, $\phi(x) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{(x-1)^2}{2\sigma_c^2}}$ and $\sigma_c = \frac{\sigma_{\mathbf{X}}}{a\sqrt{L}}$.

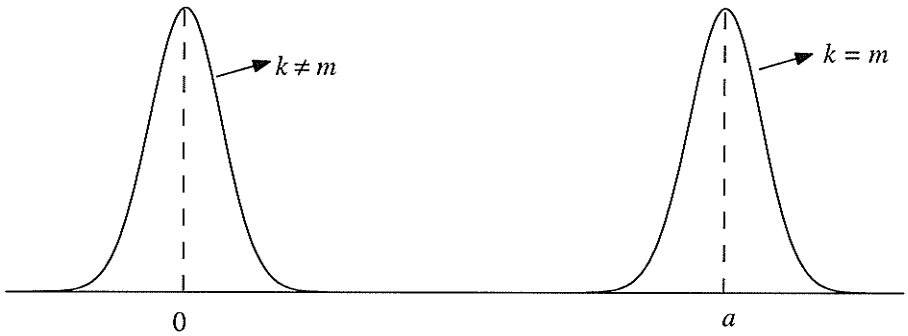


Figure 3.1: The distribution of the detection statistics

3.1.2 One-bit Watermarking via One PNS

With this approach, we generate a single i.i.d. pseudonoise sequence $\mathbf{W}_0 = (W_0[1], ..., W_0[L])$ with a key K, where $W_0[i] \sim \mathcal{N}(0, 1), i = 1, ..., L$. Then the pseudonoise vector is modulated by the message m to be embedded

$$\mathbf{W}_{\mathrm{m}} = (2m - 1)\mathbf{W}_0. \tag{3.7}$$

In other words, we use $\mathbf{W}_0$ to represent the bit '1', and its opposite $-\mathbf{W}_0$ to represent the bit '0'. The modulated PNS is then mixed with the host vector $\mathbf{X}$ either additively by (3.2) or multiplicatively by (3.3).

With the re-generated PNS $\mathbf{W}_0$, the extraction of the watermark can be performed with an optimal detector, depending on the statistics of the host features and the embedding rule. The decision is made in the following way for the estimation of the embedded message: The embedded message is decoded as

$$\hat{m} = \begin{cases} 1, & \text{if } \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_0) > T_s; \\ 0, & \text{if } \mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_0) < -T_s; \\ \text{None}, & \text{if } |\mathcal{S}(\widetilde{\mathbf{X}}, \mathbf{W}_0)| \leq T_s, \end{cases} \tag{3.8}$$

where $\mathcal{S}()$ is an optimal watermark detector function, and T_s is a pre-determined threshold for a required false alarm rate.

The error performance of the watermark detector is of great interest. To simplify the analysis, we assume that the host features are i.i.d. Gaussian random variables, i.e., $X[i] \sim \mathcal{N}(0, \sigma_{\mathbf{X}}^2), i = 1, ..., L$. Under this situation, the optimal watermark detector is the correlator (2.14). It can be shown that the correlation between the watermarked signal $\widetilde{\mathbf{X}}$ and $\mathbf{W}_0$ is

$$C(\widetilde{\mathbf{X}}, \mathbf{W}_0) \sim \begin{cases} \mathcal{N}(-a, \frac{\sigma_{\mathbf{X}}^2 + 2a^2}{L}) & \text{if } m = 0 \\ \mathcal{N}(a, \frac{\sigma_{\mathbf{X}}^2 + 2a^2}{L}) & \text{if } m = 1 \end{cases}, \tag{3.9}$$

which is illustrated in Fig. 3.2. If the message m is equally likely to be '0' and '1', and $\sigma_{\mathbf{X}} \gg a$, the error rate of watermark detection is

$$P_e = Q(\frac{a\sqrt{L}}{\sigma_{\mathbf{X}}}), \tag{3.10}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{x^2}{2}} dx$.

Comparing (3.6) and (3.10), one can see that with a, L, and $\sigma_{\mathbf{X}}$ fixed, the error rate of the watermarking system with one PNS (and its negative) is smaller than that of the system with two PNSs. Therefore, we employ implicitly the one-PNS based technique in the construction of an n-bit watermark, unless otherwise stated.

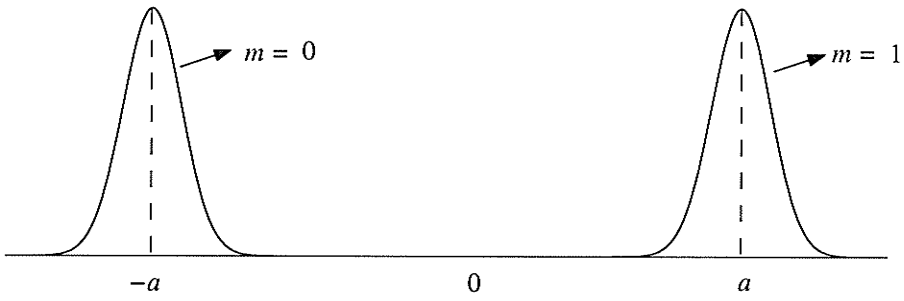


Figure 3.2: The distribution of the detection statistics

3.2 A Multibit Watermark via FDMA

In the preceding section, we have discussed how a single bit of information can be embedded into a host feature vector $\mathbf{X} = (X[1], ..., X[L])$, and later extracted from $\widetilde{\mathbf{X}}$. Now we are interested in the extension of 1-bit watermarking to n-bit watermarking. The objective is to embed n message bits $\mathbf{b} = (b_1, ..., b_n)$ in $\mathbf{X}$, by modifying it imperceptibly to produce $\widetilde{\mathbf{X}}$, and later extract the embedded bits from $\widetilde{\mathbf{X}}$ without access to $\mathbf{X}$.

Let us look into a straightforward approach based on the 1-bit watermarking. The basic idea is illustrated by Fig. 3.3. The solution is to divide the feature vector $\mathbf{X}$ into n subvectors, each of which has a length $l = L/n$ and hosts a 1-bit watermark, which has already been addressed technically in the preceding section. In this way, we can accomplish embedding $\mathbf{b}$ into the host signal. We call this class of techniques feature division multiple access (FDMA). In particular, it is space/time division multiple access if the host features are simply some samples in space/time domain, and it is frequency division multiple access if the host features are some coefficients in some transform domain like DFT domain and DCT domain.

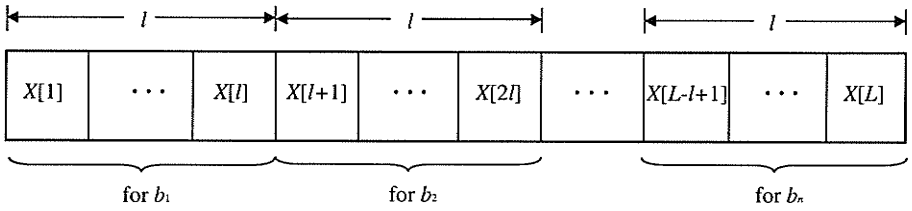


Figure 3.3: The technique of feature space division based multibit watermarking

This intuitive multiplexing technique has the advantage of easy implementation, but the watermark embedded in this way is vulnerable to signal cropping and/or signal filtering. Another disadvantage is that different feature groups may have different levels of magnitude, thus leading to uneven watermark robustness.

3.3 A Multibit Watermark via CDMA

To overcome the limitations of feature division-based multibit watermarking, we consider code division multiple access (CDMA) for n-bit watermarking. First, n different pseudonoise sequences $\mathbf{W}_j, j = 1, ..., n$ are generated, each with the same length L as the host feature vector $\mathbf{X}$. Then each pseudonoise sequence $\mathbf{W}_j$ is modulated by the message bit b_j to be embedded in the following way:

$$\mathbf{W}_{\mathrm{m}}^{(j)} = (2b_j - 1)\mathbf{W}_j, \ j = 1, \ldots, n. \tag{3.11}$$

In other words, the information of each message bit is spread over the whole spectrum of the feature vector space. The composite watermark signal is simply the summation of each individually modulated PNS:

$$\mathbf{W}_{\mathrm{m}} = \sum_{j=1}^{n} \mathbf{W}_{\mathrm{m}}^{(j)}, \tag{3.12}$$

which is to be mixed with the host feature vector, either additively or non-additively.

The prerequisite of CDMA-based watermarking is the orthogonality or quasi-orthogonality between any pair of the PNSs, which can be met when the length of PNSs is sufficiently large.

3.4 Equivalence of FDMA Watermark and CDMA Watermark

A natural question is, which approach is superior, an FDMA watermark or a CDMA watermark? One may get different answers from different perspectives. Among various perspectives, we are most concerned with the error performance of a watermarking system. To simplify analysis, we

still assume that the host feature $\mathbf{X}$ has a Gaussian distribution with variance $\sigma_{\mathbf{X}}^2$, the power of the watermark signal is a^2, and additive embedding is applied.

In the case of FDMA-based watermarking, all the parameters are the same as those of the 1-bit watermarking, except the length of the PNS for each message bit, which is now $l = L/n$. According to (3.10), the error rate is

$$P_e^{\text{FDMA}} = Q(\frac{a\sqrt{l}}{\sigma_{\mathbf{X}}}) = Q(\frac{a\sqrt{L}}{\sigma_{\mathbf{X}}\sqrt{n}}). \tag{3.13}$$

Whereas in the case of CDMA-based watermarking, all the parameters are the same as those of the 1-bit watermarking, except the power of the PNS for each message bit, which is now $a'^2 = a^2/n$. According to (3.10), the error rate is

$$P_e^{\text{CDMA}} = Q(\frac{a'\sqrt{L}}{\sigma_{\mathbf{X}}}) = Q(\frac{a\sqrt{L}}{\sigma_{\mathbf{X}}\sqrt{n}}). \tag{3.14}$$

It is worth noting that in derivation of (3.14), we assume that each PNS $\mathbf{W}_j$ is orthogonal to any other PNS $\mathbf{W}_i, i \neq j$, which is well satisfied when L is large in practice.

Comparing (3.13) and (3.14), one can see that an FDMA-based watermark is exactly equivalent to a CDMA-based watermark in terms of error performance.

However, they differ a lot in other aspects. In terms of implementation, an FDMA-based watermarking system is advantageous, because a single one PNS can be used mutually for different message bits, and thus only L/n random numbers need to be generated, while a CDMA-based watermark needs Ln random numbers to be generated. On the other hand, in terms of robustness, an FDMA-based watermark is sensitive to signal cropping and filtering, and has an uneven watermark robustness from bit to bit, but a CDMA-based watermark is free of all these problems.

3.5 A Multibit Watermark via Conventional M-ary Modulation

M-ary modulation is a concept which originated from communication theory [76, 65], and recently was proposed for digital watermarking by some researchers [56, 41, 20, 71]. It was shown that the

performance of a watermarking system can be considerably improved by M-ary modulation [41], but people tend to agree that in practice, this advantage is limited by the computational cost in message decoding when M is large, e.g., $M > 256$. However, in our work we find that with a proper design of watermark signals, this limitation is not necessarily true.

An effective method to design an n-bit watermark is to use M-ary modulation technique based on PNSs. Conventionally a group of M pseudonoise patterns $\{\mathbf{W}_0, \ldots, \mathbf{W}_{M-1}\}$ are generated independently with a secret key K, each of which is an L-element i.i.d. sequence, following Gaussian distribution $\mathcal{N}(0, 1)$. One of the prominent properties of the PNSs generated in this way is their quasi-orthogonality, i.e.,

$$\mathcal{C}(\mathbf{W}_j, \mathbf{W}_k) \approx \delta(j - k), \tag{3.15}$$

where $\mathcal{C}(\cdot)$ denotes the operation of linear correlation, which is defined as

$$\mathcal{C}(\mathbf{W}_j, \mathbf{W}_k) \triangleq \frac{1}{L} \sum_{i=1}^{L} W_j[i] W_k[i]. \tag{3.16}$$

Obviously a pseudonoise pattern $\mathbf{W}_m$ in the group can be used to represent an M-ary message symbol $m \in \{0, \ldots, M-1\}$, and thus it is capable of carrying $\log_2 M$ bits of information once chosen for data embedding. In other words, $\mathbf{W}_m$ can be viewed as a pseudonoise sequence modulated by the $\log_2 M$ bits of data to be embedded. This is the concept of M-ary modulation [41], also referred to as direct message coding [20] and orthogonal modulation [65, 71], as named by different authors.

With an additive embedding function, the M-ary message m can be embedded into the feature vector $\mathbf{X}$ by

$$\widetilde{\mathbf{X}} = \mathcal{E}(\mathbf{X}, m) = \mathbf{X} + a\mathbf{W}_m, \tag{3.17}$$

where $\widetilde{\mathbf{X}}$ is the watermarked feature vector, and a is the amplitude factor of the watermark, controlling the tradeoff between watermark visibility and watermark robustness, which is determined by the requirement of the application.

Now the important issue is how to extract the embedded data from $\widetilde{\mathbf{X}}$ without access to the original host signal $\mathbf{X}$. If $\mathbf{X}$ can be modelled as an i.i.d. sequence with Gaussian distribution,

a bank of linear correlators (matched filters) can be applied for the optimal extraction of the embedded information, as shown in Fig. 3.4, where $\mathbf{W}_0, \ldots, \mathbf{W}_{M-1}$ are re-generated PNSs with the same key K as in the embedding process, and $C(\widetilde{\mathbf{X}}, \mathbf{W}_j)$, the linear correlation between each reference pattern and the test signal is computed. With a maximum likelihood (ML) estimator, the embedded message is decoded as the index number of the reference pattern which has the maximum correlation with the test signal:

$$\hat{m} = \arg \max_{j \in \{0,\ldots,M-1\}} C(\widetilde{\mathbf{X}}, \mathbf{W}_j). \tag{3.18}$$

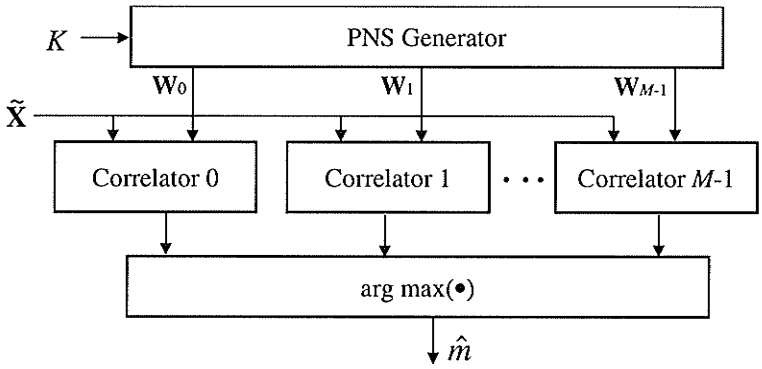


Figure 3.4: The structure of the conventional decoder for M-ary message coding.

M-ary modulation can significantly improve the performance of a watermarking system [41]. In general, the greater the value of M, the better the system performance is in terms of data error rates or data robustness. However, a large M value leads to high computational cost with the decoding structure as shown in Fig. 3.4. Because 2^n correlators are needed for an n-bit watermark, the decoding computation could be prohibitively expensive when n reaches a certain value. For instance, to extract a 16-bit watermark, 65536 correlation values have to be calculated, which could be difficult to implement in practice. Due to this difficulty, a value of $M \geq 256$ appears to be impractical with a decoding structure shown in Fig. 3.4.

To reduce the computational cost of the above M-ary watermark decoder, an improved algorithm using a tree-structure was proposed in [71]. To detect the embedded reference pattern $\mathbf{W}_m$,

all the relevant PNSs are first divided into two $\frac{1}{2}$ size groups

$$\{\mathbf{W}_0, ..., \mathbf{W}_{M-1}\} = \{\mathbf{W}_0, ..., \mathbf{W}_{\frac{M}{2}-1}\} \cup \{\mathbf{W}_{\frac{M}{2}}, ..., \mathbf{W}_{M-1}\}. \tag{3.19}$$

Then the test vector $\widetilde{\mathbf{X}}$ is correlated with the sum of all the PNSs in each group:

$$\begin{cases} c_1 = \mathcal{C}(\widetilde{\mathbf{X}}, \sum_{i=0}^{M/2-1} \mathbf{W}_i) \\ c_2 = \mathcal{C}(\widetilde{\mathbf{X}}, \sum_{i=M/2}^{M-1} \mathbf{W}_i). \end{cases} \tag{3.20}$$

If $c_1 > c_2$, the embedded pattern $\mathbf{W}_m$ must be in the first group, and otherwise in the second group. The group with $\mathbf{W}_m$ is then divided again into two $\frac{1}{4}$ size groups to decide the location of $\mathbf{W}_m$. This process continues until the exact position of $\mathbf{W}_m$ is located, whose index number is the estimate of the embedded message.

This algorithm reduces the number of correlators to $2\log_2 M$. It should be clear that the actual reduction of computation is less than that, because it introduces some other additional operations, such as summations. An issue of this approach is that it results in a higher rate of decoding errors than the direct correlation algorithm, especially for blind watermark extraction.

3.6 A Multibit Watermark via Efficient M-ary Modulation

As mentioned in the previous section, the M correlations for the extraction of an M-ary symbol can be prohibitively expensive when M is large. Another problem inherent in the conventional decoding structure shown in Fig. 3.4 is the time-consuming task of re-generating the M independent pseudonoise sequences, $\mathbf{W}_0, \ldots, \mathbf{W}_{M-1}$, which are necessary for data extraction. However, if we drop the requirement on the independence of the M pseudonoise sequences, we can solve the problem elegantly with the use of fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT), as shown subsequently.

3.6.1 An M-ary Phase Modulation Based Watermark

To overcome the computational bottleneck of the conventional M-ary modulation based watermarking system, we form the set of M reference patterns $\{\mathbf{W}_0, ..., \mathbf{W}_{M-1}\}$ with only one reference PNS in the following way:

- A reference PNS $\mathbf{W}_\mathrm{r}$ is generated as an i.i.d., Gaussian distributed sequence: $W_\mathrm{r}[i] \sim \mathcal{N}(0,1)$, $i = 1, ..., L$, where L is the length of the feature vector $\mathbf{X}$.

- Based on $\mathbf{W}_\mathrm{r}$, a set of M PNSs are generated to be circular-shift versions of $\mathbf{W}_\mathrm{r}$, satisfying

$$W_m[i] = \begin{cases} W_\mathrm{r}[i+m] & \text{if } i < L-m; \\ W_\mathrm{r}[i+m-L] & \text{otherwise.} \end{cases} \tag{3.21}$$

$$m = 0, ..., M-1; i = 1, ..., L.$$

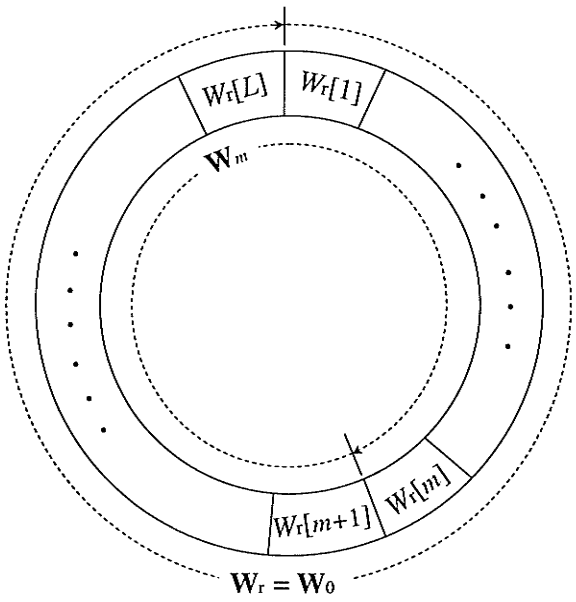


Figure 3.5: The formation of a set of circular shift PNSs based on $\mathbf{W}_\mathrm{r}$.

This process is illustrated in Fig. 3.5. It can be seen that the same PNS can be used to represent M different messages with its M phases respectively. In other words, a PNS whose phase is modulated by the message m can represent m uniquely. Drawing on the fact that $\mathbf{W}_\mathrm{r}$ is an

i.i.d. Gaussian PNS, we can show that the set of PNSs formed in this way satisfy the requirement of quasi-orthogonality expressed by (3.15), although they are not independent. This property is illustrated by Fig. 3.6, where as an example, $\mathbf{W}_{\mathrm{r}}$ is an i.i.d. normally distributed PNS with 1000 elements, and the correlations of $\mathbf{W}_{200}$ with all the circular-shift versions of $\mathbf{W}_{\mathrm{r}}$ as a function of the number of shifts are shown.

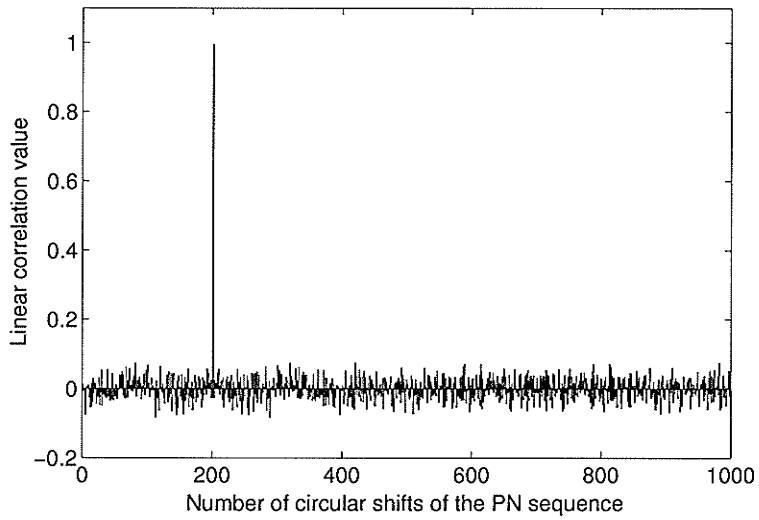


Figure 3.6: The linear correlation between a pseudonoise sequence and its circular shift versions.

Now that the set of PNSs $\{\mathbf{W}_0, ..., \mathbf{W}_{M-1}\}$ are ready, we can use them for M-ary data hiding according to (3.17). The interesting part of our algorithm lies in the extraction of the embedded data. With the circular versions of a PNS as the reference set, we no longer have to perform M correlations for data decoding as conventionally. We can compute, with a very simple method, all the correlations between the watermarked feature vector $\widetilde{\mathbf{X}}$ and the M PNSs derived from $\mathbf{W}_{\mathrm{r}}$. This computation can be implemented conveniently and efficiently by two forward FFT operations and one IFFT operation as follows,

$$\mathbf{c} = \frac{1}{L}\mathcal{F}^{-1}\left(\mathcal{F}(\widetilde{\mathbf{X}})\mathcal{F}^{*}(\mathbf{W}_{\mathrm{r}})\right), \tag{3.22}$$

where $\mathbf{c} = (c[0], c[1], ..., c[L-1])$, $c[i]$ is the correlation between $\widetilde{\mathbf{X}}$ and $\mathbf{W}_i$, $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote FFT and IFFT operations respectively.

The proof of (3.22) can be found in Appendix A.1. With $c[0], ...c[M-1]$ calculated according to (3.22), one can immediately get the estimate of the embedded message through (3.18).

3.6.2 An Extended M-ary Phase Modulation Based Watermark

It is a plain fact that the total number of PNSs derived from a given PNS $\mathbf{W}_\mathrm{r}$ of length L through circular shifting is L. If the desired value of M for M-ary data hiding satisfies $M \leq L$, the efficient method introduced above can be applied. However, if $M > L$, the above scheme does not apply. It appears that at most $\log_2 L$ bits of data can be embedded into the feature vector $\mathbf{X}$ with length L by a pseudonoise sequence. Fortunately this is not true. Next we show that this limitation can be easily circumvented.

Now the set of M reference patterns $\{\mathbf{W}_0, ..., \mathbf{W}_{M-1}\}$ are formed in the following way:

- A reference PNS $\mathbf{W}_\mathrm{r}$ is generated as an i.i.d., Gaussian distributed sequence: $W_\mathrm{r}[i] \sim \mathcal{N}(0,1)$, $i = 1, ..., M$.

- Based on $\mathbf{W}_\mathrm{r}$, a set of M PNSs are generated to be windowed circular-shift versions of $\mathbf{W}_\mathrm{r}$, satisfying

$$W_m[i] = \begin{cases} W_\mathrm{r}[i+m] & \text{if } i < M-m; \\ W_\mathrm{r}[i+m-M] & \text{otherwise.} \end{cases} \tag{3.23}$$

$$m = 0, ..., M-1; i = 1, ..., L.$$

This process is illustrated in Fig. 3.7. It is distinct from the process (3.21) in two ways. First, the length of $\mathbf{W}_\mathrm{r}$ is M, rather than L. Second, the length of $\mathbf{W}_i, i \in \{0, ..., M-1\}$ is less than that of the reference PNS $\mathbf{W}_\mathrm{r}$. In other words, $\{\mathbf{W}_0, ..., \mathbf{W}_{M-1}\}$ are derived to be windowed circular shifts of $\mathbf{W}_\mathrm{r}$.

Now let us look at how a multibit watermark is embedded and extracted with the set of PNSs derived by (3.23). In order to embed an M-ary symbol m, the corresponding $\mathbf{W}_m$ is selected

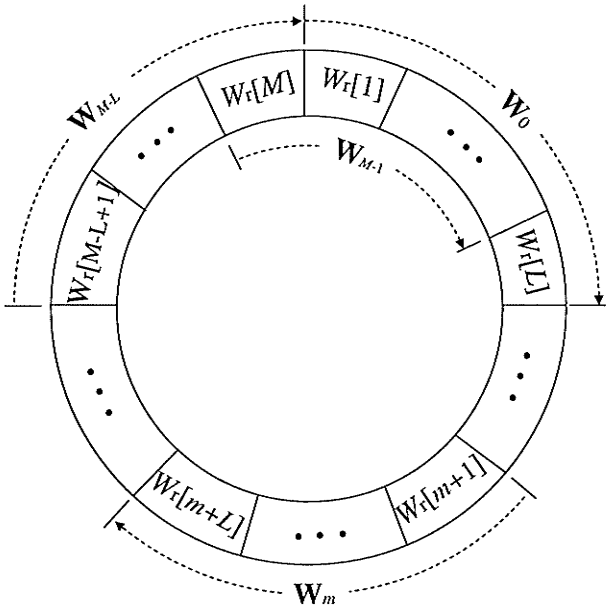


Figure 3.7: The formation of a set of windowed circular shift PNSs based on $\mathbf{W}_{\mathrm{r}}$.

from the set of PNSs, and it is embedded additively into $\mathbf{X}$ according to (3.17). For watermark extraction, we have to use a distinct strategy. Since now the watermarked feature vector $\widetilde{\mathbf{X}}$ and the reference PNS $\mathbf{W}_{\mathrm{r}}$ have different lengths, (3.22) cannot be applied directly. The solution is to first append zeros to $\widetilde{\mathbf{X}}$ so that it has the same length as $\mathbf{W}_{\mathrm{r}}$:

$$\widetilde{X}'[i] = \begin{cases} \widetilde{X}[i] & \text{for } 1 \leq i \leq L \\ 0 & \text{for } L+1 \leq i \leq M \end{cases}. \tag{3.24}$$

Then the correlations between $\widetilde{\mathbf{X}}$ and the set of PNSs $\{\mathbf{W}_i, i = 0, ..., M-1\}$ can be computed by

$$\mathbf{c} = \frac{1}{L}\mathcal{F}^{-1}\left(\mathcal{F}(\tilde{\mathbf{X}}')\mathcal{F}^*(\mathbf{W}_{\mathrm{r}})\right). \tag{3.25}$$

Summarizing the solutions to M-ary based data hiding stated above, Fig. 3.8 shows the block diagram of our proposed algorithm for M-ary watermark decoding, where the dashed block means that if $M \leq L$, the zero-padding process is not necessary, $\otimes$ indicates element-wise product, conj$(\cdot)$ denotes conjugation operation, and argmax$(\cdot)$ is the function of getting the index number of the largest correlation value.

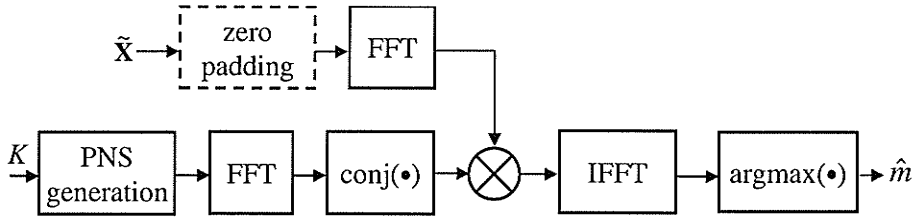


Figure 3.8: The structure of the proposed algorithm for efficient M-ary watermark extraction.

