Transform techniques based image watermarking : authentication, restoration and copyright protection

Zhu, Xunzhan

2007

Zhu, X. (2007). Transform techniques based image watermarking : authentication, restoration and copyright protection. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4089>

<https://doi.org/10.32657/10356/4089>

Nanyang Technological University

Downloaded on 26 Apr 2023 20:51:40 SGT

Transform Techniques Based Image Watermarking: Authentication, Restoration And Copyright Protection

Zhu Xunzhan

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2007

To my supervisors.

Acknowledgments

I would like to thank both of my supervisors, Professor Anthony T. S. Ho and Professor Pina Marziliano. It is their enthusiasm, patience and guidance that helped me establish the view of research. They are not only my scientific advisors, but my sincere friends. I was particularly impressed by the humor of Professor Ho and the exclusive strictness for math of Professor Marziliano. What I learned from them would be valuable for the whole of my lifetime.

I also wish to thank Professor Guan Yong Liang for sharing his valuable ideas and advices in my effort to my first journal publication. Thanks to Shen Jun, Shu Feng, Zhang Jing and Niladri, with whom I have collaborated in NTU.

I am particularly indebted to my dear husband, Stephen, who always managed to let me believe there is still sunshine when I was driven mad over those papers and codes. My life would not be so beautiful without your company.

I would like to thank my family in China. My parents can never read these English words, though. It is they who told me to face the challenges of life and try out for the best since my childhood. And my baby sister, who was also struggling on her own journey

ACKNOWLEDGEMENTS

towards the admission of a prestigious university throughout the past years. I am so glad that she accomplished her dream in the end.

Last but not least I would like to express my appreciation to the reviewers of my thesis. Thanks for their time and work which contributes significantly to the improvement of this thesis.

Xunzhan

Singapore, Aug 2006

Abstract

Copyright protection and authentication of digital multimedia contents are increasingly becoming more and more important in parallel with the tremendous development of the Internet. To counter this growing information security problem of illegal distribution and counterfeiting, much research effort has been invested in digital watermarking since the last decade. Compared to the traditional methods of protecting digital content, e.g., cryptography and digital signature, digital watermarking possesses distinct advantages in that it places information directly into the content, which is imperceptible and can neither be removed nor decoded without the required secret key. In this thesis, novel digital watermarking schemes have been proposed by incorporating a wide spectrum of transform methods ranging from orthogonal transforms to sampling theory in approximation subspaces, shedding light on some important applications of the watermarking techniques.

This thesis is divided into four parts. In the first part we address the exact authentication of digital images: we derive a fragile watermarking scheme based on encoding of the zeros of the z -transform for literal authentication of digital images. Our watermarking method is designed by exploiting the sensitivity of the positions of the zeros of the

z -transform around the unit circle to any change made on the host image. The system can detect even one bit's change on the watermarked image. Such bit-sensitivity is desirable in the authentication of master copies of important images. The second part is concerned with the issue of selective authentication of digital images. A semi-fragile watermarking scheme using the pinned sine transform (PST) is proposed. The watermarking system can localize the portions of a watermarked image that have been tampered maliciously with high accuracy. While the watermarking scheme is sensitive to any texture alteration in the watermarked image, it can survive slight legitimate operations, such as JPEG compression, which is desirable in some practical applications, such as traffic enforcement, satellite remote sensing and legal applications.

Automatic image content authentication is also investigated in this thesis. The third part presents several schemes for the self-restoration of tampered images detected by our proposed fragile and semi-fragile watermarking systems, incorporating prior knowledge embedded in the watermarked image. Firstly, we describe how the self-embedding strategy can be incorporated into our proposed semi-fragile watermarking and point out its limitations. We then propose another scheme to address these problems. The restoration of tampered images with lost blocks is formulated as an irregular sampling problem in approximation transform subspaces. The tampered blocks are then reconstructed, making use of the information embedded in the same watermarked image, through iterative projections onto convex sets in approximation subspaces. The primary advantage of this restoration process compared with traditional self-embedding methods is its increased robustness

ABSTRACT

ix

against unintentional attacks.

Finally, a robust watermarking method in the slant transform domain is proposed. Its applications include the copyright protection and owner identification of digital images. In the context of transform coding, the slant transform is considered to be a sub-optimum orthogonal transform for energy compaction. For digital watermarking, however, the energy spread becomes a significant advantage, as there is now a good spread of middle to higher frequencies with significant energies for robust information hiding. The robustness of our slant transform watermarking against various Stirmark and Checkmark attacks is illustrated and an analytical comparative study on the performance of the slant transform against other conventional transforms such as cosine transform and Walsh-Hadamard transform is also presented.

Contents

Acknowledgements	v
Abstract	vii
List of Figures	xiv
List of Tables	xvii
1 Introduction	1
1.1 Models of digital image watermarking	2
1.2 Content authentication	5
1.2.1 Exact authentication	6
1.2.2 Selective authentication	9
1.2.3 Localization and VQ attacks	13
1.2.4 Restoration	15
1.3 Robust watermarking	18
1.3.1 Spatial domain methods	18
1.3.2 Transform domain methods	21
1.3.3 Benchmarking software	24
1.4 Thesis outline	26
2 Exact authentication by zero encoding	28
2.1 Zeros of z -transform	29
2.2 The proposed fragile watermarking method by zero encoding	33
2.2.1 Watermark embedding	34
2.2.2 Image authentication	38
2.3 Performance analysis	39
2.3.1 Fragility of watermark to pixel perturbation	39
2.3.2 The setting of watermarking strength ϵ	40
2.3.3 Security	41

2.4	Experimental results	44
2.5	Summary	46
3	Selective authentication using pinned sine transform	47
3.1	The pinned sine transform	48
3.1.1	The pinned Karhunen-Loeve transform (PKLT)	49
3.1.2	The approximate pinning function and PST	52
3.2	The proposed semi-fragile watermarking using PST	54
3.2.1	Watermark embedding	54
3.2.2	Watermark detection and image authentication	59
3.3	Performance analysis	61
3.3.1	Dual-field decomposition and selective content authentication	62
3.3.2	Inter-block relationship and counterfeiting attacks	63
3.4	Simulation results	64
3.5	Summary	72
4	Self-restoration with watermarking	74
4.1	Self-embedding	75
4.1.1	The proposed method	76
4.1.2	Experiments and analysis	78
4.2	Blind restoration method based on irregular sampling	78
4.2.1	Irregular sampling and reconstruction by projections onto convex sets (POCS)	80
4.2.2	Watermark embedding	86
4.2.3	Watermark detection and image authentication	89
4.2.4	Content restoration	90
4.2.5	Minimization of the probability of false alarm	91
4.3	Performance analysis of the proposed blind restoration method	92
4.3.1	Probability of false alarm	93
4.3.2	Evaluation of restoration capability	94
4.3.3	Computational complexity	95
4.4	Simulation results of the proposed blind restoration method	97
4.4.1	The quality of watermarked images	98
4.4.2	Authentication and Restoration	99
4.5	Summary	105
5	Robust watermarking in slant transform domain	106
5.1	The slant transform	107
5.2	Watermarking in slant transform domain	112
5.2.1	Watermarking embedding	113
5.2.2	Watermarking detection	116
5.3	Perceptually adaptive watermarking	117

CONTENTS

xiii

5.4	Experiments and analysis	119
5.4.1	The robustness of the proposed scheme	120
5.4.2	The determination of threshold	123
5.4.3	Comparison of transforms	123
5.5	Summary	127
6	Summary of the thesis	128
6.1	Conclusions	128
6.2	Future works	131
Appendix A: Fast sine transform via fast fourier transform		132
Appendix B: Composing slant matrices from Hadamard matrices		134
Publications		139
Bibliography		142

List of Figures

1.1	Generic digital image watermarking scheme: (a) watermark embedding; (b) watermark detection (the part in the dashed line is optional).	3
2.1	Statistical distribution of the real zeros.	33
2.2	Fragile watermarking by zero encoding: (a) watermark embedding; (b) image authentication.	34
2.3	An example of watermarking by zero encoding: (a) an 8×8 block; (b) the zeros of the z -transform of $\mathbf{x}_0$; (c) watermarking of $\mathbf{x}_0$; (c) watermarking of $\mathbf{x}_7$	37
2.4	Offsets of the modulus of negative real zeros caused by one gray-scale value change (change one pixel value into a random new value).	40
2.5	The sensitivity of the authentication with increasing ϵ	41
2.6	The visual quality of the watermarked images with increasing ϵ	42
2.7	The probability of false alarm with increasing ϵ	42
2.8	Experimental results: (a) the original image; (b) the watermarked image. . .	44
2.9	Tampered images and authentication results: (a) the background was intentionally blurred; (b) the table was removed; (c) the positions of the man and the woman in the image were exchanged; (d) the authentication results of (a); (e) the authentication results of (b); (f) the authentication results of (c). 45	
3.1	A rectangle field of size $n \times m$	50
3.2	The PST for a typical block. Each block has one new corner and two new boundaries.	52
3.3	Semi-fragile watermark embedding process for one block.	55
3.4	The dual-field decomposition in PST for a typical block.	55
3.5	Semi-fragile watermark detection and image authentication processes for one block.	60
3.6	The dual-field decomposition in the pinned sine transform of the <i>Dubai</i> image: (a) the original image, (b) the boundary field and (c) the pinned field. 62	
3.7	The inter-block relationship in the pinned sine transform.	63

LIST OF FIGURES

xv

3.8	Simulation results: (a) original <i>Couple</i> , (b) original <i>Tank</i> , (c) original <i>Pyramids</i> and (d)-(f) the watermarked images (PSNR: (d) 33 dB; (e) 36 dB; (f) 35 dB).	65
3.9	The tradeoff of hiding rate and image quality.	66
3.10	Tampered images and authentication results: (a) the table was removed; (b) the tank was shifted; (c) the texture in some part was modified; (d) authentication result of (a); (e) authentication result of (b); (f) authentication result of (c). . .	67
3.11	The distribution of the watermark detection outputs: before and after JPEG compression (QF=40).	68
3.12	Comparisons between PST watermarking and conventional DCT watermarking: the probability of detection after (a) JPEG compression and (b) wavelet compression.	70
3.13	Sample authentication results after JPEG compression and additive noise: (a) watermarked and modified <i>Couple</i> image after JPEG compression (QF=90); (b) watermarked and modified <i>Pyramids</i> image with additive noise ($\sigma^2 = 5$); (c) authentication result of (a); (d) authentication result of (b).	71
4.1	Watermark embedding process for one block with self-embedding.	76
4.2	Watermark detection and authentication processes for one block and restoration with self-embedding.	76
4.3	The insertion of the recovery bits.	77
4.4	Reconstruction with LSB self-embedding: (a) original image; (b) self-embedded image; (c) reconstructed image, PSNR*=21.04dB (*compared to the original image).	79
4.5	Restoration results using self-embedding: (a) restoration of Figure 3.10 (a), PSNR*=22.45dB; (b) restoration of Figure 3.10 (b), PSNR=21.67dB; (c) restoration of Figure 3.10 (c), PSNR=22.31dB (*averaged PSNR of the restoration of tampered blocks compared to untampered blocks).	79
4.6	Projections onto convex sets.	82
4.7	Blind restoration based on irregular sampling: Watermark embedding process for one sub-block.	88
4.8	Blind restoration based on irregular sampling: Watermark detection, image authentication and restoration for one sub-block.	89
4.9	Restoration performance of Algorithm 4.1.	95
4.10	Different locations of tampered sub-blocks within a macro-block with $M = 3$	96
4.11	The testing images: (a) <i>Traffic</i> ; (b) <i>Chart</i>	97
4.12	The watermarked images: (a) <i>Traffic</i> ; (b) <i>Chart</i>	98
4.13	Simulation results: (a) the tampered images; (b) authentication results. . . .	99
4.14	Simulation results: (a) restored images by the proposed method; (b) restored images by Fridrich method 1 (1 LSB); (c) restored images by Fridrich method 1 (2 LSBs); (d) restored images by Fridrich method 2.	100

4.15	Simulation results: (a) the tampered images with acceptable manipulations; (b) authentication results.	102
4.16	Simulation results: (a) restored images by the proposed method; (b) restored images by Fridrich method 1 (1 LSB); (c) restored images by Fridrich method 1 (2 LSBs); (d) restored images by Fridrich method 2.	103
5.1	Original images: (a) <i>Lena</i> ; (b) <i>Singapore</i>	109
5.2	Slant transform domain logarithm of magnitude display: (a) <i>Lena</i> ; (b) <i>Singapore</i>	109
5.3	Slant transform basis vectors of order-8.	111
5.4	Robust watermarking in slant transform domain: (a) watermark embedding; (b) watermark detection.	112
5.5	Embedding locations in one selected slant transform components block. . .	114
5.6	The watermark strength levels: (a) strength level graph for <i>Lena</i> ; (b) strength level graph for <i>Singapore</i>	119
5.7	The watermarked images: (a) <i>Lena</i> ; (b) <i>Singapore</i>	120
5.8	The distribution of the values of the correlation coefficient when detected from non-watermarked images.	124
5.9	Performance comparisons between slant transform, Hadamard transform and discrete cosine transform watermarking against JPEG (DCT) compression.	125
5.10	Performance comparisons between slant transform, Hadamard transform and discrete cosine transform watermarking against JPEG-2000 (wavelet) compression.	125

List of Tables

3.1	Comparison of the probability of false alarm.	72
4.1	Comparison of the PSNRs (dB) of the watermarked images.	98
4.2	Comparison of the PSNRs (dB) of the restored portions in Figure 4.14. . . .	101
4.3	Probability of false alarm of the authentication process.	101
4.4	Comparison of the PSNRs (dB) of the restored portions in Figure 4.16. . . .	104
4.5	Robustness of the self-restoration (*compared with the original unwater- marked image).	104
5.1	Sample benchmarking results for <i>Lena</i> image.	121
5.2	Sample benchmarking results for <i>Singapore</i> image.	122

Chapter 1

Introduction

Digital multimedia technology paves the way for new applications, features, and services. The acquisition, exchange and transmission of digital images are now simple tasks with the overwhelming development of the Internet and the wide availability of high capacity digital recording devices. However, such increasing accessibility also introduces a new set of challenging problems associated with the security, authentication and illegal distribution of digital images. One potential solution for the above problem is *digital watermarking*. The term “digital watermarking” was first used by Komatsu and Tominaga [1] in 1990; still, the technique of digital watermarking did not attract much interest until the mid 1990s. Since then, a great deal of research has been dedicated to digital watermarking and this field has evolved very quickly. An extensive tutorial of digital watermarking can be found in [2, 3].

1.1 Models of digital image watermarking

Digital image watermarking involves the embedding and retrieval of imperceptible data within some host images. The hidden data is called a watermark, which typically contains information about the origin, status, or recipient of the host images. There are three main considerations in the design of a typical image watermarking system.

1. Design of the watermark signal W to be added to the host image. Typically, the watermark signal depends on a key $\mathcal{K}$ and watermark information D given as follows:

$$W = f_0(D, \mathcal{K}). \quad (1.1)$$

It may also depend on the host image $\mathbf{X}$ into which it is embedded, as follows:

$$W = f_0(D, \mathcal{K}, \mathbf{X}). \quad (1.2)$$

2. Design of the embedding method itself that incorporates the watermark signal W into the host image $\mathbf{X}$ yielding a watermarked image $\mathbf{Y}$, as follows:

$$\mathbf{Y} = f_1(\mathbf{X}, W). \quad (1.3)$$

In many cases, the key $\mathcal{K}$ is also involved in the embedding process given by

$$\mathbf{Y} = f_1(\mathbf{X}, W, \mathcal{K}). \quad (1.4)$$

3. Design of the corresponding detection method that recovers the watermark information from the watermarked and possibly attacked image $\hat{\mathbf{Y}}$ using the key and with

Introduction

3

Figure 1.1: Generic digital image watermarking scheme: (a) watermark embedding; (b) watermark detection (the part in the dashed line is optional).

help of the original image (*non-blind detection*) given as follows:

$$\hat{D} = f_2(\mathbf{X}, \hat{\mathbf{Y}}, \mathcal{K}). \quad (1.5)$$

or without the original image (*blind detection*) as follows:

$$\hat{D} = f_2(\hat{\mathbf{Y}}, \mathcal{K}). \quad (1.6)$$

The design considerations for watermark embedding and detection are illustrated in Figures 1.1(a) and 1.1(b), respectively. Figure 1.1(a) shows the generic watermarking scheme for

the embedding process. The input to the scheme is the watermark, the host image, and an optional public or secret key. The output is the watermarked image. The generic watermark detection process is depicted in Figure 1.1(b). Inputs to the scheme are the test image, the secret or public key, and, depending on the detection method, the original image and / or the original watermark. The output of the watermark detection process is either the recovered watermark or some kind of confidence measure indicating how likely it is for the given watermark at the input to be present in the image under inspection.

In terms of their functionality, the existing watermarking schemes can be classified into two main groups: *fragile watermarking* or *semi-fragile watermarking* for content authentication and *robust watermarking* for copyright protection. For both types of watermarking scheme, the following important properties are often used as performance criteria:

- *Imperceptibility*: The watermark should not destroy the visual quality of the host data;
- *Fragility / Robustness*: The ability of the watermark to reflect unauthorized operations on content in fragile watermarking or semi-fragile watermarking; or the ability of the watermark to survive normal processing of content in robust watermarking.
- *Security*: The ability of the watermark to resist hostile attacks, which are unauthorized processes specifically intended to destroy the watermark.
- *Probability of false alarm*: The frequency with which we should expect the tampered portions of watermarked content to be falsely reported in fragile or semi-fragile wa-

termarking; or the frequency with which the watermarks are falsely detected in un-watermarked content in robust watermarking.

- *Cost*: The computational complexity and cost of the embedder and detector.

In the remainder of this chapter, we examine the properties of the two groups of watermarking schemes in more details. A review of related previous works is also presented.

1.2 Content authentication

While digital media offer many distinct advantages over their analog counterparts, the ease with which they can be edited and tampered makes the protection of their integrity and authenticity a serious and important issue. In certain practical applications, such as remote sensing, legal defending, news reporting and medical archiving, there is a need for verification or authentication of the integrity of the media content. In this section, we focus on how a watermark can be used to assist in maintaining and verifying the integrity of digital images. This section also focuses on answering the following questions:

- Has the image been tampered in any way whatsoever?
- Has the semantic content of the image been altered?
- What parts of the image have been tampered?
- Can a tampered image be restored?

The remainder of this section discusses our proposed solutions to the above four challenges and review of various primary methods that have been proposed in the literature is also presented. Sections 1.2.1 and 1.2.2 give a brief introduction to exact and selective content authentication. Section 1.2.3 investigates the localization problem and Section 1.2.4 introduces the restoration methods.

1.2.1 Exact authentication

The most basic authentication task is to verify that an image has not been altered *at all*. An image is classified as inauthentic if even a single bit has been changed. Such bit-sensitive literal authentication is desirable for the storage of master copies of important images.

It is worth mentioning that besides watermarking, there are many other methods existing for solving this problem. For example, one common approach is the creation of a *digital signature*. Friedman [4, 5] proposed to apply the digital signature technology to digital cameras. He suggested creating a “trustworthy camera” in which a standard cryptographic signature was computed for each captured image using hash functions. The signature was then stored and transmitted along with the image. The verification program used the accompanying digital signature file to judge if the input digital image file in question was identical with the original one. Such digital signatures are able to detect alterations in signed data, but cannot localize those alterations. Furthermore, these signatures must be transmitted along with the images. It is thus easy to lose the signatures during transmission and normal usage of the images.

A *fragile watermark* is a message that can be invisibly embedded into an image and can be extracted to identify the authenticity of this image later on. There is no requirement in a fragile watermarking system to store any overhead data, as required in cryptographic signatures. And while it may be possible to just embed the cryptographic signatures into the images, fragile watermarks is still more advantageous in that an embedded watermark undergoes the same transformation as the image, and when the image is corrupted, it is possible to localize the tampering.

A variety of fragile watermarking methods have been proposed including some schemes for specific applications [6, 7, 8]. Schyndel *et al.* [9] discussed the embedding of imperceptible digital signature into an image through the manipulation of the LSBs. Walton [10] described techniques to embed long strings of binary bits into a set of randomly generated addresses where the LSBs of pixels were changed to match the corresponding bits in the string. The watermarking techniques in which the information is embedded into LSBs of the image pixel values are unlikely to produce visual artifacts in the image. If an image is altered, the LSBs are likely to be changed such that the verification will be able to determine the modification. Such LSBs manipulation, however, is not secure against malicious attacks: it is relatively easy to modify the visual content of an image while the LSBs are kept intact. Authentication based on the LSBs then fails to detect the tampering.

Yeung and Mintzer [11] proposed a pixel-based authentication watermark to protect the integrity of images. A logo image, which is of the same size as the source image, was used as the watermark. The binary map of the watermark image was embedded into every

pixel of the source image, using a secret key generated from pseudo-random sequence, so that any change in pixel values to the image can be detected. A modified error diffusion method was adopted to diffuse the errors introduced in processing locally to avoid visible quality degradation in watermarked images.

Wong proposed a blockwise fragile authentication watermarking [12] and improved it by using a public-key based scheme [13, 14]. Wong proposed dividing an image into blocks and independently signing each block. The signature was then embedded in LSBs of every pixel in the image. The signature was generated as follows. Firstly, a binary bit string was generated by inputting the block data into a cryptographic hash function. This binary bit string was then combined with the binary watermark signal using an exclusive or function, whose output was further encrypted with a public key cryptographic system to give the signature for each block.

In the above algorithms, the security problems in traditionally LSBs based methods were addressed through cryptographic hash function, which linked the signature to the more significant bits of the pixels. However, their blockwise independence was exploited by Holliman and Memon [15] with a Vector Quantization (VQ) counterfeiting attack¹, which is a kind of “cutting and pasting” attacks. In [16], the authentication watermarks are embedded in a wavelet-like hierarchical manner such that the watermarking scheme can be resistant to vector quantization (VQ) attacks while sacrificing localization ability.

In this thesis, a new fragile watermarking method for digital image authentication is

¹Refer to Section 1.2.3 for details of VQ attacks.

proposed based on the zero locations of the z -transform [17]. The z -transform domain is a new transform space for fragile watermark embedding. Our watermarking method is designed by exploiting the sensitivity of the positions of the zeros of the z -transform around the unit circle to any change made on the host image. The blockwise watermarking system can localize the portions of a watermarked image that have been tampered with high accuracy. On the other hand, by incorporating the phase information of the complex zeros, it would be more difficult for an attacker to find a fake block whose underlying watermark remains the same as the targeted block, which provides more security against the VQ attacks.

1.2.2 Selective authentication

Exact authentication has many potential applications, especially for the content protection of substantial documents. However, under most circumstances, the meaning of multimedia data is generally based on their semantic content rather than the bit streams. A change of a few bits rarely makes an important difference to an image. In fact, lossy compression techniques, such as JPEG, change many bits to achieve a smaller file size while the perceptual quality of the image is not significantly affected. Therefore, a *semi-fragile watermarking* which performs selective authentication is more desirable in many applications. A semi-fragile watermarking is designed to be unaffected by legitimate manipulations such as some inherent system processing, but less robust against by illegitimated distortions.

Lin and Chang [18, 19] proposed a method that could localize malicious tampering

to the image content while accepting JPEG compression to a pre-determined quality factor QF_0 . Their method achieved its goal by using an invariant relationship between two DCT coefficients in a block pair before and after JPEG compressions. Such relationship was encoded and inserted into the LSBs of rounded DCT coefficients. Their method proved to be robust to JPEG compression by both mathematical deduction and experimental results. In addition, assume that the watermarked image went through JPEG compression with a quality factor larger than or equal to QF_0 , zero probability of false alarm can be assured.

