DIGITAL WATERMARKING FOR IMAGE CONTENT AUTHENTICATION

Vom Fachbereich Informatik der Technischen Universität Darmstadt genehmigte

Dissertation

Zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.)

Von

Huajian Liu

(Master of Engineering)

geboren in Shandong, China

Referent: Prof. Dr. habil. Claudia Eckert

Koreferent: Prof. Dr. Stefan Katzenbeisser

Tag der Einreichung: 11.08.2008

Tag der mündlichen Prüfung: 24.09.2008

Darmstadt 2008

D17

Darmstädter Dissertation

To Mom

Abstract

Image content authentication is to verify the integrity of the images, i.e. to check if the image has undergone any tampering since it was created. Digital watermarking has become a promising technique for image content authentication because of its outstanding performance and capability of tampering detection. However, many challenges for watermarking techniques for image authentication still remain unsolved or need to be improved, such as tamper localization accuracy, image quality, security, synthetic image protection, and so on. In this thesis, we propose different solutions for the content authentication of natural image and synthetic images respectively, improving the tamper localization accuracy and the watermarked image quality. In addition, we develop new watermarking schemes with region of interest masking to tackle the problem of high image fidelity requirement in special applications.

First, we propose a watermarking technique for natural image authentication. By introducing the random permutation strategy in the wavelet domain, the proposed watermarking technique significantly improves the resolution of tampering detection with lower watermark payload. Due to less watermarks being embedded, the image quality is therefore improved. Furthermore, thanks to the random wavelet coefficient grouping, the scheme is intrinsically secure to local attacks. Also, scalable sensitivity of

tampering detection is enabled in the authentication process by presetting the noise filter size.

Second, we study the unique characteristics of synthetic images and develop a novel watermarking scheme for synthetic image authentication. The proposed watermarking algorithm is fully compatible with the characteristics of synthetic images. With less pixels modified, the authentication system can still achieve pixel-wise tamper localization resolution. Moreover, we propose a new embedding strategy, which enables the capability of recovering the altered image content of the authentication system. Hence, not only can the authenticator localize the tampered area but also it is able to recover the removed content and identify the forged parts.

In addition, in order to tackle the high image fidelity requirement in some special applications, we propose a framework for ROI-supporting watermarking systems, which can be applied to different watermark embedding schemes. Based on the framework, we present the non-ubiquitous watermarking schemes with ROI masking for natural images and synthetic images respectively. Unlike the common holistic watermarking schemes, the ROI-based schemes do not embed the watermark ubiquitously over the whole image but avoiding modifying the content of the specified important regions. Although there is no watermark embedded inside these regions, their integrity is still well protected as well as other image parts. The same tamper detection resolution is achieved both inside and outside the specified ROI regions.

Zusammenfassung

Die Authentifizierung von Bildinhalten verifiziert die Integrität von Bildern, in dem es beispielsweise überprüft, ob an einem Bild nach seiner Erstellung Verfälschungen durchgeführt wurden. Digitale Wasserzeichen sind inzwischen eine vielversprechende Technik zur Authentifizierung von Bildinhalten geworden. Sie bieten eine ausreichende Effizienz, um Veränderungen nachzuweisen. Es hat sich aber herausgestellt, dass einzelne Herausforderungen hinsichtlich der Bildauthentifizierung anhand von Wasserzeichen noch ungelöst sind oder verbessert werden müssen. Dazu gehören eine präzisere Lokalisierung der Veränderungen, die Bildqualität, die Sicherheit, sowie der Schutz von synthetischen Bildern. In dieser Arbeit werden verschiedene Lösungen für den Schutz des Inhaltes von natürlichen und synthetischen Bildern präsentiert, wobei die Lokalisierung der Veränderungen und auch die Qualität der Wasserzeichen verbessert werden. Zusätzlich entwickeln wir neue Wasserzeichenansätze mit der Maskierung von ROI (Region of Interest), um in speziellen Anwendungen das Problem der hohen Anforderungen an die Bildqualität zu lösen.