3.7 Computational Advantage of M-ary Phase Modulation

As noted before, the reason for the adoption of M-ary phase modulation in the design of a watermarking system is that it requires dramatically less computation than a conventional M-ary modulation based system. This computational advantage lies dominantly in the stage of watermark extraction, i.e., data decoding. Now let us compare quantitatively the computational complexity of the two methods. In the case of a conventional M-ary decoder illustrated in Fig. 3.4, the total number of operations required for the decoding of an M-ary symbol is approximately

$$T_0 = LM, \tag{3.26}$$

where L is the length of the feature vector. One operation is defined as one real multiplication plus one real addition. Apparently T_0 is a linear function of M. However, in the case of the proposed M-ary decoder illustrated in Fig. 3.8, the decoding of an M-ary symbol just involves 2 FFT and 1 IFFT operations. Because the complexity of one FFT or IFFT is $O(M \log_2 M)$ [34], the total number of operations required is approximately

$$T_1 = 3M \log_2 M. \tag{3.27}$$

To see more clearly the advantage of the proposed M-ary phase modulation over the conventional M-ary modulation, we plot in Fig. 3.9 T_0 and T_1 as a function of M in the range of our interest, for $L = 1024$.

One can see from Fig. 3.9 that the algorithm complexity of the conventional M-ary decoder is one or two orders of magnitude higher than that of the proposed M-ary phase decoder when

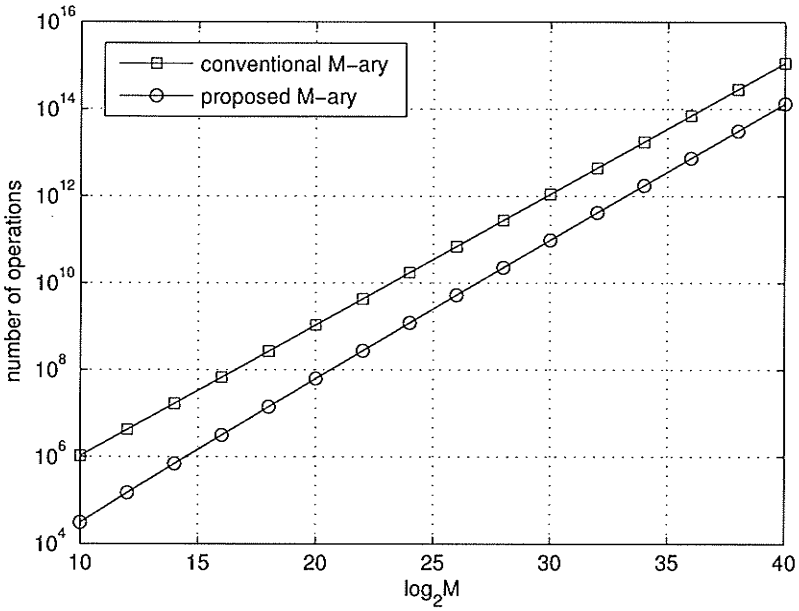


Figure 3.9: The algorithm complexity of the conventional M-ary decoder and the proposed M-ary decoder.

$L = 1024$. On the other hand, T_0 is a linear function of L, but T_1 is independent of L. This means that as L increases, the advantage of the proposed M-ary phase modulation over the conventional M-ary modulation is getting bigger linearly.

3.8 Error Performance of M-ary Modulation Based Watermarking

The algorithm proposed above makes M-ary phase modulation fully feasible in the design of spread spectrum based data hiding, even if M is very large. Now we are concerned with the performance improvement brought by M-ary phase modulation, in particular, the relationship between the value of M and the error rate of data extraction.

Theorem 3.1 *Let* $\widetilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_m$, *where* $\mathbf{X}$ *is a vector with* L *i.i.d. outcomes of* $\mathcal{N}(0, \sigma_{\mathbf{X}}^2)$, $\mathbf{W}_m$ *is a vector with* L *i.i.d. realizations of* $\mathcal{N}(0,1)$, *and* a *is a positive constant. If* $\mathbf{W}_k$ *is the* k*th circular*

shift of $\mathbf{W}_m$, *then*

$$C(\widetilde{\mathbf{X}}, \mathbf{W}_k) \sim \begin{cases} \mathcal{N}(0, \frac{\sigma_{\mathbf{X}}^2+a^2}{L}) & \text{if } k \neq m \\ \mathcal{N}(a, \frac{\sigma_{\mathbf{X}}^2+2a^2}{L}) & \text{if } k = m \end{cases}, \tag{3.28}$$

where $C(\cdot)$ *is the correlation function defined in (3.16).*

The proof of this theorem can be found in Appendix A.2. Based on this theorem, we have the following conclusion about the error probability of data extraction.

Theorem 3.2 *Let an* M*-ary message* m *be embedded into a feature vector* $\mathbf{X}$ *according to* $\widetilde{\mathbf{X}} = \mathbf{X} + a\mathbf{W}_m$, *where* $\mathbf{X}$ *has* L *i.i.d. outcomes of* $\mathcal{N}(0, \sigma_{\mathbf{X}}^2)$, $\mathbf{W}_m$ *is a vector with* L *i.i.d. realizations of* $\mathcal{N}(0,1)$, *and the constant* $a > 0$. *If* $\sigma_{\mathbf{X}} \gg a$ [1], *then the error probability of an ML estimator (3.18) is*

$$P_e \approx 1 - \int_{-\infty}^{\infty} \phi(x) \left[1 - Q(\frac{x}{\sigma_c})\right]^{M-1} dx, \tag{3.29}$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{(x-1)^2}{2\sigma_c^2}}$, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{x^2}{2}} dx$, *and* $\sigma_c = \frac{\sigma_{\mathbf{X}}}{a\sqrt{L}}$.

The proof of this theorem can be found in Appendix A.3. According to (3.29), we plot the error rate P_e as a function of σ_c^2 for various values of M, in particular, $M = 2^4, 2^8, 2^{12}, 2^{16}$, as shown in Fig. 3.10. From (3.29) and Fig. 3.10, we can draw some important conclusions. Firstly, with M and L fixed, P_e is a function of $\frac{\sigma_{\mathbf{X}}^2}{a^2}$, which can be viewed as the signal-to-noise ratio from the perspective of the host signal. It is an intuitive fact that the larger is the ratio $\frac{\sigma_{\mathbf{X}}^2}{a^2}$, the weaker is the embedded watermark signal, and therefore the more likely does the error occur. Secondly, with M and the ratio $\frac{\sigma_{\mathbf{X}}^2}{a^2}$ fixed, P_e is a function of L. As L increases, the error rate goes down. This is

[1] *In data hiding applications, this assumption is usually valid due to the watermark transparency requirement.*

also intuitive, because a larger L always reduces the variance of the detection statistic, and hence the chance of decoding error. An interesting fact is that L and $\frac{\sigma_X^2}{a^2}$ can be traded with each other. As long as $\sigma_c^2 = \frac{\sigma_X^2}{a^2 L}$ remains unchanged, P_e does not change. Finally, P_e is a function of M. As M increases, the error rate becomes higher. This is a price to pay for the increase of the amount of data embedded.

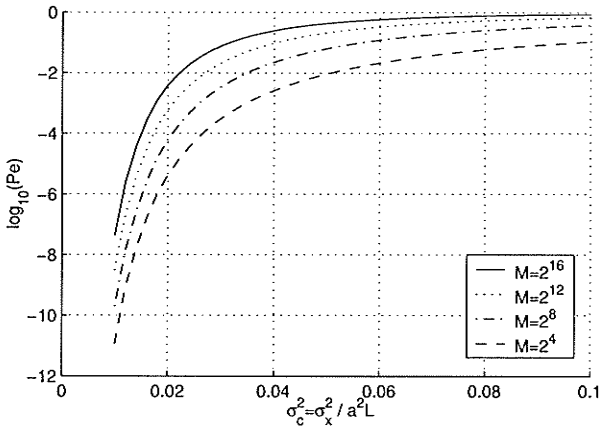


Figure 3.10: The error rates of an M-ary ML decoder.

Now we are concerned with the real performance improvement brought by M-ary phase modulation in our context of data hiding. To have a fair comparison among different cases of M values, we have to fix some parameters, including the number of bits to be embedded n, the power ratio of the feature vector and the watermark $r = \frac{\sigma_X^2}{a^2}$. Under these conditions, there are several schemes to design the watermark, such as FDMA and CDMA approaches, as mentioned in the introduction. Here we focus on the FDMA-based approach for the purpose of comparison. The general idea is as follows. An M-ary PNS represents $\log_2(M)$ bits of data, and thus for the embedding of n bits into the L-element host vector $\mathbf{X}$, we need to divide $\mathbf{X}$ into $n/\log_2(M)$ subvectors. Each subvector has a length of $L\log_2(M)/n$. A different M leads to a different number of subvectors, and hence a different length of subvectors. Our goal is to look into the error performance as a function of M. Based on (3.29), we plot a set of Pe–M curves, fixing $L = 4096, n = 16, r = \{80, 60, 40, 20\}$, as shown in Fig. 3.11. From this figure, we can see clearly that as M increases, the error ratio drops monotonically. This is particularly obvious when r is small, i.e., when the watermark signal

is strong.

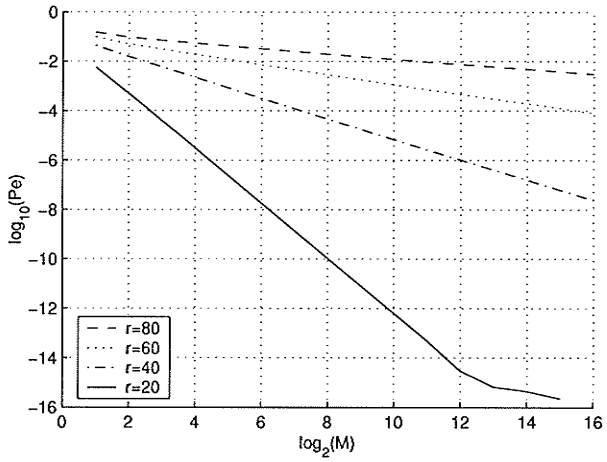


Figure 3.11: The performance improvement of M-ary phase modulation based data hiding as a function of M.

3.9 An Example of a Multibit Watermark Design

In this section, we apply the techniques introduced in previous sections in the design of a practical watermarking system, on which some experimental results are obtained and presented with details. From the results the effectiveness of the proposed algorithms and techniques are verified.

3.9.1 The Structure of an Experimental Multibit Watermarking System

In order to see the advantage of watermarks based on M-ary phase modulation, we design a multibit watermarking system via a combination of M-ary phase modulation and CDMA techniques. The structure of the watermark embedder is shown in Fig. 3.12.

First, an image $\mathbf{x}$ undergoes an 8×8 block DCT transform. In each 8×8 matrix of DCT coefficients, some mid-frequency coefficients are selected for watermarking, as illustrated by Fig. 3.13. The selected coefficients are subsequently reorganized to be a 1-D feature vector $\mathbf{X}$. A bit sequence $\mathbf{b} = (b_1, ..., b_n)$ to be embedded into $\mathbf{X}$ has to be mapped into a sequence of M-ary symbols $\mathbf{m} = (m[1], ..., m[n'])$, where $n' = n/\log_2 M$. For each M-ary symbol $m[i]$, a different reference PNS

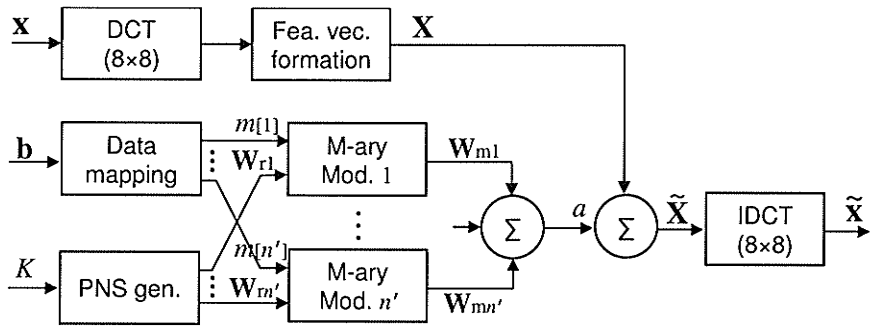


Figure 3.12: The embedder structure of the multibit watermarking system based on M-ary phase modulation plus CDMA.

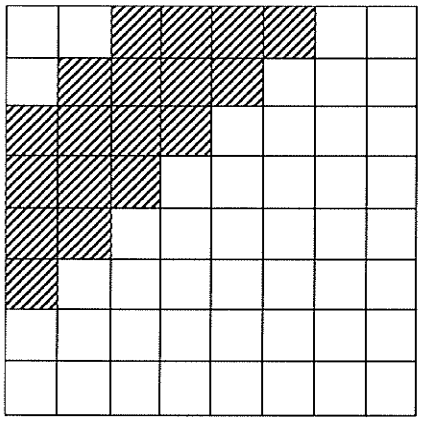


Figure 3.13: The coefficients in an 8×8 DCT block selected for data hiding.

$\mathbf{W}_{ri}$ is needed, and therefore n' reference PNSs are generated with a key K. The i^{th} PNS $\mathbf{W}_{ri}$ is modulated by the M-ary symbol $m[i]$ in the i^{th} M-ary modulator, in the way described in Section 3.6, which results in $\mathbf{W}_{mi}$. Due to the property of quasi-orthogonality, the n' modulated PNSs can be added up based on CDMA. The composite signal $\sum_{i=1}^{n'} \mathbf{W}_{mi}$ is subsequently scaled by a factor a to control the tradeoff between watermark robustness and watermark obtrusiveness, before it is combined with the feature vector $\mathbf{X}$. Each element in the resulting watermarked vector $\widetilde{\mathbf{X}}$ is substituted for its original counterpart in the DCT coefficient matrix, and finally the watermarked image $\widetilde{\mathbf{x}}$ is obtained through an inverse DCT.

A mechanism shown in Fig. 3.14 is designed for watermark extraction. A feature vector $\mathbf{X}'$ is first extracted from a possibly distorted watermarked signal $\mathbf{x}'$ through an 8×8 block DCT transform, and then fed into each of the n' M-ary demodulators. Based on the same key K, the n'

reference PNSs are re-generated, and they are used in the n' M-ary demodulators respectively for the estimation of the embedded symbols. The details of each M-ary demodulator are shown in Fig. 3.8, and explained in Section 3.6. The estimated M-ary symbols $\hat{m}[i], i = 1, ..., n'$, are subsequently mapped into the estimated bit sequence $\hat{\mathbf{b}} = (\hat{b}_1, ..., \hat{b}_n)$.

3.9.2 Experimental Results

With the watermark embedder (Fig. 3.12) and the watermark extractor (Fig. 3.14), we performed some experiments, focusing on watermark robustness to some common manipulations and the relationship between the watermark robustness and the value of M. The test images are a set of 256×256 images with 256 gray levels, shown in Fig. 3.15. For each experiment in this section, the watermark strength factor a is adjusted such that the quality of the watermarked image remains the same, PSNR = 40dB. The watermark robustness is measured by bit error rate (BER).

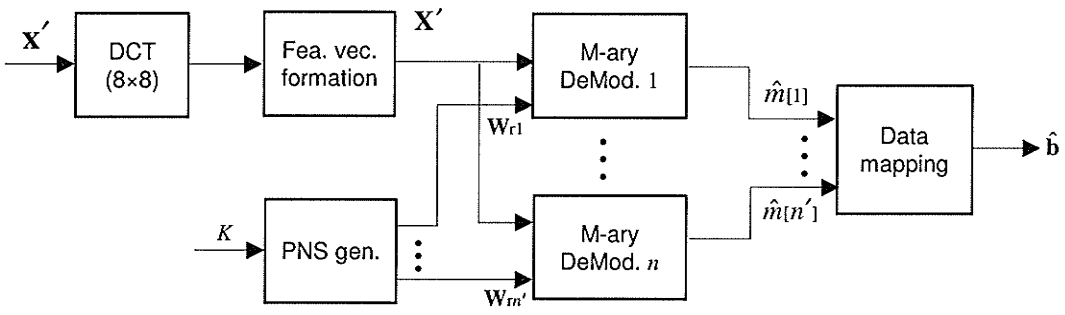


Figure 3.14: The decoder structure of the multibit watermarking system based on M-ary phase modulation plus CDMA.

Watermark Robustness to Lossy Compression

Lossy compression of images, dominantly represented by JPEG standard, is a common and easy way to process images, and therefore watermark robustness against JPEG compression is necessary. An example of JPEG compression is illustrated in Fig. 3.16(a). To look into the robustness of the designed watermark against JPEG compression, we first watermark images with the data to be embedded, and then compress the watermarked images with a number of different quality

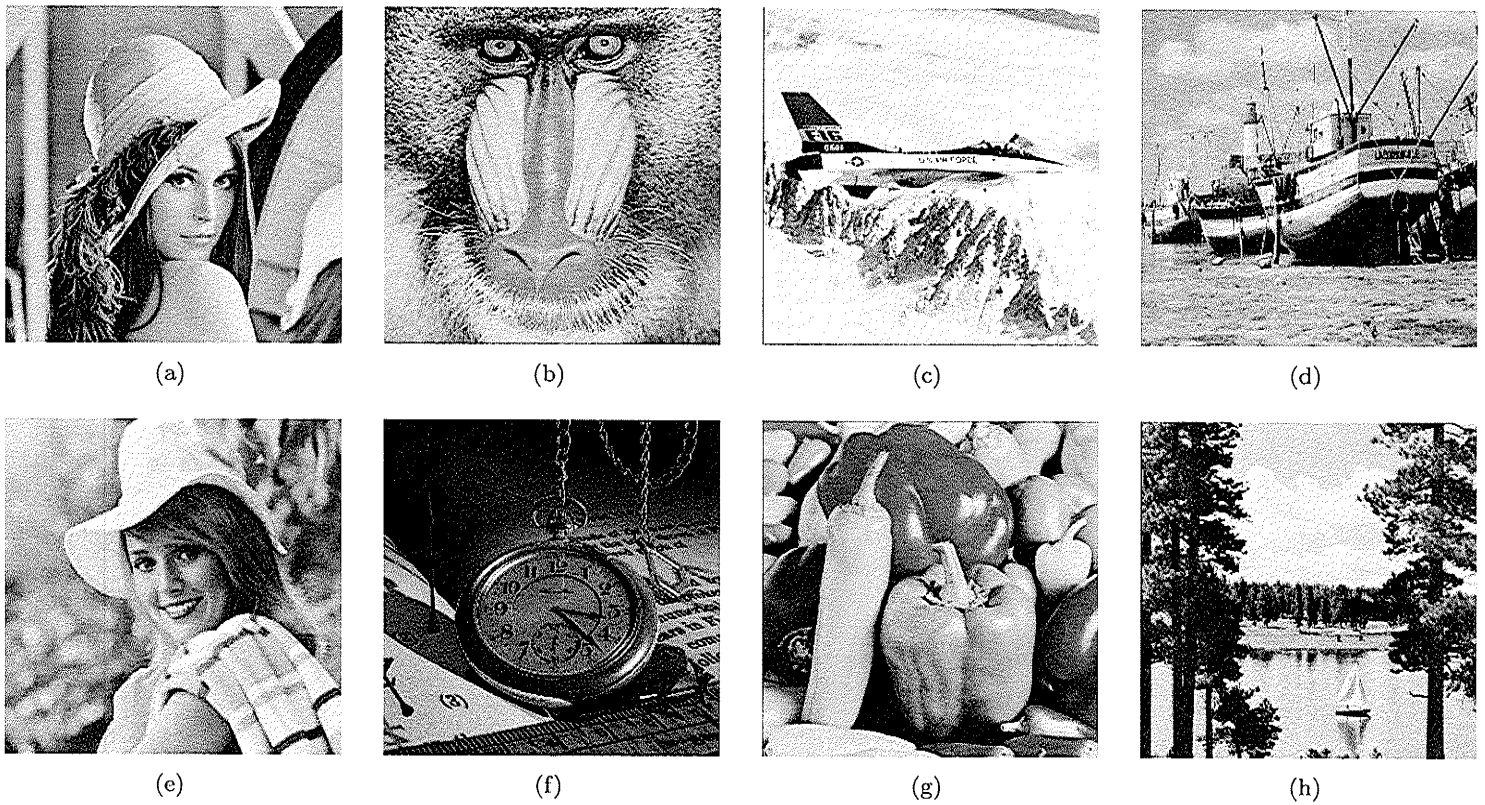


Figure 3.15: Original test images. (a) Lena. (b) Baboon. (c) F-16. (d) Fishing boat. (e) Elaine. (f) Watch. (g) Peppers. (h) Sailboat.

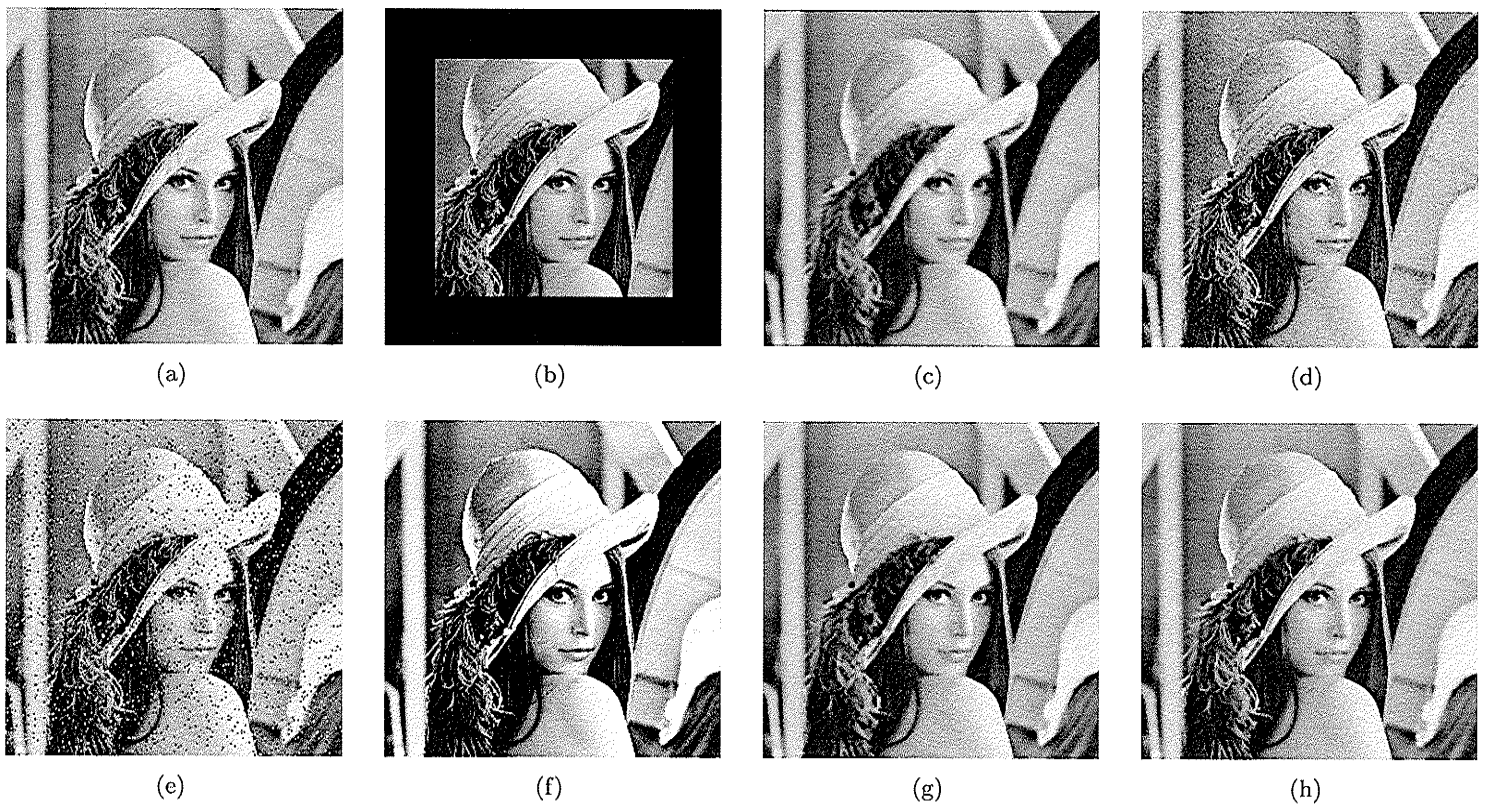


Figure 3.16: Attack examples. (a) JPEG lossy compression, QF=30. (b) Cropping, 50%. (c) Gaussian filtering, $5 \times 5, \sigma_g = 1$. (d) Gaussian noise, $\sigma = 10$. (e) Salt & pepper noise, $D = 0.05$. (f) Histogram equlization. (g) Median filtering. (h) Wiener filtering.

factors. The embedded data is estimated by the watermark extraction algorithm possibly with errors from the compressed watermarked images. Another objective of this experiment is to see the relationship between watermark robustness and the value of M. For this purpose, we take $M \in \{2, 4, 16, 256, 65536\}$.

Shown in Fig. 3.17 are a family of curves of error performance as a function of JPEG quality factors. Each point on the curves is obtained as the average value of 100 independent experiments, each of which has a different random sequence of 64 bits as its data input. From Fig. 3.17 one can see that with the increase of quality factor, BER drops monotonically. An important trend is that the value of M influence BER significantly, in particular, a larger M gives a lower BER. This result evidently shows that M-ary modulation is preferable in the design of a multibit spread spectrum-based watermarking system.

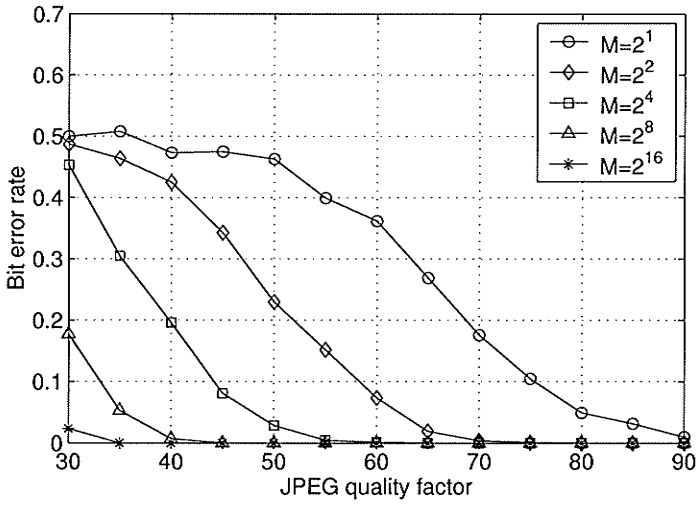


Figure 3.17: The error performance of the multibit watermarking system based on M-ary modulation plus CDMA, under JPEG lossy compression. The number of bits embedded is 64, and the quality of watermarked images is PSNR = 40dB.

Watermark Robustness to Image Cropping

Image cropping refers to the loss of some parts of an image, especially along the borders. An example of image cropping is illustrated in Fig. 3.16(b). Image cropping brings about the partial

loss of watermark information. The objective of this experiment is to look at the system's ability to recover the embedded data from incomplete watermarked images. Preferably the embedded data can be extracted at a low error rate under mild image cropping. In our experiments, we crop the watermarked images evenly along the four borders to different degrees, and record the errors in data extraction from the cropped images. The amount of data embedded is 128 bits. Shown in Fig. 3.18 is a family of BER curves, with $M \in \{2^4, 2^8, 2^{16}\}$, as a function of the remaining factor, which is the ratio of the number of remaining pixels to that of original pixels. From the figure, one can see that the watermark has outstanding robustness to image cropping, especially when $M = 2^{16}$. Even if 75% of the image pixels are cropped, the embedded data can still be extracted with very low BER at the magnitude $O(10^{-3})$.

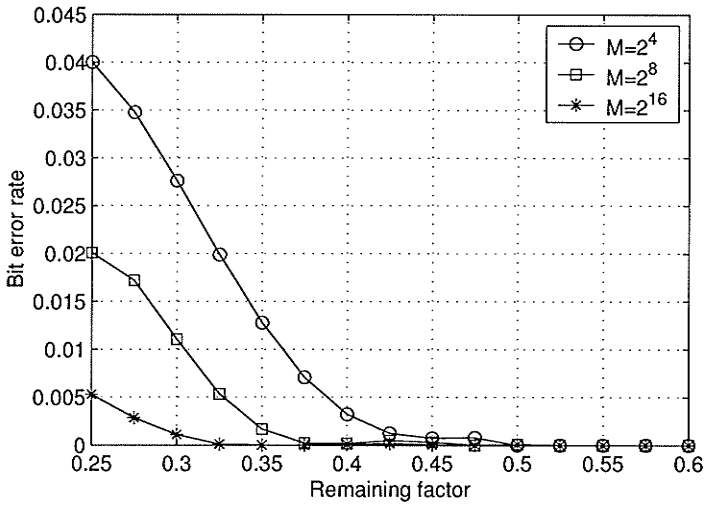


Figure 3.18: The error performance of the watermark under image cropping. The number of bits embedded is 128, and the quality of watermarked images is PSNR = 40dB.