Another semi-fragile watermarking scheme which jointly considered watermarking and compression was described by Xie and Arce [20, 21, 22], in which an edge-based message digest of the source image was used as the signature. The signature was in the form of binary data and the most significant wavelet decomposition coefficients in the SPIHT² [23] compression algorithm were modified according to this binary sequence. Let C_0 denote the compression ratio applied in the SPIHT algorithm during the signature embedding, experimental results showed that the signature was able to tolerate SPIHT compression with compression ratio lower than or equal to C_0 .

Lu and Liao [24, 25] proposed a multipurpose watermarking scheme for both copyright protection and content authentication. Two watermarks were embedded independently by quantizing the selected wavelet coefficients into masking threshold units. For the purpose of image authentication, the approach was supposed to tolerate some unintentional modifications. However, since the texture information of the source image was not

²Set Partitioning In Hierarchical Trees

explicitly considered in the framework of the scheme, the probability of false alarm caused by compression and other common image processing operations were relatively high and interactive processes may be needed to accurately localize the malicious tampering.

A telltale watermarking was described by Kundur and Hatzinakos [26]. A watermark was embedded in the discrete wavelet domain of the image by quantizing the corresponding coefficients, which allowed the localized detection of changes in the image in both spatial and frequency domain regions. A statistics-based tamper assessment function (TAF) was used to determine the extent of tampering. The presence of tampering was determined if TAF was larger than a certain threshold. TAF can be computed for specific frequencies and / or spatial regions to assess the type of tampering. For example, if JPEG compression is applied to an image, the method can assess that most of the changes have occurred to the details at the higher resolution levels. If a part of the image has been replaced or changed in addition to compression, the watermark in the lower resolutions will not remain the same. Hence, the lower resolution image can be authenticated. In addition, when filtering is applied to an image the technique can assess the frequency regions that have been most tampered with. However, their tampering detection results were very unstable. Perturbation of a wavelet coefficient to the left or to the right by a certain quantity would make the extracted mark different from the embedded one.

It is also worth mentioning that most of the recently proposed semi-fragile watermarking schemes were based on the wavelet domain [20, 21, 22, 24, 25, 26]. The localization ability in both spatial domain and frequency domain makes the wavelets a potential can-

didate for semi-fragile watermarking. However, to authenticate content, some significant features, for example, the edges of the host image are required to be encoded and embedded in the low frequencies of the wavelet decomposition. Thus, there exists a tradeoff between the visual quality of the watermarked image and the ability of the scheme to detect changes. Another drawback of these schemes is the high computation cost during the feature extraction and visual hash coding processes.

Although a variety of semi-fragile watermarking schemes have been proposed in the literature, the issue of “selective content authentication” has not been vigorously addressed. In this thesis, a novel semi-fragile watermarking scheme using the pinned sine transform (PST) [27, 28] is proposed. By exploiting the special property of PST, the proposed watermarking scheme can tolerate a wide range of common image processing operations, while it can accurately detect illegitimate operations on the content of the images. The motivation for developing a semi-fragile watermarking based on PST is due to the observation that this transform could provide an effective way to solve both the above-mentioned selective content authentication problem and the issue of exposing the “cutting and pasting” counterfeiting attacks which will be detailed in the next section. The observation is as follows. The PST conducts a decomposition of the original image into two mutually uncorrelated fields, namely, the boundary field and the pinned field. The texture information of the original image is contained in the pinned field, wherein the sine transform is equivalent to a fast Karhunen-Loeve transform (KLT). By exploiting this important property, we propose to embed a watermark signal into the sine transform domain of the pinned field for content

authentication. The proposed watermarking scheme is especially sensitive to texture alterations of the host image while permitting controlled amount of modifications to non-texture aspects of the host image.

1.2.3 Localization and VQ attacks

Localization refers to the capability of watermarking schemes to identify regions of the image that have been corrupted, while verifying that the remainder of the image has not been changed.

Most localized authentication methods rely on block-wise authentication, in which the image is divided into a number of disjoint regions, each of which is authenticated separately. If a part of the image is modified, then only the affected region would fail to authenticate. However, such block-wise schemes cannot resist the “cutting and pasting” attacks. The well-known vector quantization (VQ) counterfeiting attack proposed by Holiman and Memon [15] is one such attack, which thwarts many existing block-wise fragile or semi-fragile watermarking methods.

The success of the VQ attack is based on the assumption that the attacker has a partial knowledge of the possible watermark patterns. The attack starts by collecting a large number of watermarked images, and constructing the codebooks by categorizing all the blocks in those images so that the blocks in the same class correspond to the same watermark pattern. Suppose that the attacker has an unmarked image Z and intends to counterfeit from it an approximate image Z' which can pass the authentication system. He examines every

block of $\mathbf{Z}$, say, $\mathbf{Z}_{p,q}$ and identifies it as a member of a certain class according to the specific watermarking technique. He then replaces $\mathbf{Z}_{p,q}$ with a watermarked block in that class that minimizes the difference between this block and $\mathbf{Z}_{p,q}$. As such the attacker achieves his goal without being detected by the authentication system.

Some inter-relationship between the watermarked blocks was introduced to avoid the VQ attacks such as described in [13], however, a close relationship between uncorrelated blocks may come at the cost of reduced error localization properties and introduce confusion for the consequent authentication process. Another method which thwarts the VQ attack while maintaining localization properties was proposed by Celik *et al.* [16]. In [16], the authors proposed dividing the image into blocks in a multilevel hierarchy and calculating block signatures in this hierarchy. While signatures of small blocks on the lowest level of the hierarchy ensure tamper localization, higher level block signatures provided increasing resistance to VQ attacks. The tamper localization ability of this algorithm was bounded by the block size on the lowest level of the hierarchy which was determined by the length of the signature used. Therefore, there is a tradeoff between the localization ability and its robustness to VQ attacks.

In our semi-fragile watermarking scheme described later in this thesis, the inter-block relationship introduced in the process of PST renders the watermarking scheme intrinsically resistant to content cutting and pasting attacks, while maintaining the localization ability.

1.2.4 Restoration

When tampering was detected in an image, in certain practical applications, such as legal tenders and traffic enforcement, it is desirable for some initial estimated reconstruction of the tampered parts of the image content that may be useful for submission as court evidence.

The possibility of automatic reconstruction of tampered areas have been investigated recently in literature [29, 18, 30, 31]. One of the most popular research strategies is the self-embedding method, in which a highly compressed version of the original image is embedded into the image itself. Thus, if portions of the watermarked image are removed or destroyed, these modified regions can be replaced with their corresponding low-resolution versions. For example, In [29], Fridrich and Goljan proposed two self-embedding schemes for the self-restoration of digital images with watermarking authentication.

The first scheme began with dividing the image into 8×8 blocks and transforming each block using DCT. The primary DCT coefficients of every block were then quantized, using a quantization matrix corresponding to that of 50% quality JPEG compression. The resulted bit-string for every block, was carefully controlled so that it was exactly 64-bit long. These bits were then embedded into the least significant bit (LSB) of another block. When the watermarked content was detected to be unauthentic, the recovery bits were extracted out to establish a low-resolution reconstruction of the modified parts. However, the quality of the recovery was not satisfactory and may not be sufficient for illustrating fine details. To improve the quality of reconstruction, the authors proposed extending the recovery bits to be 128 bits for every block, and using 2 LSBs for embedding. However,

this introduced more degradation to the watermarked images. Moreover, the spatial domain based method was easily attacked. The restoration would fail if the attacker, after modifies the content, further compresses the image file or just randomizes the least significant bits.

The second scheme encoded the information of the original image into the watermarked image using differential encoding. First, a low color depth image t was generated by decreasing the gray levels of the original image g to the interval $[-8,8]$. A cyclic shift on t was then performed by an integer vector $s = (s_1, s_2)$:

$$st_{ij} = t_{(i-s_1) \bmod M, (j-s_2) \bmod N}.$$

The watermark embedding process started from the upper left corner and proceeded by rows from left to right, top to down. The watermarked image g' was defined as

$$g'_{11} = g_{11},$$

$$\bmod(g'_{ij+1} - g'_{ij}) = st_{i,j}.$$

In the reconstruction stage, a color truncated approximation to the original image can be obtained by calculating the difference between pixels. This improved scheme can tolerate the least amount of image processing operations, such as JPEG compression with quality factor above 85% and additive random noise in the range $[-2,2]$.

An alternative method for approximate correction of tampering is proposed by Kundur and Hatzinakos in [30]. The authors presented a telltale watermarking which was capable of self-restoration when tampered areas were detected. The watermark was embedded into the wavelet domain of an image. The watermark underwent the same distortions as the cover

images so that the manipulation to the image can be approximately determined through the analysis of the extracted watermark. Under the assumption that the tampering function was a low-pass filter which preserved the local mean values, and further assume that the watermark signal must undergo the same tampering with the carrier image, the image was restored by undoing the tampering. However, the restoration capability of this method is insufficient since the strict assumptions are not always true in practical applications. For the above reason, the method can only restore invertible distortions, and therefore is not useful with attacks such as removal attacks.

In this thesis, we propose a novel self-restoration method after watermark authentication. The restoration method is based on the theory of irregular sampling. Contrast to previous methods, the restoration process is robust to common image processing operations such as lossy transcoding and image filtering. Semi-fragile Watermarks are embedded into the original image, which reflects local malicious tampering on the image. When tampered blocks are detected, the restoration problem is formulated as an irregular sampling problem in approximation subspaces. These blocks are then reconstructed, making use of the information embedded in the same watermarked image, through iterative projections onto convex sets in approximation subspaces. Simulation results showed that the scheme keeps the probability of false alarm to a minimum while maintaining the data integrity of the restored image.

1.3 Robust watermarking

In contrast to fragile or semi-fragile watermarking, *robust watermarking* is designed to survive legitimate and everyday usage of content. Proposed or actual applications of robust watermarking include copyright protection, owner identification, broadcast monitoring, transaction tracking and device control. In designing a robust watermarking scheme it is important to identify the specific processes that are likely to occur between embedding and detection. Examples of processes that a watermark might need to survive include lossy compression, digital-to-analog-to-digital conversion, image filtering, noise reduction, and format conversion. Since many applications require watermarks to be detected in content that may have been altered after embedding, robust watermarking is gaining a significant amount of attention, hence much research effort has been invested in it. According to the embedding domain of the original image, the robust watermarking techniques proposed can be divided into two main groups [32]: one is the spatial domain approaches and the other is the transform domain approaches.

1.3.1 Spatial domain methods

Most of the earliest watermarking techniques embedded the watermarks into the spatial domain of the source image. Tirkel *et al.* [33, 34] proposed two spatial domain watermarking methods for grayscale images. In their first approach, the watermark was an m-sequence-derived PN code embedded in the LSBs of the image. To gain full access to the LSBs data without introducing much distortion, the image was first compressed to 7 bits

through adaptive histogram manipulation. In their second approach, the watermark was an m-sequence-derived code added to the LSBs data. The decoding process made use of the unique and optimal 20 autocorrelation functions of m-sequence.

Caronni [35] described a system to track unauthorized image distribution. He proposed a method to mark images using spatial signal modulation which was called the tagging process. A tag is a square of size $N \times N$ and all possible locations in an image where a tag could possibly be placed were identified by calculating the local region variance. A selected image area was tagged by adding or subtracting the tag with a random, zero mean noise pattern. To recover an embedded bit, the difference between the original and the tagged image was computed. The mean of an estimated tagged location was then compared to the neighboring mean to determine the embedded value.

A patchwork algorithm was proposed by Bender *et al.* [36] which involved a secret key to initialize a PN number generator that defined the locations of the cover hosting the watermark. In the insertion process, n pixels pairs were selected pseudo-randomly according to a secret key. The image was watermarked by adding 1 to the first pixel of every pair and subtracting 1 from another pixel of every pair. In the extraction stage, the presence of the watermark was judged by calculating difference of the pixels in every pair and summing up the difference values. Assume that the pixels pairs, which were randomly chosen from an image, were independent and identically distributed, a watermarked image should expect the above sum to be $2n$, otherwise, it should be approximately zero. Some extensions of this algorithm have been proposed [37, 38] to hide a message longer than

one bit and to increase the robustness of the scheme, against basic data manipulations such as the removal or translation of one pixel which would normally have destroyed the watermark.

Maes *et al.* [39] proposed a spatial watermarking approach by modifying the geometric features of an image. The method used a dense line pattern as the watermark which was generated pseudo-randomly. A set of salient points in the image was then calculated based on an edge detection filter. The detected points were warped such that a significantly large number of points were within the vicinity of the lines. In the detection process, the presence of watermark was verified if a significantly large number of points were within the vicinity of lines.

Bas *et al.* [40] proposed a new spatial approach for digital images providing more robustness to geometrical distortions. The basic idea was to bind the signature with the content of the image. Feature points detected were used to represent an image content descriptor. The set of feature points permitted image partitioning using a Delaunay tessellation. The watermark of random sequence was embedded in each triangle tessellation using classical additive scheme. The detection of the watermark was based on the same content analysis and performed using a correlation criterion.

Spatial domain algorithms are effective transparent embedding techniques and computationally inexpensive to implement. Moreover, these algorithms have significant advantages in localizing the common geometric attacks, such as rotation, cropping and translation. For example, to design a watermark that is robust against cropping, it would be

better to embed the watermark in the spatial domain which completely localizes the manipulation. When a portion of the watermark was cropped, the repetitions in the remaining signal are still accessible. However, since the energies of the watermark signal is randomly spread across all frequency bands in the spatial domain methods, attacks in the frequency domain, for example, filtering and lossy compression, may easily destroy the integrity of the embedded data, hence remove the watermark information.

1.3.2 Transform domain methods

Various methods operating in different transform domains have been presented in the literatures. Embedding in the transform domain has significant advantages in analysis of visibility and robustness to transform domain attacks. The transform domains commonly used are discrete cosine transform (DCT), discrete Fourier transform (DFT) and discrete wavelet transform (DWT).

The most widely used are discrete cosine transform (DCT) based methods. Since DCT was widely applied by the source coding community for JPEG and MPEG implementation [41, 42, 43], data embedding performed in the DCT domain is often considered more robust to attacks from JPEG and MPEG compression. The most well know DCT domain method was proposed by Cox *et al.* [44], who drew parallels between watermarking technique and spread spectrum communication and suggested that the watermark should spread over a set of visually important frequency components. The watermark which was a Gaussian distributed sequence was then embedded into the perceptually most significant

frequency components of the DCT domain of original images. Robustness tests showed that their method resisted JPEG compression down to a quality factor of 5% and was also robust against dithering, printing-scanning and collusion attacks. Cox [45, 46, 47] *et al.* has also introduced the model of informed watermarking by comparing the watermarking to communications with side information at the transmitter and / or detector.

DFT has been widely applied in signal and image processing. This transform was adopted in the field of watermarking based on its invariant properties to some geometric distortions. Ruanaidh *et al.* [48] introduced the use of the Fourier-Mellin transform for watermarking. The method was a hybrid approach between DFT and log-polar mapping. The well known property that the DFT magnitude is a circular translation invariant leads to the translation invariance of the watermarking. Invariance to image rotation and scaling was achieved by mapping the amplitude from cartesian coordinates to polar coordinates. Combining these two steps resulted in a rotation, scaling and translation (RST) transformation invariant.

The discrete wavelet transform plays an important role in signal and system analysis and is a major function in the source compression standard JPEG 2000 [49]. In recent publications, this transform has been extensively applied to image watermarking [50, 51, 52, 53, 54, 55, 56, 57]. The positive arguments of using wavelets for watermarking closely resemble those for advocating DCT for JPEG. The multi-resolution aspect of wavelets is a significant advantage and helpful in managing a good spread of the hidden message in the cover data in terms of robustness versus visibility. Hsieh *et al.* [50] proposed a multi-energy

watermarking scheme based on the qualified significant wavelet tree (QSWT). Wang *et al.* [51] selected a middle-frequency band of wavelet domain to insert a binary image watermark. Kundur *et al.* described a watermarking method using wavelet-based fusion [52]. It involved adding the wavelet coefficients of the watermark to those of the image at different resolution levels. Prior to being added, the wavelet coefficients of the watermark were modulated using a human visual model constraint based on a measure called saliency [53]. Another watermarking method in the wavelet domain known as a robust reference watermarking technique was also proposed by Kundur *et al.* [54]. They used a novel communication tool sets to improve the performance of robust watermarking systems. In particular, they related the effects of attacks on the watermark to signal interference in a fading environment and employed diversity transmission and channel estimation principles to improve performance. The watermark embedding was based on the wavelet domain decomposition coefficients quantization technique. A non-stationary parallel binary symmetric channel (BSC) model of the watermark channel was introduced to characterize signal tampering more accurately and, hence, to extract the watermark. Wang *et al.* proposed a wavelet-based watermarking algorithm which was capable of private control of the watermark [55]. The watermark used in this scheme was a binary image of size 16×16 . The watermark image was scrambled and converted into a 1-D vector. It was then inserted into the wavelet decomposition domain using randomly generated orthonormal filter banks, which formed the major part of the private key. Detection was performed using a standard correlation detector. The proposed algorithm also achieved resilience to counterfeit attacks. Barni *et*

al. [56] and Kutter *et al.* [57] developed two novel visual models to improve the perceptual quality of images watermarked in the wavelet domain.

Besides the above mentioned commonly used transform domains, other unconventional transforms have also been investigated for watermarking. For example, these include Fractals [58], Radon transform [59], singular value decomposition (SVD) [60] and Hadamard transform [61, 62, 63, 64].

Watermarking schemes performed in the frequency domain normally provide better invisibility and security than the spatial domain algorithms. In terms of robustness, transform domain algorithms provide better resistance to the filtering and compression attacks. However, there are also some concerns associated with the existing transform domain algorithms. The major problem with many of these watermarking schemes is that they are not very robust against different types of image manipulations or attacks. Moreover, the transform domain algorithms have a higher complexity than the spatial domain algorithms, with some of these techniques too complicated to implement in real-time. To address the above problems, an efficient and robust watermarking scheme using the slant transform is proposed in this thesis.

1.3.3 Benchmarking software

Similar to conditional access and copy prevention mechanisms, the existence of watermarking technology and its potential possibilities have incited researchers to come up with attempts to defeat watermarking. Some publicly available tools to test the robustness of

image watermarking techniques are discussed in this section.

Stirmark benchmark 4.0 [65, 66, 67] is a software tool to test the robustness of image watermarking techniques. It consists of 14 different types of image manipulations or attacks which include signal enhancement, compression, additive noise, image filtering, scaling, cropping, shearing, rotation, linear transformations and random geometric distortions. Among these attacks, the most powerful is the random geometric distortion. It applies minor geometric distortions to the watermarked images and introduces a practically unnoticeable quality loss in the image. But it causes almost all the current watermarking techniques to fail to detect the watermark as it destroys the synchronization. While the Stirmark benchmark is an excellent tool for measuring the robustness of watermarking algorithm, it is heavily weighted toward geometric transformations, and does not take into consideration of prior information about the watermark.

Checkmark benchmark 1.2 [68, 69] was developed to provide a better evaluation of the robustness of watermarking technologies to non-geometric attacks. Besides the attacks adopted in the Stirmark benchmark, Checkmark adds several new classes of attacks, which include wavelet compression (JPEG 2000), projection transformations and warping, copy attack, template removal attack, denoising and denoising followed by perceptual remodulation, non-linear line removal, collage attack, down / up sampling, dithering and thresholding. Most of the added attacks are non-geometric attacks that aim to remove the watermark directly or crack the entire watermarking system. There are 9 non-geometric attacks and 14 geometric attacks in Checkmark. In this thesis, both Stirmark and Checkmark benchmarks

are used in the robustness evaluation of the proposed robust watermarking scheme.

1.4 Thesis outline

The objective of this thesis is to provide a thorough discussion of digital image watermarking techniques subject to a wide range of applications. The applications involved in this thesis include exact content authentication, selective content authentication, automatic tampering correction, copyright protection and ownership identification. At the same time, we provide a broad coverage and linkage of transform techniques in signal processing, ranging from functional transform, orthogonal transform to irregular sampling in approximation subspaces. It is hoped through this coverage and linkage of these techniques would lead to novel watermarking algorithms that can yield better results than those at present available.

This thesis is divided into four parts. The first part concerns fragile watermarking for exact authentication. In Chapter 2, a fragile watermarking based on the encoding of the zeros of z -transform is described. The second part addresses the selective authentication problem: in Chapter 3, a semi-fragile watermarking scheme is introduced using the pinned sine transform (PST). The self-restoration method is investigated in the third part: in Chapter 4, we introduce self-restoration algorithms which incorporate self-embedding strategy and the irregular sampling model. The issue of copyright protection using watermarking is considered in the fourth part: a robust watermarking scheme is proposed in the slant transform domain, and comparison results with schemes from other orthogonal transform

Introduction

27

domains are also presented.

Chapter 2

Exact authentication by zero encoding

This chapter addresses the problem of exact authentication of image content. As have been mentioned in Section 1.2.1, in exact authentication, the image is classified as inauthentic even if a single bit has been changed, which is desirable for the storage of master copies of important images. A new fragile watermarking method is proposed based on the zero locations of the z -transform. Our watermarking method exploits the sensitivity of the positions of the zeros of the z -transform around the unit circle to any change made on the host image. The watermarking system can localize the portions of a watermarked image that have been tampered with high accuracy. The z -transform is a convenient yet invaluable tool for representing, analyzing and designing discrete-time signals and system [70, 71, 72, 73, 74]. However, to our knowledge, this is the first time this transform is applied for digital watermarking. It has the advantage of easy implementation and pixel-wise sensitivity to external tampering. Moreover the z -transform domain provides a better data hiding security pro-

tection than the normal LSBs check-sum fragile watermarking techniques. In Section 2.1 of this chapter, a review of the z -transform is provided, and the properties of its zeros are derived, based on which our new method is proposed. The proposed method is described in Section 2.2, followed by performance analysis and experimental results presented in Sections 2.3 and 2.4, respectively.

2.1 Zeros of z -transform

Signals are divided into *continuous signal* which are functions of a continuous variable (e.g., time), and *discrete signals* which are functions of a discrete variable. A discrete variable takes on only values which are integer multiples of a constant basic value. Thus, in fact, discrete signals are *sequences*. The general n th term of the sequence will be denoted by the symbol $f[n]$, $n = \dots, -2, -1, 0, 1, 2, \dots$. For the sequence we use the notation defined as follows:

$$\{f[n]\} = \{\dots, f[-2], f[-1], f[0], f[1], f[2], \dots\}.$$

Similarly as the operation of continuous systems is represented by differential equations, the operation of discrete systems is represented by difference equations, which can be solved with the aid of *functional transformations of sequences*. These transformations are useful tools for analysis of discrete signals and systems, which make possible an easy transition from the space of object functions represented mostly as functions of time or spatial coordinates to the space of transforms represented as functions of a complex variable or

eventually of frequency. The z -transform is the most widely used of the functional transformations of sequences in the theory of discrete signals and systems. The discrete signal $\{f[n]\}$ will mostly be assumed to be *one-sided*, i.e. $f[n] = 0$ for $n < 0$. In this thesis, we further assume that $\{f[n]\}$ is with compact support, which is the case for sequences of image pixels.

