INFORMATION HIDING IN DIGITAL IMAGES: WATERMARKING AND STEGANOGRAPHY

by

Po-Chyi Su

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

May 2003

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

UNIVERSITY OF SOUTHERN CALIFORNIA
THE GRADUATE SCHOOL
UNIVERSITY PARK
LOS ANGELES, CALIFORNIA 90089-1695

This dissertation, written by

PO-CHYI SU

under the direction of h____ dissertation committee, and approved by all its members, has been presented to and accepted by the Director of Graduate and Professional Programs, in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

____
Director

Date May 16, 2003

Dissertation Committee

C.-C. Jay Kuo *Chair*

____

____

____

Dedication

Dedicated with love to my family.

Acknowledgements

I would like to take this opportunity to express my deepest gratitude to my dissertation advisor, Prof. C.-C. Jay Kuo, for bringing me into the exciting field of multimedia signal processing and for his consistent instruction and encouragement. I learn a lot form his abundant knowledge, solid attitude and endless energy of doing research. The guidance under Prof. Kuo in these years helps me solve challenging problems and, most importantly, will benefit my entire career.

I am grateful to Prof. Antonio Ortega and Prof. Ming-Deh Huang for taking their valuable time and efforts to serve on both my qualifying examination and dissertation committees. Their rich research experience significantly improves the quality of the thesis. Also, I would like to thank Prof. Zhen Zhang and Prof. JongWon Kim for serving my qualifying examination committee and for their constructive comments and suggestions.

I take pleasure in thanking all my colleagues in Prof. Kuo's research group. Working with them broadens my scope of research and makes my staying at the University of Southern California interesting. Special thanks are extended to Dr. Houng-Jyh Wang. The several papers on digital watermarking that I worked with him have been very helpful to my doctorial study. His expertise in multimedia compression and processing deserves

a lot of credit. And I would like to thank my colleague, Chung-Ping Wu. The fruitful discussion with him on multimedia security issues clarifies many important aspects of my research.

Finally, there is nothing enough for me to show my appreciation to my family in Taiwan. I especially thank my parents and my brother for everything they give me. It is their continuous support that makes me achieve one of my most important milestones. Thank you all very much.

Contents

List of Tables

List of Figures

Abstract

Information hiding in multimedia data has drawn a lot of attention in recent years. A message is hidden in digital images, video and audio in an imperceptible manner so as not to interfere with the normal usage of these host media files while its existence can be helpful for some interesting applications. The hidden information related to content protection is usually termed *Digital Watermark*. The recent proliferation of the information-hiding research mostly comes from the development of digital watermarking techniques to meet the pressing need of content protection. The other interesting application is to transmit a large volume of data covertly in a multimedia file via information hiding. The objective in this application is to conceal the very existence of the hidden information using the innocuous multimedia data as a camouflage. Data hiding under this scenario is often termed *Steganography*, which literally means "covered writing."

In this research, we focus on information hiding in digital images. The dissertation consists of two major parts. In the first part, we investigate information hiding in the state-of-the-art still image codec, JPEG-2000. A joint compression/watermarking scheme is proposed for the copyright protection purpose. Next, we consider steganography in

JPEG-2000. Algorithms are developed to ensure that a high volume of data can be hidden reliably in JPEG-2000 compressed images.

In the second part, we aim at solving a very challenging problem in digital image watermarking, *i.e.*, synchronized detection under geometrical modifications. In order to derive a robust watermarking scheme resilient to affine transformations, we propose to embed a structural grid signal into digital images for synchronized watermark detection. A spatial-frequency composite watermarking scheme is designed so that the watermark can survive attacks such as rotation, scaling and cropping, etc. A perceptual watermarking scheme using block-based DCT methodology is also developed. It turns out that grid embedding/detection is of great help in developing generalized block-based watermarking as well. Finally, the dissertation ends with some concluding remarks and some future research topics as an extension of our current research.

Chapter 1

Introduction

1.1 Significance of the Research

The rapid growth of computational facilities and wide availability of network access lead to efficient digital data processing, delivery and storage. The superior compression technology further helps to compact digital content with high quality into a smaller data stream to facilitate its transmission and manipulation. In this multimedia era, people can not only enjoy all kinds of recreation, such as watching digital video, listening to digital audio and reading electronic publications, but also process the digitized multimedia data easily thanks to advanced utilities. Creation, authoring and dissemination of multimedia content can be realized by ordinary users nowadays instead of limited to only a few professionals.

These advantages definitely benefit our daily life but raise certain concerns. First of all, unlike traditional analog copying with which the quality of the duplicated content is degraded, a large number of perfect digital copies can be reproduced in a short period of time. The content providers could be reluctant to distribute their work in digital format since unlimited copying of digital data by users without paying any royalty will cause them a considerable amount of financial loss. The shortage of media content will eventually hinder the progress of multimedia technologies. Protection of content owners' intellectual property rights has to be enforced to ensure the owners receive what their hardworking and creativity deserve. Secondly, as great amounts of multimedia data are stored in the digital format, some digital data, such as digital photos or surveillance video, could be used as legal documents. However, digital document can be tampered or forged easily and the vulnerability of digital data may invalidate their legitimate usage as evidence on the court. Thus, the authenticity of digital data has to be taken care of seriously. That is, in such forensic cases, we have to make sure that the digital document is authentic and the information content is not modified in delivery to its destination. Therefore, it comes a pressing need to develop effective ways to deter users from illegally reproducing or misusing digitized data.

Cryptography is a classical method to prevent digital data from unauthorized access. Digital data are transformed by the encryption process so that the meaning of digital data becomes obscure to a person who intercepts the data but does not have a key for decrypting them. For example, the content owner can either encrypt a compressed video bit-stream or each of the video frames by a secret key. Users without the proper decryption key cannot correctly decompress the bitstream (or the expanded video is not viewable) since video frames are scrambled in the encryption process. However, such protection may not be enough for multimedia data since the content is no longer protected after the intended receiver decrypts the video. The intended receiver may not follow the legitimate usage of these multimedia data that he or she agreed before and may distribute them widely through high-speed networks or other convenient channels to make illegal profit. In other words, cryptography only helps protect the digital content during its transmission. To compensate the deficiency of cryptography for multimedia data, researchers have considered and developed information-hiding techniques to embed a signal in digital media to convey the information of interest. The embedded signal is known as a digital watermark.

As traditional watermarks exist in papers, the existence of a digital watermark in a document does not interfere or severely degrade the quality of the content. Although a digital watermark is generally invisible or inaudible to human beings, it can be detected or extracted through computations. Instead of being inserted in the header or the tail of the data that would be removed easily, a digital watermark is embedded in the content directly. The embedded watermark may remain in the media even after normal data processing or format conversion. It is this tight combination of the embedded signal and the carrying media content that makes digital watermarking a promising technique to protect multimedia contents. For example, the hidden watermark in a digital document can act as a proof of the authorship and/or to identify the source of the data. The presence of owner's unique watermark in an investigated document from an unauthorized possessor can prove the misuse of the document. The digital watermark may also help verify the originality of content, *i.e.* to identify if a suspected document is an original copy. Once the content is manipulated, the embedded watermark will be destroyed so that the authenticator can examine the existence of the watermark to show the integrity of contents.

The other interesting application is the point-to-point secret communication between trusting parties to convey or exchange secure information via information hiding. The secret messages are hidden within another seemingly innocuous host media data to achieve covert communication. This application can also be termed as "Steganography," which has long drawn tremendous interest in human history and may be applied in the military-related scenario.

Based on the above discussion, we can see that the research of information-hiding techniques is significant in many applications. The designer should take different characteristics of the hidden information into account to achieve specific objectives.

1.2 Contributions of the Research

In this research, we concentrate on two important issues of information hiding in digital images. The first issue is related to information hiding in JPEG-2000 [33], [84], [12], [1]. Along this direction, the following contributions have been made.

- We developed a robust watermarking scheme under the framework of this upcoming still image compression standard.
- We studied steganographic applications, *i.e.* covert communication, in JPEG-2000 by hiding a large volume of information in the JPEG-2000 compressed bit-stream.

The second issue is related to robustness of a digital watermark against geometrical attacks, which is one of the most challenging problems in digital image watermarking. Along this direction, this research has the following contributions.

- We proposed to embed structural grid signals into digital images to tackle the synchronization problem.
- A spatial-frequency composite watermarking scheme is thus derived to enable the embedded digital watermark to survive generalized geometrical modifications.

- It was shown that the algorithms of grid embedding and detection also work well in block-based DCT watermarking approaches, which are widely proposed and used in this field.

- We evaluated the performance of each system by testing its resilience in various situations.

More details of the contributions are given below.

1.2.1 Information Hiding in JPEG-2000 Compressed Images

JPEG-2000 is a new still image coding standard. It is intended as the successor of the existing JPEG standard in many important areas. In addition to its superior coding performance in both low and high bit-rate compression applications, JPEG-2000 has numerous other interesting features, including progressive recovery of an image by fidelity or resolution, Region of Interest (ROI) coding, whereby different parts of an image can be coded with different fidelity, random access to particular regions of an image without the need of decoding the entire coded stream and good resilience to bit-errors. Besides, JPEG-2000 allows efficient lossy and lossless compression within a single unified coding framework. The state-of-the-art image codec is also designed to avoid excessive computational and memory complexity. It is believed that JPEG-2000 will be used widely and its rich features and functionalities will benefit many emerging applications. As many images will be compressed by JPEG-2000 in the near future, it is worthwhile to investigate efficient information-hiding techniques in JPEG-2000 compressed images. Therefore, we carry out the pioneering research on this subject.

1.2.1.1 Robust Digital Watermarking in JPEG-2000

First of all, we develop an integrated approach to JPEG-2000 and digital watermarking[79, 80, 72]. By integrating the watermarking technique with JPEG-2000, the proposed watermark embedding and retrieval system is more efficient than the existing watermarking schemes. The watermark is embedded in the discrete host image coefficients by examining bit-planes. A binary (or bi-polar) watermark sequence is used, and the resulting watermarked image coefficients also take only discrete values. Therefore, both watermark embedding and detection occur directly in the compressed domain. The integrated scheme eliminates both the need to compress the host image after watermark embedding and the need to decompress the watermarked image before watermark detection.

In addition to efficiency, the proposed scheme has many interesting features. The embedded watermark is robust against various signal processing attacks including compression and filtering while the resulting watermarked image maintains good perceptual quality. The embedded watermark can be detected progressively so that an operation, which is enabled via watermark detection such as "never copy," can be enforced earlier without waiting for the whole image to be downloaded. Furthermore, ROI watermarking can be easily coupled with ROI coding in the proposed scheme. In this case, when we receive ROI without the insignificant background, the watermark can still be detected without ambiguity. The embedded watermark can be detected without the knowledge of the original image so that it is a "blind" watermarking scheme. Experimental results show that the proposed integrated JPEG-2000 watermarking performs very well and supports all above claims.

1.2.1.2 High-Volume Information Hiding in JPEG-2000

Next, we consider steganography in digital images compressed by JPEG-2000. The objective is to develop a data hiding scheme under the framework of JPEG-2000 so that a large amount of information can be secretly transmitted to the intended recipient in a reliable fashion. Therefore, instead of focusing on robustness as done in digital watermarking for the copyright protection purpose, we take capacity and reliability into more serious consideration in high-volume information hiding.

However, the coding structure of JPEG-2000 limits this upcoming standard from being a reliable host for reliable information hiding. We analyze the problem by examining the JPEG-2000 coding flow and determine appropriate positions for effective data hiding. Practical schemes are then proposed to embed a large amount of data into the JPEG-2000 compressed bit-stream in a reliable manner [76]. Several design issues were examined to help achieve a better balance among the complexity, capacity and visual quality. Experimental results are given to demonstrate the performance of the proposed algorithms.

1.2.2 Towards Affine-Invariant Digital Image Watermarking

Most existing watermarking schemes are based on the additive spread-spectrum method because of its robustness against noise and distortions. However, most spread-spectrum watermarking methods fail to detect the watermark when the watermarked image undergoes geometrical modifications such as cropping, rotation, scaling, or even the change of the aspect ratio. These operations are accessible to most casual users and can be applied with a low computational cost. Since images after geometrical change usually preserve the same perceptual quality of the original image, a practical watermarking scheme must be robust against geometrical attacks.

The main reason for the failure of the spread-spectrum watermarking scheme under geometrical modification is the loss of synchronization between the watermark detector and the embedded watermark. Spread-spectrum watermarking methods adopt the matched filter (or called the correlation detector) to detect the watermark. It determines the existence of a certain watermark by calculating its similarity or correlation with the extracted signal. Since the possibly existing watermark is hidden in a very strong noise, i.e. the image content, it is very difficult for the watermark detector to predict the correct position of the hidden watermark when the image is cropped, scaled or rotated. Exclusive searching is computationally impossible especially for two-dimensional digital images. Although the searching process could be simplified if the original image is available during watermark detection and some known pattern recognition or registration processes could be applied, the performance is still not satisfactory because a very precise synchronization is usually required for watermark detection. Furthermore, cropping or scaling can cause information loss of the watermark so that the embedded information may not be correctly determined. We tackle the synchronization problem by structural grid embedding. The idea is motivated by the algorithm proposed by Kutter [40]. With different underlying watermark signals being used, we develop two watermarking schemes to resist affine modifications.

1.2.2.1 Spatial-Frequency Composite Watermarking

We first propose a spatial-frequency composite watermarking scheme[73, 74] to resist cropping attack and generalized affine transformations. Two signals will be embedded into a digital image, one in the frequency domain, or more specifically, Fourier domain and the other in the spatial domain of the image. The signal embedded in the Fourier domain contains the desired hidden information, *i.e.* the digital watermark that will be carried with the host image. The signal embedded in the spatial domain, *i.e.* the grid signal, is used to achieve self-registration of the investigated image with the original image. The self-registration process can convert an image to its original orientation and scale without explicitly resorting to the original host image or the watermarked image. After the registration process as a result of spatial-domain grid signal detection, the hidden information can then be successfully determined from the registered image by detecting the frequency-domain watermark. Both detection processes of spatial-domain grid and frequency-domain watermark are blind (i.e. not requiring the original image). Experimental results will show that the embedded watermark survives a combination of manipulations applied to the watermarked images.

1.2.2.2 Perceptual Block-based DCT Watermarking

Block-based watermarking is a general type of schemes, in which the image is divided into blocks for watermark embedding and detection. A clear benefit is that local visual masking effects can be incorporated into the watermarking system. With divided blocks and their local statistics, the balance of robustness and invisibility of the watermark can be achieved in a decent way. In blocks with more activities, a stronger watermark can be embedded without being noticed and the watermark can resist more serious attacks as well. It should be noted that the block-based Discrete Cosine Transform (DCT) is utilized in many existing watermarking schemes. By decomposing the image into several frequency bands using DCT, we can embed the watermark in the significant frequency coefficients to attain a robust watermark. The watermarked image is then formed by applying inverse DCT to each image block. Besides, block-based DCT is essential in image/video compression, *e.g.* JPEG, MPEG 1/2/4. The sensitivity of the human visual system to DCT basis functions has thus been extensively studied, resulting in a recommended quantization table for JPEG [55, 91].

It is advantageous to use the visual model within this framework in block-based watermarking systems to reduce the impact of quality degradation and, at the same time, to make the watermark survive JPEG compression attack better. Real-time applications could also be achieved by applying watermarking in the same domain as image compression. Moreover, spatial division can be used to increase the amount of the embedded watermark, *i.e.* groups of blocks are embedded with different information so that multiple bits can be carried by a single image. Security of the system can also be increased by scrambling the order of the blocks so that the watermark cannot be detected correctly without the correct descrambling process.

Existing research on block-based watermarking focuses on the robustness against filtering/compression attacks and perceptual issues. However, as mentioned earlier, geometrical attacks are very difficult to resist, especially for block-based methods. The geometrical attack will seriously limit the usage and applications of block-based watermarking techniques. In block-based DCT watermarking schemes, the detector has first to determine the correct position to calculate DCT. Several trials of DCT would be needed when only simple cropping attacks are applied to the investigated image. Nevertheless, if the image undergoes certain rotation and scaling, watermark detection will become a very difficult task if no extra information or mechanisms are supplied to achieve synchronization.

To compensate the deficiency of block-based watermarking schemes, we extend our research on composite watermarking to robust and perceptual watermarking using block-based DCT [75]. Again, there will be two kinds of signals embedded into the image: the block-based watermark for carrying the hidden information and the grid signal for self-registration. Both embedding processes take human visual effects into consideration and the interference between the two signals is minimized. The complete watermark embedding/detection structures and experimental results will demonstrate the advantages of the proposed algorithm.

1.3 Organization of the Dissertation

The content of the dissertation is organized as follows. In Chapter 2, we describe the general framework of the watermarking systems followed by the discussion of some famous and early watermarking methods. In Chapter 3, we investigate information hiding in JPEG-2000. There are two parts in this chapter, *i.e.* digital watermarking and covert communication in JPEG-2000. A brief review of JPEG-2000 compression will be presented first. Next, we will propose the detailed algorithm for integrating watermarking procedures with JPEG-2000 compression. The resulting system is theoretically analyzed and empirically tested. High-volume information hiding in JPEG-2000 compressed images is then examined. Effective algorithms are proposed to achieve reliable covert communication. In Chapter 4, we tackle the synchronization problem to achieve digital watermarking schemes resisting generalized geometrical modifications. We illustrate some existing methods to cope with geometrical attacks and discuss advantages and limitations of these algorithms. Then, we propose our own solution, which makes use of structural grid signals to achieve synchronized watermark detection. We propose a novel spatial-frequency composite watermarking scheme, which shows an impressive overall performance and is resilient to generalized affine transformation. Then we apply the similar idea to block-based watermarking schemes. After discussing the properties of block-based DCT watermarking methods, we adopt a perceptual block-based DCT watermarking algorithm as the baseline scheme and enable it to survive affine transform attacks using structural grid signal embedding. Finally, possible extensions of the current research and concluding remarks of the dissertation are given in Chapter 5.

Chapter 2

Overview of Information Hiding in Digital Images

2.1 General Concept of Information Hiding

2.1.1 Information Hiding, Watermarking and Steganography

At the beginning, we would like to clarify the definitions of information hiding, watermarking and steganography. Although these three terms share many similarities and could even be interchangeable in some literature, certain fundamental differences lead us to define them as follows.

Information hiding is a general practice encompassing a broad range of applications in which the messages are embedded into the other media content for varying purposes. Watermarking and steganography are two types of information hiding. Steganography, which is derived from the Greek words meaning covered writing, hides the secret message into an innocuous host content to achieve covert communication. In order to act as a successful camouflage to conceal the very existence of the secret message, the host media content is usually chosen to have nothing to do with the hidden information. Similar to steganography, watermarking is also a procedure of imperceptibly embedding the information, *i.e.*, a digital watermark, into the content. However, a digital watermark usually represents the ownership of the content, the identity of the legitimate content user or other information used to help protect the host content. In other words, there exists a strong relationship between the embedded digital watermark and the host content. Besides, in order to achieve the intended functions, the existence of a digital watermark is usually known to the users, in contrast to the fact that the hidden information in steganography is kept secret to the public. Therefore, the dependency between the host media content and hidden information differentiates digital watermarking and steganography.

2.1.2 Generic Model of Information Hiding in Digital Images

Fig. 2.1 illustrates the generic model of information hiding systems for digital images. In the embedding process, the input data include the original image, the hidden information and a key for security issues and the output is the composite image, or watermarked image in applications of digital watermarking. The original image can be of the compressed or uncompressed format. Many schemes work in the uncompressed format to tie the hidden information more closely with the image content while operating in the compressed format may increase efficiency of implementation. The hidden information can be a text, a sequence of number, a binary logo or even a gray-level image. The key, which can be public or private depending on who has the access privilege of the hidden information, is used to increase the security of the system. The key can also be viewed as an identification number of the legitimate user. In the detection process, the input data are the image of interest and the key. The original image may or may not be required in the detection process. The output is either the hidden information itself or some kind of confidence measurement of its existence. In the latter case, the targeted hidden information may be used by the detector for comparison.

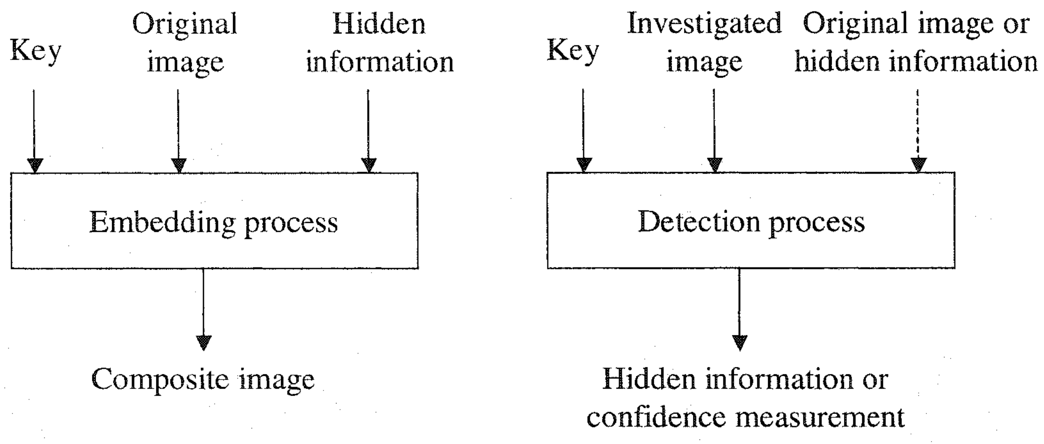


Figure 2.1: The information hiding system.

In digital watermarking, we can further adopt a viewpoint of communication to describe the model in a more precise way. Fig. 2.2(a) shows a classical communication model. In communication, the message to be transmitted is first encoded. The encoding process transforms the data stream for error correction and/or frequency spreading purposes. The encoded message is then used to modulate a carrier signal and transmitted through the channel, where it encounters additive noises. The receiver demodulates the noisy signal to a coded message. Finally, this coded message is decoded to produce the received message. One way to express a watermarking model is to apply the communication model directly. The media to be watermarked is viewed as the additive noise. Then watermarking becomes a weak signal detection problem because the watermark extractor/decoder needs to retrieve the watermark from a very noisy channel.

A more delicate way to delineate a watermarking model is shown in Fig. 2.2(b). The modulation step is replaced by the step of embedding the encoded message into some media content and the demodulation step is replaced by the step of extracting the watermark from the received signal. The noise in the transmission channel results from various media processing procedures such as compression, filtering or malicious attacks aimed to destroy the message. Note there exists a second receiver in the watermarking model, *i.e.*, the human perceptual system, which should receive a message that is perceived the same as the carrier media content. This fidelity constraint is analogous to the signal power constraint in a communication channel.

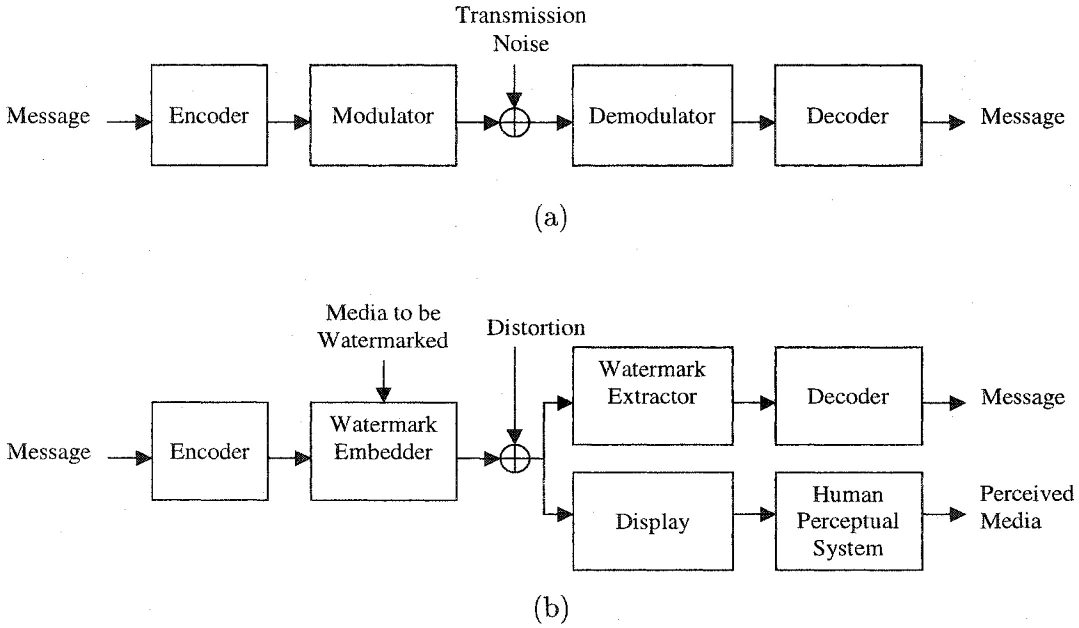


Figure 2.2: (a) The model of a communication channel and (b) the precise watermarking model.

2.1.3 Applications of Information Hiding

Several interesting applications may be achieved by using the techniques of information hiding. We briefly mention some of the applications that have been considered.

1. Copyright Protection

   The pressing need of protecting intellectual property rights is the main driving force of the research in information hiding. A digital watermark representing the copyright information can be embedded into the media data so that the owners can claim their ownership of the data in court by extracting the watermark unambiguously if someone infringes on their copyright.

2. Fingerprinting

   The application of fingerprinting works in a slightly different way to protect the media content. To avoid unauthorized duplication and distribution of the multimedia content, an author can embed a distinct label as fingerprint into each copy of the data. If an unauthorized copy is found later, the origin of the copy can be traced by retrieving the fingerprint. Besides, if the fingerprint represents certain customers' personal information such as the credit card number, the original buyers will think twice before they distribute the received data to others because they will run the risk of spreading their own personal information around.

3. Authentication

   Multimedia data in digital format facilitate modifying and editing but problems arise when the possibly tampered data are to be used for legal purposes or the authenticity of the data is important to the users. In such situations, the data must be credible, *i.e.*, the information content in the signal is not modified in transit to its destination. Information hiding provides a tamper-proofing tool for digital content. Once the content is manipulated, the embedded signal or watermark will be affected or even destroyed so that its status or existence can be used to verify the integrity of the content. Although authentication of media content can be achieved through conventional cryptographic techniques, the advantage of using information hiding is that the authenticator is inseparably bound to the content, which simplifies data handling.

4. Usage Control

   In some applications in which the multimedia content needs special hardware for copying or viewing, a digital watermark can be used to control the usage, such as the permission of viewing, listening or recording, etc. If the media player or recorder detects illegal copies based on an unmatched watermark, it will refuse to play or record the digital data.

5. Convert communication

   The nature of media data, such as images, audio and video, provides a good host for hiding the high-volume information in steganographic applications. By using the innocuous host media data as a cover, we may fool possible eavesdroppers to communicate with the trusted party secretly. From a communication viewpoint, the host media can be seen as a secret channel. Obviously, covert communication may be of military usage. Besides, some governments limit the usage of encryption services and information hiding may be a way to bypass the restriction.

6. Broadcast monitoring

   Broadcast monitoring is one of the potential applications that can be achieved by information hiding techniques. Advertisers, who pay the television or radio stations for commercial advertisements, would like to ensure those advertisements be broadcast as promised. The insertion and detection of the hidden signal may help them to verify this without involving much human effort for monitoring. Broadcast monitoring via information hiding can be achieved by either watermarking or steganography. We may embed the signal in some particular segments of the broadcast channels, which are available but not used for content transmission. The drawback is that special equipment may be required for handling the additional signal. Embedding a digital watermark in the media content as the controlled signal has the advantage of being fully compatible with the installed broadcast equipment. The primary disadvantage is its comparatively complicated embedding/detection process.

7. Annotation

   The bits embedded into the media data may comprise an annotation, giving further information about the media content. For example, a photographic image could be labeled to describe the time and place the photograph was taken, a procedure that could be done automatically by the processor in a camera. In multimedia databases, a digital watermark may represent a serial number or an index for efficient management. Besides, a digital watermark can be a flag to indicate types and properties of the content, which may possibly be used to track pornography on the network. In medical applications, embedding the date and patient's name in medical images could be a useful safety measure.

2.1.4 Requirements of Information Hiding

Although each application we mentioned above may have specific or different requirements, we list in the following some general requirements to which a designer of an information-hiding scheme should pay attention.

1. Unobtrusiveness

   The hidden information or watermark must be embedded in a sophisticated way to avoid degrading the perceptual quality of the host media. The users will not sense the existence of the embedded information by viewing or listening to the composite or watermarked media. To achieve this, many information hiding techniques make use of certain human perceptual models in the embedding process. It should be noted that this requirement has to be fulfilled in all of the applications of information hiding.

2. Robustness

   In order to achieve copyright protection, a robust watermarking scheme is required so that the embedded watermark can always remain in the multimedia data and survive all possible signal processing procedures, which degrade the data to the extent that the commercial value of the media is still maintained. These procedures are referred to "Watermark Attacks." These watermark attacks may be applied to the media content for the purposes of editing, storage or merely circumventing watermark detection. The attacks include but are not limited to compression, filtering, noise adding, geometrical modification and even anti-watermarking softwares. In data authentication, we need a fragile watermarking scheme instead so that the malicious attacks will destroy the embedded watermark. We can thus check the existence of the watermark to verify the integrity of the investigated media data. Especially, the watermark should be fragile to the distortion used to change the image content, such as replacing a portion of the image with other objects. However, the fragile watermark should not be sensitive to compression or format conversion, which is used for storage or transmission purposes. Therefore, the challenge here is to provide a watermark that can distinguish between information altering and simple signal altering transformations.

3. Detection scenario

   Detection of the hidden information can be done with or without the presence of the original media depending on the applications. In digital watermarking, the watermark that requires the original media as a reference for detection is classified as private watermark since the original media information is usually not available to the public. Therefore, the private watermark can only be detected by the content owner or the justified third party. The detection of public watermark does not resort to the original media so it is possible that all users can verify the existence of the watermark. Watermark detection without resorting to the original media is also referred as "blind" or "oblivious" watermark detection. The detection involving the original media is usually more robust but the usage is restricted since it is not practical to search the original data in the database whenever we would like to detect the watermark from a suspected data. Therefore, "blind" watermark detection has become a requirement for most watermarking schemes. In steganography, blind detection is always required.

4. Capacity

   By capacity, we refer to the ability to detect individual hidden messages with a low probability of error as the number of differently marked versions of the media increases. The amount of the embedded information must accommodate the application being considered. For example, the watermark may represent one-bit information to determine if the digital content can be reproduced or not while the watermark may contain n bits information to indicate the identification number of the intended recipient. In steganographic applications, the requirement of capacity or payload needs to be even higher. In general, information-hiding techniques should provide a way to insert as many distinguishable hidden messages as possible.

5. Low false positive rate

   While the hidden information should be detected correctly and unambiguously, the false detection of the hidden message should be as low as possible. In digital watermarking, false detection includes false negative detection and false positive detection. The false negative detection means that the watermark exists in the media but the watermark detector fails to detect it. The false positive detection means that the watermark is falsely detected in an unwatermarked media. Let us consider the case that an important ball game is broadcast and a lot of people would like to record it. However, a "never copy" watermark is falsely detected by some watermark detector in consumers' recording facilities and the recorders refuse to record the game. We can imagine how seriously the legitimate users will be affected by the false positive detection. Thus, the false positive detection rate must be made as low as possible.

6. Efficient implementation

   Efficient implementation of an information-hiding scheme is another important issue. Considering the prevalence of commercial facilities such as digital cameras, scanners and video players/recorders, we may require the hidden information to be embedded and detected during recording or playback. Complex design of an information-hiding scheme will limit its usage in real-time applications and hardware implementations.

7. Security

   An information-hiding system should be secure enough to prevent attacks from both casual users and knowledgeable hackers. Cryptography still plays an important role in maintaining the security level of information-hiding schemes.

Therefore, recent research directions on information hiding include the following: how to effectively embed an adequate amount of information in the digital media without introducing perceptual distortion and how to efficiently, correctly and unambiguously extract the hidden information from the investigated media without resorting to the original data, even though the investigated media may have been modified by signal processing procedures. Information hiding is thus becoming a challenging field, since we have to consider several design trade-offs such as effectiveness vs. efficiency, capacity vs. correctness and quality degradation vs. robustness, etc. Besides, a good understanding of media representation, signal detection and signal processing is necessary for a designer to construct a well-rounded information-hiding system.

2.2 Approaches of Information Hiding in Digital Images

Over these few years, a large number of information-hiding algorithms have been proposed. Due to the fact that surveying so many publications is not easy and that many approaches share common ideas, we briefly describe the general methodologies of information-hiding techniques and illustrate them by exemplary schemes.

2.2.1 Information Hiding with the Least Significant Bit Modification

The most straightforward approach of hiding information in digital images is to modify the least significant bit (LSB) of image pixels. In most of the images, the LSB plane doesn't contain visually significant information. We can thus replace the LSB plane with the hidden information without affecting the perceptual quality of the image. Besides, for a 8-bit gray-level image, the payload of the information hiding scheme based on LSB modification can be as high as 1/8 of the image data size. Therefore, this approach may be suitable in applications of covert communication to transmit high-volume secret data. However, as the LSB plane is insignificant to the image, it can be removed easily by such procedures as filtering or compression so its usage in robust watermarking is limited.

Most of the early research on information hiding is based on LSB modification [86, 87, 53, 93, 31]. Among them, we have to mention the work by Tirkel *et al.* [86, 87]. They first defined the term "digital watermark" to draw the analogy between the digital information embedding and the paper watermark. Besides, they also recognized the importance of digital watermarking for such applications as copyright enforcement, counterfeit protection and controlled access to image data. Two methods were proposed in their papers. In the first method, after the 8-bit image pixel is first compressed to 7-bit representation through histogram manipulation, the LSB plane is replaced by the hidden information. The watermark decoding can be done easily by checking the LSB plane. In the second method, an M-sequence [50] is circularly-shifted and added to each row of the image in the LSB plane. The watermark detection is carried out by checking the cross-correlation between rows. Therefore, their approach of correlation detection is also considered one of the pioneering researches to apply the concept of spread spectrum in information hiding. We will describe the spread spectrum watermarking in Section 2.2.4.

Other variants of LSB-based information hiding exist. Because of the fragile nature, some watermarking schemes adopted this methodology to achieve data authentication [93, 98, 95]. If malicious attacks are applied on the image, some portions of the fragile watermark in the LSB plane may be destroyed and the scheme may thus signal that the investigated image has been modified and may locate the tampered regions as well. In some cases, the embedded watermark may be a binary logo to ambiguously declare the ownership information. However, embedding such information in the LSB plane could raise certain security concerns. Voyatzis and Pitas [88] proposed toral automorphisms to scramble the logo before it is embedded into the LSB plane. A 2-D toral automorphism is a spatial transformation with periodic orbits, which means that a point in the 2-D data will be changed in position after each transform or iteration but will be transformed back to the initial position after, for example, K iterations. Therefore, the embedder transforms the logo to a chaotic form after T iterations while the detector can apply $K-T$ transforms on the scrambled logo to recover back to the original logo. The value K and T are kept confidential and only known to both embedder and detector to ensure certain degree of security.

Information hiding in halftone or dithered images may also be viewed as a LSB-based scheme since each image sample is represented by only one bit. The information is embedded by flipping the image samples. Early work can be traced back to a paper by Tanaka *et al.* [83] in 1990. Advanced work by Knox *et al.* [35] made use of different halftone patterns to embed a visible watermark. Recent work [24] embedded the hidden information in halftone images by taking the local statistics into consideration.

2.2.2 Information Hiding by Changing the Statistics

As LSB-based methods are vulnerable to some trivial attacks, the researchers think of using redundant information hiding, *i.e.*, embedding the same information for multiple times and detecting it via statistical methods. Caronni proposed a system for tracking unauthorized image distribution [10]. The marking process is called "tagging." A tag is a square random-noise pattern with size $N \times N$. The image is first divided into $N \times N$ blocks and the local variance of each block is calculated. Only locations with smaller variance will be used for tagging. A tag is then imposed onto the selected image block. The selected block is hidden with 1 bit and is only tagged if the bit value is one. To recover an embedded bit, the difference between the original and the tagged image is computed. Then the mean of a supposedly tagged block is compared to the neighboring mean to determine the bit value.

Bender *et al.* proposed a scheme called "Patchwork" [6] to increase the robustness of their LSB-based spatial-domain watermark. The scheme randomly selects N pairs of pixels, (a_i, b_i), to hide 1 bit by increasing a_i's by one and decreasing b_i's by one. The expected value of the sum of the differences between the a_i's and b_i's of N pairs will be $2N$, if the image is marked. Pitas proposed a similar idea to cast the signal in digital images [60]. To embed more bits, Langelaar *et al.* [44] extended the Patchwork scheme by splitting the image into blocks and embedding one bit in each of the blocks.

2.2.3 Information Hiding in the Frequency Domain

The methods that we illustrated above embed or detect the information or watermark in the spatial domain, *i.e.*, the luminance intensity, of the image. We may classify these approaches as spatial-domain watermarking schemes, which modify the values of image pixels directly for information hiding. In contrast to the spatial-domain watermarking, there exists frequency-domain or transform-domain watermarking, which modifies the frequency coefficients for information hiding after a proper transform such as the discrete wavelet transform (DWT), the discrete cosine transform (DCT) or the discrete Fourier transform (DFT) is applied.

Although the major difference between spatial- and frequency-domain watermarking schemes is the convenience of implementation, the two approaches can provide different functions to cope with various applications. Generally speaking, frequency-domain watermarking schemes tend to achieve a better balance between robustness and fidelity than spatial-domain schemes. First of all, the information embedded in the frequency domain will be spread in a larger region in the spatial domain. Therefore, the embedded watermark may survive some image processing procedures well. Besides, image transforms compact energy to a few transform coefficients. These coefficients have a larger magnitude, and are perceptually important to the image representation. A large distortion on significant coefficients will result in serious quality degradation. Therefore, embedding a watermark by slightly changing those significant coefficients will result in a robust watermarking scheme because significant coefficients usually remain stable even after image manipulation. The watermark embedded in significant coefficients can thus be detected more reliably. Moreover, some perceptual models are developed in the transform domain. By taking the human visual system model into consideration, we can make the embedded watermark invisible to human eyes. Therefore, the frequency-domain watermarking approaches are more popular in the watermarking literature.

Block-based DCT watermarking is a popular approach. An early block-based DCT watermarking method is proposed by Koch *et al.* [36, 37]. The image is first divided into blocks of size 8 × 8, for which DCT is computed. From these blocks, two coefficients are selected pseudo-randomly as a pair in the mid-frequency band. The selected coefficients are quantized by using the default JPEG quantization table with a relatively low JPEG quality factor to accommodate lossy JPEG compression attack. These coefficients are then modified such that the difference between them is either positive or negative, depending on the bit value. Bors *et al.* [8] proposed a similar method. Certain blocks are selected and their middle-range DCT coefficients are modified so that they fulfill a linear or circular constraint according to the information to be embedded. Langelaar *et al.* [46, 45] proposed to embed the watermark by selectively discarding the high-frequency DCT coefficients. A set of DCT blocks is chosen and divided into two subsets of equal size. The energy of the high-frequency coefficient in one subset is reduced by removing high-frequency coefficients. The information can be extracted by comparing the energy in the two subsets. The method is thus called differential energy watermarking. Busch *et al.* [9] proposed to embed the watermark into DCT blocks that are selected according to the block activity. The block activity is directly measured through DCT coefficients so that efficient watermarking can be achieved.