Watermark Robustness to Lowpass Filtering

Lowpass filtering is another common form of image processing, which can be performed conveniently either in a transform domain or directly in space domain [28]. Here we use a Gaussian filter to test the watermark robustness to this kind of attack against watermarked images. One such attack example is illustrated in Fig. 3.16(c). We apply 2^{16}-ary phase modulation, set the length of data to

be 128 bits and PSNR=40dB in all the experiments. Fig. 3.19(a) shows the test results in the cases of 3×3 and 4×4 filter sizes, while the results for 5×5 Gaussian filters are given in Fig. 3.19(b). The standard deviation of the Gaussian filter is chosen to cover a wide range: $0.5 < \sigma_g < 2$. The results are the average of 1000 repetitions. In all our experiments, BER=0 in the case of 3×3 filters regardless of σ_g, and BER=0 if $\sigma_g \leq 1.5$ in the cases of 4×4 and 5×5 filters. These results indicate that the designed watermark has outstanding robustness against the attack of lowpass filtering.

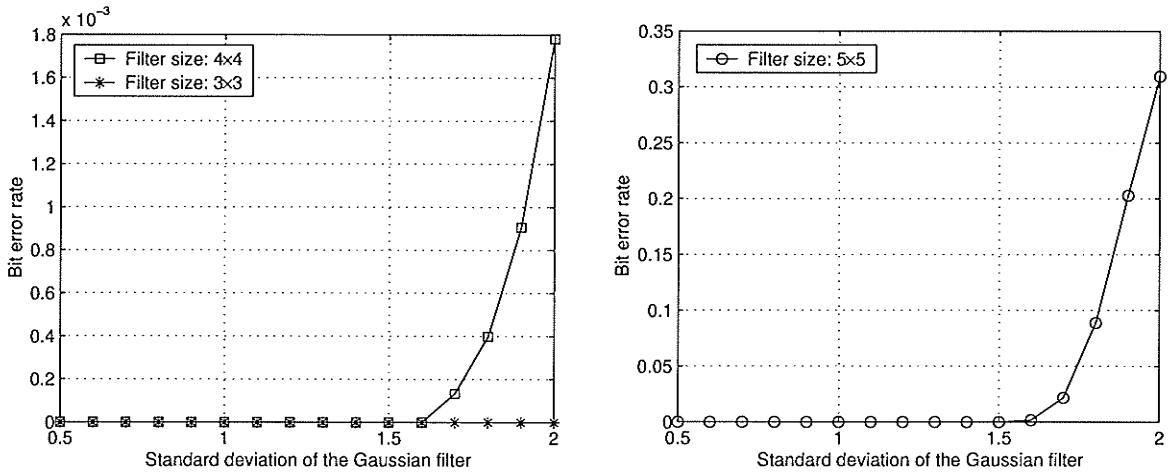


Figure 3.19: The error rate as a function of the standard deviation of Gaussian filter. (a) $\sigma_g = 3$ and 4. (b) $\sigma_g = 5$.

Watermark Robustness to Other Attacks

Besides the attacks considered above, we are concerned about the watermark robustness to some other kinds of attacks as well. A set of common image manipulations, including noise addition and image enhancement operations which are illustrated in Fig. 3.16, are applied to the watermarked images in order to test the watermark robustness. Table 3.1 lists the error rates under these attacks. Throughout all the tests, we use 2^{16}-ary phase modulation, embed 128 bits of data and make PSNR=40dB. The table shows the embedded data are robust enough against most commonly used image processing operations.

Table 3.1: Watermark robustness to other common attacks

Type of attack	Parameter of attack	BER
White Gaussian noise	$\sigma = 5$	0
	$\sigma = 10$	0
	$\sigma = 15$	0
Salt & pepper noise	$D = 0.01$	0
	$D = 0.03$	0
	$D = 0.05$	3.33×10^{-2}
Histogram equalization	N/A	0
Median filtering	f. size=2×2	0
	f. size=3×3	0
	f. size=4×4	5.89×10^{-2}
Wiener filtering	f. size=2×2	0
	f. size=3×3	0
	f. size=4×4	7.19×10^{-4}
Sharpening (in Paintshop Pro)	Moderate	0
	High	0

3.10 Chapter Summary

In this chapter, we have focused on the design of a multibit watermarking system based on spread spectrum technique. Starting from the simplest 1-bit watermark, several techniques have been introduced to construct an n-bit watermark, including FDMA, CDMA, and M-ary modulation. Compared with an FDMA/CDMA-based watermark, an M-ary modulation-based watermark is advantageous because it is capable of carrying more information with the same watermark energy. Conventional M-ary modulation has been limited to $M \leq 256$, due to the heavy computation associated with the correlation-based signal detection. However, with the proposed M-ary phase modulation, which is based on the circular shifts of a reference PNS, the amount of computation in watermark detection is drastically reduced. Furthermore, we also provide the design of an extended M-ary phase modulated watermark based on a set of windowed circular shifts of a PNS of length M, which breaks the restriction on the value of M due to the length of the feature vector. A practical design of a multibit watermark based on M-ary phase modulation plus CDMA has been presented. The simulation results show that M-ary phase modulation has greatly improved the tradeoff among a watermark's transparency, robustness and information capacity.

Chapter 4

Geometrically Robust Image Watermarking in Cartesian Coordinates

4.1 Introduction

In Chapter 2 and Chapter 3, we have elaborated on some techniques and algorithms that can be used to enhance the watermark robustness, and given an example of a coherent watermarking system. With those techniques, we have achieved good watermark robustness to a variety of common attacks, such as lossy compression and lowpass filtering. Such robustness makes the watermark applicable in some situations. It is worth noting that the robustness of a coherent watermarking system necessitates the synchronization of the watermarked feature vector and the reference PNS pattern, i.e., each element of the watermarked feature vector has been modified by the same-position element of the reference PNS pattern. However, in applications where a watermarked image has to undergo geometric distortions, the embedded data are likely to be lost. By geometric distortion, we mean that an image object is changed geometrically. This category of distortions includes

some common ones like image rotation, scaling, and flipping, and some other uncommon ones like shearing, aspect ratio change, and even random warping. The watermark designed in Chapter 3 does not survive geometric distortions for two reasons. Firstly, under geometric distortion the host features like DCT coefficients are changed wildly and irregularly. Secondly, a geometric distortion damages inevitably the synchronization which is crucial for the correct detection of the spread spectrum watermark. It is a well known fact that many existing image watermarking algorithms are vulnerable to geometric attacks [62]. Despite the efforts and progress made in this direction, the watermark robustness to geometric distortions has not been well addressed, and it still remains an open problem. In this chapter and Chapter 5, we deal with this problem and provide solutions by taking advantage of the invariance property of circularly orthogonal moments, i.e., Zernike moments and pseudo-Zernike moments.

4.2 Existing Approaches: a Brief Overview

Some research has been done to deal with the watermark's vulnerability to geometric distortions. A variety of methods have been proposed [56, 40, 23, 60, 2, 1, 3, 47, 25, 73, 24, 38, 80, 81], which are summarized and categorized as follows.

4.2.1 Distortion Inversion

This is an intuitive approach. In order to combat geometric manipulations, a registration pattern is inserted into the host signal along with the watermark [60, 23], or the watermark is designed with a special structure [40], so that in the stage of watermark detection the involved geometric distortions can be identified and measured, and thus the distortions can be removed by an inversion process. The nature of this approach is the incorporation of two watermarks, one for data payload and the other for distortion detection. This may bring two issues. On the one hand, the existence of a second watermark either causes additional distortion of the cover signal or decreases the amount of data payload. On the other hand, for correct watermark extraction, both watermarks must be

robust, which is often difficult to achieve.

4.2.2 Image Normalization

The second category is based on image normalization [1, 24]. An image can be normalized to a certain position, orientation and size [67], which are invariant to image translation, rotation and scaling, respectively. The host image is normalized prior to watermark insertion, and the image is denormalized back to its original look after watermark insertion. At the watermark extractor, the watermarked image has to undergo the same normalization process before watermark detection. An outstanding disadvantage of this approach is that an image has to experience transformations twice in the watermarking process, which inevitably causes extra quality degradation of the image on top of the watermark-induced distortion. Another problem is that the image distortion due to the insertion of the watermark and subsequent attacks is likely to result in a slight change of the normalized position, orientation or size, which is a fatal issue for the synchronization-sensitive watermark detection.

4.2.3 Invariant Watermarking

The third category, in which we are most interested, is based on the invariance properties of some image features. Different image features have different invariance properties. Image features that have been used for invariance watermarking include Fourier-Mellin transform domain coefficients, geometric moment invariants and Zernike moments.

Watermarking in Fourier-Mellin Domain

O'Ruanaidh *et al.* first reported rotation-scaling-translation (RST) invariant watermarks in the Fourier-Mellin domain [56]. Given an image $f(k,l), k = 1,...,N, l = 1,...,N$, a discrete Fourier transform modulus (DFTM) is taken first to form a new function

$$f_1(u,v) = \left| \sum_{k=1}^{N} \sum_{l=1}^{N} f(k,l) \exp[-j2\pi(uk+vl)/N] \right|, \; u,v = 1,...,N. \tag{4.1}$$

Subsequently a log-polar transform is performed on $f_1(u,v)$, resulting in another function $f_2(p,q) = f_1(u,v)$, where (p,q) and (u,v) are related by

$$\begin{cases} u = e^p \cos(q) \\ v = e^p \sin(q) \end{cases} . \tag{4.2}$$

Then another DFTM with respect to $f_2(p,q)$ is taken

$$f_3(s,t) = \left| \sum_{p=1}^{N} \sum_{q=1}^{N} f_2(p,q) \exp[-j2\pi(ps+qt)/N] \right|, \ s,t = 1,...,N. \tag{4.3}$$

The resulting function $f_3(s,t)$ is RST invariant, because $f_1(u,v)$ is translation-invariant due to the translation-invariant property of the DFT [28], and the transform (4.2) converts image rotation and scaling into translations in the (p,q)-plane, which are made invariant by another DFT-modulus operation. The RST invariant coefficients $f_3(s,t)$ are the watermarking space. Although this approach seems flawless in theory, the involved forward and inverse log-polar transforms pose an implementation difficulty [56] and lead to severe image degeneration, which is a big limitation in watermarking applications.

Watermarking via Geometric Moment Invariants

Using geometric moments, one can obtain two sets of invariants. One set consists of seven invariants to orthogonal transformations, including rotation, scaling, and flipping [31]:

$$\psi_1 = \eta_{20} + \eta_{02} \tag{4.4}$$

$$\psi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{4.5}$$

$$\psi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{4.6}$$

$$\psi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{4.7}$$

$$\begin{aligned}\psi_5 =& (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{4.8}$$

$$\psi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{4.9}$$

$$\begin{aligned}\psi_7 =& (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &- (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].\end{aligned} \tag{4.10}$$

The other set is composed of four affine transformation invariants [31, 66]:

$$I_1 = (\mu_{20}\mu_{02} - \mu_{11}^2)/\mu_{00}^4, \tag{4.11}$$

$$I_2 = (\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{03}\mu_{21}^3 - 3\mu_{21}^2\mu_{12}^2)/\mu_{00}^{10}, \tag{4.12}$$

$$I_3 = (\mu_{20}\mu_{21}\mu_{03} - \mu_{20}\mu_{12}^2 - \mu_{11}\mu_{30}\mu_{03} + \mu_{11}\mu_{21}\mu_{12} + \mu_{02}\mu_{30}\mu_{12} - \mu_{02}\mu_{21}^2)/\mu_{00}^7, \tag{4.13}$$

$$\begin{aligned}I_4 =& (\mu_{20}^3\mu_{03}^2 - 6\mu_{20}^2\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{02}\mu_{21}\mu_{03} + 9\mu_{20}^2\mu_{02}\mu_{12}^2 - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} \\ &- 8\mu_{11}^3\mu_{30}\mu_{03} - 6\mu_{20}\mu_{02}^2\mu_{30}\mu_{12} + 12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} + 9\mu_{20}\mu_{02}^2\mu_{21}^2 \\ &+ 12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} - 6\mu_{11}\mu_{02}^2\mu_{30}\mu_{21} + \mu_{02}^3\mu_{30}^2)/\mu_{00}^{11}.\end{aligned} \tag{4.14}$$

The η's in the above equations denote the normalized central moments of an image $f(x, y)$:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{1+(p+q)/2}}, \; p + q > 2, \tag{4.15}$$

where μ_{pq} is the central moment, defined by

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x - \bar{x})^p(y - \bar{y})^q f(x, y)dxdy, \tag{4.16}$$

where $(\bar{x}, \bar{y})$ is the centroid of the image $f(x, y)$.

These two sets of invariants can be employed for image watermarking [2, 59]. It is a well-known fact that although $\{x^p y^q\}$ is a complete basis set, it is not orthogonal [18]. This leads to difficulty of direct watermark embedding in the space of geometric moments. To circumvent this problem, one can use the orthogonal Legendre moments [18] to embed the watermark, and use the aforementioned invariants to detect the watermark, as performed in [59].

However, there exist some limitations in this approach, which are listed below.

- Due to the small number of invariants, it is difficult to embed a multibit watermark, although it is possible to embed a zero-bit watermark.
- The nonlinear nature of the invariants makes them potentially unstable. A small error in the computation of μ_{pq} may result in unpredictable error of the invariants.
- Geometric moments, or central moments in (4.16), are defined for analog images, and so are the invariants $\psi_1, ..., \psi_7$ and $I_1, ..., I_4$. For digital images, the invariance property is often compromised to some extent due to numerical errors involved in moment computation [43, 44].

Watermarking via Zernike Moments

Zernike moments [69, 70, 37] are a type of orthogonal moments defined on the unit disk with an attractive property that their magnitudes are invariant to image rotation, which is potentially suitable for invariant image watermarking. However, so far only some primitive research in this direction has been reported in literature.

Zernike moments were first reported as watermark features by Farzam *et al.* [25] to achieve watermark robustness to rotation, additive noise and JPEG compression. An image is divided into some concentric regions, whose Zernike moments are subsequently modulated by the watermark signal. One of the weaknesses of this approach is that the cumulative geometric errors along the borders of the concentric regions inevitably lead to severe image degradation, making the quality

of the watermarked image unacceptable.

More recently, in [38] the authors proposed embedding a zero-bit watermark by modifying a feature vector consisting of Zernike moments with orders below 5. One problem of this approach is that it is hardly possible to adapt the algorithm to multibit watermarking, because only a very small number of moments were used.

In these two approaches, the accuracy issue of moment computation [43, 45, 58], which is crucial for digital images, was disregarded. As to be shown in this chapter, the rotational invariance property of Zernike moments are not uniformly ideal, and it is even not valid for some Zernike moments. Without consideration of this problem, the usefulness of Zernike moments for invariant watermarking has to be compromised.

In the remainder of this chapter, we consider in depth the accuracy issue of Zernike moments, and accordingly propose a multibit watermarking algorithm, which achieves an impressive performance. Along with Zernike moments, pseudo-Zernike moments, another kind of orthogonal moments on the unit disk, are also studied for image watermarking. Both theoretical and experimental results are presented, which are based on our recent research [80, 81, 82, 84].

4.3 Zernike Moments and Pseudo-Zernike Moments

4.3.1 Zernike/Pseudo-Zernike Functions

The Zernike basis is a set of complete and orthogonal functions on the unit disk $\mathbb{D} = \{(x, y) : x^2 + y^2 \leq 1\}$, defined by [90, 9]:

$$V_{pq}(x, y) = R_{pq}(\rho)e^{jq\theta}, \tag{4.17}$$

where $\rho = \sqrt{x^2 + y^2}$, $\theta = \tan^{-1}(y/x)$. Here p is a non-negative integer and q is an integer that takes positive, negative , or zero values such that $p - |q|$ is even and non-negative. The radial Zernike polynomial $R_{pq}(\rho)$ is defined by the following formula

$$R_{pq}(\rho) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)! \rho^{p-2s}}{s!(\frac{p+|q|}{2} - s)!(\frac{p-|q|}{2} - s)!}. \tag{4.18}$$

A modified version of Zernike basis is the so-called pseudo-Zernike basis [6, 70], which is also a set of complete and orthogonal functions on the unit disk $\mathbb{D}$, and has the same form of definition as (4.17) with two exceptions. One is that q is only restricted to be $|q| \leq p$, and the other is that the radial polynomial $R_{pq}(\rho)$ is defined differently by

$$R_{pq}(\rho) = \sum_{s=0}^{p-|q|} \frac{(-1)^s (2p+1-s)! \rho^{p-s}}{s!(p+|q|+1-s)!(p-|q|-s)!}. \tag{4.19}$$

Both Zernike basis and pseudo-Zernike basis satisfy the following orthogonality condition:

$$\int\int_{\mathbb{D}} V_{pq}(x,y) V^*_{p'q'}(x,y) dxdy = \frac{\pi}{p+1} \delta_{pp'} \delta_{qq'}, \tag{4.20}$$

where the asterisk denotes complex conjugate.

4.3.2 Zernike/Pseudo-Zernike Moments

Like any other orthogonal and complete basis, the Zernike/Pseudo-Zernike basis can be used to decompose an analog image function $f(x,y)$:

$$f(x,y) = \sum_{p=0}^{\infty} \sum_{\{\text{permissible } q\}} A_{pq} V_{pq}(x,y), \tag{4.21}$$

where A_{pq} is the Zernike moment (ZM) or Pseudo-Zernike moment (PZM) of order p with repetition q, which is defined by

$$A_{pq} = \frac{p+1}{\pi} \int\int_{\mathbb{D}} f(x,y) V^*_{pq}(x,y) dxdy. \tag{4.22}$$

For digital images, (4.22) cannot be applied directly. Assume an $N \times N$ image matrix $F(i,j), i = 1, ..., N, j = 1, ..., N$ is given. In order to compute its ZMs/PZMs, one has to first map $F(i,j)$ into the function $f(x_i, y_j)$ defined in $[1,1] \times [1,1]$, such that $f(x_i, y_j) = F(i,j), i = 1, ..., N, j = 1, ..., N$, where $x_i = (2i - N - 1)/N$ and $y_j = (2j - N - 1)/N$. Viewing $f(x_i, y_j)$ as a pixel-wise constant image function, one can approximate (4.22) by [43, 45, 58]

$$\check{A}_{pq} = \frac{p+1}{\pi} \sum_i \sum_j \nu_{pq}(x_i, y_j) f(x_i, y_j), \tag{4.23}$$

where the values of i and j are taken such that $(x_i, y_j) \in \mathbb{D}$, and

$$\nu_{pq}(x_i, y_j) = \int_{x_i-\frac{\lambda}{2}}^{x_i+\frac{\lambda}{2}} \int_{y_j-\frac{\lambda}{2}}^{y_j+\frac{\lambda}{2}} V_{pq}^*(x, y)dxdy, \tag{4.24}$$

where $\lambda = \frac{2}{N}$ is the pixel width or height. For the computation of the factor $\nu_{pq}(x_i, y_j)$, the most commonly used formula, which is also the simplest one, is

$$\nu_{pq}(x_i, y_j) \approx \lambda^2 V_{pq}^*(x_i, y_j). \tag{4.25}$$

Some more complex numerical techniques, such as the multidimensional cubature formulas introduced in [43, 45] can be used to improve the accuracy of the factor $\nu_{pq}(x_i, y_j)$.

We have to point out that the ZMs/PZMs of discrete images computed via (4.23) and (4.25) are not accurate in a strict sense. As analyzed in [43, 45, 58], there exist two sources of errors, namely geometric error and numerical integration error, in the computation of ZMs/PZMs. The former is due to the fact that the areas covered by the pixels involved in the computation of moments never sum up exactly to the area of the unit disk. The latter comes from the approximation of $\nu_{pq}(x_i, y_j)$ via a formula like (4.25).

Due to the close similarity of Zernike moments and pseudo-Zernike moments, we do not differentiate between their notations in this thesis. We use A_{pq} to denote both ZMs and PZMs, and $V_{pq}(\cdot)$ to denote both Zernike polynomials and pseudo-Zernike polynomials. Furthermore, since we are primarily concerned with digital images, we deal with $\check{A}_{pq}$ defined in (4.23), rather than A_{pq} defined in (4.22), and therefore we shall simply use A_{pq} to denote $\check{A}_{pq}$, unless otherwise stated.

4.3.3 The Invariance Properties of ZMs/PZMs

The reason we use ZMs/PZMs for image watermarking is that they have some very important properties, i.e., their magnitudes are invariant under image rotation and image flipping. We now elaborate on these properties.

Rotation Invariance

If an image $f(x,y)$ is rotated α degrees counterclockwise, it can be shown that the ZMs/PZMs of the resulting image are

$$A_{pq}^{(\alpha)} = A_{pq} e^{-jq\alpha}, \tag{4.26}$$

where A_{pq} and $A_{pq}^{(\alpha)}$ denote the ZMs/PZMs of the original image $f(x,y)$ and the rotated image, respectively. This leads to $|A_{pq}^{(\alpha)}| = |A_{pq}|$. Therefore, if a watermark is inserted in the magnitudes of ZMs/PZMs, it is robust to rotation. We have to note that this property holds for analog images, however, for discrete images whose ZMs/PZMs are typically computed via (4.23) and (4.25), this property has to be compromised to some extent. We have a detailed analysis in the next section.

Flipping Invariance

Another interesting property of ZMs/PZMs is about image flipping, either horizontal or vertical. If we define the horizontally flipped version of the image $f(x_u, y_v)$ as $f^{(\text{hf})}(x_u, y_v) = f(-x_u, y_v)$ whose ZMs/PZMs are denoted by $A_{pq}^{(\text{hf})}$, and the vertically flipped version of the image $f(x_u, y_v)$ as $f^{(\text{vf})}(x_u, y_v) = f(x_u, -y_v)$ whose ZMs/PZMs are denoted by $A_{pq}^{(\text{vf})}$, then

$$\begin{aligned}
A_{pq}^{(\text{hf})} &= k \sum_{\{(x_u,y_v)\in\mathbb{D}\}} V_{pq}^*(x_u, y_v) f^{(\text{hf})}(x_u, y_v) \\
&= k \sum_{\{(x_u,y_v)\in\mathbb{D}\}} R_{pq}(\rho) e^{-jq\theta} f(-x_u, y_v) \\
&= k \sum_{\{(x_u,y_v)\in\mathbb{D}\}} R_{pq}(\rho) e^{-jq(\pi-\theta)} f(x_u, y_v) \\
&= k \sum_{\{(x_u,y_v)\in\mathbb{D}\}} (-1)^q R_{pq}(\rho) e^{jq\theta} f(x_u, y_v) \\
&= (-1)^q A_{pq}^*
\end{aligned} \tag{4.27}$$

where $k = \frac{(p+1)\lambda^2}{\pi}$. Similarly,

$$\begin{aligned} A_{pq}^{(\mathrm{vf})} &= k \sum_{\{(x_u,y_v)\in\mathbb{D}\}} V_{pq}^*(x_u,y_v) f^{(\mathrm{vf})}(x_u,y_v) \\ &= k \sum_{\{(x_u,y_v)\in\mathbb{D}\}} R_{pq}(\rho) e^{-jq\theta} f(x_u,-y_v) \\ &= k \sum_{\{(x_u,y_v)\in\mathbb{D}\}} R_{pq}(\rho) e^{-jq(-\theta)} f(x_u,y_v) \\ &= k \sum_{\{(x_u,y_v)\in\mathbb{D}\}} R_{pq}(\rho) e^{jq\theta} f(x_u,y_v) \\ &= A_{pq}^*. \end{aligned} \tag{4.28}$$

In either case, the magnitudes of ZMs/PZMs do not change, i.e., $|A_{pq}^{(\mathrm{hf})}| = |A_{pq}|$ and $|A_{pq}^{(\mathrm{vf})}| = |A_{pq}|$. This property is of significance in watermarking applications, since if a watermark is embedded into ZMs/PZMs, it is robust to image flipping. Image flipping is an easy and effective form of attack to which many existing watermarking algorithms are vulnerable.

Scaling Invariance

Although in theory ZMs/PZMs are not invariant to image scaling, we can still obtain scaling invariance in practice, either by changing the unit disk region accordingly or resizing the image to a canonical size, provided that the unit disk is made to cover the same contents of the image. It is worth noting that this property holds well only when the image is scaled moderately. When the image is scaled to a much smaller size, this property is compromised due to the loss of information. In watermarking scenarios, it is often reasonable to assume that the images are scaled only slightly, because otherwise the value of the host signal would be lost.

4.4 Non-ideal Invariance of ZMs/PZMs of Digital Images

As stated above, for a digital image, due to geometric error and numerical integration error, ZMs/PZMs cannot be computed accurately. This inaccuracy of ZMs/PZMs has a negative im-

pact on the aforementioned rotation-invariance property, making it hold only approximately. Our concern is how close this approximation is. We argue that the magnitude invariance of a particular ZM/PZM depends on its computation accuracy. We have observed that computed with (4.23) and (4.25), different moments have different levels of computation accuracy. To see this clearly, let us look at an example. Assume we have an 128×128 image of a constant graylevel 127. Based on (4.23) and (4.25), we have calculated its ZMs and PZMs up to order 12, whose magnitudes are shown in Table 4.1 and Table 4.2, respectively. In theory, all the ZMs/PZMs of this image except A_{00} should be zero, but in fact we can see from these tables that a number of moments deviate from zero, and some of them even have considerable magnitudes. In general, we have the following results.

Theorem 4.1 *The Zernike/pseudo-Zernike moments of a constant image $f(x_i, y_j) = T$, computed via (4.23) and (4.25), are*

$$A_{pq} = \begin{cases} T(1 + O(\lambda^\gamma)), & \text{if } p = q = 0 \\ \text{nonzero}, & \text{if } p \neq 0 \text{ and } q = 4i, i \in \text{integer} \\ 0, & \text{otherwise,} \end{cases}$$

where λ is the pixel width, and $1 \leq \gamma < 3/2$ is the exponent characterizing the geometric error.

The proof of this theorem can be found in Appendix A.4. This theorem is significant for invariant watermarking with intended robustness against geometric distortions. From this theorem, we know that not all the ZMs/PZMs of a discrete image can be accurately computed. When $q = 4i$, $i \in$ integer, A_{pq} is not accurate, and therefore $|A_{pq}|$ is not invariant to image rotation. Because of this, those A_{pq}'s with $q = 4i$ are not suitable for invariant watermarking.

4.5 Watermark Embedding

Based on the above conclusion about the rotational invariance of ZMs/PZMs, we can design a watermark with robustness to image rotation, flipping, and scaling. As a rule of thumb, one

Table 4.1: Magnitudes of Zernike moments up to order 12 with $m \geq 0$ for a 128×128 constant image

	q=0	1	2	3	4	5	6	7	8	9	10	11	12
p= 0	127.24												
1		0.0000											
2	0.7083		0.0000										
3		0.0000		0.0000									
4	1.1689		0.0000		0.0049								
5		0.0000		0.0000		0.0000							
6	1.6117		0.0000		0.0004		0.0000						
7		0.0000		0.0000		0.0000		0.0000					
8	2.0287		0.0000		0.0136		0.0000		0.2140				
9		0.0000		0.0000		0.0000		0.0000		0.0000			
10	2.4110		0.0000		0.0375		0.0000		0.2169		0.0000		
11		0.0000		0.0000		0.0000		0.0000		0.0000		0.0000	
12	2.7487		0.0000		0.0757		0.0000		0.1900		0.0000		0.0153

Table 4.2: Magnitudes of pseudo-Zernike moments up to order 12 with $q \geq 0$ for a 128×128 constant image

	q=0	1	2	3	4	5	6	7	8	9	10	11	12
p= 0	127.24												
1	0.4728	0.0000											
2	0.7044	0.0000	0.0000										
3	0.9329	0.0000	0.0000	0.0000									
4	1.1482	0.0000	0.0000	0.0000	0.0049								
5	1.3682	0.0000	0.0000	0.0000	0.0002	0.0000							
6	1.5486	0.0000	0.0000	0.0000	0.0080	0.0000	0.0000						
7	1.7667	0.0000	0.0000	0.0000	0.0203	0.0000	0.0000	0.0000					
8	1.8816	0.0000	0.0000	0.0000	0.0377	0.0000	0.0000	0.0000	0.2140				
9	2.1156	0.0000	0.0000	0.0000	0.0612	0.0000	0.0000	0.0000	0.1993	0.0000			
10	2.1188	0.0000	0.0000	0.0000	0.0918	0.0000	0.0000	0.0000	0.1704	0.0000	0.0000		
11	2.4006	0.0000	0.0000	0.0000	0.1305	0.0000	0.0000	0.0000	0.1252	0.0000	0.0000	0.0000	
12	2.2284	0.0000	0.0000	0.0000	0.1785	0.0000	0.0000	0.0000	0.0614	0.0000	0.0000	0.0000	0.0153

can circumvent the non-ideal property of invariance by avoiding those inaccurate ZMs/PZMs for watermarking. The structure of the watermark embedder are depicted in Fig. 4.1. The main ideas are explained below.