Als erstes wird ein Wasserzeichen zur Authentifizierung von natürlichen Bildern vorgestellt. Die angewandte Wasserzeichentechnik verbessert die Erkennung von Veränderungen im Bild durch eine zufällige Auswahl von Wavelet-Koeffizienten signifikant. Dabei wird nur eine geringe Wasserzeichenkapazität benötigt, wodurch die

Bildqualität verbessert werden kann. Aufgrund der zufälligen Auswahl der Wavelet-Koeffizienten ist das Schema sicherer gegenüber lokalen Angriffen. Das Erkennen von Bildmanipulationen ist durch Einstellungen der Filtergröße im Authentifizierungs-prozess zusätzlich skalierbar.

Weiterhin identifizieren wir einheitliche Charakteristiken von synthetischen Bildern und stellen ein neues Wasserzeichenverfahren für Bilder dieser Art vor. Das vorgestellte Wasserzeichenmodel ist komplett anwendbar für die Authentifizierung synthetischer Bilder. Selbst nach der Modifikation weniger Bildpixeln lokalisiert das Authentifizierungssystem die Modifikation pixelgenau. Wir präsentieren eine neue Einbettungsmethode, die eine Wiederherstellung des verfälschten Bildinhaltes ermöglicht. Das Authentifizierungssystem lokalisiert den veränderten Inhalt, stellt den ursprünglichen Inhalt wieder her und identifiziert die verfälschten Bildkomponenten.

Um in einigen Anwendungen die Anforderung der hohen Bildqualität zu bewältigen, stellen wir ein Framework für ROI unterstützende Wasserzeichensysteme vor, dass von diversen Wasserzeichenalgorithmen benutzt werden kann. Wir präsentieren ein Wasserzeichenschema mit ROI Maskierung, dass sowohl für natürliche als auch synthetische Bilder angewandt werden kann. Vor der Einbettung wird eine Vorauswahl von ROIs durchgeführt. Entgegen üblicher Wasserzeichen bettet das ROI-basierte Schema das Wasserzeichen nicht gleichmäßig in das komplette Bild ein. Es vermeidet eine Veränderung der vorausgewählten Regionen im Bild. Auch wenn kein Wasserzeichen in diese Regionen eingebettet wird, so ist doch deren Integrität genau so gut geschützt wie die der übrigen Teile des Bildes. Es wurden die gleichen Detektionsergebnisse innerhalb und außerhalb der ROIs erreicht.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Dr. Claudia Eckert, for her guidance and support during the development of this work. I would also like to thank my second advisor Prof. Dr. Stefan Katzenbeisser for his valuable comments and suggestions on this thesis. My thanks also go to the other members of my doctoral examination committee: Prof. Dr. Oskar von Stryk (the chair), Prof. Dr. Johannes Buchmann and Prof. Dr. Karsten Weihe. In addition, I would also like to thank Prof. Dr. Erich J. Neuhold and Prof. Dr. Thomas Hofmann for their guidance and encouragement in the early stage of my Ph.D work.

I am grateful to Dr. Martin Steinebach for his mentoring and constant support since the very beginning of my Ph.D work. His thoughtful comments and encouragement on my research and thesis are highly appreciated. I would also like to thank all other members of MERIT and TAD divisions of Fraunhofer IPSI and SIT institutes for their help and collaborations. Special thanks to Lucilla Croce Ferri, Sascha Zmudzinski, Enrico Hauer, Stefan Thiemert and Patrick Wolf for plenty of enlightening discussions and valuable help during my research work. In addition, I thank Dr. Zhou Wang, Enrico Hauer and Sascha Zmudzinski for their great help in all kinds of aspects of my life in Germany.

Acknowledgements

This work was accomplished with the financial and technical support of Fraunhofer IPSI and SIT institutes. During my work at IPSI and SIT, I have also got lots of help from the colleagues of other divisions besides MERIT and TAD. All these kind help and collaborations are deeply appreciated. In particular, I would like to express my appreciation to Barbara Lutes for her constant kind help during my work at IPSI.

Finally, my greatest thanks go to my Mom without whose love, nurturing and support I could never accomplish this thesis. I would also like to thank my sister and our whole family for their love and support during the course of my research work.