Many other schemes are based on global image transforms, such as global DCT, wavelet and the Fourier transform. Most of them adopt the concept of spread spectrum watermarking to achieve the robustness against such attacks as compression and filtering processes. We will discuss them more thoroughly later.

2.2.4 Spread Spectrum Watermarking

With the strong relationship between watermarking and communication as described in Section 2.1.2, many watermarking schemes are based on the additive spread-spectrum method, which is inspired by the spread-spectrum modulation technique in the digital communication system. This technique provides more security and resistance to channel noises for digital communication. Similarly, the spread-spectrum watermarking scheme can resist more serious content distortion. The hidden information is represented by a pseudo-random signal with a low amplitude, which is added to the host data and then detected by using a correlation receiver or a matched filter. The watermarked image can keep good perceptual quality since the value of each pseudo-random number is small.

Besides, the pseudo-random signal is usually generated by a key, which is also suitable to information-hiding applications.

We discuss the spread spectrum watermarking approaches according to the domain where the watermark embedding/detectin is operated. In other words, we divide the schemes into the spatial-domain watermarking and the frequency-domain watermarking.

2.2.4.1 Spread Spectrum Watermarking in the Frequency-domain

A large portions of spread-spectrum watermarking schemes operate in the frequency or transform domain for better performance. An early spread-spectrum transform-domain scheme was proposed by Cox *et al.* [14]. They first suggested that the watermark signal must be placed in the perceptually significant components of the content to survive common signal processing procedures. A watermark sequence of length N is added to the largest N coefficients (except for the DC coefficient) after the global DCT transform is applied to the image. By scaling the pseudo-random sequence by a weighting factor α and the magnitude of the host coefficient, the scaled watermark symbol is added to the selected host coefficient to form the watermarked coefficient. The watermark is then retrieved by subtracting the original coefficients from coefficients of the suspected image. A correlation detector is used to calculate the similarity between the original watermark sequence and the extracted one. If the watermark embedded in the image matches with the tested watermark, a large correlation response will be generated and the watermark can thus be determined.

Piva *et al.* [61] proposed another global DCT-based method but required no original image for watermark detection. The global DCT coefficients are reordered by using the zig-zag scan. Some fixed low- to middle-frequency coefficients (e.g. 16000th to 25000th coefficients for an image of size 512 by 512) were selected for watermark embedding and retrieval to reach the balance between image fidelity and robustness. The absolute value of the coefficient was used to scale the watermark energy. A detection threshold value, which depends only on the information of the tested image, was set to determine the existence of the watermark. The embedded watermark can survive many serious distortions.

Wavelet transform is also commonly used in digital watermarking. Xia *et al.* [97] considered a multi-resolution watermarking method in the wavelet domain. The idea was a direct extension of Cox's algorithm [14, 13] to the DCT domain and the original image was needed in watermark retrieval. After the wavelet transform is applied, the watermark is detected first in higher frequency subbands. If the watermark is not detected, coefficients in lower frequency subbands are taken into account for watermark detection. Wang *et al.* [89, 90, 77] investigated a blind wavelet-based watermarking scheme, which did not require the original image for watermark detection. The inserted watermark signal was adaptively scaled by different weighting values of the subband to maintain the high quality of the watermarked image. The largest wavelet coefficient in each subband was chosen as a standard value. Other significant coefficients were truncated to the standard value multiplied by a weighting factor. Then, the watermark sequence was added to truncated coefficients to form watermarked coefficients. In watermark detection, these standard values can be accurately determined because they are perceptually significant to the image and usually remain stable after image manipulation. Therefore, the watermark can be retrieved by subtracting coefficients of a suspected image by scaled standard values. Finally, the similarity between extracted and tested watermarks is calculated. One advantage of this scheme is that the collusion attack, by which the attackers average several copies with different watermarks, will fail to generate an unwatermarked image. Several other schemes were developed based on the wavelet transform, such as [32, 38, 99, 82, 4]. The difference between these schemes usually lies in the way the watermark is weighted to decrease visual artifacts.

Note that image compression and frequency-domain watermarking share some common characteristics. In image compression, we encode significant frequency coefficients first because these coefficients convey more fundamental visual information about the image. In watermarking, we choose significant coefficients for watermark casting to enhance its robustness since these coefficients often remain stable after the attack. With this similarity, Su *et al.* [79] proposed to integrate the image watermarking method with the state-of-the-art image compression standard, JPEG-2000. Efficiency can be achieved since the most expensive computation related to the image transform has been already computed as one part of compression and decompression algorithms. The integrated watermarking scheme has various interesting properties, including progressive watermark detection and region-of-interest (ROI) watermarking. This integrated scheme will be presented in Section 3.2.

Swanson *et al.* [81] proposed a perceptual watermarking scheme, which explicitly makes use of the human visual model to embed and detect the watermark from the block-DCT domain. For each DCT block, a frequency mask is computed and scaled by the DCT of a pseudo-random sequence. This watermark is added to the corresponding DCT block. Spatial masking is then applied to ensure the invisibility of the watermark. Another image-adaptive, block-based DCT watermarking scheme was proposed by Podilchuk *et* *al.* [62, 64]. They incorporated Watson's model [91] to determine the Just Noticeable Difference (JND) of each DCT coefficient. In Watson's model, the JND value determines the maximum amount of quantization noise that can be tolerated at every frequency location without affecting the visual quality. Therefore, JND can also be used to indicate the maximum energy that the watermark can be embedded to guarantee its invisibility. The watermark is modulated onto the coefficients in the block DCT coefficients if the value of the coefficient is larger than the JND. Watermark detection is based on the correlation between the difference of the original and the investigated images.

Many spread-spectrum watermarking schemes utilized the Fourier transform, such as [54, 57, 56, 73]. The main reason of this choice is to enable the embedded watermark to survive geometrical attacks, which cause severe problems to watermark detection. In order to discuss this issue in more detail, we will explain the properties of Fourier transform, which are important in digital watermarking, and review those schemes in Section 4.2.

2.2.4.2 Spread Spectrum Watermarking in the Spatial-domain

The spread-spectrum method can also be applied in the spatial domain. The LSB scheme [87], which we mentioned before, is one of the early algorithms to utilize the correlation detector for more robust watermark detection. Wolfgang *et al.* [93] developed a watermarking scheme called the Variable-Watermark Two-Dimensional algorithm (VM2D). The one-dimensional M-sequence is extended to a two-dimensional watermark block. The watermark block is then added to the image repeatedly and detected by the matched filter. Detection of the above two watermarks does not require the original image. However, they are not robust enough since the watermark is operated in the LSB plane.

Instead of working on gray-level images, Kutter *et al.* [42] exploited humans' unequal detection of different colors and proposed to embed the watermark in the blue image component. They applied the method of amplitude modulation by adding the blue image component with the watermark bits, which are weighted by the luminance values in the spatial domain. In watermark detection, preprocessing through filtering is used to help estimate the original value for more accurate detection.

A robust block-based spatial-domain watermarking scheme was developed at the Philips Research Laboratories [34], in which the video is treated as a sequence of still images and a two-dimensional spatial watermark pattern is embedded into every frame. The pattern is embedded in the frame block by block. Each pixel is embedded with the watermark symbol scaled by a global weighting factor and a local weighting factor. In watermark detection, a filter is first applied to remove the correlation between neighboring pixels. Then, the watermark is detected by circular convolution of the filtered block with the tested pattern. More than one pattern will be embedded into a block and the distance between the peaks is used to carry the watermark payload. In this scheme, watermark detection is invariant to spatial shift.

A lot more schemes, such as [15], [17] and [26], extended the spatial-domain still image watermarking approaches to video frames for video watermarking. Besides, theoretical performance analysis of the amplitude modulation scheme in the spatial domain is done by Hernándes *et al.* [28, 27].

It should be noted that the spatial-domain watermarking still has its advantage. For the frequency-domain watermarking schemes, geometrical attacks such as cropping and rotation to an image will cause a serious synchronization problem of watermark detection.

The spatial-domain watermarking scheme may have a better chance to resist geometrical attacks because of the facts that attaining the spatial information from image pixels is much easier and that the spatial-domain watermark is embedded and detected directly in image pixels without the need of image transforms.

2.2.5 Other Information Hiding Algorithms

Quantization is the other important approach of information hiding. Chen and Wornell [11] generalized and proposed the watermarking approach using quantization and termed it as Quantized Index Modulation (QIM). QIM is based on a set of N-dimensional quantizers. The quantizers satisfy a distortion constraint and are designed such that the reconstruction values from one quantizer are separated far away from the reconstruction points of every other quantizer. The message to be transmitted is used as an index for selecting a quantizer, which embeds the information by quantizing the image data. In the decoding process, a distance metric is evaluated for all quantizers and the index of the quantizer with the smallest distance identifies the embedded information. Many schemes use the methodology of quantization for embedding high-volume data, such as [70] or for authentication purposes, such as [39].

Salient point or feature extraction may be useful in information hiding. An interesting approach is to embed the information by modifying the geometrical feature of the image [66]. A dense line pattern is pseudo-randomly generated. A set of salient points in the image is extracted. The detected points are then warped such that a significantly large number of points are within the vicinity of lines. In the detection process, the detector verifies if a large number of points are within the vicinity of lines. A few other schemes also make use of salient point or feature extraction to achieve certain objectives, such as robustness against geometrical attacks [5, 41]. The feasibility of these methods will be determined by the reliability and precision of the feature extraction process.

Some of the information hiding methods are designed to tailor such palette-based formats as GIF or PNG. Machado [51] proposed a scheme, called "EZ Stego," to hide the information in GIF compressed images. The color palette in an image is first sorted by luminance. In the reordered palette, neighboring palette entries are typically near to each other so the scheme can embed the message in the LSB of the indices pointing to the palette colors without degrading the image quality much. Fridrich *et al.* proposed a similar idea for data hiding in GIF [20]. Besides, permuting the image palette in a special order can also be used for information hiding [43]. The advantage of this method is that the appearance of the image will not be affected. But the special palette generated by the scheme may not only raise suspicion but also be vulnerable when the image is re-saved by other image editors.

One special type of information-hiding schemes embeds the data without introducing any distortion to the host image [21, 22]. The basic idea of this invertible embedding is to make use of certain redundancy in the host media data. The embedder compresses the redundancy part, such as the LSB of the image pixels, into a smaller size to leave some room for data hiding. The detector extracts the hidden data and expands the compressed redundant part to reconstruct the original data. The same idea can also be extended to several image formats [23].

Chapter 3

Information Hiding in JPEG-2000 Compressed Images

In this chapter, we deal with an interesting issue of information hiding, including digital watermarking and covert communication, in the JPEG-2000 still image compression standard. First of all, we provide a brief review of the basic architecture of JPEG-2000 in Section 3.1, which is intended to offer sufficient information to help understand concerns in designing an information-hiding scheme in this image standard. Next, we will give the detailed implementation and analysis of the proposed integrated information-hiding scheme with the JPEG-2000 compression standard. The watermark embedding and detection procedures are presented in Section 3.2. We also discuss the decision and analysis of the threshold value used for watermark detection with experimental results illustrated.

Then, we turn our attention to steganographic applications of JPEG-2000. We point out some challenges of information hiding in wavelet-based codecs and present effective schemes under the framework of JPEG-2000 in Section 3.3 to achieve covert communication. Experimental results will be shown to demonstrate the feasibility of the proposed methods. Finally, concluding remarks are given in Section 3.4.

3.1 Brief Review of JPEG-2000

The basic coding engine of JPEG-2000 is based on the Embedded Block Coding with Optimized Truncation (EBCOT) scheme proposed by Dr. Taubman. Roughly speaking, JPEG-2000 can be viewed as a block-based and bit-plane coder. By the block-based coder, we mean that the basic coding unit is a block instead of the whole image as used in coding schemes such as SPIHT [67] and EZW [68]. A bi-orthogonal wavelet transform is first applied to the image, and each subband of wavelet coefficients is divided into blocks of samples. Each block is then encoded independently to generate a separate bit-stream without resorting to any information from other blocks. The bit-stream can be truncated to a variety of discrete lengths with respect to different distortion measures. Once the entire image has been compressed, a post-processing operation passes over all the compressed blocks, and determines the extent to which each block's bit-stream should be truncated to achieve the target bit-rate. The final bit-stream is formed by concatenating the truncated bit-streams of all blocks together. JPEG-2000 is a bit-plane coding, *i.e.*, the most significant bits for all samples in the code block are handled first, then the next most significant bits and so on until all bit-planes are processed. With its block-based and bit-plane coding paradigm, JPEG-2000 can achieve several important features under the same framework.

Now, let us take a closer look at the structure of the JPEG-2000 coding standard. The block diagram of JPEG-2000 is shown in Fig. 3.1. The JPEG-2000 coding flowchart consists of a few major blocks. The major blocks in the encoder side include the forward image transform, quantization, tier-1 encoding and tier-2 encoding while those in the decoder side are composed of the tier-2 decoding, tier-1 decoding, dequantization and the inverse image transform. Basically, JPEG-2000 is a pipeline structure with a so-called "push and pull" model adopted. In the encoding process, a push model is employed. Image samples are pushed into the forward transform stage, which then pushes the transformed samples to the quantization stage as soon as those data become available. Likewise, the quantization stage pushes quantized symbol indices to the encoding stage, which pushes compressed bits to form the code stream. In the decoding process, the dual of the encoder's push paradigm, *i.e.*, a pull model, is employed. Image samples are pulled out of the reverse transform stage, which pulls transformed samples from the dequantizer. The dequantizer in turn pulls data out of the decoder, which pulls or reads the bit-stream from the compressed file. This pipeline structure also attempts to minimize the internal memory size so as to facilitate its hardware implementation. We will examine and explain the functions of each major block in the following sections.

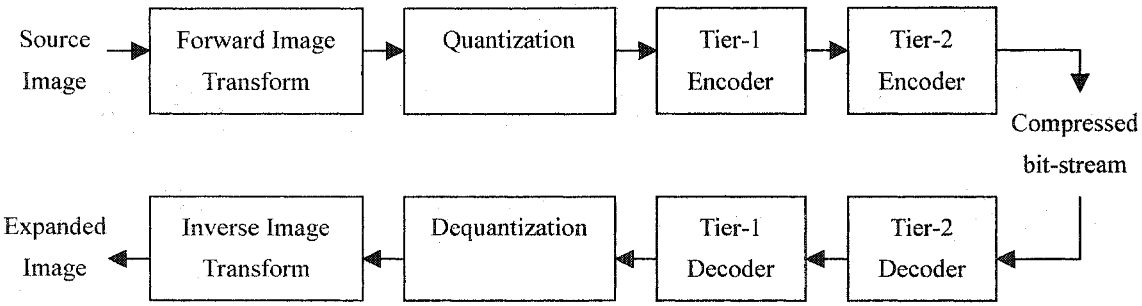


Figure 3.1: The block diagram of JPEG-2000.

3.1.1 Forward/Inverse Image Transform

The image transform stage consists of the inter-component transform and the intra-component transform. After the image is divided into tiles and certain preprocessing is applied to the tile samples so that they have a nominal dynamic range approximately centered about zero, the inter-component transform is first applied to the tile-component data to reduce the correlation between components, leading to improved coding efficiency. Two inter-component transforms are defined: the irreversible color transform (ICT) for lossy compression and the reversible color transform (RCT) usually for lossless compression. Both of the transforms map the image data from the RGB domain to the YCrCb color space.

Following the inter-component transform in the encoder is the intra-component transform, *i.e.* the wavelet transform. Through the wavelet transform, tile components are decomposed into different resolution levels, which contain a number of subbands. Both reversible interger-to-interger and irreversible real-to-real wavelet transforms are available in JPEG-2000. The irreversible transform is implemented by means of the Daubechies 9/7 filter while the reversible transform is implemented by means of the 5/3 filter. Two filtering modes, *i.e.* the convolution-based and the lifting-based modes, are supported.

The inverse intra-component and inter-component transforms in the decoder essentially undo the effect of the forward transforms in the encoder. Unless the transforms are reversible, the inversion may only be approximate due to the finite precision arithmetic effect.

3.1.2 Quantization/Dequantization

In the encoder, after inter-component and intra-component transforms, the resulting coefficients are quantized. Quantization helps to achieve better compression by representing transform coefficients with the minimum precision required for the desired level of image quality. Transform coefficients are quantized using scalar quantization with a deadzone. A different quantizer is employed for each subband with the quantizer step size as its only parameter. The quantization process can be written as

$$V(x,y) = \lfloor \frac{|U(x,y)|}{\triangle} \rfloor \times sgn(U(x,y)), \tag{3.1}$$

where $\triangle$ is the quantizer step size, $U(x,y)$ is the input subband signal, and $V(x,y)$ is the output quantizer index for the subband. The baseline codec has two modes of operation, *i.e.* integer mode and real mode. Lossless compression is always operated in the integer mode, in which the quantizer step sizes are always fixed at one to effectively bypass the quantization. In the real mode, the quantizer step sizes are chosen in conjunction with rate control. The step sizes used by the encoder are conveyed to the decoder via the code stream. The step sizes signaled are not absolute but relative quantities. That is, the quantizer step size for each subband is specified relative to the nominal dynamic range of the subband signal. In the decoder, the dequantization stage approximately reverses the effect of quantization to obtain the quantized transform coefficient. The dequantization process can be written as

$$U(x,y) = \{V(x,y) + 0.5 \times sgn(V(x,y))\} \times \triangle, \tag{3.2}$$

where $V(x, y)$ is the input quantizer index for the subband, and $U(x, y)$ is the reconstructed subband signal. It should be noted that quantization of transform coefficients is one of the two major sources of information loss in the coding path of JPEG-2000.

3.1.3 Tier-1 Coding

At the encoder side, the quantization stage is followed by the tier-1 encoding. The quantizer indices for each subband are partitioned into code blocks. Code blocks are rectangular in shape with the same dimension, except at image boundaries where some blocks may have smaller dimensions. The nominal size of blocks is a free parameter with certain constraints. A typical choice for the nominal code block size is 64×64. After a subband has been partitioned into code blocks, each of the code blocks is independently coded. The coding is performed using the bit-plane coder and each bit-plane is processed by several coding passes. Therefore, the output of the tier-1 encoding process is a collection of coding passes for the code blocks.

At the decoder side, the bit-plane coding passes for the code blocks are input to the tier-1 decoder, these passes are decoded, and the resulting data are assembled into subbands. In lossy compression, the reconstructed quantizer indices may only be approximations to the quantizer indices at the encoder since the code stream may only include a subset of the coding passes generated by the tier-1 encoding process. In the lossless case, the reconstructed quantizer indices must be the same as the original ones at the encoder side and all coding passes must be included for lossless coding.

In each bit-plane, there are three coding passes: 1) significance, 2) refinement and 3) cleanup pass. All of the three coding passes scan the samples of a code block in the same fixed order. The code block is partitioned into stripes with the nominal height of four samples. If the code block height is not a multiple of four, the height of the bottom stripe will be less than this nominal value. The bit-plane encoding process generates a sequence of symbols for each coding pass. All of the symbols are either entropy coded or raw coded. For entropy coding, a context-based adaptive binary arithmetic coder, or more specifically, the MQ coder, is employed. For raw coding, the binary symbols are emitted as raw bits with simple bit stuffing. Both the entropy and raw coding processes ensure that certain bit patterns never occur in the output, allowing such patterns to be used for the error resilience purpose. Next, we examine the three coding passes in more detail.

1. Significance Pass

   The first coding pass for each bit-plane is the significance pass. This pass is used to convey significance with its sign information for samples that have not yet been found to be significant and are predicted to become significant during the processing of the current bit plane. A sample is predicted to become significant if any 8-connected neighbor has already been found to be significant. The symbols generated during the significance pass may or may not be arithmetically coded. If arithmetic coding is employed, the binary symbol conveying significance information is coded using one of nine contexts. The particular context used is selected based on the significance of the sample's 8-connected neighbors and the orientation of the subband with which the sample is associated. If the arithmetic coding is used, the sign of a sample is coded as the difference between the actual and predicted sign value. Otherwise, the sign information is coded directly. Sign prediction is performed using the significance and the sign information for 4-connected neighbors.

2. Magnitude Refinement Pass

   The second coding pass for each bit-plane is the magnitude refinement pass. This pass signals subsequent bits after the most significant bit for each sample. If a sample was found to be significant in a previous bit plane, the next most significant bit of that sample is conveyed using a single binary symbol. Like the significance pass, the symbols of the magnitude refinement pass may or may not be arithmetically coded. If arithmetic coding is adopted, each refinement symbol is coded using one of three contexts. The particular context is selected based on if the second MSB position is being refined and the significance of 8-connected neighbors.

3. Cleanup Pass

   The third coding pass for each bit-plane is the cleanup pass. This pass is used to convey significance and its sign information for those samples that have not yet been found to be significant and are predicted to remain insignificant during the processing of the current bit-plane. The key difference between the cleanup and significance pass is that the cleanup pass conveys information about samples that are predicted to remain insignificant, rather than those that are predicted to be significant. The other important difference is that the samples in cleanup passes are not processed individually but sometimes processed in groups. As we mentioned earlier, a code block is partitioned into stripes with a nominal height of four samples. Then, stripes are scanned from top to bottom, which we refer to as a vertical scan. If

> the vertical scan contains four samples and all of the samples are predicted to remain insignificant, the so-called "aggregation mode" is entered. When this occurs, the four samples of the vertical scan are examined. If all four samples are insignificant, an all-insignificant aggregation symbol is coded, and the processing of the vertical scan is complete. Otherwise, a some-significant aggregation symbol is coded, and two binary symbols are then used to code the number of leading insignificant samples in the vertical scan. The symbols generated during the cleanup pass are always arithmetically coded. When the aggregation mode is not employed, the significance and the sign coding functions as in the case of the significance pass.

To sum up, cleanup passes always employ arithmetic coding. In the case of significance and refinement passes, two possibilities exist, depending on whether the so-called "lazy mode" is enabled. If the lazy mode is enabled, only the significance and refinement passes for the four most significant bit-planes use arithmetic coding, while the remaining such passes are raw coded. Otherwise, all significance and refinement passes are arithmetically coded. The lazy mode significantly reduces the computational complexity of bit-plane coding by decreasing the number of symbols that must be arithmetically coded. The cost of lazy mode coding is reduced coding efficiency at low bit-rate compression. Consecutive coding passes that use the same encoding scheme (*i.e.*, arithmetic or raw coding) constitute a "segment." All of the coding passes in a segment can collectively form a single codeword or each coding pass can form a separate codeword as determined by the termination mode. Two termination modes are supported: per-pass termination and per-segment termination. In the first case, all coding passes are terminated. In the second case, only the last coding pass of a segment is terminated. Terminating all coding passes facilitates improved error resilience at the expense of decreased coding efficiency.

3.1.4 Tier-2 Coding

The tier-1 encoding is followed by the tier-2 encoding. The input to the tier-2 encoding process is the set of bit-plane coding passes generated during the tier-1 encoding. The coding passes are packaged into data units called packets, in a process referred to as packetization. The resulting packets are then output to the final code stream. Each packet is comprised of two parts: a header and a body. The header indicates which coding passes are included in the packet, while the body contains the actual coding pass data. The packetization process imposes a particular organization on coding pass data in the output code stream. This organization facilitates many of the desired features including rate scalability and progressive recovery by fidelity or resolution. Rate scalability is achieved through quality layers. Each coding pass is either assigned to one of the layers or discarded. The coding passes containing the most important data are included earlier in the lower layers, while the coding passes associated with finer details are included later in the higher layers. During decoding, the reconstructed image quality improves incrementally with each successive layer processed. Since some coding passes may be discarded in the case of lossy compression, the tier-2 coding is the second primary source of information loss in the coding path.

In the tier-2 coding, one packet is generated for each component, resolution level, layer, and precinct. A precinct is essentially a group of code blocks within a subband. The precinct partitioning for a particular subband is derived from a partitioning of its parent LL band (*i.e.*, the LL band at the higher resolution level). Each resolution level has a nominal precinct size. Each of the resulting precinct regions is then mapped into its child subbands at the next lower resolution level. Precinct boundaries always align with code block boundaries. Since coding pass data from different precincts are coded in separate packets, using smaller precincts reduces the amount of data contained in each packet, which leads to improved error resilience, while the coding efficiency is degraded due to the increased overhead of more packets. It should be noted that a packet can be empty. Empty packets are sometimes necessary since a packet must be generated for every component-resolution-layer-precinct combination even if the resulting packet conveys no new information. In the decoder, the tier-2 decoding process extracts the coding passes from the code stream in a process referred to as depacketization and then associates each coding pass with its corresponding code block.

The major task of the tier-2 coding is to achieve rate control by selecting subsets of coding passes to include in the code stream. Given a target bit-rate, each of the independent code block bit-streams is truncated in an optimal way so as to minimize distortion subject to the bit-rate constraint. The idea is referred to as Post-Compression Rate-Distortion (PCRD) optimization since the rate control is applied after all the subband samples have been compressed in the tier-1 coding. The advantages of PCRD optimization is its reduced complexity. To generate an embedded bit-stream without PCRD, we may have to process the image several times and a large buffer is required to store the whole image, which could not be accessible in certain applications. With PCRD, the image needs only be compressed once to generate an out-of-order bit-stream, which is generally much smaller than the original image and hence can be buffered easier. The rate control is then applied on this compressed bit-stream to generate the embedded bit-stream.

3.1.5 Rate Control

We briefly describe the rate control mechanism of JPEG-2000 as follows. For each code block, B_i, the embedded bit-stream is truncated to the rates, $R_i^{n_i}$. The contribution from B_i to the distortion improvement in the reconstructed image is denoted by $D_i^{n_i}$, for each truncation point, n_i. If an additive distortion metric that approximates Mean Squared Error (MSE) is assumed, the overall image distortion, D, can be calculated by,

$$D = \sum_i D_i^{n_i} = \sum_i \left\{ \omega_{b_i}^2 \sum_{\mathbf{k} \in \mathbf{B_i}} (s_i^{n_i}[\mathbf{k}] - s_i[\mathbf{k}])^2 \right\}, \tag{3.3}$$

where $s_i[\mathbf{k}]$ denotes the two-dimensional sequence of subband samples in the code block B_i, $s_i^{n_i}[\mathbf{k}]$ denotes the quantized representation of these samples associated with truncation point n_i, and ω_{b_i} denotes the L2-norm of the wavelet basis function for the subband, b_i, to which the code block belongs. This approximation is valid provided the wavelet transform's basis functions are orthogonal and the quantization errors in each of the samples are uncorrelated.

The truncation points, n_i, are selected to minimize distortion subject to a constraint, R^{max}, on the available bit-rate, *i.e.*

$$R^{max} \geq R = \sum_i R_i^{n_i}. \tag{3.4}$$

Any set of truncation points, n_i^λ, which minimizes

$$D(\lambda) + \lambda R(\lambda) = \sum_i (D_i^{n_i^\lambda} + \lambda R_i^{n_i^\lambda}) \tag{3.5}$$

for some λ, is optimal in the sense that the distortion cannot be reduced without increasing the overall rate and vice versa. Thus, a value of λ can be found such that the truncation points which minimize (3.5) yield $R(\lambda) = R^{max}$, then this set of truncation points must also be an optimal solution to the R-D optimization problem. Since we only have a discrete set of truncation points, it may not be possible to find a value of λ for which $R(\lambda)$ is exactly equal to R^{max}. However, since the code blocks are relatively small and there are many truncation points, it is sufficient to find the smallest value of λ such that $R(\lambda) \leq R^{max}$.

The optimal truncation points, n_i^λ, for any given λ, can be determined efficiently based on a small amount of summary information collected during the generation of each code block's embedded bit-stream. A simple algorithm to find the truncation point, n_i^λ, which minimizes $D_i^{n_i^\lambda} + \lambda R_i^{n_i^\lambda}$, is as follows:

```
1) Initialize $n_i^\lambda = 0$
2) For j=1,2,3, ...,
    - Set $\triangle R_i^j = R_i^j - R_i^{n_i^\lambda}$ and $\triangle D_i^j = D_i^{n_i^\lambda} - D_i^j$.
    - If $\triangle D_i^j / \triangle R_i^j > \lambda$ then update $n_i^\lambda = j$
```

Since this algorithm has to be executed for many different values of λ, a set of feasible truncation points, N_i, should be determined first. Let $j_1 < j_2 < j_3 ...$ be an enumeration of these feasible truncation points and let the corresponding distortion-rate "slopes" be given by $S_i^{j_k} = \triangle D_i^{j_k} / \triangle R_i^{j_k}$ where $\triangle R_i^{j_k} = R_i^{j_k} - R_i^{j_{k-1}}$ and $\triangle D_i^{j_k} = D_i^{j_{k-1}} - D_i^{j_k}$. These slopes should be strictly decreasing. If $S_i^{j_{k+1}} \geq S_i^{j_k}$, then these truncation point, j_k, will not be selected by the above algorithm, regardless of the value of λ. When restricted to a set of truncation points whose slopes are strictly decreasing, the above algorithm reduces to the selection $n_i^\lambda = max\left\{j_k \in N_i | S_i^{j_k} > \lambda\right\}$ so that each such point must be a valid candidate for some value of λ. The set of feasible truncation points, N_i, can be determined using a conventional convex hull analysis immediately after the bit-stream for B_i has been generated. The rates, $R_i^{j_k}$ and slopes $S_i^{j_k}$, for each $j_k \in N_i$, are kept in a compact form along with the embedded bit-stream until all code blocks have been compressed. The search for the optimal λ and n_i^λ can thus be proceeded in a straightforward manner.

3.1.6 Region of Interest Coding

Region of interest (ROI) coding is one interesting feature supported by JPEG-2000. ROI coding makes it possible to encode regions in which users are more interested with better quality than the rest of the image. For the extreme case, the specified ROI can be encoded losslessly while the remaining parts of the image are encoded with low bit-rates. When ROI is small compared with the whole image, the transmission time and the storage space can be greatly saved.

In JPEG2000, two types of ROI functionality are defined. The first one is "ROI during encoding," in which ROI is specified when the image is compressed. The other one is "ROI during decoding" that supports interactive browsing. In JPEG-2000 VM, the "ROI during encoding" mode is implemented. The implementation is based on the maximum shift method. The principle is to scale or shift coefficients so that the bits associated with the ROI are placed in higher bit-planes than the bits associated with the background.

During the embedded coding process, the most significant ROI bit-planes will be placed in the bit-stream before any background bit-planes of the image. Thus the ROI will be decoded or refined before the rest of the image. If the bit-stream is truncated without fully decoded, the ROI will be of higher fidelity.

Therefore, the implementation of ROI in JPEG-2000 is shown as follows:

1. The wavelet transform of the image is calculated.
2. After the ROI is chosen in the image domain, a ROI mask is derived as shown in Fig. 3.2 to indicate the set of coefficients that are required for ROI.
3. The wavelet coefficients are quantized and the quantized coefficients are stored in a sign magnitude representation.
4. Coefficients outside the ROI are downscaled by a specified scaling value.
5. The resulting coefficients are encoded.

The decoder reverses these steps to reconstruct the image. It should be noted that the scaling value assigned to the ROI and the coordinates of the ROI are added to the bit-stream. The decoder also performs the ROI mask generation but scales up the background coefficients.

3.2 Digital Watermarking in JPEG-2000 Compressed Images

3.2.1 Framework for Watermarking

Before presenting the implementation of the proposed watermarking scheme, we briefly describe the basic framework of the watermarking method to be adopted. Similar to most

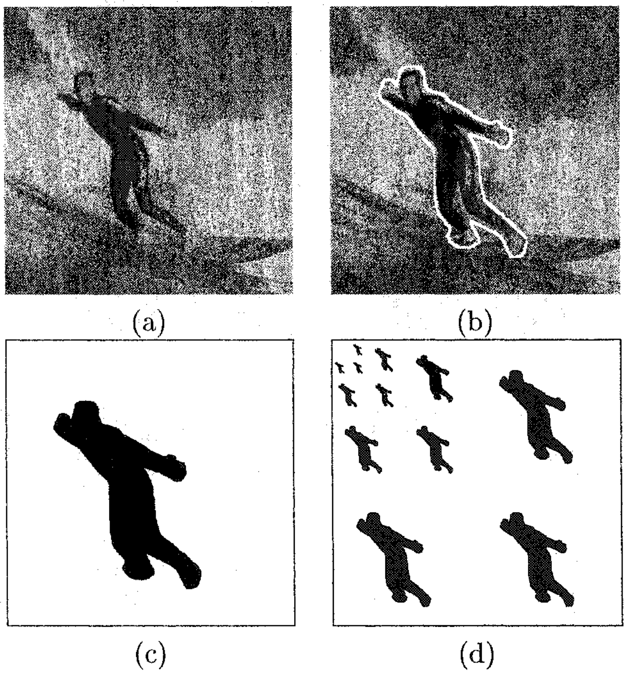


Figure 3.2: (a) The image to be compressed, (b) the contour of ROI, (c) the spatial-domain ROI mask, and (d) the wavelet-domain ROI mask.

robust watermarking schemes given in Section 2.2, the additive spread-spectrum watermarking method is chosen due to its decent characteristics in robustness, unobtrusiveness and security to watermarking applications. After a proper transform (the wavelet transform in our scheme) is applied to the image, the watermark is added onto the selected frequency coefficient by

$$I^{'}(x,y) = I(x,y) + \alpha(x,y) \times W(x,y), \tag{3.6}$$

where $I^{'}(x,y)$ is the watermarked coefficient and $I(x,y)$ is the original coefficient with the coordinate (x,y) in the spatial position. $I(x,y)$ is chosen based on its magnitude, *i.e.*, the coefficient with the large magnitude is selected for watermark embedding. $W(x,y)$ is the corresponding watermark symbol, which can be a real number or only take values, 1 and

-1. The weighting factor $\alpha(x, y)$ is a positive number used to adjust the amount of added watermark energy. The value of α is usually adjusted according to the magnitude of the frequency coefficients or the different subband characteristics so that the balance between robustness and fidelity of the resulting watermarked image can be achieved. The inverse transform is then applied to form the watermarked image.

In watermark detection, a correlation detector is used to determine if the watermark exists in the tested image. It is based on the fact that if the coefficients and the watermark sequence are independent, the inner product of the watermark and the coefficient sequences will be close to 0. If the target watermark sequence is added to the coefficient sequence, we will get a peak response from the inner product. We show this basic idea as follows; $I^*(x, y)$ is the wavelet coefficient of the suspected image. We make use of the correlation detector to determine if the wavelet coefficients are embedded with a specific watermark sequence $W^*(x, y)$. The correlation response ρ of the watermark detector can be expressed as

$$\rho = \sum_{(x,y)} (I^*(x, y) \times W^*(x, y)) \tag{3.7}$$

By assuming that $I^*(x, y)$ is formed by casting watermark symbol $W(x, y)$ onto the original coefficient $I(x, y)$ without any modification, then we can express ρ as

$$\begin{aligned} \rho &= \sum_{(x,y)} ((I(x, y) + \alpha(x, y) \times W(x, y)) \times W^*(x, y)) & (3.8) \\ &= \sum_{(x,y)} (I(x, y) \times W^*(x, y)) + \sum_{(x,y)} (\alpha(x, y) \times W(x, y) \times W^*(x, y)) & (3.9) \end{aligned}$$

After calculating the expected value of both sides, we get

$$\mathcal{E}[\rho] = \mathcal{E}\left[\sum_{(x,y)} (I(x,y) \times W^*(x,y))\right] + \mathcal{E}\left[\sum_{(x,y)} (\alpha(x,y) \times W(x,y) \times W^*(x,y))\right], \quad (3.10)$$

where $\mathcal{E}[\cdot]$ is the expected value. The first term on the right-hand side of (3.10) is zero if the tested watermark sequence $W^*(x,y)$ and the coefficients $I(x,y)$ are independent. Similarly, the second term is also zero if $W(x,y)$ and $W^*(x,y)$ are independent or $W(x,y)$ does not exist. If the image is embedded with $W^*(x,y)$, *i.e.*, $W(x,y) = W^*(x,y)$, then the expected value of the correlation response $\mathcal{E}[\rho]$ will be close to

$$\mathcal{E}[\rho] = \mathcal{E}\left[\sum_{(x,y)} \alpha(x,y) \times W^{*^2}(x,y)\right], \quad (3.11)$$

which is much larger than zero. Therefore, we can simply examine the peak response and compare it with a threshold value to determine the existence of watermark without any difficulty.

In the proposed system, the watermark is embedded after coefficients are quantized so that the watermark can be easily embedded into the bitstream. Watermark detection is done before the dequantization stage. We choose the non-reversible kernel for watermark embedding and detection in the following discussion since lossy compression offers a broader application scope than the lossless one. The same idea can however be applied to the reversible kernel without modification.

3.2.2 Watermark Embedding

To make the embedded watermark robust against attacks, significant coefficients are chosen for watermark casting. Significant coefficients are those with a larger magnitude. Because coefficients in each subband have been normalized to have unit gain in JPEG-2000 implementation, the significant coefficients in each subband tend to have their highest non-zero bit in the same bit-plane. Therefore, we can apply the same watermark embedding rule in each subband. JPEG-2000 divides the subband into blocks which is the basic coding unit so that we also use the coding block as the basic unit for watermarking.

In JPEG-2000 implementation, the original I-bit image samples are level shifted to a nominal range of -2^{I-1} to 2^{I-1} and then shifted up by $P - I - G$ bits to fit within the P-bit implementation precision. G is the number of guard bits, which is included to avoid the occurrence of overflow so that the frequency coefficients can be represented with the fixed-point precision. The wavelet transform kernels are then normalized so that the low-pass analysis filters always have a unit DC gain and the high-pass analysis filters always have a unit Nyquist gain. This means that the nominal range of subband coefficients will be in the range of -2^{P-G-1} to 2^{P-G-1}. In the representation of the coefficient, the Most Significant Bit (MSB) of the coefficient indicates the sign value and the remaining $P - 1$ bits represent the absolute magnitude of the coefficient. This simplifies the case when the magnitude is required only, because we can avoid the calculation of the absolute value.