Definition 2.1 Consider a discrete signal $f[n]$, $n \in \mathbb{Z}$, with compact support $[0, N - 1]$, that is, $f[n] = 0$ for $\forall n \notin [0, N - 1]$. The z -transform is defined as follows:

$$\mathcal{Z}\{f[n]\} = F(z) = \sum_{n=0}^{N-1} f[n]z^{-n} \quad (2.1)$$

under the assumption that the series is convergent [75], where z is a complex variable.

The region where the z -transform converges is known as the region of convergence (ROC), and in this region the values of $F(z)$ are finite. For the image pixel value sequences, which are causal and with finite duration, the ROC is everywhere except at $z = 0$.

Moreover, there also exists a 2-D z -transform, which can be used in image processing applications as a useful generalization of the Fourier series [76]. However, the calculation of the zeros of 2-D z -transform is very complicated and number of the zeros of a 2-D sequence may be infinite [77]. In this thesis, the 1-D z -transform is used for fragile watermark embedding and authentication of image pixel sequences for the consideration of complexity.

Definition 2.2 The zeros of $F(z)$, denoted as $\{z_i\}$, $i = 1, \dots, N - 1$, are the points on which $F(z) = 0$.

The significance of the zeros is more apparent if we rewrite (2.1) in the form

$$F(z) = \frac{A}{z^{N-1}} \prod_{i=1}^{N-1} (z - z_i), \quad (2.2)$$

where A is a constant. Let us consider the polar form of the complex variable z :

$$z = |z|e^{j\theta},$$

where $|z|$ is the modulus and $\theta = \angle z$ is the phase angle of z . The phase angles of the zeros contain most of the information of the images [78]. To preserve the visual quality of the watermarked images, it is expected to keep the phases of the zeros unchanged. However, to achieve this, it is necessary to convert zeros from Cartesian plane to the polar plane, which is computationally expensive and the mapping usually involves rounding the numbers which introduce inaccuracy to the results. For the above reasons, we avoid manipulation on the phases of the complex zeros and only embed the watermark into real zeros. On the other hand, in most of cases, for a real sequence, there are a lot of complex zeros while there is only one real zero. Choosing complex zeros may result in difficulty in synchronization in the detection. In the following, we shall establish the sufficient conditions under which the existence of real zeros can be ensured, and it can also be proved that these real zeros must be negative.

Theorem 2.1 Suppose that $f[n]$, $n = 0, 1, \dots, N-1$ is a non-negative real sequence which is not uniformly zero. Then the z -transform of $f[n]$ must have negative real root(s) if $f[0] \neq 0$, $f[N-1] \neq 0$ and N is even.

Proof: By comparing Equations (2.1) and (2.2) we obtain

$$F(z) = \sum_{n=0}^{N-1} f[n]z^{-n} = \frac{A}{z^{N-1}} \prod_{i=1}^{N-1} (z - z_i). \quad (2.3)$$

From Equation (2.3) it can be concluded that $\{z_i\}$ must either be real or occur in conjugate pairs to ensure that the coefficients $f[n]$ are real. Meanwhile, since $f[N-1] \neq 0$, the z -transform series has even number of items. Therefore, the total number of roots should be odd. It follows that there is at least one real root. Let us consider z_{pr} , which is a root of $F(z)$ such that it is real and positive. Then we have

$$F(z) \big|_{z=z_{pr}} = \sum_{n=0}^{N-1} f[n](z_{pr})^{-n} = 0. \quad (2.4)$$

Since $f[n]$ is non-negative, we have that

$$f[n] = 0 \quad \forall n, \quad (2.5)$$

which contradicts the hypotheses that the sequence is not uniformly zero. In addition, we have

$$F(z) \big|_{z=0} = \sum_{n=0}^{N-1} f[n](z)^{-n} \big|_{z=0} = \infty, \quad (2.6)$$

which implies that the z -transform does not exist at $z = 0$ for the reason that the series does not converge at this point. In other word, $F(z)$ must have real root(s), and this (these) real root(s) can only be negative. ■

Figure 2.1 shows the distribution of the real zeros of the z -transform of pixel sequences with $N = 8$, which are sampled from a database of 1,000 natural images [79]. We found that all the real zeros are negative. This is consistent with our assumption above. Moreover, most of the negative real zeros are distributed in the neighborhood of the unit circle.

Figure 2.1: Statistical distribution of the real zeros.

Another problem is the number of negative real zeros of Equation (2.1). According to Kac formula [80, 81], the expected number of negative real zeros of a random polynomial of degree 7 with independently and identically distributed (*i.i.d.*) coefficients is approximately 1. Through our investigation, in the case of image pixel sequences of length 8, there is only one negative real zero under most circumstances. From our experiments, the probability of occurrence of multiple negative real zeros is determined to be only 0.02%.

2.2 The proposed fragile watermarking method by zero encoding

Based on the analytical concepts discussed in the previous section, we now describe our proposed fragile watermarking algorithm. The watermark embedding and image authentication processes are shown in Figures 2.2(a) and 2.2(b), respectively.

(a)

(b)

Figure 2.2: Fragile watermarking by zero encoding: (a) watermark embedding; (b) image authentication.

2.2.1 Watermark embedding

The original image, $\mathbf{X}$, is divided into non-overlapping blocks of size $N \times N$, where N is an even positive integer. By arranging it row by row, each block can be expressed as a sequence of vectors $\{\mathbf{x}_m\}$, $m = 0, 1, \dots, N - 1$, where $\mathbf{x}_m = \{x_m[n]\}$, $n = 0, 1, \dots, N - 1$.

We then perform 1-D z -transform on every vector using a variation of Equation (2.1):

$$F_m(z) = \sum_{n=0}^{N-1} x_m[n] z^{-n}, \quad (2.7)$$

where $m = 0, \dots, N - 1$. Now by setting

$$F_m(z) = 0, \quad (2.8)$$

we obtain the zeros of $F_m(z)$ by solving Equation (2.8), which are denoted as $\{z_{m,i}\}$, $i = 1, \dots, N - 1$, $m = 0, \dots, N - 1$. The watermark $\mathbf{w}$ is embedded by slightly perturbing the locations of the zeros, where $\mathbf{w}$ is a binary sequence of length N . The watermark bits are randomly generated for every block and the initial seed number is contained in a secret key file. A watermark signal of N bits long is embedded into every block, that is, one bit is embedded into every vector. To avoid the complex number computation, we embed the authentication watermark by slightly modifying the modulus of negative real zeros, which are denoted by z_{nr} . As proved in Section 2.1, since N is even, which is conventionally adopted in image processing, there must be at least one real negative zero in every zero set. When there is no negative real zero, that is, $x_m[0] = 0$ or $x_m[N - 1] = 0$ or the the pixel sequence is uniformly zero, a minor modification is performed to Equation (2.7) to solve the problem:

$$F_m(z) = \sum_{n=0}^{N-1} (x_m[n] + \alpha) z^{-n} \quad (2.9)$$

where α is a small positive offset. In our experiment, α is set to be approximately 10^{-4} , which is experimentally proved to be sufficient to solve the above problem without introducing visible changes to the reconstructed image (PSNR¹ > 40 dB). When there are

¹PSNR stands for peak signal-to-noise ratio defined by:

$$PSNR = 20 \cdot \log_{10} \left(\frac{255}{\sqrt{MSE}} \right),$$

where MSE is the mean squared error between two $m \times n$ images I and K and is defined by:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2.$$

multiple negative real zeros, we would embed the watermark bit into each of them. Since such a case is exceptionally rare as mentioned in Section 2.1, it does not introduce any additional distortion on the watermarked image.

Besides the negative real zero, there are $\frac{N}{2} - 1$ pairs of complex zeros for every vector (the number of complex zeros would be less if multiple negative real zeros exist). Let V_m be the number of complex zeros in the vector $\mathbf{x}_m$, whose phase angles satisfy $\theta \in (0, \frac{\pi}{2}]$, and let K be a user defined key, we have

$$\mathbf{h} = \mathcal{H}(K, V_0, V_1, \dots, V_{N-1}), \quad (2.10)$$

where $\mathcal{H}$ is a hash function and $\mathbf{h}$ is the N bits output. We then compute

$$\mathbf{p} = \mathbf{w} \oplus \mathbf{h}, \quad (2.11)$$

where $\oplus$ is the exclusive *or* function. The watermarking is performed by embedding $\mathbf{p}$ into the negative real zeros. The embedding process is defined as follows:

$$z'_{nr} = \begin{cases} -1 + 1.5\epsilon + U(z_{nr} + 1)(0.5\epsilon) & \text{if } p=1 \\ -1 - 1.5\epsilon - U(-z_{nr} - 1)(0.5\epsilon) & \text{if } p=0, \end{cases} \quad (2.12)$$

and

$$U(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases},$$

where $z_{nr}, z'_{nr} \in \mathbb{R}^-$ are the original and the watermarked zeros, respectively, and $0 < \epsilon <$

Figure 2.3: An example of watermarking by zero encoding: (a) an 8×8 block; (b) the zeros of the z -transform of x_0 ; (c) watermarking of x_0 ; (d) watermarking of x_7 .

0.1. The small positive offset ϵ determines the fragility of the watermarking scheme. The setting of ϵ will be discussed in Section 2.3.2.

As an illustrative example, consider the 8×8 pixel block in Figure 2.3(a). By performing z -transform to x_0 , we obtain its z -transform $F_0(z)$, whose roots are shown in Figure 2.3(b). There is a unique negative real root $z_{nr} = -0.9580$ among them and it is inside the unit circle. This negative real zero is then watermarked using Equation (2.12). The watermarking process is also pictorially shown in Figure 2.3(c). The watermarking process is similar for the rest of the rows. Figure 2.3(d) shows the watermarking process of x_7 , whose z -transform has a negative real zero $z_{nr} = -1.0538$ and it is outside the unit circle. Depending on the watermark sequence and the zero locations, occasionally, we need to

force some zeros to cross the unit circle for the encoding. This may introduce some minor distortion to the local areas of the watermarked image. However, from the experimental results such a distortion is negligible and not perceptually visible.

After the watermark embedding process, we transform the zeros back to the sequence using Equations (2.2). We then obtain another vector, $\mathbf{x}'_m$, which is slightly different from the one before watermarking. By applying the above process to all the relevant blocks we obtain the watermarked image $\mathbf{Y}$.

2.2.2 Image authentication

In the authentication process, we need both the watermarked image and the secret key to identify the watermark. Let the watermarked image after passing through various communication channels be $\hat{\mathbf{Y}}$. The watermark sequence $\mathbf{w}$ is generated using the initial state number contained in the key. The authentication process also starts by dividing the image into small blocks of size $N \times N$. By applying the z -transform to every row in every block, the zeros are obtained. We find the values of $\{V_m\}$ and compute $\mathbf{h} = \mathcal{H}(K, V_0, V_1, \dots, V_{N-1})$. We then obtain $\mathbf{p}$ by Equation (2.11). The negative real zeros $\{\hat{z}_{nr}\}$ are compared to the bits of $\mathbf{p}$ based on the following conditions:

$$R_m = \begin{cases} 1 & \text{if } (\hat{z}_{nr} > -1 + \epsilon \text{ and } p = 1) \\ & \text{or } (\hat{z}_{nr} < -1 - \epsilon \text{ and } p = 0) , \\ 0 & \text{otherwise} \end{cases} \quad (2.13)$$

where $\vee$ is an OR function and R_m is a Boolean variable which indicates the authenticity of the pixel vector $\mathbf{x}_m$, $m = 0, \dots, N - 1$. $R_m = 0$ represents the altered regions while $R_m = 1$ represents intact areas. Similarly, when no negative real zero is found, we apply Equation (2.9) instead of Equation (2.7). The authenticity of a block can be expressed as

$$R = R_0 \vee R_1 \vee \dots \vee R_{N-1}. \quad (2.14)$$

After checking all the image blocks, the tampered parts are located to determine the authenticity of the input image.

2.3 Performance analysis

In this section, we provide the performance analysis of the proposed fragile watermarking scheme in terms of fragility and security.

2.3.1 Fragility of watermark to pixel perturbation

As mentioned previously, a fragile watermarking scheme should detect any change to the watermarked image. From our observation, the zero locations of z -transform are very sensitive to the value change of even a single pixel, which renders the z -transform domain ideal for fragile watermarking. The relationship between pixel and zero values are mathematically given by Equation (2.2). However, the expansion form of Equation (2.2) seems to be too complex to allow strict theoretical analysis. Nevertheless, we have investigated this property experimentally. We collected 1,000 gray-scale natural images [79] and calculated

Figure 2.4: Offsets of the modulus of negative real zeros caused by one gray-scale value change (change one pixel value into a random new value).

the negative real zeros of the z -transform of their pixel sequences as described in Section 2.2.1. We then randomly changed one pixel value in every sequence, to another random selected new value, and calculated the amplitudes of the offsets of the negative real zeros, which are illustrated in Figure 2.4. It can be observed that even a single pixel's change unavoidably disturb the zero locations. In addition, we found that approximately 98% of the negative real zeros tend to shift towards the unit circle and as illustrated in Figure 2.4, the distribution peaks at an offset of 0.18, are sufficient to change the watermark detection results.

2.3.2 The setting of watermarking strength ϵ

In Figure 2.5, we illustrate the sensitivity of the scheme to alterations on the images with different settings of the parameter ϵ . The sensitivity is measure in term of the detection probability of 1 pixel's alteration. We tested this detection probability in blocks of 1,000 images. As we can observe, smaller ϵ achieved higher sensibility, and increasing ϵ resulted

Figure 2.5: The sensitivity of the authentication with increasing ϵ .

in slowly decreasing sensitivity. On the other hand, as shown in Figure 2.6, increasing ϵ also resulted in degrading visual quality of the watermarked images. The visual quality of the images was measured by PSNR. However, since the positions of the zeros are extremely sensitive, even small rounding during the processing causes them to shift, and in turn leads to false alarms. The solution is keeping the ϵ from being too small, which results in the tradeoff between the sensitivity and the probability of false alarm in term of the setting of ϵ . However, as illustrated in Figure 2.7, the probability of false alarm is considered to be acceptable with $\epsilon > 0.015$.

2.3.3 Security

The most important security issue for block-independent fragile watermarking are the “cutting and pasting” counterfeiting attacks. Suppose every block is watermarked independently using the same watermark signal. The attacker then would swap two blocks in the

Figure 2.6: The visual quality of the watermarked images with increasing ϵ .

Figure 2.7: The probability of false alarm with increasing ϵ .

same watermarked image. The authentication would succeed while the image has been altered. Such swapping attack can be efficiently prevented by using pseudo-random distributed watermark for each block. More sophisticated replacement attack has been proposed by Holliman and Memon [15]. Suppose the attacker wants to alter some or all of the pixels in block x so that it becomes another block $\hat{x}$. He would generate many approximations of the block $\hat{x}$, which are slight perturbations of $\hat{x}$. He would then select the block that has the same hash output as x . Some counter solutions have been proposed [14, 16] based on a tradeoff between accuracy in localization and security. In our scheme, we avoid the “cutting and pasting” attacks by incorporating the hashed phase information of the complex zeros into the watermark. Unlike LSBs based methods, for the attack to be successful, the attacker has to manipulate the phases of the complex zeros of $\hat{x}$, which would result in significant decrease of the image quality. Moreover, when the phases of the complex zeros are changed, it is not likely that the visual content of the block would remain similar to the target block, since the phases of zeros have an important role in describing the content of an image [78]. From the experimental results in [78], we found that most of the details of the images can be reconstructed given only accurate zero phases, while the reconstruction is far less satisfactory when only given the zero magnitudes.

Figure 2.8: Experimental results: (a) the original image; (b) the watermarked image.

2.4 Experimental results

For our experiment, we used the 512×512 gray scale image *Couple* as shown in Figure 2.8(a) to test our authentication algorithm. A block size of 8×8 was used, which is commonly used in image processing applications. The watermarked image is displayed in Figure 2.8(b). We observed that the watermarked image looks identical to the original image, with a PSNR value of approximately 40 dB. We modified the content of the watermarked image as shown in Figure 2.9: (a) the background was intentionally blurred; (b) the table was removed and (c) the positions of the man and the woman in the image were exchanged. The authentication results of the attacked images are shown in Figures 2.9(d)-(f), respectively. We can see that the modified areas were accurately detected and identified.

Exact authentication by zero encoding

45

Figure 2.9: Tampered images and authentication results: (a) the background was intentionally blurred; (b) the table was removed; (c) the positions of the man and the woman in the image were exchanged; (d) the authentication results of (a); (e) the authentication results of (b); (f) the authentication results of (c).

2.5 Summary

We discussed a novel fragile watermarking method based on the encoding of the zeros of the z -transform. The watermark bits were embedded by slight perturbation of the zero locations. The zeros of the z -transform around the unit circle are very sensitive to any change on the host image. Even a single pixel's change unavoidably disturb the zero locations. This important property provided the scheme with special sensitivity to any alteration to the watermarked image and the ability of accurate localizing. In addition, the proposed method is more secure than normal fragile watermarking techniques based on LSBs embedding. Simulation results confirmed the applicability of the proposed algorithm.

Chapter 3

Selective authentication using pinned sine transform

The preceding chapter covers the issue of exact authentication using fragile watermarking. In this chapter, we focus on selective content authentication by semi-fragile watermarking. Selective authentication addresses the problem of detecting any modification on image content by illegitimate distortions, such as malicious content counterfeit, while allowing modification by legitimate distortions, such as JPEG compression.

The motivation for developing a semi-fragile watermarking based on the pinned sine transform (PST) is due to the observation that this transform could provide an effective way to solve the above semantic content authentication problem. The observation is as follows. The PST conducts a decomposition of the original image into two mutually uncorrelated fields, that is, the boundary field and the pinned field. The texture information of the orig-

inal image is contained in the pinned field, wherein the sine transform is equivalent to a fast Karhunen-Loeve transform (KLT). By exploiting this important property, we propose to embed a watermark signal into the sine transform domain of the pinned field for content authentication. As illustrated in this chapter, the proposed watermarking scheme is especially sensitive to texture alterations of the host image, while it can tolerate common image processing operations which do not affect the semantic content of the images.

In Section 3.1, the PST is briefly reviewed. Our proposed watermarking method for image authentication is detailed in Sections 3.2.1 and 3.2.2. In Section 3.3.1, we discuss how the proposed scheme ensures a selective content authentication. The proposed scheme's robustness to vector quantization (VQ) counterfeiting attacks is demonstrated in Section 3.3.2, followed by experimental results in Section 3.4.

3.1 The pinned sine transform

The orthogonal transforms effect a reduction in the correlation which exists between the individual data samples and result in a set of coefficients the members of which may be separately processed. Obviously the most efficient transform is that which minimizes the correlation between the obtained transform coefficients. The “ultimate” transform is the Karhunen-Loeve transform (KLT), which results in a transform domain covariance matrix the components of which are uncorrelated. However, this transform is impossible to implement in practice, unless the ensemble of images to be transmitted has a well-known

underlying structure. Jain [82, 83, 84] and Meiri [85] have shown that if the image is a Gauss-Markov field of certain type, there is a KLT which is implementable as a sine transform. The next section give a description a transform based on the above property, the pinned KLT (PKLT), and an easily implementable approximation to the pinning function is described in Section 3.1.2 which leads to the pinned sine transform (PST).

3.1.1 The pinned Karhunen-Loeve transform (PKLT)

A random sequence $x(n)$ is called a degree- p Gauss-Markov sequence if the conditional expectations satisfy the relation

$$E[x(n)|x(n-1), x(n-2), \dots] = E[x(n)|x(n-1), \dots, x(n-p)], \quad \forall n. \quad (3.1)$$

If an image field $v(\cdot, \cdot)$ is stochastically represented as a degree-2 Gauss-Markov field, which is reasonable for natural images which do not contain rapidly varying details, it solves the stochastic Poisson equation [28]

$$\frac{\partial^2 v(x, y)}{\partial x^2} + \frac{\partial^2 v(x, y)}{\partial y^2} - \pi^2 \alpha^2 v(x, y) = \xi(x, y) \quad (3.2)$$

where $\xi(x, y)$ is a 2-D Gaussian white noise, and α is a scaling parameter related to the correlation between data which are one length unit apart. Small α values correspond to highly correlated images.

The solution of Equation (3.2) inside a rectangle of size $n \times m$ as shown in Figure 3.1 can be represented as sum of four components:

$$v(x, y) = v_c(x, y) + w(x, y) + u(x, y) + v_1(x, y) \quad (3.3)$$

Figure 3.1: A rectangle field of size $n \times m$.

where $v_1(x, y)$ is the “pinned field” that vanishes on the block edges (hence the name “pinned”) and solves Equation (3.2) for the zero boundary conditions; $v_0(x, y) \triangleq v_c(x, y) + w(x, y) + u(x, y)$ is the “pinning field” representing the boundary conditions existing on the block boundaries and is a solution of the homogeneous version of Equation (3.2), i.e.,

$$\frac{\partial^2 v(x, y)}{\partial x^2} + \frac{\partial^2 v(x, y)}{\partial y^2} - \pi^2 \alpha^2 v(x, y) = 0; \quad (3.4)$$

the “corner field” $v_c(x, y)$ solves Equation (3.4) and agrees with $v_0(x, y)$ at the four corners:

$$\begin{aligned} & v_c(x, y) \sinh \pi \alpha_1 m \sinh \pi \alpha_1 n \\ &= v(0, 0) \sinh \pi \alpha_1 (m - x) \sinh \pi \alpha_1 (n - y) \\ &+ v(0, n) \sinh \pi \alpha_1 m - x \sinh \pi \alpha_1 y \\ &+ v(m, 0) \sinh \pi \alpha_1 x \sinh \pi \alpha_1 (n - y) \\ &+ v(m, n) \sinh \pi \alpha_1 x \sinh \pi \alpha_1 y \end{aligned} \quad (3.5)$$

where

$$\alpha_1 \triangleq \alpha / \sqrt{2},$$

and $\sinh(\cdot)$ is the hyperbolic sine function defined as

$$\sinh(x) = \frac{e^x - e^{-x}}{2},$$

and $w(x, y)$ and $u(x, y)$ are also solutions of Equation (3.4) with boundary conditions which vanish on the horizontal and vertical edges, respectively, and they satisfy on the other edges the following conditions:

$$\begin{aligned} w(0, x) &= V_{x0}(x) - v_c(x, 0) \\ w(x, n) &= V_{xn}(x) - v_c(x, n) \\ u(0, y) &= V_{0y}(y) - v_c(0, y) \\ u(m, y) &= V_{my}(y) - v_c(m, y) \end{aligned} \tag{3.6}$$

where $V_{x0}(\cdot)$, $V_{xn}(\cdot)$, $V_{0y}(\cdot)$ and $V_{my}(\cdot)$ are the block boundaries as illustrated in Figure 3.2, and $w(x, y)$ is given by

$$w(x, y) = \sum_{r=1}^{\infty} \left\{ V_{x0_r} \frac{\sinh \left[\left(r^2 + m^2 \alpha^2 \right)^{1/2} \pi \frac{n-y}{m} \right]}{\sinh \left[\left(r^2 + m^2 \alpha^2 \right)^{1/2} \pi \frac{n}{m} \right]} + V_{xn_r} \frac{\sinh \left[\left(r^2 + m^2 \alpha^2 \right)^{1/2} \pi \frac{y}{m} \right]}{\sinh \left[\left(r^2 + m^2 \alpha^2 \right)^{1/2} \pi \frac{n}{m} \right]} \right\} e_r, \tag{3.7}$$

where the $e_r = \sqrt{2/m} \sin(k\pi x/m)$ are the sine transform basis functions, and V_{x0_r} , V_{xn_r} are the sine transform coefficients of, respectively, $V_{x0}(\cdot)$, $V_{xn}(\cdot)$ and $u(x, y)$ is given by a similar expression. The pinned field is then given by

$$v_1(x, y) = v(x, y) - v_0(x, y). \tag{3.8}$$

When we retain the boundary field $v_0(x, y)$ and apply the sine transform to the pinned field

Figure 3.2: The PST for a typical block. Each block has one new corner and two new boundaries.