Contents

Chapter 1 Introduction

1.1 Motivation

With the rapid growth of multimedia systems and popularity of Internet, there has been a vast increase in the use and distribution of digital media data. Digital images become more and more popular in various applications. People can not only conveniently obtain and exchange digital images, but also can easily manipulate them [ZST04]. By using the powerful personal computer and image editing software, even an inexperienced user is able to edit a picture at will, such as adding, deleting or replacing specific objects. Some powerful software, like Adobe Photoshop, can even help a common amateur, who doesn't have any professional skills, to make 'perfect' manipulations without introducing any noticeable traces [CMB01]. Figure 1-1 shows an example of image manipulation, which was published by Spiegel Online in April 2005 [SO05]. In the Deutsch Bank annual report of 2004, an old photo of the board of management from the annual report of 2003 was reused after some manipulations. As can be seen in Figure 1-1 (b), one person on the right was removed and another person on the left was moved to the right side. The tampered image looks visually perfect and genuine. The "new" photo conveys to the viewer the information that the management board met again in 2004, but in fact they did not.

(a)

(b)

Figure 1-1 Example of image manipulation: (a) Photo of the board of management of Deutsch Bank in the annual report of 2003, (b) Photo of the board of management of Deutsch Bank in the annual report of 2004.

It is very hard, if not impossible, for a human to judge whether an image is authentic or not by perceptual inspection. As a result, the old proverb "Words are but wind, but seeing is believing." is not true any more in this digital era. Therefore, no visual data can be considered trustworthy before passing certain integrity authentication. A pressing security need is emerging to protect the visual data against illegal content tampering and manipulation [ZS03].

Visual data authentication is to verify whether the visual content has undergone any tampering since it has been created [CMB01]. It resembles the problem of message

authentication that has been well studied in [S95] yet has some unique features. The traditional cryptographic solution to message authentication is to make use of *digital signatures*. A digital signature scheme normally consists of two algorithms: one for signing which involves the user's secret or private key, and the other for verifying signatures which involves the user's public key [G04][MOV96].

However, cryptographic solutions, like digital signatures, are not well suited for visual data authentication due to their characteristics. First, digital signature provides only bitwise authentication. The targeted data must be identical to the original copy in order to be considered as authentic by the digital signature. Even one bit difference will render the whole content unauthentic. In the visual data applications, the perceptual content instead of its binary representation should be authenticated because the same visual content may have different yet equivalent representations. Due to the massive volume, visual data are usually stored and distributed in compressed ways. Such compression methods are often not lossless and will render the compressed data slightly different from the original copy. For example, many digital images on the Web are commonly stored in JPEG format, which compresses the image in a lossy way. In JPEG compression, the image data is quantized; some image data, to which the human eyes are not sensitive, is even discarded. Moreover, the data may also undergo other incidental distortions in the transmission, such as random bit errors and packet loss. Obviously, a digital signature can not survive these inevitable common processing of the data, while all of these distortions are acceptable in visual data applications because they are usually imperceptible and do not break the data's integrity. In other words, the conventional digital signature technique can not distinguish between the incidental distortions and intentional manipulations, also known as malicious manipulations, of the visual contents.

In addition, digital signatures can neither provide any localization information of the manipulations nor have any capability of recovering the original data. For visual content authentication, the capability of localizing the manipulations is a particularly desirable feature in most applications [ZS03]. Not only the content integrity needs to be

verified, but also the knowledge of the tampered positions is very useful. With the help of the localization information, other parts of the content can still remain trustworthy and useful when the original data is not available. Knowing the exact position where the manipulation occurs can also help to infer an adversary's motives in applications like forensic evidences. Recovering the original data from the tampered version is also a desirable feature, which helps to estimate the extent of the modifications and reveals how the original content looked like, although it is not always possible.

Moreover, digital signatures are stored separately and externally attached to the data to provide the authentication. The need of additional storage significantly decreases their compatibility and portability in the practical applications. Furthermore, this property also renders them easily to be removed either incidentally or intentionally [CMB01]. For example, given a JPEG image with a digital signature stored in the JPEG file header as metadata, when the JPEG image is converted to another format that has no space for the signature in its header, the stored signature will be lost.