Therefore, we select significant coefficients by examining the highest non-zero bit (not including the sign bit) that is higher than a certain bit-plane with index q. That is, coefficient $I_{b_s}(x,y)$ in the block b_s of subband s with the coordinate (x,y) will be chosen for watermark embedding if

$$\|I_{b_s}(x,y)\| \geq 2^q, \tag{3.12}$$

where we define that the Least Significant Bit (LSB) of coefficients form the bit-plane with index "0." The strategy to select coefficients matches the bit-plane coding well since coefficients to be coded earlier will be cast with the watermark first. The watermark in our scheme is a random number sequence taking two values 1 and -1. First, a seed, which can be viewed as a user ID number, is used to generate the watermark sequence with the length equal to the number of coefficients in a coding block. The sequence forms a watermark map with a dimension equal to that of a block. For a block of size $\gamma \times \gamma$, the watermark map is $W_{b_s}(x,y)$, where $x, y \in [0, \gamma)$, and the coefficient $I_{b_s}(x,y)$ that satisfies (3.12) is modified to $I'_{b_s}(x,y)$ by

$$I'_{b_s}(x,y) = I_{b_s}(x,y) + (W_{b_s}(x,y) \times 2^{\delta_{b_s}}) \tag{3.13}$$

where $W_{b_s}(x,y) = \pm 1$ is the watermark element in the position (x,y) on the watermark map associated with block b_s and δ_{b_s} is the number of bit-shift that depends on the implementation precision P and the watermark energy. In general, the shifted number is chosen to be

$$\delta_{b_s} = P - I - G + \alpha_{b_s}. \tag{3.14}$$

As mentioned earlier, P, G and I are the implementation precision, the guard bit, and the image sample precision, respectively, and α_{b_s} is the watermark scaling factor. We may increase the embedded watermark energy by shifting the watermark a few bits to the left. It should be noted that α_{b_s} can vary in different subbands or blocks so that we can adjust it according to different subband or block characteristics. Generally, only bitplanes around $P-I-G+\alpha_{b_s}$ will be affected by watermark embedding so the bitplane-based watermark embedding method does not affect the coding efficiency much. Besides, by experiments, the setting of δ_{b_s} achieves a pretty good balance between image quality and robustness of the watermark. Special care must be taken that we do not cast the watermark in blocks of the DC band since it may lead to serious fidelity degradation in the watermarked image.

3.2.3 Watermark Detection

As done in the embedding procedure, we only pick coefficients that satisfy (3.12) for watermark detection. We use the same bit-plane as a reference so that only the coefficients that are possibly embedded with watermark are taken into consideration. If we embed and detect the watermark in all of the wavelet coefficients, a similar objective might be achieved, but the watermarking process will be less effective. (The computational load will increase a lot especially when we have to test a lot of watermark sequences to see which one is embedded in the image).

However, we have to take the significance of different subbands into account during watermark detection. A value called "extra LSB" and denoted by β_{b_s} is determined along with the JPEG-2000 normalization process and can be interpreted as the number of insignificant bit-planes in the coding block b_s. β_{b_s} is smaller in subbands that need better precision and larger in high-frequency subbands. Thus, the calculation of correlation response is done as

$$\rho = \frac{\sum_{s} \sum_{b_s} \sum_{\substack{(x,y) \\ (\|I^*_{b_s}(x,y)\| \geq 2^q)}} (I^*_{b_s}(x,y) \times 2^{-\beta_{b_s}}) \times (W_{b_s}(x,y) \times 2^{(\delta_{b_s}-\beta_{b_s})})}{\sum_{s} \sum_{b_s} \sum_{\substack{(x,y) \\ (\|I^*_{b_s}(x,y)\| \geq 2^q)}} 2^{(2\delta_{b_s}-2\beta_{b_s})}}, \tag{3.15}$$

where δ_{b_s} is defined in (3.14). $I^*_{b_s}$ is the coefficient of the investigated image. Note that there is a difference between (3.7) and (3.15). In (3.15), we normalize the correlation response by the sum of squares of selected watermark symbols. Since the watermark symbol takes value 1 or -1, $W^2_{b_s}(x,y)$ is equal to 1 and omitted in the denominator. There are a couple of reasons that we decide to normalize the correlation response. First of all, the value of the response will not be affected by the number of selected coefficients. Consequently, the same correlation response can be used in different scenarios, e.g. normal watermark detection and progressive watermark detection, which will be discussed in Section 3.2.4. Second, this normalization process will help in explaining the high value of the correlation response of the watermarked image in our scheme. That is, the value of (3.7) in a watermarked image will be even larger than (3.11). This phenomenon will be discussed in Section 3.2.6.

3.2.4 Progressive Watermark Detection

Progressive watermark detection is one of the most attractive features for watermarking in JPEG-2000 compressed images. When a large image is being decompressed, it is not efficient to detect the watermark after the whole image is formed. This is especially true for Internet applications. A fully-embedded compression scheme lets the user truncate the image at any time to get his or her "best" image. Thus, it is desirable that the watermark can also be detected progressively. JPEG-2000 is a bit-plane coder which can support the fully-embedded feature. Significant coefficients, which have been embedded with the watermark in our scheme, will be encoded and decoded first so that progressive watermark detection can be achieved easily. However, we should set a threshold value η to indicate the minimum number of coefficients needed for watermark detection. If the correlation response is higher than the current threshold, which is used to decide the existence of watermark, but the selected coefficients are less than η, the detection process should continue in other blocks to avoid possible false alarm. The derivation of the threshold in the proposed watermarking scheme will be discussed in Section 3.2.6.

3.2.5 Region of Interest Watermark

Most watermarking techniques embed the watermark in the entire image without taking the image content into account. For many applications, a certain portion of an image is more important than others. Especially, for object-oriented images, regions that cover the main objects are of major concern to the image owner. For example, in a picture with a person appearing at the center, the image viewer usually cares more about the person than the background of the picture. The portion that attracts more attention of an image viewer is the Region of Interest, *i.e.*, ROI. It is desirable to embed a more robust watermark in ROI to give it better protection [78].

ROI is usually selected by image owners or users in the spatial domain. After selecting ROI in an image, it is straightforward to embed the watermark in the spatial domain, *i.e.*, modifying values of image pixels directly so that we can decide as our will on what portions of the image should be embedded with what kind of watermarks. It is worthwhile to point out that there may be different ROI in the same image depending on different applications. However, in order to achieve a robust watermark, frequency-domain is preferred for watermark embedding. Thanks to the combined spatial-frequency characteristic of wavelet transform, we can make use of the spatial self-similarity between wavelet subbands to determine the coefficients belonging to ROI for watermark embedding.

In addition to copyright enforcement, ROI watermarking can also be used as data labeling to assist content retrieval in image databases. In image archiving, high performance data representations and structures are essential to image database management. By using traditional database indexing methods, it is not easy to index the location, size and relationships of the objects in the image. In the newly-proposed database indexing techniques, the objects are extracted by low-level feature extraction or segmentation. Nevertheless, it is very difficult to achieve perfect image segmentation and image understanding methods may need to be included to improve the performance of object extraction. Related techniques of image understanding are not mature yet for object retrieval. ROI watermarking could bridge the gap between the traditional and newly-proposed indexing methods.

As illustrated in Section 3.1.6, to achieve ROI coding, the encoder keeps the coefficients that belong to ROI unchanged while downscales the other coefficients that do not belong to ROI by a few bits. The encoding process is done as usual while the coordinates of ROI and scaling values are put in the bit-stream header for transmission. When the SNR progressive mode is used, ROI will be sent before the background. The decoder can detect ROI by examining the magnitude of received coefficients since all ROI coefficients are larger than other coefficients outside ROI. The decoder may have to upshift the received coefficients when necessary. Under this scenario, we do not have to change the proposed algorithm because only coefficients in ROI with a larger magnitude will be embedded with the watermark. All coefficients outside ROI will be downshifted so that they will not satisfy the criterion in (3.12) for watermark embedding and retrieval. To conclude, our scheme can support ROI watermarking automatically.

3.2.6 Threshold Decision and Analysis

It is essential to determine a threshold value so that the existence of a watermark sequence can be detected by comparing the value of the correlation response with the selected threshold value. There are two main parts in this section. First of all, we determine the threshold value to decrease the possibility of false positive detection. False positive detection occurs when the watermark is falsely detected in an image that contains no watermark or the wrong watermark ID is detected. Since the usage of the image will be limited once a certain watermark is found, false positive detection will bring much more inconvenience to the legitimate users. Therefore, the threshold value should be decided carefully. Second, we examine the peak correlation response of the watermarked image and show that the existence of the watermark can generate a large correlation response.

In watermark detection, we compute the sum of multiplications of the shifted coefficients with the corresponding watermark symbol in the watermark map, and then divide it by a weighting factor, *i.e.*, the weighted norm of the watermark sequence as indicated in (3.15). Here, we assume that the correlation response ρ follows the Gaussian distribution due to the Central Limit Theorem.

The variable ρ in (3.15) has a mean equal to zero if the watermark does not exist. The variance of ρ can be estimated by

$$\sigma_\rho^2 \simeq \frac{\displaystyle\sum_{s}\sum_{b_s}\sum_{\substack{(x,y)\\(\|I^*_{b_s}(x,y)\|\geq 2^q)}} {I^*_{b_s}}^2(x,y) \times 2^{(2\delta_{b_s}-4\beta_{b_s})}}{\left\{\displaystyle\sum_{s}\sum_{b_s}\sum_{\substack{(x,y)\\(\|I^*_{b_s}(x,y)\|\geq 2^q)}} 2^{(2\delta_{b_s}-2\beta_{b_s})}\right\}^2}, \tag{3.16}$$

where all entities in above are the same as those in (3.15).

By definition, the Gaussian Integral Function [71] (or simply the Q function) can be written as

$$Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}}\, e^{\frac{-x^2}{2}} dx. \tag{3.17}$$

If random variable $Y(u)$ follows the Gaussian distribution with mean m and variance σ^2, the probability that $Y(u) > a$ can be expressed as

$$Pr\{Y(u) > a\} = Q(\frac{a-m}{\sigma}). \tag{3.18}$$

We should set up the threshold value according to the Q function so that the false alarm rate is lower than a given probability. As a result, the threshold value is actually a function of the variance of ρ. Here, we simply define the threshold as

$$T = \tau \times \sigma, \tag{3.19}$$

where τ is a scaling parameter. On one hand, we can lower the false alarm rate by raising the τ value. On the other hand, we can reduce the τ value so that detection of the embedded watermark can be more easily achieved even under very serious attacks at the expense of a higher false alarm rate. For example, if the desired false alarm rate is around 10^{-12}, we should choose the threshold value as $T = 7\sigma$ since

$$Pr\{Y(u) > T\} = Q(\frac{T}{\sigma}) = Q(7) = 1.28 \times 10^{-12}. \tag{3.20}$$

In progressive watermark detection, the probability of false alarm can be larger because the number of the selected coefficients may not be large enough. We choose a larger τ to get a higher threshold to lower the probability of false alarm as much as possible. It is also possible to adjust the τ value to adapt to different detection situations.

After setting up the threshold, we would like to analyze the peak value of the correlation response when a certain watermark is found to exist in an image. To simplify the analysis, it is assumed that the extra LSB β_{b_s} and the bit-shift number δ_{b_s} are both equal to δ in all blocks. If the watermark exists, (3.15) can be modified as

$$\rho = \frac{\sum\left\{(I_d + W_d \times 2^{\delta}) \times 2^{-\delta} \times W_d\right\}}{\sum(W_d \times W_d)} = \frac{\sum(I_d \times W_d)}{\sum(W_d \times W_d)} \times 2^{-\delta} + 1, \tag{3.21}$$

where W_d and I_d are the watermark symbol and the original coefficient selected by the watermark detector. To be more precise, we calculate the expected value on both sides as

$$\mathcal{E}[\rho] = \mathcal{E}\left[\frac{\sum(I_d \times W_d)}{\sum(W_d \times W_d)}\right] \times 2^{-\delta} + 1. \tag{3.22}$$

As discussed in Section 3.2.1, one may think that the first term on the right-hand side of (3.22) is zero so that the expected value of the maximal correlation peak is unity.

However, the peak can be substantially larger than 1 if a certain watermark exists in the image as argued below. Let I_e denote the original coefficient selected by the watermark embedder, *i.e.*, the coefficient satisfies (3.12), and W_e be its respective watermark symbol, *i.e.*, the watermark symbol to be cast on the selected coefficient. First, we calculate the expected cumulative sum of $I_e \times W_e$:

$$\mathcal{E}[R_e] = \mathcal{E}\left[\sum (I_e \times W_e)\right]. \tag{3.23}$$

Note that $\mathcal{E}[R_e] = 0$ because I_e and W_e are independent. Let us divide $\mathcal{E}[R_e]$ into two parts, *i.e.*,

$$\mathcal{E}[R_e] = \mathcal{E}[R_{e_1}] + \mathcal{E}[R_{e_2}] = \mathcal{E}\left[\sum (I_{e_1} \times W_{e_1})\right] + \mathcal{E}\left[\sum (I_{e_2} \times W_{e_2})\right]. \tag{3.24}$$

I_{e_2} is the coefficient satisfying both of the following conditions:

$$2^q \leq \|I_{e_2}\| < 2^q + 2^\delta, \quad q > \delta, \tag{3.25}$$

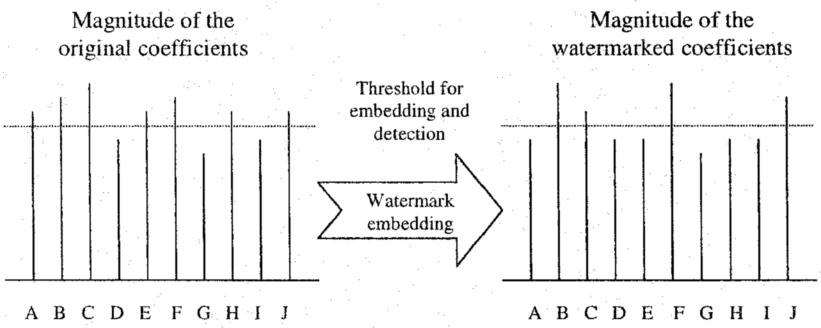
and

$$I_{e_2} \times W_{e_2} < 0, \tag{3.26}$$

where W_{e_2} is the respective watermark symbol of I_{e_2}. Obviously, $\mathcal{E}[R_{e_2}]$ is a negative number so that $\mathcal{E}[R_{e_1}]$ has to be positive to make $\mathcal{E}[R_e]$ equal to 0. However, owing to the process of watermark detection, coefficients I_{e_2} in R_{e_2} will not be picked by the detection process in watermark retrieval since its highest non-zero bit is $q - 1$, which is lower than

q. Therefore, the expected value of the correlation response calculated in the detection process is equal to

$$\mathcal{E}[\rho] = \frac{\mathcal{E}[R_e] - \mathcal{E}[R_{e_2}]}{\mathcal{E}\left[\sum W_{e_2}^2\right]} \times 2^{-\delta} + 1 = \frac{\mathcal{E}[R_{e_1}]}{\mathcal{E}\left[\sum W_{e_2}^2\right]} \times 2^{-\delta} + 1 > 1 \tag{3.27}$$


Magnitude of the original coefficients
Threshold for embedding and detection
Magnitude of the watermarked coefficients
Watermark embedding
A B C D E F G H I J
A B C D E F G H I J

Coeff.	Sign of Coeff.	Sign of watermark	Selected for embedding?	Change of magnitude	Selected for detection?
A	+	-	Yes	-	No
B	+	+	Yes	+	Yes
C	-	+	Yes	-	Yes
D	+	-	No	No change	No
E	-	+	Yes	-	No
F	+	+	Yes	+	Yes
G	-	-	No	No change	No
H	-	+	Yes	-	No
I	+	+	No	No change	No
J	-	-	Yes	+	Yes

Figure 3.3: Extra correlation gain from coefficient selection.

The number of coefficients I_{e_2} is quite large so that the first term of the right-hand side in (3.27) can be larger than 1 to generate a peak value ρ that could be even equal to 2 when the watermark exists in the image. Therefore, the first term of the right-hand side in (3.22) is positive and does make a contribution to the peak correlation response when a watermark is embedded. An example is shown in Fig.3.3. During watermark embedding, the coefficients D, G and I will keep unchanged because their magnitude is lower than the embedding threshold. They will not be chosen for watermark detection either because the threshold value for watermark detection is the same with the one for embedding. The coefficients A, E and H are chosen for watermark embedding but will not be chosen for watermark detection since their magnitudes are lower than the threshold value for watermark detection after watermark embedding. These coefficients are I_{e2} and the ignorance of I_{e2} in watermark detection will increase the correlation response since the sign of these coefficiensts and that of the watermark symbols are different.

One main concern of the correlation-based watermark detection is the efficiency issue. In applications where one is required to determine which watermark ID number is embedded, we may have to check all possible ID numbers. By assuming the total number of users is 2^{32}, it is not practical to try from ID number 0 to $2^{32} - 1$ to determine the exactly embedded watermark ID number. Thanks to the block coding of JPEG-2000, we can simplify the detection structure. We first divide all coding blocks into n subset S_i, $i = 1, \cdots, n$. We can embed k bits in each of the subset S_i. Therefore, the total number of bits that can be embedded in the image is $k \times n$. In watermark detection, we only need to check 2^k different numbers in each subset to decide the k-bit value. $k \times n$ bits can be decoded correctly after n subsets are processed. If we also consider the sign of the correlation response, we can check only 2^{k-1} watermark candidates in each block. For larger images, we are able to have more divided subsets to allow a larger watermark capacity. However, because spread-spectrum watermarking is adopted in the system, a spreading gain must be maintained to reliably embed and retrieve the watermark. Therefore, there exists a tradeoff between robustness and capacity.

3.2.7 Experimental Results

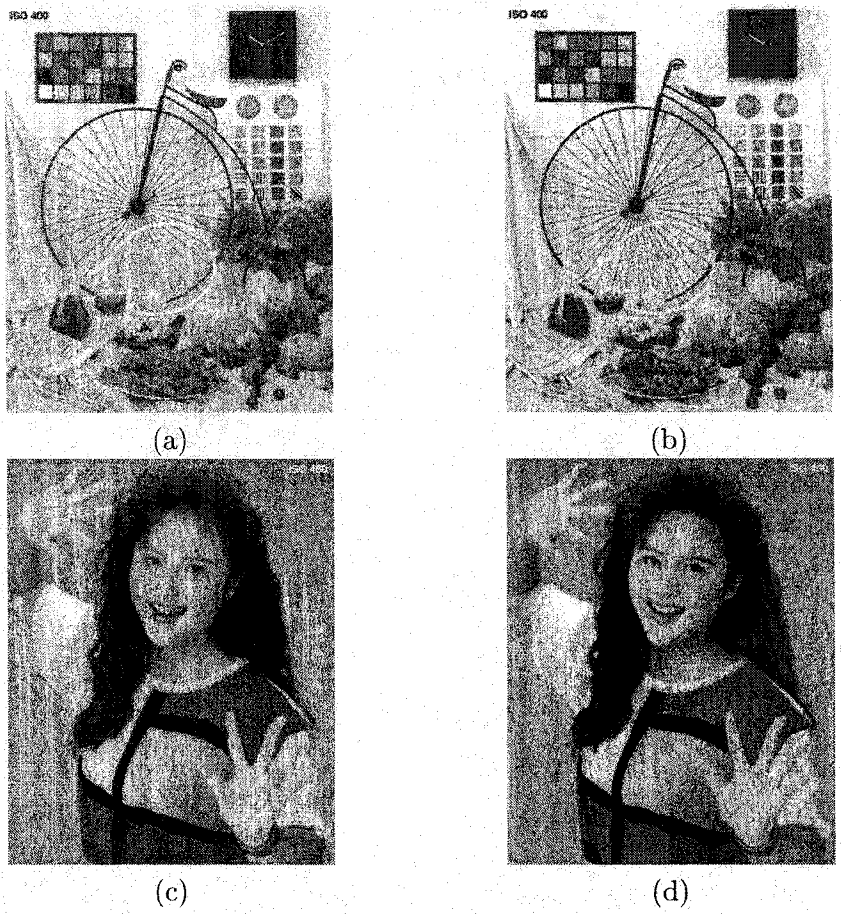
In this section, we show some experimental results to demonstrate the robustness of the proposed watermarking scheme. The embedded watermark has an ID number 500, which is actually the seed to generate the random watermark sequence. The watermark is embedded when the image is compressed. It can be detected when the JPEG-2000 bit-stream is expanded. We test 1000 watermark ID numbers to see if the correct one is detected without ambiguity. A threshold value calculated with (3.19) is used to determine if there exists a certain watermark in the image.

First of all, we examine the parameters to be used in the experiments. In JPEG-2000 implementation, the coefficient precision P can be either 32 or 16. We choose 32 since it is commonly used in software or hardware designs today. The guard bit G is chosen as 2, which has been shown a reasonable number to avoid the overflow problem. Therefore, the nominal range of subband coefficients will be in $(-2^{29}, 2^{29})$. The image sample precision I is equal to 8 for gray-level images. In the experiments, we let q in (3.12) be equal to 24 so that if a coefficient that has the non-zero bit higher than or equal to 24 will be viewed as a significant coefficient for watermark embedding or retrieval. The watermark sequence W_{b_s}, which takes value 1 or -1, is left-shifted by $22 + \alpha_{b_s}$ bits and then added to the selected coefficients to form watermarked coefficients as indicated in (3.14). The value of "extra LSB," which indicates the number of insignificant bit-planes in a coding block, is determined by JPEG-2000. We do not make any change on it. In watermark detection, our algorithm tends to generate a very high correlation response if the watermark exists. This allows us to set a higher threshold value to avoid any possibility of false alarm. The threshold scaling parameter τ in (3.19) is set to 7. If the progressive watermark detection is used, we choose τ to be 8.5. As mentioned in Section 3.2.4, we should define the minimum number η of the selected coefficients to claim the existence of watermark. In the experiment, η is chosen to be 500.

Two JPEG-2000 test images, "Bike" and "Woman," with size 2048 by 2560, as shown in Fig. 3.4 (a) and (c), respectively, are used to demonstrate the invisibility of the embedded watermark. Because of the large size of the images, we encode them with a lower bit rate equal to 0.5 bpp so that they can be stored and transmitted efficiently. The SNR progressive mode is enabled to demonstrate the fully-embedded feature and progressive watermark detection. When the watermark function is disabled, the peak signal to noise ratio (PSNR) between the original and the compressed images of "Bike" and "Woman" are 33.54 dB and 33.70 dB, respectively. The PSNR values between the original and watermarked/compressed images are 32.49 dB and 33.09 dB, respectively. It is clear that the quality degradation resulting from watermark insertion is very little. The watermarked images are shown in Fig. 3.4(b) and (d).

Next, we demonstrate the correlation response in the watermark detection process. Detection results are shown in Fig. 3.5. We can see clearly that there exists a peak with the watermark ID number 500 in both cases. The peak value of the correlation response is much larger than the threshold value T, which is shown as the break line in the figures. The responses of other watermarks are much lower than T. The target watermark can thus be determined unambiguously.

It usually takes a while to decode such large images completely from the entire bitstream. Furthermore, watermark detection may involve a lot of coefficients so that the watermarking process is also time-consuming. We take the "Woman" image for example.



(a)

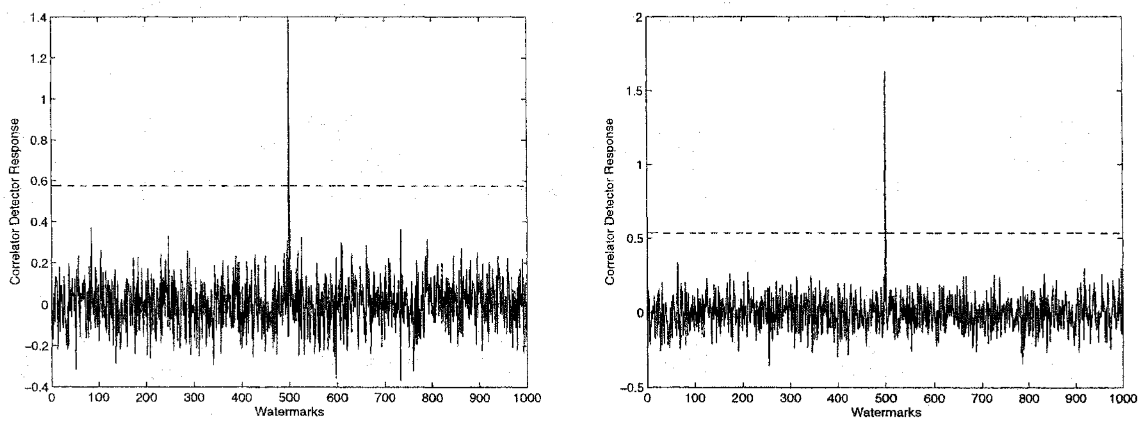
(b)

(c)

(d)

Figure 3.4: Original images v.s. compressed/watermarked images: (a) original "Bike," (b) watermarked "Bike" (PSNR:32.49dB), (c) original "Woman" and (d) watermarked "Woman" (PSNR:33.09dB). Both images are with size 2048 × 2560 and are compressed with 0.5 bpp.

There are over 400K coefficients selected for watermark detection. Actually, the number of selected coefficients necessary for watermark detection can be reduced. We use progressive watermark detection by taking advantage of the fully-embedded feature of JPEG-2000 to speed up the watermarking process. There are two ways to demonstrate this property. The first one is to adopt progressive watermark detection. During the decoding process, whenever the correlation response is larger than the threshold T and the number of selected coefficients is larger than η, we stop watermark detection and claim the existence

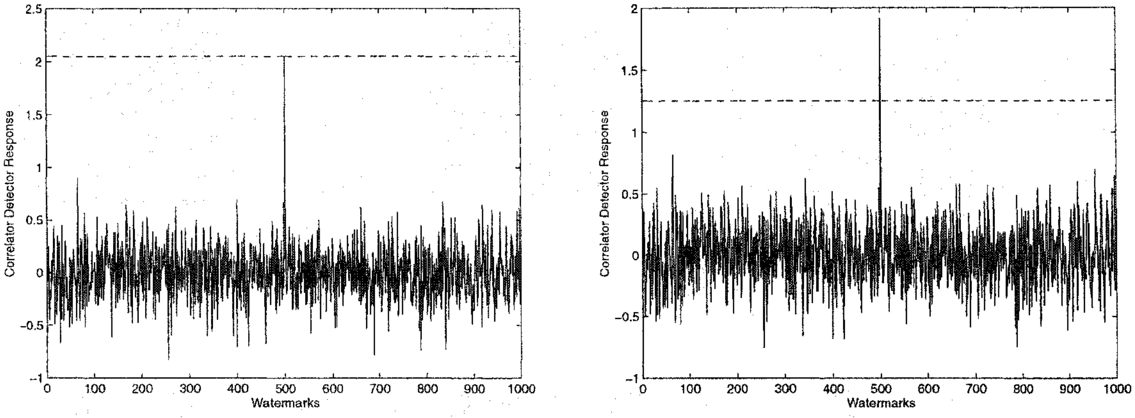


(a) (b)

Figure 3.5: Watermark detection results for (a) "Bike" and (b) "Woman" with 1000 watermark sequences tested.

of watermark. We test "Woman" image in progressive watermark detection. The result is shown in Fig. 3.6(a). The number of selected coefficients is 32,611. The value is still large because we set up a conservative threshold to avoid false alarm. In the second approach, we do not use progressive detection but specify the decoding rate of the image. Fig. 3.6(b) shows the detection result when the image is decoded at a bit rate equal to 0.01. The PSNR of the decoded image is 22.98 dB. In this case, the number of selected coefficients is 6,511. The existence of watermark with ID number 500 is detected in both cases, yet the speed of watermark detection is greatly improved.

The next example is ROI watermarking. ROI coding is especially useful for large images. We use the other JPEG-2000 test image, "Aerial2" (2048 × 2048) shown in Fig. 3.7(a), as an example because application of ROI is important for aerophotography. We assign two rectangular regions which cover two constructions in image "Aerial2" as the desired ROI. We then enable the SNR progressive mode to encode the image, and then decode it with a lower bit rate. We can see from the Fig. 3.7(b) that the two regions



(a)

(b)

Figure 3.6: Progressive watermark detection: (a) watermark detection by using the progressive mode and (b) watermark detection of the image at a bit rate of 0.01 bpp

are well reconstructed while other parts of the image remain blurred. The ROI coding can be verified by the spatial difference of the transmitted image and the roughly decoded image as shown in Fig. 3.7(c) where the lighter the pixel is, the larger the difference between the two images. The detection result shown in Fig. 3.7(d) indicates that the proposed watermarking scheme matches the ROI feature of JPEG-2000, and the embedded watermark can be detected without any difficulty. Besides, with ROI watermarking, it is also possible to embed different watermark ID numbers into different objects in the same image. In this case, the watermark can be viewed as a function of data labeling, which may benefit content-based retrieval in the management of multimedia databases as we mentioned before.

Although wavelet-based coding schemes have the advantages over the block DCT-based coding method in terms of the rate-distortion tradeoff performance, reconstructed images still suffer from various coding artifacts such as ringing effect, graininess, and blotchiness, etc. In JPEG-2000 VM, a postprocessing technique is used to reduce these artifacts so

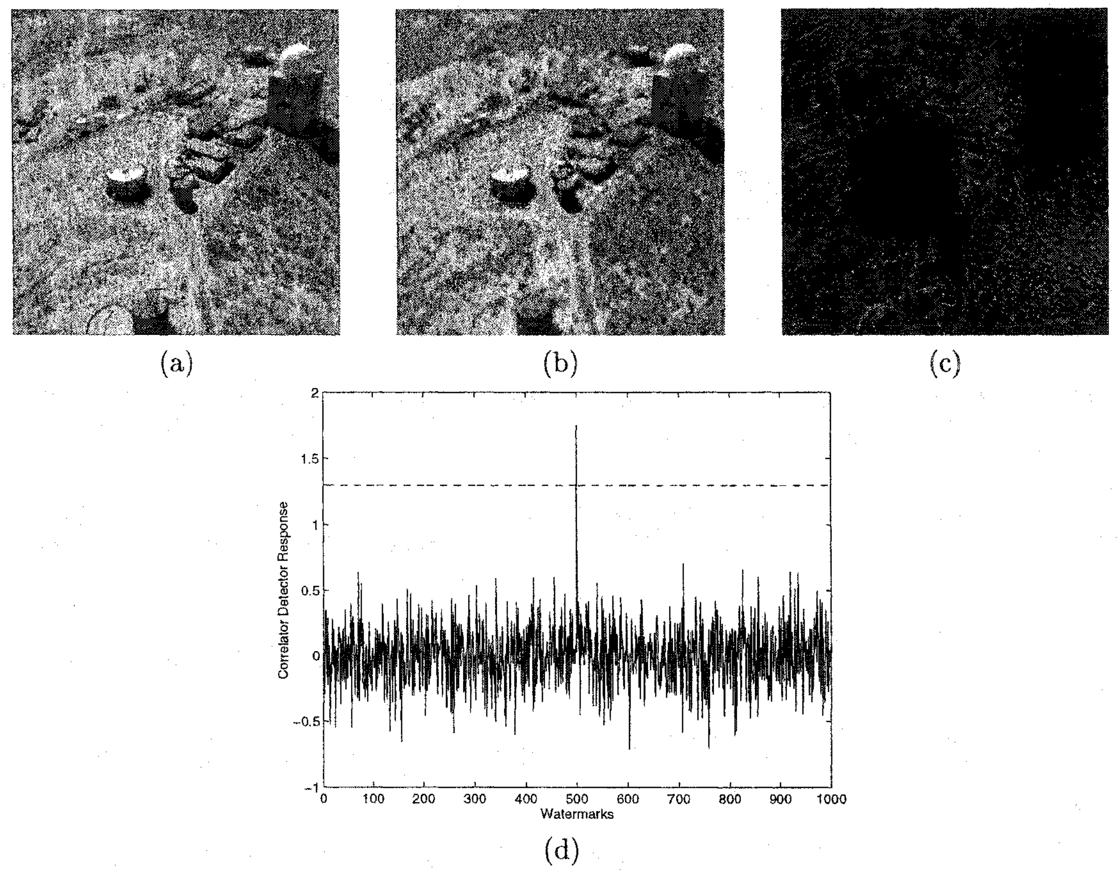
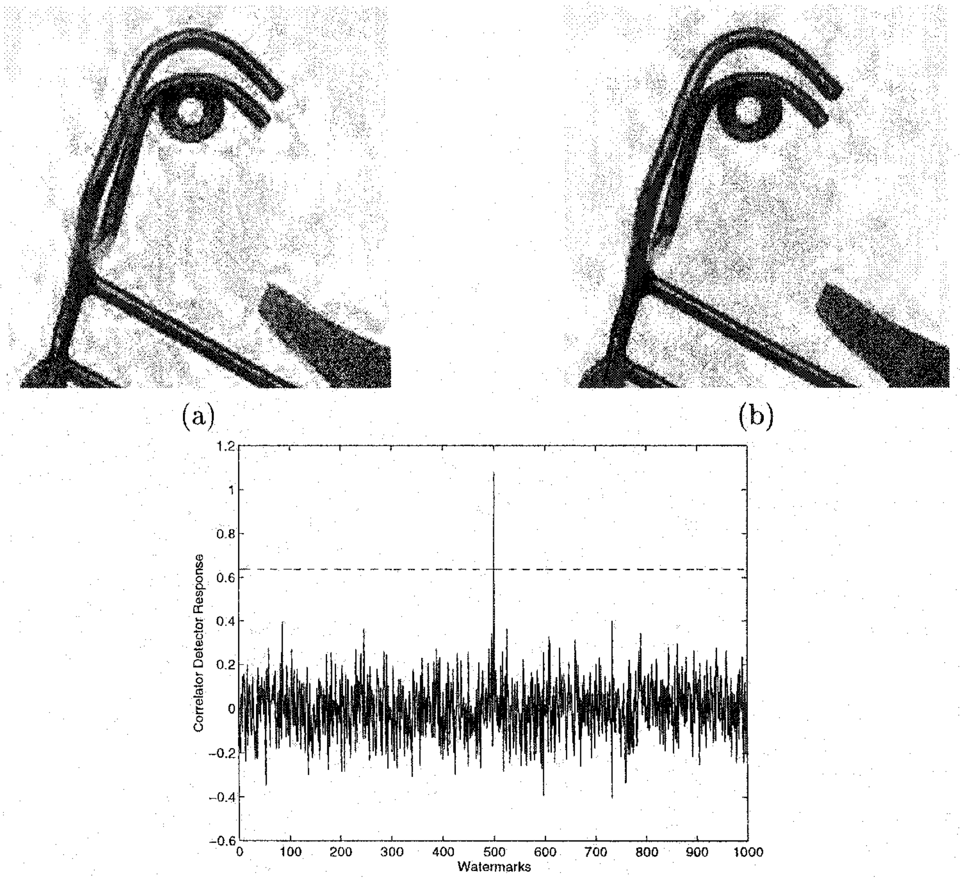


Figure 3.7: ROI watermarking: (a) the fully reconstructed image from ROI coding bitstream, (b) the decoded image with a bit rate of 0.4 bpp, (c) the spatial difference of these two images (the lighter the pixel is, the larger the difference and the two black rectangles are the assigned ROI) and (d) the watermark detection result.

that the overall visual quality of decoded images can be improved substantially. Therefore, we apply the JPEG-2000 postprocessing technique [33, 69] to the decoded image and then test the performance of watermark detection. We decode the "Bike" image with 0.125 bpp and then feed it to the postprocessing stage with three iterations. The effect of the postprocessing can be understood by comparing the two images, before and after postprocessing. Fig. 3.8(a) and (b) show only part of the "Bike" image. The ringing artifact around the bike handler in Fig. 3.8(a) is greatly reduced in Fig. 3.8(b) so that the visual quality is improved. The watermark detection result demonstrated in Fig.

3.8(c) indicates that our watermarking scheme can be coupled very well with JPEG-2000 including the postprocessing procedure.



(a)

(b)

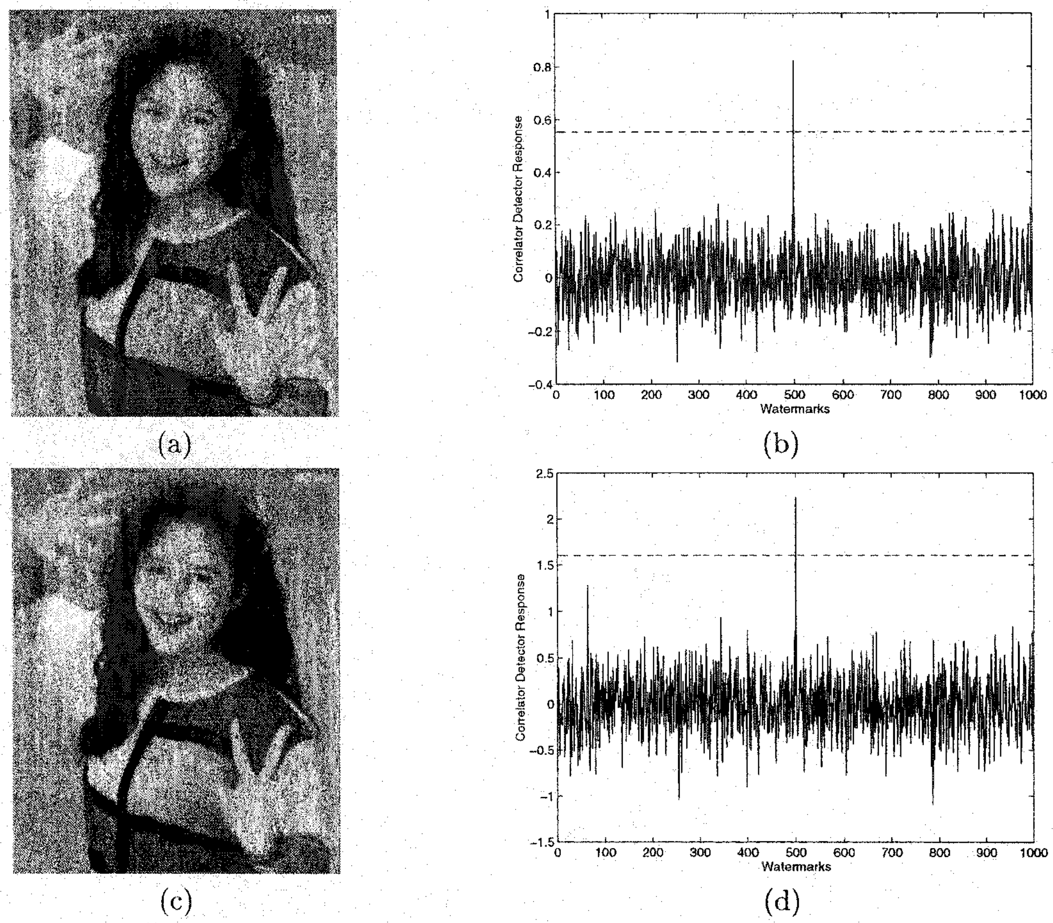
(c)

Figure 3.8: Postprocessing: (a) part of the "Bike" image before postprocessing, (b) part of the "Bike" image after postprocessing, in which the ringing artifacts are greatly reduced and (c) the watermark detection result.

We then apply a series of attacks to show the robustness of the proposed watermarking schemes. First, we consider compression attacks, *i.e.*, to compress the image with other schemes at low bit rates. The well-known DCT-based codec, JPEG, and the wavelet-based codec, SPIHT, are the two compression attacks under test. The attacked image is encoded into the JPEG-2000 bitstream for watermark detection. Since perceptual loss caused by compression varies in different images, some images can be compressed with a higher compression ratio yet preserving good image quality. To verify that the watermark is robust against JPEG and SPIHT attacks, we choose to compress the image with extremely low bit rates. That is, the "Woman" image is compressed by JPEG with quality factor equal to 1 and by SPIHT with a bit rate of 0.005 bpp. The resulting images and detection results are shown in Fig. 3.9. In the JPEG-attacked image as shown in Fig. 3.9(a), a very serious blocking artifact appears since JPEG encodes an image block by block. The SPIHT-attacked image, as shown in Fig. 3.9(c), is blurred very much since only a few coefficients are used to reconstruct the whole image. The PSNR values of the JPEG- and SPIHT-attacked images are 22.61 dB and 22.50 dB, respectively. Although the two images are compressed to an unacceptable degree, the embedded watermark still survives well as shown in Figs.3.9(b) and (d).