$v_1(x, y)$, the sine transform coefficients of $v_1(x, y)$ are orthogonal, i.e., the sine transform is equivalent to the KLT when applied to the pinned field $v_1(x, y)$. A strict proof can be found in [28].

3.1.2 The approximate pinning function and PST

The complexity of the pinning function in Equations (3.5-3.7) makes the PKLT difficult to implemented in practice. In view of this drawback, the complicated PLKT pinning function is approximated by simpler interpolation. The approximate pinning function agrees with the original values at block boundaries and is a good estimator of in-block pixels. A discrete version was also derived. An approximation to the pinning function v_0 , which has the same general form as the exact pinning field was proposed in as:

$$\tilde{v}_0(x, y) = \tilde{v}_c(x, y) + \tilde{w}(x, y) + \tilde{u}(x, y). \quad (3.9)$$

The corner function v_c is approximated by *bilinear interpolation* of the four corners and is given by

$$\tilde{v}_c(x, y) = V_{00} + \frac{V_{m0} - V_{00}}{m}x + \frac{V_{0n} - V_{00}}{n}y + \frac{V_{00} + V_{mn} - V_{0n} - V_{m0}}{mn}xy. \quad (3.10)$$

The boundaries $\tilde{w}$ and $\tilde{u}$ have the form:

$$\tilde{w}(x, 0) = V_{x0}(x) - \tilde{v}_c(x, 0)$$

$$\tilde{w}(x, n) = V_{xn}(x) - \tilde{v}_c(x, n) \quad (3.11)$$

$$\tilde{w}(0, y) = \tilde{w}(m, y) = 0$$

and likewise for $\tilde{u}$. The in-block values are then *linearly interpolated* as follows:

$$\tilde{w}(x, y) = \tilde{g}(x) + \frac{\tilde{h}(x) - \tilde{g}(x)}{n}y \quad (3.12)$$

where

$$\tilde{g}(x) \triangleq \tilde{w}(x, 0), \quad \tilde{h}(x) \triangleq \tilde{w}(x, n),$$

with a similar expression for $\tilde{u}(x, y)$.

This procedure, which consists of linear and bilinear interpolation for approximating the actual pinning field, is called the *boundary field*. By subtracting the boundary field from the original field produces an approximate pinned field. This pinned field resulting from the bilinear pinning function does not have the property that its sine transform is exactly equivalent to the KLT, however, it is a good practical approximation to the PKLT scheme. Its pinning function is simple to implement and it does not depend on specific

model parameters. In [28], this approximation to the PKLT was called the pinned sine transform (PST).

3.2 The proposed semi-fragile watermarking using PST

In this section, we present our proposed semi-fragile watermarking method using the pinned sine transform. The watermark embedding process is described in Section 3.2.1, and the processes of watermark detection and image authentication are described in Section 3.2.2.

3.2.1 Watermark embedding

The watermark embedding process is described in Figure 3.3 and the details are described as follows. The original image $\mathbf{X}$ is partitioned into non-overlapping blocks of size $N \times N$ as shown in Figure 3.4. Let us consider a typical block $\mathbf{x}_{m,n}$, where m and n are the coordinate numbers of this block, we define its corner response as

$$\mathbf{c}_{m,n} = \{c_{11}, c_{1N}, c_{N1}, c_{NN}\} \quad (3.13)$$

and its boundary response as

$$\mathbf{b}_{m,n} = \{\mathbf{b}_{1x}, \mathbf{b}_{Nx}, \mathbf{b}_{y1}, \mathbf{b}_{yN}\} \quad (3.14)$$

as illustrated in Figure 3.4. The corner response is obtained using the corner function:

$$\mathbf{c}_{m,n} = \mathcal{C}[\mathbf{x}_{u,v}; m-1 \leq u \leq m+1, n-1 \leq v \leq n+1] \quad (3.15)$$

Figure 3.3: Semi-fragile watermark embedding process for one block.

Figure 3.4: The dual-field decomposition in PST for a typical block.

More specifically, the corner function is defined as follows:

$$\begin{aligned}
 c_{11} &= \frac{\mathbf{x}_{m,n}(1,1) + \mathbf{x}_{m-1,n-1}(N,N) + \mathbf{x}_{m-1,n}(N,1) + \mathbf{x}_{m,n-1}(1,N)}{4} \\
 c_{1N} &= \frac{\mathbf{x}_{m,n}(1,N) + \mathbf{x}_{m-1,n}(N,N) + \mathbf{x}_{m-1,n+1}(N,1) + \mathbf{x}_{m,n+1}(1,1)}{4} \\
 c_{N1} &= \frac{\mathbf{x}_{m,n}(N,N) + \mathbf{x}_{m,n-1}(N,N) + \mathbf{x}_{m+1,n-1}(1,N) + \mathbf{x}_{m+1,n}(1,1)}{4} \\
 c_{NN} &= \frac{\mathbf{x}_{m,n}(N,N) + \mathbf{x}_{m,n+1}(N,1) + \mathbf{x}_{m+1,n}(1,N) + \mathbf{x}_{m+1,n+1}(1,1)}{4},
 \end{aligned} \tag{3.16}$$

and the boundary response is defined by the boundary function

$$\mathbf{b}_{m,n} = \mathcal{B}[\mathbf{x}_{u,v}: m-1 \leq u \leq m+1, n-1 \leq v \leq n+1] \quad (3.17)$$

which is further defined as

$$\begin{aligned} \mathbf{b}_{1x}(i) &= \frac{\mathbf{x}_{m,n}(1,i) + \mathbf{x}_{m-1,n}(N,i)}{2} \\ \mathbf{b}_{Nx}(i) &= \frac{\mathbf{x}_{m,n}(N,i) + \mathbf{x}_{m+1,n}(1,i)}{2} \\ \mathbf{b}_{y1}(j) &= \frac{\mathbf{x}_{m,n}(j,1) + \mathbf{x}_{m,n-1}(j,N)}{2} \\ \mathbf{b}_{yN}(j) &= \frac{\mathbf{x}_{m,n}(j,N) + \mathbf{x}_{m,n+1}(j,1)}{2}. \end{aligned} \quad (3.18)$$

As we can see from Equations (3.15)-(3.18), the processing of one block should involve all the blocks surrounding it, and we can observe in Figure 3.4 that in a sequential processing of blocks, only one new corner c_{NN} and two new boundaries $\mathbf{b}_{Nx}$ and $\mathbf{b}_{yN}$ are required to be computed for a new input block.

The boundary field of $\mathbf{x}_{m,n}$ is achieved by the pinning function [28]:

$$\mathbf{x}_{m,n}^b = \mathcal{PIN}[\mathbf{c}_{m,n}, \mathbf{b}_{m,n}]. \quad (3.19)$$

Corresponding to the above general form, the specific form of the pinning function is defined as follows:

$$\begin{aligned} \mathbf{x}_{m,n}^b(i, j) &= \mathbf{x}_{m,n}(1, 1) + (c_{1N} - c_{11}) \frac{(i-\frac{1}{2})}{N} \\ &\quad + (c_{N1} - c_{11}) \frac{(j-\frac{1}{2})}{N} \\ &\quad + (c_{11} + c_{NN} - c_{N1} - c_{1N}) \frac{(i-\frac{1}{2})(j-\frac{1}{2})}{N^2} \\ &\quad + \mathbf{g}_x(i) + (\mathbf{h}_x(i) - \mathbf{g}_x(i)) \frac{j-\frac{1}{2}}{N} \\ &\quad + \mathbf{g}_y(j) + (\mathbf{h}_y(j) - \mathbf{g}_y(j)) \frac{i-\frac{1}{2}}{N} \end{aligned} \quad (3.20)$$

where

$$\begin{aligned}
 \mathbf{g}_x(i) &= \mathbf{b}_{Nx}(i) - \left(c_{N1} + \frac{c_{NN}-c_{N1}}{N} \left(i - \frac{1}{2} \right) \right) \\
 \mathbf{h}_x(i) &= \mathbf{b}_{1x}(i) - \left(c_{11} + \frac{c_{1N}-c_{11}}{N} \left(i - \frac{1}{2} \right) \right) \\
 \mathbf{g}_y(j) &= \mathbf{b}_{yN}(j) - \left(c_{1N} + \frac{c_{NN}-c_{1N}}{N} \left(j - \frac{1}{2} \right) \right) \\
 \mathbf{h}_y(j) &= \mathbf{b}_{y1}(j) - \left(c_{11} + \frac{c_{N1}-c_{11}}{N} \left(j - \frac{1}{2} \right) \right)
 \end{aligned} \tag{3.21}$$

are the pinned boundaries. The pinned field $\mathbf{x}_{m,n}^p$ is then given by

$$\mathbf{x}_{m,n}^p = \mathbf{x}_{m,n} - \mathbf{x}_{m,n}^b. \tag{3.22}$$

Next, we perform a sine transform¹ to this pinned field block as follows:

$$\mathbf{x}_{m,n}^{p(s)} = \mathbf{S}_N \mathbf{x}_{m,n}^p \mathbf{S}_N^T \tag{3.23}$$

where $\mathbf{S}_N$ is the sine transform matrix of order N which is defined as [76]

$$S_N(i, j) = \sqrt{\frac{2}{N+1}} \sin \frac{\pi(i+1)(j+1)}{N+1} \tag{3.24}$$

where $0 \leq i, j \leq N-1$.

We use a pseudo-random binary sequence as the watermark for image authentication. The length of the sequence L and its initial state number is contained as part of the secret key file $\mathcal{K}$. The watermark embedding process proceeds by embedding the pseudo-random sequence into each sine transformed pinned field block. The watermark signal intended to be embedded into this block is denoted as:

$$\mathbf{w}_{m,n} = \left\{ w_{m,n}[k] \right\} \tag{3.25}$$

¹Refer to Appendix A for the fast algorithm for discrete sine transform.

with $k \in \mathcal{L} = \{1, 2, \dots, L\}$ and $w_{m,n}[k] \in \{0, 1\}$. In the middle to high frequency bands of $\mathbf{x}_{m,n}^{p(s)}$, we select, according to the length of the watermark sequence, L coefficients for watermarking modulation. The watermarking function is then given by

$$\hat{\mathbf{x}}_{m,n}^{p(s)} = \mathcal{F} \left[\mathbf{x}_{m,n}^{p(s)}, \mathbf{w}_{m,n}, \mathcal{K} \right] \quad (3.26)$$

where $\hat{\mathbf{x}}_{m,n}^{p(s)}$ is the block of watermarked sine transform coefficients. More specifically, a specific bit $w_{m,n}(k)$ is embedded into a coefficient $x_{m,n}^{p(s)}(k)$ according to the following algorithm:

Algorithm 3.1 *Watermark embedding of semi-fragile watermarking by PST*

if $w_{m,n}(k) = 1$ **then**

if $x_{m,n}^{p(s)}(k) > \lambda$ **then**

$$\hat{x}_{m,n}^{p(s)}(k) = x_{m,n}^{p(s)}(k)$$

else

$$\hat{x}_{m,n}^{p(s)}(k) = \alpha_1$$

end if

else if $w_{m,n}(k) = 0$ **then**

if $x_{m,n}^{p(s)}(k) < -\lambda$ **then**

$$\hat{x}_{m,n}^{p(s)}(k) = x_{m,n}^{p(s)}(k)$$

else

$$\hat{x}_{m,n}^{p(s)}(k) = \alpha_2$$

end if

end if

where

- $\hat{x}_{m,n}^{p(s)}(k)$ is the corresponding watermarked coefficient;
- $0 < \lambda < 255$ is a threshold which can be determined by users; its value will affect the tradeoff between the perceptual quality of the watermarked image and the robustness of the semi-fragile watermark;
- α_1 and α_2 are floating point values chosen randomly from $[\lambda/2, \lambda]$ and $[-\lambda, -\lambda/2]$, respectively.

The watermarked pinned field block is obtained by the inverse 2-D sine transform:

$$\hat{\mathbf{x}}_{m,n}^p = \mathbf{S}_N^T \hat{\mathbf{x}}_{m,n}^{p(s)} \mathbf{S}_N \quad (3.27)$$

and a watermarked block is therefore achieved by

$$\hat{\mathbf{x}}_{m,n} = \hat{\mathbf{x}}_{m,n}^p + \mathbf{x}_{m,n}^b. \quad (3.28)$$

After processing all the blocks, the watermarked image is the union of all the watermarked blocks:

$$\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_{m,n}\}. \quad (3.29)$$

3.2.2 Watermark detection and image authentication

The watermark detection and image authentication processes is illustrated in Figure 3.5.

The detection system receives as input a watermarked and possibly tampered image $\tilde{\mathbf{X}}$.

Figure 3.5: Semi-fragile watermark detection and image authentication processes for one block.

Similar to the watermarking process, a decomposition is performed on $\tilde{\mathbf{X}}$ by Equations (3.13)-(3.22), and then we obtain the sine transform coefficients of its pinned field by Equation (3.23). Let us consider the sine transform components matrix of a certain watermarked pinned field block:

$$\tilde{\mathbf{x}}_{m,n}^{p(s)} = \left\{ \tilde{x}_{m,n}^{p(s)}(k) \right\}. \quad (3.30)$$

The retrieved and possibly corrupted watermark $\tilde{\mathbf{w}}_{m,n}$ is decided based on the watermark detection function:

$$\tilde{\mathbf{w}}_{m,n} = \mathcal{G} \left[\tilde{\mathbf{x}}_{m,n}^{p(s)}, \mathcal{K} \right]. \quad (3.31)$$

More specifically, $\mathcal{G}[\cdot]$ is described by the algorithm below:

Algorithm 3.2 *Watermark detection of semi-fragile watermarking by PST*

if $\tilde{x}_{m,n}^{p(s)}(k) \geq 0$ **then**

$$\tilde{w}_{m,n}(k) = 1$$

else

$$\tilde{w}_{m,n}(k) = 0$$

end if

where $\check{w}_{m,n}$ is the extracted watermark.

The original watermark signal $w_{m,n}$ is also generated using the initial state number in the $\mathcal{K}$, and this binary sequence with elements $\{0, 1\}$ is mapped into a corresponding bipolar sequence with elements $\{-1, 1\}$. The watermark bits are compared via the normalized cross-correlation function [86]:

$$\rho = \frac{\sum_{k=1}^L \check{w}_{m,n}(k) w_{m,n}(k)}{\left[\sum_{k=1}^L (\check{w}_{m,n}(k))^2 \right]^{1/2} \left[\sum_{k=1}^L (w_{m,n}(k))^2 \right]^{1/2}}, \quad (3.32)$$

where $\rho \in [-1, 1]$.

The integrity of the block $\check{x}_{m,n}$ is evaluated according to the value of ρ . The block is considered to be maliciously tampered if $\rho < \gamma$, where the threshold γ is determined mathematically or experimentally so as to maximize the probability of tampering detection subject to a given probability of false alarm. In our current simulations, γ is experimentally set to tolerate unavoidable non-malicious modifications in some practical applications, such as JPEG compression and noise addition, while maintaining the sensitivity of the authentication process to malicious modification on the content of the watermarked images.

3.3 Performance analysis

Two main issues which concern semi-fragile watermarking are selective authentication and the robustness of vector quantization (VQ) counterfeiting attacks. In this section, we present a theoretical analysis of the performance of the proposed scheme in terms of the above issues.

Figure 3.6: The dual-field decomposition in the pinned sine transform of the *Dubai* image: (a) the original image, (b) the boundary field and (c) the pinned field.

3.3.1 Dual-field decomposition and selective content authentication

The semi-fragile watermarking provides a selective authentication on the content of images. Our scheme aims at protecting the primary textures, such as edges, of the images. To this end, the watermark should not survive the authentication process if such textures are tampered or damaged. The results of the PST dual-field decomposition of the 512×512 *Dubai* image using Equations (3.13)-(3.22) are shown in Figure 3.6. We find that the boundary field is only a blurred version of the original image, while the pinned field is a good characterization of edges, which largely reflects the texture information in the original image. The watermark can be embedded into the pinned field as an indicator of the authenticity of the watermarked image. Moreover, since most common image manipulations tend to preserve such primary features of images, this embedding method ensures that the watermark does not suffer significantly from such legitimate manipulations.

Figure 3.7: The inter-block relationship in the pinned sine transform.

3.3.2 Inter-block relationship and counterfeiting attacks

The most important malicious attacks to existing watermarking authentication schemes are the “cutting and pasting” attacks. The well-known vector quantization (VQ) counterfeiting attack proposed by Holliman and Memon [15] is one such attack which thwarts many existing block-wise fragile watermarking methods. A brief review of the VQ attack by Holliman and Memon has been provided in Section 1.2.3. In the following section, we explain why our scheme can survive the VQ attack.

In our scheme, we exploit the intrinsic inter-block dependence in PST to detect the above counterfeiting attacks. The “PST style” encoding in Equations (3.13)-(3.22) introduces an inter-block relationship to the pinned sine transformed images as shown in Figure 3.7. The watermarking of any particular block also depends on its location in the image instead of depending only on its own content. Thus, simple VQ counterfeiting attack can be exposed by this encoding style since the counterfeit of one block affects all the surrounding

blocks. Moreover, as mentioned in Section 1.2.3, the success of the VQ attack relies on the codebooks which is constructed by categorizing all the blocks in the watermarked images. However, for the proposed scheme, the construction of codebooks would be very difficult for the reason that the identification of one block would take all the blocks around it into consideration.

3.4 Simulation results

Three 512×512 gray-scale images with different contents and textures were used to test our PST authentication algorithm. The block size in our experiments was 8×8 . The original images are shown in Figure 3.8. The images shown in Figures 3.8(a) and 3.8(b) are simple natural images. Figure 3.8(c) is a satellite image with complex texture and fine details. Figures 3.8(d)-(f) display the respective watermarked images. We can see that the watermarked images look identical to the original images, with PSNR greater than 33 dB. In Figure 3.9, we show the tradeoff between the length of the watermark and the visual quality of the watermarked images. In the following experiments, a watermark length of $L = 6$ was used, with which the visual quality of the images is acceptable while the robustness of the system is ensured, and finally we set $\lambda = 50$.

We modified the content of the watermarked images in a similar way to the “cutting and pasting” attacks: all the modifications were performed by cutting and pasting blocks in the same or similar watermarked images. The modification results are shown in Figures

(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.8: Simulation results: (a) original *Couple*, (b) original *Tank*, (c) original *Pyramids* and (d)-(f) the watermarked images (PSNR: (d) 33 dB; (e) 36 dB; (f) 35 dB).

Figure 3.9: The tradeoff of hiding rate and image quality.

3.10(a)- 3.10(c). The modifications made to the respective images are: the table in the bottom right corner was removed from the *Couple* image; the tank was shifted in the *Tank*; and in the *Pyramids*, some geographical textures were modified. As illustrated in Figures 3.10(d)-3.10(f), the modified areas were accurately detected and identified. We define the probability of tampering detection P_{TD} of the authentication scheme as

$$P_{TD} = \frac{\text{NUM}_{\text{detected}}}{\text{NUM}_{\text{modified}}} \quad (3.33)$$

where $\text{NUM}_{\text{modified}}$ is the number of modified blocks, and $\text{NUM}_{\text{detected}}$ is the number of detected blocks. In our experiments, P_{TD} without non-malicious attacks was always higher than 98%.

We also tested the insensitivity of our algorithm to compression. As shown in Figure 3.11, before compression, the output ρ of the watermark detection system sharply peaked at

(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.10: Tampered images and authentication results: (a) the table was removed; (b) the tank was shifted; (c) the texture in some part was modified; (d) authentication result of (a); (e) authentication result of (b); (f) authentication result of (c).

Figure 3.11: The distribution of the watermark detection outputs: before and after JPEG compression (QF=40).

1; after compression, the values of ρ decreased as shown in the same figure. To illustrate the advantage of PST watermarking, we compare the performance of PST watermarking with that of DCT watermarking. In the DCT watermarking, the same watermark embedding method was used and the same middle frequency bands coefficients were selected as those in the PST watermarking. The comparison was based on the same PSNR values of the watermarked images and the results were obtained through averaging the outcomes of the 1,000 test images [79]. We found that after the compression, the drop in the detector output ρ for the PST watermarking was smaller than that of the DCT watermarking. This indicates that the PST watermarking is less sensitive to JPEG compression than the DCT watermarking, which makes it a better candidate for semi-fragile watermarking. Given a certain value of the threshold γ , the probability of detection P_D is shown as the shaded area in Figure 3.11. It is apparent from this figure that the P_D of the PST scheme is larger than that of DCT. The collective comparison results with $\gamma = 0.1$ and varying compression quality factor (QF) values are reported in Figure 3.12. The higher values of P_D indicated the better detection performance of PST over DCT.

The performance of our algorithm against JPEG compression and additive noise was also tested. After content modification, the watermarked image in Figure 3.10(a) was JPEG compressed with a QF of 90% and the watermarked image in Figure 3.10(c) is added a additive white Gaussian noise of zero mean and a variance of $\sigma^2 = 5$, as shown in Figure 3.13(a) and Figure 3.13(b), respectively. As indicated in Figure 3.13(c) and 3.13(d), the modified area still can be correctly identified.

(a)

(b)

Figure 3.12: Comparisons between PST watermarking and conventional DCT watermarking: the probability of detection after (a) JPEG compression and (b) wavelet compression.

(a)

(b)

(c)

(d)

Figure 3.13: Sample authentication results after JPEG compression and additive noise: (a) watermarked and modified *Couple* image after JPEG compression (QF=90); (b) watermarked and modified *Pyramids* image with additive noise ($\sigma^2 = 5$); (c) authentication result of (a); (d) authentication result of (b).

Attacks	The probability of false alarm	
	The proposed method	Lin's method
<i>JPEG compression (90%)</i>	0.0124	0.0000
<i>JPEG compression (80%)</i>	0.0267	0.0000
<i>Additive noise</i>	0.0143	0.8976
<i>Gaussian filtering</i>	0.0156	0.9676
<i>Unsharpening</i>	0.0068	0.7845
<i>Contrast enhancement</i>	0.0048	0.7765

Table 3.1: Comparison of the probability of false alarm.

In Table 3.1, we list the probability of false alarm of the proposed algorithm subject to some common image processing operations, as well as the comparison with the previous Lin's method [18]. We observe that the overall performance of our proposed method is relatively better than Lin's method.

3.5 Summary

We investigated the problem of the selective content authentication of digital images through a novel semi-fragile watermarking using the pinned sine transform (PST). The watermark is embedded into the pinned field of PST, which contains the texture information of the original image. This important property of the pinned field provides the scheme with special sensitivity to any texture alteration to the watermarked image. The effectiveness of the new method has been demonstrated by using natural scene images and satellite images. We selected satellite images for the reason that texture information is especially critical to this kind of images. In the authentication process, the probability of tampering detection

was higher than 98%. The scheme was very robust to “cutting and pasting” counterfeiting attacks. It was also able to tolerate some common image processing manipulations; the probability of false alarm caused by JPEG compression and wavelet compression is lower than that of equivalent DCT scheme.