Therefore, new security solutions are demanded for visual data authentication which should meet the special requirements of the corresponding applications. *Digital watermarking* is such a technique concerning multimedia security [CMB01]. Compared with cryptography, it is much better suited for visual data protection. Digital watermarking could be completely compatible with multimedia systems because it is not only transparent to both the viewer and the system but also able to survive the common media processing as well. Image content authentication is one of the application fields of digital watermarking. For image content authentication, it has many advantages over digital signatures. It can fulfill the above-mentioned requirements, not only being able to verify the integrity of multimedia content but also providing much more tampering information. Due to its outstanding advantages in multimedia data protection, digital watermarking has become a very active research field and been widely accepted as a very promising technique for multimedia security.

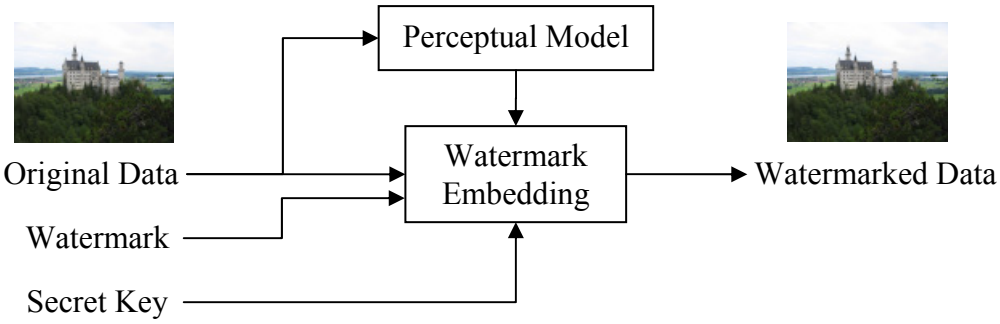


Figure 1-2 Watermark embedding process

1.2 Digital Watermarking

Digital watermarking is a technique that hides some additional information, which is called a 'watermark', into the 'cover data' by slightly modifying the data content. The term 'cover data', also known as host data, is used to describe the original media data, such as audio, image and video. After a watermark is embedded, the cover data becomes the 'watermarked data'. The process of inserting a watermark into the cover data is known as embedding, while the process of extracting or verifying the presence of a watermark is known as watermark detection or extraction.

Figure 1-2 and Figure 1-3 illustrate the general watermark embedding and detection processes respectively. Unlike the conventional visible paper watermarks, the watermarked data is perceptually identical to the cover data, i.e. the embedded watermark is invisible or inaudible. In order to insert the watermark in an imperceptible way, the watermark embedding process usually uses perceptual models to control the modification amount, known as watermark strength, adaptively in the different parts and components of the data. Although it is imperceptible to human observers, the embedded watermark is detectable by the watermark detector afterwards. At the watermark detector side, the original data is an optional input. If the detector requires the original data in order to extract the watermark, we call it 'private' watermarking; otherwise, it is known as 'public' or 'blind' watermarking. The latter is more feasible in

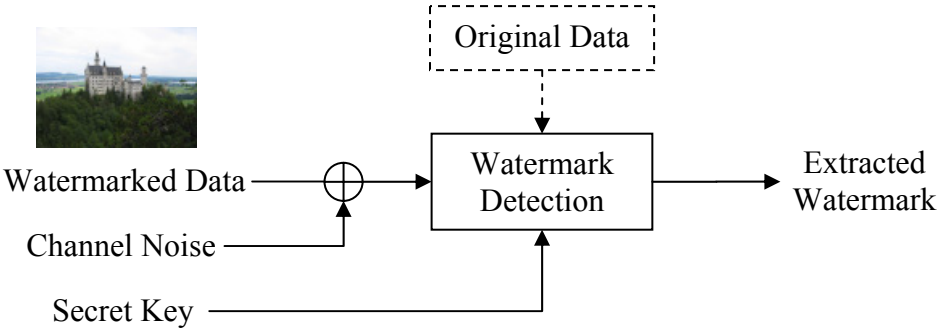


Figure 1-3 Watermark detection process

practical applications, because the original data is often not available at the detector side, especially in the authentication applications. A secret key is usually used to control the embedding and detection processes, which ensures the security of the whole watermarking system. Without the access to the secret key, the proper watermark can not be successfully embedded, detected or removed.

Generally speaking, *digital watermark* has many characteristics, out of which we selectively outline the most important ones below [CMB01][KP00][CS07].