GIF is a popular file format for graphics. Unlike JPEG, the maximum number of colors that can be used for a picture is 256. For some images, annoying visual degradation will not be generated when they are converted to the GIF format. Thus, color reduction is another important type of attack that a watermark must resist. We tested two kinds of attacks. The first one is to reduce the number of colors from 256 to 4 for a gray-level image. The second one is image halftoning that is used quite often in FAX, newspapers, or other publications. The attacked images with the detection results are shown in Fig. 3.10. The distinctive peak of the correlation response indicates the survival of the watermark even though the attacked image is visually different from the original one.

With the advances of graphical tools, users may edit an image with some artwork. Thus, it is interesting to test the robustness of the proposed watermarking scheme by



(a) (b)

(c) (d)

Figure 3.9: Compression attacks: (a) the watermarked "Woman" image compressed by JPEG with quality factor equal to 1 (PSNR: 22.61 dB), (b) the watermark detection result of the JPEG-attacked image, (c) the watermarked "Woman" image compressed with SPIHT at a bit rate of 0.005 bpp (PSNR: 22.50 dB) and (d) the watermark detection result of the SPIHT-attacked image.

using a popular editing tool, e.g. the Paint Shop Pro. In order to demonstrate the attack effect, we choose the two classic images, "Lena" and "Baboon" shown in Fig.3.11(a) and (c) as the tested images because of their diverse image characteristics and structures. Both of the images are of the size 512 × 512. We encode "Lena" with bit rate 0.35 bpp and "Baboon" with 2 bpp so that the PSNR values of the resulting compressed/watermarked images (compared with the original image) are around 34 dB. For "Lena" image, if the

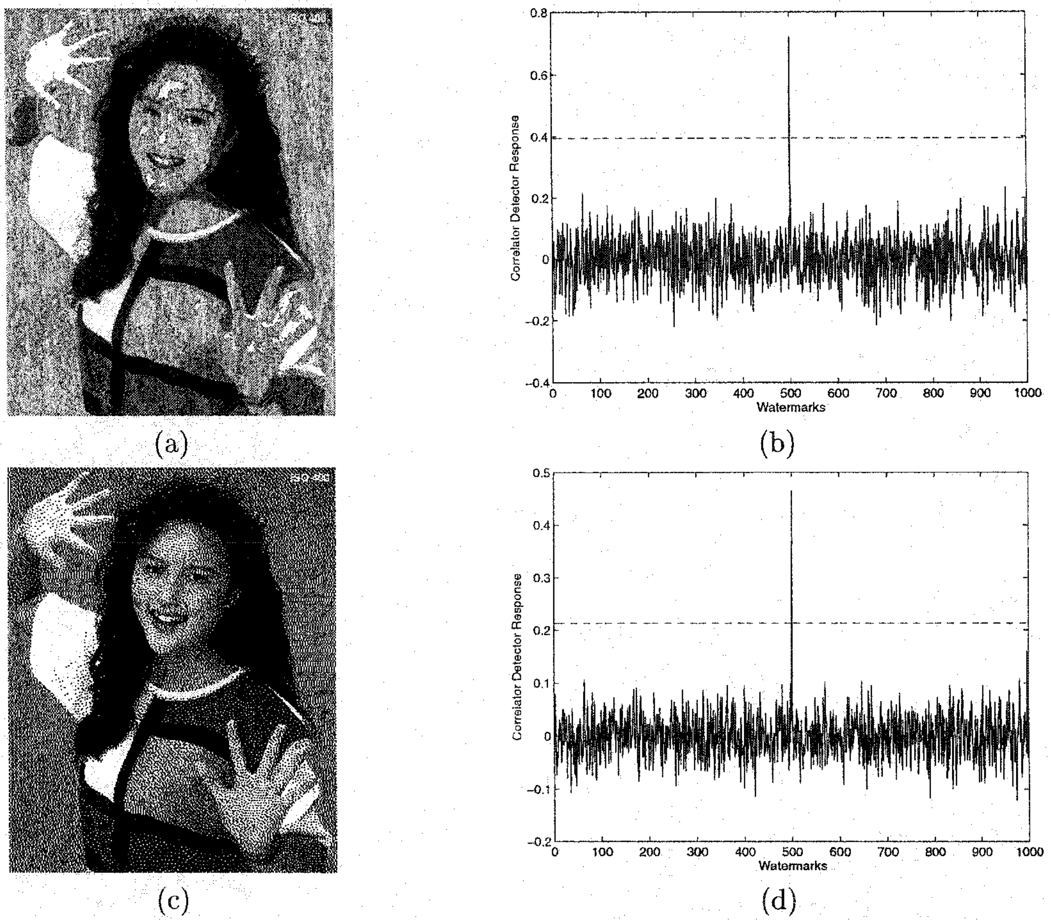


Figure 3.10: Color reduction attacks. (a) The attacked image with 4 colors and (c) the image undergoing halftoning. (b) and (d) are the detection response of (a) and (c) respectively

watermark function is disabled, the PSNR value is 35.64 dB, while that of the watermarked/compressed image is 34.44 dB. For "Baboon" image, the PSNR values of the compressed and the compressed/watermarked images are 34.98 dB and 34.11 dB, respectively. Thus, it is clear that quality degradation resulting from watermark insertion is limited. The two watermarked images are shown in Fig.3.11(b) and (d).

Detection results are shown in Fig. 3.12. The tests include 0) no attack, 1) sharpening, 2) edge enhancement, 3) low-pass filtering, 4) high-pass filtering, 5) dilating, 6) eroding, 7)

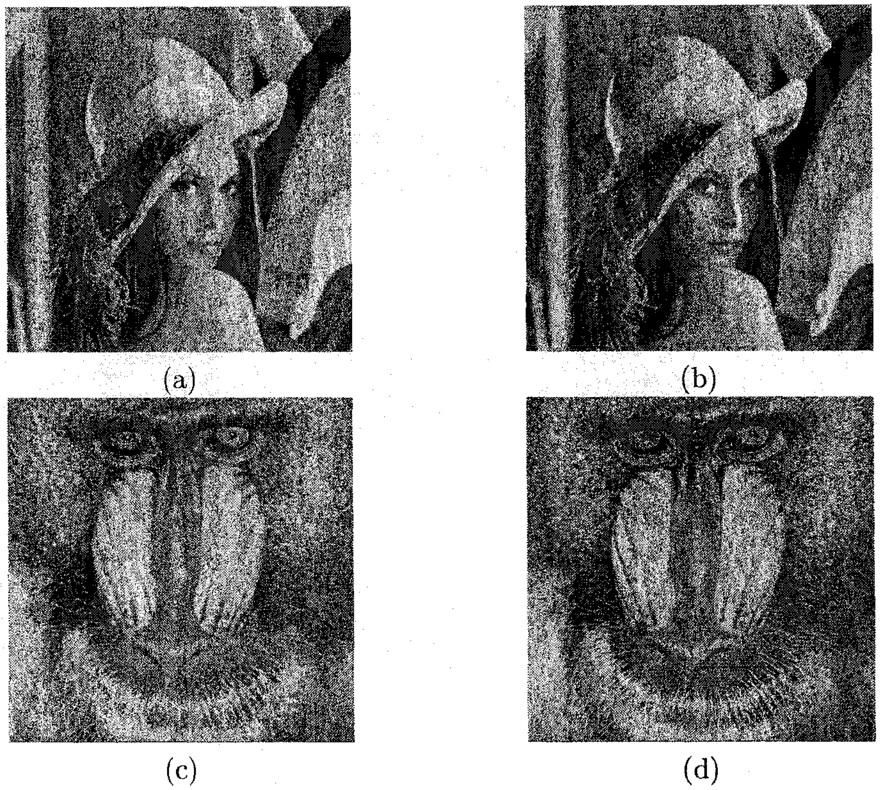
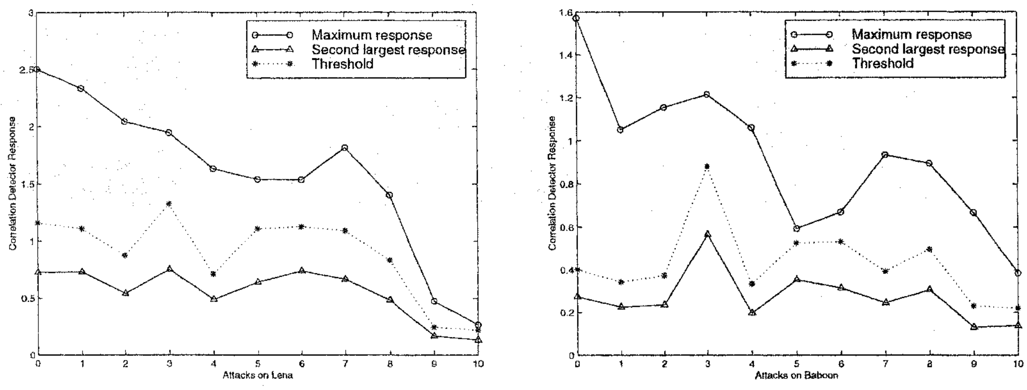


Figure 3.11: Images in extensive watermark testing: (a) the original "Lena," (b) the compressed/watermarked "Lena" (bit-rate: 0.35 bpp, PSNR: 34.44 dB), (c) the original "Baboon" and (d) the compressed/watermarked "Baboon" (bit-rate: 2 bpp, PSNR: 34.11 dB).

histogram equalization, 8) mosaic, 9) 25% uniform noise adding and 10) 25% random noise adding. We show the maximum response (resulting from the embedded watermark), the second largest response (related with one of the other 999 watermarks) and the detection threshold. We can see that different attacks do have varying effects on the embedded watermark while the watermark is still robust under these attacks since the watermark is embedded in the significant coefficients that usually remain stable after most image processing operations.



(a) (b)

Figure 3.12: Detection results of extensive watermark testing on the watermarked images, (a) "Lena" and (b) "Baboon."

Finally, we would like to measure the false positive rate of the proposed system. As mentioned in Section 3.2.6, there are two cases in false positive detection: (1) the watermark is detected in an un-watermarked image and (2) the wrong watermark ID is detected. In the first case, *i.e.*, the watermark is found in an un-watermarked image, the best method to measure the probability of false positive detection is to detect watermarks in numerous clear (un-watermarked) images by using the proposed watermarking scheme. This part is difficult to achieve due to the shortage of content sources. However, we believe that the Gaussian assumption should hold in this case so that the false positive rate should be covered by our analysis. The major concern of the accuracy of the Gaussian assumption comes from the second case, *i.e.*, a wrong watermark is detected from a watermarked image with different ID. At this point, we would like to verify it by experimental data. To do so, we generated 100000 watermark sequences, constructed the watermarked image by embedding one of the watermark sequences and used 100000 watermarks (one correct watermark and the other 99999 incorrect watermarks) for detection. We counted the number

Error rate	Number of wrong watermark detected	Expected number of false detection
5×10^{-3}	525	500
10^{-3}	99	100
5×10^{-4}	55	50
10^{-4}	12	10
8×10^{-5}	10	8
5×10^{-5}	7	5
3×10^{-5}	3	3
10^{-5}	1	1

Table 3.1: The number of false positive detections measured from 100000 tested watermark sequences vs. estimated number of false positive detections based on Gaussian assumption.

of tested watermarks with a correlation response higher than the threshold value set according to the allowable false positive rate. We tested the false positive rate from 5×10^{-3} to 10^{-5}. We measured more points around 10^{-4} to 10^{-5} because these measurements may be more correct and important. The results are shown in Table 3.1. We see that the number of false detection matches pretty well with the predicted false detection based on the Gaussian assumption. Table 3.1 verifies the suitability of our analysis. By following this trend, we expect that the false positive analysis will work when a higher threshold value (with lower false positive rate) is set.

3.3 Steganography in JPEG-2000 Compressed Images

In this section, we consider a different scenario of information hiding in JPEG-2000 from the watermarking scheme presented in Section 3.2. We focus on developing a steganographic scheme under the framework of JPEG-2000 so that a high volume of information can be secretly transmitted to the intended recipient in a more reliable fashion. The steganographic application may be of certain military usage. It can also be useful when the information is sensitive in some way that the sender and the receiver would like to exchange it through a public channel without being noticed by the third party. In our case, a digital image serves as a host or is viewed as a camouflage to cover the existence of the hidden information. Instead of focusing on the robustness issue as done in most of the digital image watermarking research, we take capacity, reliability and security into more serious consideration. The goal is to ensure that the suitable amount of hidden information be transmitted without errors.

Two approaches to achieve steganography in digital images are commonly used. The first approach is to embed the information into the imagery data without taking any file format into consideration, as the way many digital watermarking schemes are operated. The second approach is to explicitly work on a specific image format. We believe that the second approach may be more appropriate in the application of covert communication. First of all, digital images are usually compressed to facilitate their storage or transmission. In natural images, the lossy compression is more commonly employed to form a compact representation of the image. By embedding the secret information into the imagery data, we run the risk of losing it by the subsequent image compression, which contradicts the requirement of reliability. Besides, embedding the hidden information in the compressed images may also help avoid the problem of attacks since transcoding or signal processing procedures rarely happen in the process of transmission if the images have been stored in a certain format. However, embedding information in the compressed bit-stream is much more difficult than in the imagery domain since the space for hosting the hidden information has been significantly squeezed by modern compression methods.

Many of previous information-hiding schemes operating on the compressed domain were based on JPEG. The choice is partly because that most of the still images circulated nowadays are compressed with JPEG. The other reason is that, as a block DCT codec, JPEG lends itself to a good candidate for information embedding due to its fixed block structure. As JPEG-2000 is viewed as a promising image standard in the near future, we develop novel and feasible approaches to effectively achieve steganography in JPEG-2000 compressed images.

3.3.1 Challenges of Steganography in JPEG-2000

Before designing a steganographic scheme under the framework of JPEG-2000, we have to first determine an appropriate position in JPEG-2000 coding flow for effective information embedding. From the structure given in Fig. 3.1, there are three positions to be considered. We examine their suitability for information embedding below.

(1) Image Transform

After the intra-component image transform, the image data are transformed to wavelet coefficients. If we modify the data at this stage, the scheme will be equivalent to many existing wavelet-based watermarking algorithms, which may take other wavelet-based codecs as attacks. For digital watermarking, the payload is usually low, and multiple embedding with the majority detection or the spread-spectrum concept can be applied. The embedded information can thus have sufficient robustness against lossy compression of another codec. However, multiple embedding is not suitable in the steganographic application as the required payload is usually high and we have to make efficient use of the already limited bandwidth.

(2) Quantization

Quantization is an important step in image compression, which reduces certain visual redundancy for efficient coding. As mentioned in Section 3.1, quantization is the primary source of information loss. We have to avoid losing the hidden data due to coarser quantization by embedding them in the quantization indices. The solution works for JPEG (as many JPEG-based data hiding schemes operate on the quantization indices), but is not good for steganography in JPEG-2000. It should be noted that wavelet-based coders usually truncate the compressed bit-stream to fulfill the targeted bit-rate. In JPEG-2000, PCRD optimization strategy is adopted so that the truncation mechanism is activated after the whole image has been compressed. If embedding the information at this stage, we cannot predict exactly which quantization index or bit-plane of an index will be included in the final code stream. The embedded information will not be perfectly recovered unless the lossless compression mode is chosen.

(3) Coding

If the information is embedded in the output of tier-2 coding, *i.e.*, the JPEG-2000 packets, it can be guaranteed that all the embedded information will be received without error and in a correct order because we avoid the two major sources of information loss, *i.e.*, quantization and bit-stream truncation. However, we will have difficulty in modifying the packets for information embedding since the bit-streams may have been compactly compressed by the arithmetic coder. Careless modification could result in failure of expanding the compressed image.

3.3.2 Progressive Embedding of a Hidden Image and Its Drawbacks

There does exist a solution to partially achieve high-volume information hiding in wavelet-based codecs. From the previous discussion, we know that the hidden information could be lost after the subsequent truncation of the compressed bit-stream if it is embedded in the quantization index. However, an intuitive concept indicates that certain indices may have a better chance of survival since they are of more significance. To be more specific, wavelet coefficients in lower frequency subbands are usually more important than those in higher frequency bands. An extreme example is resolution progressive transmission, in which lower frequency subbands will be sent prior to higher ones. In this case, lower frequency bands should be preserved well at high bit-rates. On the other hand, although some portions of the embedded information may be lost, the recipient can still receive enough information if the significant portions are transmitted successfully. Therefore, if the hidden information is a digital image of a smaller size, we may transmit it by embedding in the quantization indices. This general idea should work in most of the wavelet codecs.

We briefly describe the idea and factors that should be considered. First of all, we decompose the hidden image with the wavelet transform. The number of wavelet decomposing levels and the image size should be related to the host image. For example, if the host image is 512×512 and decomposed with 3 levels, we may set the rule that the hidden image is one fourth the size, *i.e.*, 256×256, with the same decomposing levels. We may need to pad the sides of the hidden image when its size is smaller than required. This strategy is to make sure that no image specific information be necessarily known by the recipient. Besides, it should be noted that coefficients in each level be represented by a fixed number of bits, which is also known by both sides. Then we embed the wavelet subbands of the hidden image according to their importance into the counterparts in the host image. Basically, lower (higher) frequency subbands of the hidden image will be embedded in the lower (higher) frequency subbands of the host image. Embedding is based on modification of lower bit-planes of the host quantization index, *i.e.*, replacing its least significant bits with the bits of the hidden information. The number of the bit-planes used for information embedding will affect both capacity and quality of the resulting image. Embedding follows certain predefined agreements between the information embedder and the recipient. A rule of thumb is that the sign of a coefficient, which is comparatively important than the magnitude, should be embedded more carefully. In addition, bit-planes of lower frequency subbands of the hidden image should also be better embedded than those of the higher frequency subbands. Permutation can be applied to the hidden information before embedding to increase uncertainty. The recipient should be able to permute it back correctly to reconstruct the hidden image. We should not embed any data in the lowest frequency band, *i.e.*, the DC band, of the host image to avoid generating unpleasant artifacts.

Many existing wavelet-based watermarking schemes may emphasize the function of progressive embedding/detection. However, there are some drawbacks in the idea of progressive hidden image transmission. First of all, we know from the previous discussion that many parameters have to be known in advance by both the sender and the receiver so that the hidden image can be correctly extracted and perceived. Nevertheless, if a lot of information has to be shared beforehand between the sender and the receiver through a certain side channel, the steganographic scheme becomes impractical. Besides, if the hidden information is an image, the eavesdropper may have more chances to detect its existence, given that the characteristics of an image are quite different from that of the host signal. Encryption might not be applicable here since we cannot guarantee that the hidden information be transmitted without errors due to the truncation of JPEG-2000 or other wavelet-codec and an encrypted bit-stream is usually not robust to any error. The most we can do to increase the security level is to permute the position of the wavelet coefficients of the hidden image as mentioned above. It is not clear how this permutation procedure will prevent the eavesdropper from detecting the hidden image. Finally, the requirement of embedding an image as the hidden information significantly limits the usage of this algorithm. The embedded information should be general binary data, in any form such as texts, encrypted bit-streams, raw/compressed images, or else, to achieve practical covert communication. It is apparent that the method presented above may not meet our requirements.

In the following sections, we would like to develop a practical steganographic scheme to convey high-volume general binary data in a more secret and reliable manner.

3.3.3 Information Embedding with Lazy Mode Coding

As analyzed in Section 3.3.1, in order to transmit general binary data secretly without error, we should embed the hidden information in the JPEG-2000 packets. However, we have to avoid modifying the bit-stream that is entropy coded for its correct decoding. In JPEG-2000, a so-called "lazy mode" coding option is introduced, in which the arithmetic coding procedure is completely bypassed for most of the significance and the magnitude refinement coding passes. Thanks to this lazy mode coding option, we propose a steganographic scheme, which solves all the above-mentioned problems to achieve reliable covert communication in JPEG-2000.

This lazy mode choice for the proposed scheme can be further justified below. It has been observed that, at high bit-rates, the symbols produced by the significance and refinement passes have distributions close to a uniform one so that there is no substantial benefit from arithmetic coding. Bypassing the MQ coder can thus reduce the complexity and improve the execution speed without degrading the coding performance. Since images used to hide a large volume of data are usually compressed at high bit-rates, it is appropriate to embed information in JPEG-2000 with the lazy mode enabled. Besides, our embedded information could also be uniformly distributed given that encryption is usually applied. Therefore, these passes turn out to be good candidates for modification with less chance of being noticed.

The proposed scheme is to embed the data in the selected magnitude refinement passes. With the lazy mode, except for the four most significant bit-planes, the remaining such passes are raw coded so we can simply modify the raw data without causing problems. However, we should ensure that certain bit patterns never occur in the output, since such patterns may be used for the error resilient purpose.

The choice of the magnitude refinement pass for information embedding is explained as follows. Among the three types of passes, the significance pass carries significance and necessary sign information. Any modification of the significant passes could cause either sign flipping or decoding errors. For the cleanup pass, it is sometimes run-length coded so its modification is still prohibited. The magnitude refinement pass carries subsequent bits after the most significant bit for each sample. The significant bits can act as visual masking to make the modification of these subsequent bits less obvious. By considering these factors, we conclude that only the magnitude refinement pass is suitable for information embedding.

3.3.4 Selection of Refinement Passes for Embedding

In order to avoid degrading the composite image severely, we may only use a subset of the raw coded magnitude refinement passes for information embedding. We describe three scenarios of selecting suitable magnitude refinement passes as follows.

(1) Fixed number of the lowest bit-planes

The most straightforward method is to examine the bit-planes where these raw coded magnitude refinement passes are located. Given that the total number of meaningful bit-planes in a subband is K, which is signaled explicitly in the code stream, the modification for information embedding is restrained to those bit-planes lower than $K - G$. The smaller G is assigned, the more bit-planes can be modified so that more hidden information can be carried. Basically, this idea is similar to the common LSB-based information-hiding methodologies, in which only the lowest few bit-planes are modified to avoid introducing visible artifacts. The subtle difference is that, in this steganographic scheme, not all the data in those bit-planes but the data included in the magnitude refinement passes are affected by the embedding process. The advantage of this embedding scenario is its simple implementation since both the information embedding and extraction can be done efficiently in the tier-2 coding. Besides, both of the information embedder and extractor only need to know the parameter G to achieve successful secret communication. The amount of the information that has to be transmitted through other subsidiary channels is thus significantly reduced.

(2) Bit-planes below the MSB

A more sophisticated way is to take the MSB of the quantization index into account. In this embedding scenario, the digit of a quantization index is allowed to be modified for information embedding if it is located at the bit-plane that is below the MSB by at least P bit-planes. This idea is somewhat analogous to those watermarking schemes that use the magnitude of a host coefficient to scale the embedded watermark [14, 13]. If a host signal is large in magnitude, we may modify it with a greater scale so that more information or a stronger watermark can be embedded owing to the masking effect. Therefore, the capacity of the steganographic scheme may thus be improved without further affecting the visual quality. In addition, we can intentionally ignore some bit-planes of certain coefficients for embedding due to a more advanced visual masking model or increased security concerns. However, the complexity of the implementation increases accordingly since the information embedding/extraction processes become coefficient-wise, instead of simply viewing the whole bit-planes as a group. In this way, the tier-1 coding has to be involved in the information embedding/extraction processes. Besides, we have to deal with problems such as the varying length of the coding passes and the special pattern for error resilience in a more careful manner.

(3) Backward embedding

A better way of selecting suitable magnitude refinement passes for information embedding is to consider the importance of these passes to the overall quality of the compressed image. As explained in Section 3.1.4, the tier-2 coding achieves rate scalability through multiple quality layers. Each coding pass is either assigned to one of the layers or discarded according to its rate-distortion slope, which is calculated in the tier-1 coding and passed to the tier-2 coding for organizing the code stream. The coding passes with larger rate-distortion slopes are included earlier in the lower layers, while the coding passes with smaller rate-distortion slopes are included later in the higher layers.

Our goal is to hide as much information as possible with the minimal impact on the image quality. Obviously, the embedding process should function in the opposite order of the tier-2 coding by selecting less important coding passes earlier for modifying. Therefore, we propose the idea of backward embedding to take account of the importance of the passes to the overall image quality. After the tier-2 coding determines the passes that will be included into the code stream, an extra procedure embeds the information backward, starting from the last included refinement pass. On one hand, modifying these insignificant passes may be similar to discarding them, which does not severely affect the image, compared to those passes included earlier in the lower layers. On the other hand, the length of these passes could be larger so that we can actually hide more information. The embedding procedure can be carried out until the image quality has been degraded within an acceptable level. A termination pattern may be necessary to signal the end of embedding so that the decoder can learn when to stop extracting the hidden information.

3.3.5 Issues on Backward Embedding

By considering the complexity and the performance, backward embedding is adopted in the proposed steganographic scheme. With multiple layers being employed in JPEG-2000, backward embedding can easily select those passes that are less important to the compressed image for information embedding without considering the location of the bit-plane and the associated coefficient. The embedding process can thus be done very efficiently since only the tier-2 coding is involved in the embedding process and the coding structure of JPEG-2000 can be kept almost the same except that an extra procedure to embed the data in a backward fashion is necessary. It is noteworthy that a major benefit of the proposed JPEG-2000 steganographic scheme over the existing JPEG schemes is its controllable rate-distortion trade-off. Here, the rate means the capacity of the hidden information while the distortion is referred to as the additional degradation resulting from the information embedding process. In existing JPEG embedding schemes, the effect on image quality due to information hiding is usually unpredictable since it is difficult to achieve good rate-control in the JPEG standard. In contrast, our scheme may exploit the characteristics of wavelet-codecs to achieve a better balance between the payload of the hidden information and the resulting image quality.

As the embedding process starts from the last included magnitude refinement pass, the ending point of embedding will decide the distortion of the composite image. A simple scenario for controlling capacity and distortion goes as follows. If the image will be compressed with the bit-rate equal to B bpp, we can guarantee that the composite image will have the quality of the image compressed with C bpp, where $C < B$, by embedding the raw coded magnitude refinement passes until the one that is included in both the bit-stream with B bpp and the bit-stream with C bpp. The idea is easy to implement but the estimation of the quality is very conservative since we do not modify all the three coding passes in a bit-plane but magnitude refinement passes, which may only occupy a small portion. We can see this argument from an extreme case that C is equal equal to 0 while the resulting composite image will still have a reasonably good quality. Therefore, to demonstrate the advantages of the proposed JPEG-2000 steganographic scheme over other existing schemes based on the JPEG standard, a more accurate quality measurement is necessary.

If MSE is used as the quality measure, the most accurate way is to calculate the additional distortion in the spatial (or the image) domain. However, the complexity is too high in this approach since we have to expand the compressed bit-stream several times after each embedding of a pass. A more practical way is to evaluate the additional distortion in the wavelet domain along with the generation of the bit-stream. In JPEG-2000, the overall distortion in terms of MSE of the compressed image and the original image is estimated by (3.3) and the rate-distortion slope can then be derived as described in Section 3.1.4. The distortion estimation is based on the two assumptions, *i.e.* the orthogonal wavelet basis functions and uncorrelated quantization errors. Although neither of the assumptions are held perfectly, the estimation is acceptable in the case of compression. Following the same route, we can calculate the additional distortion introduced by information embedding.

During information embedding, distortion happens when the bit flips from 0 to 1 or from 1 to 0, *i.e.*, when the original bit and the embedding bit are different. The additional distortion can then be calculated by the sum of the difference between the original quantization index and the index after possible bit flipping and scaled with the quantization step size and the L-2 norm of the wavelet base function. In the implementation, we need not keep the original quantized coefficient and the resulting coefficient to calculate their difference but evaluate it on the fly with each bit-plane processed. For the same index, if a bit in a certain bit-plane changes from 1 to 0, we record it as a negative change while if a bit changes from 0 to 1, we view it as a positive change. The difference between the index values before and after information embedding can be calculated by taking the sum of the positive changes subtracted by the sum of the negative changes. We give a quick example. If the original value is 44 (101100) and the value after embedding is 50 (110010). Each of the positive change and negative change happens twice. The sum of positive changes will be $1 \times 2^4 + 1 \times 2^1 = 18$ and the sum of negative changes will be $1 \times 2^3 + 1 \times 2^2 = 12$ so the difference will be 6. This way of calculation is straightforward, and the benefit is its adaptation to the bit-plane coding structure.

However, the exact determination of distortion comes with a few drawbacks, which increase the complexity of the implementation. First of all, the information embedding is carried out in the tier-2 coding. At this stage, what the tier-2 encoder sees is only a bit-stream. It knows nothing more than the position of the bit-plane or the code-block to which the pass belongs. When flipping a bit in a pass, we may not know exactly which coefficient will be affected. This problem may be solved by passing more information from tier-1 coding to tier-2 coding. Since only the raw coded magnitude refinement passes will be embedded with the hidden information, the tier-1 encoder may have to send extra information to the tier-2 encoder, indicating the correspondence between the bit in the pass and its associated coefficient. For a 64×64 block, the extra information may be 64×64 bits long for each pass with 1 representing that the pass includes the bit information and 0 representing the null information. The encoder is then able to scan with the same order to identify which coefficient will be affected by a certain bit flipping to evaluate the distortion. Nevertheless, the other problem exists. We have to keep the distortion value for each coefficient in each code block, which increases the memory consumption significantly and contradicts the requirement of efficient memory usage in the tier-2 coding. Next, we provide two rough evaluation methods, which simply operate in the bit-plane level and we will then validate their practibility by comparing them with the exact distortion approach.

The first method is to calculate the distortion by summing up MSE of bit flipping in each bit-plane. If a bit flipping happens, we add it to the overall distortion without taking account of the coefficient. The additional distortion $\triangle D$ with one bit flipping is expressed as

$$\triangle D = \omega_b^2 \times (\triangle_b)^2 \times (2^p)^2, \tag{3.28}$$

where ω_b is the L-2 norm of the wavelet basis function, $\triangle_b$ is the quantization step size associated with band b and p is the bit-plane position.

It is apparent that the distortion calculated in this way is only a rough estimation since the exact change of the distortion with each coefficient is not recorded. We look at two simple but extreme cases. In the first case, when the original value is 100 and the value after embedding is 011, the actual squared error is 1 while we overestimate it with 21. In the second case, when the original value is 111 and the value after embedding is 000, the squared error is 49 while we underestimate it with 21. However, in most common cases, we found that the overestimate and underestimate compensate each other and this method provides us a pretty good estimation of the additional distortion introduced by information embedding.

The second estimation method is to utilize the distortion of each pass calculated during the tier-1 encoding. The distortion improvement of each pass is estimated by the difference between the distortion measured by (3.3) before and after including the pass. The distortion is recorded and then evaluated with the rate increase of this pass for rate control as mentioned in Section 3.1.5. We can view this step as a measurement of importance of the pass since the inclusion of the pass with large distortion improvement makes great impact on the compressed image. We believe that, in information embedding, if the host signal and the hidden signal are both of the uniform distribution, the distortion introduced by embedding or modifying the content of the pass would be very similar to discarding the whole pass. Therefore, we may also use this distortion measurement of a pass as an estimation of the distortion resulting from information embedding. A clear benefit of this method is that the overhead of embedding is made as small as possible since the embedding process shares the same procedure of distortion measurement in the coding process. It should be noted that the mid-point reconstruction rule is often employed in the distortion measurement in the tier-1 coding as shown in (3.2). Information embedding, however, results in bit flipping without mid-point reconstruction as done in the dequantization step. Therefore, we multiply the distortion estimated in the tier-1 coding by 2 as a measurement of the distortion introduced in information embedding of this pass.

3.3.6 Steganalysis of the Proposed Information-Hiding Scheme

As mentioned before, robustness is not the main issue of steganographic applications since we do not expect the attacker will modify the image content by either transcoding or other signal processing procedures, especially in the case that the secret information is hidden in a compressed file for storage or circulation. Capacity, reliability and security are the three major concerns. In our information-hiding scheme with the JPEG-2000 standard, we can guarantee that the secret transmission be carried out without errors by embedding the information in raw coded magnitude refinement passes. We can achieve high-volume covert communication by choosing an appropriate amount of passes for information embedding. The remaining issue to be discussed is the security of the proposed scheme. In this section, we try to play the role of an eavesdropper to clarify some security issues to which we should pay attention when designing a steganographic scheme.

The first step the eavesdropper may take is to analyze the bit-stream structure. One drawback of the proposed scheme is that the information-hiding procedure is operated in a special mode of JPEG-2000, *i.e.* the lazy mode. Some people may question that the attacker may suspect the existence of certain hidden information in the JPEG-2000 bit-stream if it is compressed with the lazy mode. Eventually, this problem depends on how popular the lazy mode will be. From our viewpoint, the lazy mode coding operation is very useful in high bit-rate image compression. The complexity can be significantly reduced by employing the lazy mode coding because the computationally expensive MQ coding is bypassed while coding efficiency will not be affected much, especially in the high bit-rate coding, which is a very possible case for information embedding. The ROI coding may be the only scenario that the lazy mode is not appropriate to be applied. Therefore, we do not see many reasons for not adopting the lazy mode coding in a broad range of imagery applications.

Next, the eavesdropper may analyze the data in the magnitude refinement passes to see if any unusual distribution appears. As mentioned before, the reason why the MQ coder does not improve coding efficiency in the magnitude refinement passes in the lower few bit-planes is that the distribution of these data is close to a uniform one. Therefore, if we can encrypt or scramble the data in some way so that the hidden information also has a uniform distribution, the chance that the eavesdropper can tell the difference will be small. However, we have to make sure that some special patterns designed for increased error resilience in JPGE-2000 should not appear in the modified bit-stream. Aside from the purpose of correct expanding the compressed bit-stream, this cautious strategy can prevent that the appearance of the mark at the wrong position reveals the existence of the hidden information.

The eavesdropper may further expands the compressed bit-stream to see if any abnormal situation happens. In the proposed steganographic scheme, we do not change the length of the magnitude refinement passes but modify the binary content. In general cases, the modified magnitude refinement passes generate the same number of symbols with the original passes. However, some special situations may happen when more symbols or less symbols are generated than expected. This comes from the fact that extra bits are added by the encoder to avoid generating error resilience patterns as described before. The inaccurate number of symbols will not affect the normal operation of the coding but may give a loophole for the eavesdropper to sense the existence of the hidden information. Besides, the bit-stuffing at the end of the pass to comply with the byte boundary may appear differently if embedding is done in a careless way. We may not embed information into the bits that are used for bit-stuffing if a unified bit-stuffing byte is adopted in most of the JPEG-2000 coders. In other words, we should only modify the bits that come from the magnitude refinement passes and avoid the bits used in the simplified implementation or any type of markers to ensure better security.

A more advanced eavesdropper may examine the behavior of wavelet coefficients to see if possible hidden information exists. This is actually an interesting topic to investigate if information embedding has different effects on wavelet coefficients from the quantization process. We believe that modifying the JPEG-2000 packets may result in some intriguing phenomenon on the inverse wavelet transform. We may study this subject by avoid the quantization step, *i.e.*, by operating the information embedding in the lossless compression mode. We leave this part as future research.

3.3.7 Experimental Results

The implementation of our steganographic scheme was based on JASPER [1]. JASPER is a free reference code of JPEG-2000 offering the baseline coding with an excellent performance and thus serves as a good framework. In the experiment, we used the four well-known gray-level images, "Lena," "Boat," "Peppers" and "Baboon" as the host images, all with the same size of 512 × 512, to carry the generalized binary information. We assume that the image will be compressed into the bit-stream with a high bit-rate, *i.e.* 2 bpp. It should be noted that the length of the bit-stream is not changed by the information embedding process. In other words, the embedding process will only affect the quality of the image by modifying the bit-stream content, *i.e.* certain magnitude refinement passes, as described in Section 3.3.3.

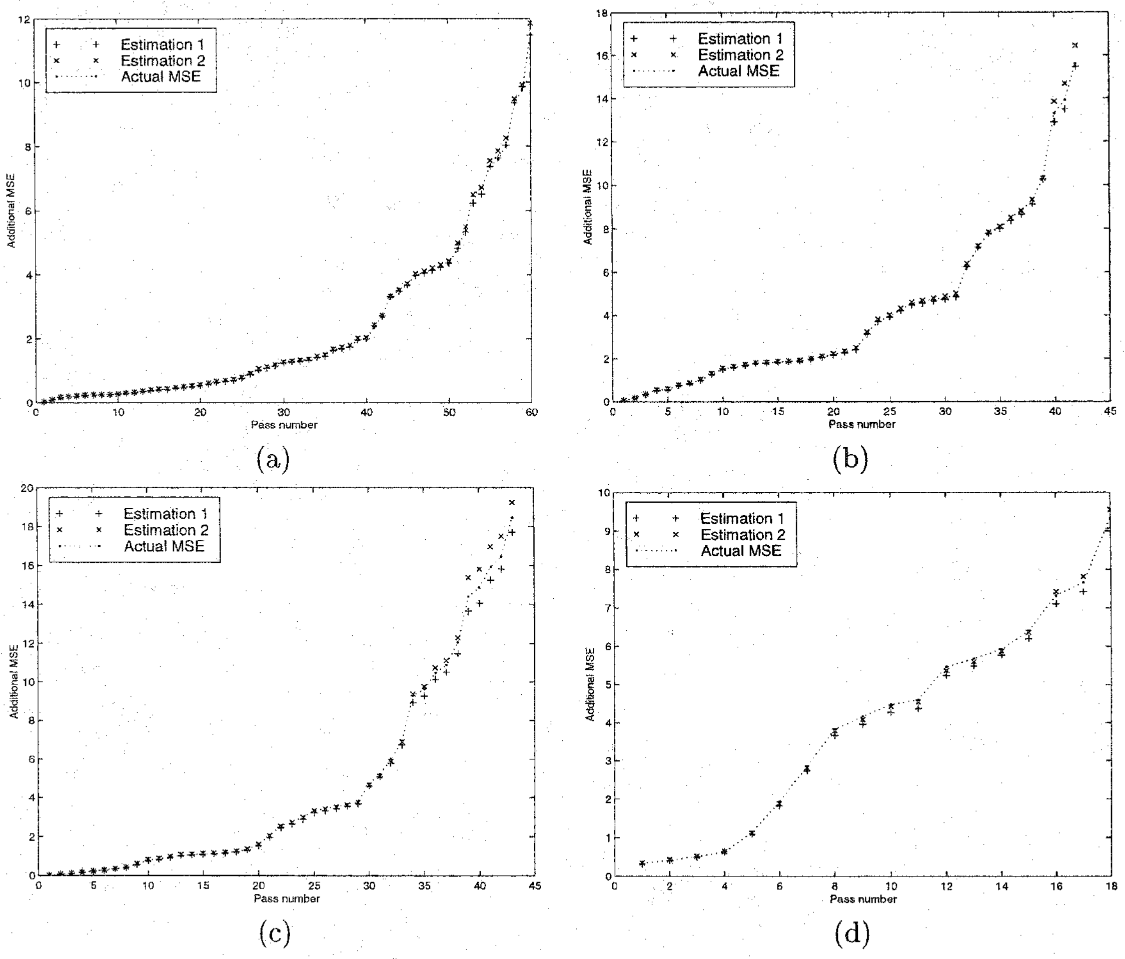
First of all, we would like justify the claim that the lazy mode coding does not give an inferior performance compared with the normal mode in the case of high bit-rate image coding. We compressed "Lena," "Baboon," "Boat" and "Peppers" images from 0.5 bpp to 2 bpp with the normal mode and the lazy mode and then compared the PSNR values of the expanded images in each case. The results are shown in Table 3.2. We see that the difference of PSNR values is very limited. It should be noted that the larger difference, such as Peppers in 2 bpp, comes from the fact that the two compressed bit-streams are

Table 3.2: Performance comparison using the lazy and the normal modes (PSNR in dB)

Bit-rate(bpp)	0.50	0.75	1.00	1.25	1.50	1.75	2.00
Lena(normal)	37.06	38.92	40.31	41.55	42.75	43.94	45.12
Lena(lazy)	37.01	38.87	40.26	41.50	42.69	43.87	45.04
Boat(normal)	33.15	35.11	36.61	37.98	39.30	40.61	41.91
Boat(lazy)	33.14	35.06	36.55	37.91	39.21	40.52	41.81
Peppers(normal)	35.60	37.00	38.23	39.44	40.66	41.89	43.11
Peppers(lazy)	35.55	36.94	38.17	39.36	40.55	41.76	42.96
Baboon(normal)	25.47	27.41	29.06	30.58	32.02	33.41	34.73
Baboon(lazy)	25.47	27.41	29.06	30.59	32.03	33.41	34.73

different in their lengths although we have tried to make them as close to the target bit-rate as possible. This result may suggest that the lazy mode is applicable in many cases and the compression/decompression speed is thus tremendously improved without much quality degradation.