Chapter 4

Self-restoration with watermarking

The problem of localized authentication of image content has been studied in the previous chapters. We have introduced the fragile watermarking based on zero encoding to determine if an image has been altered and the semi-fragile watermarking scheme using PST to determine if an image has been altered illegitimately. This naturally leads to an examination of whether an altered image can be restored which is desirable in many applications, for example, for protecting important images used as courtroom evidence, insurance claims and journalistic photographs.

Self-embedding is a strategy for automatic restoration with fragile or semi-fragile watermarking. In the next section, a short review of the self-embedding method is reviewed, we describe how the self-embedding strategy can be incorporated into our proposed watermarking method and point out its shortcoming. We then propose another novel restoration scheme which provides solution to the problem which is based on the irregular sampling

theory. In addition, we suggest a new scheme which further minimizes the probability of false alarm of semi-fragile watermarking caused by legitimate operations. In Section 4.2 we give a brief review of irregular sampling theory in approximation spaces and introduce our proposed approach based on it. The performance analysis and simulation results are reported in Sections 4.3 and 4.4, respectively.

4.1 Self-embedding

The concept of self-embedding for tampered image restoration is to embed a highly compressed version of the original image as a watermark into the image itself. Thus, if some parts of the watermarked image are detected to be removed or destroyed, these modified regions can be replaced with their corresponding low-resolution versions. We incorporated the self-embedding technique into our semi-fragile watermarking method presented in the previous chapter. The low-resolution image is generated in the PST domain. As mentioned in the previous section, for the pinned field, the sine transform coder is equivalent to a fast KLT coder, which results in optimal coding. Another significant advantage of the PST coder over common used DCT techniques is that it suppresses significantly the block-effect appearing in the recovered image when the compression rate is high by retaining the continuity between blocks [28].

Figure 4.1: Watermark embedding process for one block with self-embedding.

Figure 4.2: Watermark detection and authentication processes for one block and restoration with self-embedding.

4.1.1 The proposed method

To facilitate a restoration process, the watermarking embedding and detection processes in Sections 3.2 are modified slightly as shown in Figures 4.1 and 4.2. The watermarking embedding begins with setting the least significant bits (LSBs) of all the pixels of the original image to be zero (LSB nulling); and we denote this zero-LSB image to be $\mathbf{X}^0$. The watermark is embedded as described in Section 3.2.1 into $\mathbf{X}^0$ instead of $\mathbf{X}$. At the same time, as illustrated in Figure 4.1, we generate the “recovery bits” for the self-embedding. The recovery bits are obtained by compressing the pinned field, two “new boundaries” $\mathbf{b}_{Nx}(i)$ and $\mathbf{b}_{yN}(j)$, and one “new corner” $\mathbf{c}_{NN}$, as illustrated in Figure 3.4, of every block into a

Figure 4.3: The insertion of the recovery bits.

64-bit binary string using a sine transform coder. The recovery bits of a certain block, say, $\mathbf{x}_{m_1, n_1}$ is inserted into the LSBs of another block $\mathbf{x}_{m_2, n_2} = \Omega(\mathbf{x}_{m_1, n_1})$, where Ω is a pre-determined mapping function. When the watermarked image is detected to be tampered, the recovery bits are extracted out to obtain a reconstruction of the tampered portions. The above concepts are illustrated in Figure 4.3.

Similarly, in the detection system, the watermarked and possibly tampered image $\tilde{\mathbf{X}}$ should be first divided into two components: its LSBs and a residual image $\tilde{\mathbf{X}}^0$ with all its LSBs set to zero. We then input $\tilde{\mathbf{X}}^0$ to the authentication process as described in Section 3.2.2. If any block of the image, say, $\tilde{\mathbf{x}}_{m_1, n_1}$ is found to have been removed or tampered, it can be restored using the information hidden in the LSBs of $\tilde{\mathbf{x}}_{m_2, n_2}$, under the assumption that $\tilde{\mathbf{x}}_{m_2, n_2}$ has been confirmed to be reliable by the authentication process. However, the restoration fails if both of $\tilde{\mathbf{x}}_{m_1, n_1}$ and $\tilde{\mathbf{x}}_{m_2, n_2}$ are tampered.

4.1.2 Experiments and analysis

The original test image *Traffic* is shown in Figure 4.4(a). Figure 4.4(b) shows the watermarked image with self-embedding. Figure 4.4(c) shows the reconstructed image obtained through uncompressing the recovery bits embedded in LSB. Figure 4.5 shows the restoration results of the tampered image in Figure 3.10 using self-embedding. The reconstruction portions are shown to be visually acceptable with $\text{PSNR} > 20$ dB. However, since the recovery bits are embedded in the LSBs of the source image, the recovery results will not be correct after some common image processing operations, for example, JPEG compression and image filtering. Moreover, the method is not able to recover very fine details, for example, the license numbers in Figure 4.4(c) are still not recognizable. In the remainder of this chapter, we shall discuss another automatic restoration scheme which is robust to common image processing operations and be able to restore fine details.

4.2 Blind restoration method based on irregular sampling

This section describes an improved restoration scheme based on irregular sampling theory. After a brief review of the irregular sampling in approximation subspaces in Section 4.2.1, the newly proposed watermarking method is discussed in Section 4.2.2. In Sections 4.2.3 and 4.2.4, we describe the processes of image authentication and restoration. The newly proposed method also helps to minimize the probability of false alarm as described in Section 4.2.5.

(a)

(b)

(c)

Figure 4.4: Reconstruction with LSB self-embedding: (a) original image; (b) self-embedded image; (c) reconstructed image, PSNR*=21.04dB (*compared to the original image).

(a)

(b)

(c)

Figure 4.5: Restoration results using self-embedding: (a) restoration of Figure 3.10 (a), PSNR*=22.45dB; (b) restoration of Figure 3.10 (b), PSNR=21.67dB; (c) restoration of Figure 3.10 (c), PSNR=22.31dB (*averaged PSNR of the restoration of tampered blocks compared to untampered blocks).

4.2.1 Irregular sampling and reconstruction by projections onto convex sets (POCS)

In this section, we shall give a short review of the irregular sampling by projections onto convex sets (POCS), which our new proposed method is based on. The topic of this chapter is to detect tampered blocks in a watermarked image and then restore them. Assume that we have no prior knowledge of the corruption channel, the problem can be formulated as obtaining an incomplete set of data with lost packets, which can be cast as an irregular sampling problem. Marziliano *et al.* [87, 88] suggested solving the problem of reconstructing irregularly sampled signal by projection onto convex sets [89, 90] under the assumption that the signal belongs to two linear closed convex sets with non-empty intersection in a Hilbert space.

Definition 4.1 If the inner product defines a norm, then the inner product space is called a *Hilbert space* [91].

Definition 4.2 A subset l of $\mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, is said to be convex if together with any x_1 and x_2 it also contains $\mu x_1 + (1 - \mu)x_2$ for all μ , $0 \leq \mu \leq 1$.

Definition 4.3 Let S be a subset of V , where V is a vector space. The *closure* of S , written $\bar{S}$, is simply S together with all of its limit points. S is said to be *closed* iff $S = \bar{S}$.

Definition 4.4 Given $f \in \mathcal{H}$, the projection $g \triangleq \mathcal{P}_i f$ of f onto the closed convex set l_i is that unique element $g \in l_i$ that satisfies

$$\inf_{x \in l_i} \|f - x\| = \|f - g\|,$$

where $\|x\| \triangleq \langle x, x \rangle^{1/2}$ is the norm in the L_2 -space.

The method of projection onto convex sets (POCS) is a recursive algorithm for finding a point in the intersection of m given closed convex sets. Suppose a known property of an unknown image is viewed as a constraint that restricts the image to lie in a well-defined closed convex set l_i , thus for m properties there are m sets $l_i, i = 1, \dots, m$, and the image must lie in the intersection

$$l_0 \triangleq \bigcap_{i=1}^m l_i. \quad (4.1)$$

The image restoration problem is then to find a point, i.e., an image that lies in l_0 . Let $\mathcal{P}_i$ denote the projector for the set l_i , let us define

$$\mathcal{P} \triangleq \mathcal{P}_m \mathcal{P}_{m-1} \cdots \mathcal{P}_1; \quad (4.2)$$

if l_0 is nonempty the POCS algorithm is given by

$$f^{(n+1)} = \mathcal{P} f^{(n)} \quad (4.3)$$

As exemplified in Figure 4.6, assume that the image lies in the intersection of two convex sets, l_a and l_b , the reconstruction of the image involves two projections: the first projection is onto l_a and the second projection is onto l_b , where $\mathcal{P} = \mathcal{P}_b \mathcal{P}_a$, and $\mathcal{P}_a$ and $\mathcal{P}_b$ are the projection operators onto the two convex sets, respectively.

To facilitate the watermarking based restoration problem, we define the following convex sets:

Convex set 1: l_a denotes the subset of $\mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, composed of all

Figure 4.6: Projections onto convex sets.

functions whose cosine transform coefficients satisfy the constraint

$$\text{sgn}(F(u, v)) = \Gamma(u, v) \quad (4.4)$$

in a prescribed region Δ of the frequency domain, with

$$\text{sgn}(y) = \begin{cases} 1 & y \geq 0 \\ 0 & y < 0 \end{cases},$$

where $F(u, v)$ is the DCT coefficient of $f(x, y)$, and $\Gamma(u, v) \in \{0, 1\}$ is a known binary function. If f_1, f_2 belong to l_a and

$$f_1 \leftrightarrow F_1(u, v)$$

$$f_2 \leftrightarrow F_2(u, v)$$

then given $0 \leq \mu \leq 1$, we have

$$\mu f_1 + (1 - \mu) f_2 \leftrightarrow \mu F_1 + (1 - \mu) F_2. \quad (4.5)$$

Since

$$\text{sgn}(\mu F_1 + (1 - \mu) F_2) = \mu \Gamma(u, v) + (1 - \mu) \Gamma(u, v) = \Gamma(u, v), \quad (4.6)$$

then l_a is convex.

The closure of l_a is obvious. Suppose $\{f_n\} \in l_a$ and $f_n \rightarrow f$, which is equivalent to

$$\lim_{n \rightarrow \infty} \|f_n - f\| = 0. \quad (4.7)$$

We then have

$$\lim_{n \rightarrow \infty} \|F_n - F\| = 0, \quad (4.8)$$

which leads to

$$\text{sgn}(F(u, v)) = \Gamma(u, v). \quad (4.9)$$

The projection of an arbitrary $f \in \mathcal{H}$ onto l_a is realized by

$$\mathcal{P}_a f \leftrightarrow \begin{cases} F(u, v) & (u, v) \in \Delta, \text{sgn}(F(u, v)) = \Gamma(u, v) \\ 0 & (u, v) \in \Delta, \text{sgn}(F(u, v)) \neq \Gamma(u, v) \\ F(u, v) & (u, v) \notin \Delta \end{cases} \quad (4.10)$$

During watermark embedding, the polarity information of the primary cosine transform coefficients is extracted, hashed into the watermark signal, and embedded into the image itself. This prior knowledge is extracted out on the detection phase and used to restore the tampered blocks.

Convex set 2: l_b denotes the subset of $\mathcal{H}$ composed of all functions f which satisfy the constraint

$$\frac{1}{|\Delta|} \iint_{\Delta} f(x, y) \, dx dy = \rho \quad (4.11)$$

for any prescribed $\rho \in R$. The projection onto l_b is realized by

$$\mathcal{P}_b f \leftrightarrow \begin{cases} f + c, & (x, y) \in \Delta \\ f, & (x, y) \notin \Delta \end{cases}, \quad (4.12)$$

where the real constant c is chosen so that

$$\frac{1}{|\Delta|} \iint_{\Delta} (f(x, y) + c) \, dx dy = \rho. \quad (4.13)$$

In our scheme, the mean value of the tampered block is estimated from its surrounding blocks. It is readily shown that l_b is convex. If f_1, f_2 belong to l_b and given $0 \leq \mu \leq 1$, then

$$\frac{1}{|\Delta|} \iint_{\Delta} \mu f_1 + (1 - \mu) f_2 \, dx dy = \mu \rho + (1 - \mu) \rho = \rho.$$

Suppose $\{f_n\} \in l_b$ and $f_n \rightarrow f$. Schwarz's inequality yields

$$\begin{aligned} & \left| \rho - \frac{1}{|\Delta|} \iint_{\Delta} f(x, y) \, dx dy \right|^2 \\ &= \frac{1}{|\Delta|^2} \left| \iint_{\Delta} (f_n(x, y) - f(x, y)) \, dx dy \right|^2 \\ &\leq \frac{1}{|\Delta|^2} \left(\iint_{\Delta} |f_n(x, y) - f(x, y)| \, dx dy \right)^2 \\ &\leq \frac{1}{|\Delta|} \iint_{\Delta} |f_n(x, y) - f(x, y)|^2 \, dx dy \rightarrow 0; \end{aligned} \quad (4.14)$$

that is,

$$\frac{1}{|\Delta|} \iint_{\Delta} f(x, y) \, dx dy = \rho \quad (4.15)$$

and l_b is closed.

Proof of $\mathcal{P}_b$: From the definition of projection, suppose $g = \mathcal{P}_b f$ minimizes

$$I = \|f - g\|^2 = \iint_{\Delta} (f - g)^2 \, dx dy \quad (4.16)$$

while satisfies the side restriction

$$\frac{1}{|\Delta|} \iint_{\Delta} g(x, y) \, dx dy = \rho. \quad (4.17)$$

Let $\delta(x, y)$ denotes an impulse function whose center is on (x, y) . Obviously, with a sufficient small positive μ ,

$$\tilde{g}(x, y) = g(x, y) - \mu\delta(x_0, y_0) + \mu\delta(x_1, y_1) \quad (4.18)$$

also satisfies Equation (4.17). Moreover, the value $\tilde{I}$ obtained by substituting $\tilde{g}$ for g in Equation (4.16) is given by

$$\begin{aligned} \tilde{I} = I &+ \iint_{\Delta} 2(f-g) (\mu\delta(x_0, y_0) - \mu\delta(x_1, y_1)) \, dx dy \\ &+ \iint_{\Delta} (\mu\delta(x_0, y_0) - \mu\delta(x_1, y_1))^2 \, dx dy. \end{aligned} \quad (4.19)$$

Since $I \leq \tilde{I}$ and $\mu > 0$,

$$\lim_{\mu \rightarrow 0} \frac{\tilde{I} - I}{\mu} = \iint_{\Delta} 2(f-g)(\delta(x_0, y_0) - \delta(x_1, y_1)) \, dx dy \geq 0. \quad (4.20)$$

So that we yield

$$f(x_0, y_0) - g(x_0, y_0) \geq f(x_1, y_1) - g(x_1, y_1). \quad (4.21)$$

But the argument is reversible, whence,

$$f(x_1, y_1) - g(x_1, y_1) \geq f(x_0, y_0) - g(x_0, y_0) \quad (4.22)$$

and so that

$$f(x_0, y_0) - g(x_0, y_0) = f(x_1, y_1) - g(x_1, y_1). \quad (4.23)$$

In short, for any $(x, y) \in \Delta$, there exists a constant c such that

$$g(x, y) = f(x, y) + c, \quad (4.24)$$

where the constant c is chosen to satisfy the requirement

$$\rho = \frac{1}{|\Delta|} \iint_{\Delta} (f(x, y) + c) \, dx dy. \quad \blacksquare \quad (4.25)$$

Convex set 3: l_c denotes the set of all functions in $\mathcal{H}$ which assume prescribed values Θ

over a closed region Δ . The projection onto l_c is realized by

$$\mathcal{P}_c f = \begin{cases} \Theta(x, y) & (x, y) \in \Delta \\ f(x, y) & (x, y) \notin \Delta \end{cases}. \quad (4.26)$$

Here, $\Theta(x, y)$ are the values of the known samples and the convexity and closure of the l_c can be readily observed.

Let us define the composite projection operation

$$\mathcal{P} = \mathcal{P}_c \mathcal{P}_b \mathcal{P}_a \quad (4.27)$$

then the irregularly sampled signal can be restored through the iteration

$$f^{(i+1)} = \mathcal{P} f^{(i)}. \quad (4.28)$$

4.2.2 Watermark embedding

The watermark embedding process begins with dividing the original image into sub-blocks of size $N \times N$. These sub-blocks are then grouped into macro-blocks which contain $M \times M$ sub-blocks in raster-scanned order. Our authentication is based on sub-blocks while the restoration is based on macro-blocks. In the following, the words “block” and “sub-block” will be used alternatively.

Consider a macro-block $\mathbf{X}_\mu$, it is DCT transformed and the coefficients are re-ordered by zigzag scanning. The first $L \cdot M^2$ coefficients are kept and denoted as $\mathbf{d}_\mu$, where L is the number of watermark bits to be embedded into every $N \times N$ sub-block. For example, $L = 6$, and $M = 3$, then the length of $\mathbf{d}_\mu$ is 54. The polarity information, $\mathbf{p}_\mu$, is generated by

$$p_\mu(k) = \begin{cases} 1 & d_\mu(k) \geq 0 \\ 0 & d_\mu(k) < 0 \end{cases}, \quad (4.29)$$

where $k = 0, 1, \dots, L \cdot M^2 - 1$. The macro-blocks are then formed into pairs using a pre-determined mapping function Ω . Suppose $\Omega(\mu) = \nu$, then $\mathbf{p}_\mu$, the polarity information of macro-block $\mathbf{X}_\mu$ is to be embedded into $\mathbf{X}_\nu$, and $\mathbf{p}_\nu$ is to be embedded into $\mathbf{X}_\mu$, with L bits into each $N \times N$ block. A pseudo-random binary signal $\mathbf{h}$ is generated and its initial state number is contained as part of the secret key file $\mathcal{K}$. The watermark signal $\mathbf{w}_\mu$ is then obtained by XOR-ing the pseudo-random signal with the polarity information:

$$\mathbf{w}_\mu = \mathbf{h} \oplus \mathbf{p}_\mu. \quad (4.30)$$

The watermark is partitioned into M^2 parts, and each part is embedded into individual sub-blocks.

The watermark embedding process for one sub-block is described in Figure 4.7. Consider an $N \times N$ block $\mathbf{x}$, it is first decomposed into two fields¹, the boundary field $\mathbf{x}^b$ and the pinned field $\mathbf{x}^p$, which can be described as:

$$\mathbf{x} = \mathbf{x}^b + \mathbf{x}^p. \quad (4.31)$$

¹Refer to Section 3.2.1 for the specific process.

Figure 4.7: Blind restoration based on irregular sampling: Watermark embedding process for one sub-block.

Next, we perform the sine transform to the pinned field block as follows:

$$\mathbf{x}^{p(s)} = \mathbf{S}_N \mathbf{x}^p \mathbf{S}_N^T. \quad (4.32)$$

In the middle to high frequency bands of $\mathbf{x}^{p(s)}$, we select, according to the length of the watermark signal, L coefficients for watermarking modulation. The watermarked sine transform coefficients $\hat{\mathbf{x}}^{p(s)}$ is obtained as described in Algorithm 3.1.

The watermarked pinned field block $\hat{\mathbf{x}}^p$ is obtained by the inverse 2-D sine transform:

$$\hat{\mathbf{x}}^p = \mathbf{S}_N^T \hat{\mathbf{x}}^{p(s)} \mathbf{S}_N \quad (4.33)$$

and a watermarked block is therefore achieved by

$$\hat{\mathbf{x}} = \mathbf{x}^b + \hat{\mathbf{x}}^p. \quad (4.34)$$

Figure 4.8: Blind restoration based on irregular sampling: Watermark detection, image authentication and restoration for one sub-block.

4.2.3 Watermark detection and image authentication

The watermark detection and image authentication processes for one sub-block is illustrated in Figure 4.8. The detection system receives as input a watermarked and possibly tampered image. Similar to the watermarking process, the polarity information is extracted from every macro-block and is partitioned into M^2 parts, with every part corresponding to one sub-block in its paired macro-block. Here, we assume that the pre-determined mapping function Ω is known to both encoder and decoder. Since we limit ourselves to the situations in which tampering is only in the form of content modification, the synchronization issue after geometrical attacks is not considered here.

The embedded watermark $\tilde{w}$ is extracted from every sub-block using Algorithm 3.2. We then XOR the extracted watermark with the corresponding part of the polarity information of its paired macro-block:

$$\tilde{h} = \tilde{w} \oplus \tilde{p}. \quad (4.35)$$

The original pseudo-random signal $\mathbf{h}$ is also generated using the initial state number in $\mathcal{K}$. The bits in $\check{\mathbf{h}}$ and $\mathbf{h}$ are then compared by the normalized cross correlation function ρ , whose value lies in $[-1, 1]$. Assume γ is a properly set threshold, the block is considered to be maliciously tampered if $\rho < \gamma$. The threshold is determined mathematically or experimentally so as to maximize the probability of tampering detection subject to a given probability of false alarm.

4.2.4 Content restoration

If some parts of the watermarked image were detected to be removed or destroyed, they would be automatically restored using the method described in Section 4.2.1. While our authentication is based on sub-blocks, the restoration is based on the macro-blocks. The macro-block containing tampered blocks is viewed as an irregular sampled signal with lost samples on the locations of the tampered blocks. The tampered blocks are then restored using the following algorithm:

Algorithm 4.1 *Restoration of tampered blocks*

$$\text{Init } \mathbf{X}^{(0)} = \mathcal{P}_0 \mathbf{X}$$

$$\mathbf{X}^{(i+1)} = \mathcal{P} \mathbf{X}^{(i)}$$

where the projection operator $\mathcal{P}$ is as that defined in Equation (4.27), and $\mathcal{P}_0$ is defined by

$$\mathcal{P}_0 \mathbf{y} = \begin{cases} 0 & \text{if } n \in \Delta \\ y(n) & \text{if } n \notin \Delta \end{cases} \quad (4.36)$$

with Δ denoting the tampered sub-blocks. The polarity information, $\Gamma(u, v)$, has been extracted as described in the Section 4.2.3 from the paired macro-block of the tampered macro-block. The mean value of the tampered block is estimated by the mean value of the un-tampered blocks in the same macro-block.

4.2.5 Minimization of the probability of false alarm

In practical implementation, the DCT domain polarity information extracted from the macro-blocks may change after its contained sub-blocks are watermarked. This in turn leads to false alarm during image authentication. False alarms also occur when images are subjected to lossy transcoding or other common image processing operations which are viewed legitimate as they preserve the content of the image. Therefore, two compensative schemes are developed to minimize the probability of false alarm. Firstly, a detection process is performed immediately after the embedding. If false alarm occurs on any block, an iterative embedding procedure is performed on those blocks based on the newly extracted polarity information.

The second solution is to be executed in the authentication process. When the system detects “wrong” blocks, a verification process is activated before outputting the final tampered block alarm. The verification process is a variation of the restoration process:

Algorithm 4.2 *Verification of tampered blocks*

$$Init \quad \mathbf{X}^{(0)} = \mathbf{X}$$

$$\mathbf{X}^{(i+1)} = \mathcal{P}\mathbf{X}^{(i)}.$$

This iteration procedure is performed on the detected “wrong” block for η times and the restored result is compared with the input test image, where η is a small positive integer and it was heuristically set to be 5 in our experiment. In a false alarm case, that is, the content of the block actually has not been modified, the initial guess which is input to Algorithm 4.2 is then approximately the same as the true block. As such, the iteration process would converge very fast. The verification process would indicate a false alarm if the PSNR of the restored image compared to the input test image is larger than a threshold T_v after η iterations.

4.3 Performance analysis of the proposed blind restoration method

In this section, we analyze the performance of the proposed hybrid watermarking and restoration system. The system performance is evaluated from three aspects: the probability of false alarm, the self-restoration capability and computational complexity.