Transparency: Imperceptibility is one of the most important characteristics of a digital watermark. All kinds of watermarks must satisfy this requirement. The embedded watermark must be transparent to the viewer and must not introduce any undesirable artifacts, which will cause quality degradation and even destroy the data's commercial value.

Robustness/Fragility: This property is highly application dependent. Depending on the intended application, the embedded watermark should be immune to or easily destroyed by the intentional content modifications or incidental distortions. Or it must satisfy both the requirements simultaneously, that is to say, the watermark can survive incidental distortions caused by the common signal processing, like channel noise, filtering, lossy compression, re-sampling, print and scan, etc., but will be easily impaired by the intentional manipulations of the media content.

Payload: The term "watermark payload" refers to the number of bits a watermark encodes within the cover data or within a unit of time [CMB01]. It is also known as watermark bit rate in the cases of audio and video watermarking. The required watermark payload varies greatly from application to application.

Portability: Unlike the digital signatures that are stored separately or appended as metadata in the file header, the embedded watermark is inseparable from the host data. It does not get removed after changing the data's store format or their digital/analog representations. This property makes digital watermarking particularly suitable for multimedia data protection. Compared to the traditional cryptography, the digital watermark always remains present in the multimedia data. It therefore can provide further protection to multimedia data after decryption, making the security intrinsically a part of the content.

Security: Different from the robustness, the watermark security refers to the ability to resist intentional or hostile attacks. It should be difficult for an adversary to remove or forge a watermark without the knowledge of the proper secret key even if the watermarking algorithm is publicly known. For robust watermarking, any attempts to remove or destroy a watermark should result in severe quality degradation of the host data before the watermark is lost or becomes undetectable. For the authentication watermarks, such attempts should destroy the data's authenticity.

According to different applications, digital watermarking has to comply with more specific requirements in the above-listed aspects. In general, the digital watermarking applications can be categorized into four main groups: copyright protection, fingerprinting, content authentication and annotation [CMB01][KP00]. These include both security-related and non-security applications. Every kind of application has different levels of robustness and security requirements. In the following, we briefly introduce these four kinds of different applications and the corresponding requirements.

***Copyright Protection*:** The embedded watermark is used to claim the ownership of the host data, which is typically an exclusive owner or producer's identifier. This kind of

application requires a very high level of robustness. The watermark is expected to remain in the protected data until the data's quality is degraded so severely that the content becomes commercially useless. The watermarking process is usually protected by a secret key that only the owner knows. Without the secret key, any unauthorized party cannot embed, detect or remove a valid watermark.

***Fingerprinting*:** Unlike the copyright protection applications in which the same watermark is embedded in all the copies, in fingerprinting applications a unique watermark is embedded in each individual copy of the host data in order to trace back illegal copies and find the piracy origin. The watermark is usually a customer or buyer's identifier instead of the owner's. High robustness against both incidental and intentional distortions is also required. In addition, because each copy contains a different watermark, the embedded watermarks must also be secure to collusion attacks.

***Content Authentication*:** The watermark information is embedded into the host data in a fragile way to monitor if the host is modified or not. Based on the application requirements, different levels of robustness are specified. According to the robustness levels, authentication watermarks fall into two classes: fragile watermark and semi-fragile watermark [CMB01]. Both kinds of authentication watermarks can identify intentional or malicious content manipulations. Fragile watermarks have the lowest robustness and are extremely sensitive to any signal sample value's change, while semi-fragile watermarks can survive moderate signal processing and hence are able to distinguish incidental distortions from malicious content manipulations. As discussed in the previous section, in content authentication applications, localizing the manipulations and recovering the original data are usually desirable capabilities. In order to identify the whole host data, high watermark payload is commonly required for authentication watermarks. Also high security level of preventing unauthorized embedding and detection must be ensured.

Annotation Watermark: Additional data-related information is embedded into the host data as content annotation. Thus, more information is conveyed together with the

transmission of the host data. The embedded information can be anything related to the content. For example, an image or a song could contain additional embedded information on its author, type, copyright or a link to a Web address where more related information can be retrieved. Annotation watermarks require a moderate robustness against the common signal processing and the lowest security level.