One of the main advantages of this steganography scheme over other JPEG-based schemes is its controllable distortion during the process of information embedding. As we mentioned before, we can estimate the additional distortion in MSE in the wavelet domain quite precisely. However, in order to simplify the structure without increasing memory consumption, we presented two methods to roughly estimate the distortion introduced by information embedding, *i.e.* the method one estimating the overall MSE by adding up errors in each bit-plane and the method two utilizing the existing distortion value calculated in the tier-1 coding. We would like to verify their applicability by some experiments. Figure 3.13 shows the additional MSE measured in "Lena," "Boat," "Peppers" and "Babbon." The horizontal axis represents the number of magnitude refinement passes that are chosen for information embedding. As more passes are modified for embedding the secret data, the MSE increases accordingly. The dash lines are the actual MSE while the "+"



(a)

(b)

(c)

(d)

Figure 3.13: Additional MSE estimation of (a) Lena, (b) Boat, (c) Peppers and (d) Baboon.

and "x" mark the estimated MSE values calculated by the method one and method two respectively. We can see that the two methods perform pretty well at the beginning of the estimation but the errors accumulate as more passes are processed. An interesting phenomenon indicates that the method one seems to underestimate the distortion while the method two tends to overestimate it. Some calibrating steps may need to be taken to achieve a more accurate measurement.

In steganographic applications, we are interested in the relationship between the capacity and the resulting composite image. We embedded the four images with the same binary data and examined the MSE increase of the image due to the embedding process associated with the payload of the hidden information. We can see from Fig. 3.14 that the capacity varies in the four images even though they are compressed with the same ratio. This phenomenon should not be too surprising since the compression affects images in different ways. In our scheme, the capacity is eventually determined by the number of magnitude refinement passes that are raw coded so the distribution of the data across the three passes will directly affect the payload. Fig. 3.14 also shows that, with more information being embedded, the MSE value grows as expected. However, they do not relate to each other linearly. We take "Lena" as an example. Embedding the first 3000 bytes of the binary data only results in about 1 additional MSE of each pixel in average but embedding the next 1000 bytes quickly increases the MSE to 2. The last 100 bytes even cause the MSE to change by more than 9. Therefore, the embedding process should evaluate this curve to decide how much information is appropriate to be embedded. It should be noted that the MSE increase may be roughly estimated in conjunction with the embedding process so that we can stop embedding at the point that a minimal acceptable quality is reached.

Since the payload is quite large in our experiment, we may consider using an image with a smaller size as the intended binary data for information embedding. We compressed the image, "F-16" (128×128), into a JPEG-2000 bit-stream and embedded it into the four host images. The relationship between the PSNR of the composite image and that of the hidden image along with embedding is shown in Fig. 3.15. The result demonstrates the benefits of backward embedding, in which the less important refinement passes are used to carry the more important information of the hidden image owing to the layered structure

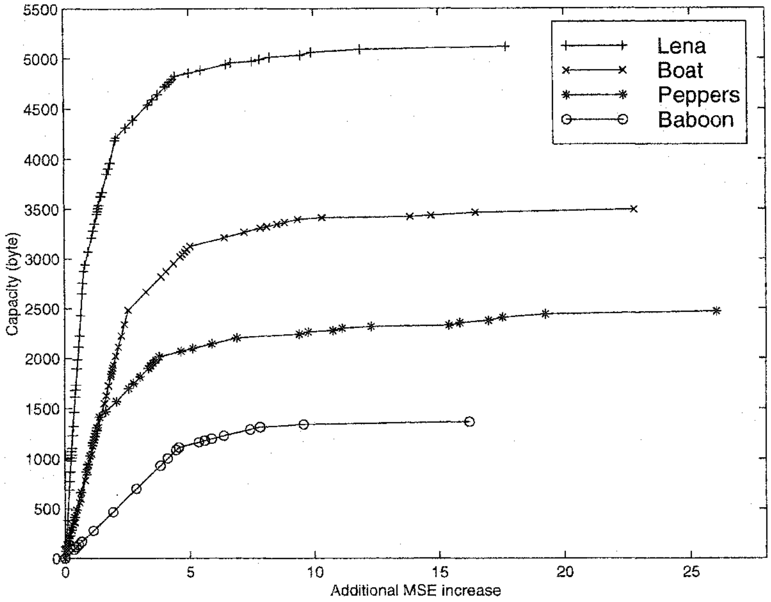


Figure 3.14: Capacity vs. additional MSE of the composite image.

of JPEG-2000. Progressive transmission of the hidden image can thus be achieved. At the beginning of the embedding, the composite image degrades little while the PSNR of the hidden image boosts quickly. At the latter part of the embedding, the large sacrifice of the composite image only helps to improve the finer detail of the hidden image so the increase of the PSNR value is limited. Under this scenario, the embedding process should proceed until both the composite and the hidden images have an acceptable quality.

3.4 Conclusion

An integrated approach to image compression and watermarking was first presented. JPEG-2000 provides various features for different applications while the proposed watermarking method can be coupled with JPEG-2000 to provide a way to assert copyright information for JPEG-2000 compressed images. The watermark sequence is embedded

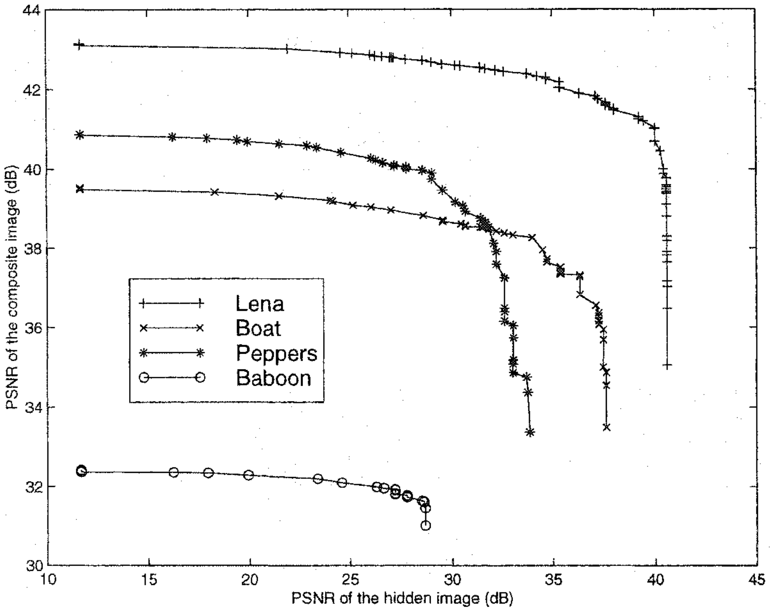


Figure 3.15: PSNR of the composite image vs. PSNR of the hidden image

into significant wavelet coefficients so that it is robust against general signal processing attacks. The detection process does not resort to the help from the original image. Progressive watermark detection is also supported so that the watermark retrieval process can be done faster. ROI watermarking is achieved easily under the same framework. This integrated scheme can be viewed as a base-line watermarking scheme, and several variants can be derived based on this basic scheme.

We then discussed the issue of steganography in JPEG-2000. Feasible information hiding schemes were proposed to reliably hide a high volume of data into JPEG-2000 compressed images for covert communication. The "lazy mode" coding option of JPEG-2000 was employed, and its usage and functions were explained and justified. Several design issues were examined to develop a well-rounded steganographic scheme in this state-of-the-art image coding standard. Experimental results were shown to demonstrate the practicability of the proposed algorithms and the decent performance.

Chapter 4

Towards Affine-Invariant Digital Image Watermarking

In this chapter, we aim at solving the challenging synchronization problem of watermark detection in digital images. The goal is to enable the embedded watermark to survive attacks such as cropping, rotation, scaling, shearing, change of the aspect ratio or other generalized geometrical attacks, *i.e.* affine attacks. The basic idea of our proposed methodology is to make use of structural grid signals for synchronized watermark detection without resorting to the original image. We will show that structural grid signal embedding can be applied to a broad range of watermarking schemes to achieve resilience to geometrical distortions.

We will first examine some important characteristics of Discrete Fourier Transform (DFT) in Section 4.1, since DFT plays a vital role in most of the schemes that are resistant to geometrical attacks. Next, we review some existing algorithms dealing with geometrical transformations in Section 4.2. Then, we describe our proposed solutions. We will present the idea of embedding and detection of structural grid signals to tackle the synchronization problem in Section 4.3. With different types of underlying watermarks being used, two affine-invariant watermarking schemes are illustrated, *i.e.* a spatial-frequency composite watermarking scheme in Section 4.4 and a block-based DCT watermarking scheme in 4.5. Experimental results of both schemes will be shown to demonstrate their good performance. Some comments on grid embedding/detection are provided in Section 4.6.1. Concluding remarks are given in Section 4.7.

4.1 Discrete Fourier Transform of Images

DFT is a powerful tool in a wide variety of signal processing applications, including filtering and spectral analysis. Efficient algorithms were developed for its numerical computation. In image watermarking, DFT is also frequently used since some characteristics of DFT can be exploited to make the embedded watermark resist geometrical modifications. Let us review some properties of DFT on 2-dimensional image data.

Let a given image be a function $f(x, y)$ defined on an integer valued Cartesian grid $0 \leq x < N_1$, $0 \leq y < N_2$. Its DFT $F(m, n)$ is defined as

$$F(m,n) = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f(x,y) e^{\frac{-j2\pi mx}{N_1} + \frac{-j2\pi ny}{N_2}}, \tag{4.1}$$

and the inverse transform is defined as

$$f(x,y) = \frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} F(m,n) e^{\frac{j2\pi mx}{N_1} + \frac{j2\pi ny}{N_2}}. \tag{4.2}$$

In some cases, it is convenient to represent the DFT coefficient by its phase θ and magnitude A, *i.e.*,

$$\begin{aligned} \theta(m,n) &= \arctan(Im[F(m,n)]/Re[F(m,n)]), \\ A(m,n) &= \sqrt{Re[F(m,n)]^2 + Im[F(m,n)]^2}, \end{aligned} \tag{4.3}$$

where $Im[\cdot]$ and $Re[\cdot]$ are the imaginary and real parts of the Fourier coefficient. It should be noted that the magnitude of the Fourier transform is even symmetric and the phase part of the Fourier transform is odd symmetric when the transform is applied on real numbers.

We then examine effects of translation, rotation and scaling of an image on its DFT coefficients.

1. Translation

Suppose that we shift the image $f(x,y)$ to the point (x^*,y^*), so that it becomes $f(x-x^*,y-y^*)$. The DFT of $f(x-x^*,y-y^*)$, $F^t(m,n)$, can be expressed as

$$F^t(m,n) = \sum_{x=0}^{N_1-1}\sum_{y=0}^{N_2-1} f(x-x^*,y-y^*)e^{-2\pi j(\frac{mx}{N_1}+\frac{ny}{N_2})}. \tag{4.4}$$

We then rewrite it as

$$F^t(m,n) = \left\{\sum_{x=0}^{N_1-1}\sum_{y=0}^{N_2-1} f(x-x^*,y-y^*)e^{-2\pi j(\frac{m}{N_1}(x-x^*)+\frac{n}{N_2}(y-y^*))}\right\} e^{-2\pi j(\frac{mx^*}{N_1}+\frac{ny^*}{N_2})}. \tag{4.5}$$

If the translation is applied circularly in both the horizontal and vertical directions, or a periodic repetition of the image in all directions is assumed, the DFT of the shifted image then becomes

$$F^t(m,n) = F(m,n)e^{-2\pi j(\frac{mx^*}{N_1}+\frac{ny^*}{N_2})}. \tag{4.6}$$

Thus, we can see that the difference of the DFT of the shifted image and that of the original image is the term $e^{-2\pi j(\frac{mx^*}{N_1}+\frac{ny^*}{N_2})}$, which is a shift in phase. In other words, the circular shift of the image will only affect the DFT phase and leave the DFT magnitude intact.

It should be noted that the cropping operation in image editing is not equivalent to a circular shifting process. We can view cropping as a combination of circular translation plus noise that comes from the deletion of some data samples.

2. Rotation

We rewrite here the DFT with $N_1 = N_2 = N$,

$$F(m,n) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{\frac{-2\pi j}{N}(mx+ny)}. \tag{4.7}$$

We can then introduce polar coordinates on (x, y) and (m, n) as follows: $x = r\cos\theta$, $y = r\sin\theta$, $m = \omega cos\phi$ and $n = \omega sin\phi$. Since $mx+ny = r\omega(\cos\theta\cos\phi+\sin\theta\sin\phi) = r\omega\cos(\theta-\phi)$, we can express the DFT as

$$F(\omega,\phi) = \sum_{r}\sum_{\theta} f(r,\theta)e^{\frac{-2\pi j}{N}r\omega\cos(\theta-\phi)}. \tag{4.8}$$

Now, we rotate $f(r,\theta)$ by θ^*. It is apparent that

$$\begin{aligned} F^r(\omega,\phi) &= \textstyle\sum_r \sum_\theta f(r,\theta+\theta^*)e^{\frac{-2\pi j}{N} r\omega\cos(\theta-\phi)} \\ &= F(\omega,\phi+\theta^*). \end{aligned} \tag{4.9}$$

Therefore, rotating the 2-dimensional data through an angle θ^* causes the Fourier representation to be rotated through the same angle, *i.e.*, $f(x\cos\theta^* - y\sin\theta^*, x\sin\theta^* + y\cos\theta^*) \leftrightarrow F(m\cos\theta^* - n\sin\theta^*, m\sin\theta^* + n\cos\theta^*)$.

3. Scaling

If we scale the 2-dimensional data by α in one axis and β in the other axis, we obtain

$$\begin{aligned} F^s(m,n) &= \textstyle\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(\alpha x,\beta y)e^{\frac{-2\pi j}{N}(mx+ny)} \\ &= \tfrac{1}{\alpha\beta}F(\tfrac{m}{\alpha},\tfrac{n}{\beta}). \end{aligned} \tag{4.10}$$

We can see from the above that scaling an axis in the spatial domain causes a reciprocal scaling in the frequency domain.

With these properties, DFT has various benefits in watermarking to resist generalized geometrical attacks as described in the following sections.

The other important application of DFT in watermarking is its efficient calculation of the circular convolution. First of all, let m be an integer. The integer $n = mod(m,N)$ is the integer for which $0 \leq n < N$ and $n-m$ is a multiple of N. Let us consider the two one-dimensional data, d_1 and d_2, with finite length N. The circular convolution $Conv_{(d_1,d_2)}$ can be expressed as

$$Conv_{(d_1,d_2)}(m) = \sum_{n=0}^{N-1} d_1(n) \times d_2(mod(m-n,N)). \tag{4.11}$$

The autocorrelation function can be viewed as a special case of the circular convolution when the data sequence d is cicularly convolved with the inverse order of itself. That is, the autocorrelation function $A_u(m)$ can be expressed as

$$A_u(m) = \sum_{n=0}^{N-1} d(n) \times d(mod(n+m,N))). \tag{4.12}$$

It should be noted that autocorretion function $A_u(m)$ has a symmetric structure, *i.e.*, $A_u(m) = A_u(-m)$ because of modulus algebra.

Calculating the circular convolution in a straightforward manner is computationally expensive when the size of data samples grows. However, DFT can be used to calculate it efficiently [65]. The convolution of two sequences d_1 and d_2 is actually equivalent to

$$Conv_{(d_1,d_2)} = \mathcal{F}^{-1}\{\mathcal{F}(d_1) \odot \mathcal{F}(d_2)\}, \tag{4.13}$$

where $\odot$ denotes the point to point multiplication, $\mathcal{F}(\cdot)$ the Fourier Transform and $\mathcal{F}^{-1}(\cdot)$ the inverse Fourier Transform.

4.2 Previous Work on the Digital Watermark Resilient to Geometrical Attacks

There are basically three main approaches to make the watermark resilient to geometrical attacks. The first approach is to embed and detect the watermark in a rotation, scaling and translation (RST) invariant domain. The other two methods attempt to determine and then invert the possible geometrical modification. One method embeds a matching template for watermark recovery while the other method, called the self-reference scheme, uses the autocorrelation function to detect the geometrical reference.

4.2.1 Watermarking in RST Invariant Domain

O'Ruanaidh *et al.* [54] proposed a watermarking method based on the Fourier-Mellin transform. They suggested that the watermark should be embedded to be invariant to common image transformations including rotation, scaling and translation (RST) so that the watermarking process is performed in an RST invariant domain. The diagram of the RST invariant watermarking scheme is shown in Fig. 4.1.

In the first step, the Fourier transform of the image is computed. Since shifting in the spatial domain results in a phase shift in the frequency domain, the image representation will be translation-invariant if we only keep the amplitude of Fourier coefficients. In the second step, the log-polar mapping (LPM) is applied so that the amplitude of the Fourier transform is changed from the Cartesian coordinate to the log-polar coordinate, *i.e.*, the

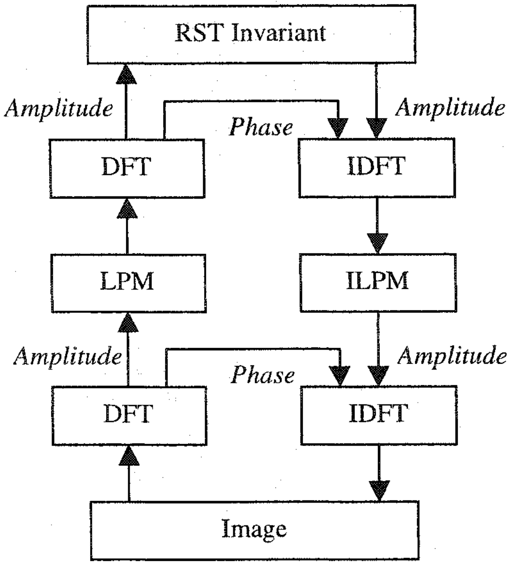


Figure 4.1: The diagram of a watermarking scheme built on the RST invariant domain.

coordinates with the logarithmic radius and the angle axes. Consider a point $(x, y) \in R$, where

$$\begin{aligned} x &= e^{\mu} \cos\theta, \\ y &= e^{\mu} \sin\theta, \end{aligned} \tag{4.14}$$

and where e^{μ} is the radius and θ is the angle in the range $(-\pi, \pi]$. Then, LPM maps a point from (x, y) to (μ, θ). We see that every point (x,y) corresponds to a unique point (μ,θ). Thus, the scaling effect can be converted to the translational effect, *i.e.*,

$$(\rho x, \rho y) \leftrightarrow (\mu + ln(\rho), \theta). \tag{4.15}$$

Similarly, the rotational effect can be converted to the translational effect, *i.e.*,

$$(x\cos(\theta + \delta) - y\sin(\theta + \delta), x\sin(\theta + \delta) + y\cos(\theta + \delta)) \leftrightarrow (\mu, \theta + \delta). \tag{4.16}$$

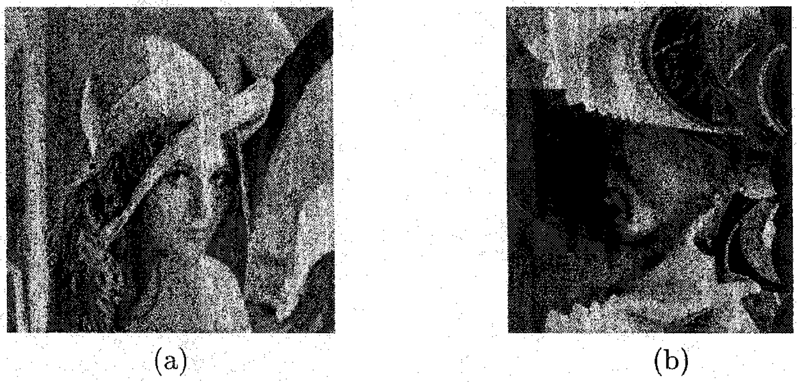
To summarize, LPM will change both scaling and rotation into horizontal and vertical shifts in the new coordinate. By computing the DFT of the log-polar map and keeping the DFT amplitude only, we are led to a rotation and scaling invariant representation.

By combining the above two steps together, we can achieve invariance in rotation, scale and translation transformations. Taking the Fourier transform of a log-polar map is actually equivalent to computing the Fourier-Mellin transform. The watermark is then embedded into the amplitude of the Fourier-Mellin transform by using the spread spectrum modulation. The watermarked image is constructed by applying inverse DFT and the inverse log-polar mapping (ILPM) twice. The phases in both Fourier transforms are not modified but simply integrated into the watermarked amplitudes for the inverse Fourier transforms. A similar method that replaces the log-polar mapping with the log-log mapping (LLM) can be used to resist the change of the aspect ratio, where the horizontal and vertical scaling ratios may be different.

Although Fourier-Mellin based watermarking is a very elegant approach, it has several limitations. First of all, the watermark will not survive under the combination of rotation and the change of the aspect ratio since neither LPM nor LLM will generate an invariant domain for watermark detection. Second, since all procedures must be done in the discrete domain, both LPM and ILPM will cause a loss of image quality from sampling. In other words, there has been information loss in converting an image to the log-polar coordinate and then transferring it back to the Cartesian coordinate without any watermarking procedure.

Consider the Lena image shown in Fig. 4.2(a) and its log-polar mapping in the spatial domain shown in Fig. 4.2(b), we see that more sampling points in the region far from the center are needed to limit the loss of information during the mapping process. This will increase the internal memory usage for intermediate image buffering. Besides, some form of interpolation is always needed when we change the coordinates. The interpolation scheme will definitely affect the performance of watermark detection. The more delicate the interpolation method is, the better precision we will achieve. Nevertheless, it demands a higher computational load. Furthermore, interpolation only performs well if neighboring samples are of a similar value. The suitability of interpolation in the transform domain is questionable.

Lin *et al.* [49] proposed another watermarking scheme. Without pursuing the "strong invariance," they embedded the watermark in the log-polar mapping of the magnitude of the Fourier transform. The detection process involves a comparison of the watermark with all cyclic rotations of the extracted watermark. Instead of directly applying spread-spectrum watermarking, which views the host image as noises, they introduced a mixing function such that the output of the mixing function is perceptually similar to the image signal and has a high correlation with the watermark. The mixing function adopted in this algorithm is a weighted sum of the image and watermark signals. We believe that the use of the mixing function is the key to its success of resisting the common rotation and scaling modifications. The mixing function should reduce the inaccuracy problem resulting from the coordinate change. A similar idea based on the Radon transform was proposed by Wu *et al.* [96].



(a) (b)

Figure 4.2: (a) The Lena image and (b) the transformed Lena image with the log-polar mapping.

4.2.2 Embedding Template for Registration

The second approach is to embed an auxiliary template in the frequency or the spatial domain of the image. The embedded information is thus composed of two parts: the watermark signal that conveys the necessary payload and a template, which usually contains no information but is used to recover the image back to its original shape. Once the template is successfully determined and the image is recovered by the reverse transformation, the watermark can be retrieved afterwards.

Fleet *et al.* [18] proposed to embed sinusoidal signals in the spatial domain of the image. The sinusoidal signals can provide a coordinate frame for registration. These sinusoids can be detected by examining the frequency domain because the frequency of the selected sinusoids rarely appears in natural images. Pereira *et al.* [57] proposed the other method to resist geometrical attacks by embedding a template in the frequency domain. Both the template and watermark signal are cast in the Fourier transform coefficients. The template consists of additionally embedded peaks in the Fourier spectrum. That is, the watermark embedder casts extra peaks in some locations of the Fourier spectrum. The locations of the peaks are predefined or selected by a secret key. The strength of the template is determined adaptively by using the local statistics of the Fourier spectrum. These embedded local peaks will reflect the geometrical modifications applied to the images because of the rotation and scaling properties as shown in Section 4.1. The watermark is embedded by a differential coding scheme. Two points that are 90 degrees apart are modified such that the difference is equal to the desired message bit. In watermark detection, all local maxima in the magnitude of the Fourier transform are extracted by using a small window. The geometrical transformation is then estimated by a point matching algorithm between extracted peaks and the reference template points. Since the extracted peaks may not be the one inserted by the watermark embedder, a method to prune the search space is proposed to reduce the computational cost. Some iterations may be needed to obtain an accurate affine matrix, which determines the possible geometrical attacks applied on the image. The matching algorithm may also be simplified with the help of LPM and LLM [56]. After the affine transform is found and the inverse affine transformation is applied, the watermark can be detected in the Fourier domain.

4.2.3 Self-Reference Scheme by Autocorrelation

The third solution is to embed a reference pattern several times into the image such that generalized geometrical transformation can be reflected by applying autocorrelation on the investigated image. Kutter [40] proposed a method to embed the same reference pattern at shifted locations. Four patterns consisting of pseudo-random numbers are embedded in the image. The four patterns are not totally different but linearly shifted copies of each other. The initial pattern is a two-dimensional random number array. The second

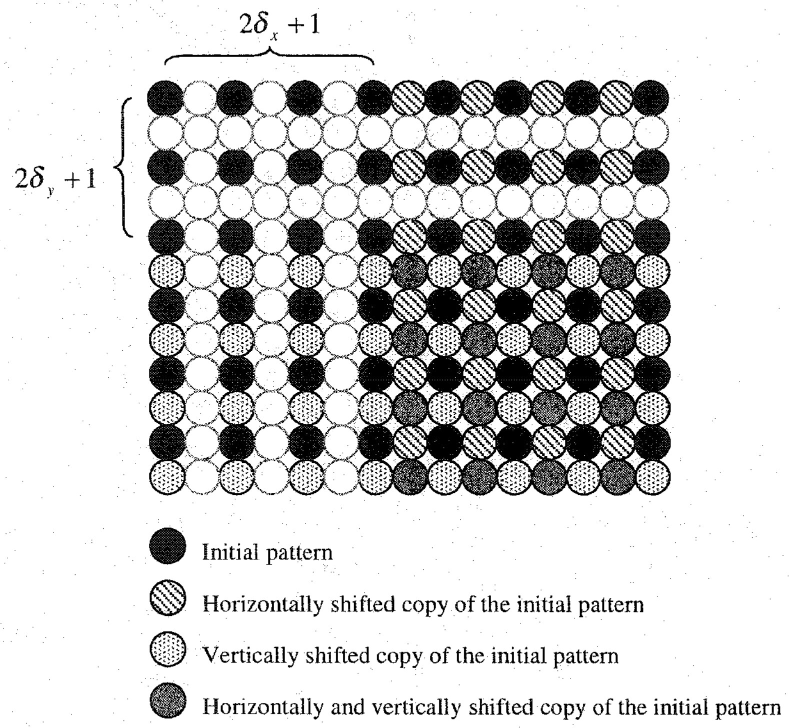

$2\delta_x + 1$
$2\delta_y + 1$
Initial pattern
Horizontally shifted copy of the initial pattern
Vertically shifted copy of the initial pattern
Horizontally and vertically shifted copy of the initial pattern

Figure 4.3: Multiple embedding of the same registration pattern, where black circles are the initial pattern, circles with lines are the horizontally shifted pattern, circles with dots are the vertically shifted pattern and gray circles are the horizontally and vertically shifted pattern.

pattern is then formed by horizontally shifting the first pattern by δ_x columns. Similarly, the third pattern is formed by vertically shifting the first pattern by δ_y. Finally, the fourth pattern is formed by shifting the first pattern by δ_x horizontally and δ_y vertically. The four patterns are embedded as shown in Fig. 4.3. The first pattern is embedded at locations with odd rows and odd columns. The second pattern is embedded in locations with odd rows and even columns. The third pattern is embedded in locations with even rows and odd columns. The fourth pattern is embedded in locations with even rows and even columns. Thus, the same random number is embedded at four different locations, *i.e.*, (x, y), $(x + 2 \times \delta_x + 1, y)$, $(x, y + 2 \times \delta_y + 1)$ and $(x + 2 \times \delta_x + 1, y + 2 \times \delta_y + 1)$.

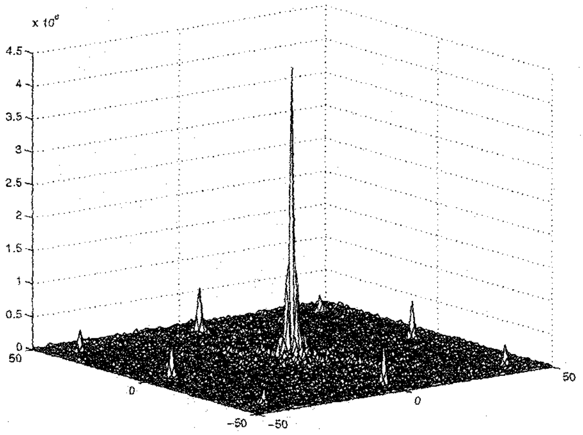


Figure 4.4: Autocorrelation of the watermarked image.

In the watermark recovery process, an estimate of the embedded pattern based on a prediction filter is applied. Then, the two-dimensional autocorrelation is computed. As shown in Fig. 4.4, nine peaks can be detected in the autocorrelation function. The center peak represents the energy of the filtered image while the other eight peaks, which are symmetric around the center owing to the symmetric structure of autocorrelation, are generated by the four embedded patterns. The locations of these peaks can be used to compute the geometrical transformation for its inversion. Experimental results show that the algorithm can resist generalized geometrical transformations including the change of the aspect ratio, rotation and shearing. However, the interleaved embedding seems vulnerable to the block-based coding scheme such as JPEG compression since the watermark is viewed as the high frequency noise. If the image is slightly scaled (or rotated) and then compressed with JPEG, the autocorrelation procedure may fail to detect peaks for registration. Besides, the watermark that carries the payload in Kutter's scheme is embedded in the spatial domain. It is not easy to perform synchronization of the spatial domain watermark when the image is cropped since the autocorrelation cannot detect the translation in this case.

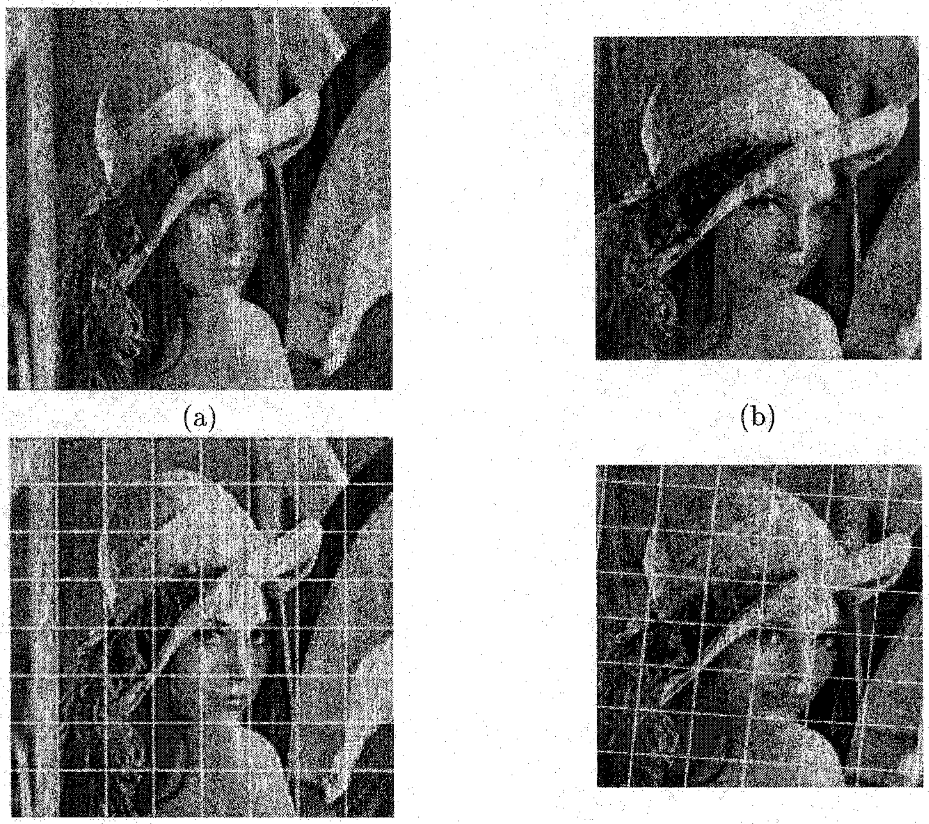
4.3 Structural Grid Signals for Synchronized Watermark Detection

Unlike filtering or lossy compression, geometrical attacks do not remove the embedded watermark but introduce the synchronization problem for the watermark detector. To resist geometrical attacks, the detector has to find a way to locate the right position where the watermark is embedded so that subsequent watermark detection can be successful. It is not an easy task since the hidden information is a weak signal but is interfered with a strong noise, *i.e.* the host image. The requirement of blind detection makes this task even more challenging. As discussed in Section 4.2, we believe that a better solution is to embed an extra signal explicitly for synchronized watermark detection. A clear advantage of using this extra signal is to avoid ambiguous detection so as to decrease the complexity of watermark detection. If a certain geometrical attack is detected undoubtedly by extracting this extra signal, the detector can easily invert the attacked image back to its original shape for watermark detection without the need of guessing and detecting repeatedly. Besides, spread-spectrum based watermark detection usually requires very precise synchronization. The auxiliary signal for synchronization will increase accuracy of geometrical recovering so that the hidden watermark can be detected much easier.

We have to first determine where this auxiliary signal for synchronization should be embedded/detected and what properties it should have. Since geometrical attacks are applied to the pixel domain directly, a straightforward but better idea is to embed and detect this signal in the spatial-domain, *i.e.* image pixels. Resorting to image transforms is usually prohibited since image transforms require a collection of image data while the structure of these data may have already been modified by geometrical attacks.

By comparing the two Lena images before and after certain geometrical attacks as shown in Figure 4.5 (a) and (b), we can tell that the image has undergone certain geometrical transformations. However, given the fact that the watermark detector does not have the original image for reference, it is difficult for the detector to be aware of this geometrical change. The degradation caused by the geometrical attacks is limited especially when we only view the attacked image. Let us impose a grid on the Lena image as shown in Figure 4.5 (c) and apply the same geometrical modifications to it. From the skewed grid as shown in Figure 4.5 (d), we have some clues about the procedures that have been operated on this image. This is basically what motivates the idea of grid signals embedding/detection to resist geometrical attacks.

Although we cannot impose a visible grid on the image as shown in Figure 4.5 (c), we can achieve this in a rather implicit manner. We construct the grid by embedding structural signals into the image. The structural signals are formed by horizontally and vertically repeating the same pseudo random pattern, or the grid pattern. The energy of the grid signals is small compared to the host image so as not to introduce any unpleasant artifacts. However, the embedding process of grid signals should take into account the host image features and human visual effects to increase the robustness against watermark attacks. Therefore, we may also view the grid signal as a spatial-domain watermark.



(a) (b) (c) (d)

Figure 4.5: (a) The original Lena image, (b) the geometrically attacked Lena image, (c) the Lena image imposed with a grid and (d) the geometrically attacked Lena image with a grid imposed.

4.3.1 Construction of the Grid Pattern

The pseudo-random pattern, or the grid pattern, used to tile the grid structure should have a good autocorrelation property, *i.e.*, the two dimensional autocorrelation function of the pattern should generate a single delta pulse. To achieve this, we first take a look at the autocorrelation function of one dimensional signal with finite length N. The autocorrelation function $A_u(m)$ of signal $d(n)$ can be expressed as

$$A_u(m) = \sum_{n=0}^{N-1} d(n) \times d(mod(n+m, N))). \tag{4.17}$$

As mentioned in Section 4.1, we can calculate (4.17) efficiently by resorting to DFT:

$$\mathbf{A_u} = \mathcal{F}^{-1}\{\mathcal{F}(\mathbf{d}) \odot \mathcal{F}^{*}(\mathbf{d})\}, \tag{4.18}$$

where $\odot$ denotes the point to point multiplication, $\mathcal{F}(\cdot)$ the Fourier Transform and $\mathcal{F}^{-1}(\cdot)$ the inverse Fourier Transform and * the complex conjugate. The autocorrelation function and signal are expressed in vector form, $\mathbf{A_u}$ and $\mathbf{d}$, respectively.

Now we require that a single delta pulse be shown in the autocorrelation function, *i.e.*,

$$\mathbf{A_u} = [\mathbf{N}, \mathbf{0}, \mathbf{0}, ..., \mathbf{0}], \tag{4.19}$$

which corresponds to

$$\mathcal{F}(\mathbf{d}) \odot \mathcal{F}^{*}(\mathbf{d}) = [\mathbf{1}, \mathbf{1}, \mathbf{1}, ..., \mathbf{1}]. \tag{4.20}$$

In other words, the magnitudes of the Fourier Transform of signal d are all equal to 1.

Therefore, to generate an $M \times M$ pseudo-random pattern with a good autocorrelation property, we can generate a random pattern and calculate its two dimensional DFT. After we force the magnitudes of the DFT coefficients equal to M to make the pattern have a unit variance and leave the phases unchanged, the grid pattern is formed by taking the inverse DFT.

4.3.2 Grid Signal Embedding

As mentioned earlier, the operations of grid signals will be performed in the spatial (or the pixel) domain. Since image features vary in different parts of the image, the embedding of grid signals should take image characteristics into account so that the embedded grid signals are robust against attacks and fulfill the visual constraint as well. In other words, we weigh the same importance on the grid signals for synchronization and the watermark carrying the necessary hidden information. There are two reasons for this design. First, the payload of digital watermarking for copyright protection is usually not high so there exists some room for extra signal embedding. Second, geometrical modifications are very common to image users. It deserves special effort to embed the auxiliary signal for synchronization so that a practical watermarking scheme can be attained.