4.3.1 Probability of false alarm

Semi-fragile watermarking is designed to detect illegitimate manipulations. It should be able to survive acceptable manipulations based on a pre-determined acceptable level of probability of false alarm. After a watermarked image goes through a legitimate image processing operation, a false alarm may occur when either the watermark bits extracted from a sub-block are not correct or the polarity information extracted from its paired macro-block has changed. For performance estimation, we assume that for the latter situation, all the watermark signal embedded in the contained blocks of its paired macro-block has been destroyed. Consider a watermarked coefficient $\hat{x}^{p(s)}(k)$ which has been tampered with, as follows:

$$\tilde{x}^{p(s)}(k) = \hat{x}^{p(s)}(k) + n(k), \quad (4.37)$$

with $n(k)$ assumed to be Gaussian distributed with zero mean and variance σ^2 . The probability that the watermark does not survive the tampering is given as:

$$P_{FA}(k) = P\{\text{sgn}(\tilde{x}^{p(s)}(k)) \neq \text{sgn}(\hat{x}^{p(s)}(k))\}. \quad (4.38)$$

Then the probability of false alarm for a sub-block is:

$$P_{FA} = (m^2 + 1) \prod_{k=1}^{\ell} \left[\frac{1}{2} \text{erfc} \left(\frac{|\hat{x}^{p(s)}(k)|}{\sqrt{2}\sigma} \right) \right] \quad (4.39)$$

$$\leq (m^2 + 1) \left[\frac{1}{2} \text{erfc} \left(\frac{\lambda}{2\sqrt{2}\sigma} \right) \right]^{\ell}.$$

where $\text{erfc}(\cdot)$ is the complementary error function defined as [92]

$$\text{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-t^2} dt.$$

We observe that P_{FA} is closely related to the length and the strength of the watermark signal. Watermark signal with longer length and stronger strength leads to a lower probability of false alarm, however, with increasing degradation in the watermarked image. There is a tradeoff between the system performance and the quality of the watermarked image. For example, if we set $N = 8$, $M = 3$, $L = 6$ and $\lambda = 10$, then the upper bound of P_{FA} is 0.0418 for $\sigma = 20$, that is, a distorted image with PSNR of approximately 22 dB compared to the watermarked image. Consider the 512×512 *Traffic* image shown in Figure 4.4(a), based on the above parameter setting, the PSNR of the watermarked image compared to the original image is approximately 35 dB. If we increase the parameters to be $L = 8$ and $\lambda = 20$, we obtain a lower probability of false alarm of $P_{FA} < 0.0001$, but the PSNR of the watermarked image is degraded to 31 dB. However, if we adopt the verification algorithm as described in Section 4.2.5 and use the tolerance bound γ during authentication, the probability of false alarm will decrease significantly as illustrated by our experimental results in Section 4.4.

4.3.2 Evaluation of restoration capability

The self-restoration capability is evaluated from both the quality of the restored image and the speed of convergence of the reconstruction algorithm. A performance analysis of Algorithm 4.1 on a database of 1,000 natural images [79] is shown in Figure 4.9. During the evaluation, we set $N = 8$, $M = 3$ and $L = 6$, which ensure an acceptable tradeoff between the restoration performance and the visual quality of the watermarked images. After the

Figure 4.9: Restoration performance of Algorithm 4.1.

images are watermarked as described in Section 4.2.2, each sub-block was set to be null and restored based on Algorithm 4.1. We group the tampered sub-blocks into three types: the blocks inside a macro-block, those on the boundary of a macro-block and those on the four corners of a macro-block as illustrated in Figure 4.10. An average result is reported for each group. We notice that the restoration process is insensitive to the block's location within the macro-block. Moreover, the PSNR of the reconstructed block converged to a satisfactory value ($> 27\text{dB}$) after approximately 50 iterations.

4.3.3 Computational complexity

The computation overhead of the proposed algorithm can be partitioned into three parts: watermark generation, watermark embedding and detection, and restoration operations.

Figure 4.10: Different locations of tampered sub-blocks within a macro-block with $M = 3$.

Since the computation overheads of the first two operations are negligible compared to that of the restoration operations, in this section we will focus on analyzing the computational complexity of the restoration operations, which depends on the required number of iterations.

The restoration operations include two parts: verification of tampered blocks and restoration of tampered blocks. As discussed in the previous section, for restoration of tampered blocks, the restoration algorithm converges to satisfactory results after approximately 50 iterations. The computation overhead of the verification of tampered blocks, however, with a fixed number of iterations η , also depends on the number of blocks which report an initial false alarm. The iteration number for verification of tampered blocks can be expressed as

$$n = \eta \cdot \left(\frac{L_1 \times L_2}{N^2} \right) \cdot P_{FA}, \quad (4.40)$$

with P_{FA} bounded as in Equation (4.39) and $L_1 \times L_2$ is the size of the image. In our simulation, the system took approximately 3 seconds to achieve a correct authentication

Figure 4.11: The testing images: (a) *Traffic*; (b) *Chart*.

result for a 512×512 watermarked image which had been JPEG compressed with a quality factor of 75. The simulation was performed in MATLAB on a PC with Pentium 4 2.26 GHz CPU and 256 MB RAM.

4.4 Simulation results of the proposed blind restoration method

In this section, the performance of the proposed method is evaluated, and comparison with existing techniques is also presented. The 256×256 gray-scale images *Traffic* and *Chart* as shown in Figure 4.11 were used to test our method. In our experiment, the parameters were set as follows: $n = 8$, $m = 3$, $\ell = 6$, $\lambda = 10$, $\gamma = 0.5$ and $T_v = 20$ dB.

Figure 4.12: The watermarked images: (a) *Traffic*; (b) *Chart*.

Image	The proposed method	Fridrich 1 (1 LSB)	Fridrich 1 (2 LSBs)	Fridrich 2 (differential coding)
<i>Traffic</i>	35.55	51.06	43.91	31.57
<i>Chart</i>	38.01	54.02	46.65	29.23

Table 4.1: Comparison of the PSNRs (dB) of the watermarked images.

4.4.1 The quality of watermarked images

Figure 4.12 displays the watermarked images. We can see that the watermarked images look identical to the original image. Table 4.1 shows the comparison of PSNRs of the watermarked images with those of Fridrich methods [29] (LSB embedding methods using 1 bit and 2 bits respectively, and the differential coding method). As expected, the proposed method introduced relatively more degradation than the LSB methods, however, the quality is far better than the differential coding method.

Figure 4.13: Simulation results: (a) the tampered images; (b) authentication results.

4.4.2 Authentication and Restoration

In the watermarked image Figure 4.12(a), the licence numbers were maliciously modified. As illustrated in Figure 4.13(a), modifications with different area sizes were performed. The authentication and restoration results are shown in the Figure 4.13(b) and Figure 4.14(a), respectively. We observe that the tampered blocks were accurately detected and the restored results were visually acceptable. Figures 4.14(b)-(d) show the restoration results by Fridrich methods for a comparison, and Table 4.2 show the PSNRs of the restored portions. We observe that the proposed method out-performed the LSB methods. The differential coding method obtained better restoration results than the proposed method here, however, as shown in Table 4.1, the visual quality of the watermarked images resulted

(a)

(b)

(c)

(d)

Figure 4.14: Simulation results: (a) restored images by the proposed method; (b) restored images by Fridrich method 1 (1 LSB); (c) restored images by Fridrich method 1 (2 LSBs); (d) restored images by Fridrich method 2.

Attacks	The proposed method	Fridrich 1 (1 LSB)	Fridrich 1 (2 LSBs)	Fridrich 2 (differential coding)
Col. 1	22.80	17.70	21.50	28.82
Col. 2	20.56	17.41	21.01	29.00
Col. 3	15.98	17.23	20.14	28.99

Table 4.2: Comparison of the PSNRs (dB) of the restored portions in Figure 4.14.

JPEG compression		Other attacks	
Quality	P_{FA}	Attacks	P_{FA}
$QF=90$	0.0000	<i>Gaussian filtering</i>	0.0000
$QF=70$	0.0000	<i>Unsharpening</i>	0.0000
$QF=50$	0.0069	<i>Contrast enhancement</i>	0.0000
$QF=30$	0.0127	<i>Salt & pepper noise</i>	0.0054
$QF=10$	0.0138	<i>Median filtering</i>	0.0090

Table 4.3: Probability of false alarm of the authentication process.

from differential coding was not satisfactory. We also found that, in our proposed method, the quality of the restored image degrades when the area of the tampered portion becomes larger. This problem can be solved by increasing the size of macro-blocks, although this will also increase the computation overhead.

To test the probabilities of false alarm of the authentication caused by various image processing operations, after modification of contents, the images were further degraded by other acceptable manipulations. Table 4.3 lists testing results obtained on an image database of 1,000 images. It can be observed that our scheme maintains a low probability of $P_{FA} < 0.01$ for JPEG compression with $QF \geq 50$ and other processing attacks including the median filtering, which is considered to be very difficult for the successful detection of watermarks. Figures 4.15 and 4.16 shows the restoration results after both

Figure 4.15: Simulation results: (a) the tampered images with acceptable manipulations; (b) authentication results.

content alteration and JPEG compression. Some portion in Figure 4.12(b) was removed and the image was further JPEG compressed. The first column in Figures 4.15 and 4.16 illustrates the results after JPEG compression with quality factor of 90%, and the second column illustrates the results after JPEG compression with quality factor of 80%. The LSB embedding methods failed to restore the removed parts. The differential coding method survived the 90% JPEG compression, however, failed when the quality factor decreased to 80%. The proposed approach achieved acceptable result with PSNR of approximately 16

(a)

(b)

(c)

(d)

Figure 4.16: Simulation results: (a) restored images by the proposed method; (b) restored images by Fridrich method 1 (1 LSB); (c) restored images by Fridrich method 1 (2 LSBs); (d) restored images by Fridrich method 2.

Attacks	The proposed method	Fridrich 1 (1 LSB)	Fridrich 1 (2 LSBs)	Fridrich 2 (differential coding)
JPEG (QF=90)	16.21	6.94	6.83	11.33
JPEG (QF=80)	16.19	6.35	6.42	4.66

Table 4.4: Comparison of the PSNRs (dB) of the restored portions in Figure 4.16.

JPEG compression				
Quality	PSNR of Restored Blocks (dB)*			
	The proposed method	Fridrich 1 (1 LSB)	Fridrich 1 (2 LSBs)	Fridrich 2 (differential)
<i>QF=90</i>	23.94	5.30	5.67	14.4
<i>QF=70</i>	22.75	5.48	5.73	7.49
<i>QF=50</i>	22.56	5.13	5.66	7.42
<i>QF=30</i>	19.27	5.13	5.76	7.42
<i>QF=10</i>	13.65	4.81	5.04	7.95

Other attacks				
Attacks	PSNR of Restored Blocks (dB)*			
	The proposed method	Fridrich 1 (1 LSB)	Fridrich 1 (2 LSBs)	Fridrich 2 (differential)
<i>Gaussian filtering</i>	22.62	5.99	5.94	8.71
<i>Unsharpening</i>	20.03	5.00	5.17	8.04
<i>Contrast enhancement</i>	22.63	3.54	3.50	8.43
<i>Salt & pepper noise</i>	20.87	19.01	17.17	14.59
<i>Median filtering</i>	12.5	8.05	7.48	8.78

Table 4.5: Robustness of the self-restoration (*compared with the original unwatermarked image).

dB. The PSNRs of the restored parts are listed in Table 4.4.

A benchmark evaluation of the robustness of restoration using 1,000 images was also performed and the results are reported in Table 4.5. As illustrated in Table 4.5, for the proposed approach, satisfactory results (e.g., PSNR of approximately 20 dB) are assured, except for the extreme situations of JPEG compression with $QF = 10$ and median filtering.

In all the cases, the proposed method obtained far better results than Fridrich's methods.

4.5 Summary

This chapter describes the concept of self-embedding strategy and its incorporation into PST based semi-fragile watermarking scheme for automatic restoration. Experimental results were shown and the limitation was pointed out. An improved method was then suggested for automatic image content authentication and restoration using irregular sampling. The problem of restoration of digital images was expressed as an irregular sampling problem in discrete cosine transform subspace. The tampered image can be reconstructed through iterative projections onto convex sets. Prior knowledge was hashed into the watermark signal and embedded into the pinned field of PST of the original image, containing the texture information of the original image. A new scheme which further minimized the probability of false alarm of semi-fragile watermarking caused by legitimate operations was also suggested. A low probability of false alarm of $P_{FA} < 0.01$ was achieved after the image was JPEG compressed with $QF > 50$ or subject to other common processing attacks. Experimental results showed that accurate authentication and restoration with PSNR of approximately 20 dB were still possible after lossy transcoding and other common image processing operations such as filtering.

Chapter 5

Robust watermarking in slant transform domain

In the previous chapters, we have described the fragile and semi-fragile watermarking for authentication and restoration of images. In this chapter, we focus on robust image watermarking for copyright protection. The slant transform has been applied to many image processing applications, such as transform coding and image restoration [93, 94]. However, through our initial survey, this transform has not been reported or investigated by other researchers for digital watermarking. The slant transform was designed to match basis vectors to areas of constant luminance slope [95]. Many fast algorithms have been previously proposed for this transform [96, 97, 98]. In terms of transform coding, the slant transform is considered to be a sub-optimum orthogonal transform for energy compaction. For digital watermarking, however, the energy spread becomes a significant advantage, as

there is now a good spread of middle to higher frequencies with significant energies for robust information hiding [99, 100, 101].

In this chapter, a robust watermarking scheme in the slant transform domain is proposed. We use a small logo image as the watermark, which can serve to identify copyright protection of company trademarks or logos. The logo watermark provides more textured details and more robust against attacks than messages of other forms for industrial applications. The performance of the slant transform for digital watermarking will be evaluated based on its robustness against various Stirmark and Checkmark attacks [67, 69]. An analytical comparative study on performance results of the slant transform for image watermarking against other conventional transforms such as Hadamard transform and cosine transform will also be presented. In Section 5.1, we recall some basic notions of the slant transform and its relationship to the Walsh-Hadamard transform. Our proposed watermarking method is then elaborated in Section 5.2. In Section 5.3, we discuss the optimization of data integrity based on human perception. Experimental results and analysis are presented in Section 5.4, which include a comparison of the slant transform to the Hadamard transform [63, 64] and the discrete cosine transform.

5.1 The slant transform

The slant transform was introduced to image coding by Enomoto and Shibata [102] in 1971 and developed by Pratt *et al.* [93, 94]. In this section, we give a brief review of the slant

transform representation of image data, which is used in the watermarking embedding and detection processes. The relationship between the slant transform and the Walsh-Hadamard transform is also presented.

Let $\mathbf{U}$ be the original image of size $N \times N$, its 2-D slant transform is given by

$$\mathbf{V} = \mathbf{S}_N \mathbf{U} \mathbf{S}_N^T \quad (5.1)$$

where $\mathbf{S}_N$ is the $N \times N$ unitary slant matrix. The inverse transformation to recover $\mathbf{U}$ from the transform components matrix $\mathbf{V}$ is given by

$$\mathbf{U} = \mathbf{S}_N^T \mathbf{V} \mathbf{S}_N. \quad (5.2)$$

Figure 5.2 contains the slant transform displays of the two original images of Figure 5.1. The logarithm of the absolute value of each transform coefficient is displayed rather than the absolute value itself in order to reduce the dynamic range of the display. The typical energy distribution of the slant transform of an image can be observed from these pictures. The slant transform is a member of the orthogonal transforms. It has a constant function for the first row, and has a second row which is a linear (slant) function of the column index. The matrices are formed by an iterative construction that exhibits the matrices as products of sparse matrices, which in turn leads to a fast transform algorithm.

The slant transform matrix of order-2 is given by

$$\mathbf{S}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (5.3)$$

Figure 5.1: Original images: (a) *Lena*; (b) *Singapore*.

Figure 5.2: Slant transform domain logarithm of magnitude display: (a) *Lena*; (b) *Singapore*.

The slant matrix of order-4 is obtained by the operation

$$\mathbf{S}_4 = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ a_4 & b_4 & -a_4 & b_4 \\ \hline 0 & 1 & 0 & -1 \\ -b_4 & a_4 & b_4 & a_4 \end{array} \right] \bullet \left[\begin{array}{c|c} \mathbf{S}_2 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{S}_2 \end{array} \right] \quad (5.4)$$

where a_4 and b_4 are scaling constants. The orthonormality conditions lead to:

$$a_4 = 2b_4 \text{ and } b_4 = \frac{1}{\sqrt{5}}. \quad (5.5)$$

Equation (5.4) can be generalized to give the slant matrix of order- N in terms of the slant matrix of order- $N/2$ by the following recursive relation:

$$\mathbf{S}_N = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|cc|cc} 1 & 0 & & & 1 & 0 \\ a_N & b_N & & & -a_N & b_N \\ \hline & & \mathbf{0} & & & \\ \hline \mathbf{0} & \mathbf{I}_{N/2-2} & & & \mathbf{0} & \mathbf{I}_{N/2-2} \\ \hline 0 & 1 & & & 0 & -1 \\ -b_N & a_N & & & b_N & a_N \\ \hline & & \mathbf{0} & & & \\ \hline \mathbf{0} & \mathbf{I}_{N/2-2} & & & \mathbf{0} & -\mathbf{I}_{N/2-2} \end{array} \right] \bullet \left[\begin{array}{c|c} \mathbf{S}_{N/2} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{S}_{N/2} \end{array} \right]. \quad (5.6)$$

The matrix $\mathbf{I}_{N/2-2}$ is the identity matrix of dimension $N/2 - 2$. The constants a_N, b_N , may be computed by the formulas

$$a_{2N} = \left(\frac{3N^2}{4N^2 - 1} \right)^{1/2} \text{ and } b_{2N} = \left(\frac{N^2 - 1}{4N^2 - 1} \right)^{1/2}. \quad (5.7)$$

Figure 5.3: Slant transform basis vectors of order-8.

Figure 5.3 displays a plot of the slant transform basis vectors of S_8 represented as waveforms. Wang [96] proved that the slant transform if approached by a series of steps could gradually change the transform to a Hadamard transform¹. This demonstrates the close relationship between the slant transform and the Walsh-Hadamard transform. From our study in later sections, we can see that the slant transform and the Hadamard transform display similar properties when applied to digital watermarking.

¹The details are described in Appendix B.

(a)

(b)

Figure 5.4: Robust watermarking in slant transform domain: (a) watermark embedding; (b) watermark detection.

5.2 Watermarking in slant transform domain

The slant transform image watermarking scheme and detection scheme are illustrated in Figures 5.4(a) and 5.4(b). In our scheme, a visually recognizable pattern is used as a watermark. It is embedded by modifying the slant transform coefficients of relevant blocks of the host image. In the detection stage, the original image is not necessary, which refers to a *blind detection* process.

5.2.1 Watermarking embedding

Copyright information in the form of a trademark or logo can be created as a pattern for watermarking. In our experiment, a gray-scale image of size $M \times M$ is used as the watermark. The watermark image, $\mathbf{D}$, is first transformed into a set of slant transform coefficients by Equation (5.1). A slant transform matrix of order- M is applied to this image, and then a $M \times M$ slant transform components matrix $\mathbf{W}$ is obtained. A 1-D watermark signal W is achieved by viewing $\mathbf{W}$ column by column, we denote it by

$$W = \{w_m\} \quad (5.8)$$

where

- $m \in \mathcal{M} = \{0, 1, \dots, M^2 - 1\}$;
- w_0 is the DC component and is stored in the secret key file $\mathcal{K}$ for subsequent watermark extraction, which is based on the consideration that usually the DC coefficients are of very high values. Therefore the DC value is not suitable for imperceptible embedding;
- the rest are the AC components used for watermark embedding.

Let the original image be $\mathbf{X}$. The watermarking process begins by decomposing it into a set of non-overlapped $N \times N$ blocks. An m-sequence random number generator is used to select a certain number of blocks for watermark embedding. The initial state number of the m-sequence random number generator has been produced earlier and is contained in $\mathcal{K}$.

Figure 5.5: Embedding locations in one selected slant transform components block.

Consider a certain selected block $\mathbf{x}_{m,n}$, where m and n are the coordinate numbers for blocks, we conduct an order- N slant transform on it and denote the transformed block as $\mathbf{x}_{m,n}^{(s)}$. Let $x_i \in \mathbf{x}_{m,n}^{(s)}$ denote a certain slant transform coefficient of this selected block, where $i \in \mathcal{N} = \{1, 2, \dots, N^2\}$. Among the N^2 coefficients, L middle and high frequency coefficients are used for modulation. The coefficient selection affects the performance of the watermarking scheme significantly. The high frequency components are relatively vulnerable to compression operations, while the low frequency components must be retained for visual quality of the watermarked image. Therefore, most existing watermarking schemes choose to embed the watermark into the middle frequency band [44, 63, 64]. In our scheme, embedding locations are chosen as shown in Figure 5.5 are adopted, where $N = 8$ and $L = 16$. These locations are selected based on a tradeoff analysis between robustness and data integrity.

Suppose the labelling set of those selected coefficient is $\mathcal{S} = \{n_1, \dots, n_L\}$, the embedding formula is given by

$$y_i = \begin{cases} \alpha w_{m_i} & i \in \mathcal{S} \\ x_i & \text{otherwise} \end{cases} \quad (5.9)$$

where

- $w_{m_i} \in W$, $m_i \in \mathcal{M}' = \{1, 2, \dots, M^2 - 1\}$, is the AC watermark coefficient to be embedded into the position of x_i ;
- y_i is the corresponding watermarked coefficient;
- α denotes the watermark strength factor that controls the tradeoff between visual quality of the watermarked image and robustness of the watermarking scheme. The determination of α value will be discussed in Section 5.3.

After embedding, the original coefficient x_i is replaced by y_i , hence a new $N \times N$ matrix of slant transform coefficients of image block is obtained. The inverse slant transform is then applied to the matrix using Equation (5.2) to reconstruct the luminance matrix of the watermarked image block. After performing the watermark embedding for all the selected blocks of the original image, a watermarked image, $\mathbf{Y}$, is obtained. As indicated earlier, the secret key file $\mathcal{K}$ is required for subsequent decoding. The reason why we choose the slant transform coefficients for watermark embedding instead of transforms with better energy compaction, for example, DCT, is based on the observation that, the amplitudes of those types of transform coefficients are more uneven, which make the strength control of the embedding more complex as well as introducing additional extra overheads.

5.2.2 Watermarking detection

In the detection stage, as indicated in Figure 5.4(b), the watermarking system receives as input, the test image $\hat{\mathbf{Y}}$ and its appended secret key file $\mathcal{K}$. First, the positions of the blocks with watermark embedded are computed using the m-sequence random number generator and the initial seed value stored in $\mathcal{K}$. All the selected blocks are slant transformed and similar to the embedding process, we denote the (m, n) transformed block as

$$\hat{\mathbf{y}}_{m,n}^{(s)} = \{\hat{y}_i\} \quad (5.10)$$

where $\hat{y}_i$ denotes a certain coefficient in this block and $i \in \mathcal{N} = \{1, 2, \dots, N^2\}$. Let the retrieved watermark signal be denoted by $\hat{W}$, then the watermark extraction formula is given as follows:

$$\hat{w}_{n_i} = \begin{cases} \hat{y}_i / \alpha & i \in \mathcal{S} \\ \hat{y}_i & \text{otherwise} \end{cases} \quad (5.11)$$

where $\hat{w}_{n_i} \in \hat{W}$, $n_i \in \mathcal{M}' = \{1, 2, \dots, M^2 - 1\}$, are the AC slant transform coefficients of the retrieved watermark; and α and $\mathcal{S}$ have the same meaning as described in Section 5.2.1.