In this thesis, we will mainly focus on digital watermarking techniques for image content authentication. In the following chapter, we will discuss more specifically the framework and requirements of the fragile and semi-fragile watermarks. Following the overview of the related work and the existing challenges, we propose several novel watermarking schemes for content authentication of different types of images

1.3 Thesis Organization and Contributions

This thesis is organized as follows. In Chapter 2, we first introduce the basic framework of the watermarking technique for image authentication and then summarize the previous work and identify the existing challenges. From Chapter 3 to Chapter 5, we present the main contributions of this thesis, which are listed as follows.

In Chapter 3, we propose a semi-fragile watermarking scheme for natural image authentication. The proposed scheme reduces the necessary watermark payload by applying a random permutation process in the wavelet domain to build up a random mapping of all the image locations. With a lower watermark payload, the authenticator still achieves high tampering localization capability. The maximal resolution of tamper detection is not bounded by the unit size that is used to embed the watermark. Because the embedded watermark is distributed in the selected wavelet coefficients that are especially suitable for watermark embedding, namely, causing less perceptual artifacts, the quality of the watermarked image is improved. Furthermore, the random permutation process enhances the security of the whole system against local attacks. By embedding the watermark in different wavelet decomposition levels, the proposed scheme can achieve robustness against incidental distortions, such as JPEG

compression. Multi-level resolution authentication can be enabled by embedding the watermark in all the levels of the wavelet decomposition in order to identify different extents of the distortions.

In Chapter 4, we propose a novel watermarking scheme for synthetic image authentication. By identifying the challenges of embedding watermark in simple images, in the proposed algorithm every watermark bit is utilized to identify a group of pixels which are referred to each other in a random way. Thus, all pixels of the image instead of blocks are identified by much fewer watermark bits. The low watermark payload enables to impose more strict criteria on the selection of the embedding positions. Only the pixels whose change causes the least visible notification are used to embed the watermark. The watermark imperceptibility is therefore improved. Thanks to the random permutation and the statistical detection based on the density of unverified pixels, the proposed scheme can localize the tampered region with pixel-wise resolution. In the embedding process, we propose a new quantization strategy to improve the odd-even and look-up table embedding method by introducing a dummy quantization entry. Based on a statistical detection of the types of the unverified pixels, the proposed achieves the capability of recovering the original data in a binary way, which is enough for the most applications of synthetic images such as text images.

In Chapter 5, we identify the special requirements of some particular applications, which require extreme high image fidelity in important regions that are referred to as region of interest (ROI). In order to fulfill such requirements, we propose a framework for ROI-based watermarking. A non-ubiquitous watermark is applied to the targeted image for a ubiquitous integrity protection. Based on the proposed framework, we extend the watermark embedding schemes proposed in Chapter 3 and Chapter 4 to support the ROI concept. The watermark embedding only occurs outside the predefined or interactively selected regions of interest. The ROI(s) is kept intact during the watermark embedding process. Thus high fidelity in the preferred image parts is achieved, while the content integrity for the whole image is still protected. The image authenticator can localize the manipulations both inside and outside the ROI(s). One

important advantage of the proposed scheme is that no ROI information is required in the watermark detection and image authentication processes. Furthermore, the localization resolution of tampered areas remains equal inside and outside of the watermarked regions. The proposed framework can also be applied to other watermark embedding schemes to support ROI-based watermarking.

Finally, we conclude the thesis in Chapter 6 and discuss some possible directions for the future study.

Chapter 2 Preliminaries

In this chapter we first introduce a general framework of the watermarking techniques for image authentication to explain how authentication watermarks work. Then we specify the general requirements and features of an effective watermarking system of content authentication. Afterwards, a comprehensive review of the existing watermarking techniques for image authentication is presented. Two categories of authentication watermarks, fragile watermarks and semi-fragile watermarks, are discussed respectively, followed by an introduction of the watermarking techniques for synthetic images. Finally, we identify the existing problems and challenges with regard to the watermarking techniques for image content authentication.

2.1 General Framework of Authentication Watermark

Content authentication is one of the main application fields of digital watermarking. In contrast to other kinds of applications, like copyright protection, fingerprinting and annotation, the objective of content authentication is to verify the integrity of the test data and detect any possible manipulation.