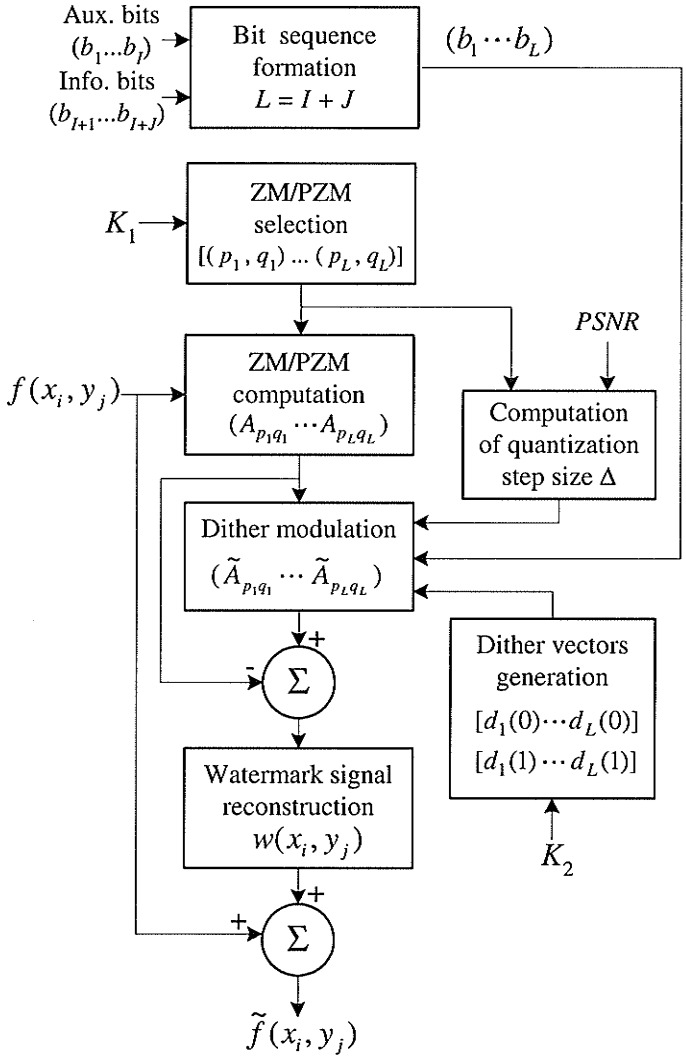


Figure 4.1: The structure of the watermark embedder

4.5.1 Structure of Embedded Bit Sequence

The embedded bit array consists of two parts, as indicated by Fig. 4.2. The first part is an I-bit auxiliary sequence $(b_1 \ldots b_I)$, which is followed by the second part, a J-bit informative sequence $(b_{I+1} \ldots b_{I+J})$. The auxiliary part is a fixed bit sequence, known to both the watermark embedder

and watermark extractor, whose purpose is to facilitate the determination of the unit disk. Due to the possible geometric distortions, such as image resizing, it is crucial for the watermark extractor to have the exact knowledge of the image region covered by the unit disk in order for the informative bits to be extracted. We address this issue by embedding the fixed auxiliary bit sequence $(b_1...b_I)$. When the adopted region is not the correct unit disk region, the extracted auxiliary bit sequence $(b'_1...b'_I)$ displays randomness, and thus does not agree well with the embedded sequence $(b_1...b_I)$. On the contrary, if the unit disk region is adopted correctly, the two sequences match well. It can be shown that the probability that a uniformly distributed random bit sequence $(b'_1...b'_I)$ matches the fixed sequence $(b_1...b_I)$ at more than H bit positions is

$$P(I,H) = 2^{-I} \sum_{i=H}^{I} \binom{I}{i}. \tag{4.29}$$

For example, if the given sequence is 16 bit long, of which 14 bits are matched by an extracted bit sequence, we are almost sure that the correct unit disk region is located, because $P(16,14) \approx 2.1 \times 10^{-3}$, meaning that the probability that it is not the unit disk region is only 2.1×10^{-3}.

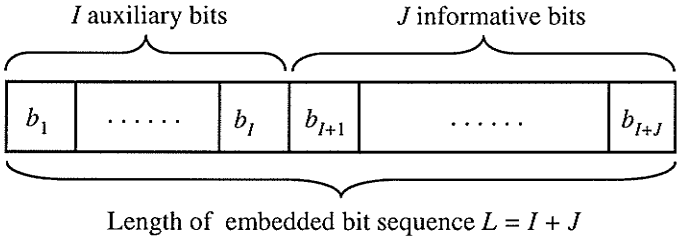


Figure 4.2: The structure of the embedded bit sequence

4.5.2 Selection of ZMs/PZMs

The following factors have to be taken into account in selection of ZMs/PZMs for data hiding.

- As shown in the previous section, the combination of geometric error and numerical integration error makes the computation of some ZMs/PZMs inaccurate, thus compromising their invariance property. As a result, some ZMs/PZMs can be computed more accurately, hence

more suitable for invariant data hiding than others. According to Theorem 4.1, all A_{pq}'s with repetition $q = 4i$, $i \in$ integer, have to be ruled out for data hiding.

- Due to rounding errors which become increasingly significant as the the order goes up, there exists a certain value $p_{\max}$, which makes A_{pq} with $p > p_{\max}$ inaccurate, even if $q \neq 4i$. Therefore only those A_{pq}'s with $p \leq p_{\max}$ are reliable and selected for data hiding.

- Because of the conjugate symmetry $A_{pq}^* = A_{p,-q}$, only about half of ZMs/PZMs have independent magnitudes, and in practice we only choose those A_{pq}'s with $q \geq 0$.

Considering all these factors, the set of applicable ZMs/PZMs can be denoted by $\mathbb{S} = \{A_{pq}, p \leq p_{\max}, q \geq 0, q \neq 4i\}$. The cardinalities of $\mathbb{S}$, denoted by $|\mathbb{S}|_{\text{ZM}}$ and $|\mathbb{S}|_{\text{PZM}}$ in the cases of ZMs and PZMs, respectively, can readily be obtained with straightforward algebra:

$$|\mathbb{S}|_{\text{ZM}} = \begin{cases} \frac{3p_{\max}^2+8p_{\max}}{16} & \text{if } p_{\max} = 4i \\ \frac{3p_{\max}^2+10p_{\max}+3}{16} & \text{if } p_{\max} = 4i+1 \\ \frac{3p_{\max}^2+8p_{\max}+4}{16} & \text{if } p_{\max} = 4i+2 \\ \frac{3p_{\max}^2+10p_{\max}+7}{16} & \text{if } p_{\max} = 4i+3 \end{cases}, \tag{4.30}$$

and

$$|\mathbb{S}|_{\text{PZM}} = \begin{cases} \frac{3p_{\max}^2+6p_{\max}}{8} & \text{if } p_{\max} = 4i \\ \frac{3p_{\max}^2+6p_{\max}-1}{8} & \text{if } p_{\max} = 4i+1 \\ \frac{3p_{\max}^2+6p_{\max}}{8} & \text{if } p_{\max} = 4i+2 \\ \frac{3p_{\max}^2+6p_{\max}+3}{8} & \text{if } p_{\max} = 4i+3 \end{cases}, \tag{4.31}$$

where i is any nonnegative integer.

4.5.3 Modification of ZMs/PZMs

We apply dither modulation for the modification of ZMs/PZMs. Dither modulation is a special form of quantization index modulation, which was first proposed for data hiding in [13]. With a

base quantizer $\mathcal{Q}(\cdot)$, the dither modulation function is defined as

$$f_{\mathrm{DM}}(x, m) = \mathcal{Q}(x - d(m)) + d(m) \tag{4.32}$$

where x is a scalar variable to be quantized, m is a message symbol to be embedded in x, and $d(m)$ is the dither scalar associated with m. Dither modulation has the property that the quantization cells and reconstruction points of any given quantizer are shifted versions of the quantization cells and reconstruction points of any other quantizer. Due to this special structure of quantizers, dither modulation has the advantage of easy implementation. In our work, we adopt binary dither modulation, i.e., $m \in \{0, 1\}$, and one independent magnitude of ZM/PZM is to carry one bit information. In practice one independent magnitude of ZM/PZM can be used to carry more than one bit, or more than one independent magnitude of ZMs/PZMs to carry one bit, which we discuss later. A uniform scalar quantizer is adopted as our base quantizer $\mathcal{Q}(\cdot)$, i.e.,

$$\mathcal{Q}(x) = \mathcal{R}\left(\frac{x}{\Delta}\right)\Delta, \tag{4.33}$$

where $\mathcal{R}(\cdot)$ is the rounding operation, Δ is the step size of quantization.

Assume a bit sequence $\mathbf{b} = (b_1, ..., b_L), L \leq |\mathbb{S}|$ and $b_i \in \{0, 1\}$, is to be embedded in an image $f(x_i, y_j), i, j = 1, 2, ..., N$. For the sake of security, we use a secret key K_1 to pseudorandomly choose L ZMs/PZMs from $\mathbb{S}$ to form a moment vector $\mathbf{Z} = (A_{p_1 q_1}, ..., A_{p_L q_L})$, $A_{p_i q_i} \in \mathbb{S}$.

Now each bit from $\mathbf{b}$ is to be embedded into an element of $\mathbf{Z}$ via dither modulation. The magnitude of $A_{p_i q_i}, i = 1, ..., L$ is quantized according to (4.32) and (4.33), i.e.,

$$\begin{aligned} |\tilde{A}_{p_i q_i}| =& f_{\mathrm{DM}}(|A_{p_i q_i}|, b_i) \\ =& \mathcal{R}\left(\frac{|A_{p_i q_i}| - d_i(b_i)}{\Delta}\right)\Delta + d_i(b_i), i = 1, ..., L \end{aligned} \tag{4.34}$$

where $d_i(\cdot)$ is the dither function for the ith quantizer satisfying $d_i(1) = \frac{\Delta}{2} + d_i(0)$. The dither vector $(d_1(0), ..., d_L(0))$, whose elements are uniformly distributed over $[0, \Delta]$, is pseudorandomly generated with another key K_2, which is used to further increase the secrecy and security of the embedded signal.

As a result, a new vector $\tilde{\mathbf{Z}} = (\tilde{A}_{p_1q_1}, ..., \tilde{A}_{p_Lq_L})$ can be obtained via the following calculation:

$$\tilde{A}_{p_iq_i} = \frac{|\tilde{A}_{p_iq_i}|}{|A_{p_iq_i}|} A_{p_iq_i}, i = 1, ..., L, \tag{4.35}$$

where $\tilde{A}_{p_iq_i}$ is the dither quantized version of $A_{p_iq_i}$.

It is worth noting that in quantizing each $A_{p_iq_i}$, if $q_i \neq 0$, its conjugate $A_{p_i,-q_i}$ must be quantized simultaneously to ensure that they always have the same magnitudes, so that the reconstructed image is real.

4.5.4 Determination of Quantization Step Size

In using (4.34) to dither-modulate the selected ZMs/PZMs, we must first decide the quantization step size Δ. Basically, Δ determines the tradeoff between the visibility and robustness of a watermark. A larger Δ gives better watermark robustness, but makes the watermark more visible, and vice versa. In practice, the value of Δ can be decided by the required quality of a watermark image, which, in our work, is the peek signal-to-noise ratio (PSNR) defined by

$$\text{PSNR}(f, \tilde{f}) = 10 \log_{10} \frac{255^2}{\sigma_e^2}, \tag{4.36}$$

where f and $\tilde{f}$ are the original image and the watermarked image, respectively, both with dimensions $N \times N$, and

$$\sigma_e^2 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left\{ \tilde{f}(x_i, y_j) - f(x_i, y_j) \right\}^2 \tag{4.37}$$

is the squared error, whose relationship to Δ is expressed by the following theorem.

Theorem 4.2 *Given a selected set of ZMs/PZMs $\{A_{p_kq_k}\}_{k=1}^{L}$ of a stochastic image $f(x_i, y_j)$, which are to be dither-modulated by (4.34) to generate a watermarked image $\tilde{f}(x_i, y_j)$, the expected value of the squared error σ_e^2 defined in (4.37) is related to the quantization step size Δ by*

$$E\{\sigma_e^2\} = \frac{\pi \Delta^2}{24} \sum_{k=1}^{L} (p_k + 1)^{-1}. \tag{4.38}$$

See Appendix A.5 for the proof of this theorem. With this theorem, given a required PSNR (in dB), we can estimate the quantization step Δ by the following formula

$$\Delta = 255 \left[10^{\frac{\text{PSNR}}{10}} \frac{\pi}{24} \sum_{k=1}^{L} (p_k + 1)^{-1} \right]^{-0.5}. \tag{4.39}$$

4.5.5 Formation of the Watermarked Image

The reconstructed watermarked image is composed of two parts. One part is the image components contributed by the moments modified for watermarking:

$$f_{\tilde{\mathbf{Z}}}(x_i, y_j) = \sum_{i=1}^{L} [\tilde{A}_{p_i q_i} V_{p_i q_i}(x_i, y_j) + \tilde{A}_{p_i, -q_i} V_{p_i, -q_i}(x_i, y_j)]. \tag{4.40}$$

The other part is the image components contributed by the rest of the moments, i.e., the moments which are not modified:

$$f_{\text{rem}}(x_i, y_j) = f(x_i, y_j) - f_{\mathbf{Z}}(x_i, y_j), \tag{4.41}$$

where $f_{\mathbf{Z}}(x_i, y_j)$ is the image components contributed by the moments selected for watermarking before they are modified:

$$f_{\mathbf{Z}}(x_i, y_j) = \sum_{i=1}^{L} [A_{p_i q_i} V_{p_i q_i}(x_i, y_j) + A_{p_i, -q_i} V_{p_i, -q_i}(x_i, y_j)]. \tag{4.42}$$

Therefore we obtain the watermarked image by combining the two parts

$$\tilde{f}(x_i, y_j) = f_{\text{rem}}(x_i, y_j) + f_{\tilde{\mathbf{Z}}}(x_i, y_j). \tag{4.43}$$

Due to the linearity of the image reconstruction process, (4.40) can be rewritten as

$$\begin{aligned} f_{\tilde{\mathbf{Z}}}(x_i, y_j) &= \sum_{i=1}^{L} (A_{p_i q_j} + \varepsilon_{p_i q_j}) V_{p_i q_i}(x_i, y_j) + \sum_{i=1}^{L} (A_{p_i, -q_j} + \varepsilon_{p_i, -q_j}) V_{p_i, -q_i}(x_i, y_j) \\ &= w(x_i, y_j) + \sum_{i=1}^{L} [A_{p_i q_i} V_{p_i q_i}(x_i, y_j) + A_{p_i, -q_i} V_{p_i, -q_i}(x_i, y_j)] \end{aligned} \tag{4.44}$$

where $\varepsilon_{p_iq_i} = \tilde{A}_{p_iq_i} - A_{p_iq_i}$ and $\varepsilon_{p_i,-q_i} = \tilde{A}_{p_i,-q_i} - A_{p_i,-q_i}$ are the quantization noise signals of the moments $A_{p_iq_i}$ and $A_{p_i,-q_i}$, respectively, and

$$w(x_i, y_j) = \sum_{i=1}^{L} [\varepsilon_{p_iq_i} V_{p_iq_i}(x_i, y_j) + \varepsilon_{p_i,-q_i} V_{p_i,-q_i}(x_i, y_j)] \tag{4.45}$$

is the reconstructed watermark signal, which results from the quantization noise of the selected ZMs/PZMs.

Therefore (4.43) turns into

$$\begin{aligned} \tilde{f}(x_i, y_j) &= f_{\text{rem}}(x_i, y_j) + f_{\mathbf{Z}}(x_i, y_j) + w(x_i, y_j) \\ &= f(x_i, y_j) + w(x_i, y_j). \end{aligned} \tag{4.46}$$

4.6 Data Extraction

The process of watermark extraction is shown in Fig. 4.3. Suppose there is a test image $f'(x_i, y_j)$, which is a distorted version of $\tilde{f}(x_i, y_j)$ after some possible manipulations, such as rotation and scaling. Our goal is to get an estimate of the hidden bit sequence from $f'(x_i, y_j)$ at a low error rate. First, with the same key K_1 as in the process of watermark insertion, the identities of ZMs/PZMs involved in the data embedding process can be determined, which is denoted by $[(p_1, q_1), ..., (p_L, q_L)]$. The subsequent data extraction process can be described by the following two steps.

4.6.1 Locating the Unit Disk Region

The exact location of the unit disk region on the test image is crucial for the extraction of the embedded data. To facilitate the search for the unit disk region, we have to use the embedded auxiliary bit sequence. For an assumed unit disk region, we extract the first I bits $(b'_1...b'_I)$, and then compare them to the bits in the known sequence $(b_1...b_I)$. As stated before, if the two sequences match well, the assumption of the unit disk region is correct. Otherwise the search process continues, until the two sequences match well.

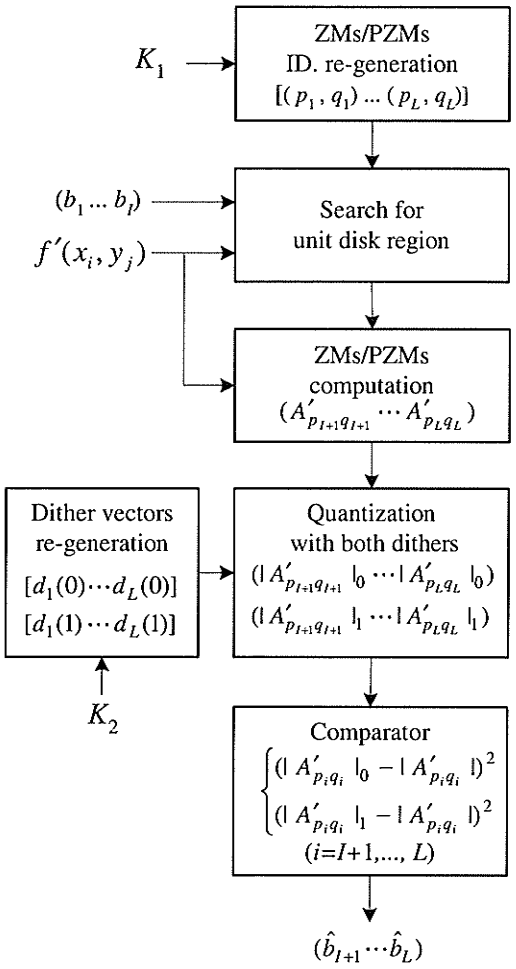


Figure 4.3: The structure of watermark extractor

4.6.2 Informative Data Extraction

Once the unit disk region is found, we can proceed to the extraction of the informative data. First the relevant moment vector $\mathbf{Z}' = (A'_{p_{I+1}q_{I+1}}, ..., A'_{p_Lq_L})$ is computed. Then with the same key K_2 as in the embedder, the same two dither vectors $(d_1(0), ..., d_L(0))$ and $(d_1(1), ..., d_L(1))$ are re-generated. Using the same quantizer as in (4.34), we quantize the magnitude of each $A'_{p_iq_i}$ with the two corresponding dithers respectively,

$$|A'_{p_iq_i}|_j = \mathcal{R}\left(\frac{|A'_{p_iq_i}| - d_i(j)}{\Delta}\right)\Delta + d_i(j), \tag{4.47}$$

where $i = I+1, ..., L$, $j = 0, 1$, $|A'_{p_iq_i}|_j$ denotes the quantized value of $|A'_{p_iq_i}|$ with dither $d_i(j)$, and $\mathcal{R}(\cdot)$ is the rounding operation.

By comparing the distances between $|A'_{p_iq_i}|$ and its two quantized versions, we obtain the estimate of the bit embedded on $|A_{p_iq_i}|$

$$\hat{b}_i = \arg \min_{j\in\{0,1\}} (|A'_{p_iq_i}|_j - |A'_{p_iq_i}|)^2, \quad i = I+1, ..., L. \tag{4.48}$$

which is so-called minimum distance decoder.

4.7 Simulation Results

In this section we examine the robustness of the proposed watermarking algorithm to various forms of attacks. Unless otherwise stated, the test images are 256×256 with 256 graylevels, and the unit disk is chosen such that it is fully inside an image and touches the four borders. As an example, Fig. 4.4(b) is the watermarked version of Fig. 4.4(a), in which an array of 128 bits is embedded, while Fig.4.4(c) is the absolute difference between Fig.4.4(a) and Fig. 4.4(b), multiplied by 25 for better display.

(a) (b) (c)

Figure 4.4: An example of using the proposed algorithm. (a) Original Lena of size 256 × 256. (b) Lena watermarked with 128 bits. (c) Exaggerated difference of (b) and (a).

4.7.1 The Quality of Watermarked Images

The PSNR of a watermarked image is determined by two main factors. On the one hand, given a fixed number of bits to be embedded, the PSNR is determined by the quantization step size Δ

of the dither modulation imposed on the magnitudes of ZM/PZM. A larger Δ leads to a stronger watermark, but results in a lower PSNR, and vice versa. On the other hand, given a fixed Δ or watermark strength, the number of bits to be embedded decides the PSNR of the watermarked image. The more bits are embedded, the lower value is PSNR, and vice versa. The relationship between PSNR and these two factors is clearly reflected in Fig. 4.5(a) and Fig. 4.5(b), which are the experimental results from ZM-based and PZM-based algorithms respectively. Every data point in these two figures is the average of 100 individual tests. It can be verified that Fig. 4.5(a) and Fig. 4.5(b) exactly agree with Theorem 4.2.

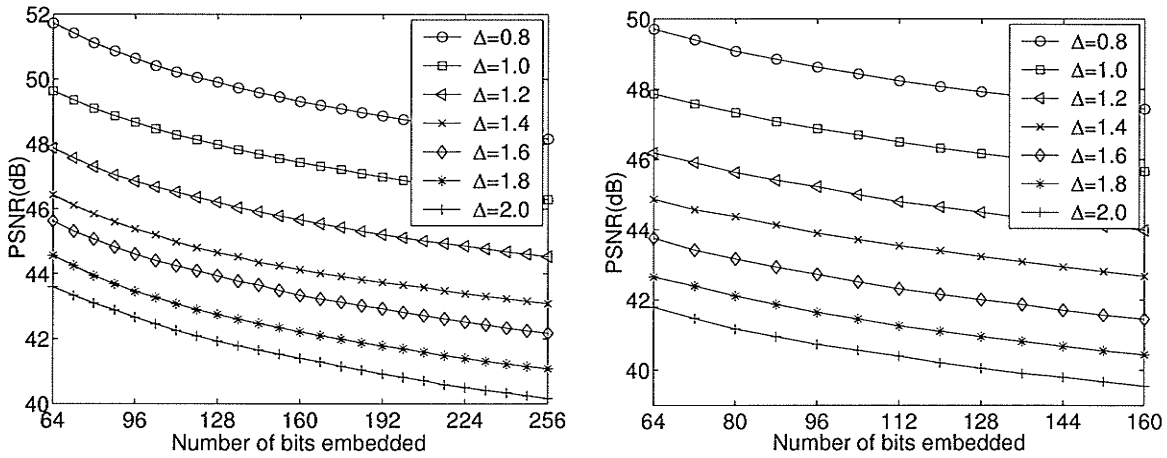


Figure 4.5: The quality of watermarked images is affected by the number of bits embedded and the quantization step size. (a) ZM-based watermarking results. (b) PZM-based watermarking results.

In all our experiments, we chose quantization step size Δ such that the resulting PSNR $>$ 40dB, which guarantees a good watermark transparency.

4.7.2 Robustness to Image Rotation

We are particularly interested in the watermark performance, i.e., the bit error rate (BER), under the attack of image rotation. We used Fig. 4.6(a) as the test image, and 160-bit long random sequences as the information to embed. The quantization step sizes Δ were set such that an average PSNR $\approx$ 42.5dB and 46.4dB for ZM-based and PZM-based algorithms respectively. Fifteen different

rotation angles $\theta = \{3°, 6°, ..., 45°\}$ were tried. The rotated version of each watermarked image was computed via bilinear interpolation and the resulting additional black borders were partially cropped so that the image sizes remained the same. Fig. 4.6(b) is an example of the rotated images. To obtain the BER at a certain rotation angle θ, 100 different randomly generated bit sequences were tried and the the BER was taken as the average of the 100 cases. Fig. 4.6(c) shows the simulation results, both for ZM-based algorithm and the PZM-based algorithm, illustrated by the dotted line and the solid line respectively, from which we see an excellent watermark robustness to image rotation with the maximum BER at $O(10^{-3})$. It can also be seen from the figure that BER is related irregularly with θ.

4.7.3 Robustness to Image Scaling

Image scaling is another common form of geometric attacks. We looked at BERs under 16 different scaling levels. A 256×256 watermarked image was scaled to smaller sizes, ranging from 128×128 to 248×248 with an interval 8 of side length. Fig. 4.7(a) is an example of the scaled images. Prior to watermark extraction, the scaled images were scaled back to the size 256×256. It is worth noting that although they were scaled back to their original sizes, the images were quite different from their unscaled versions due to the double scaling transforms they underwent. The scaling operation was performed via bilinear interpolation. Fig. 4.7(b) shows the results for embedding 80 bits and 160 bits by ZM-based watermarking, while Fig. 4.7(c) gives the results for embedding 80 bits and 160 bits by PZM-based watermarking. Each data point in the figures is the average of 100 test results on different randomly generated bit sequences. The figure shows a trend that BER decreases as the scaling lessens and information amount drops.

4.7.4 Robustness to Image Flipping

Image flipping, either horizontal or vertical, is a very easy attack to perform, however it is so effective that it fails many existing watermarking algorithms. Shown in Fig. 4.8(a) is a watermarked image.

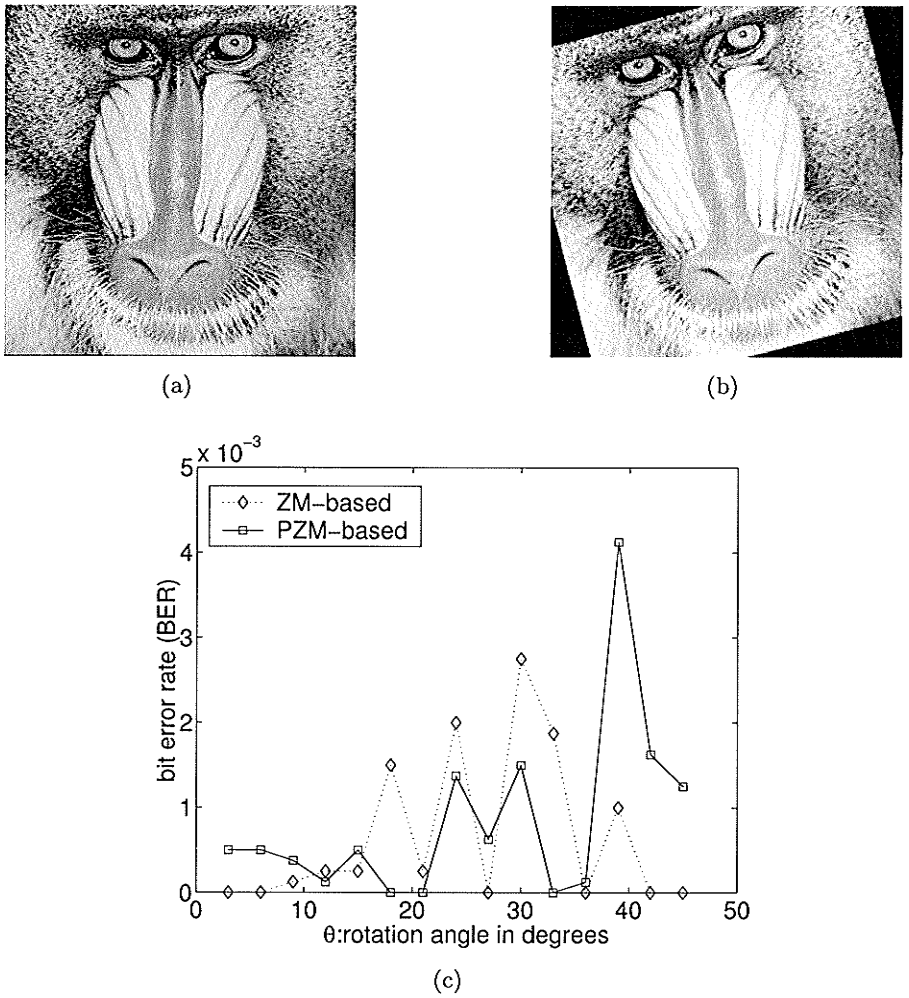


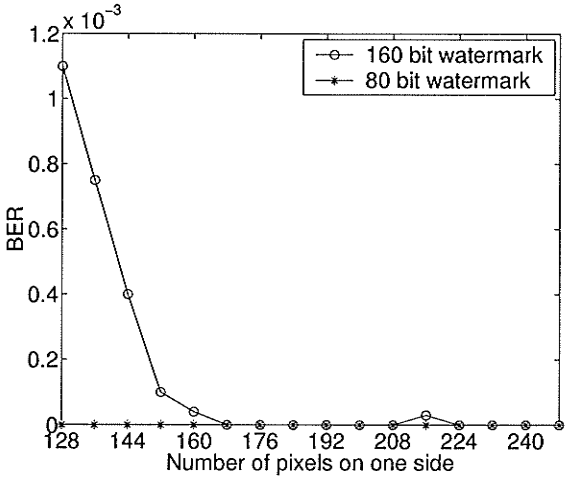
Figure 4.6: Watermark robustness to rotation. (a) Original Baboon image of size 256×256. (b) Baboon image watermarked with 160 bits followed by a 15° rotation. (c) BER as a function of rotation angles.