It should be noted that the same size M will be used in all of the watermarked images so the value must be set such that all suitably large images can be protected by the watermark. Since the size of the image may not be equal to a multiple of M, boundaries of the image may be simply ignored for grid signal embedding. A rule of thumb is to set M as a multiple of 8, which will help achieve robustness against JPEG compression since the general block size of DCT adopted in JPEG compression is equal to 8.

4.3.3 Grid Signal Detection

We now consider how to detect the affine attack using the embedded structural grid signal. The affine attack can be defined by a 2×2 matrix plus translation. The whole image will be transformed by the same matrix with certain shift. In other words, for an image pixel located at (x, y), its new position, $(x^{'}, y^{'})$, after the affine transformation can be determined using the six parameters in the following model:

$$\begin{bmatrix} x^{'} \\ y^{'} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_s \\ y_s \end{bmatrix}. \tag{4.21}$$

Detection of the grid signal is based on autocorrelation, *e.g.* Kutter's scheme [40]. In our system, the structural grid signals are responsible for the determination of the four parameters in the affine matrix, *i.e.* a_{00}, a_{01}, a_{10} and a_{11}. Special care has to be taken to the two translation parameters, x_s and y_s so that the synchronization problem can be fully solved.

Under the assumption that the grid signal embedded in a pixel can be appropriately estimated or extracted, we can get the autocorrelation function as shown in Fig. 4.6(a). We represent the autocorrelation function by $A_u(x, y)$ where the central peak in Fig. 4.6(a) is denoted by $A_u(0, 0)$. If the watermarked image is not geometrically modified, we can find peaks in locations $(i \times M, j \times M)$, where i and j are integers. The distance between peaks corresponds to the block size M of the grid structure. In Figs. 4.6(b) and (c), we show the autocorrelation function after the image is rotated and scaled. It is clear that detected peaks can reflect the geometrical transformation applied to the image.

In the following two sections, we will make use of the grid embedding/detection technique to develop complete affine-invariant watermarking schemes.

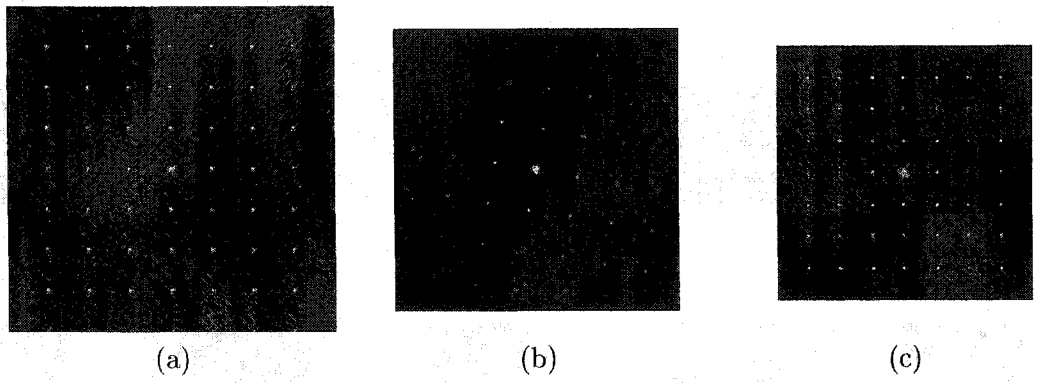

(a) (b) (c)

Figure 4.6: The autocorrelation function of the estimated grid structure in (a) the watermarked image, (b) the rotated watermarked image and (c) the scaled watermarked image.

4.4 Spatial-Frequency Composite Watermarking Scheme

In this section, we propose the spatial-frequency composite scheme to resist geometrical transformations. A frequency-domain watermark, or more specifically, Fourier-domain watermark is first embedded into the host image to convey necessary information. A spatial-domain watermark, *i.e.*, the grid signal, is embedded in the frequency-domain watermarked image to achieve image registration. Magnitudes of Fourier transform are selected for frequency-domain watermark embedding due to their insensitivity to translation of the image. The detection is done in the reverse order. The spatially embedded structure is recognized by looking at the autocorrelation function of the image, which would contain the corresponding peaks as we mentioned before. These peaks are analyzed to identify any affine distortions. The inverse affine transform is then used to adjust the attacked image back to its original orientation and scale. The frequency-domain watermark can be detected easily to unveil the information that the watermarked image carries.

We will present the system in the order of embedding and detection procedures, *i.e.*, frequency-domain watermark embedding, spatial-domain watermark embedding, spatial-domain watermark detection and frequency-domain watermark detection. The block diagram of the composite watermarking scheme is shown in Fig. 4.7.

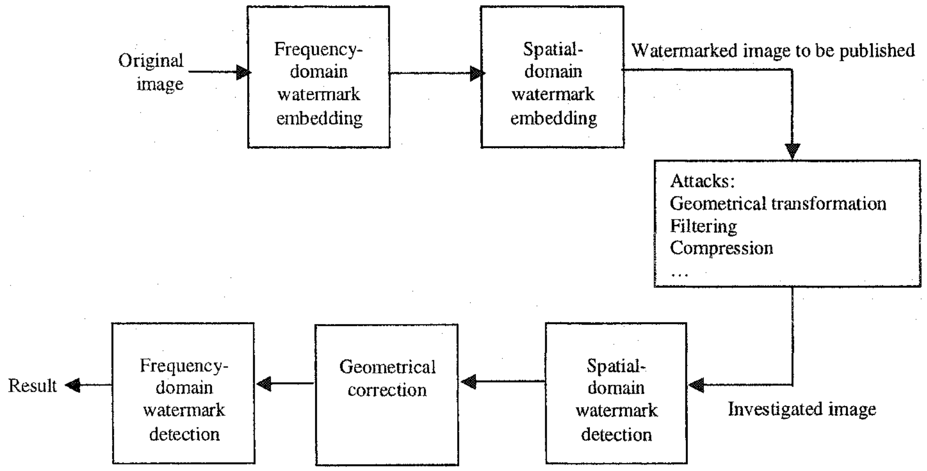
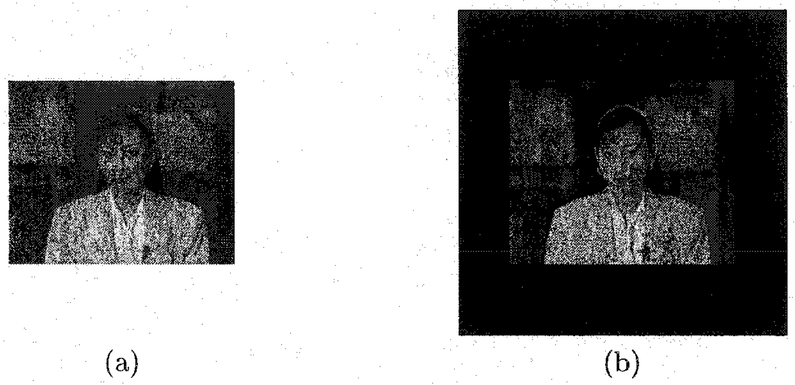

Original image
Frequency-domain watermark embedding
Spatial-domain watermark embedding
Watermarked image to be published
Attacks:
Geometrical transformation
Filtering
Compression
...
Investigated image
Spatial-domain watermark detection
Geometrical correction
Frequency-domain watermark detection
Result

Figure 4.7: The block diagram of the spatial-frequency composite watermarking scheme.

4.4.1 Frequency-Domain Watermark Embedding

The frequency-domain watermark is used to carry the necessary payload. First of all, an N by N DFT is applied to the original image, where N is chosen to be an integer power of 2, *i.e.*, $N = 2^k$. One advantage of this choice is to make the DFT calculation more efficient. Besides, by setting this agreement between the watermark embedder and the watermark detector, the requirement of the presence of the original image can be avoided in the detection phase. However, the width w and the height h of the image may not be the integer power of two. We can form a temporary image of size 2^k by 2^k, where $2^{k-1} < max(w, h) <= 2^k$, by padding the image with zero. An example is shown in Fig.

4.8. If the size of the original image is 352×288, a temporary image with size 512×512 is generated by zero-padding the part surrounding the original image. After the watermark is embedded, we simply crop the added portion to keep the original size of the image. We do not expect that the image will be modified to a very large degree since a substantial amount of visual degradation will be introduced and the commercial value of the image can be significantly reduced by doing so.



(a)

(b)

Figure 4.8: (a) The original image of size 352×288 and (b) the padded image of size 512×512 for watermark embedding.

The watermark is embedded in DFT coefficients of the middle frequency range. Low frequency coefficients are avoided since most of the energy of the image resides in the low frequency range. Even a slight modification in low frequency coefficients will result in serious perceptual distortion. The high frequency range is also ignored for watermarking since lossy image coders tend to remove high frequency coefficients, which are usually not significant to the human visual system. Thus, after applying DFT and shifting the spectrum to make the DC coefficient appear at the center, we embed the watermark in coefficients that are located within a circular strip with radius between r_h and r_l.

The other issue worth our attention is the cross artifact in the image's energy spectrum. Let us take a look at the "Lena" image and its DFT magnitude as shown in Figs. 4.9 (a) and (b), respectively. We see that there exist strong energy coefficients along the horizontal and vertical axes. No matter how the image is cropped or scaled, the cross artifact will still appear near the axes since the cross artifact is not related with the frequency content of the underlying image data but the discontinuity of pixel values on the image boundaries. The energy of discontinuous boundaries will be more dominant if we pad the image with zero when the image size is not equal to the integer power of two. To make the embedded watermark reliably detected by the watermark detector, we should avoid embedding the watermark in the region that is close to the horizontal and vertical axes. We only embed the watermark in the angular range of $(\delta^\circ, 90^\circ - \delta^\circ)$, $(90^\circ + \delta^\circ, 180^\circ - \delta^\circ)$, $(-\delta^\circ, -90^\circ + \delta^\circ)$ and $(-90^\circ - \delta^\circ, -180^\circ + \delta^\circ)$ to avoid the region with the cross artifact. The region for watermark embedding (and detection) in the Fourier spectrum can thus be shown in Fig. 4.9(c). The four sectors are chosen and the other black portion is omitted.

It should be noted that magnitudes of DFT coefficients are even symmetric, *i.e.*, they are symmetric around the center since the image data take real values. When embedding the watermark in the Fourier domain, we should keep this symmetric property. Thus, the watermark will first be embedded in coefficients in the upper two sectors. The resulting watermarked coefficients will be mapped to coefficients in the lower two sectors.

After determining the region for watermarking, we start to embed the watermark in each of the coefficients in the region. We only modify magnitudes of Fourier coefficients and leave the phase unchanged. The magnitude of the watermarked coefficient, $|C^{'}(x,y)|$ is formed by

$$|C^{'}(x,y)| = |C(x,y)| \times (1 + \alpha_f \times w_f(x,y)), \tag{4.22}$$

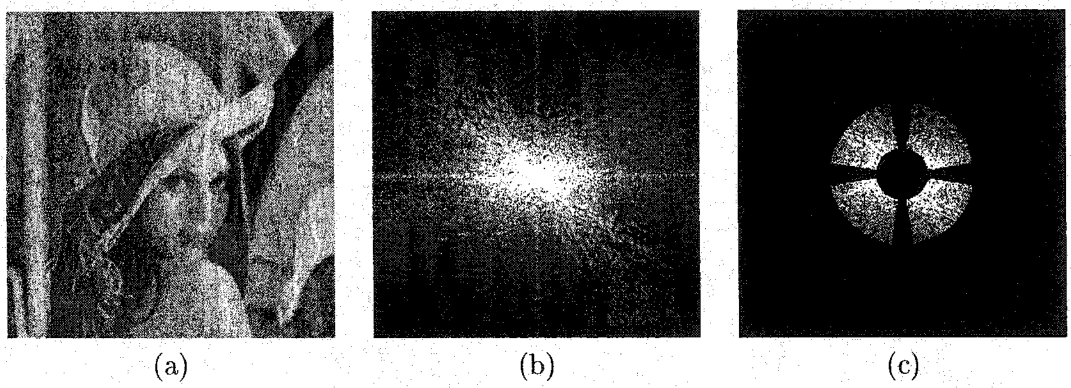


Figure 4.9: (a) The Lena image, (b) magnitudes of the DFT coefficients of the Lena image and (c) the region chosen for watermarking.

where $C^{'}(x, y)$ is the original coefficient, and $w_f(x, y)$ is the watermark symbol, which can be of real or binary value. Parameter α_f is used to scale the total watermark energy. We also use the magnitude of the coefficient to weigh the embedded watermark to avoid adding too much watermark energy on a coefficient with a smaller value. In order to make the frequency-domain watermark less sensitive to the error of the rotation and scaling in the recovery process, the same watermark symbol may be embedded in a larger region instead of a single point. We embed the watermark by examining the position of the coefficient, that is, the angle and the radius. Consequently, the watermark is embedded by using

$$|C^{'}(r, \theta)| = |C(r, \theta)| \times (1 + \alpha_f \times w_f(r, \theta)), \tag{4.23}$$

where we calculate radius r and angle θ and round them to nearest integers. The corresponding watermark symbol is then embedded on the magnitude. Finally, the watermarked coefficient is formed by combining the watermarked magnitude with the original phase.

The other point we would like to emphasize is that the watermark value is horizontally (or vertically) symmetric, *i.e.* $w_f(x, y) = w_f(x, N - y)$. The major concern here is to make the watermark scheme resist the mirror attack, *i.e.*, flipping the image horizontally around the vertical axis. In many cases, such as landscape images, mirroring an image is apparently acceptable to most of image users. Many existing watermarking schemes do not take this issue into consideration so that the complexity of the watermark detector is increased for the need to detect the watermark in the flipped image once more. In fact, flipping an image horizontally causes the magnitude of the Fourier spectrum to flip as well. Therefore, embedding the same watermark value in the two positions horizontally symmetric to each other will allow the watermarking scheme to be resilient to mirror manipulation.

One drawback of embedding the watermark in the Fourier domain is the ignorance of local statistics in the host image [49]. Although the PSNR value of the resulting DFT watermarked image with respect to the original image is quite large (usually larger than 45dB), the embedded watermark could cause visible artifacts in images with various characteristics. Let us consider the image "Statue of Liberty" as shown in Fig. 4.10(a). We enhance the energy of the watermark to exaggerate the artifact introduced by the frequency-domain watermark embedding for explanation. From the watermarked image shown in Fig. 4.10(b), we see that the watermark is hidden well in the statue, but it becomes quite visible in the sky. Therefore, a postprocessing procedure in the spatial domain is required to make the embedded watermark adapt to the local characteristics of the image. The spatial postprocessing procedure should take human visual model into account to decide the maximum amount that a pixel can be changed while invisible to the human being. Methods such as clamped high-pass filter [34] or Girod's spatial masking model [25] maybe suitable for preventing the embedded watermark from being perceived.

The evaluation of the two methods has been reported in [52]. For both simplicity and generality, we adopt the clamped high-pass filter as the tolerable error level, which defines the maximum offset a pixel can be made. The effect is that the energy of the embedded watermark can be constrained in visually sensitive regions.

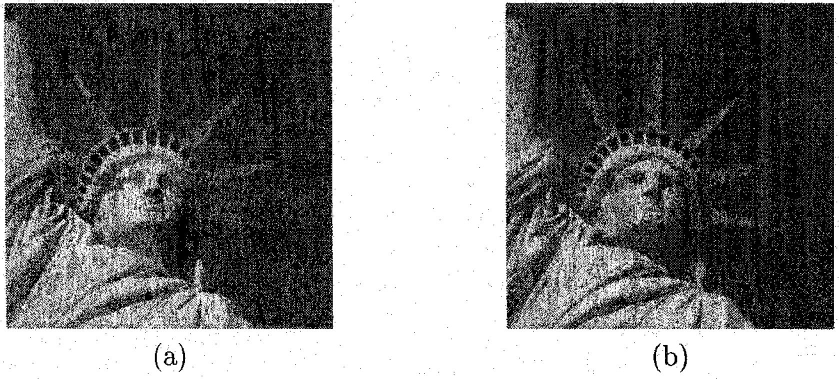


Figure 4.10: The embedding of a stronger watermark in an image with varying characteristics at different locations: (a) the original "Statue of Liberty" (b) the watermarked "Statue of Liberty"

4.4.2 Spatial-Domain Watermark Embedding

The pseudo random pattern is repeatedly embedded in the image in a tiled grid as shown in Fig. 4.11 to form the spatial-domain watermark, *i.e.*, the grid signal.

Let the pixel of the image output by the DFT watermarking process described in Section 4.4.1 be denoted by $I(x,y)$, where (x,y) is the position, $0 \leq x < W$, $0 \leq y < H$, and W, H are the width and height of the image, respectively. The watermarked pixel $I^{'}(i,j)$ is formed by

$$I^{'}(x,y) = I(x,y) + \alpha_s \times \lambda(x,y) \times w_s(mod(x,M), mod(y,M)), \tag{4.24}$$

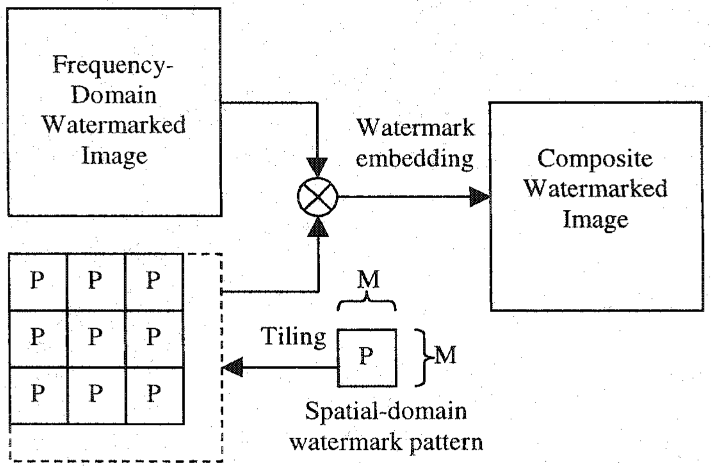

Frequency-Domain Watermarked Image
Watermark embedding
Composite Watermarked Image
P P P
P P P
P P P
M
Tiling
P
M
Spatial-domain watermark pattern

Figure 4.11: Spatial-domain watermark embedding.

where α_s is the global weighting factor and w_s is the corresponding spatial-domain watermark symbol in the pseudo-random pattern. The local weighting factor, $\lambda(x, y)$, is again determined by the clamped high-pass filter. That is, $\lambda(x, y)$ is obtained by filtering the DFT-watermarked image with a Laplacian high-pass filter

$$F_c = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \times \frac{1}{8}, \tag{4.25}$$

taking the absolute value and limiting the offset within a predefined maximum value, λ_{max}. Filtering with the clamped high-pass filter can be viewed as an activity measurement so that stronger watermark energy will be embedded in the area with more activity while less watermark energy is added in the smooth region to guarantee the invisibility constraint of the watermark.

4.4.3 Spatial-Domain Watermark Detection

As we mentioned before, the affine attack can be defined by an affine matrix plus translation. The structural grid signals will be responsible for determining the four parameters in the affine matrix while certain strategies have to be taken to deal with the translation. Due to the translation-invariant property of the Fourier magnitude, the two shift parameters will not affect the watermark payload as we choose to embed the watermark in Fourier magnitudes.

The spatial-domain watermark is detected based on autocorrelation. With the help from FFT, we can calculate the autocorrelation function very efficiently. Before calculating the autocorrelation function of the investigated image to extract embedded peaks for geometrical correction, we perform the prefiltering on the image. The necessity of the prefiltering process can be explained below.

First, the correlation detector is known to be optimal in an additive white Gaussian channel, when the receiver has full knowledge of the modulation function used to transmit the messages. However, the channel noise in watermarking, *i.e.*, the image content, does not have this property since the mean value is clearly not close to zero and the distribution of pixel values varies in different images. Besides, the strong correlation between image pixels will also undermine the performance of the correlation detector. Therefore, the prefiltering process can decorrelate the image pixels on one hand and change the noise distribution without affecting watermark detection on the other hand. The filtered image will have the mean close to zero and a decreased variance. The probability distribution of the filtered image can possibly be modeled as the generalized Guassian noise so that the performance of watermark detection is boosted [16].

Second, we can view the prefiltering as a watermark prediction filter. The Fourier spectrum of the spatial domain watermarked image is shown in Fig. 4.12. The small watermark peaks appear in the figure across the low to high frequency bands. Each small peak is separated from each other by a fixed distance as a result from the repeated embedding of the spatial-domain watermark. If some of these watermark peaks can be successfully extracted or estimated, the spatial domain watermark can be detected.

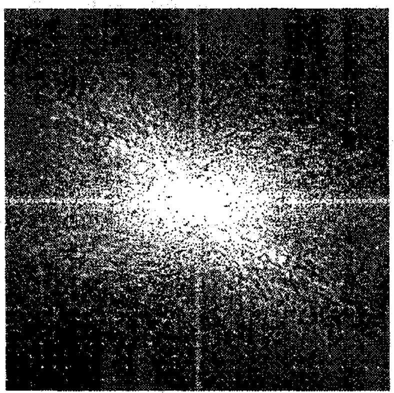


Figure 4.12: DFT magnitudes of the composite watermarked image.

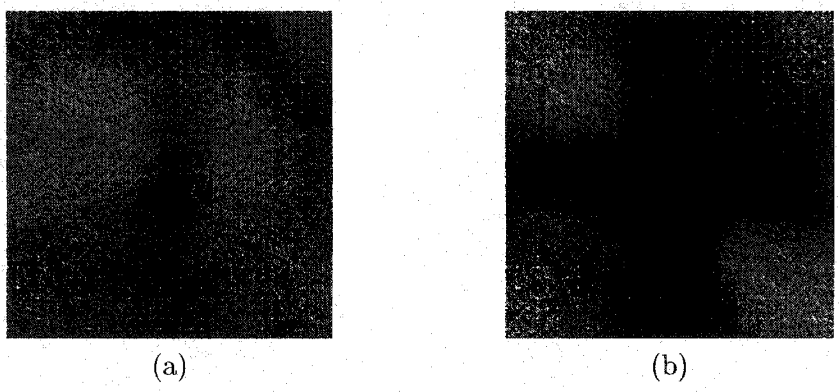
To further understand the usage of estimation filters, we examine two types of filters applicable in spatial-domain watermark detection since they can make the prefiltering process computationally inexpensive. The two examined filters are

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{4.26}$$

and

$$H_2 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \tag{4.27}$$

The Fourier spectrum of the two filtered images by using filters H_1 and H_2 are shown in Fig. 4.13 (a) and (b), respectively. We see that the image filtered by H_1 has more energy in lower to middle frequency components while the image filtered by H_2 has stronger energy in high frequency coefficients. In terms of watermark prediction, it would be better to choose H_2 because watermark can be successfully extracted by filtering out the image signal. However, lossy image compression tends to remove the insignificant high frequency coefficients. Most of the watermarks existing in high frequency coefficients will be removed and the filter may only preserve noise introduced by JPEG compression. Therefore, H_2 may not work well when the watermarked image is geometrically modified and compressed with lossy image codecs. If we would like to accommodate image compression, H_1 seems better in detecting the watermark because it can predict the watermark in the middle frequency coefficients, which will also be preserved in lossy compression.


(a)

(b)

Figure 4.13: The Fourier spectrum of an image filtered by (a) H_1 and (b) H_2.

However, there exists another problem that may limit the usage of H_1. Let us consider the image shown in Fig 4.14(a). Black boundaries are added to sides of the image. Images with boundaries are quite common, such as one frame of the video in letter-box format. If we apply the cross shape filter H_1 to filter the image and calculate its autocorrelation, there will be three horizontal and three vertical ridges as shown in Fig 4.14(b). These ridges will dominate and add difficulty in extracting autocorrelation peaks especially when the image is compressed. These ridges are generated owing to the structure of the filter. Note that the sum of the raw or column cofficients of H_1 is a positive number. After prefiltering the image, there will be a positive line along the black boundary. These positive numbers will generate ridges when we calculate the two-dimensional autocorrelation function. Although the problem could be solved by detecting ridges and selecting other peaks, we actually prevent this situation by changing the filter structure. For example, ridges will disappear as shown in Fig 4.14(c) when we apply H_2 because the sum of each raw and column of this Laplacian operator is zero. Without those ridges, detection of the peaks will become easier and efficient.

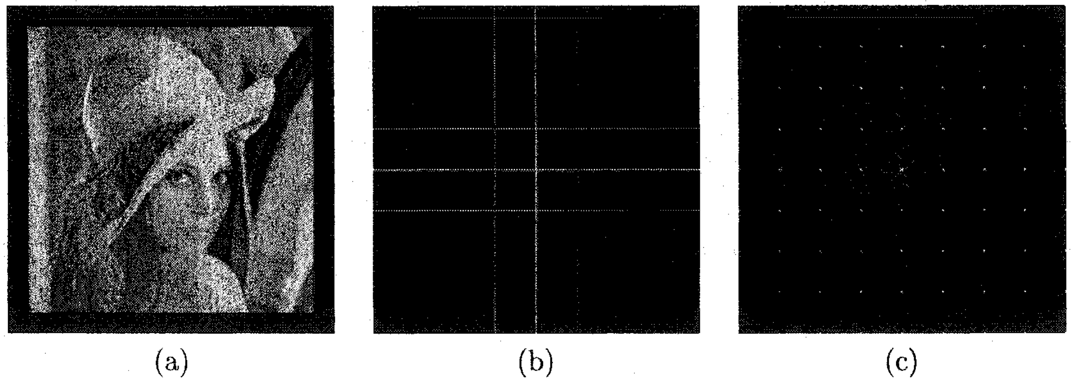

(a) (b) (c)

Figure 4.14: (a) The Lena image with black boundaries, (b) the autocorrelation function of the H_1 filtered image and (c) the autocorrelation function of the H_2 filtered image

Based on the above consideration, a possible better filter with the sum of its raw and column coefficients equal to zero is shown as follows.

$$H_3 = \begin{bmatrix} 0 & 1 & -2 & 1 & 0 \\ 1 & 2 & -6 & 2 & 1 \\ -2 & -6 & 16 & -6 & -2 \\ 1 & 2 & -6 & 2 & 1 \\ 0 & 1 & -2 & 1 & 0 \end{bmatrix} \tag{4.28}$$

The H_3 filter tends to keep the balance between compression and the accuracy of watermark prediction so the watermark can be detected easier after the image is geometrically modified and then compressed with mild JPEG compression.

A more sophisticated way to detect the spatial-domain watermark is to make use of Wiener filtering. The first step is to predict the original signal and the second step is to subtract the investigated signal by the predicted original signal to retrieve the spatial-domain watermark. In our implementation, an adaptive Wiener filtering algorithm [48] is adopted. The space-variant filter $h(n_1, n_2)$, with $-M \leq n_1, n_2 \leq M$, is determined by

$$h(n_1, n_2) = \begin{cases} \frac{\sigma_o^2 + \frac{\sigma_w^2}{(2M+1)^2}}{\sigma_o^2 + \sigma_w^2}, & n_1 = n_2 = 0 \\ \frac{\frac{\sigma_w^2}{(2M+1)^2}}{\sigma_o^2 + \sigma_w^2}, & 0 < |n_1| \leq M, 0 < |n_2| \leq M \\ 0, & otherwise \end{cases} \tag{4.29}$$

where σ_o^2, σ_w^2 are the variances of the original image and the watermark, respectively. Instead of assuming a fixed σ_o^2 and σ_w^2 for the entire image, they are estimated locally. The variance $\sigma_w^2(x, y)$ of the watermark is estimated by calculating the local variance of the investigated image filtered by F_c as shown in (4.25) and multiplying it by α_s^2. The variance σ_o^2 of the original signal is estimated by

$$\sigma_o^2 = \begin{cases} \sigma_i^2(x,y) - \sigma_w^2(x,y), & \sigma_i^2(x,y) > \sigma_w^2(x,y) \\ 0, & \sigma_i^2(x,y) \leq \sigma_w^2(x,y) \end{cases} \tag{4.30}$$

where $\sigma_i^2(x,y)$ is the local variance of the investigated image.

After prefiltering the image or predicting the spatial-domain watermark and calculating its autocorrelation function, we calculate the second derivative to make the peak detection process easier. That is, we filter the autocorrelation function by the filter shown in (4.27). Then we will be able to detect peaks if the spatial-domain watermark exists. A simple solution for peak detection is to make use of the eight peaks around the center. To detect those peaks, first of all, we ignore the region around the center for peak detection since the appearance of peaks in this region may be caused by the image structure itself instead of the embedded watermark. Next, we find the local maximum that is closest to the center of the image. The peak is assumed to be one of eight peaks closest to the center. The determination of the local maximum or peak is done by examining if $A_u(x,y)/A_u(0,0)$ exceeds some threshold. It is actually the correlation coefficient of the extracted watermark with its shifted version. To detect other peaks, we will erase values around peaks already detected, and the values of the opposite peaks across the center in the autocorrelation function because of its symmetric structure, *i.e.*, $A_u(x,y) = A_u(-x,-y)$. The other local maximum closest to the center is again chosen as the peak, which will also be assumed as one of the eight peaks closest to the center. It should be noted that utilization of other peaks will definitely help determine a more accurate affine matrix because of the periodic structure of the peaks.

However, the position of the detected peak $(\tilde{x}, \tilde{y})$ will always take an integer value. To increase the accuracy of detected peaks, we use the value of the autocorrelation function in a square block centered at $(\tilde{x}, \tilde{y})$ of size $(2\delta + 1) \times (2\delta + 1)$ to determine a more precise position of the peak. The final position of the peak can be calculated by

$$\begin{aligned} x^* &= \tilde{x} + \tfrac{1}{\mathcal{K}} \textstyle\sum_{i=\tilde{x}-\delta}^{\tilde{x}+\delta} \sum_{j=\tilde{x}-\delta}^{\tilde{x}+\delta} \left\{ A(i,j) \times (i - \tilde{x}) \right\}, \\ y^* &= \tilde{y} + \tfrac{1}{\mathcal{K}} \textstyle\sum_{i=\tilde{y}-\delta}^{\tilde{y}+\delta} \sum_{j=\tilde{y}-\delta}^{\tilde{y}+\delta} \left\{ A(i,j) \times (j - \tilde{y}) \right\}, \end{aligned} \tag{4.31}$$

where $A(i,j) = Max(A_u(i,j), 0)$ and $\mathcal{K}$ is the number of nonzero $A(i,j)$ excluding $A(\tilde{x}, \tilde{y})$. From empirical data, the precision can be increased by this weighting process.

Given the location of detected peaks (x^*, y^*), the correction of the geometrical attack becomes a point-matching problem. The matching process can be done in a simple way because of the periodic structure of the peaks. By determining the relationship between positions of a point before and after the transformation, the affine matrix defined in (4.21) can be found. If the image is purely scaled and/or rotated, two parameters will be needed to describe the affine matrix since it can be determined by the rotation angle and the scaling ratio. In this case, we only need one peak to determine the affine matrix. If the image is modified by shearing, the change of the aspect ratio or other linear transformations, we need two peaks to calculate the correct affine matrix.

Let us denote positions of the two points before transformation by (x_1, y_1), (x_2, y_2), and positions of the two detected points after transformation by (x_1^*, y_1^*), (x_2^*, y_2^*). The inverse affine matrix can be calculated by using

$$\begin{bmatrix} a_{00}^* & a_{01}^* \\ a_{10}^* & a_{11}^* \end{bmatrix}^{-1} = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} \times \begin{bmatrix} x_1^* & x_2^* \\ y_1^* & y_2^* \end{bmatrix}^{-1} \tag{4.32}$$

4.4.4 Frequency-Domain Watermark Detection

Before detecting the frequency-domain watermark, we first use the inverse affine transform to recover the image back to the original scale and orientation. For the pixel of the investigated image at location $(x^{'}, y^{'})$, its recovered position (x^*, y^*) can be calculated via

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} a_{00}^* & a_{01}^* \\ a_{10}^* & a_{11}^* \end{bmatrix}^{-1} \begin{bmatrix} x^{'} \\ y^{'} \end{bmatrix} \tag{4.33}$$

Since x^* and y^* may not take integer values, some interpolation method must be used. Note that our scheme does not require a very accurate precision and linear interpolation method is adopted. Although the recovered image does not have good quality, the watermark can be successfully detected since we embed the watermark in the middle-frequency coefficients of the image. The finer parts of the image represent the high frequency components, which are not required in watermark detection.

However, we should pay attention to the case of rotation attacks. Let us consider the image shown in Fig. 4.15(b), which is generated by rotating the original "Lena" image (Fig. 4.15(a)) 15 degrees to the right. The size of the two images are different since it is assumed that all the image users will remove the additional background, *i.e.*, the portion that is not originally appear in the image (usually filled with zero value in most of the image editors), since slanting boundaries do not look pleasant to image users. In watermark detection, we apply the autocorrelation process and correctly convert the attacked image to its original orientation and scale as shown in Fig. 4.15(c). From the Fourier spectrum of the corrected image as shown in Fig. 4.15(d), the cross artifact will not be constrained near the horizontal and vertical axes. These additional artifacts come from slanting boundaries as we recover the image and pad the background with zeros. One method to get rid of these cross artifacts is to crop only the central part of the recovered image as most of the image users will do. However, the effect of cropping an image on its Fourier coefficients is equivalent to shifting the image and adding noise to the Fourier coefficients. Cropping more image content will definitely affect the detection of the frequency-domain watermark. Therefore, instead of cropping the image, we choose to reduce the cross artifacts by filling the additional background of the image with its mean value (Fig. 4.15(e)). In normal cases, this procedure will decrease the discontinuity between image boundaries and other parts of the image. By comparing Fig. 4.15(d) and (f), we see that the cross artifact is significantly reduced and the watermark detection will thus be improved.

After recovering the image to its original shape, we paste it on an $N \times N$ block, where N is the smallest number equal to the power of two that can cover the recovered image. In normal cases, the size should be the same with the one used in embedding. If the computational load is not a serious issue in watermark detection, $2N \times 2N$ DFT can also be tested if the watermark is not found in $N \times N$ DFT. The other method is to calculate

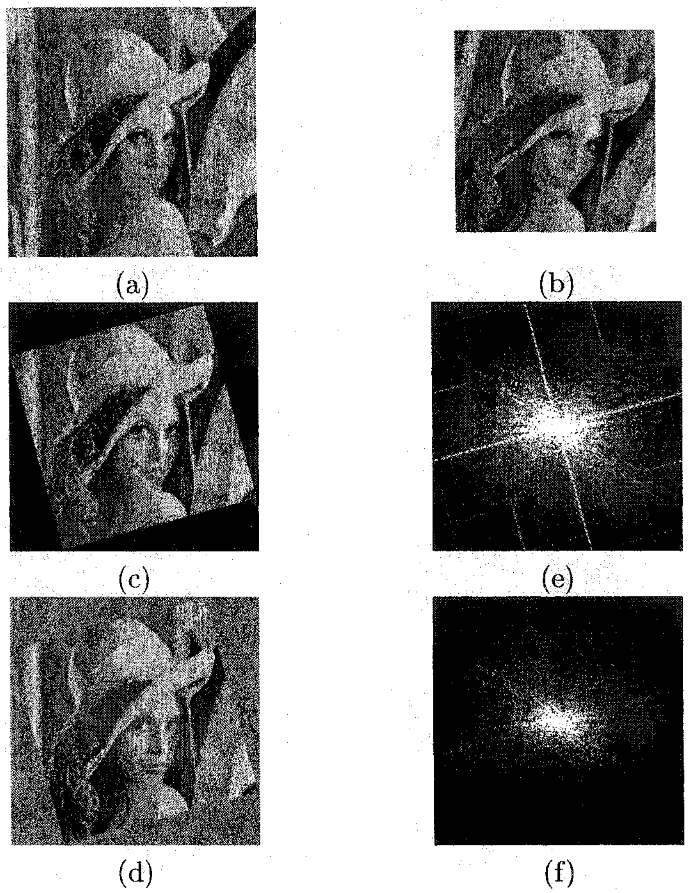


Figure 4.15: (a) The original Lena image, (b) the image rotated by 15 degrees and cropped, (c) the recovered image by filling the background with zeros, (d) DFT magnitudes of the zero-padded recovered image, (e) the recovered image by filling the background with the mean value and (f) DFT magnitudes of the mean-filled recovered image.

the DFT in a larger dimension, e.g. $4N \times 4N$. If the watermark is not detected, we then detect the watermark from the downsampled coefficients.

In frequency-domain watermark detection, we encounter the same problem in spatial-domain watermark detection since the magnitudes of the Fourier transform only take positive numbers so the distribution is definitely non-Gaussian. We follow the same strategy as adopted in spatial-domain watermark detection, *i.e.*, filtering Fourier magnitudes via the use of a prediction filter.

The correlation response is used to determine the existence of a watermark. We need only the two upper sectors in the DFT magnitudes as shown in Fig. 4.9 to detect the watermark. Let $C_1^*(r,\theta)$ and $C_2^*(r,\theta)$ be the filtered magnitudes of the Fourier coefficients in the upper/left sector and the upper/right sector, respectively. $w_f^{'}(r,\theta)$ is the tested watermark symbol at the corresponding position. The correlation response is calculated by

$$\rho = \frac{\sum_{r=r_l}^{r_h} \sum_{\theta=\delta}^{\frac{\pi}{2}-\delta} \left\{ C_1^*(r,\theta) \times w_f^{'}(r,\theta) \right\} + \sum_{r=r_l}^{r_h} \sum_{\theta=\frac{\pi}{2}+\delta}^{\pi-\delta} \left\{ C_2^*(r,\theta) \times w_f^{'}(r,\theta) \right\}}{\sqrt{\sum_{r=r_l}^{r_h} \sum_{\theta=\delta}^{\frac{\pi}{2}-\delta} C_1^{*2}(r,\theta) + \sum_{r=r_l}^{r_h} \sum_{\theta=\frac{\pi}{2}+\delta}^{\pi-\delta} C_2^{*2}(r,\theta)}} \tag{4.34}$$

The weighting factor shown in the denominator is to make the correlation response have a normal distribution based on the Central Limit Theorem. Consequently, the detection threshold can be set according to the Q function to maintain the false positive rate below a given probability as discussed in the threshold analysis of JPEG-2000 watermarking in Section 3.2.6.

Since the frequency-domain watermark carries information about the image, let us examine its capacity. The energy in an image is seldom evenly distributed in the angular frequency of the Fourier spectrum. Images frequently have a large amount of energy in one group of directions, while having lower energy in the orthogonal group of directions. Therefore, one possible method is to divide the Fourier spectrum into S sections according to different angles. The value of S is limited by the spreading gain as the spread-spectrum watermarking method is utilized. We calculate the summation of the correlation responses in one sector and the other sector ninety degrees apart from it. The sign of the correlation response of the summation can be used to represent one-bit information. In this case, the correlation response in (4.34) should be modified to be the sum of the absolute value of the correlation response in each binary data. If we would like to increase the security of the system, we can consider the other method, which makes use of a secret key to decide random coefficients for calculating the correlation response of each bit. This method is applicable if the image has been geometrically corrected and the synchronization problem has been solved by using the spatial-domain watermark.