Since authenticity is a relative concept, a reference is always needed to verify the targeted test data. For instance, in the traditional message authentication applications, digital signatures are usually used as references. The receiver reproduces a digest from

the received data and uses it to verify the appended signature that is generated from the original data. If they match, the received data will be deemed as authentic. Otherwise, they have been modified. In watermarking systems, the embedded watermark serves as a reference. As the watermark is embedded in the data, it will undergo the same transformations as the data itself. When the data is corrupted, the watermark will also be changed. Therefore, the integrity of the data can be verified by comparing the extracted watermark and the original one.

Figure 2-1 illustrates the general framework for a watermarking system for image content authentication, which basically consists of two parts: the watermark embedder and the watermark detector. In the watermarking systems for authentication, the watermark is used as an authentication code, which can be a random sequence, a visual binary logo or certain content-related features. This authentication code is embedded by the sender into the original image (also known as the cover or host image) in an imperceptible way. Although it is transparent to human observers, the code can be extracted by the watermark detector under certain circumstances. The detection conditions are determined by the system designation corresponding to the particular application requirement. For instance, what kinds of image distortions should be tolerable and should not impair the embedded watermark. At the receiver side, the extracted code is compared with the original to verify the integrity of the received image. If any mismatch occurs, it indicates that the image content has been manipulated. Therefore, in authentication applications, the watermark detector is also called the authenticator. To ensure the security of the whole system, a secret key is usually used in the authentication code generation, watermark embedding and retrieval processes. No knowledge of the secret key prevents the attacker from changing the embedded watermark or forging an image with a valid authentication code embedded.

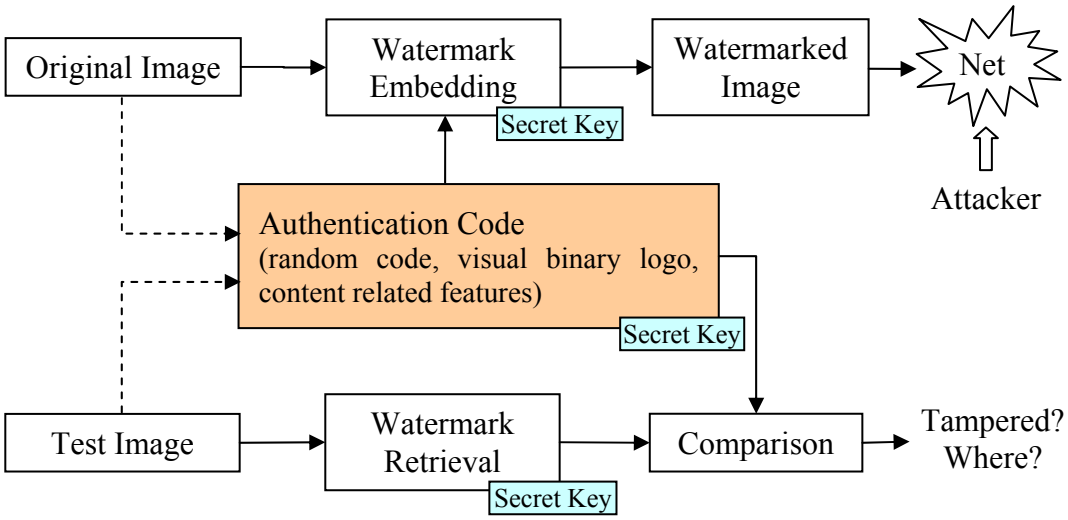


Figure 2-1 General framework of watermarking for image authentication

Basically, for an effective image watermarking system for the content authentication, the following features are usually desired.