The extracted AC components and the DC component of the watermark stored in $\mathcal{K}$ are rearranged into a slant transform component matrix $\hat{\mathbf{W}}$. The extracted watermark image, $\hat{\mathbf{D}}$, is then obtained by an inverse slant transform using Equation (5.2).

5.3 Perceptually adaptive watermarking

A robust watermark is expected to survive common signal preprocessing operations. However, the improvement of robustness of watermarking systems can often be traded for good data integrity of the watermarked image. Therefore, one major concern of any robust digital watermarking system is the trade-off design between the strength of the watermark signal and the visual quality of the watermarked images.

In our scheme, a perceptual mask model is incorporated to adaptively control the perceptibility of watermarks. The strength of the watermark α is adjusted based on the local textures and edge characteristics of the source image. The perceptibility of noise varies with the texture of its background. The human visual system (HVS) is insensitive to noise in coarse textured regions compared to smooth areas. For the coarse texture areas, the distortion visibility is low. Therefore, the watermark of higher strength can be embedded into these areas.

The perceptual mask model is generated by a Canny edge detection algorithm. The Canny edge detection was selected because of its ability in detecting weak edges by using two different thresholds. It is applied to each selected block and the 7 blocks immediately surrounding the selected blocks. We then count the number of edge points in each block. Large numbers of edges points indicate highly textured regions while smaller values indicate smoother areas or areas with outstanding edges. The perceptual mask model then

involves calculation of a weighted texture score which is defined as

$$\Psi_{m,n} = \psi_{m,n} * \tilde{\psi}_{m,n} \quad (5.12)$$

where $\psi_{m,n}$ is the number of edge points of block $\mathbf{x}_{m,n}$ and $\tilde{\psi}_{m,n}$ is the average edge number of the neighborhood of 8 blocks with $\mathbf{x}_{m,n}$ as the center block. The adoption of the second item is based on the observation that the perceptibility of noise on a certain region depends on the texture of its surrounding regions, and not just depends on the texture of itself. For example, the noise in a small smooth region is less visible if this region is surrounded by textured areas. However, the noise would be obvious on a textured region if it was immediately surrounded by very smooth areas. The adoption of the second item avoids the misjudgement of such blocks.

The watermark strength factor α then can be determined as

$$\alpha_{m,n} = \beta \cdot \Psi_{m,n} \quad (5.13)$$

where β is the user defined global scaling factor. The same perceptual model needs to be calculated in both the embedding and detection sides.

Based on the above perceptual mask model, the watermark strength levels for the images in Figure 5.1 are shown in Figure 5.6. We can observe that the classification of the blocks is very consistent as compared to subjective evaluation.

Figure 5.6: The watermark strength levels: (a) strength level graph for *Lena*; (b) strength level graph for *Singapore*.

5.4 Experiments and analysis

The two gray-scale images shown in Figure 5.1 were used to test our proposed algorithm. These images contain different distinct textures: the *Lena* image is a natural image with a significant amount of smooth areas while the *Singapore* is a satellite image with many fine textures. The watermarked images are shown in Figure 5.7. Results obtained show that there are no perceptually visible degradations on the watermarked images with a PSNR of approximately 41 *dB* for *Lena* and 38 *dB* for *Singapore*, respectively. In the experiment, the parameters were set as $M = 64$, $N = 8$, $L = 16$ and $\beta = 0.6$.

Figure 5.7: The watermarked images: (a) *Lena*; (b) *Singapore*.

5.4.1 The robustness of the proposed scheme

The slant transform watermarking scheme is benchmarked using Stirmark 4.0 [67] and Checkmark 1.2 [69]. These are the most commonly used benchmarking tools in digital watermarking. Sample results of the above two images are reported in Table 5.1 and Table 5.2, respectively. The quality of the retrieved watermark is evaluated by its normalized correlation coefficient ρ with the original watermark, where $\rho \in [-1, 1]$.

We can see that the slant transform watermarking is relatively robust to additive noise and some common image processing techniques such as compression and convolution filtering, with most of the values of $\rho > 0.8$. It can also be found that most of the correlation coefficients in Table 5.2 are smaller than those in Table 5.1. This is due to the fact that there are more AC energies concentrating in the high frequency components in highly textured images than smooth images. Hence, textured images, such as *Singapore*, suffer more from

Benchmarking operations	Extracted watermark	Correlation coefficient
Hard thresholding		$\rho = 0.9646$
Soft thresholding		$\rho = 0.9076$
Gaussian filtering		$\rho = 0.9640$
Wiener filtering		$\rho = 0.9500$
JPEG compression		$\rho = 0.9273$
Wavelet compression		$\rho = 0.9446$
Sharpening		$\rho = 0.9529$
Additive noise		$\rho = 0.9514$

Table 5.1: Sample benchmarking results for *Lena* image.

Benchmarking operations	Extracted watermark	Correlations coefficient
Hard thresholding		$\rho = 0.8883$
Soft thresholding		$\rho = 0.7552$
Gaussian filtering		$\rho = 0.9410$
Wiener filtering		$\rho = 0.8122$
JPEG compression		$\rho = 0.9080$
Wavelet compression		$\rho = 0.8553$
Sharpening		$\rho = 0.9106$
Additive noise		$\rho = 0.9595$

Table 5.2: Sample benchmarking results for *Singapore* image.

common image processing operations than smooth images such as *Lena*. Watermarking robustness is therefore more an issue for textured images though it is commonly recognized that highly textured images provide more capacity for data hiding without introducing too much degradation to the visual quality of original images.

5.4.2 The determination of threshold

The extracted watermark image is compared to the original watermarked image based on correlation coefficient, and the watermark is judged to be exist if the correlation coefficient is higher than a certain threshold. The threshold should be decided such that the probability of false alarm is minimum. We test the values of the correlation coefficients detected from 1,000 non-watermarked images using random keys, and the results are shown in Figure 5.8. We find the values sharply clustered in the small interval around the zero point. Based on this observation, we set the threshold as 0.15 or above without introducing any false alarm.

5.4.3 Comparison of transforms

In Section 5.1, we have mentioned that the slant transform has a close relationship and similar properties to the Hadamard transform. Ho *et al.* [63, 64] reported some research work associated with the fast Hadamard transform for robust watermark embedding and retrieval of images and characters. Their results suggested that the Hadamard transform based scheme was superior to those based on DCT when the processing noise was high. We conducted an analytical comparison on the robustness of the slant transform to the

Figure 5.8: The distribution of the values of the correlation coefficient when detected from non-watermarked images.

Hadamard transform [63, 64] as well as the DCT against compression attacks. The results are presented in Figures 5.9 and 5.10. Figure 5.9 shows the robustness comparison of the three transforms against JPEG (DCT) compression. The results against JPEG-2000 (wavelet) compression are depicted in Figure 5.10. The reported results are averaged results based on 1,000 testing images [79]. For all the other transforms, the watermarks were also embedded into the middle frequency bands, and the comparison is based on the same PSNR values of $40dB$ for all the images. As expected, we find the slant transform and Hadamard transform have similar performances in terms of robustness against compression; however, the slant transform scheme gives better results in terms of robustness against compression over the other two transforms. As indicated in the graphs, the slant transform survives compression far better than the DCT. The enhanced robustness is due to the fact that there

Figure 5.9: Performance comparisons between slant transform, Hadamard transform and discrete cosine transform watermarking against JPEG (DCT) compression.

Figure 5.10: Performance comparisons between slant transform, Hadamard transform and discrete cosine transform watermarking against JPEG-2000 (wavelet) compression.

are more useful middle frequency bands in the slant transform. Furthermore, energies in the middle to high frequency bands of the cosine transform may spread to more middle or low frequency bands in the slant transform. As such, while some middle to high frequency bands in the DCT domain might be completely eliminated in the JPEG compression, there would always be some slant transform bands remained intact, which in turn leads to the better reservation of the watermark energies in the slant transform watermarking.

It is also interesting to note that the advantage of the slant transform is much more significant in more textured images. Since the slant transform is designed to match basis vectors to areas of constant luminance slope, it is good for compact energy in “smooth” images. However, in terms of watermarking, it works better for “textured” images, for the energy spread becomes a significant advantage in watermarking and offers a good spread of middle to higher frequencies with significant energies for robust information hiding. For the above reasons, the slant transform watermarking can be exploited for the copyright protection of complex textured images, such as satellite images [99].

Unfortunately, the proposed scheme cannot survive the geometrical operations in Stir-mark, such as down scaling, rotation and translation. As such the Slant transform is not suitable for consideration for RST invariant applications.

5.5 Summary

We presented a robust watermarking technique for embedding grayscale logo image watermark into a carrier image based on the slant transform. The slant transform has more useful middle and high frequency bands than several high gain transforms such as discrete cosine transform. Therefore, it was more robust than DCT domain algorithm under various attacks. Comparison results between the slant transform, the Hadamard transform and the discrete cosine transform were presented. In particular, after JPEG compression, the probability of detection of the slant transform watermark obtained a significant improvement over the Hadamard and the DCT. The advantage of the slant transform watermarking is more significant for images with complex textures such as satellite images. Possible applications of this technique include copyright protection and ownership identification.

Chapter 6

Summary of the thesis

This chapter summarizes the thesis. We provide conclusions of our work as well as the proposal for future works.

6.1 Conclusions

Several novel watermarking schemes are presented in this thesis. Their applications include image content authentication, restoration and copyright protection, and so on. By theoretical analysis and simulation results we have shown their advantages over other methods presented in recent literature.

We proposed a fragile watermarking scheme based on encoding of the zeros of the z -transform for exact authentication of digital images. It was designed by exploiting the sensitivity of the positions of the zeros of the z -transform around the unit circle to any change made on the host image. Experimental results showed that the watermarking system

could localize the portions of a watermarked image that had been tampered maliciously with high accuracy and provided bit-sensitivity to any tampering on the image file, which is desirable in the literal authentication of master copies of important images. In addition, the proposed method is more secure than normal fragile watermarking techniques based on LSBs embedding.

To address the problem of selective authentication of digital images, which provides more general usage, a semi-fragile watermarking scheme using the pinned sine transform (PST) was proposed. The probability of tampering detection of the authentication system was higher than 98%. While the watermarking scheme was sensitive to any texture alteration in the watermarked image, it could survive slight legitimate operations, such as JPEG compression, which is desirable in some practical applications, such as traffic enforcement, satellite remote sensing applications and legal applications. Simulation results demonstrated that the proposed algorithm was less sensitive to legitimate image processing operations such as compression than that of the equivalent DCT scheme. The inter-relationship between blocks introduced by the PST decomposition rendered the watermarking scheme resistant to the content “cutting and pasting” attacks.

The possibility of self-restoration of detected tampered portions was also investigated. We have incorporated the self-embedding strategy into our semi-fragile watermarking scheme to facilitate the automatic restoration upon detection of tampering. The restoration result was equivalent to a low-quality compressed original and was visually acceptable. Unfortunately, this restoration could not survive any common image processing operations.

To enhance the robustness of the self-restoration, we proposed another blind restoration method based on the irregular sampling theory. When tampered blocks were detected, the problem was formulated as an irregular sampling problems in an approximation subspace. The tampered blocks were then reconstructed using a variant of the projection onto convex sets (POCS) method, incorporating the prior knowledge embedded in the same watermarked image. Simulation results showed that accurate detection and restoration with PSNR of approximately 20 dB were still possible after common image operations such as JPEG compression and image filtering. We also suggest a new scheme which further minimize the probability of false alarm of semi-fragile watermarking caused by legitimate operations. A low probability of false alarm of $P_{FA} < 0.01$ was achieved after the image was JPEG compressed with $QF > 50$ or subject to other common processing attacks.

For copyright protection and owner identification, a robust watermarking method in the slant transform domain was proposed. The slant transform method achieved better improvements in terms of robustness against some common image processing operations, relative to other methods based on conventional transforms such as Hadamard transform and discrete cosine transform. The slant transform method could survive up to 85% of the non-geometric attacks from Stirmark and Checkmark. In particular, after compression attacks, the probability of detection of the slant transform watermarking obtained an improvement over the Hadamard transform and Cosine transform. The advantage of the slant transform method was more significant for highly textured images, which usually suffer more from common image operations such as compression. Hence the slant transform watermarking

can be exploited for the copyright protection of those images with complex textures, such as satellite images.

6.2 Future works

Our future work will focus on improvement of the watermarking-based self-restoration scheme. Limitation of the proposed method is that large gaps of size more than one macro-block may not be successfully restored. However, our work can be viewed as a new perspective in unifying the watermarking, authentication and restoration problems into an integrated field that would help in improvements of the probability of false alarm and restoration. Future work may focus on addressing the problem of reconstructing tampered images through watermarking when the tampered area is large compared to the size of the macro-block. Another future area of research is the incorporation of different prior knowledge and embedding methods which may lead to better quality of the restored image and more robustness of the authentication and restoration processes.

Appendix A

Fast sine transform via fast Fourier transform

It is possible to employ a modified FFT to calculate the coefficients of the sine transform, which are given by

$$C_p = \sqrt{\frac{2}{N+1}} \sum_{k=1}^N f_k \sin \frac{\pi p k}{N+1}, \quad p = 1, 2, \dots, N \quad (\text{A.1})$$

where the indexing has been modified to allow the sine argument to correspond with the Fourier kernel. If we carry out a DFT of length $2(N+1)$ we obtain

$$\text{DFT}(2N+2)_p = \frac{1}{\sqrt{2(N+1)}} \sum_{k=0}^{2(N+1)-1} f_k \exp \left[-j \frac{2\pi p k}{2(N+1)} \right]. \quad (\text{A.2})$$

Taking the negative imaginary part gives

$$\text{IM} (\text{DFT}(2N+2)_p) = \frac{1}{\sqrt{2(N+1)}} \sum_{k=0}^{2N-1} f_k \sin \frac{\pi p k}{N+1} \quad (\text{A.3})$$

and therefore the sine transform coefficients are

$$C_p = 2 \operatorname{Im} \left(\text{DFT}(2N+2)_p \right), \quad p = 1, 2, \dots, N \quad (\text{A.4})$$

where $f_k = 0$ for $k = 0$ and $k \geq N + 1$.

The coefficients of 2-D sine transform can be obtained by performing column-transform and row-transform separately using the above method.

Appendix B

Composing slant matrices from Hadamard matrices

The matrix of the slant transform may be composed by multiplying a series of sparse matrices on the Hadamard matrix [96].

Let $\mathbf{S}_N^*$ denote the slant matrix of order N whose vectors are in the Hadamard sequence. Then $\mathbf{S}_2^*$ is given by

$$\mathbf{S}_2^* = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (\text{B.1})$$

The slant matrix $\mathbf{S}_4^*$ is obtained by

$$\mathbf{S}_4^* = \frac{1}{\sqrt{2}} \left[\begin{array}{c|cc|c} 1 & & & \\ \hline & a_4 & -b_4 & \\ \hline & b_4 & a_4 & \\ \hline & & & 1 \end{array} \right] \bullet \left[\begin{array}{c|c} \mathbf{S}_2^* & \mathbf{S}_2^* \\ \hline \mathbf{S}_2^* & -\mathbf{S}_2^* \end{array} \right] \quad (\text{B.2})$$

where

$$a_4 = 2b_4 \text{ and } b_4 = \frac{1}{\sqrt{5}}. \quad (\text{B.3})$$

The slant matrix $\mathbf{S}_{2N}^*$ is obtained similarly;

$$\mathbf{S}_{2N}^* = \frac{1}{\sqrt{2}} \left[\begin{array}{c|cc|c} \mathbf{I}_{N/2} & & & \\ \hline & a_{2N} & 1 & -b_{2N} \\ & & \ddots & \\ & & & 1 \\ \hline & b_{2N} & a_{2N} & 1 \\ & & & \ddots \\ & & & & 1 \\ \hline & & & & \mathbf{I}_{N/2} \end{array} \right] \bullet \left[\begin{array}{c|c} \mathbf{S}_N^* & \mathbf{S}_N^* \\ \hline \mathbf{S}_N^* & -\mathbf{S}_N^* \end{array} \right] \quad (\text{B.4})$$

where

$$a_{2N} = \left(\frac{3N^2}{4N^2 - 1} \right)^{1/2} \text{ and } b_{2N} = \left(\frac{N^2 - 1}{4N^2 - 1} \right)^{1/2}. \quad (\text{B.5})$$

Let us use the slant matrix $\mathbf{S}_8^*$ as an example, according to Equation (B.4),

$$\mathbf{S}_8^* = \frac{1}{\sqrt{2}} \left[\begin{array}{c|cc|c} \mathbf{I}_2 & & & \\ \hline & a_8 & 1 & -b_8 \\ & & & \\ \hline & b_8 & a_8 & 1 \\ & & & \\ \hline & & & \mathbf{I}_2 \end{array} \right] \bullet \left[\begin{array}{c|c} \mathbf{S}_4^* & \mathbf{S}_4^* \\ \hline \mathbf{S}_4^* & -\mathbf{S}_4^* \end{array} \right]. \quad (\text{B.6})$$

The last matrix may be decomposed as

$$\left[\begin{array}{c|c} \mathbf{S}_4^* & \mathbf{S}_4^* \\ \hline \mathbf{S}_4^* & -\mathbf{S}_4^* \end{array} \right] = \left[\begin{array}{c|c} \mathbf{I}_4 & \mathbf{I}_4 \\ \hline \mathbf{I}_4 & -\mathbf{I}_4 \end{array} \right] \left[\begin{array}{c|c} \mathbf{S}_4^* & \\ \hline & \mathbf{S}_4^* \end{array} \right]. \quad (\text{B.7})$$

Let $\mathbf{M}_4^*$ denote the first matrix on the right-hand side of (B.2) and $\mathbf{M}_8^*$ denote the first matrix on the right-hand side of (B.6), namely

$$\mathbf{M}_4^* = \left[\begin{array}{c|cc|c} 1 & & & \\ \hline & a_4 & -b_4 & \\ & b_4 & a_4 & \\ \hline & & & 1 \end{array} \right] \quad (\text{B.8})$$

and

$$\mathbf{M}_8^* = \left[\begin{array}{c|c|c|c} \mathbf{I}_2 & & & \\ \hline & a_8 & -b_8 & \\ & & 1 & \\ \hline & b_8 & a_8 & \\ & & & 1 \\ \hline & & & \mathbf{I}_2 \end{array} \right]. \quad (\text{B.9})$$

Then $\mathbf{S}_8^*$ becomes

$$\mathbf{S}_8^* = \frac{1}{\sqrt{2}} \mathbf{M}_8^* \left[\begin{array}{c|c} \mathbf{I}_4 & \mathbf{I}_4 \\ \hline \mathbf{I}_4 & -\mathbf{I}_4 \end{array} \right] \left[\begin{array}{c|c} \mathbf{S}_4^* & \\ \hline & \mathbf{S}_4^* \end{array} \right]. \quad (\text{B.10})$$

The last matrix may be decomposed as

$$\begin{aligned} \left[\begin{array}{c|c} \mathbf{S}_4^* & \\ \hline & \mathbf{S}_4^* \end{array} \right] &= \frac{1}{\sqrt{2}} \left[\begin{array}{c|c|c|c} 1 & & & \\ \hline & a_4 & -b_4 & \\ & b_4 & a_4 & \\ & & & 1 \\ \hline & & & \\ & & 1 & \\ & & & a_4 & -b_4 \\ & & b_4 & a_4 \\ & & & & 1 \end{array} \right] \bullet \left[\begin{array}{c|c|c|c} \mathbf{S}_2^* & \mathbf{S}_2^* & & \\ \hline \mathbf{S}_2^* & -\mathbf{S}_2^* & & \\ & & \mathbf{S}_2^* & \mathbf{S}_2^* \\ & & \mathbf{S}_2^* & -\mathbf{S}_2^* \end{array} \right] \\ &= \frac{1}{\sqrt{2}} \left[\begin{array}{c|c} \mathbf{M}_4^* & \\ \hline & \mathbf{M}_4^* \end{array} \right] \left[\begin{array}{c|c|c|c} \mathbf{S}_2^* & \mathbf{S}_2^* & & \\ \hline \mathbf{S}_2^* & -\mathbf{S}_2^* & & \\ & & \mathbf{S}_2^* & \mathbf{S}_2^* \\ & & \mathbf{S}_2^* & -\mathbf{S}_2^* \end{array} \right]. \quad (\text{B.11}) \end{aligned}$$

Since Hadamard matrices $\mathbf{H}_2$ and $\mathbf{H}_4$ are [103]

$$\mathbf{H}_2 = \mathbf{S}_2^* = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (\text{B.12})$$

and

$$\mathbf{H}_4 = \frac{1}{\sqrt{2}} \left[\begin{array}{c|c} \mathbf{H}_2 & \mathbf{H}_2 \\ \hline \mathbf{H}_2 & -\mathbf{H}_2 \end{array} \right] = \frac{1}{\sqrt{2}} \left[\begin{array}{c|c} \mathbf{S}_2^* & \mathbf{S}_2^* \\ \hline \mathbf{S}_2^* & -\mathbf{S}_2^* \end{array} \right]. \quad (\text{B.13})$$

Therefore

$$\left[\begin{array}{c|c} \mathbf{S}_4^* & \\ \hline & \mathbf{S}_4^* \end{array} \right] = \left[\begin{array}{c|c} \mathbf{M}_4^* & \\ \hline & \mathbf{M}_4^* \end{array} \right] \left[\begin{array}{c|c} \mathbf{H}_4 & \\ \hline & \mathbf{H}_4 \end{array} \right]. \quad (\text{B.14})$$

Substituting (B.14) into (B.6),

$$\mathbf{S}_8^* = \frac{1}{\sqrt{2}} \mathbf{M}_8^* \left[\begin{array}{c|c} \mathbf{I}_4 & \mathbf{I}_4 \\ \hline \mathbf{I}_4 & -\mathbf{I}_4 \end{array} \right] \left[\begin{array}{c|c} \mathbf{M}_4^* & \\ \hline & \mathbf{M}_4^* \end{array} \right] \left[\begin{array}{c|c} \mathbf{H}_4 & \\ \hline & \mathbf{H}_4 \end{array} \right]. \quad (\text{B.15})$$

But

$$\left[\begin{array}{c|c} \mathbf{I}_4 & \mathbf{I}_4 \\ \hline \mathbf{I}_4 & -\mathbf{I}_4 \end{array} \right] \left[\begin{array}{c|c} \mathbf{M}_4^* & \\ \hline & \mathbf{M}_4^* \end{array} \right] = \left[\begin{array}{c|c} \mathbf{M}_4^* & \\ \hline & \mathbf{M}_4^* \end{array} \right] \left[\begin{array}{c|c} \mathbf{I}_4 & \mathbf{I}_4 \\ \hline \mathbf{I}_4 & -\mathbf{I}_4 \end{array} \right] \quad (\text{B.16})$$

and

$$\frac{1}{\sqrt{2}} \left[\begin{array}{c|c} \mathbf{I}_4 & \mathbf{I}_4 \\ \hline \mathbf{I}_4 & -\mathbf{I}_4 \end{array} \right] \left[\begin{array}{c|c} \mathbf{H}_4 & \\ \hline & \mathbf{H}_4 \end{array} \right] = \frac{1}{\sqrt{2}} \left[\begin{array}{c|c} \mathbf{H}_4 & \mathbf{H}_4 \\ \hline \mathbf{H}_4 & -\mathbf{H}_4 \end{array} \right] = \mathbf{H}_8 \quad (\text{B.17})$$

where $\mathbf{H}_8$ is Hadamard matrix of order eight. Therefor, $\mathbf{S}_8^*$ is obtained as

$$\mathbf{S}_8^* = \mathbf{M}_8^* \cdot \mathbf{M}_{8,4}^* \cdot \mathbf{H}_8 \quad (\text{B.18})$$

where

$$\mathbf{M}_{8,4}^* = \left[\begin{array}{c|c} \mathbf{M}_4^* & \\ \hline & \mathbf{M}_4^* \end{array} \right]. \quad (\text{B.19})$$

The first subscript of $\mathbf{M}_{8,4}^*$ denotes the order of the matrix, and second subscript represents

that all the submatrices which are in the diagonal of matrix $\mathbf{M}_{8,4}^*$ are $\mathbf{M}_4^*$.