4.4.5 Experimental Results

The Lena image of size 512×512 is used in this section for experiments. To begin with, we describe parameters to be used in the composite system. In frequency-domain watermark embedding, the watermark is embedded in coefficients with its radius in the range of [32,128] to achieve the balance between robustness and visual quality and its angle in the range of $(9^\circ, 81^\circ)$, $(99^\circ, 171^\circ)$, $(-9^\circ, -81^\circ)$ and $(-99^\circ, -171^\circ)$ to avoid the region with the cross artifact. The watermark weighting factor α_f in (4.23) is set to 0.6. The bi-polar watermark is used, *i.e.*, the watermark symbols take two values, -1 and 1. In spatial-domain watermarking, the size M of the block is set to 64 so that the image of size at least 128×128 can be protected by the composite scheme. The global watermark weighting parameter, α_s, is set to be 0.1. The maximum offset of the watermarked image from the original image is 9. In frequency-domain watermark detection, our goal is to achieve a false positive rate of 10^{-8} or lower while the correlation response is assumed to be a random variable with a normal distribution. Therefore, the threshold value is set to 5.61. The PSNR value of the composite watermarked image with respect to the original image is 38.3dB. The resulting watermarked image is shown in Fig. 4.16(b).

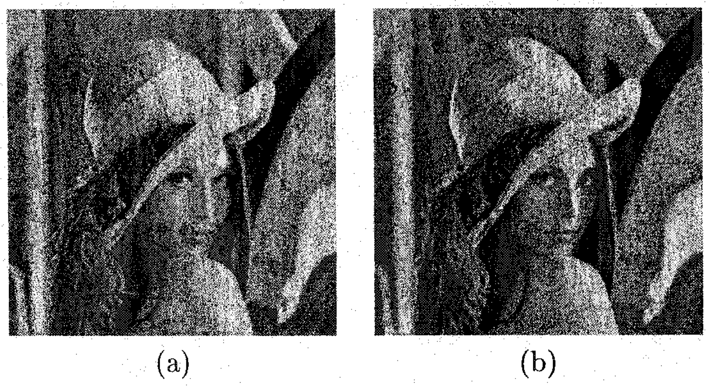


Figure 4.16: (a) The original "Lena" and (b) the watermarked "Lena."

StirMark [59] provides a generic set of tools for the robustness test of image watermarking algorithms. We use this benchmark tool to test the resilience of the proposed composite watermarking system to various attacks. The attacks performed by StirMark can be classified into two categories. The first category is the noise-type attack, which includes filtering, color quantization and JPEG compression. The second category is the geometrical attack, which includes general image editing operations such as cropping, rotation, scaling and the random geometrical distortion, which applies varying types of geometrical attacks to different regions of an image. Up to now, no satisfactory solution has been found to deal with the random geometrical distortion. We will focus on the noise-type and the generalized geometrical attacks in our experiments. Since most users will process the image and then save the image in a compressed format, most experiments are done with JPEG compression.

1. JPEG Compression

   First, we test the JPEG compression attacks on the watermarked image. Since the compression attack does not change the geometry of the image, we detect the watermark from the frequency-domain directly. The JPEG quality factor of the tests vary from 90 to 10. The experimental results are shown in Fig. 4.17. All correlation responses are well above the threshold value 5.61, which is indicated as a broken line. Therefore, our frequency-domain watermark survives JPEG compression quite well.

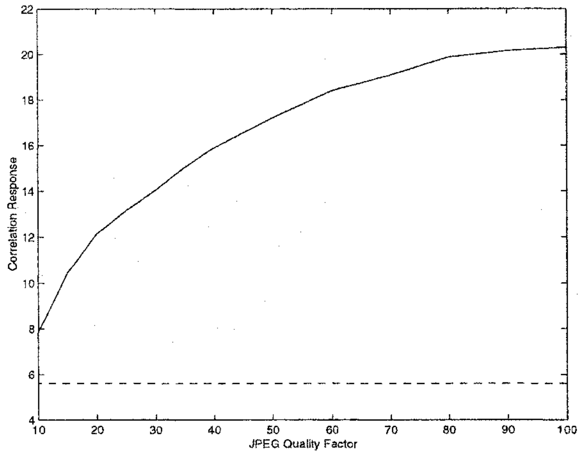


Figure 4.17: Robustness test of JPEG compression.

2. Filtering

Next, we consider the filtering attack. StirMark performs six types of filtering attacks including 2×2, 3×3, 4×4 median filters, frequency mode Laplacian removal attack [3], 3×3 Gaussian filtering and 3×3 sharpening. The correlation responses in the above six cases are 18.66, 18.42, 17.22, 10.57, 19.14 and 23.38, respectively. We see that the embedded watermark is robust against these filtering processes as expected.

3. Cropping

In most cases, image users are interested in the central part of the image. StirMark crops the images by removing 1, 2, 5, 10, 15, 20, 25, 50 and 75% of borders. The correlation responses are shown in Table 4.1. As shown in the table, the watermark is successfully detected if the size of the remaining image is larger than a quarter of

Width Cropped(%)	Width	Region Left(%)	Correlation Response
0	512	100.00	20.29
1	506	97.67	18.04
2	501	95.75	14.55
5	486	90.10	16.19
10	460	80.72	14.89
15	435	72.18	11.98
20	409	63.81	10.99
25	384	56.25	12.35
50	256	25.00	8.08
75	128	6.25	0.26

Table 4.1: The correlation responses of different cropping attacks.

the original image. The watermark is not detected when the image is cropped more than 75% of rows and columns. However, the commercial value of this small image has lost.

4. Rotation and Scaling

The correlation responses of the rotation, scaling and rotation/scaling are shown in Table 4. In rotation, the image is rotated by ±2°, ±1°, ±0.75°, ±0.5° and ±0.25°. Some larger degrees such as 5°, 10° and 15° are also tested. The correlation response remains high when a rotation of a smaller angle is applied. The correlation response drops a bit for the rotation of a larger angle since some portion of the image is cropped to preserve the central part of the image. For scaling attacks, the ratios with 0.5, 0.75, 0.9, 1.1, 1.5 and 2.0 are tested. When the image is enlarged, the correlation response is close to the optimal value since we will not lose much image information when we recover the image by down-sizing. However, the watermark is not detected when the image is scaled to one half of its original size. The reason

Rotation		Rotation/Scaling		Scaling	
Angles	Response	Angles	Response	Ratio	Response
−2.00°	13.15	−2.00°	15.07	0.5	-0.13
−1.00°	17.53	−1.00°	18.11	0.75	9.72
−0.75°	14.32	−0.75°	13.58	0.9	13.20
−0.50°	14.71	−0.50°	16.80	1.1	18.95
−0.25°	15.72	−0.25°	12.09	1.5	19.58
0.25°	16.51	0.25°	12.38	2.0	19.20
0.50°	13.91	0.50°	17.60	—	—
0.75°	15.72	0.75°	14.08	—	—
1.00°	14.99	1.00°	17.75	—	—
2.00°	15.11	2.00°	14.45	—	—
5.00°	10.61	5.00°	14.23	—	—
10.00°	13.54	10.00°	14.24	—	—
15.00°	9.05	15.00°	11.83	—	—

Table 4.2: Correlation responses of rotation, scaling and rotation/scaling attacks.

comes from the fact that the spatial-domain watermark is vulnerable to the JPEG compression when the image is down-sampled. If the JPEG compression is not applied, the watermark can still be detected without difficulty. The third attack is a combination of rotation, cropping and scaling. That is, the image is rotated, the center is cropped and scaled to its original size. All the correlation responses are higher than the threshold value 5.61, which shows the capability of the proposed system to survive generalized geometrical modifications.

5. Other Generalized Geometrical Transformations

Other generalized geometrical transformations such as shearing, change of the aspect ratio, columns/rows removal and linear transform are also tested in the experiments.

We show some typical examples of these geometrical attacks in Fig. 4.18. Correlation responses of Fig. 4.18(a), (b), (c) and (d) are 12.60, 18.32, 12.97 and 11.34, respectively. The watermark is unambiguously detected.

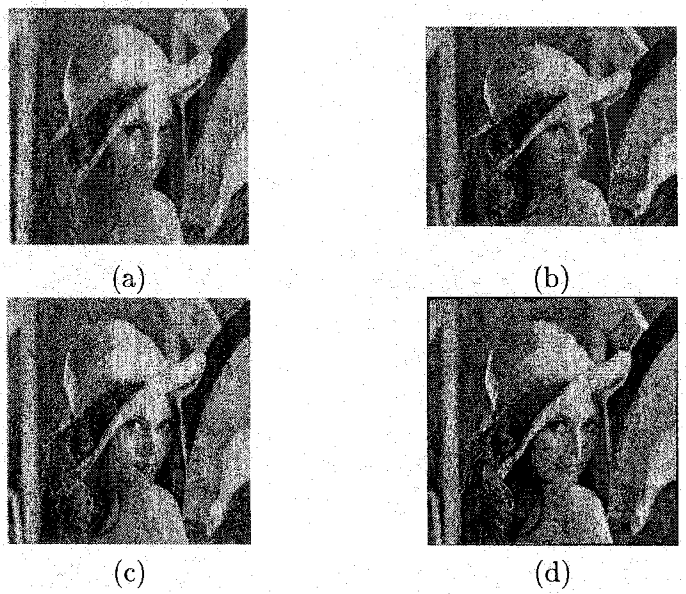
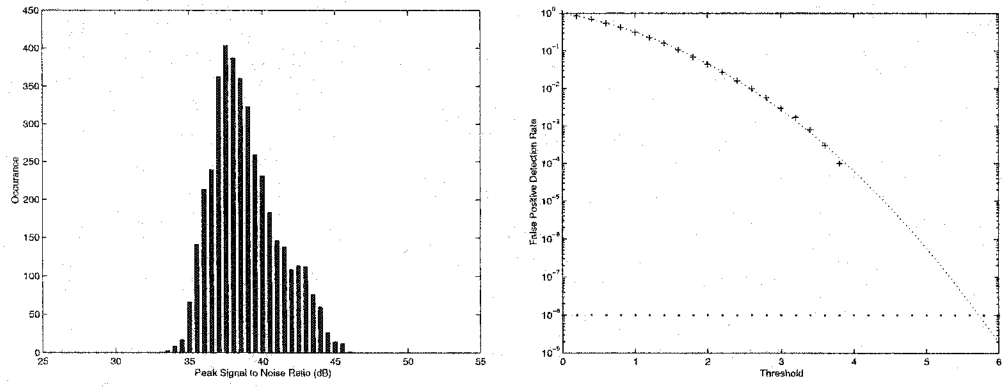


Figure 4.18: (a) Shearing the image by 5% in both horizontal and vertical directions, (b) the change of the aspect ratio by scaling the height by 0.8, (c) removing 17 rows and 5 columns, and (d) linear transformation with $a_{00} = 1.007$, $a_{01} = -0.01$, $a_{10} = -0.01$ and $a_{11} = 1.012$, where a_{00}, a_{01}, a_{10} and a_{11} are the four parameters of affine matrix as defined in (4.21).

Finally, we apply extensive tests to obtain the PSNR value of watermarked images and perform the false positive analysis on unwatermarked images. Fig. 4.19(a) shows the histogram of the PSNR value of 4000 watermarked images, which are provided by Corel Corporation. The average PSNR is 38.9dB, very close to the PSNR of the watermarked Lena image (38.3dB). We see that most of the watermarked images have a high PSNR value around the range of 37dB and 39dB. The minimum and maximum PSNR values are 33.5dB and 45.8dB, respectively, which correspond to a noisy image (which allows more watermark energy to be embedded) and a smooth image (which allows less watermark energy to maintain perceptual quality).

Fig. 4.19(b) shows the false positive rate analysis. We detect the watermark in 10,000 natural images, which are not embedded with watermarks. The X-axis is the threshold for watermark detection. If the absolute value of the correlation response calculated from (4.34) is larger than the threshold, the watermark is falsely detected. The Y-axis is the false positive rate. The '+' marks in Fig. 4.19(b) come from the experimental data, which are compared with the breaking line curve derived from the Q function. The dotted straight line indicates the false positive rate (10^{-8}) corresponding to the threshold 5.73 used in the experiment. We can see that the false positive rate is under our control and it is expected that the very low false positive rate (10^{-8}) can be achieved in the proposed composite watermarking scheme.



(a) (b)

Figure 4.19: Extensive tests for (a) the PSNR value and (b) the false positive rate analysis.

4.5 Perceptual Block-based DCT Watermarking Resisting Affine Attacks

As we mentioned earlier, the methodology of block-based watermarking is widely used because of its benefits of the increased perceptual performance, larger payload and possible compatibility with the current image compression standard, JPEG. However, the vulnerability of the block-based scheme against affine attacks severely limits its scope of applications. In this section, we attempt to solve the synchronization problem of general block-based watermarking schemes by using grid embedding/detection. We illustrate the usage of grid signals by a perceptual, affine-invariant, block-based DCT watermarking scheme.

Before describing the proposed approach, we would like to address the issue why we will present one more scheme in another domain. We first discuss transform-based watermarking briefly. The Fourier transform, the block-based discrete cosine transform and the wavelet transform are the three most popular ones used in transform-based or frequency-domain watermarking. We compare advantages and disadvantages of using these transforms as follows.

The Fourier transform outperforms the other two in terms of the functionality. It is much easier to attain a digital watermark surviving geometrical attacks by using the Fourer transform because of its invariant properties. Our proposed composite scheme is a good example as we chose the Fourier domain for watermark embedding/detection due to its shift invariance. However, the drawback of watermarking algorithms based on the Fourier transform is the lack of perceptual measurement in a pure frequency domain. In our spatial-frequency composite watermarking scheme, the scaled magnitude of the Fourier coefficient is used as the watermark weighting factor and a spatial masking is applied afterwards to ensure that the embedded signals fulfill the visibility constraint. Without this postprocessing step, the artifacts caused by the Fourier-domain watermark may appear, especially in some flat regions. The requirement of postprocessing in the spatial-domain may not only increase the complexity of the system but affect the embedded Fourier-domain watermark to a certain degree. Therefore, the major concern in Fourer-based watermarking is the visibility of embedded watermarks.

Since the wavelet transform and the block-based DCT have good capability of energy compaction, many image codecs, including JPEG and JPEG-2000, resort to either one of the two transforms for efficient coding. Many watermarking schemes also make use of them to increase robustness against filtering and compression attacks. Certain efficiency can thus be attained by embedding the digital watermark into the same domain as compression, as what we have done in the JPEG-2000 watermarking scheme. Moreover, many perceptual models have been developed based on block-based DCT or the wavelet transform for efficient quantization [2, 58, 91, 92]. Watermarking schemes utilizing the two transforms can achieve better watermark adaptation by using these existing visual models as well. More sophisticated visual models can help maintain a good balance between the requirements of robustness and invisibility.

Nevertheless, geometrical correction is always a major problem for these transforms. For block-based DCT watermarking, the ability to recover the block position accurately is the key to correct watermark detection. This is however a difficult task. The watermarking scheme based on the wavelet transform has a similar drawback as the transform is neither shift nor rotation invariant. Most of previous research either ignored this problem or assumed that the original image is available, which severely limits the scope of applications.

Our goal is to develop a perceptual watermarking scheme resilient to geometrical attacks. To develop a perceptual watermarking, the Fourier-based approach may not be appropriate and the wavelet-based and block-based watermarking are better choices. To make the watermark resist geometrical attacks, we exclude the wavelet transform since cropping will make wavelet coefficients totally different from their original values and such attacks as rotation or scaling make the synchronization problem even worse. Therefore, we propose a scheme based on block-based DCT to demonstrate the idea of grid embedding to compensate the loss of synchronization in general block-based watermarking schemes as well as show the decent performance along with possible compatibility with JPEG.

We will first discuss Watson's visual model [91] in Section 4.5.1, which determines the Just Noticeable Difference (JND) of DCT coefficients and will be used to scale the embedded watermark energy to achieve perceptual watermarking. In Section 4.5.2, we demonstrate the processes of signal embedding and detection. We apply Watson's perceptual model in both the watermark and the grid signal embedding so that the modification of the DCT coefficient is limited by the JND to fulfill the invisibility constraint. Experimental results in Section 4.5.5 will be used to show the robustness of the proposed method.

4.5.1 Watson's Perceptual Model

Watson [91] proposed a perceptual model in block DCT domain to achieve an image-adaptive quantization table. The goal is to quantize the image as much as possible while maintain the quality of the image. It is worthwhile to note that quantization and watermarking share some similarity. Quantization modifies the data to obtain a more compacted representation while watermark changes the data to carry hidden information. Both modifications require that the introduced distortion be not perceived by the human eyes. Therefore, it is reasonable to make use of the model in watermarking to achieve imperceptible watermark embedding and to obtain a robust scheme by maximizing the embedded watermark energy.

In Watson's perceptual model, the estimation of the visual threshold begins with an image independent perceptual threshold proposed by Ahumada and Peterson [2, 58]. The visual threshold $T_{i,j}$ associated with each DCT coefficient with frequency indices (i,j) is measured under various conditions and can accommodate varying display luminance, resolution and viewing distance, etc. To be more specific, the log of $T_{i,j}$ can be approximated by a parabola in log spatial frequency,

$$\log(T_{i,j}) = \log(\frac{T_{min}}{1 - 1.2(\frac{f_{i,0}f_{0,j}}{f_{i,0}^2+f_{0,j}^2})^2}) + K(\log(\sqrt{f_{i,0}^2 + f_{0,j}^2}) - \log(f_{min}))^2, \tag{4.35}$$

where T_{min}, K and f_{min} are functions of the total luminance. T_{min} is the luminance threshold at f_{min}, the frequency where the threshold is smallest and K determines the steepness of the parabola. $f_{i,0}$ and $f_{0,j}$ are vertical and horizontal spatial frequencies respectively. For a $N \times N$ block, they can be expressed as

$$f_{i,0} = \frac{i}{2Nw_i}, f_{0,j} = \frac{j}{2Nw_j}, \tag{4.36}$$

where w_i/w_j are vertical/horizontal width of a pixel in degrees of visual angle.

Two masking effects are taken into account: the luminance masking and the contrast masking effects. Luminance masking refers to the dependency of the visual threshold and the mean luminance of the local image region. The luminance-adjusted threshold $M^l_{i,j,k}$ is calculated by

$$M^l_{i,j,k} = T_{i,j}(\frac{c_{0,0,k}}{\bar{c}_{0,0}})^{a_T}, \tag{4.37}$$

where a_T controls the degree of luminance masking and a typical value of 0.65 is used. $\bar{c}_{0,0}$ is the average of DC coefficients for the image or a nominal value of 1024, corresponding to gray-level 8-bit images, and $c_{0,0,k}$ is the DC term of DCT for block k.

Contrast masking indicates that the threshold for a visual pattern would be reduced in the presence of other patterns, particularly those of similar spatial frequency and orientation. The luminance-adjusted threshold is then adjusted for the component contrast via

$$M^c_{i,j,k} = |c_{i,j,k}|^{s_{i,j}}(M^l_{i,j,k})^{1-s_{i,j}}, \tag{4.38}$$

where $c_{i,j,k}$ is the DCT coefficient and $s_{i,j}$ is the exponent controls the degree of contrast masking.

Since contrast-adjusted threshold aims to increase the visual threshold by considering the additional effect, the larger value between the contrast-adjusted threshold and luminance-adjusted threshold will be chosen as the final masked threshold,

$$M_{i,j,k} = Max[M^c_{i,j,k}, M^l_{i,j,k}], \tag{4.39}$$

The final masked threshold $M_{i,j,k}$ is also called Just Noticeable Difference (JND). The additive noise is assumed to be invisible if it is less than the JND. In this research, we will view Watson's model as the visibility constraint and any embedded signal should not violates this constraint to insure perceptual quality.

4.5.2 Structure of the Proposed Scheme

Our system block diagram is shown in Fig. 4.20. We embed a grid structure into the watermarked image to determine applied geometrical modifications without explicitly resorting to the original image. Detection of the grid structure includes the autocorrelation and cross-correlation processes to recover the image to its original scale, orientation and position. After the registration process as a result of grid detection, the hidden information can be successfully determined from the registered image by watermark detection.

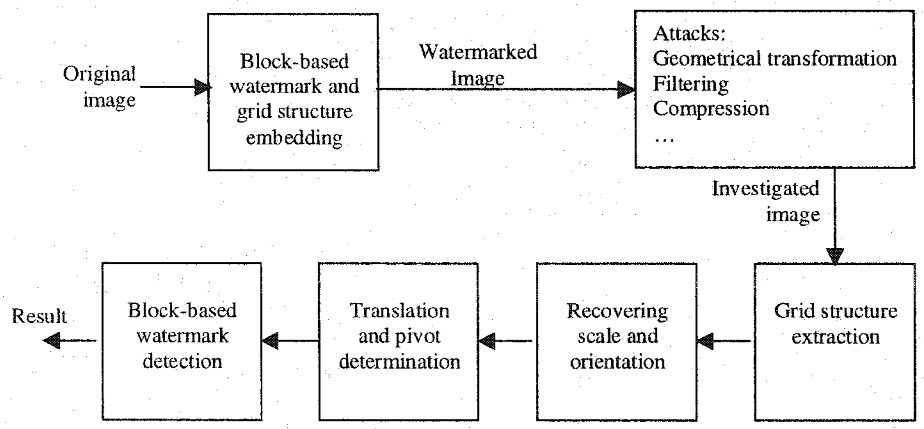

Original image
Block-based watermark and grid structure embedding
Watermarked Image
Attacks:
Geometrical transformation
Filtering
Compression
...
Investigated image
Grid structure extraction
Recovering scale and orientation
Translation and pivot determination
Block-based watermark detection
Result

Figure 4.20: The watermarking scheme with grid embedding and detection

4.5.3 Signal Embedding

The signal embedding processes are shown in Fig. 4.21. In the proposed block-based scheme, two signals are embedded in the image. One is the watermark signal and the other is the grid signal, corresponding to the frequency-domain watermark and the spatial-domain watermark, respectively, in the composite watermarking scheme presented in Section 4.4. We do not term the grid signal as the spatial-domain watermark here, since the grid signal will be shaped in the block DCT domain to be explained later.

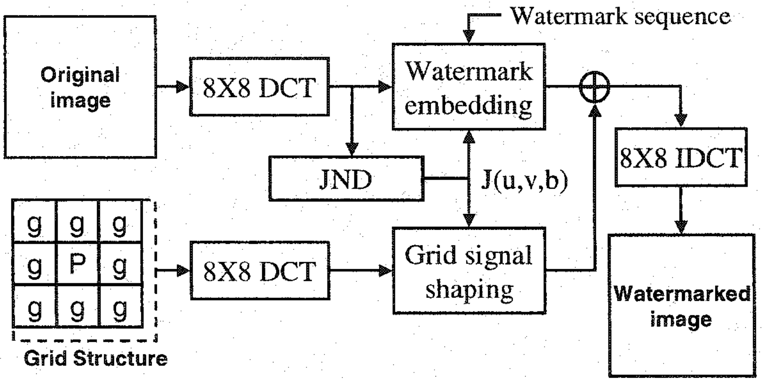


Figure 4.21: Signal embedding of the proposed block-based scheme.

In the composite watermarking scheme, spatial-domain watermarking is embedded in the image that has been embedded with the frequency-domain watermark. The detection of the frequency-domain watermark is more or less interfered by the existence of the other embedded signal. The situation is not so severe for Fourier-domain watermarks since the global Fourier transform is considered to be robust against noise adding. However, in the block-based watermarking scenario, we should be more cautious in doing so. Some block-based DCT watermarking schemes rely on more accurate detection, and grid signal embedding in the spatial domain without special attention may not be acceptable. To make grid embedding/detection applicable in general block-based DCT watermarking schemes, it is necessary that DCT coefficients selected for watermark embedding should not be modified further by the grid embedding process.

Therefore, we divide DCT coefficients into two parts: one part for carrying the watermark and the other part for hosting the grid signal. As done in JPEG compression, DCT is applied to 8×8 blocks to achieve signal decomposition. A classification scheme is shown in Fig. 4.22. The lower middle frequency coefficients at the shaded positions are selected for watermark embedding since they are comparatively reliable for watermark detection after possible filtering or compression attacks are applied. The grid signal is embedded in the remaining coefficients. It should be noted that some of the low frequency coefficients are also left for grid signal embedding so that the grid signal can have reasonable resistance against the compression attack. We can view the process of classification as dividing the communication channel into the "watermarking channel" and the "synchronization channel." The interference between the two channels is minimized by doing so.

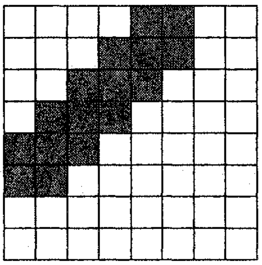


Figure 4.22: The watermark is embedded in the shaded positions while the grid signal exists only in the other positions.

Again, the watermark signal is a pseudo-random sequence, which should be kept confidential so that the security of the watermarking system can be further ensured. For selected coefficient $c_{i,j,k}$, the corresponding watermarked coefficient $c^{'}_{i,j,k}$ is formed by

$$c^{'}_{i,j,k} = c_{i,j,k} + M_{i,j,k} \times w_{i,j,k}, \tag{4.40}$$

where $M_{i,j,k}$ is JND determined in (4.39) to control watermark energy and $w_{i,j,k}$ is the corresponding watermark symbol taking 1 and -1 so that the amount of modification is equal to JND. The embedding method is very similar with the one proposed by Podilchuk *et al.* [62].

Special care must be taken to deal with the cropping attack. As mentioned before, cropping can cause a serious problem for block-based watermark detection. If the watermark sequence is embedded from the beginning of the image (left-upper corner) to the end of the image (right-bottom corner), removing a small portion of the image on the four sides will result in synchronization loss, and detection of the embedded watermark will fail. A simple solution would be repeatedly embedding a shorter watermark sequence into several parts of the image. However, this may limit the robustness and capacity of the watermark. Besides, some search algorithms are still needed to cope with the trivial cropping attack. A low-complexity method is thus required to make the watermark resist simple cropping attack. It should be noted that cropping is usually done on sides of an image. In a photograph, the central portion usually represents the Region of Interest (ROI), which is of more importance. It is less possible to remove ROI since the value of the photograph will be reduced by doing so. Therefore, the watermark should be embedded in the region where cropping is difficult to be applied. We therefore choose a point near the center of the image as the "pivot." The watermark sequence should be embedded starting from the pivot with an inside-out manner as shown in Fig. 4.23. If the position of the pivot can be accurately determined by both the watermark embedder and the watermark detector, multiple-bit watermarking can be achieved easily, and robustness and security can also be increased.

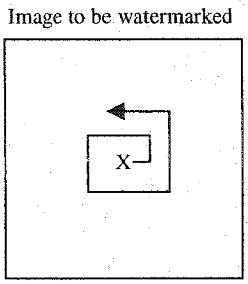


Figure 4.23: Embedding of the watermark sequence. "X" represents the pivot. The watermark sequence is embedded starting from the pivot with an inside-out manner.

Instead of using a simple Laplacian filter as an activity measurement, we utilize the JND value decided by Watson's visual model as our absolute visibility criterion for both watermark embedding and grid signal embedding in this block-based DCT watermarking scheme. In other words, the weighting of the grid signal will also be done in the block DCT domain. A practical solution is to multiply the grid signal by a value of S, which results in a strong grid signal. The exaggeratedly scaled grid signal will be shaped in the DCT domain to ensure its invisibility using Watson's model.

We calculate the 8 × 8 DCT of the grid signal. If the coefficient $c_{i,j,k}$ is allowed to be modified via grid embedding, *i.e.*, located in the region outside the shaded positions as shown in Fig. 4.22, the coefficient after grid embedding, $c'_{i,j,k}$, will be

$$c'_{i,j,k} = c_{i,j,k} + sign(g_{i,j,k}) \times Min(|g_{i,j,k}|, M_{i,j,k}), \tag{4.41}$$

where $g_{i,j,k}$ is the DCT coefficient of the grid signal. Thus, we can guarantee that the embedding of the grid signal will not introduce visible artifacts since the modification is within the JND.

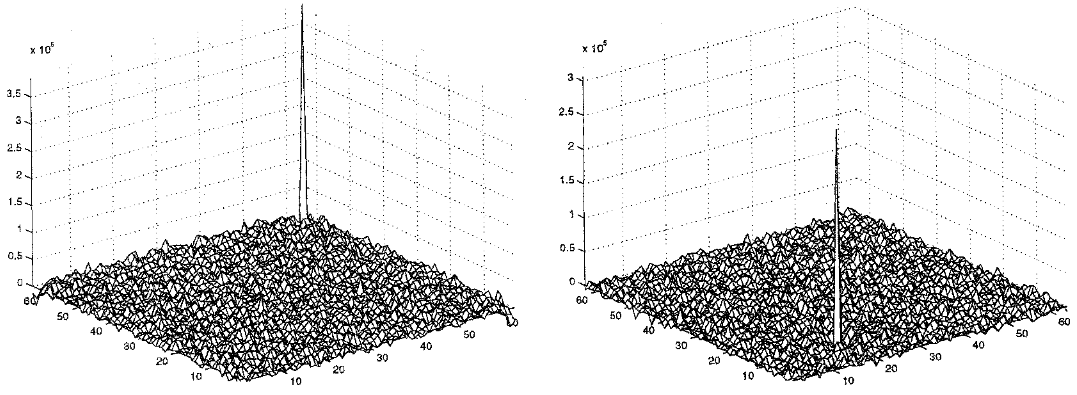
To help determine the pivot point, which is used to indicate the start of the watermark sequence, we intentionally choose a different pseudo-random pattern and embed it into the region covering the pivot as shown in Fig. 4.21. This pattern can be totally different from other patterns, or simply the negation of other patterns that construct the grid structure. Due to the different characteristic of this pattern, the detector can identify this region, and find out the pivot for subsequent detection of the watermark sequence.

4.5.4 Signal Detection

The signal detecting processes include the detection of the grid signal and the watermark signal. The autocorrelation and cross-correlation procedures are the two important tools for grid detection. In short, autocorrelation helps determine the affine matrix for recovering the image to its original orientation and scale while cross-correlation is used to measure the translational offset in a grid. The grid covering the pivot is determined by calculating the correlation once more. The whole self-registration process is then completed.

Now, we are ready to present the detail of the detection processes. We make use of the Wiener filtering method described in Section 4.4.3 to extract the grid signal because of its decent performance. The autocorrelation process is applied to the extracted grid signal in the same way as the composite scheme to determine the four parameters in the affine matrix, *i.e.*, a_{00}, a_{01}, a_{10} and a_{11}, in Equation (4.21) for the inverse affine transformation if necessary.

After the image is transferred back to its original orientation and scale by applying the inverse affine matrix, we have to determine the spatial translation. We do not measure x_s and y_s in Equation (4.21) explicitly since cropping can be done with any amount and the size of the original image is not supposed to be known by the watermark detector. But the horizontal/vertical shifts in a grid can be determined. That is, we can find out that the horizontal and vertical shifts via $\beta_x \times M + x_b$ and $\beta_y \times M + y_b$, respectively. Although the exact values of β_x and β_y are not known, we can at least verify the correct coordinate of the grid structure. To determine x_b and y_b, we fold up the extracted signal that is geometrically adjusted by the inverse affine matrix into a folded grid of size $M \times M$. That is, signal values are summed up if they are at the same position in the grid. We then calculate the cross-correlation of the folded grid and the embedded pattern, which is assumed to be known to the detector. Again, cross-correlation can be calculated efficiently by using DFT. Fig. 4.24 shows the cross-correlation of the folded grid and the grid pattern. Fig. 4.24(a) is the case of the watermarked image without cropping while Fig. 4.24(b) is the case of the cropped image. The horizontal and vertical shifts of the correlation peak reveal values of x_b and y_b.



(a)

(b)

Figure 4.24: The cross-correlation function of the folded grid and the embedded pattern in (a) the watermarked image without cropping and (b) the watermarked image with cropping.

With the help from the affine matrix and translational offset parameters, the grid structure of the investigated image can be matched with that in the unattacked watermarked image. The final step of the self-registration process is to determine the pivot. Identification of the pivot is also achieved by the correlation method. We calculate the correlation between each $M \times M$ grid with the pseudo-random pattern. As described earlier, the grid that contains the pivot is different from others so that the correlation value can clearly indicate which grid contains the pivot. For example, if the unique grid is embedded with a negative grid pattern, we will obtain a negative correlation response in this grid while correlation responses in other grids are much larger than zero. Synchronization is then fully recovered, and we can apply the block-based watermark detection to uncover the hidden information.

To determine the existence of a watermark, the normalized correlation response is as the confidence measurement, which is calculated via

$$\rho = \frac{\sum_k \sum_{(i,j)\in S} c^*_{i,j,k} \times w'_{i,j,k}}{\sqrt{\sum_k \sum_{(i,j)\in S} (c^*_{i,j,k})^2}}, \tag{4.42}$$

where S is the set of selected coefficients for watermarking in a DCT block, $c^*_{i,j,k}$ is the DCT coefficient of the investigated image and $w'_{i,j,k}$ is the test watermark sequence. Watermark decoding can also be achieved using the matched filter as coefficients or blocks are divided into sets, in which a watermark bit is embedded. The sign of the correlation response in each set represents the decoded bit.

A more sophisticated detection structure is to take the distribution of DCT coefficients into account. We assume that the DCT coefficients (excluding DC term) can be reasonably modeled as a generalized Gaussian distribution. The probability density function can be expressed as

$$f(c) = Ae^{-|\beta c|^\gamma}, \tag{4.43}$$

where

$$\beta = \frac{1}{\sigma}\left(\frac{\Gamma(3/\gamma)}{\Gamma(1/\gamma)}\right)^{1/2}, \tag{4.44}$$

$$A = \frac{\beta\gamma}{2\Gamma(1/\gamma)}, \tag{4.45}$$

and γ is the shape parameter and σ is the standard deviation.

If the distribution of DCT coefficients is modeled correctly, watermark detection may be improved by using the maximum likelihood detection. It should be noted that the matched filter is a special case of the maximum likelihood detection.

The selection of the shape parameter γ could bring certain impact on the performance of the watermark decoder. There are two methods to determine the value of γ. The first method is to estimate the shape parameter from a set of data based on the first- and second-order moments. Let ρ be

$$\rho = \frac{\sigma^2}{E^2[|c-\mu|]}, \tag{4.46}$$

where $E[\cdot]$ is the expectation and μ is the mean of the set of DCT coefficients. The shape parameter γ for the distribution of DCT coefficient can be found by solving

$$\frac{\Gamma(1/\gamma)\Gamma(3\gamma)}{\Gamma^2(2/\gamma)} = \rho. \tag{4.47}$$

Equation (4.47) can be solved by a lookup table that is generated by letting γ vary over the range of values that could possibly be expected for this parameter in small steps.

However, we found that the previous method could underestimate the shape parameter. The reason is that the maximum likelihood detection is based on the assumption of the generalized Gaussian distribution. The assumption may fail when the image is complex and many outliers with large values appear. The assumption could be valid if we ignore those large coefficients for watermark detection. However, in our watermark detection, we take the contrast masking into consideration as shown in Equation (4.38). It is clear that the masking is directly related with the value of the coefficient. That is, we embed more watermark energy in stronger coefficients, which is the interference of the watermark. Watermark hidden in those coefficients could be more reliable than others. Since coefficients of a larger value could not be possibly modeled, the matched filter or the correlation detector will be the best method to detect the existence of the watermark. In this case, γ should be set to 2 and the maximum likelihood detector is degenerated to the matched filter.

In this scenario, the parameter could be very difficult to predict to achieve the optimum result. Here comes the second solution. We select a small portion of coefficients as "probing coefficients" and embed them with a known watermark. These probing coefficients will be used to detect the very existence of the watermark by Equation (4.42) and compare it with the threshold determined by the Q function to control the false alarm rate. At the same time, the probing coefficient will help select the best shape parameter in the investigated image for best watermark decoding afterwards.

4.5.5 Experimental Results

In the experiment, we assume that one-bit information is embedded in the image. Multiple-bit watermarking can be achieved easily by block allocation when perfect synchronization is reached. Again, the size of the grid pattern M is set to 64. The threshold value is set to 5.61, which corresponds to a false positive rate of 10^{-8}. The Lena image of size 512×512 is used for test. The viewing distance for invisibility test is equal to 3.56 multiplied by the picture size, which results in 32 pixels per degree of visual angle. The PSNR value of the watermarked image with respect to the original image is 37 dB. The original and watermarked Lena images are shown in Fig. 4.25.

First of all, we would like to test the robustness of the watermarked image under cropping attacks. We randomly cropped a region with size 400×400 from the watermarked image and then compressed it by JPEG with a quality factor varying from 90 to 10. The

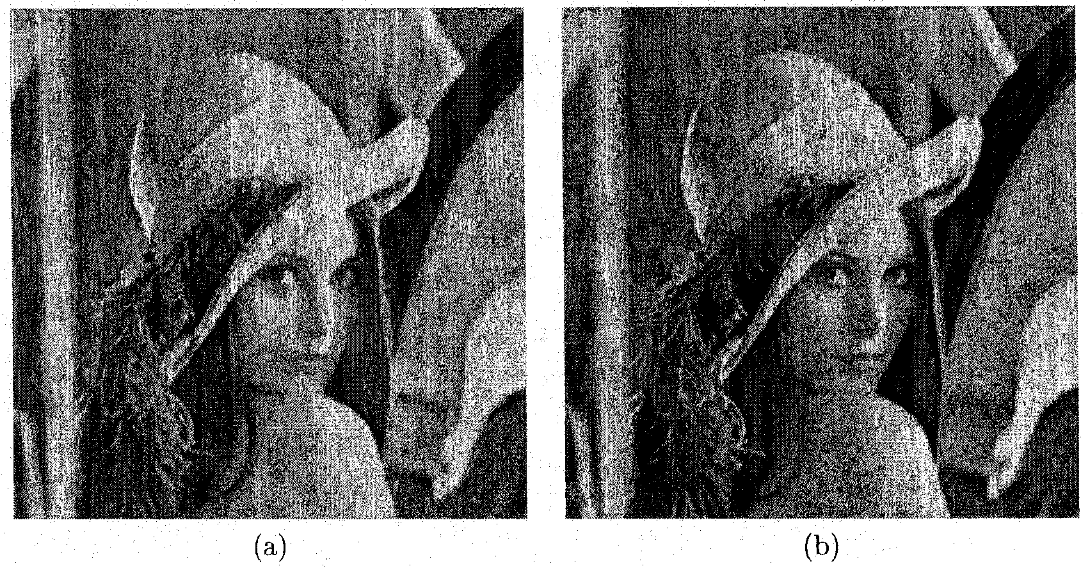


Figure 4.25: (a) The original Lena image and (b) the watermarked Lena image with a PSNR value of 37dB.

cropping operation results in non-zero values of translation, *i.e.*, $x_b \neq 0$ and $y_b \neq 0$. As shown in Fig. 4.26(a), all the correlation responses are above the threshold value 5.61, which is represented by a dotted line. The result demonstrates the resilience of the watermarking scheme against the combination of cropping and JPEG compression.

Next, we cropped the central part of the images by removing 2, 5, 10, 15, 20, 30, 40, 50, 60 and 75% of borders and compressing the resulting images by JPEG with a high quality factor (90) to see the effect of the pure cropping attack on the watermarked images. Results are shown in Fig. 4.26(b). Cropping causes information loss and the correlation response declines as the size of the remaining image becomes smaller. However, we see that correlation responses keep high when the image has a suitably large size and the value of the image is preserved. Here, it is assumed that the region that contains the pivot is not cropped since it lies in ROI of an image.