Fig. 4.8(b) is the horizontally flipped version of Fig. 4.8(a), while Fig. 4.8(c) is the vertically flipped version of Fig. 4.8(a). As mentioned above, the proposed approach is inherently immune, and hence perfectly robust against such an attack. All the experiments with this kind of attack, both for ZM-based scheme and PZM-based scheme, yielded a BER=0.

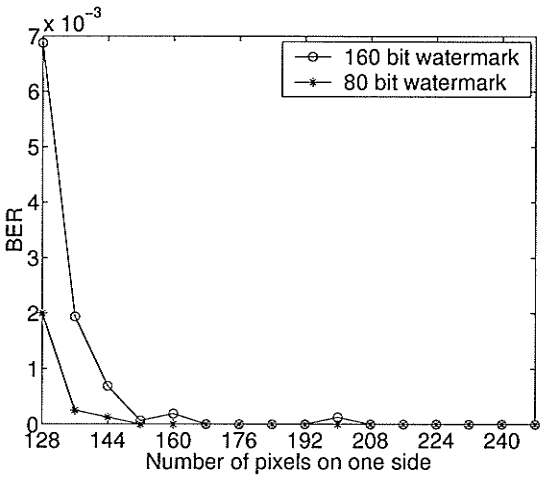
4.7.5 Robustness to Image Compression

With the wide-spread use of standard JPEG image compression, lossy compression is a highly common form of image processing. We looked at BERs under different JPEG compression levels

(a)



(b)



(c)

Figure 4.7: Watermark robustness to image scaling. (a) A scaling example: watermarked Lena scaled by 75% of side length. (b) BER as a function of scaled image size in the case of ZM-based watermarking. (c) BER as a function of scaled image size in the case of PZM-based watermarking.

with quality factors from 20 to 90 with an interval of 2. Fig. 4.9(b) shows the results for embedding 64, 128 and 256 bits respectively in the image of Fig. 4.4(a) by means of ZM-based algorithm while Fig. 4.9(c) shows the results for embedding 64, 128 and 160 bits respectively in the same source image via PZM-based scheme. Each data point in the figures is the average of 100 individual results, which were obtained from 50 different randomly generated bit sequences. It can be seen that BERs decrease rapidly as the quality factor increases and the number of bits embedded drops. Considering that a JPEG quality factor less than 50 gives an obviously degraded image and hence is unlikely to be used by an attacker in practice, the robustness to JPEG lossy compression is

(a) (b) (c)

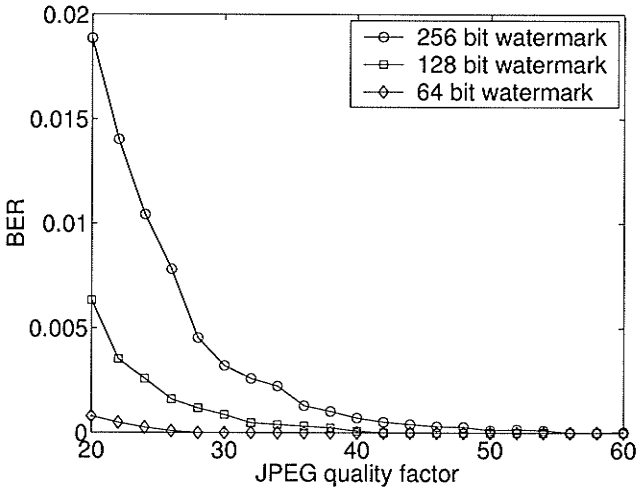
Figure 4.8: Watermark robustness to image flipping. (a) Watermarked image of F16. (b) Horizontally flipped version of (a). (c) Vertically flipped version of (a).

remarkable in our algorithms.

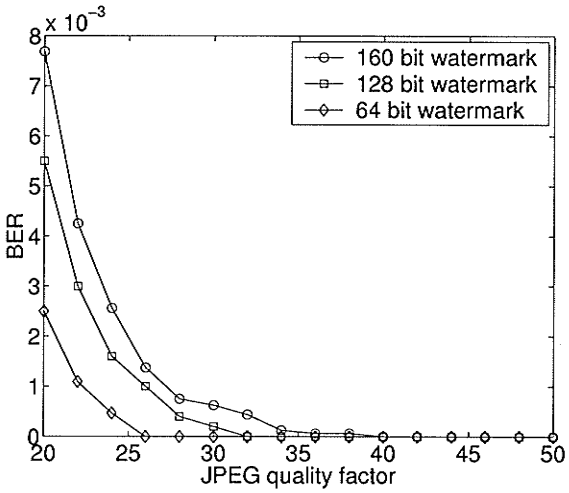
4.7.6 Robustness to Lowpass Filtering

Lowpass filters are a family of filters that are commonly applied in image processing [28], including averaging filters and Gaussian filters etc., and therefore, lowpass filters are of interest to watermark designers. We introduce here the test results on the watermarked images undergoing Gaussian filters. The test image is the Lena image, the payload is 128 bits, and the average PSNR≈41.7dB. Shown in Fig. 4.10(a) is an example of Gaussian filtered watermarked image with 5×5 window and $\sigma_{gf} = 0.9$, which is apparently blurred. We recorded BERs under different levels of filter strength, σ_{gf}, ranging from 0.5 to 2 with an interval of 0.1. Fig.4.10(b) shows the results for the ZM-based watermarking scheme, while Fig.4.10(c) illustrates the results for the PZM-based algorithm. In both cases, 3×3, 5×5 and 7×7 window sizes of the Gaussian filters were tried. For every data point in the figures, 50 different bit sequences were generated and tested on the watermarked images, and then the average of 50 individual results was taken. The results display excellent watermark robustness to lowpass filtering. No error was observed when $\sigma_{gf} < 0.9$.

(a)



(b)



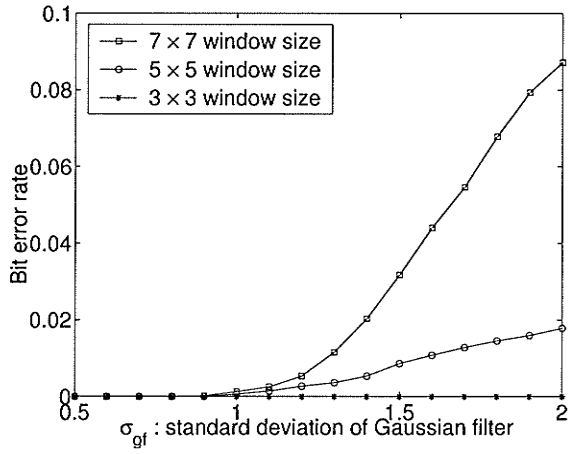
(c)

Figure 4.9: Watermark robustness to JPEG lossy compression. (a) A compression example: watermarked Lena image compressed by JPEG with quality factor 30. (b) BER as a function of JPEG quality factor in the case of ZM-based watermarking. (c) BER as a function of JPEG quality factor in the case of PZM-based watermarking.

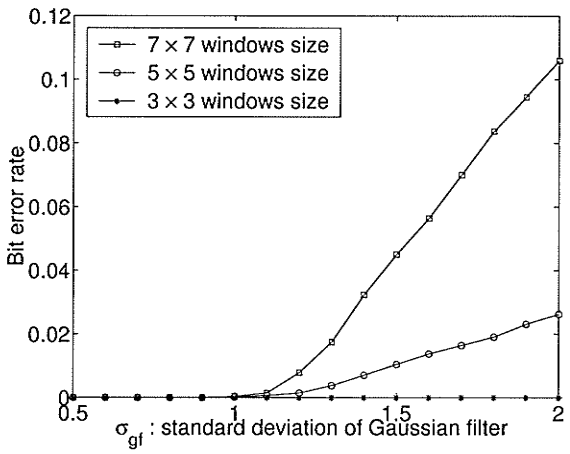
4.7.7 Robustness to Additive Noise

Additive Gaussian noise is considered here to be an attack because it approximately models some interferences the watermarked images may undergo. We recorded BERs under different levels of noise whose standard deviations range from 0 to 15 with an interval of 0.5. Fig. 4.11(b) and Fig. 4.11(c) show the results for ZM-based and PZM-based algorithms respectively, embedding 64, 96 and 128 bits in the image of Fig. 4.4(a), with PSNR $\approx$ 42dB in each case. For every data point in the figures, 100 different matrices of pseudo-random Gaussian noise were tested on the

(a)



(b)



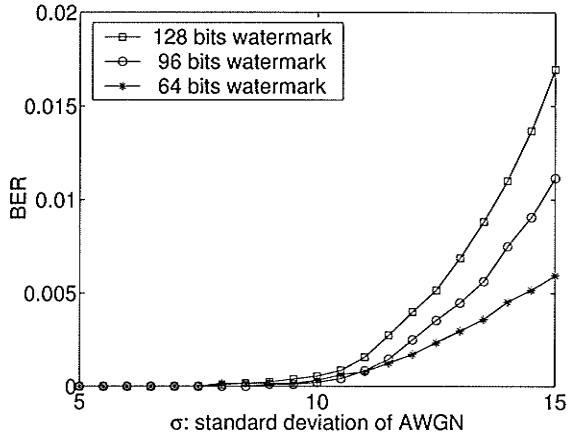
(c)

Figure 4.10: Watermark robustness to Gaussian filtering. (a) An example: Watermarked Lena after a Gaussian filter with a 5×5 window size and $\sigma_{gf} = 0.9$. (b) BER as a function of the standard deviation of the filter in the case of ZM-based watermarking. (c) BER as a function of the standard deviation of the filter in the case of PZM-based watermarking.

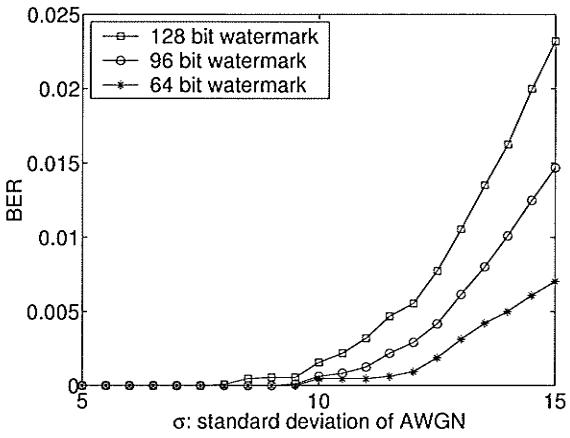
watermarked image, and then the average of 100 individual results was taken. The figures show that the algorithm has an outstanding performance on the attack of additive Gaussian noise. In all of our tests, we found no error when $\sigma < 5$. Even for $\sigma = 10$, BER $\leq O(10^{-4})$ if the payload is below 128 bits.

It is worth noting that when $\sigma = 5$, the attacked image displays obvious quality degradation, as shown in Fig. 4.11(a), which means that an attacker has to control the noise strength such that $\sigma < 5$ in order to maintain the practical value of the watermarked image.

(a)



(b)



(c)

Figure 4.11: Watermark robustness to additive Gaussian noise. (a) An example: Watermarked Lena with Gaussian noise, $\sigma = 5$. (b) BER as a function of the standard deviation of AWGN in the case of ZM-based watermarking. (c) BER as a function of the standard deviation of AWGN in the case of PZM-based watermarking.

4.7.8 Stirmark Test Results

Now we evaluate the watermark robustness to some of the attacks provided by the Stirmark3.1 benchmarking tool [42, 61]. We used the images shown in Fig. 3.15 as original images, each of which has a 512×512 image size. A unit disk with a 256-pixel diameter was made to cover the central circular region. The auxiliary sequence used to detect the unit disk contains 16 bits, while the informative data to embed is 64 randomly generated bits. Each of the test images, after the 80 bits are embedded, was fed into the Stirmark tool. Then Stirmark performed various attacks on the watermarked image, and produced a series of attacked images. We used the JPEG-compressed

version of these attacked images for data extraction and recorded the number of erroneous bits for each image. The average bit error rates over all the test images are listed in Table 4.3, where -1 denotes failure to extract the embedded data. It can be seen that the proposed algorithm has excellent robustness to image rotation, scaling, cropping and JPEG compression with quality factors over 20; it has good robustness to image aspect ratio change, mild removal of lines and JPEG with very low quality factors. However, the embedded data cannot be extracted in several cases including excessive scaling and cropping, shearing and random band etc.

Table 4.3: Stirmark test results

Attack type	Average BER
Remove 17 rows and 5 columns	0.0430
Remove 5 rows and 1 columns	0.0605
Cropping 10%	0
Cropping 20%	0
Cropping 50%	0
Cropping 75%	-1
Gaussian filtering 3x3	0
JPEG 15	0.0137
JPEG 20	0.0039
JPEG 25-90	0
Change aspect ratio x:0.80 y:1.00	0.0020
Change aspect ratio x:0.90 y:1.00	0
Change aspect ratio x:1.00 y:1.10	0.0137
Rotation -0.25	0.0215
Rotation -2.00	0
Rotation 10.00	0
Rotation 90.00	0
Scale 0.25	-1
Scale 0.50	0.0020
Scale 0.75	0
Scale 0.90	0
Scale 1.50	0
Scale 2.00	0
Sharpening 3x3	0.1328
Shearing x:5.00 y:0.00	-1
Stirmark random bend	-1

4.8 Discussion

4.8.1 Comparison of ZM Watermarks to PZM Watermarks

Based on the simulation results on ZM-based and PZM-based algorithms, we can see that their performances are quite close in most cases. It is necessary to point out that in all the experiments carried out, we set the quantization steps such that for the same experiment, the two algorithms would output watermarked images of the same quality. For example, in the case of 128-bit watermark, we set Δ=2 for the ZM-based algorithm while Δ=1.6 for the PZM-based algorithm, such that both algorithms produced watermarked images with PSNR $\approx$ 42dB. If we use the same Δ value for the two algorithms, the PZM-based watermarks notably outperform the ZM-based watermarks in terms of robustness, but at the cost of lower quality of watermarked images. The reason for this is that for the same number of bits to be embedded, fewer lower order moments get involved in ZM-based watermarks than in PZM-based watermarks, because there are approximately twice as many PZMs as ZMs due to the constraint $p - |q| =$ even on ZMs. Roughly speaking, low order moments, which are low-frequency image components in nature, have stronger impacts on the image quality than high order moments, but have better robustness to signal distortions such as lowpass filtering and lossy compression.

4.8.2 The Detection of the Unit Disk Region

It is crucial that the unit disk region has to be located correctly for the extraction of the embedded data. As stated before, the auxiliary bit sequence is deployed to facilitate the search for the region. In our experiments, we use finite steps of trial for the detection of the unit disk. In theory it is possible to design an algorithm to locate the unit disk in an elegant way, because the modified ZMs/PZMs of the disk region have distinguishing properties due to quantization. This interesting issue is beyond the scope of this thesis, and deserves independent research.

4.8.3 Extension to M-ary Dither Modulation

In case of large payloads, e.g., 512 bits, we can use one independent ZM/PZM magnitude to carry more than one bit of information. The dither modulation does not have to be binary, as what we have shown in previous sections, but rather, it can be M-ary in general, i.e., a set of M quantizers can be used, for which (4.32) still applies, where $m \in \{0, 1, ..., M-1\}$. Obviously with M-ary dither modulation, an independent magnitude of ZM/PZM carries $\log_2 M$ bits of information. This is an effective approach to the increase of data capacity of watermarks. Nevertheless, there is a price to pay for this gain of payload. At the same level of watermark-induced distortion, M-ary dither modulation gives higher bit error rates than binary dither modulation. The rigorous theoretical derivation of information capacity of ZM/PZM-based watermarks is beyond the scope of this thesis.

On the other hand, in the case of small payloads, more than one ZM/PZM magnitude can be combined to carry one bit of information in order to gain extra watermark robustness. If l independent moment magnitudes are employed to carry one bit, and binary dither modulation is performed on each magnitude, then the 1-D minimum distance decoder represented by (4.48) would be replaced by an l-D minimum distance decoder. As a result, the watermark robustness (measured by BER) is improved.

4.8.4 Implementation Issue of the Proposed Algorithm

Although the proposed ZM/PZM watermarking algorithm is quite straightforward, the amount of computation is considerable. The bottleneck of speed lies in the computation of Zernike/pseudo-Zernike polynomials at the grid points of an image inside the unit disk. In particular, the computation of the radial polynomial $R_{pq}(\rho)$ and the exponential factor $e^{jq\theta}$ is very time-consuming. For the simulations we reported in this thesis, all the radial polynomials and the exponential factors to be used were computed and stored as tables in computer memory in advance. In this fashion, most of the computation was reduced to table lookup, and the speed was greatly improved. For example, on a 1.8G Pentium 4-based computer, the time for embedding a 128 bit sequence into a

256×256 image requires about 10.9 seconds, while the watermark extraction process takes about 5.3 seconds. In applications where computer memory is limited, fast algorithms [53, 54] can be adopted to improve this situation.

4.9 Chapter Summary

In this chapter, we have investigated the invariance properties of ZM/PZM, analyzed and verified the non-ideal rotational invariance of ZMs/PZMS of digital images, and pointed out how to circumvent the non-ideal invariance in applications. On basis of this, we proposed a multibit watermarking scheme based on modification of ZMS/PZMs. By quantizing the magnitudes of a group of selected ZMs/PZMs through quantization index modulation, hundreds of bits can be embedded into an image imperceptibly. An analytical result revealing the relationship of the quantization step size, the quality of the watermarked image, and the number of embedded bits has been derived. For extraction of the embedded message from a watermarked image with distortions, we use a minimum distance decoder, following the quantization of same group of selected ZMs/PZMs computed from the possibly manipulated image. An auxiliary bit sequence is used to address the location of the unit disk region in a distorted image. On successful detection of the unit disk region, the embedded informative bit sequence can be extracted at low or even zero error rates from a distorted watermarked image. Experimental results show that the embedded data are robust against typical geometric distortions, such as image rotation, scaling, flipping, cropping and aspect ratio change, as well as other common attacks such as lossy compression, additive noise and lowpass filtering.

Chapter 5

Geometrically Robust Image Watermarking in Polar Coordinates

5.1 Introduction

In Chapter 4, we introduced an effective approach to data hiding through modification of Zernike or pseudo-Zernike moments of images, which leads to watermark robustness against common geometric distortions including image rotation, scaling and flipping etc., as well as other manipulations such as lossy compression, noise addition and lowpass filtering. However, due to the inaccuracy involved in moment computation, the invariance property of some moments does not hold. This reduces considerably the number of ZMs/PZMs that can be used for data hiding, resulting in a disadvantage of the ZM/PZM-based watermarking algorithm when used in the Cartesian coordinate system.

The accuracy issue of ZMs of digital images was brought to attention in [43, 45]. It has been demonstrated that two kinds of errors, namely geometric error and numerical integration error, are inherent and inevitable [43, 45, 58] in the computation of Zernike moments. Without dealing with the accuracy of ZMs/PZMs appropriately, some attractive properties of Zernike moments would be compromised. For instance, the property of magnitude invariance to image rotation depends on the accuracy of Zernike moments. With the existing Cartesian methods to compute ZMs/PZMs,

some of the moment magnitudes are not truly rotationally invariant. Furthermore, the existing errors have such a negative impact on image analysis and reconstruction that when the order of moments reaches a critical value, the resulting reconstruction error increases.

In this chapter, we further improve the ZM/PZM-based data hiding approach introduced in Chapter 4. To simplify the structure of this chapter, we only consider the case of ZM-based methods, however, all the techniques and conclusions can be applied to PZM-based methods as well. In the first part of this chapter, we deal with the accuracy issue in the computation of ZMs. We show that if ZMs are calculated in an appropriate way in the polar coordinate system rather than in the conventional Cartesian coordinate system, the accuracy loss from geometric error and numerical integration error can be avoided. We present a detailed description of the proposed approach including the derivation of formulas for ZM computation in the polar coordinate system, the polar pixel arrangement scheme and the image conversion via interpolation. With the proposed approach, although the accuracy of ZMs is still not perfect due to the Cartesian-polar image conversion, it is shown to be improved greatly. As a result, the magnitude invariance of ZMs gets close to ideal [85], and also, one can perform image analysis and reconstruction via ZMs of very high orders [88].

In the second part of this chapter, we apply the proposed polar ZMs in the design of a data hiding system [86]. Due to the significant improvement of moment accuracy, all ZMs up to a high order are now eligible for data hiding. The effectiveness of the polar ZM-based watermarking system is to be compared with that of the Cartesian ZM-based watermarking system, which verifies the advantage of the polar Zernike moments for geometrically robust data hiding.

5.2 Accuracy Problem in Traditional ZM Computation

As stated in Chapter 4, for an analog image function $f(x, y)$, its Zernike moment of order p with repetition q is defined by

$$A_{pq} = \frac{p+1}{\pi} \int\int_{\mathbb{D}} f(x, y) V_{pq}^*(x, y) dx dy, \tag{5.1}$$

where $\mathbb{D} = \{(x, y) : x^2 + y^2 \leq 1\}$ is the unit disk .

However, for digital images, (5.1) cannot be applied directly. Given a digital image $f(x_i, y_j), i = 1, ..., N, j = 1, ..., N$, $x_i = (2i - N - 1)/N$ and $y_j = (2j - N - 1)/N$, it can be viewed as a pixel-wise constant function within the unit circle, and thus its ZMs can be approximated by

$$\check{A}_{pq} = \frac{p+1}{\pi} \sum_i \sum_j \nu_{pq}(x_i, y_j) f(x_i, y_j), \tag{5.2}$$

where the values of i and j are taken such that $(x_i, y_j) \in \mathbb{D}$, and

$$\nu_{pq}(x_i, y_j) = \int_{x_i - \frac{\lambda}{2}}^{x_i + \frac{\lambda}{2}} \int_{y_j - \frac{\lambda}{2}}^{y_j + \frac{\lambda}{2}} V^*_{pq}(x, y) dx dy, \tag{5.3}$$

where $\lambda = \frac{2}{N}$ is the pixel width/height. For the computation of the factor $\nu_{pq}(x_i, y_j)$, some methods of numerical integration can be applied. The most commonly used formula, which is also the simplest one, is the following:

$$\nu_{pq}(x_i, y_j) \approx \lambda^2 V^*_{pq}(x_i, y_j). \tag{5.4}$$

Some more complex numerical techniques, based on the multidimensional cubature algorithms, were introduced in [45] in order to improve the accuracy of computing $\nu_{pq}(x_i, y_j)$.

Nevertheless, as pointed out in [43, 45, 58], the accuracy of ZMs computed via (5.2) and (5.4) suffers from two sources of errors, namely geometric error and numerical integration error. The former is due to the fact that the total area covered by all the square pixels involved in the computation of Zernike moments via (5.2) is not exactly the unit disk, as illustrated by the ragged border in Fig. 5.1. The latter results from the numerical integration via an approximation formula like (5.4). Although some techniques can be deployed [45] to alleviate the inherent accuracy problem, the aforementioned errors can never be eradicated provided that the computation of Zernike moments is performed with the Cartesian coordinate system.

5.3 Computing Zernike Moments in the Polar Coordinate System

The cause of errors in computing ZMs via (5.2) lies in the adoption of Cartesian coordinates for the computation, which is motivated by the fact that digital images are represented by square pixels.

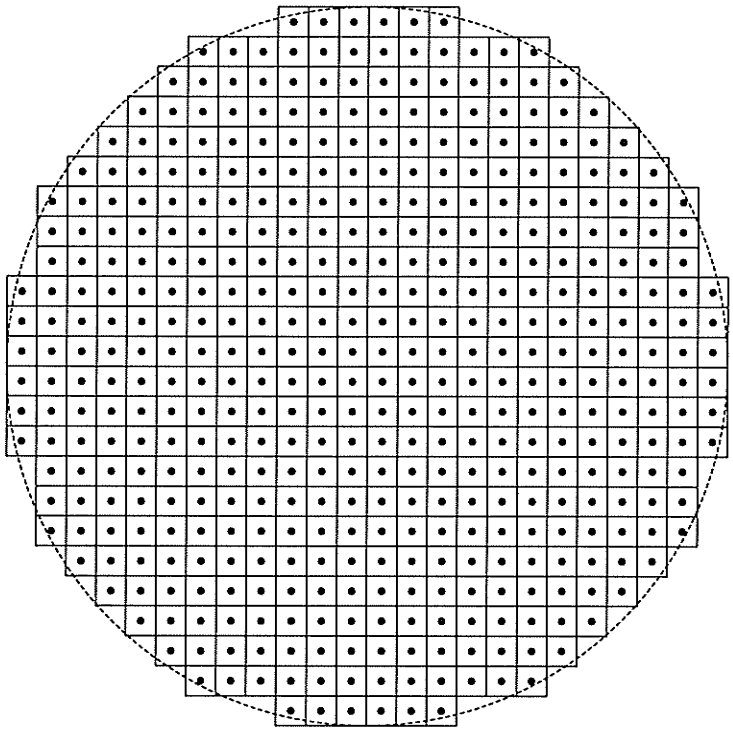


Figure 5.1: An illustration of the Cartesian pixel grid for computation of Zernike moments.

However, this practice of ZM computation does not take into account the circular nature of Zernike polynomials. In this section we present an algorithm for computation of ZMs in polar coordinates, in which neither geometric error nor numerical integration error is present.

5.3.1 Principles

To remove geometric error and numerical integration error in ZM computation, we need to take a different approach from the existing Cartesian methods. It is intuitive that geometric error can be avoided by using non-square pixels, whose areas sum up to that of the unit disk. Furthermore, we can use an analytical method instead of numerical approximation for the pixel-wise integration of basis functions, which is accurate and efficient in moment computation [26]. The definition of Zernike polynomials in (4.17) reveals that they are expressed directly by polar coordinates ρ and θ. This prompts us that adoption of polar coordinates could facilitate the computation of ZMs. For this purpose, we rewrite the definition of Zernike moment (5.1) in its equivalent form based on

polar coordinates

$$A_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho\cos\theta, \rho\sin\theta) R_{pq}(\rho) e^{-jq\theta} \rho d\rho d\theta. \tag{5.5}$$

If an image is approximated by a piecewise constant function composed of constant-intensity sectors, denoted by Ω_{uv}, which are concentric about the origin and non-overlapping, i.e.,

$$\bigcup_{(u,v)} \Omega_{uv} = \mathbb{D}, \tag{5.6}$$

and

$$\Omega_{uv} \cap \Omega(u', v') = \emptyset, \ \forall \ (u, v) \neq (u', v'), \tag{5.7}$$

we can get an approximate version of (5.5) as follows.

$$\hat{A}_{pq} = \frac{p+1}{\pi} \sum_u \sum_v \hat{f}(\rho_{uv}, \theta_{uv}) \omega_{pq}(\rho_{uv}, \theta_{uv}), \tag{5.8}$$

where $\hat{f}(\rho_{uv}, \theta_{uv})$ is the estimated image intensity of Ω_{uv}, centered at (ρ_{uv}, θ_{uv}), and the double summation is performed over all the sectors inside the unit disk. The factor $\omega_{pq}(\rho_{uv}, \theta_{uv})$ is an integral over Ω_{uv}:

$$\begin{aligned} \omega_{pq}(\rho_{uv}, \theta_{uv}) &= \int \int_{\Omega_{uv}} R_{pq}(\rho) e^{-jq\theta} \rho d\rho d\theta \\ &= \int_{\rho_{uv}^{(s)}}^{\rho_{uv}^{(e)}} R_{pq}(\rho) \rho d\rho \int_{\theta_{uv}^{(s)}}^{\theta_{uv}^{(e)}} e^{-jq\theta} d\theta, \end{aligned} \tag{5.9}$$

where $\rho_{uv}^{(s)}$ and $\rho_{uv}^{(e)}$ denote the starting and ending radii of Ω_{uv} respectively, while $\theta_{uv}^{(s)}$ and $\theta_{uv}^{(e)}$ denote the starting and ending angles of Ω_{uv} respectively. Fig. 5.2 illustrates the variables introduced above.