By generalizing Equation B.20, the slant transform matrix $\mathbf{S}_N^*$ may be obtained as

$$\mathbf{S}_N^* = \mathbf{M}_N^* \cdot \mathbf{M}_{N,N/2}^* \cdot \cdots \mathbf{M}_{N,4}^* \cdot \mathbf{H}_N \quad (\text{B.20})$$

where $\mathbf{H}_N$ is the Hadamard matrix of order N ,

$$\mathbf{M}_K^* = \begin{bmatrix} \mathbf{I}_{K/4} & & & \\ & a_N & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \\ & b_N & & & \\ & & a_N & & \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \\ & & & & & & \mathbf{I}_{K/4} \end{bmatrix}, \quad (K = 4, 8, 16, \dots, N) \quad (\text{B.21})$$

and

$$\mathbf{M}_{N,K}^* = \text{diagonal} \left[\underbrace{\mathbf{M}_K^*, \mathbf{M}_K^*, \dots, \mathbf{M}_K^*}_{N/K} \right] \quad \left(K = 4, 8, \dots, \frac{N}{2} \right) \quad (\text{B.22})$$

are modification transforms which change the Hadamard transform gradually into the slant transform, and a_N and b_N are determined by (B.5).

Publications

Journal papers

1. X. Zhu, A. T. S. Ho and P. Marziliano, “A new semi-fragile image watermarking with robust tampering restoration using irregular sampling,” *Signal Processing: Image Communication (SPIC)*, accepted with minor revision, Jan 2007.
2. A. T. S. Ho, X. Zhu and Y. L. Guan, “Image content authentication using pinned sine transform,” *EURASIP Journal on Applied Signal Processing (JASP)*, *Special Issue on Multimedia Security and Rights Management*, vol. 2004 (14), pp. 2174-2184, Oct. 2004.
3. A. T. S. Ho, X. Zhu and P. Marziliano, “Fragile watermarking based on encoding of the zeros of the z -transform,” *IEEE Trans. on Information Forensics and Security*, revised and resubmitted, Jun. 2006.

Conference papers

1. X. Zhu, A. T. S. Ho, and P. Marziliano, "Image Authentication and Restoration Using Irregular Sampling for Traffic Enforcement Applications," in Proc. *IEEE International Conference on Innovative Computing, Information and Control (ICICIC-06)*, Beijing, China, Aug. - Sept. 2006.
2. A. T. S. Ho, X. Zhu, B. Vrusias, and J. Armstrong, "Digital Watermarking and Authentication for Crime Scene Analysis," *IEE International Symposium for Crime Detection and Prevention (ICDP 2006)*, London, June 2006.
3. A. T. S. Ho, X. Zhu, and W. M. Woon, "A semi-fragile pinned sine transform watermarking system for content authentication of satellite Images," in Proc. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Seoul, Korea, July 2005.
4. A. T. S. Ho and X. Zhu, "Traffic image authentication based on pinned sine transform," in Proc. *IEE International Symposium on Imaging for Crime Detection and Prevention (ICDP)*, London, UK, June 2005.
5. A. T. S. Ho, X. Zhu, and J. Shen, "Authentication of biomedical images based on zero location watermarking," in Proc. *the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Kunming, China, Dec. 2004.
6. A. T. S. Ho, X. Zhu, Y. L. Guan and P. Marziliano, "Slant Transform Watermark-

PUBLICATIONS

141

ing for Textured Images,” in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Invited Session on Information Assurance and Data Hiding II*, Vancouver, Canada, May 2004.

7. X. Zhu, A. T. S. Ho, “A Slant transform watermarking for copyright protection of satellite images,” in *Proc. 4th International Conference on Information, Communications & Signal Processing / 4th IEEE Pacific-Rim Conference On Multimedia (ICICS-PCM)*, Singapore, Dec. 2003.
8. A. T. S. Ho, X. Zhu and J. Shen, “Slant transform watermarking for digital images,” in *Proc. Visual Communications and Image Processing (VCIP), SPIE, Session on Image and Video Security and Watermarking I*, Lugano, Switzerland, July 2003.

Others

1. A. T. S. Ho, X. Zhu and Y. L. Guan, “Two-source image content authentication,” *EEE Research Bulletin 2004*, Jan. 2004.

Bibliography

- [1] N. Komatsu and H. Tominaga, “A proposal on digital watermark in document image communication and its application to realizing a signature,” *Electronics and Communications in Japan*, vol. 73(5), pp. 208–218, 1990.
- [2] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Francisco, Calif, USA: Morgan Kauffman Publishers, 2001.
- [3] F. Hartung and M. Kutter, “Multimedia watermarking techniques,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079 –1107, 1999.
- [4] G. L. Friedman, “The trustworthy digital camera: restoring credibility to the photographic image,” *IEEE Trans. Consumer Electron.*, vol. 39, pp. 905–910, Nov. 1993.
- [5] ———, “Digital camera with apparatus for authentication of images produced from an image file,” *United States Patent 5,499,294*, 1996.
- [6] M. Chen, Y. He, and R. L. Lagendijk, “A fragile watermark error detection scheme

BIBLIOGRAPHY

143

- for wireless video communications,” *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 201–211, Apr. 2005.
- [7] H.-Y. S. Lin, H.-Y. M. Liao, C.-S. Lu, and J.-C. Lin, “Fragile watermarking for authenticating 3-D polygonal meshes,” *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 997–1006, Dec. 2005.
- [8] C.-T. Li, “Digital fragile watermarking scheme for authentication of JPEG images,” *IEE Proc.-Vis. Image Signal Process.*, vol. 151, no. 6, pp. 460–466, Dec. 2004.
- [9] R. G. van Schyndel, A. Z. Trikel, and C. F. Osborne, “A digital watermark,” in *IEEE Int. Conf. Image Process.*, Austin, Texas, Nov. 1994, pp. 86–90.
- [10] S. Walton, “Image authentication for a slippery new age,” *Dr. Dobbs’s Journal of Software Tools for Professional Programmers*, pp. 18–26, Apr. 1995.
- [11] M. Yeung and F. Mintzer, “An invisible watermarking technique for image verification,” in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Santa Barbara, CA, Oct. 1997, pp. 680–683.
- [12] P. W. Wong, “A watermark for image integrity and ownership verification,” in *Proc. of IS & T PIC conference*, Portland, Oregon, May 1998.
- [13] —, “A public key watermark for image verification and authentication,” in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Chicago, Illinois, Oct. 1998, pp. 455–459.

- [14] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Processing*, vol. 10, no. 10, pp. 1593–1601, Oct. 2001.
- [15] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Trans. Image Processing*, vol. 9, pp. 432–441, Mar. 2000.
- [16] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization," *IEEE Trans. Image Processing*, vol. 11, no. 6, pp. 585–595, June 2002.
- [17] A. T. S. Ho, X. Zhu, and P. Marziliano, "Fragile watermarking based on encoding of the zeros of the z -transform," *IEEE Trans. on Information Forensics and Security*, June 2006, revised and resubmitted.
- [18] C.-Y. Lin and S.-F. Chang, "Semi-fragile watermarking for authenticating JPEG visual content," in *SPIE International Conf. on Security and Watermarking of Multimedia Contents II*, vol. 3971, San Jose, USA, Jan. 2000, pp. 140–151.
- [19] —, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 153–168, Feb. 2001.
- [20] L. Xie and G. R. Arce, "A class of authentication digital watermarks for secure

BIBLIOGRAPHY

145

- multimedia communication,” *IEEE Trans. Image Processing*, vol. 10, no. 11, pp. 1754–1764, Nov. 2001.
- [21] —, “A blind wavelet based digital signature for image authentication,” in *Proceedings of the EUSIPCO 98*, Island of Rhodes, Greece, Sept. 1998.
- [22] —, “Joint wavelet compression and authentication watermarking,” in *IEEE Int. Conf. Image Process.*, Chicago, USA, Oct. 1998.
- [23] A. Said and W. A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits Syst. Video Technol.*, pp. 243–250, Mar. 1996.
- [24] C.-S. Lu and H.-Y. M. Liao, “Multipurpose watermarking for image authentication and protection,” *IEEE Trans. Image Processing*, vol. 10, no. 10, pp. 1579–1592, Oct. 2001.
- [25] —, “Structural digital signature for image authentication: an incidental distortion resistant scheme,” *IEEE Trans. Multimedia*, vol. 5, no. 2, pp. 161–173, June 2003.
- [26] D. Kundur and D. Hatzinakos, “Digital watermarking for telltale tamper-proofing and authentication,” in *Proceedings of the IEEE: Special Issue on Identification and Protection of Multimedia Information*, vol. 87, July 1999, pp. 1167–1180.
- [27] A. T. S. Ho, X. Zhu, and Y. L. Guan, “Image content authentication using pinned sine transform,” *EURASIP Journal on Applied Signal Processing, Special Issue on*

- Multimedia Security and Rights Management*, vol. 2004, no. 14, pp. 2174–2184, Oct. 2004.
- [28] A. Z. Meiri and E. Yudilevich, “A pinned sine transform image coder,” *IEEE Trans. Commun.*, vol. COM-29, pp. 1728–1753, Dec. 1981.
- [29] J. Fridrich and M. Goljan, “Images with self-correcting capabilities,” in *IEEE International Conf. on Image Processing*, Kobe, Japan, Oct. 1999, pp. 792–796.
- [30] D. Kundur and D. Hatzinakos, “Semi-blind image restoration based on telltale watermarking,” in *Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems & Computers*, vol. 2, Nov. 1998, pp. 933–937.
- [31] X. Zhu, A. T. S. Ho, and P. Marziliano, “A new semi-fragile image watermarking with robust tampering restoration using irregular sampling,” *Signal Processing: Image Communication*, Jan. 2007, accepted with minor revision.
- [32] M. Miller, I. J. Cox, and J. Bloom, “Watermarking in the real world: an application to DVD,” in *Proc. Wksp. Multimedia and Security at ACM Multimedia 98*, Bristol, U.K., Sept. 1998.
- [33] A. Tirkel, G. Rankin, R. V. Schyndel, W. Ho, N. Mee, and C. Osborne, “Electronic water mark,” in *Proc. DICTA’93*, Sydney, Australia, Dec. 1993, pp. 666–672.
- [34] R. G. Schyndel, A. Z. Tirkel, and C. F. Osborne, “A digital watermark,” in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Austin, Texas, Nov. 1994, pp. 86–90.

BIBLIOGRAPHY

147

- [35] G. Caronni, "Assuring ownership rights for digital images," in *Proc. Reliable IT Systems, VIS '95*, Vieweg Publishing Company, 1995.
- [36] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for data hiding," in *Proc. SPIE 2420, Storage and Retrieval for Image and Video Databases III*, Feb. 1995, pp. 164–173.
- [37] G. C. Langelaar, J. C. A. V. der Lubbe, and R. L. Lagendijk, "Robust labeling methods for copy protection of images," in *Proc. SPIE 3022, Storage and Retrieval for Image and Video Databases V*, Feb. 1997, pp. 298–309.
- [38] I. Pitas and T. H. Kaskalis, "Applying signatures on digital images," in *IEEE Workshop on Nonlinear Signal and Image Processing*, Thessaloniki, Greece, Oct. 1995, pp. 460–463.
- [39] M. J. J. B. Maes and C. W. A. M. V. Overveld, "Digital watermarking by geometric warping," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Chicago, IL, Oct. 1998, pp. 424–426.
- [40] P. Bas, J. M. Chassery, and B. macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. Image Processing*, vol. 11, no. 9, pp. 1010–1028, Sept. 2002.
- [41] G. K. Wallance, "The jpeg still picture compression standard," *Communication of the ACM*, vol. 34, no. 4, pp. 40–44, 1991.

- [42] W. B. Pennebaker and J. L. Mitchell, *JPEG still image data compression standard*. New York: Van Nostrand Reinhold Company, 1990.
- [43] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York: Academic Press, 1990.
- [44] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, Dec. 1997.
- [45] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proc. IEEE, Special Issue on Identification and Protection of Multimedia Information*, vol. 87, pp. 1127–1141, July 1999.
- [46] I. J. Cox and M. L. Miller, "Facilitating watermark insertion by preprocessing media," *EURASIP Journal on Applied Signal Processing*, vol. 14, pp. 2081–2092, 2004.
- [47] M. L. Miller, G. J. Doerr, and I. J. Cox, "Applying informed coding and embedding to design a robust, high capacity watermark," *IEEE Trans. Image Processing*, vol. 13, no. 6, pp. 792–807, June 2004.
- [48] J. J. K. O. Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Santa Barbara, CA, Oct. 1997, pp. 536–539.

BIBLIOGRAPHY

149

- [49] M. D. Adams, *The JPEG-2000 Still Image Compression Standard*. ISON/IEC JTC 1/SC 29/WG 1 N 2412, Dec. 2002.
- [50] M. S. Hsieh, D. C. Tseng, and Y. H. Huang, "Hiding digital watermarks using multiresolution wavelet transform," *IEEE Trans. Ind. Electron.*, vol. 48, pp. 875–882, Oct. 2001.
- [51] Y. Wang, J. F. Doherty, and R. E. Dyck, "A wavelet-based watermarking algorithm for ownership verification of digital images," *IEEE Trans. Image Processing*, vol. 11, pp. 77–88, Feb. 2002.
- [52] D. Kunder and D. Hatzinakos, "A robust digital image watermarking method using wavelet-based fusion," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Santa Barbara, CA, Oct. 1997, pp. 544–547.
- [53] T. A. Wilson, S. K. Rogers, and L. R. Myers, "Perceptual-based hyperspectral image fusion using multiresolution analysis," *Optical Engineering*, vol. 34, pp. 3154–3164, 1995.
- [54] D. Kundur and D. Hatzinakos, "Diversity and attack characterization for improved robust watermarking," *IEEE Trans. Signal Processing*, vol. 49, no. 10, pp. 2383–2396, Oct. 2001.
- [55] Y. W. Wang, J. F. Doherty, and R. E. V. Dyck, "A wavelet-based watermarking algo-

- rithm for ownership verification of digital images,” *IEEE Trans. Image Processing*, vol. 11, no. 2, pp. 77–88, Feb. 2002.
- [56] M. Barni, F. Bartolini, and A. Piva, “Improved wavelet-based watermarking through pixel-wise masking,” *IEEE Trans. Image Processing*, vol. 10, pp. 783–791, May 2001.
- [57] M. Kutter and S. Winkler, “A vision-based masking model for spread-spectrum image watermarking,” *IEEE Trans. Image Processing*, vol. 11, pp. 16–25, Jan. 2002.
- [58] J. Puate and F. Jordan, “Using fractal compression scheme to embed a digital signature into an image,” in *Proc. SPIE 2915, Video Techniques and Software for Full-Service Networks*, 1999, pp. 108–118.
- [59] M. Wu, M. L. Miller, A. Bloom, and I. J. Cox, “A rotation, scale and translation resilient public watermark,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, Phoenix, AZ, Mar. 1999.
- [60] R. Z. Liu and T. N. Tan, “An SVD-based watermarking scheme for protection rightful ownership,” *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 121–128, Mar. 2002.
- [61] B. J. Falkowski and L. S. Lim, “Image watermarking using the complex Hadamard transform,” in *Proc. IEEE Int. Symposium on circuits and Systems*, Geneva, Switzerland, May 2000, pp. 573–576.
- [62] S. A. M. Gilani and A. N. Skodras, “Watermarking by multiresolution Hadamard

BIBLIOGRAPHY

151

- transform,” in *Proc. Electronic Imaging and Visual Arts*, Florence, Italy, Mar. 2001, pp. 73–77.
- [63] A. T. S. Ho, J. Shen, and S. H. Tan, “Robust digital image-in-image watermarking algorithm using the fast Hadamard transform,” in *SPIE’s 47th Annual Meeting on Optical Science and Technology*, Seattle, USA, July 2002.
- [64] —, “Character-embedded watermarking algorithm using the fast Hadamard transform for satellite images,” in *SPIE’s 47th Annual Meeting on Optical Science and Technology*, Seattle, USA, July 2002.
- [65] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Attacks on copyright marking systems,” *Lecture Notes in Computer Science of Springer*, vol. 1525, pp. 218–238, 1998.
- [66] F. A. P. Petitcolas and R. J. Anderson, “Evaluation of copyright marking systems,” in *Proc. IEEE Int. Conf. Multimedia Comput. Syst.*, Florence, Italy, June 1999.
- [67] [Online]. Available: <http://www.cl.cam.ac.uk/fapp2/watermarking/stirmark/>.
- [68] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet, and T. Pun, “Second generation benchmarking and application oriented evaluation,” in *Information Hiding Workshop*, Pittsburgh, PA, USA, Apr. 2001.
- [69] [Online]. Available: <http://watermarking.unige.ch/Checkmark/index.html>.

- [70] E. C. Ifeachor and B. W. Jervis, *Digital signal processing: a practical approach*. Wokingham, England: Addison-Wesley Publishing Company, 1993.
- [71] A. J. Noushin and M. A. Fiddy, "Some new findings on the zeros of band-limited functions," *J. Opt. Soc. Am. A*, vol. 16, no. 7, pp. 1857–1863, July 1999.
- [72] C. W. Liao, M. A. Fiddy, and C. L. Byrne, "Imaging from the zero locations of far-field-intensity data," *J. Opt. Soc. Am. A*, vol. 14, no. 12, pp. 3155–3161, Dec. 1997.
- [73] R. H. T. Bates, B. K. Quek, and C. R. Parker, "Some implications of zero sheets for blind deconvolution and phase retrieval," *J. Opt. Soc. Am. A*, vol. 7, no. 3, pp. 468–479, Mar. 1990.
- [74] A. T. S. Ho, "Image information coding, compression and restoration incorporating prior knowledge," Ph.D. dissertation, University of London, Sept. 1983.
- [75] R. Vich, *z-Transform Theory and Applications*. Dordrecht: D. Reidel Publishing Company, 1987.
- [76] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [77] C. R. Parker, B. L. Satherley, and P. J. Bones, "Image reconstruction from zeros of the z-transform," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. v, 19–22, Apr. 1994, pp. V/465 –V/468.

BIBLIOGRAPHY

153

- [78] P. T. Chen, M. A. Fiddy, C. W. Liao, and D. A. Pommet, “Blind deconvolution and phase retrieval from point zeros,” *J. Opt. Soc. Am. A*, vol. 13, pp. 1524–1531, July 1996.
- [79] [Online]. Available: <http://www.ars.usda.gov/is/graphics/photos/>.
- [80] M. Kac, “On the average number of real roots of a random algebraic equation,” *Bull. Amer. Math. Soc.*, vol. 49, pp. 314–320, 1943.
- [81] ———, “A correction to ‘on the average number of real roots of a random algebraic equation,’” *Bull. Amer. Math. Soc.*, vol. 49, p. 938, 1943.
- [82] A. K. Jain, “A fast Karhunen-Loeve transform for a class of random processes,” *IEEE Trans. Commun.*, vol. COM-24, pp. 1023–1029, Sept. 1976.
- [83] ———, “Image data compression: A review,” *Proc. IEEE*, vol. 69, pp. 349–389, Mar. 1981.
- [84] P. M. Farrelle and A. K. Jain, “Recursive block coding— a new approach to transform coding,” *IEEE Trans. Commun.*, vol. COM-34, pp. 161–179, Feb. 1986.
- [85] A. Z. Meiri, “The pinned Karhunen-Loeve transform of a two dimensional Gauss-Markov field,” in *Proc. SPIE Conf. Image Processing*, San Diego, CA, 1976.
- [86] W. K. Pratt, *Digital Image Processing*, 2nd ed. NY: John Wiley & Sons, INC, 1991.
- [87] P. Marziliano and M. Vetterli, “Irregular sampling in approximation subspaces,” in *SampTA’99*, Loen, Norway, Aug. 1999.

- [88] P. Marziliano, "Sampling innovations," Ph.D. dissertation, N.2369 EPFL, Apr. 2001.
- [89] D. C. Youla, "Generalized image restoration by the method of alternating orghogonal projections," *IEEE Trans. Circuits Syst.*, vol. CAS-25, no. 9, pp. 694–702, Sept. 1978.
- [90] —, "Mathematical theory of image restoration by the method of convex projections," in *Image Recovery: Theory and Application*, H. Stark, Ed. Florida: Academic Press, 1987, pp. 29–77.
- [91] [Online]. Available: <http://mathworld.wolfram.com/>.
- [92] J. Spanier and K. B. Oldham, *An Atlas of Functions*. Washington, DC: Hemisphere, 1987.
- [93] W. K. Pratt, L. R. Welch, and W. H. Chen, "Slant transform for image coding," *Proc. Symp. Appl. Walsh Function*, Mar. 1972.
- [94] W. K. Pratt, W. H. Chen, and L. R. Welch, "Slant transform coding," *IEEE Trans. Commun.*, vol. COM-22, Aug. 1974.
- [95] R. J. Clarke, *Transform Coding of Image*. New York: Academic Press, 1985.
- [96] Z. D. Wang, "New algorithm for the slant transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 551–555, Dec. 1982.

BIBLIOGRAPHY

155

- [97] M. M. Anguh and R. R. Martin, "A truncation method for computing slant transforms with applications to image processing," *IEEE Trans. Commun.*, vol. 43, pp. 2103–2110, June 1995.
- [98] J. F. Yang and C. P. Fan, "Centralized fast slant transform algorithms," *IEICE Trans. Fundamentals*, vol. E80-A, pp. 705–71, Apr. 1997.
- [99] A. T. S. Ho, X. Zhu, Y. L. Guan, and P. Marziliano, "Slant transform watermarking for textured images," in *Proc. IEEE International Symposium on Circuits and Systems, Invited Session on Information Assurance and Data Hiding II*, Vancouver, Canada, May 2004.
- [100] X. Zhu and A. T. S. Ho, "A Slant transform watermarking for copyright protection of satellite images," in *4th Int. Conf. on Information, Communications & Signal Processing / 4th IEEE Pacific-Rim Conf. On Multimedia*, Singapore, Dec. 2003.
- [101] A. T. S. Ho, X. Zhu, and J. Shen, "Slant transform watermarking for digital images," in *Proc. Visual Communications and Image Processing, SPIE, Session on Image and Video Security and Watermarking I*, Lugano, Switzerland, July 2003.
- [102] H. Enomoto and K. Shibata, "Orthogonal transform coding system for television signals," *IEEE Trans. Electromagn. Compat.*, vol. EMC-13, pp. 11–17, Aug. 1971.
- [103] N. Ahmed and K. R. Rao, *Orthogonal Transform for Digital Signal Processing*. New York: Springer-Verlag, 1975.