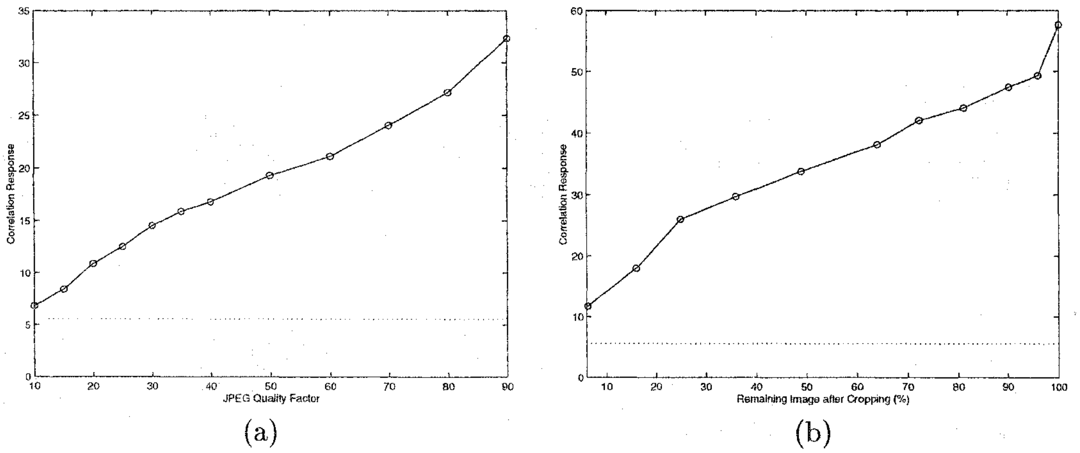
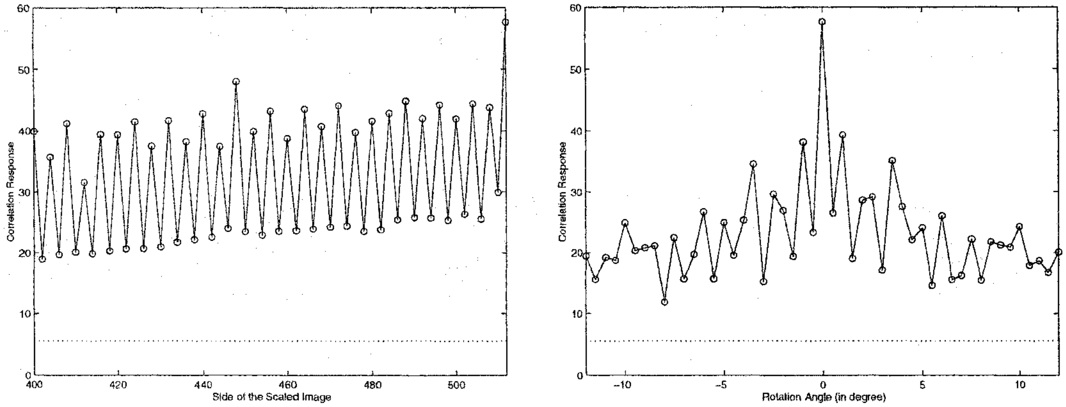


Figure 4.26: The robustness test of cropping.

One major concern is the precision issue. This is especially when the image is rotated and scaled since watermark detection usually requires very good precision, *i.e.*, perfect synchronization. Thus, we performed extensive tests on rotation and scaling to show the applicability of the proposed watermarking system. In scaling attacks, we scaled the image from 512×512 to 400×400 and changed the width and height of the image by 2 each time. The correlation responses are shown in Fig. 4.27(a). It is clear that the watermark survives well under scaling attacks and the lack of perfect precision does not cause a catastrophic result, but makes the correlation response fluctuate a bit. The information loss in down-sampling makes the correlation response a tendency to decline. In rotation attacks, we rotate the image every 0.5° from -12° to 12°. The result is shown in Fig. 4.27(b). By comparing these two figures, it appears that the system is more sensitive to rotation than scaling. Rotation does not introduce much information loss but synchronization loss resulting from the rotation process is more difficult to recover. Although the result shows that the watermark can still be detected, we believe that the performance can be improved by taking more situations into account to increase the precision of peak determination. This will however increase the complexity of the watermark detector as well.



(a) (b)

Figure 4.27: Robustness test of scaling and rotation.

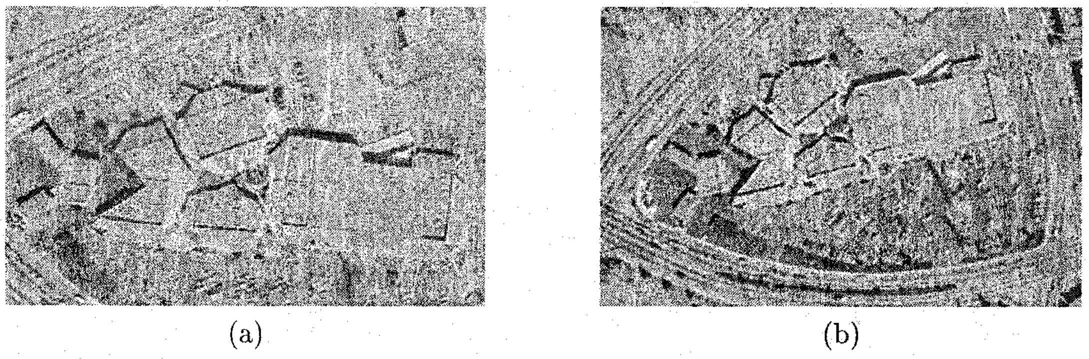
4.6 Comments on Grid Signal Embedding/Detection

4.6.1 Other Applications of Grid Embedding

The embedded grid signal can help recover the geometrically modified image back to its original scale and orientation for watermark detection. This geometrical correction functionality in digital images can also be of help to other applications. One possible usage of the grid signal is to assist image registration in database management. Given a picture of interest, we may need to search relevant images in the database so that we can dig out images with similar characteristics from the database or we can classify this picture and add it to the group of images with the same type for efficient archiving. In the remote sensing application, we may need to compare an airborne image with archived images to help us extract and understand useful information. Radar images can be used jointly with a map database for map updating, improved analysis and determining sensor platform positions.

Since the volume of images in an image database is usually very large, it is improper to involve the manual effort in the processes of comparison and matching. Those processes should be done in an automatic way through computing. The automated matching of an image to a database is achieved in an iterative process, where we start with an appropriate sensor position, search for corresponding points and compute refinements to the sensor position/attitude [47]. Correlation, edge-based matching, region-based matching and feature extraction, etc. are common tools for image registration.

However, images are usually taken at different time or varying situations such as different distances, directions, viewing angles, foci and light conditions, etc. The phenomenon is more manifest in airborne images. For example, the two aerial pictures as shown in Fig. 4.28(a) and (b) are taken from the same scene with a different direction and height. The trivial differences of scaling and rotation increase the computing burden of image registration since the comparison may have to be done in a different scaling, rotation and processing. It is clear that if the two images can be aligned with the same scale and orientation, the complexity of the matching process can be significantly reduced.



(a)

(b)

Figure 4.28: Aerial images taken from the same scene with different directions and heights.

A possible solution is delineated as follows. Before the image is taken, the direction and the height of the sensor are measured. Next, the grid signal is generated, resized and scaled according to the height and direction measured earlier. The size of the grid should be set corresponding to a predetermined height of the sensor. The grid should be pointed to a specific direction (north or south) and a rectangular grid may be more suitable in this scenario. After the picture is taken, the grid signal is weighted by taking the human perceptual model into account and embedded into the airborne picture. It should be noted that the energy of the grid signal might not be as strong as required in the watermarking application since we do not expect much malicious attack applied to the image. However, a reasonable degree of robustness is still needed to cope with possible transcoding and/or processing.

For an airborne image that will be matched with images in the database, the grid signal is first extracted and the autocorrelation function is calculated. After scaling and rotating the image by comparing the peaks of the autocorrelation function of the extracted grid signal with the known constellation, we can then apply the existing methods to extract features and match them with those of the image stored in the database. To sum up, with the objects aligned in the same direction and scale from the help of grid embedding/detection, the comparison and matching process can be done in a much easier manner.

4.6.2 Robustness against Stirmark Random Geometrical Distortion

The random geometrical distortion offered in Stirmark causes the most severe damage to digital image watermarking schemes. To further understand how Stirmark's random geometrical distortion affects the image, we apply the grid to the Lena image and compare it with its random distorted version as shown in Figs. 4.29(a) and (b). We can see that each block is slightly stretched, sheared, shifted, bent and rotated by a small and different random amount. As described earlier, since different operations are applied to different regions of the image, the geometrical manipulation cannot be described by a single affine matrix and the autocorrelation method fails to determine the grid structure for the inverse transformation.

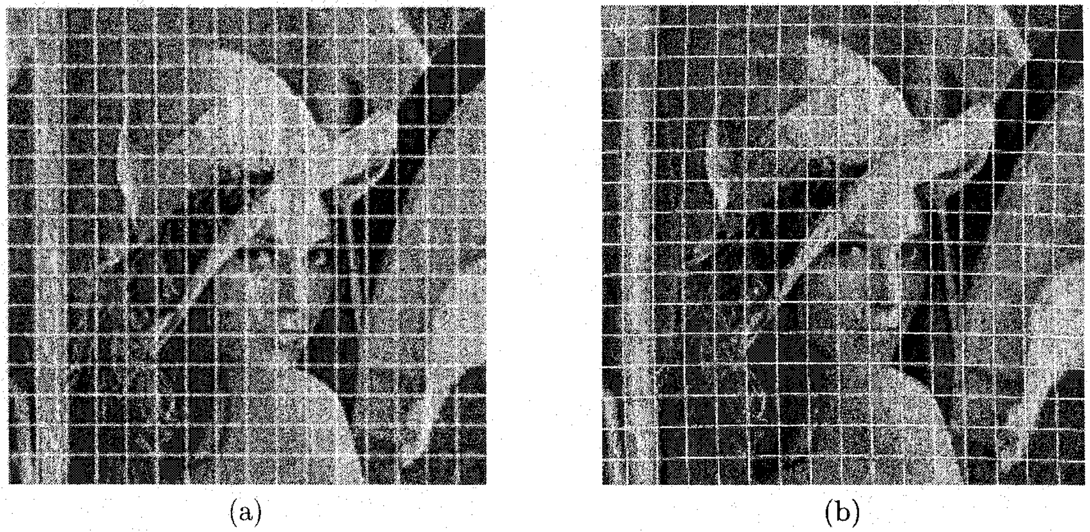

(a) (b)

Figure 4.29: (a) Lena with the grid (b) Lena with the grid after Stirmark random geometrical distortion.

However, with repetitive grid pattern embedding, we have a better chance to restore the image back to its original shape. The idea is to slightly distort the grid pattern in various ways and match it with the extracted signal so that the most possible local geometrical modification can be identified. The inverse operation can then be applied for reconstructing the local region. It is apparent that the method significantly increases the complexity of watermark detection and the false positive detection rate could be raised as well. Besides, the block size should be chosen carefully. If the block size is too large, we may not be able to reconstruct the image well and detection of the hidden watermark may still fail. However, if the block size is too small, the ability of the grid signal to resist generalized affine modifications may be decreased. The satisfactory reconstruction of the image with small grids is a problem to be solved too.

4.6.3 The Limit of Grid Embedding for Digital Watermarking

Although grid signals can help detect the geometrical transformation so that the distorted image can be recovered, periodic embedding of the grid pattern has a major drawback. For a 512 × 512 image with a 64 × 64 grid signal repeatedly embedded, there exist peaks appearing every 8 rows and columns in the Fourier spectrum of the image. This phenomenon hints that the grid signal is vulnerable to so-called template attacks [29], in which the peaks shown in the Fourier domain are removed and geometrical modifications are applied to the image afterwards. Besides, these peaks raise certain security concerns since they may reveal the existence of grid signals and the attacker may thus have enough knowledge about the repetitive grid pattern so that they can inverse the process to erase the embedded grid. After the grid signal is removed, the ability of combating affine attack is compromised. In the spatial-frequency composite watermarking scheme, a successful attack has to include geometrical attacks such as scaling or rotation to introduce the synchronization problem. In the perceptual block-based scheme, the detector may not be able to locate the starting point of the embedded watermark, *i.e.*, the pivot, since the pattern covering the pivot is removed. The subsequent watermark detection task will become more difficult.

In fact, this drawback is basically a problem of trade-off. The explicit peaks reveal the existence of the grid signal. However, if we try to hide these peaks to enhance the security of the watermarking scheme, we are making the hidden watermark more vulnerable to geometrical attacks as well. Without the original image at hand, the watermark detector and attackers may have a similar amount of knowledge about the hidden information when the image has been geometrically modified. One possible solution to segment the image by using feature points. These feature points should be detected reliably even after geometrical or filtering attacks. As the image is segmented into several parts, we separate these parts into two groups. For grids located within one group, the grid pattern is positively modulated. For grids in the other group, the grid pattern is negatively modulated. By doing so, we may break the periodic nature of the grid signals so that the Fourier spectrum will not show those additive peaks. The watermark detector will extract the embedded signal and segment the image in the same way as the watermark embedder does. Then, the detector negates the extracted grid signal in the group in which the grid pattern is negatively modulated in the embedder. Then we should be able to determine the possible affine distortion by the autocorrelation process described before. It is apparent that a reliable image segmentation algorithm is the key of this methodology. Besides, the watermark embedder and detector must have the same way to separate the segmented parts into positively modulated and negatively modulated groups efficiently. Otherwise, a further synchronization problem may arise due to this segmentation process.

4.7 Conclusion

Geometrical attacks cause a great deal of trouble to digital image watermarking. We tackled the synchronization problem resulting from geometrical modifications via structural grid signal embedding and detection to attain affine-invariant watermarking schemes. We first developed a spatial-frequency composite watermarking method, in which two watermarks are embedded into two domains separately in a sophisticated way to achieve the balance between imperceptibility and robustness. The embedded spatial-domain watermark, *i.e.*, the grid signal, is used to accurately recover an attacked image to its original orientation and scale. The frequency-domain watermark is responsible for carrying the watermark payload. Since the frequency-domain watermark is translation-resilient, the embedded information can be correctly determined from the possibly cropped or shifted version of the watermarked image. Next, we extended the idea of grid embedding/detection to the block-based watermarking, which is commonly adopted in many existing algorithms but extremely vulnerable to geometrical attacks. With the help from the grid signal and using the autocorrelation and cross-correlation, we can determine the affine matrix and translation to recover the modified image by inverse transformation. The block-based DCT watermark detector can then detect the watermark from the geometrically corrected image. Besides, it should be noted that grid embedding/detection should be helpful to many block-based watermarking schemes, which do not necessarily work in the DCT domain. Some comments on grid signal embedding/detection were given at the end to show its advantages and limitations.

Before ending this chapter, we would like to compare the two proposed affine-invariant schemes. Although both schemes achieve robustness against generalized geometrical attacks by adopting the similar methodology of structural grid signal embedding/detection, there exists a subtle difference because of different domains where the watermarking processes are operated. The composite scheme employs a global Fourer-transform for watermarking, which attains shift-invariance and better robustness with the cost of inferior visual performance. The block-based scheme makes use of a formal visual model to fulfill the constraint of invisibility. The interference between the watermark and the grid signal is smaller in the block-based scheme as well. However, compared with the composite scheme, the block-based scheme is less resilient to geometrical attacks since a more precise synchronization is required for separating the image into blocks. Besides, successful watermark detection relies on an accurate determination of the pivot point, which increases the difficulty of the self-registration process.

Chapter 5

Future Work and Conclusion

5.1 Future Work

Before concluding the dissertation, we would like to describe some possible future extensions of the current research.

5.1.1 System Refinement of the Proposed Algorithms

Some components in the proposed algorithms may need further improvements. We examine them in the following, and consider possible solutions and strategies to be adopted in the future.

1. Affine-invariant digital image watermarking

   One concern of the affine-invariant digital watermarking schemes is the robustness against the attack obtained by combining geometrical transformations and a higher ratio lossy compression. For the composite watermarking scheme, the hidden watermark cannot be detected if the image is down-sampled to less than one fourth of its original size and compressed with JPEG compression of a mild ratio. One possible solution is to improve the frequency-domain watermark using the log-polar mapping. It is expected that the resulting watermark can be detected if the image is purely rotated and scaled. This part can be implemented separately since the frequency-domain watermark should not severely contradict with the spatial-domain watermarking scheme. The drawbacks are that the computational load is increased and that the information loss during the mapping process has to be handled. Besides, some numerical numbers in the design of the proposed schemes have not yet been determined exactly. Although the experiments have demonstrated their decent performance, we believe that the result can be further improved if certain parameters can be assigned more accurately with more attacks and images being tested.

2. Steganography in JPEG-2000

   For the steganographic scheme developed under the framework of the JPEG-2000 standard, a more rigorous steganalysis may be necessary to ensure the security of the proposed system. However, the purpose of steganalysis is not to uncover the content of the secret message but to determine the very existence of the hidden information. Therefore, unlike the cryptanalysis, which can be done in a mathematical way, the steganalysis may only be carried out by careful examinations and repeated tests. It should be noted that the security of the steganographic scheme still relies heavily on techniques of cryptography as the hidden information is usually encrypted before embedded.

5.1.2 Development of Video Watermarking

Video watermarking techniques have a broader scope of applications than image watermarking. The main reason is that entertainment products such as TV news items, show programs, movies are worth a lot of value. Their wide distribution to the consumer market makes the content vulnerable to intellectual property right violation. Besides, video stored in the digital format may replace the videotape in the near future, since digital video presents an enormous improvement in video quality. Consequently, the video watermarking technique can be integrated with the encryption technology as a copy control tool to deter illegal reproduction. Although many video watermarking schemes have been proposed, there still leave some room for improvement.

At the first thought, there is not much difference between video and image watermarking since we can view video as an image sequence so that watermark embedding and detection can be performed on a frame-by-frame basis. However, some specific issues of video watermarking have to be taken care of.

1. Complexity

   One of the major concerns of video watermarking is its complexity. It is usually required that the watermark can be detected in real time. If the watermark is not detected from the investigated video, the detector has to keep detecting the watermark without affecting the performance of video playback or recording. Once an unmatched watermark is detected, the video player or recorder will stop its operation since the investigated video may be an illegal copy. Thus, the watermark detector should have a moderate complexity so that the detection can be done efficiently.

Besides, the amount of watermark payload is directly related with the complexity of implementation. Ideally, we can increase the watermark payload by testing more orthogonal watermark patterns in a video clip of interest. However, the time of the matching process will increase accordingly. The adequate amount of watermark detection has to be justified by empirical results.

2. Temporal Artifact

   In terms of visibility, one of the most distinct differences between image and video watermarking schemes is its temporal artifact. By viewing only one frame, the watermark may not look obtrusive to the human being. However, their visual appearance in moving pictures can be quite annoying. For example, it has been found that if the watermark is embedded into the wavelet coefficients of each frame, the ringing artifact may become more visible. Therefore, special attention has to be paid to the temporal artifact in video watermarking.

3. False Positive Detection

   The volume of video data is quite large. This large volume of data adds more flexibility to watermarking. For example, the watermark can be embedded frame by frame or only in some frames selected by a secret key, etc. However, the probability of false positive detection will also increase if the design and analysis of watermark detection is not done carefully. As discussed earlier, false positive detection will cause a lot of inconvenience to legitimate users so that the false positive analysis should be more precise in video watermarking.

4. Video Watermark Attack

   Some video watermarking attacks such as frame dropping/swapping or rate changing must be considered. These attacks can be viewed as geometrical modifications in the temporal domain. Besides, robustness against the higher-ratio compression is also an important issue. Increased robustness of the system against scaling and the change of the aspect ratio is still necessary since these processes are quite common in video editing or playing. However, the mechanism designed to resist these attacks may increase the computational complexity as well.

5.1.3 Universal Watermark Detector

Many watermarking schemes have been proposed and each of them may have various and unique characteristics. An interesting issue is to develop a universal watermark detector, which is used to determine the existence of any hidden signal in an image of interest without much information about the hidden data. One of the applications of this research is to approximately estimate the amount of watermarked images that are circulated in the network. This measurement may help to evaluate the potential of watermarking research and the demand of copyright protection of digital images more accurately. The other application is for the national security purpose. A universal watermark detector may help to screen out suspected images, in which certain groups of people may embed secret information for covert communication.

As discussed before, different watermarking schemes may adopt varying methodologies, such as the spread spectrum or quantization, or choose different transforms. All of these make developing a universal watermark detector a very challenging task. However, if we focus on detecting the existence of a robust watermark that can resist generalized geometrical attacks or a large volume of secret data transmitted by steganographic schemes, such a universal watermark detector may exist.

For example, as many schemes make use of periodical signals or embed artificial peaks in the Fourier domain to resist generalized geometrical attacks, we may examine these signals in the Fourier domain to possibly detect the existence of a hidden watermark used for copyright protection. Besides, embedding a robust watermark or a large amount of data may unusually modify the statistics of the host signal. Therefore, there may exist certain analytical or statistical ways to differentiate clean images and images embedded with certain hidden information.

5.2 Conclusion

In this dissertation, we explored the field of information hiding in digital images, an innovative idea of invisibly embedding additional information into digital images for several interesting applications. We focused on two important issues, *i.e.*, digital watermarking/covert communication in the upcoming image standard, JPEG-2000, and the robustness of a digital watermark against geometrical attacks. We provided practical solutions to these challenging problems and discussed the feasibility of the proposed schemes via theoretical analysis and experimental support. In the pioneering research of JPEG-2000 information-hiding, a robust watermarking scheme was developed for the copyright protection purpose. The watermarking and the compression procedures are combined in a sophisticated way to achieve efficiency and features such as progressive watermark detection and ROI watermarking. A steganographic scheme was then designed to reliably and secretly transmit high-volume information in JPEG-2000 compressed images for covert communication. In the development of digital watermarking schemes towards affine-invariance, we proposed to embed structural grid signals to solve the synchronization problem in blind watermark detection. By using the methodology of grid signal embedding/detection, we presented two affine-invariant schemes and their superior performances were demonstrated.

Building a comprehensive multimedia infrastructure relies on a well-designed security framework. In our opinion, research on information hiding should be of tremendous help to construct a better multimedia system. The research of digital watermarking may provide a final defense line for protecting intellectual property rights. The study of steganography should urge researchers to pay extra attention to security-related issues. We hope that the advance of digital technologies brings convenience to our modern life and our research can help move toward this direction.

Reference List

[1] M. D. Adams, "The JPEG-2000 still image compression standard," Tech. Rep., ISO/IEC JTC 1/SC 29/WG 1, Sep. 2001.

[2] A. J. Ahumada and H. A. Peterson, "Luminance-model-based DCT quantization for color image compression," in *Proc. SPIE*, San Jose, CA, 1992, vol. 1666, pp. 365–374.

[3] R. Barnett and D. E. Pearson, "Frequency mode L.R. attack operator for digitally watermarked images," *Electronics Letters*, vol. 34, no. 19, pp. 1837–1839, Sep. 1998.

[4] M. Barni, F. Bartolini, V. Cappellini, A. Lippi, and A. Piva, "A DWT-based technique for spatio-frequency masking of digital signatures," in *Proc. SPIE International Conference Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 1999, vol. 3657, pp. 31–39.

[5] P. Bas, J.-M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. on Image Processing*, vol. 11, no. 9, pp. 1014–1028, Sep. 2002.

[6] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM System Journal*, vol. 35, no. 3, pp. 313–336, 1996.

[7] D. Benham, N. Memon, B.-L. Yeo, and M. Yeung, "Fast watermarking of DCT-based compressed images," in *Proc. International Conference Image Science, Systems and Technology*, Las Vegas, NV, June 1997.

[8] A. Bors and I. Pitas, "Image watermarking using DCT domain constraints," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Lausanne, Switzerland, Sep. 1996.

[9] C. Busch, W. Funk, and S. Wolthusen, "Digital watermarking: from concepts to real-time video applications," *IEEE Computer Graphics and Applications, Image Security*, vol. 19, no. 2, pp. 25–35, Jan. 1999.

[10] G. Caronni, "Assuring ownership rights for digital images," in *Proc. Reliable IT Systems, VIS*, Vieweg, Germany, 1995, pp. 251–263.

[11] B. Chen and G. W. Wornell, "Quantization index modulation methods for digital watermarking and information embedding of multimedia," *Journal of VLSI Signal Processing*, vol. 27, no. 1/2, pp. 7–33, Feb. 2001.

[12] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.

[13] I. J. Cox, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for images, audio and video," in *Proc. IEEE International Conference on Image Processing*, Lausanne, Switzerland, July 1996, pp. 243–246.

[14] I. J. Cox, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, 1997.

[15] V. Darmstaedter, J.-F. Delaigle, D. Nicholson, and B. Macq, "A block based watermarking technique for MPEG-2 signals: Optimization and validation on real digital TV distribution links," in *Proc. European Conference Multimedia Applications, Services and Techniques*, Berlin, Germany, May 1998, pp. 190–206.

[16] G. Depovere, T. Kalker, and J-P Linnartz, "Improved watermark detection reliability using filtering before correlation," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Chicago IL, Oct. 1998, pp. 430–434.

[17] J. Dittmann, M. Stabenau, and R. Steinmetz, "Robust MPEG video watermarking technologies," in *Proc. ACM Multimedia*, Bristol, U.K., Sep. 1998, pp. 71–80.

[18] D. J. Fleet and D. J. Heger, "Embedding invisible information in color images," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Santa Barbara, CA, Oct. 1997, pp. 532–535.

[19] J. Fridrich, "A new steganographic method for palette-based images," in *Proc. PICS 52nd Annual Conference*, Savannah, GA, April 1999.

[20] J. Fridrich and R. Du, "Secure steganographic methods for palette images," in *Proc. The 3rd Information Hiding Workshop*, New York, 2000.

[21] J. Fridrich, M. Goljan, and R. Du, "Distortion-free data embedding," in *4th Information Hiding Workshop*, Berlin, Germany, 1998, vol. 2137, pp. 27–41.

[22] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, California, Jan. 2001, vol. 3971, pp. 197–208.

[23] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding for all image formats," in *Proc. SPIE Photonics West, Electronic Imaging 2002*, San Jose, California, Jan. 2002, vol. 4675.

[24] M. S. Fu and O. C. Au, "Data hiding watermarking for halftone images," *IEEE Trans. on Image Processing*, vol. 11, no. 4, pp. 477–484, April 2002.

[25] B. Girod, "The information theoretical significance of spatial and temporal masking in video signals," in *Proc. SPIE International Conference on Human Vision, Visual Processing and Digital Display*, Los Angeles, CA, Jan. 1989.

[26] F. Hartung and B. Girod, "Digital watermarking of raw and compressed video," in *Proc. SPIE Digital Compression Technologies and Systems for Video Communication*, Los Angeles, CA, Oct. 1996, vol. 2952, pp. 205–213.

[27] J. R. Hernández, F. Pérez-Gonzalez, and J. M. Rodríguez, "Performance analysis of a 2-D multipulse amplitude modulation scheme for data hiding and watermarking still images," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 510–524, May 1998.

[28] J. R. Hernández, F. Pérez-Gonzalez, J. M. Rodríguez, and G. Nieto, "The impact of channel coding on the performance of spatial watermarking for coyright protection," in *Proc. IEEE International Conference Acoustics, Speech and Signal Processing 1998 (ICASSP 98)*, Seattle, WA, May 1998, vol. 5, pp. 2973–2976.

[29] A. Herrigel, S. Voloshynovskiy, and Y. Rytsar, "The watermark template attack," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 2001, pp. 4314–4346.

[30] M. Holliman, N. Memon, B.-L. Yeo, and M. Yeung, "Adaptive public watermarking of DCT-based compressed image," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 1998.

[31] C.-T. Hsu and J.-L. Wu, "Hidden signatures in images," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Lausanne, Switzerland, Sep. 1996, pp. 223–226.

[32] H. Inoue, A. Miyazaki, A. Yamamoto, and T. Katsura, "A digital watermark based on the wavelet transform and its robustness on image compression," in *Proc. IEEE International Conference Image Processing (ICIP)*, Chicago, IL, Oct. 1998, vol. 1, pp. 391–395.

[33] JPEG 2000 Document, "JPEG 2000 verification model 5.0 (technical description)," Tech. Rep., ISO/IEC JTC 1/SC 29/WG 1, July 1999.

[34] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "A video watermarking system for broadcast monitoring," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 1999, vol. 3657.

[35] K. T. Know and S. Wang, "Digital watermarks using stochastic screens - a halftoning watermark," in *Proc. SPIE International Conference Storage and Retrieval for Image and Video Databases*, San Jose, CA, Feb. 1997, pp. 310–316.

[36] E. Koch, J. Rindfrey, and J. Zhao, "Copyright protection for multimedia data," in *Proc. International Conference Digital Media and Electronic Publishing*, Leeds, U.K., Dec. 1994.

[37] E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," in *Proc. IEEE Workshop Non-Linear Signal and Image Processing*, Neos Marmaros, Thessaloniki, Greece, June 1995, pp. 452–455.

[38] D. Kundur and D. Hatzinakos, "A robust digital image watermarking method using wavelet-based fusion," in *Proc. IEEE International Conference Image Processing (ICIP)*, Santa Barbara, CA, Oct. 1997, vol. 1, pp. 544–547.

[39] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," *Proc. IEEE*, vol. 87, pp. 1167–1180, July 1999.

[40] M. Kutter, "Watermarking resisting to translation, rotation, and scaling," in *Proc. of SPIE Multimedia Systems and Applications*, Boston, MA, Nov. 1998, vol. 3528, pp. 423–431.

[41] M. Kutter, S. K. Bhattacharjee, and T. Ebrahimi, "Towards second generation watermarking schemes," in *Proceedings 6th International Conference on Image Processing (ICIP'99)*, Kobe, Japan, Oct. 1999, vol. 1, pp. 320–323.

[42] M. Kutter, F. Jordan, and F. Bossen, "Digital signature of color images using amplitude modulation," in *Proc. of SPIE Storage and Retrieval for Image and Video Databases*, San Jose, CA, Feb. 1997, vol. 3022, pp. 518–526.

[43] M. Kwan, "Gifshuffle," http://www.darkside.com.au/gifshuffle.

[44] G. Langelaar, J. C. A. ven der Lubbe, and R. Lagendijk, "Robust labeling method for copy protection of images," in *Proc. SPIE, Electronic Imaging*, San Jose, CA, Feb. 1997.

[45] G. C. Langelaar and R. L. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Trans. on Image Processing*, vol. 10, no. 1, pp. 148–158, Jan. 2001.

[46] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Watermarking by DCT coefficient removal: Statistical approach to optimal parameter settings," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 1999, vol. 3657.

[47] F. W. Leberl, *Radargrammetric Image Processing*, Artech House, Inc., 1990.

[48] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall, 1990.

[49] C.-Y. Lin, M. Wu, M. L. Miller, J. A. Bloom, I. J. Cox, and Y. M. Lui, "Rotation, scale, and translation resilient public watermarking for images," in *Proc. SPIE Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 2000, vol. 3971, pp. 90–98.

[50] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.

[51] R. Machado, "Ez stego," http://www.stego.com.

[52] W. Macy and M. Holliman, "Quality evaluation of watermarked video," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, 2000, vol. 4675.

[53] K. Matsui and K. Tanaka, "Video-steganography: How to secretly embed a signature in a picture," *J. Interactive Multimedia Association Intellectual Property Project*, vol. 1, no. 1, pp. 187–205, May 1994.

[54] J O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," in *Proc. IEEE International Conference Image Processing*, Santa Barbara, CA, Oct. 1997, pp. 536–539.

[55] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*, New York: Van Nostrand, 1993.

[56] S. Pereira, J. J. O'Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun, "Template based recovery of fourier-based watermarks using log-polar and log-log maps," in *Proc. IEEE Multimedia Systems 99, International Conference on Multimedia Computing and Systems*, Florence, Italy, June 1999, pp. 870–874.

[57] S. Pereira and T. Pun, "Fast robust template matching for affine resistant image watermarking," in *International Workshop on Information Hiding*, Dresden, Germany, 1999.

[58] H. A. Peterson, A. J. Ahumada, and A. B. Watson, "An improved detection model for DCT coefficient quantization," in *Proc. SPIE, Human Vision, Visual Processing, and Digital Display*, Bellingham, WA, 1993, vol. 1913, pp. 191–201.

[59] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Information Hiding, Second International Workshop*, Portland, Oregon, April 1998, pp. 219–239.

[60] I. Pitas, "A method for signature casting on digital images," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Lausanne, Switzerland, Sep. 1996.

[61] A. Piva, M. Barni, F. Bartolini, and V. Cappellini, "DCT-based watermark recovering without resorting to the uncorrupted original image," in *Proc. IEEE International Conference on Image Processing*, Santa Barbara, CA, July 1997, vol. 1, pp. 520–523.

[62] C. Podilchuk and W. Zeng, "Perceptual watermarking of still images," in *Proc. of Workshop Multimedia Signal Processing*, Princeton, NJ, June 1997.

[63] C. I. Podilchuk and W. Zeng, "Watermarking of the JPEG bit-stream," in *Proc. International Conference Image Science, Systems and Technology*, Las Vegas, NV, June 1997.

[64] C. I. Podilchuk and Wenjun Zeng, "Image-adaptive watermarking using visual models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.

[65] R. A. Roberts and C. T. Mullis, *Digital Signal Processing*, Addison Wesley, 1987.

[66] P. M. J. Rongen, M. J. J. J. B. Maes, and C. W. A. M. van Overveld, "Digital image watermarking by salient point modification," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 1999, vol. 3657.

[67] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[68] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[69] M.-Y. Shen and C.-C. J. Kuo, "Artifact reduction in low bit rate wavelet coding with robust nonlinear filtering," in *Proc. IEEE Second Workshop on Multimedia Signal Processing*, Redondo Beach, CA, Dec. 1998.

[70] K. Solanki, N. Jacobsen, S. Chandrasekaran, U. Madhow, and B. S. Manjunath, "High-volume data hiding in images: Introducing perceptual criteria into quantization based embedding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, Fl, May 2002.

[71] H. Stark and J. W. Woods, *Probability, Random Processes and Estimation Theory for Engineers*, Prentice Hall, 1994.

[72] P.-C. Su and C.-C. J. Kuo, "An efficient implementation of digital image watermark," in *International Symposium on Multimedia Information Processing (ISMIP99)*, Taipei, Taiwan, Dec. 1999.

[73] P.-C. Su and C.-C. J. Kuo, "An image watermarking scheme to resist generalized geometrical transformations," in *Proc. SPIE Photonics East*, Boston, MA, Nov. 2000.

[74] P.-C. Su and C.-C. J. Kuo, "Spatial-frequency composite watermarking for digital image copyright protection," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 2000.

[75] P.-C. Su and C.-C. J. Kuo, "Synchronized detection of the block-based watermark with invisible grid embedding," in *Proc. SPIE Photonics West*, San Jose, CA, Jan. 2001.

[76] P.-C. Su and C.-C. J. Kuo, "Information embedding in JPEG-2000 compressed images," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Bangkok, Thailand, May 2003.

[77] P.-C. Su, H.-J. Wang, and C.-C. J. Kuo, "Blind digital watermarking for cartoon and map images," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 1999, vol. 3657.

[78] P.-C. Su, H.-J. Wang, and C.-C. J. Kuo, "Digital image watermarking in regions of interest," in *Proc. PICS 52nd Annual Conference*, Savannah, GA, April 1999.

[79] P.-C. Su, H.-J. Wang, and C.-C. J. Kuo, "Digital watermarking on EBCOT compressed images," in *Proc. SPIE's 44th Annual Meeting*, Denver, CO, July 1999.

[80] P.-C. Su, H.-J. Wang, and C.-C. J. Kuo, "An integrated approach to image watermarking and JPEG-2000 compression," *Journal of VLSI Signal Processing*, vol. 27, no. 1/2, pp. 35–53, Feb. 2001.

[81] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Lausanne, Switzerland, Sep. 1996.

[82] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Multiresolution scene-based video watermarking using perceptual models," *IEEE Journal Selected Areas in Communications*, vol. 16, pp. 540–550, May 1998.

[83] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multi-level image," in *Proc. IEEE Military Communications Conference*, Monterey, CA, 1990, pp. 216–220.

[84] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. on Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.

[85] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 572–588, Sep. 1994.

[86] A. Z. Tirkel, G. A. Rankin, R. G. van Schyndel, W. J. Ho, N. Mee, and C. F. Osborne, "Electronic watermark," in *Digital Image Computing, Technology and Applications (DICTA'93)*, Sidney, Australia, 1993, pp. 666–673.

[87] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Austin, TX, 1994, vol. 1, pp. 86–90.

[88] G. Voyatzis and I. Pitas, "Applications of toral automorphisms in image watermarking," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Lausanne, Switzerland, Sep. 1996.

[89] H.-J. Wang, P.-C. Su, and C.-C. J. Kuo, "Wavelet based blind watermark retrieval technique," in *Proc. SPIE Photonics East - Symposium on Voice, Video, and Data Communications*, Boston, MA, Nov. 1998, vol. 3528.

[90] H.-J. Wang, P.-C. Su, and C.-C. J. Kuo, "Wavelet-based digital image watermarking," *Journal of Optics Express*, vol. 3, no. 12, pp. 491–496, Dec. 1998.

[91] A. B. Watson, "DCT quantization matrices visually optimized for individual images," in *Proc. SPIE, Human Vision, Visual Processing, and Digital Display*, Bellingham, WA, 1993, vol. 1913, pp. 202–216.

[92] A. B. Watson, G. Y. Yang, A. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise," *IEEE Trans. on Image Processing*, vol. 6, no. 8, Aug. 1997.

[93] R. B. Wolfgang and E. J. Delp, "A watermark for digital images," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Lausanne, Switzerland, July 1996, pp. 219–222.

[94] R. B. Wolfgang and E. J. Delp, "Fragile watermarking using the VW2D watermark," in *Proc. SPIE International Conference Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 1999, vol. 3657, pp. 204–213.

[95] P. Wong, "A public key watermark for image verification and authentication," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Chicago, IL, Oct. 1998, vol. 1, pp. 455–459.

[96] M. Wu, M. L. Miller, J. A. Bloom, and I. J. Cox, "A rotation, scale, and translation resilient public watermark," in *Proc. IEEE International Conference Acoustics, Speech, and Signal Processing*, Phoenix, AZ, March 1999.

[97] X. G. Xia, C. G. Boncelet, and G. R. Arce, "A multiresolution watermark for digital images," in *Proc. IEEE International Conference on Image Processing*, Santa Barbara, CA, July 1997, vol. 1.

[98] M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Santa Barbara, CA, July 1997, pp. 680–683.

[99] W. Zhu, Z. Xiong, and Y.-Q. Zhang, "Multiresolution watermarking for images and video: A unified approach," in *Proc. IEEE International Conference Image Processing (ICIP)*, Chicago, IL, Oct. 1998, vol. 1, pp. 465–468.