The formula in (5.9) is the product of two integrals, whose exact values can be obtained analytically as follows.

$$\int_{\rho_{uv}^{(s)}}^{\rho_{uv}^{(e)}} R_{pq}(\rho) \rho d\rho = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)! [(\rho_{uv}^{(e)})^{p-2s+2} - (\rho_{uv}^{(s)})^{p-2s+2}]}{(p-2s+2) s! (\frac{p+|q|}{2} - s)! (\frac{p-|q|}{2} - s)!}. \tag{5.10}$$

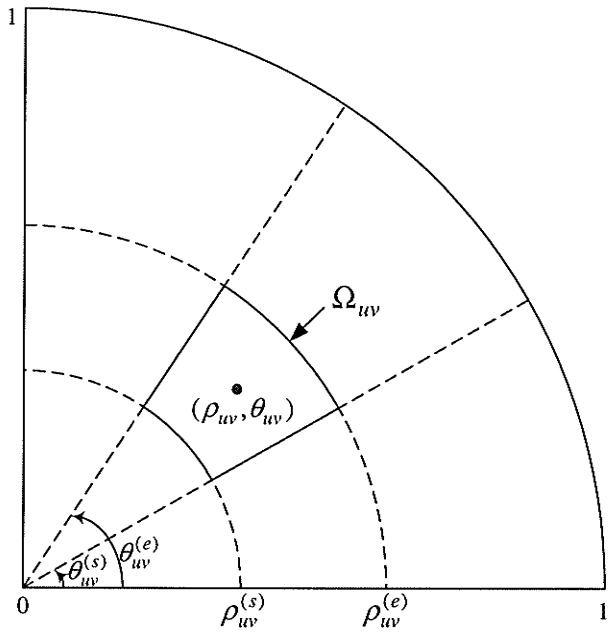


Figure 5.2: An illustrative sector, Ω_{uv}, indicates a polar pixel. The location of Ω_{uv}, (ρ_{uv}, θ_{uv}), is defined by $\rho_{uv} = (\rho_{uv}^{(s)} + \rho_{uv}^{(e)})/2$ and $\theta_{uv} = (\theta_{uv}^{(s)} + \theta_{uv}^{(e)})/2$.

$$\int_{\theta_{uv}^{(s)}}^{\theta_{uv}^{(e)}} e^{-jq\theta} d\theta = \begin{cases} \frac{j}{q}\left[e^{-jq\theta_{uv}^{(e)}} - e^{-jq\theta_{uv}^{(s)}}\right], & q \neq 0 \\ \theta_{uv}^{(e)} - \theta_{uv}^{(s)}, & q = 0 \end{cases} . \tag{5.11}$$

Combining (5.8), (5.9), (5.10) and (5.11), we obtain the exact formula for ZMs of $\hat{f}(\cdot, \cdot)$, without introducing any geometric error or numerical integration error in the process.

5.3.2 A Polar Pixel Structure for ZM Computation

As we have already noted, in order to eliminate geometric and numerical integration errors, we must represent $f(\rho_{uv}, \theta_{uv})$ over the sectors satisfying conditions (5.6) and (5.7). If we imagine each sector as a fan-shape pixel whose value is determined by that of its central point, then the question what should be the arrangement of these fan-shape pixels arises . There are numerous schemes satisfying conditions (5.6) and (5.7). An example of the most obvious structure is shown in Fig. 5.3, in which the unit disk is divided uniformly along both the radial and angular directions. This scheme has the advantage of easy implementation. However, it behaves poorly in terms of image

representation. This is due to the fact that the areas of the sectors vary greatly, and it is impossible to achieve both efficiency and accuracy of information representation. In fact, if the inner sectors are required to be large enough to represent image information efficiently, the outer sectors are too large to accurately represent the original image information. On the other hand, if the outer sectors are required to be small enough to accurately represent the image information, then there are too many inner sectors for the scheme to be efficient.

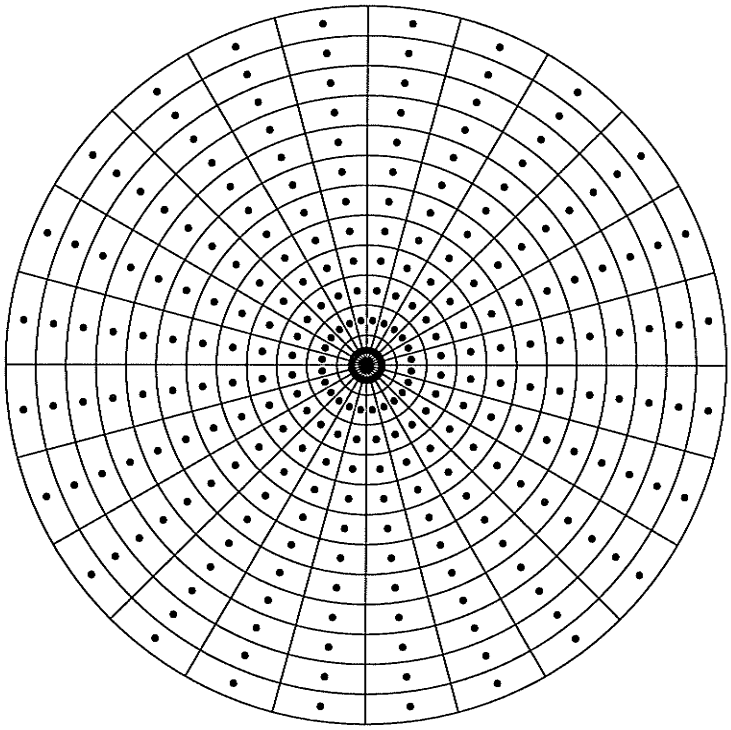


Figure 5.3: An example of a tentative polar pixel grid for efficient computation of Zernike moments.

To overcome the aforementioned problems of the partition scheme shown in Fig. 5.3, we need to design a more appropriate one. The following criteria are used for finding a suitable polar pixel partition of the image plane.

- All the sector areas should be approximately equal. Note that in Cartesian coordinates all pixels are of equal size.

- The number of polar pixels inside the unit circle should not be smaller than that of the Carte-

sian pixels inside the unit circle, so that the necessary image resolution could be maintained without loss of information.

- The polar pixels should be as 'square' as possible, i.e., the lengths of the borders of a sector should be close enough, so that the image distortion due to the coordinate system conversion could be kept at a low level.

- In order to facilitate the storage and computation processes, the polar pixel structure should be as simple and regular as possible.

Following these guidelines, we propose a pixel arrangement scheme illustrated in Fig. 5.4. The details of this structure are listed below.

- The unit disk is uniformly divided along the radial direction into U sections, with the separating circles located at $\{\frac{k}{U}, k = 1, ..., U\}$.

- The kth ring-shape section is equally divided into $V(2k-1)$ sectors by radii starting from the origin, with angles $\{(i-1)\frac{2\pi}{V(2k-1)}, i = 1, ..., V(2k-1)\}$. V is the number of sectors contained in the innermost section.

It can be shown by simple algebra that the unit disk is divided into VU^2 sectors, each of which has an area of $\frac{\pi}{VU^2}$. The values of U and V should be set properly. A small value of VU^2 is advantageous in terms of computation and implementation, but may represent inadequately the image information. On the other hand, a large value of VU^2 is beneficial for representation of the image, but entails heavy workload. In practice, we recommend setting $V = 4$ and $\frac{N}{2} \leq U \leq N$ for an $N \times N$ image.

Equipped with the above introduced scheme of the polar pixel arrangement, the formula for ZM computation in (5.8) can be further rewritten as

$$\hat{A}_{pq} = \frac{p+1}{\pi} \sum_{u=1}^{U} \sum_{v=1}^{V(2u-1)} \hat{f}(\rho_{uv}, \theta_{uv}) \int\int_{\Omega_{uv}} R_{pq}(\rho) e^{-jq\theta} \rho d\rho d\theta, \tag{5.12}$$

where the integral can be explicitly evaluated using (5.10) and (5.11).

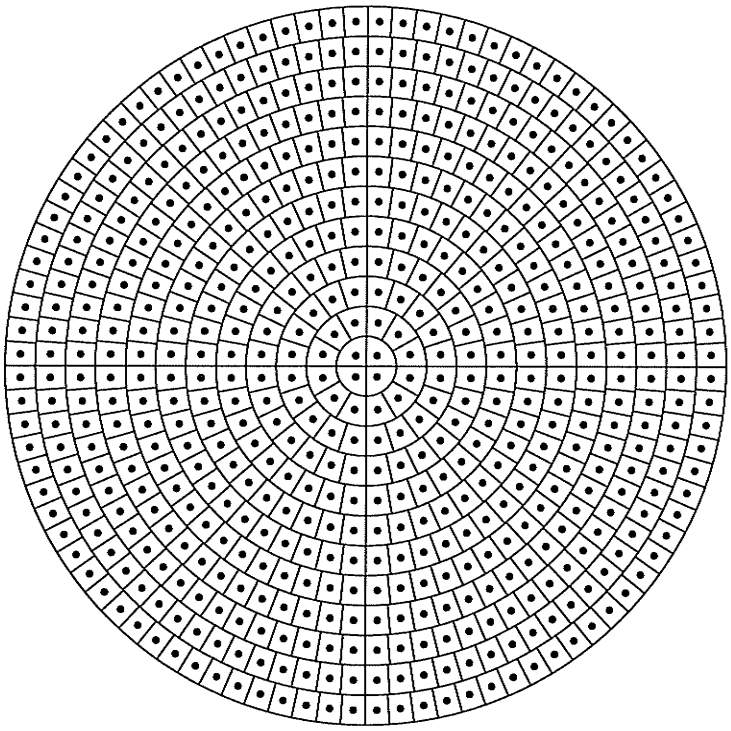


Figure 5.4: The proposed structure of polar pixels for efficient computation of Zernike moments.

5.3.3 Image Representation in Polar Coordinates

We have seen in the previous section that computing ZMs in polar coordinates results in neither geometric error nor numerical integration error, provided that the image is represented by a certain polar pixel structure as in Fig. 5.4. However, a digital image is usually defined by a set of square Cartesian pixels, as shown in Fig. 5.1. It can be verified that the locations of most of the polar pixels do not coincide with those of the Cartesian pixels. Therefore, we have to derive the polar counterpart of a given Cartesian image before computing its ZMs in polar coordinates. This issue can be resolved by applying an image interpolation procedure.

There are a number of existing image interpolation techniques [36, 48] which we can use to determine the values of polar pixels. The simplest and least accurate one is the nearest point interpolation, by which the value of a polar pixel is set to that of the closest Cartesian pixel. Another technique is the bilinear interpolation, which determines the value of a point by the

linearly weighted average of the four neighboring pixel values. The bilinear technique yields better image quality, and has been widely used in many applications. A more advanced approach, which we adopt in our work, is bicubic interpolation [12, 36, 48]. It uses the 16 neighboring pixels to compute the value of an interpolating point. This usually yields a smooth image with very small interpolation error. The 1-D kernel function representing the bicubic interpolation is a cubic spline:

$$h(x) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}|x|^2 + 1, & |x| \leq 1 \\ -\frac{1}{2}|x|^3 + \frac{5}{2}|x|^2 - 4|x| + 2, & 1 < |x| \leq 2 \\ 0, & \text{otherwise.} \end{cases} \tag{5.13}$$

The image value of Ω_{uv} can be estimated via the 2-D cubic convolution between the image function $f(x_i, y_j)$ and the kernel $h(x)h(y)$, i.e.[1],

$$\hat{f}(\rho_{uv}, \theta_{uv}) = \sum_{i=m-1}^{m+2} \sum_{j=n-1}^{n+2} f(x_i, y_j) h(\frac{\rho_{uv}\cos\theta_{uv} - x_i}{\lambda}) h(\frac{\rho_{uv}\sin\theta_{uv} - y_j}{\lambda}), \tag{5.14}$$

where $m = \lfloor \frac{\rho_{uv}\cos\theta_{uv}}{\lambda} \rfloor + \frac{N}{2}$ and $n = \lfloor \frac{\rho_{uv}\sin\theta_{uv}}{\lambda} \rfloor + \frac{N}{2}$, and $\lambda = 2/N$ is the pixel width.

5.4 Accuracy Analysis of the Algorithm

As we have shown in the previous section, the proposed polar approach avoids entirely both geometric error and numerical integration error in computing Zernike moments. If a digital image is represented directly by a grid of pixels structured as in Fig. 5.4, its Zernike moments can be computed accurately by the proposed algorithm without incurring the aforementioned errors. In practice, however, an image is defined over the square grid and it is necessary to convert it into its polar coordinates counterpart. This conversion is done via an interpolation procedure like the formula in (5.14) resulting in some interpolation error.

Our concern in this section is to describe the influence of the interpolation error on the accuracy of the estimate $\hat{A}_{pq}$ defined in (5.12). To this end let us consider the convolution interpolation

[1]The formula for $\hat{f}(\rho_{uv}, \theta_{uv})$ has to be modified slightly for the pixels near the border of the unit disk.

scheme $\hat{f}(\rho,\theta)$ as in (5.14) with the kernel function $h(x)$ such that for $t \geq 1$,

$$|\hat{f}(\rho,\theta) - f(\rho,\theta)| \leq C(f,h)\lambda^t, \tag{5.15}$$

where λ is the pixel width. In (5.15) the constant $C(f,h)$ depends on the smoothness of $f(\cdot,\cdot)$ and the interpolation kernel $h(\cdot)$, and the number t characterizes the accuracy of the interpolation scheme. It is known that $t = 1$ corresponds to the nearest-neighbour algorithm, and $t = 2$ to the linear interpolation, whereas the cubic convolution method meets (5.15) with $t = 3$, [36, 48, 12].

Let us consider the estimate $\hat{A}_{pq}$ in (5.12) with $\hat{f}(\rho_{uv},\theta_{uv})$ in (5.14). We wish to evaluate the size of the difference between $\hat{A}_{pq}$ and the true Zernike moment A_{pq}. To do so we assume that the analog image function $f(x,y)$ is in the class of functions with bounded variation on $\mathbb{D}$. The following theorem shows that the difference $\hat{A}_{pq} - A_{pq}$ decreases like $\max(\lambda^t,\lambda)$.

Theorem 5.1 *Let $f(\cdot,\cdot)$ be a function of bounded variation on $\mathbb{D}$. Let $\hat{A}_{pq}$, defined in (5.12), be the estimate of A_{pq} utilizing the interpolation scheme in (5.14) which satisfies the assumption in (5.15). Then we have*

$$|\hat{A}_{pq} - A_{pq}| \leq \left(\frac{p+1}{\pi}\right)^{1/2} \left\{8C^2(f,h)\lambda^{2t} + 4f_{\max}\mathcal{V}(f)\lambda^2\right\}^{1/2}, \tag{5.16}$$

where $\mathcal{V}(f)$ is the total variation of $f(\cdot,\cdot)$ over $\mathbb{D}$ and $f_{\max} = \max_{(x,y)\in\mathbb{D}} f(x,y)$.

The proof of this theorem can be found in Appendix A.6. The bound in (5.16) contains two unrelated terms. The first one, being of order $O(\lambda^t)$, characterizes the applied interpolation scheme of order t. On the other hand, the second term in (5.16) is of order $O(\lambda)$ and describes the discretization error in replacing the integral in (5.5) by the sum appearing in (5.12). This term can be reduced by putting some further smoothness conditions on $f(\cdot,\cdot)$. In fact, if $f(\cdot,\cdot)$ is

differentiable then the term $O(\lambda^2)$ in (5.16) is replaced by the term of order $O(\lambda^3)$. Hence under the assumption of Theorem 5.1 we have

$$\hat{A}_{pq} = A_{pq} + C_1(p+1)^{1/2}\lambda, \tag{5.17}$$

for some positive constant C_1 depending on $f(\cdot,\cdot)$ and $h(\cdot)$.

It is worth noting that as mentioned in [58], the square grid based method given in (5.2) and (5.4) exhibits the following error

$$\check{A}_{pq} = A_{pq} + C_2(p+1)^{1/2}\lambda^{\alpha}, \tag{5.18}$$

where $\frac{1}{2} \leq \alpha < \frac{3}{4}$ is the exponent characterizing the lattice points approximation of a circle and C_2 is a positive constant depending on $f(\cdot,\cdot)$. For image functions possessing the first derivative, the result in (5.17) improves to $O(\lambda^{3/2})$, whereas (5.18) remains the same.

Thus, we have obtained the qualitative result that the proposed method is more accurate than any method based on the square partition of the image plane.

5.5 Empirical Evaluation of the Algorithm Accuracy

In this section, we investigate empirically the improvement of ZM accuracy resulting from the proposed polar approach. It is illustrated from three different perspectives, namely, the moment magnitudes of a constant image, the image reconstruction from a finite set of computed ZMs, and the ZM's magnitude invariance to image rotation.

5.5.1 Improvement of Zernike Moment Accuracy

To illustrate the accuracy of Zernike moment computation, we use a 128×128 image with a constant intensity value 127 as the test image, on which both the conventional Cartesian method and the proposed polar method are applied for the computation of Zernike moments. The reason we choose the constant image is that, in theory, all its Zernike moments $A_{pq} = 0$ except that $A_{0,0} = 127$.

Therefore by looking at the magnitudes of the computed moments, we can tell the performance of the algorithm in terms of accuracy. Considering the symmetry property of Zernike moments, $A_{pq} = A^*_{p,-q}$, we are only concerned with those with $q \geq 0$.

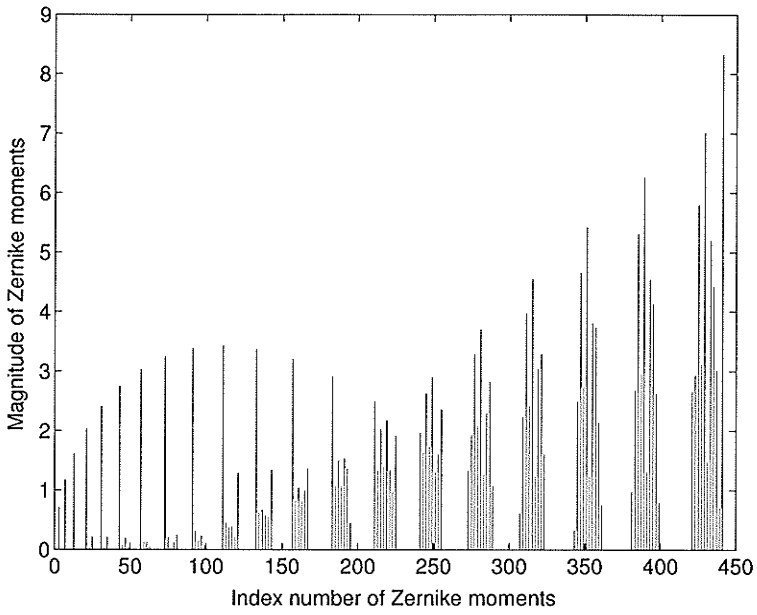


Figure 5.5: The magnitudes of some low-order Zernike moments, of a 128 × 128 constant image, computed with the Cartesian method.

Fig. 5.5 is the result based on the conventional Cartesian method, with magnitudes of Zernike moments up to order 40 of the constant image as a function of the index number of the moments. The moments are placed in the order $\{A_{1,1}, A_{2,0}, A_{2,2}, A_{3,1}, A_{3,3}, A_{4,0}, A_{4,2}, ..., A_{40,40}\}$. There are 440 moments in all.

In contrast, the result obtained from the proposed polar method is shown in Fig. 5.6, in which all the moments have magnitudes below $O(10^{-10})$. Comparing Fig. 5.5 and Fig. 5.6, we can immediately conclude that the proposed approach is vastly superior to the conventional method.

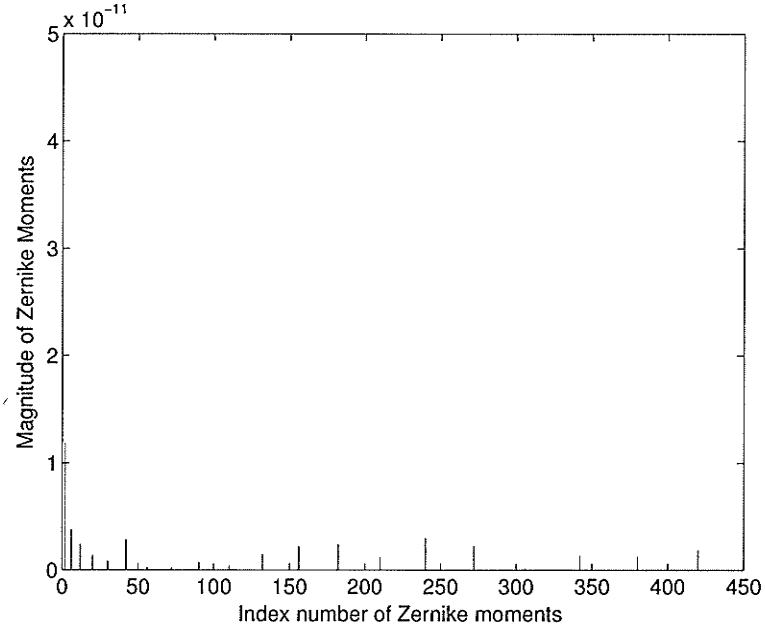


Figure 5.6: The magnitudes of some low-order Zernike moments, of a 128 × 128 constant image, computed with the proposed polar method.

5.5.2 Improvement of Image Reconstruction

Image reconstruction from a finite number of moments is performed with the following formula:

$$\hat{f}(x,y) = \sum_{i=1}^{L} [\hat{A}_{p_i q_i} V_{p_i q_i}(x,y) + \mathbf{I}(q_i \neq 0) \hat{A}_{p_i,-q_i} V_{p_i,-q_i}(x,y)], \tag{5.19}$$

where L is the number of ZMs with non-negative repetitions for image reconstruction, and $\mathbf{I}(\cdot)$ is the indicator function. It was shown in [45] that the reconstruction error consists of two parts. One part of the error is due to the finite value of the parameter L, and the other comes from the inaccuracy of Zernike moments $\hat{A}_{pq}$. The former can be reduced by increasing L, while the latter was believed to be inevitable due to the inherent geometric error and numerical integration error [45, 58]. As we have already shown above, the geometric error and numerical integration error that used to be inherent in Cartesian coordinate system can actually be avoided by appealing to the polar coordinate system. Therefore, the reconstruction error due to the inaccuracy of $\hat{A}_{pq}$ can now be greatly improved. This is confirmed by simulation results. We use the 128 × 128 Lena image shown in Fig. 5.10(a) to illustrate the image reconstruction performance. The Zernike moments up

to the order of 200 are computed with the conventional Cartesian approach and the proposed polar approach respectively. Then the computed moments are used to reconstruct the images. Shown in Fig. 5.7 are some of the images reconstructed with the Cartesian system-based moments, while Fig. 5.8 displays the images reconstructed with the polar system-based moments. It can be seen from Fig. 5.7 that some erroneous pixels along the border of the unit circle are very obtrusive (clearer on a computer monitor than on paper) in Cartesian ZM-based images, and the number of erroneous pixels increases quickly as the order goes up. However, such erroneous pixels don't exist in the images based on polar ZMs.

To compare more objectively the performances of the two approaches in terms of image reconstruction, we experimented on more different numbers of ZMs for image recovery. To be specific, ZMs up to order $\{p = 2i\}_{i=0}^{100}$ were used to reconstruct the image respectively. The quality of each reconstructed image is measured in terms of PSNR. The test results are shown in Fig. 5.9, in which two important conclusions can be drawn. First, for small orders of ZMs, approximately $p < 20$, the quality of the reconstructed images via polar ZMs is similar to that of the reconstructed images via Cartesian ZMs. But as p becomes larger, the former gets significantly better than the latter. Second, the quality of polar ZM-reconstructed images increases monotonically with p. However, in the Cartesian case, as p increases to a certain point, approximately 40, the image quality reaches its maximum value, after which the image quality deteriorates. This is because the reconstruction error incurred by geometric error and numerical integration error increases with p, and at some point it outweighs the quality gain from the population increase of ZMs [45].

5.5.3 Improvement of Magnitude Invariance

If image $f(x, y)$ is rotated α degrees counterclockwise, the magnitudes of its Zernike moments remain the same according to (4.26), i.e., $|A_{pq}^{(\alpha)}| = |A_{pq}|$. This property holds if ZMs are computed accurately, and otherwise it has to be compromised. Therefore, an inspection of ZM's magnitude change before and after image rotation reveals the accuracy of moment computation. If $|A|$ denotes

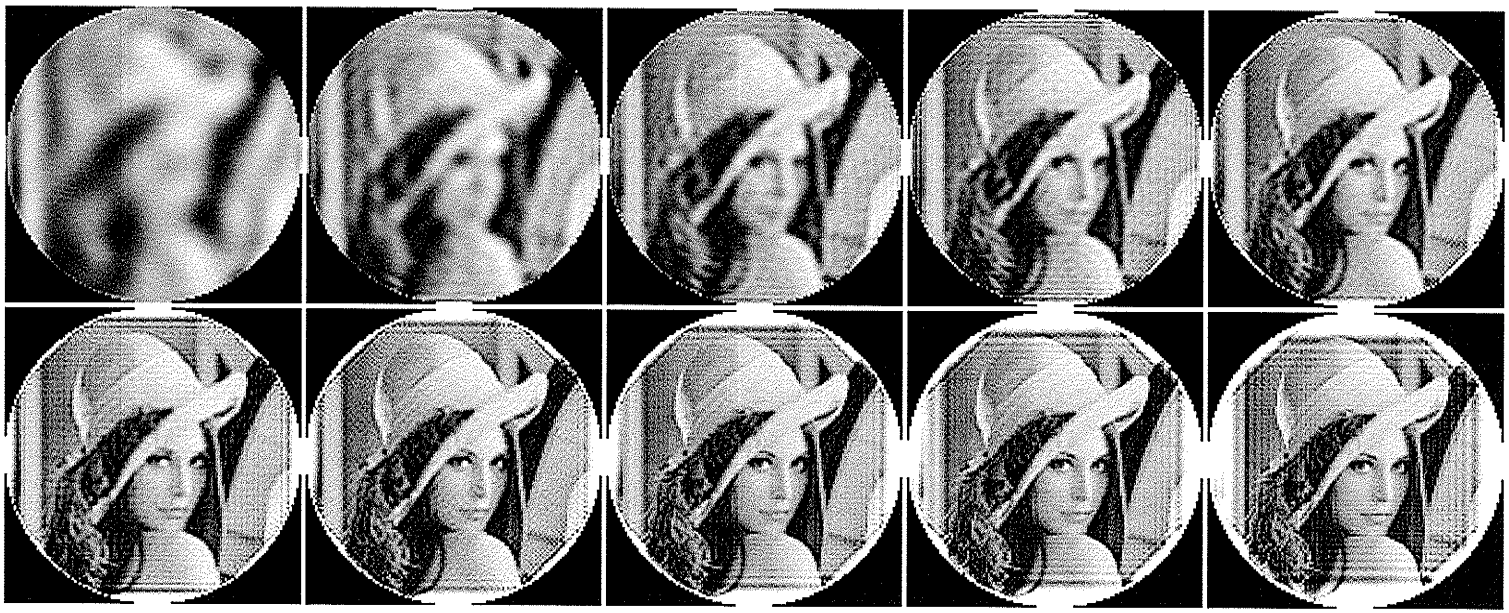


Figure 5.7: Image reconstruction from ZMs computed in Cartesian system. First row from left to right: images reconstructed from ZMs up to order 20, 40, 60, 80 and 100 respectively. Second row from left to right: images reconstructed from ZMs up to order 120, 140, 160, 180 and 200 respectively.

Figure 5.8: Image reconstruction from ZMs computed in polar system. First row from left to right: images reconstructed from ZMs up to order 20, 40, 60, 80 and 100 respectively. Second row from left to right: images reconstructed from ZMs up to order 120, 140, 160, 180 and 200 respectively.

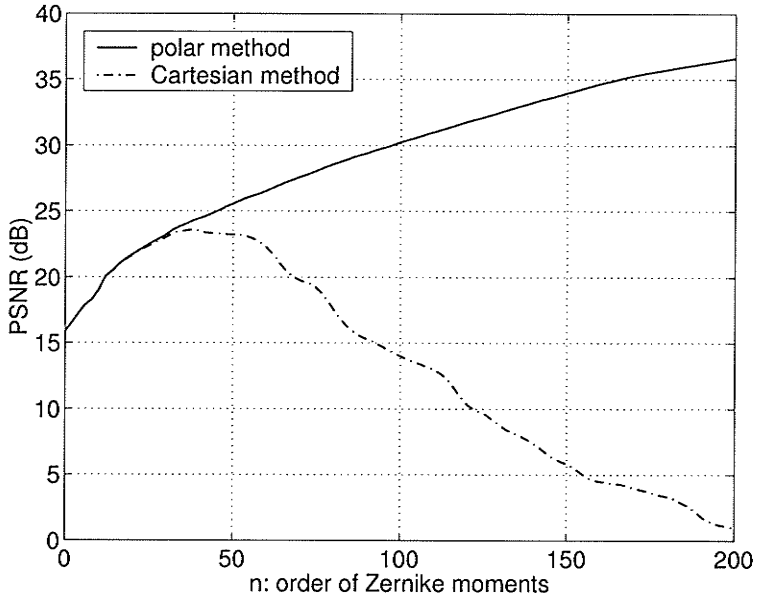
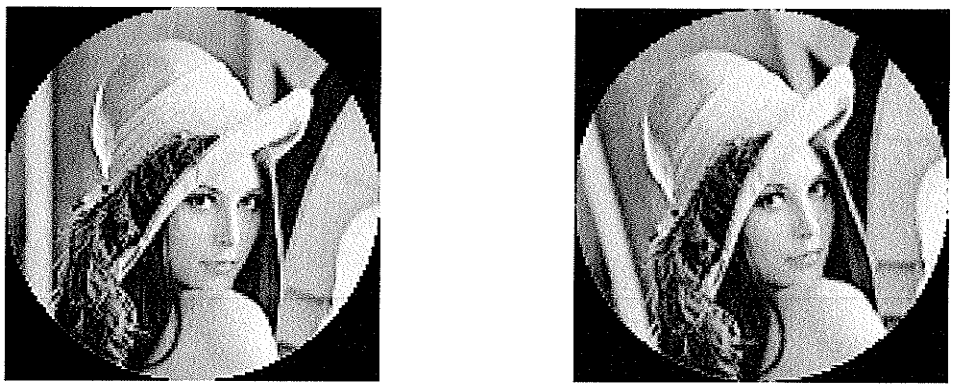


Figure 5.9: The image reconstruction quality in terms of PSNR as a function of the order of Zernike moments, for a comparison of Cartesian and polar moments.

the ZM magnitude of the image before rotation, while $|A^{(r)}|$ denotes that of the image after rotation, we are interested in the signal $\Delta A = |A^{(r)}| - |A|$. As an example, the Lena image is rotated by 15° to yield the image of Fig. 5.10(b). Shown in Fig. 5.11(a) is ΔA in the case of Cartesian approach, while Fig. 5.11(b) illustrates ΔA as the result of the polar approach. It is clear that the polar ZMs greatly outperform the Cartesian ZMs in terms of rotational invariance.

To quantitatively evaluate the improvement of magnitude invariance, we define the mean-square-



(a) (b)

Figure 5.10: The test images. (a) Original 128 × 128 Lena. (b) Lena rotated by 15°.

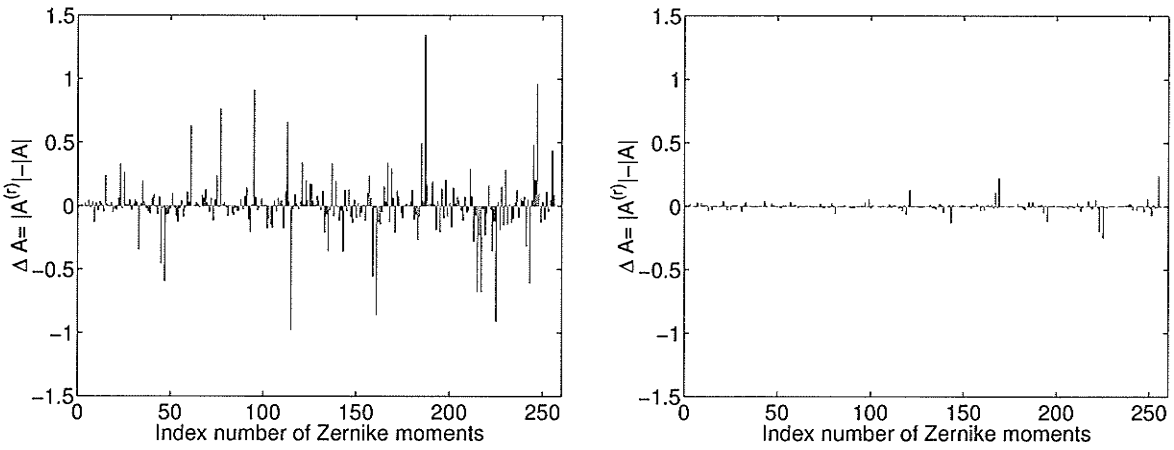


Figure 5.11: The proposed method improves Zernike moments' rotational invariance. (a) Conventional Cartesian approach: the magnitude difference of the first 256 ZMs of Original Lena and those of 15°-rotated Lena. (b) Proposed polar approach: the magnitude difference of the first 256 ZMs of Original Lena and those of 15°-rotated Lena.

error of ZM magnitudes as

$$\mathrm{MSE}(|A|, |A^{(r)}|) = \frac{1}{L}\sum_{i=1}^{L}(|A_{p_iq_i}| - |A^{(r)}_{p_iq_i}|)^2, \tag{5.20}$$

where L is the number of ZMs involved in the evaluation. Depicted in Fig. 5.12 are $\mathrm{MSE}(|A|, |A^{(r)}|)$ in cases of both the Cartesian approach and the polar approach. We experimented with rotation angles from 0° to 90°, with a 2.25° interval. For each rotation angle, we computed the first 256 ZMs of the rotated image with both Cartesian approach and polar approach respectively, and then obtained the corresponding MSE values according to (5.20). The advantage of the polar approach is obvious. To be more specific, the ratio of two average MSEs $\overline{MSE}_{Cartesian}/\overline{MSE}_{polar} \approx 61.66$.

5.6 Multibit Watermarking with Polar Zernike Moments

With the accuracy greatly improved by means of the proposed algorithm, the polar Zernike moments are more suitable for invariant watermarking than their Cartesian counterparts. The design of the polar ZM-based watermarking system is similar to that of the Cartesian ZM-based system explained in Chapter 4. Depicted in Fig. 5.13 and Fig. 5.14 are the algorithm illustrations of the watermark

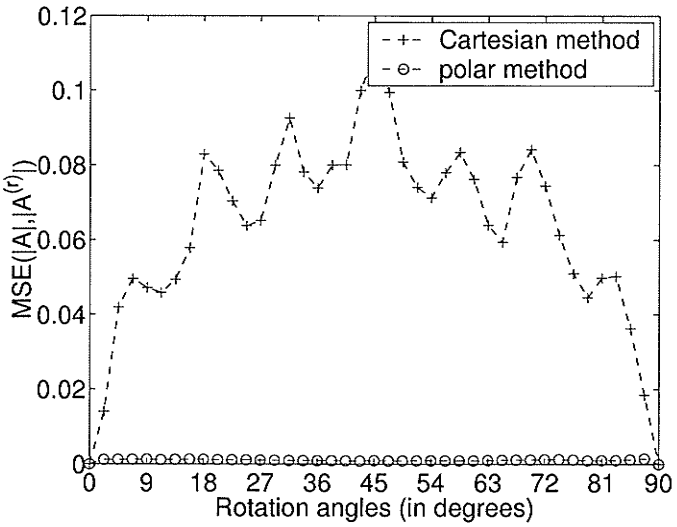


Figure 5.12: Mean-square-error of ZM magnitudes of images shown in Fig. 5.10(a) and Fig. 5.10(b).

embedder and the watermark extractor, respectively.

The polar-ZM based watermarking system shown in Fig. 5.13 and Fig. 5.14 works in almost the same way as the Cartesian ZM-based system, with the following exceptions.

- For the computation of polar Zernike moments, a bicubic interpolation-based image conversion is first performed according to (5.13) and (5.14), which results in a polar image $\hat{f}(\rho_u, \theta_v)$ whose pixels are arranged in a structure shown in Fig. 5.4. Subsequently polar Zernike moments are computed according to 5.12.

- In the Cartesian ZM-based watermarking system described in Chapter 4, A_{pq} with $q = 4i, i \in$ integer, are ruled out for data hiding. However, the current polar ZM-based watermarking is not restricted by this condition.

- Due to the reduced image reconstruction error and rotational invariance error, the polar ZM-based watermarking achieves a better tradeoff between watermark transparency and robustness, as to be witnessed by the simulation results.

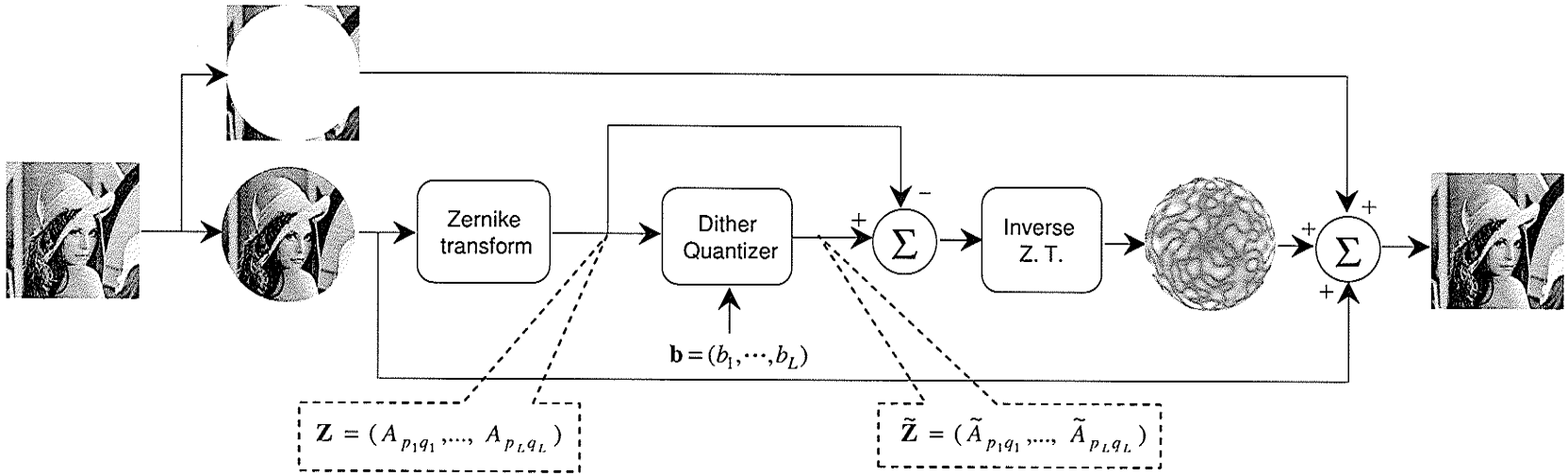


Figure 5.13: The process of ZM-based data embedding.

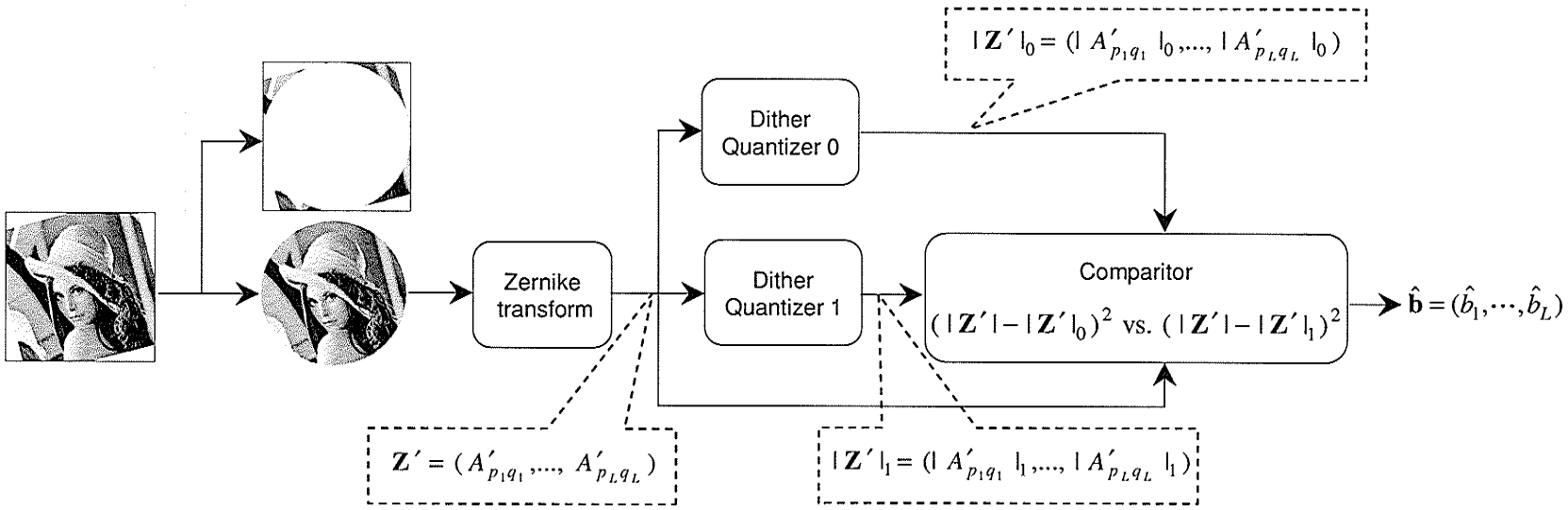


Figure 5.14: The process of data extraction from ZM-watermarked images.

5.7 Simulation Results for Polar ZM-based Watermarking

We carried out some experiments to verify the proposed polar scheme for data hiding algorithm, with emphasis on the watermark robustness to image rotation, scaling and flipping, as well as JPEG lossy compression. We use images shown in Fig. 3.15 as the original images for the tests.

First we look at the watermark robustness measured by the bit error rate, in the case of image rotation. With a data payload of 256 bits to embed, we choose a quantization step such that the average PSNR=42dB. Rotation angles range from 0° to 45° with an increment of 2.25°. The BER are obtained as the average over all the test images, for each of which 100 different randomly generated bit sequences are tested. As a result, no error takes place in all our experiments. Fig. 5.15 shows the result, along with the result from the Cartesian approach [81]. From this experiment we see clearly the superiority of the proposed algorithm.

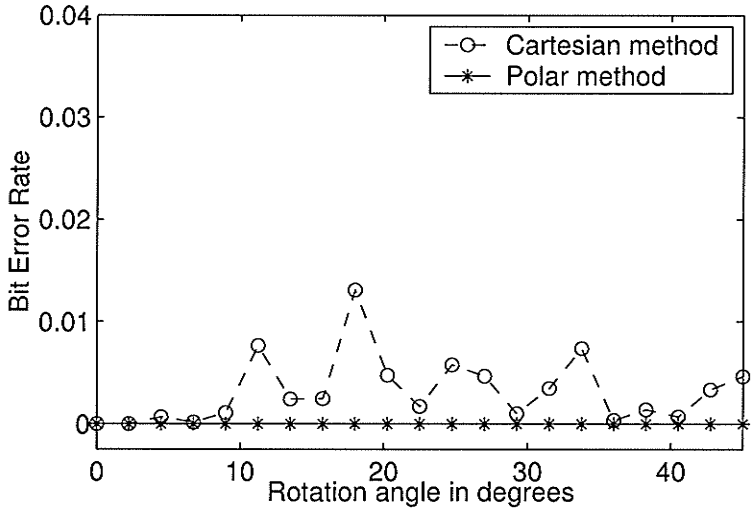


Figure 5.15: Watermark robustness to image rotation.

Image flipping, either horizontal or vertical, is also performed as an attack to the watermarked images. Just as in the Cartesian approach, the polar algorithm is robust against such an attack as well. In all the experiments with image flipping attack, no extraction error occurs, i.e., BER=0.

Image scaling (resizing) is another common form of geometric attacks. We looked at BERs under 16 different scaling levels. A 256×256 watermarked image is scaled to smaller sizes, ranging

from 128×128 to 248×248 with interval 8 of side length. Fig. 5.16 shows the average results for embedding 256 bits with PSNR=42dB in the images of Fig. 3.15, each tested with 100 different bit sequences. Compared with the Cartesian method, the advantage of the proposed polar algorithm is significant.

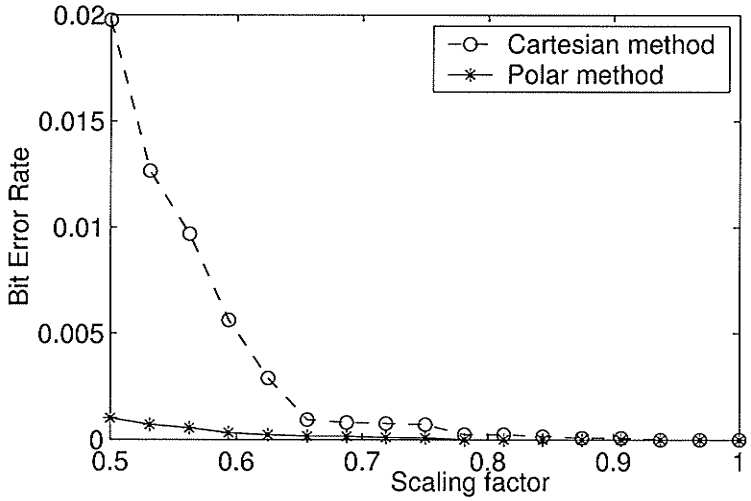


Figure 5.16: Watermark robustness to image scaling (resizing).

Finally we test for lossy compression. The quantization step is taken such that PSNR=40dB, and 256 bits are embedded. We looked at BERs under 20 different JPEG compression levels with quality factors from 20 to 60 with an interval of 2. Fig. 5.17 shows the result, which is the average of 100 individual results. It can be seen that BER decreases rapidly as the quality factor increases, and the proposed polar approach is slightly better than its Cartesian counterpart.

5.8 Chapter Summary

In this chapter, we have proposed a novel approach for high precision computation of ZMs for digital images. In contrast to the traditional Cartesian coordinates-based method which is applied in the watermarking system in Chapter 4, this approach is designed under the polar coordinate system. Detailed aspects of the algorithm, such as the lattice structure of polar pixels and the generation of the polar image from its Cartesian counterpart, have been investigated. It was shown

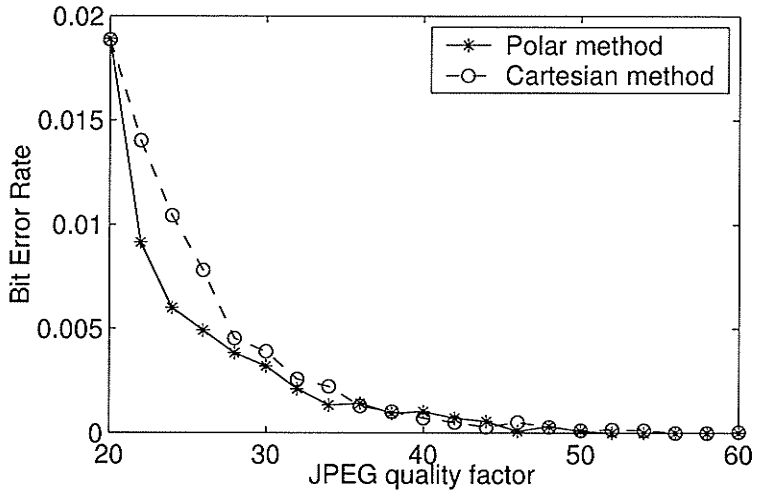


Figure 5.17: Watermark robustness to JPEG lossy compression.

that with this algorithm, the two inherent kinds of errors from which the Cartesian method suffers do not exist. For a digital image given in a Cartesian format, the accuracy of its ZMs is determined by the interpolation scheme involved. Due to the accuracy improvement of ZMs, their invariance property has significantly enhanced, making every ZM eligible for data hiding. The structure of the watermarking system based on polar ZMs is basically the same as that of the system presented in Chapter 4, except the component for ZM computation. Experimental results show that the watermarking system based on polar ZMs is superior to that based on Cartesian ZMs in terms of the error rates of extracted data. The polar ZMs make the watermarks more robust against typical geometric distortions, including image rotation, scaling, and JPEG lossy compression etc.

Chapter 6

Summary and Future Work

6.1 Conclusions

In this thesis we deal with some fundamental issues in image watermarking, focusing on the design of a multibit watermark with robustness to common types of distortion.

From the perspective of watermark communication, most of the proposed algorithms in the area of digital watermarking fall into two categories: coherent systems and non-coherent systems. The former usually employ pseudo-noise sequences as the carrier of the data to be embedded, and rely on the same pseudo-noise sequences for data retrieval, often with a correlator. The advantage of coherent systems include a high level of security and good robustness to additive noise, signal amplitude scaling etc. However, they are not suitable for embedding a high volume of data, and they are sensitive to synchronization of PNSs. Non-coherent systems usually take advantage of quantization techniques for data communication, and hence do not depend on any reference pattern for data extraction. They are capable of embedding a large data payload, and are easy to implement, but are sensitive to change of signal amplitudes.

There are several ways to enhance the robustness of a watermark. Firstly, an appropriate form of watermark signal and use of side information about the cover signal are a good starting point. Secondly, the design of a robust watermark embedder is important, which includes the selection

of robust host features and adaptive signal mixing via an HVS model or multiplicative embedding etc. Thirdly, an optimal watermark detector is crucial. The most commonly used correlator is optimal only for Gaussian features and additive embedding. However, the signal features are often non-Gaussian. The Generalized Gaussian distribution is a good model for DCT/DWT coefficients, and DFT coefficients are better described by the Weibull distribution. The structure of an optimal watermark detector is determined by the statistical model of the host features and the watermark mixing rule, and can be derived according to the Neyman-Pearson criterion or the Bayes rule.

Based on a zero-bit watermark, a multibit watermark can be designed in a number of ways. The popular alternatives include the frequency/space division multiplexing (FDMA) techniques, and the code division (CDMA) multiplexing technique. In terms of error performance, an FDMA-based multibit watermark is equivalent to a CDMA-based one. A more effective form of a multibit watermark is an M-ary watermark, which displays advantages over a CDMA/FDMA watermark, because it is capable of carrying more information with the same watermark energy (host distortion).

M-ary modulation has been limited to small values of M, in particular, $M \leq 256$, due to the computational workload associated with the correlation-based watermark detection. However, we have devised an efficient way to implement an M-ary modulation based watermarking system. We show that if the set of M PNSs are circular shifts of ONE reference PNS, the M correlations can be computed with 2 FFT operations plus 1 IFFT operation, whose workload is almost negligible compared to that of direct calculation of the M correlations, especially when M is large.

It appears that an M-ary watermark mixed with a feature vector of length L can carry $\log_2 L$ bits of information at most, i.e. the value of M is restricted to be $M \leq L$, because L is the total number of circular shifts of a PNS with length L. However we show that this limitation can be circumvented. The solution is that a reference PNS of length M, $M > L$, is used, and its M circular shifts can be used for M-ary modulation after being shortened by a window of width L. Such an extended M-ary watermark can be decoded equally efficiently with 2 FFT operations plus 1 IFFT

operation, with the feature vector zero-padded to the length of M first.

Geometric attacks are a type of attack which is easy to perform, and nevertheless they make most of the existing watermarking algorithms vulnerable. Researchers have been tackling the watermark robustness to this type of attacks, but have not addressed the problem satisfactorily. The approaches existing in literature can be roughly classified into three types, i.e., a template-based inversion techniques, normalization-based technique, and invariance-based techniques. Our algorithms presented in this thesis, which are based on Zernike/pseudo-Zernike moments, fall into the third type. The rationale of our ZM-based approach lies in one of the attractive properties of ZMs, namely, the magnitude invariance to image rotation and flipping.

ZMs can be computed under Cartesian coordinate system, as performed conventionally. We show that not all the Cartesian ZMs are suitable for invariant data hiding, because the moments A_{pq} with $q = 4i$ ($i \in$ integer) cannot be computed accurately, and hence their magnitudes are not invariant at all. Based on this finding, a ZM-based multibit watermarking scheme was designed, in which the magnitudes of ZMs are quantized via dither modulation. Simulation results show that such a ZM-based watermark has robustness against geometric attacks including image rotation, flipping, scaling, moderate cropping, aspect ratio change etc., and a number of common image processing manipulations such as lossy compression, noise addition, filtering, and so on.

The accuracy problem of ZMs can be addressed effectively in the polar coordinate system. We formulate a novel approach to accurate ZM computation. Under this approach, a given Cartesian image is first resampled into a polar image whose trapezoid-like pixels are arranged in a structure as illustrated by Fig. 5.4. Formulas to compute the ZMs of such a polar image are derived, with zero geometric error and integral error, from which Cartesian ZMs suffer inevitably. The only error of ZMs in this approach comes from the interpolation-based image resampling. However, this error is shown to be much smaller than the geometric error and integral error inherent in Cartesian ZMs. The considerably improved accuracy of ZMs yields much better rotational invariance and a significantly lower level of image reconstruction error. These two factors are crucial in the successful

design of the polar ZM-based data hiding.

With the improved rotational invariance, every ZM is now eligible for date hiding. The experimental results have verified the advantages of the polar ZM-based watermark.

6.2 Contributions

Of all the work finished in this thesis, the following results are the major contributions to the field of digital watermarking.

1. A framework on the design of a robust watermark has been proposed, which covers three different aspects, i.e., the design of a robust watermark signal form, the design of a robust watermark embedder, and the design of a robust watermark extractor. For each aspect, some concrete measures were suggested.

2. M-ary phase modulation, which is a special kind of of M-ary modulation, was devised. Based on a circular PNS, it reduces the formidable computational workload in the demodulation of an embedded M-ary message to a very low level by means of a couple of FFT operations.

3. An extended M-ary phase modulation scheme was designed, which overcomes the limitation of the number of host features used for data hiding. Based on this scheme, M is no longer limited by the length of the feature vector. For instance, an M-ary watermark with $M = 2^{20}$ is practical and easy to implement.

4. A theoretical analysis on the error performance of PNS-based M-ary modulation was given, which reveals that the use of an M-ary modulator is advantageous in the design of a multibit watermark, and a larger M is always preferable.

5. We have found and proved that for digital images, only a subset of ZMs/PZMs can be accurately computed in Cartesian coordinates. Based on this, we have shown that some ZMs/PZMs are suitable for watermarking while others should be avoided.

6. A detailed design of a multibit watermark in the ZM/PZM domain was given, which has robustness to some common geometric distortions including image rotation, scaling and flipping etc., as well as other regular processing such as lossy compression and various filtering manipulations.

7. A novel algorithm under polar coordinate system has been formulated for the accurate computation of Zernike moments. It was shown to be advantageous over the conventionally used Cartesian algorithms. It improves significantly the rotational invariance of ZMs, and reduces the error of image reconstruction with ZMs. The polar ZM based data hiding yields better tradeoffs among watermark transparency, robustness and the amount of data embedded.

8. The proposed polar coordinates based methodology for the accurate computation of Zernike moments can be applied to the computation of other circular moments, such as Pseudo-Zernike moments, rotational moments [10, 70], and orthogonal Fourier-Mellin moments [68].

9. The findings and algorithms regarding the computation of Zernike moments are bound to have positive influence on other fields such as image analysis, pattern recognition, medical imaging and ophthalmology [32, 33, 11], in addition to digital watermarking.

6.3 Future Work

As future research directions, the following aspects are worth exploring.

- For blind spread spectrum-based watermark detection, the most commonly used statistical models for the description of host features include Gaussian and generalized Gaussian distributions. In practice, we find that these models are often not very accurate. A better model would improve the performance of a watermark detector. In this direction, a Gaussian mixture model is worth exploring. On the other hand, nonparametric detection can be considered. Nonparametric detectors have such advantages as better adaptability and easier

implementation than parametric detectors. Little has been found in literature on the use of nonparametric detectors for watermarking purposes. The design and performance analysis of nonparametric watermark detectors are definitely a research direction.

- The concept of informed watermarking will be incorporated into the spread spectrum-based algorithm. Informed watermarking, or watermarking with side information [20], refers to the technique of watermark coding or watermark embedding with the cover signal taken into account. Through informed watermarking, the overall performance of a watermarking system can be enhanced.

- An important issue in the ZM/PZM-based watermarking systems proposed in this thesis is the location of the unit disk region, which is crucial for the correct extraction of the embedded bit sequence. In the thesis we deal with this issue by a trial-and-error approach. The systems will be more practical if the search process can be automated.

- The robustness of ZM/PZM-based watermarks are to be extended to cover more distortion types such as the general affine transformations, and even the more demanding image warping operations.

- All the algorithms in this thesis are proposed without explicit consideration of intentional attacks, which leads to the issue of watermark security. In situations like fingerprinting, the security of a watermark is crucial. More work needs to be done for the proposed algorithms to enhance watermark security.

- With the algorithms proposed in this thesis, hundreds of bits can be embedded imperceptibly into an image of size as small as 256×256. Undoubtedly this information capacity of watermarks is sufficient for the needs of many applications. A theoretical analysis of upper bounds for the data capacity of watermarks is yet to be performed [16, 52].

Appendix A

Appendices

A.1 Derivation of (3.22)

The linear correlation between $\mathbf{X}$ and $\mathbf{W}_k$

$$c[k] = \frac{1}{L}\sum_{i=0}^{L-1} X[i]W_k[i] = \frac{1}{L}\sum_{i=0}^{L-1} X[i]W_0[i-k], \qquad k = 0, ..., L-1, \tag{A.1}$$

whose DFT

$$\begin{aligned} C[u] =& \frac{1}{L}\sum_{k=0}^{L-1}\left[\sum_{i=0}^{L-1} X[i]W_0[i-k]\right] e^{-j\frac{2\pi}{L}uk} \\ =& \frac{1}{L}\sum_{i=0}^{L-1} X[i]\sum_{k=0}^{L-1} W_0[i-k]e^{-j\frac{2\pi}{L}uk} \\ =& \frac{1}{L}\sum_{i=0}^{L-1} X[i]e^{-j\frac{2\pi}{L}ui}\sum_{k=0}^{L-1} W_0[i-k]e^{j\frac{2\pi}{L}u[i-k]} \\ =& \frac{1}{L}\mathcal{F}(\mathbf{X})\mathcal{F}^*(\mathbf{W}_0), u = 0, ..., L-1 \end{aligned} \tag{A.2}$$

which leads to

$$c[k] = \frac{1}{L}\mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{X})\mathcal{F}^*(\mathbf{W}_0)\right), k = 0, ..., L-1. \tag{A.3}$$

$\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote DFT and IDFT operations respectively in the above equations.

A.2 Proof of Theorem 3.1

The correlation between $\tilde{\mathbf{X}}$ and $\mathbf{W}_k$ is

$$\mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_k) = \mathcal{C}(\mathbf{X}, \mathbf{W}_k) + a\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k). \tag{A.4}$$

Let's look at the first item on the right hand side. According to the Central Limit Theorem, $\mathcal{C}(\mathbf{X}, \mathbf{W}_k)$ follows Gaussian distribution when L is sufficiently large. Based on the fact that $\mathbf{X}$ and $\mathbf{W}_r$ are independent, the mean and variance of $\mathcal{C}(\mathbf{X}, \mathbf{W}_k)$ can be obtained:

$$E\{\mathcal{C}(\mathbf{X}, \mathbf{W}_k)\} = E\{\frac{1}{L}\sum_{i=0}^{L-1} X[k]W_k[i]\} \tag{A.5}$$

$$= \frac{1}{L}\sum_{i=0}^{L-1} E\{X[i]\}E\{W_k[i]\} = 0. \tag{A.6}$$

$$Var\{\mathcal{C}(\mathbf{X}, \mathbf{W}_k)\} = E\left\{\left(\frac{1}{L}\sum_{i=0}^{L-1} X[i]W_k[i]\right)^2\right\} \tag{A.7}$$

$$= \frac{1}{L^2}\sum_{i=0}^{L-1} E\{(X[i])^2(W_k[i])^2\}$$

$$+ \frac{1}{L^2}\sum_{\{(i,j),i\neq j\}} \underbrace{E\{(X[i]W_k[i])(X[j]W_k[j])\}}_{=0} \tag{A.8}$$

$$= \frac{1}{L^2}\sum_{i=0}^{L-1} \underbrace{E\{(X[i])^2\}}_{=\sigma_{\mathbf{X}}^2}\underbrace{E\{(W_k[i])^2\}}_{=1} = \frac{\sigma_{\mathbf{X}}^2}{L}. \tag{A.9}$$

We can analyze the second item on the right hand side of (A.4) in a similar way. When $k \neq m$,

$$E\{\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k)\} = 0, \tag{A.10}$$

$$Var\{\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k)\} = \frac{1}{L}. \tag{A.11}$$

When $k = m$,

$$E\{\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k)\} = \frac{1}{L}\sum_{i=0}^{L-1} E\{(W_m[i])^2\} = 1, \tag{A.12}$$

$$Var\{\mathcal{C}(\mathbf{W}_m, \mathbf{W}_k)\} = E\left\{\frac{1}{L^2}\left(\sum_{i=0}^{L-1}(W_m[i])^2\right)^2\right\} - 1 \tag{A.13}$$

$$= \frac{1}{L^2}\sum_{i=0}^{L-1} E\{(W_m[i])^4\}$$

$$+ \frac{2}{L^2}\sum_{\{(i,j),i\neq j\}} \underbrace{E\{(W_m[i])^2\}E\{(W_m[j])^2\}}_{=1} - 1 \tag{A.14}$$

$$= \frac{1}{L^2}3 + \frac{2}{L^2}\frac{L(L-1)}{2} - 1 = \frac{2}{L}. \tag{A.15}$$

Note that in (A.14), $E\{(W_m[i])^4\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} x^4 e^{-x^2/2}dx = 3$, and involved in the second summation there are $\binom{L}{2}$ products in total.

Combining (A.4),(A.6),(A.9),(A.10),(A.11),(A.12) and (A.15), we obtain

$$\mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_k) \sim \begin{cases} \mathcal{N}(0, \frac{\sigma_{\mathbf{X}}^2 + a^2}{L}) & \text{if } k \neq m \\ \mathcal{N}(a, \frac{\sigma_{\mathbf{X}}^2 + 2a^2}{L}) & \text{if } k = m \end{cases}.$$

A.3 Proof of Theorem 3.2

Let $\mathbf{c} = (c[0], ..., c[M-1])$ where $c[k] = \mathcal{C}(\tilde{\mathbf{X}}, \mathbf{W}_k)$. According to Theorem 3.1, we have

$$\frac{c[m]}{a} \sim \mathcal{N}(1, \sigma_1^2), \text{ and}$$
$$\frac{c[i]}{a} \sim \mathcal{N}(0, \sigma_0^2), i \in \{0, ...M-1\} \text{ but } i \neq m,$$

where $\sigma_1^2 = \frac{r+2}{L}$, $\sigma_0^2 = \frac{r+1}{L}$, and $r = \frac{\sigma_{\mathbf{X}}^2}{a^2}$. When $\sigma_{\mathbf{X}}^2 \gg a^2$, $\sigma_1^2 \approx \sigma_0^2 \approx \frac{r}{L}$.

Let $c_{\max} = \max_{\{i,i\neq m\}}\{c[i]\}$, $F_{\max}(x)$ and $F_i(x)$ denote the distribution function of $c_{\max}/a$ and $c[i]/a$ respectively, then $F_{\max}(x) = \prod_{\{i,i\neq m\}} F_i(x)$. Thus

$$P(\frac{c_{\max}}{a} < x) = \prod_{i=0,\neq m}^{M-1} P(\frac{c[i]}{a} < x) = \left[1 - Q(\frac{x}{\sigma_c})\right]^{M-1},$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{x^2}{2}} dx$, and $\sigma_c = \sqrt{\frac{r}{L}} = \frac{\sigma_X}{a\sqrt{L}}$. Following the ML principle in (3.18), the embedded symbol m is correctly decoded only when $c[m] > c_{\max}$, and therefore the probability of correct estimation is

$$P_c = \int_{-\infty}^{\infty} \phi(x) P(\frac{c_{\max}}{a} < x) dx,$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{(x-1)^2}{2\sigma_c^2}}$ is the probability density function of $\frac{c[m]}{a}$. Finally we have

$$P_e = 1 - \int_{-\infty}^{\infty} \phi(x) \left[1 - Q(\frac{x}{\sigma_c})\right]^{M-1} dx.$$

A.4 Proof of Theorem 4.1

Assume we have an image with constant intensity T. For an arbitrary pixel point $A(x, y)$ within the unit circle in Fig. A.1, there must be 3 other pixel points whose coordinates have a fixed relationship with $A(x, y)$: $B(-y, x)$, $C(-x, -y)$ and $D(y, -x)$. Equivalently in polar coordinates, they are: $A(\rho, \theta)$, $B(\rho, \frac{\pi}{2} + \theta)$, $C(\rho, \pi + \theta)$ and $D(\rho, \frac{3\pi}{2} + \theta)$.

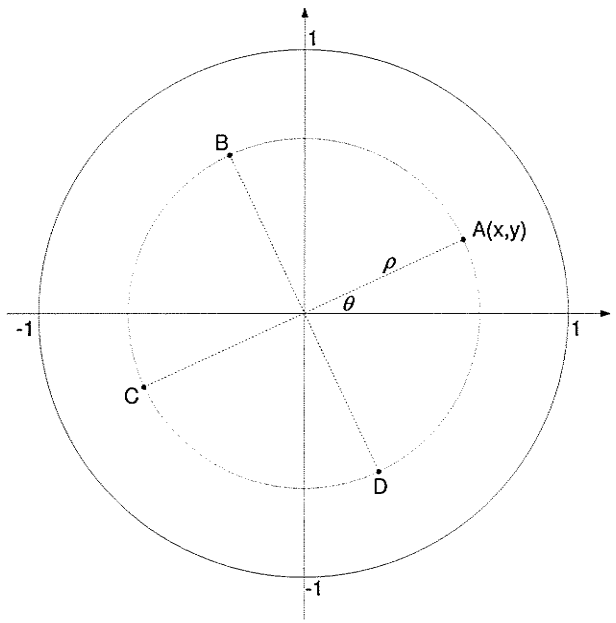


Figure A.1: The pixel points on a circle

First let's group all the pixel points inside the unit circle into pairs like (A, C) and (B, D), and

then

$$\begin{aligned} A_{pq} &= k \sum_{\{(x_u,y_v)\in\mathcal{D}\}} TV_{pq}^*(x_u,y_v) \\ &= kT \sum_{\{\text{All pairs}\}} R_{pq}(\rho)e^{-jq\theta} + R_{pq}(\rho)e^{-jq(\pi+\theta)} \\ &= kT \sum_{\{\text{All pairs}\}} R_{pq}(\rho)e^{-jq\theta}[1+(-1)^q] \end{aligned} \tag{A.16}$$

which is zero if q is odd, unknown otherwise.

Next we are concerned with the property of A_{pq} when q is even. Let's group all the pixel points inside the unit circle in another way, to form pairs like (A,B) and (C,D), and then

$$\begin{aligned} A_{pq} &= k \sum_{\{(x_u,y_v)\in\mathcal{D}\}} TV_{pq}^*(x_u,y_v) \\ &= kT \sum_{\{\text{All pairs}\}} R_{pq}(\rho)e^{-jq\theta} + R_{pq}(\rho)e^{-jq(\frac{\pi}{2}+\theta)} \\ &= kT \sum_{\{\text{All pairs}\}} R_{pq}(\rho)e^{-jq\theta}[1+(-j)^q] \end{aligned} \tag{A.17}$$

which is zero if q is even but not divisible by 4, and nonzero in general if q is a multiple of 4.

The proof of $A_{pq} = T(1+O(\lambda^\gamma))$ if $p=q=0$ is omitted. See [45, 58] for details.

A.5 Proof of Theorem 4.2

Let $\{A_{p_k,q_k}\}_{k=1}^L$ denote the selected set of ZMs/PZMs of an $N \times N$ stochastic image process $f(x_i,y_j)$, and $\{\tilde{A}_{p_k,q_k}\}_{k=1}^L$ denote the dither modulated version of $\{A_{p_k,q_k}\}_{k=1}^L$, i.e., the corresponding ZM/PZM set of the watermarked image $\tilde{f}(x_i,y_j)$. Then the average square error of the watermarked image

$$\sigma_e^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}[\tilde{f}(x_i,y_j) - f(x_i,y_j)]^2 \tag{A.18}$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\sum_{k=1}^{L}(\varepsilon_{p_k,-q_k}V_{p_k,-q_k} + \varepsilon_{p_k,q_k}V_{p_k,q_k})\right)^2 \tag{A.19}$$

$$= \frac{1}{4}\sum_{k=1}^{L}\left(\frac{\pi}{p_k+1}|\varepsilon_{p_k,-q_k}|^2 + \frac{\pi}{p_k+1}|\varepsilon_{p_k,q_k}|^2\right) \tag{A.20}$$

$$= \frac{\pi}{2}\sum_{k=1}^{L}\frac{1}{p_k+1}|\varepsilon_{p_k,q_k}|^2 \tag{A.21}$$

$$= \frac{\pi}{2}\sum_{k=1}^{L}\frac{1}{p_k+1}\left||\tilde{A}_{p_k,q_k}|e^{j\theta_k} - |A_{p_k,q_k}|e^{j\theta_k}\right|^2 \tag{A.22}$$

$$= \frac{\pi}{2}\sum_{k=1}^{L}\frac{1}{p_k+1}(|\tilde{A}_{p_k,q_k}| - |A_{p_k,q_k}|)^2. \tag{A.23}$$

In (A.19), $\varepsilon_{p_k,q_k} = \tilde{A}_{p_k,q_k} - A_{p_k,q_k}$ and $\varepsilon_{p_k,-q_k} = \tilde{A}_{p_k,-q_k} - A_{p_k,-q_k}$, $V_{p_k,-q_k}$ and V_{p_k,q_k} are Zernike/pseudo-Zernike polynomials. (A.19) follows from the linearity of the inverse Zernike/pseudo-Zernike transform, see (4.44) and (4.45) for details. (A.20) is due to the Parseval formula [1][49]. Eq. (A.21) results from the symmetry property of ZM/PZM. Now we get the expected value of square error

$$E\{\sigma_e^2\} = \frac{\pi}{2}\sum_{k=1}^{L}\frac{1}{p_k+1}E\{(|\tilde{A}_{p_k,q_k}| - |A_{p_k,q_k}|)^2\} \tag{A.24}$$

$$= \frac{\pi}{2}\sum_{k=1}^{L}\frac{1}{p_k+1}\frac{\Delta^2}{12} \tag{A.25}$$

$$= \frac{\pi\Delta^2}{24}\sum_{k=1}^{L}\frac{1}{p_k+1}. \tag{A.26}$$

Equation (A.25) follows from the well-known fact that the quantization noise is uniformly distributed in $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ and the expected power of the quantization noise is $\frac{\Delta^2}{12}$.

[1]Strictly speaking, the Parseval formula holds only approximately since we employ the discrete version of the Zernike moments not the continuous one.

A.6 Proof of Theorem 5.1

Let us first observe that

$$\hat{A}_{pq} - A_{pq} = \frac{p+1}{\pi} \sum_{u=1}^{U} \sum_{v=1}^{V(2u-1)} \int\int_{\Omega_{uv}} [\hat{f}(\rho_{uv}, \theta_{uv}) - f(\rho, \theta)] R_{pq}(\rho) e^{-jq\theta} \rho d\rho d\theta. \tag{A.27}$$

By Cauchy-Schwarz inequality we obtain

$$\left|\hat{A}_{pq} - A_{pq}\right| \leq \frac{p+1}{\pi} \left\{ \int\int_{\mathbb{D}} \left| R_{pq}(\rho) e^{-jq\theta} \right|^2 \rho d\rho d\theta \right\}^{1/2} \times \left\{ \sum_{u=1}^{U} \sum_{v=1}^{V(2u-1)} \int\int_{\Omega_{uv}} \left| \hat{f}(\rho_{uv}, \theta_{uv}) - f(\rho, \theta) \right|^2 \rho d\rho d\theta \right\}^{1/2}. \tag{A.28}$$

Since $\int\int_{\mathbb{D}} |R_{pq}(\rho) e^{-jq\theta}|^2 \rho d\rho d\theta = \frac{\pi}{p+1}$, it remains to consider the second term in (A.28).

Let us notice that

$$\int_{\Omega_{uv}} \int \left| \hat{f}(\rho_{uv}, \theta_{uv}) - f(\rho, \theta) \right|^2 \rho d\rho d\theta \leq L_1 + L_2, \tag{A.29}$$

where

$$L_1 = 2 \int_{\Omega_{uv}} \int \left| \hat{f}(\rho_{uv}, \theta_{uv}) - f(\rho_{uv}, \theta_{uv}) \right|^2 \rho d\rho d\theta, \tag{A.30}$$

and

$$L_2 = 2 \int_{\Omega_{uv}} \int |f(\rho_{uv}, \theta_{uv}) - f(\rho, \theta)|^2 \rho d\rho d\theta. \tag{A.31}$$

By virtue of (5.15)

$$L_1 \leq 2C^2(f, h) \lambda^{2t} \lambda^2, \tag{A.32}$$

where, without loss of generality, we assume that the area of Ω_{uv} is not greater than λ^2.

Concerning the term L_2, we have

$$L_2 \leq 2 \underset{\Omega_{uv}}{Osc}(f) \int_{\Omega_{uv}} \int |f(\rho_{uv}, \theta_{uv}) - f(\rho, \theta)| \rho d\rho d\theta \leq 4 f_{\max} \underset{\Omega_{uv}}{Osc}(f) \lambda^2, \tag{A.33}$$

where $\underset{\Omega_{uv}}{Osc}(f) = \max_{(\rho,\theta),(\rho',\theta')\in\Omega_{uv}} \{|f(\rho,\theta) - f(\rho',\theta')|\}$ is the oscillation of $f(\rho,\theta)$ over the sector Ω_{uv}, and $f_{\max} = \max_{(x,y)\in\mathbb{D}} f(x,y)$.

Substituting (A.29), (A.32) and (A.33) into (A.28), and noting that $\mathcal{V}(f) = \sum_{u=1}^{U} \sum_{v=1}^{V(2u-1)} \underset{\Omega_{uv}}{Osc}(f)$ we can obtain the following bound.

$$|\hat{A}_{pq} - A_{pq}| \leq \left(\frac{p+1}{\pi}\right)^{1/2} \left\{8C^2(f,h)\lambda^{2t} + 4f_{\max}\mathcal{V}(f)\lambda^2\right\}^{1/2}. \tag{A.34}$$

This proves the assertion of Theorem 5.1.

Bibliography

[1] M. Alghoniemy and A. H. Tewfik. Geometric distortion correction through image normalization. In *IEEE Int. Conf. Multimedia and Expo*, pages 1291–1294, 2000.

[2] M. Alghoniemy and A. H. Tewfik. Image watermarking by moment invariants. In *IEEE Conference on Image Processing*, pages 73–76, 2000.

[3] M. Alghoniemy and A. H. Tewfik. Geometric invariants in image watermarking. *IEEE Transactions on Image Processing*, 13:145–153, 2004.

[4] R. J. Anderson and F. A. P. Petitcolas. On the limits of steganography. *IEEE Trans. Selected Areas in Communications*, 16(4):474–481, 1998.

[5] M. Barni, F. Bartolini, A. Rosa, and A. Piva. Optimum decoding and detection of multiplicative watermarks. *IEEE Trans. Signal Processing*, 51(4):1118–1123, 2003.

[6] A. B. Bhatia and E. Wolf. On the circle polynomials of Zernike and related orthogonal sets. *Proc. Cambridge Philosophical Soc.*, 50:40–48, 1954.

[7] R. E. Blahut. *Theory and Practice of Error Control Codes.* Addison-Wesley., 1983.

[8] J. A. Bloom, I. J. Cox, T. Kalker, and J. P. M. G. Linnartz. Copy protection for dvd video. *Proceedings of the IEEE*, 87(7):1267–1276, 1999.

[9] M. Born and E. Wolf. *Principles of Optics.* Pergamon, Oxford, 1975.

[10] J. F. Boyce and W. J. Hossack. Moment invariants for pattern recognition. *Patter Recognition Letters*, 1(5-6):451–456, 1983.

[11] C. E. Campbell. A new method for describing the aberrations of the eye using Zernike polynomials. *Optom. Vision Science*, 80(1):77–83, 2003.

[12] E. Catmull and R. Rom. A class of local interpolating splines. In *Computer Aided Geometric Design*, pages 317–326. R. E. Barnhill and R. F. Riesenfeld, Eds. New York: Academic, 1974.

[13] B. Chen and G. W. Wornell. Quantization index modulation methods: a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423–1443, 2001.

[14] Q. Cheng and T. S. Huang. An additive approach to transform-domain information hiding and optimum detection structure. *IEEE Trans. Multimedia*, 3(3):273–284, 2001.

[15] Q. Cheng and T. S. Huang. Robust optimum detection of transform domain multiplicative watermarks. *IEEE Trans. Signal Processing*, 51(4):906–924, 2003.

[16] A. S. Cohen and A. Lapidoth. The Gaussian watermarking game. *IEEE Transactions on Information Theory*, 48(6):1639–1667, 2002.

[17] M. H. M. Costa. Writing on dirty paper. *IEEE Trans. Information Theory*, 29(3):439–441, 1983.

[18] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Vol. I.* New York: Interscience, 1953.

[19] I. J. Cox, J. Killian, T. Leighton, and T. Shanmoon. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Processing*, 6(12):1673–1687, 1997.

[20] I. J. Cox, M. L. Miller, and J. A. Bloom. *Digital Watermarking.* Morgan Kaufmann Publishers, 2001.

[21] S. Craver, N. Memon, B. Yeo, and M. Yeung. On the invertibility of invisible watermarking techniques. In *Proc. Int. Conf. Image Processing*, volume 1, pages 540–543, 1997.

[22] S. Craver, N. Memon, B. Yeo, and M. Yeung. Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications*, 16(5):573–586, 1998.

[23] 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. 3rd Int. Information Hiding Workshop*, pages 315–330, 1999.

[24] P. Dong and N. P. Galasanos. Affine transform resistant watermarking based on image normalization. In *IEEE Int. Conf. Image Pro.*, pages 489–492, 2002.

[25] M. Farzam and S. Shirani. A robust multimedia watermarking technique using Zernike transform. In *IEEE Int. Workshop Multimedia Signal Processing*, pages 529–534, 2001.

[26] J. Flusser. Refined moment calculation using image block representation. *IEEE Transactions on Image Processing*, 9(11):1977–1978, 2000.

[27] J. Fridrich and M. Goljan. Digital image steganography using stochastic modulation. In *Proc. EI SPIE Santa Clara*, pages 191–202, 2003.

[28] R. Gonzalez and R. Woods. *Digital Image Processing.* Prentice-Hall, Inc., 2002.

[29] J. R. Hernandez, M. Amado, and F. Perez-Gonzalez. Dct-domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE Transactions on Image Processing*, 9(1):55–68, 2000.

[30] J. R. Hernandez and F. Perez-Gonzalez. Statistical analysis of watermarking schemes for copyright protection of images. *Proceedings of the IEEE*, 87(7):1142–1166, 1999.

[31] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, 8:179–187, 1962.

[32] D. R. Iskander, M. J. Collins, and B. Davis. Optimal modeling of corneal surfaces with Zernike polynomials. *IEEE Trans. Biomedical Engineering*, 48(1):87–95, 2001.

[33] D. R. Iskander, M. J. Collins, M. R. Morelande, and M. Zhu. Analyzing the dynamic wavefront aberrations in the human eye. *IEEE Trans. Biomedical Engineering*, 51(11):1969–1980, 2004.

[34] A. K. Jain. *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[35] S. M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory.* Prentice-Hall, 1998.

[36] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. ASSP*, 29(6):1153–1160, 1981.

[37] A. Khotanzad and Y. H. Hong. Invariant image recognition by Zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):489–497, 1990.

[38] H. S. Kim and H. K. Lee. Invariant image watermark using Zernike moments. *IEEE Trans. Circuits and Systems for Vid.*, 13(8):766–775, 2003.

[39] D. Kundur and D. Hatzinakos. Digital watermarking for telltale tamper proofing and authentication. *Proceedings of the IEEE*, 87(7):1167–1180, 1999.

[40] M. Kutter. Watermarking resisting to translation, rotation, and scaling. In *Proc. SPIE Multimedia Systems Applications*, pages 423–431, 1998.

[41] M. Kutter. Performance improvement of spread spectrum based image watermarking schemes through M-ary modulation. *Lecture Notes in Computer Science*, 1728:238–250, 1999.

[42] M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In *Proc. Electronic Imaging '99. Security and Watermarking of Multimedia Contents*, volume 3657, 1999.

[43] S. X. Liao. *Image Analysis by Moments.* Ph.D. Thesis, University of Manitoba, 1993.

[44] S. X. Liao and M. Pawlak. On image analysis by moments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(3):254–266, 1996.

[45] S. X. Liao and M. Pawlak. On the accuracy of Zernike moments for image analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(12):1358–1364, 1998.

[46] C. Y. Lin. *Watermarking and Digital Signature Techniques for Multimedia Authentication and Copyright Protection.* Ph.D. Thesis, Columbia University, 2000.

[47] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Liu. Rotation, scale, and translation resilient watermarking for images. *IEEE Trans. Image Processing*, 10(5):767–782, 2001.

[48] E. Maeland. On the comparison of interpolation methods. *IEEE Transactions on Medical Imaging*, 7(3):213–217, 1988.

[49] S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition.* Academic Press, 1999.

[50] S. G. Mallat. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[51] L. M. Marvel, C. G. Boncelet, and C. T. Retter. Spread spectrum image steganography. *IEEE Transactions on Image Processing*, 8(8):1075–1083, 1999.

[52] P. Moulin and J. A. O'Sullivan. Information-theoretic analysis of information hiding. *IEEE Transactions on Information Theory*, 49(3):563–593, 2003.

[53] R. Mukundan and K. Ramakrishnan. Fast computation of Legendre and Zernike moments. *Pattern Recognition*, 28(9):1433–1442, 1995.

[54] R. Mukundan and K. Ramakrishnan. *Moment Functions in Image Analysis: Theory and Applications.* World Scientific, 1998.

[55] A. Nikolaidis and I. Pitas. Asymptotically optimal detection for additive watermarking in the dct and dwt domains. *IEEE Transactions on Image Processing*, 12(5):563–571, 2003.

[56] J. O'Ruanaidh and T. Pun. Rotation, scale and translation invariant spread spectrum digital image watermarking. *Signal Processing*, 66(8):303–317, 1998.

[57] A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Process.* 4th Edition, Mc. Graw Hill, 2002.

[58] M. Pawlak and S. X. Liao. On the recovery of a function on a circular domain. *IEEE Transactions on Information Theory*, 48(10):2736–2753, 2002.

[59] M. Pawlak and Y. Xin. Robust image watermarking: an invariant domain approach. In *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2002*, volume 2, pages 885–888, 2002.

[60] S. Pereira and T. Pun. Robust template matching for affine resistant image watermarks. *IEEE Transations on Image Processing*, 9(6):1123–1129, 2000.

[61] F. A. P. Petitcolas. *http://www.petitcolas.net/fabien/watermarking/stirmark/.* [Online].

[62] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on copyright marking systems. In *Proc. Workshop Information Hiding*, pages 218–238, 1998.

[63] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information hiding–a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[64] C. I. Podilchuk and W. Zeng. Image-adaptive watermarking using visual models. *IEEE Trans. Selected Areas in Communications*, 16(4):525–539, 1998.

[65] J. G. Proakis. *Digital Communications, 4th Edition.* New York, McGraw Hill, 2000.

[66] T. H. Reiss. The revised fundamental theorem of moment invariants. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(8):830–834, 1991.

[67] I. Rothe, H. Susse, and K. Voss. The method of normalization to determine invariants. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(4):366–375, 1996.

[68] Y. Sheng and L. Shen. Orthogonal Fourier-Mellin moments for invariant pattern recognition. *JOSA-A*, 11(6):1748–1757, June 1994.

[69] M. R. Teague. Image analysis via the general theory of moments. *J. Optical Soc. Am.*, 70:920–930, 1980.

[70] C. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.

[71] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu. Anti-collusion fingerprinting for multimedia. *IEEE Transactions on Signal Processing*, 51(4):1069–1087, 2003.

[72] H. L. van Trees. *Detection, Estimation, and Modulation Theory.* John Wiley & Sons, Inc., 2001.

[73] S. Voloshynovskiy, F. Deguillaume, and T. Pun. Multibit digital watermarking robust against local nonlinear geometrical distortions. In *IEEE Conference on Image Processing*, pages 999–1102, 2001.

[74] Z. J. Wang, M. Wu, H. V. Zhao, W. Trappe, and K. J. R. Liu. Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation. *IEEE Transactions on Image Processing*, 14(6):804–821, 2005.

[75] A. B. Watson. Efficiency of an image code based on human vision. *J. Opt. Soc. Amer. A*, 4(12):2401–2417, 1987.

[76] S. G. Wilson. *Digital Modulation and Coding.* Prentice Hall, 1996.

[77] R. Wolfgang, C. I. Podilchuk, and E. J. Delp. Perceptual watermarks for digital images and video. *Proceedings of IEEE*, 87(7):1108–1126, 1999.

[78] M. Wu and B. Liu. A public key watermark for image verification and authentication. In *IEEE International Conference on Image Processing*, volume 1, pages 455–459, 1998.

[79] M. Wu and B. Liu. Watermarking for image authentication. In *IEEE Int. Conf. Image Processing*, volume 2, pages 437–441, 1998.

[80] Y. Xin, S. Liao, and M. Pawlak. Geometrically robust image watermarking via pseudo-Zernike moments. In *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2004*, volume 2, pages 939–942, 2004.

[81] Y. Xin, S. Liao, and M. Pawlak. A multibit geometrically robust image watermark based on Zernike moments. In *International Conference on Pattern Recognition (ICPR) 2004*, volume IV, pages 861–864, 2004.

[82] Y. Xin, S. Liao, and M. Pawlak. Using Zernike moments selectively for invariant watermarking. In *Proceedings of Image and Vision Computing New Zealand 2004*, pages 405–410, Akaroa, New Zealand, 21–23 November 2004.

[83] Y. Xin, S. Liao, and M. Pawlak. Accurate computation of Zernike moments in polar coordinates. Submitted to *IEEE Transactions on Image Processing*, 2005.

[84] Y. Xin, S. Liao, and M. Pawlak. Circularly orthogonal moments for robust data hiding. Submitted to *IEEE Transactions on Image Processing*, 2005.

[85] Y. Xin, S. Liao, and M. Pawlak. On the improvement of rotational invariance of Zernike moments. In *International Conference on Image Processing (ICIP) 2005*, Genova, Italy, 11–14 Sept. 2005.

[86] Y. Xin, S. Liao, and M. Pawlak. Robust data hiding with image invariants. In *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2005*, Saskatoon, Canada, 1–4 May 2005.

[87] Y. Xin and M. Pawlak. Multibit data hiding based on CDMA. In *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2004*, volume 2, pages 935–938, 2004.

[88] Y. Xin, M. Pawlak, and S. Liao. Image reconstruction with polar Zernike moments. *Lecture Notes in Computer Science*, volume 3687:394–403, Springer-Verlag, 2005.

[89] W. Zeng and B. Liu. A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images. *IEEE Transactions on Image Processing*, 8(11):1534–1548, 1999.

[90] F. Zernike. Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode. *Physica*, 1(6):689–704, 1934.

[91] H. V. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu. Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting. *IEEE Transactions on Image Processing*, 14(5):646–661, 2005.