1. Imperceptibility: the embedded watermark should be invisible under the normal viewing condition, i.e. high image fidelity must be maintained;
2. Capability of detecting whether an image has been maliciously tampered or not;
3. Capability of localizing the manipulations with good accuracy: the authenticator should be able to identify the locations of the manipulated regions with a desirable resolution and verify other regions as authentic;
4. Compatibility: the authentication system should be able to survive the incidental distortions caused by the common image processing to some extent, i.e. being able to distinguishing the incidental distortions from intentional/malicious tampering;
5. Portability: the authentication information should be embedded in the host image and no separate storage is required;

6. Security against forgery and unauthorized operations should be ensured.

Among the above listed features, portability is an intrinsic property of the digital watermark as we introduced in Chapter 1 since a watermark is always inserted into the host data itself. However, other several features are mutually competitive with each other. For example, the imperceptibility is determined by the embedding strength and the total watermark payload. Stronger embedding can make the watermark survive more distortions caused by the common image processing and higher watermark payload will usually render a better resolution of tamper localization. But stronger embedding and higher watermark payload will both degrade the image quality and cause the embedded watermark to be more visible. Therefore, a reasonable tradeoff should be found according to the application requirements in designing an effective watermarking system for content authentication.

2.2 State Of The Art

In the literature, a variety of watermarking techniques have been proposed for image content authentication. According to the types of the authentication they provide, the existing watermarking algorithms can be classified into two categories: watermarking for exact/hard authentication and watermarking for selective/soft authentication [CMB01][ZST04]. The watermarking algorithms for exact authentication are usually referred to as fragile watermarks, and the watermarking algorithms for selective authentication are known as semi-fragile watermarks.

2.2.1 Fragile Watermarks

Fragile watermarks provide a strict tamper detection, which has minimal tolerance of content manipulations. Even one single bit alteration will impair the embedded watermark and render the image inauthentic. Therefore, it resembles a digital signature in authentication function except that it does not need separate storage. Actually, many

fragile watermarking algorithms also make use of the cryptography techniques to achieve a high security level.

The simplest fragile watermarking algorithm is the so-called LSB watermark, in which the least significant bits (LSB) of pixels are modified to embed the desired watermark information [W98][YM97][FGB00][F02][CSST01]. Since the change of the least significant bit of pixel value is assumed to be imperceptible, the whole corresponding bit plane can be replaced by random or structured watermark patterns. In [W98], a public key LSB fragile watermark algorithm was proposed for image integrity verification. The image is divided into non-overlapping blocks and in each block the LSB plane is replaced by the XOR result of the watermark bitmap and the hash value of the image size and the pixel values in the block except the LSB. The XOR result is encrypted using the user's private key before embedding. In the detection process, the LSB plane is extracted from each block and decrypted using the corresponding public key. Then the embedded watermark bitmap in each block is recovered from the decrypted information by doing XOR operation again with the hash value recalculated from the same block. If the watermark bitmap is complete, the corresponding image block is deemed as authentic. Otherwise, the corrupted position indicates the location of the alterations. Because it separately authenticates the image blocks, this fragile watermarking algorithm was subsequently observed to be vulnerable to the vector quantization (VQ) attacks (also referred to as collage attack) [HM00]. Therefore, some improved algorithms were proposed in [CMTWY99] and [WM00]. In [CMTWY99], overlapping blocks are used in order to resist the VQ attack. This method, however, causes a significant loss of tampering localization capability. Therefore, Wong et al. proposed another improved scheme in [WM00], in which a unique image-dependent block ID is added into the hashing procedure to prevent the VQ attacks. This method preserves the tampering localization property of the original technique.

Another popular fragile watermark algorithm was proposed in [YM97], which is known as the Yeung-Mintzer scheme. This scheme uses a binary function (a look-up table), generated by a secret key, to enforce every pixel to map to the corresponding bit

value in a secret logo. Either a binary logo or a random pattern can be used in this method. An error diffusion process follows the watermark embedding to improve the watermarked image quality. Because every pixel is individually watermarked, the Y-M scheme can achieve pixel-wise tamper localization accuracy. The security of this algorithm was examined in [SMW99], followed by some simple modifications. It was reported that the search space for inferring the look-up table can be significantly reduced if the secret logo is known. In [FGM00] and [FGM02], it was further proven that even if the used logo image is kept secret it is still possible for the adversary to deduce the secret embedding function or successfully perform a VQ attack when multiple watermarked images with the same secret key are available. An improvement of the Y-M scheme was proposed in [FGB00]. The improved scheme introduces the neighborhood dependency in the mapping function to thwart the aforementioned attacks, although this modification decreases the tamper localization capacity. Nevertheless, in [WZLL04], Wu et al. further discussed that only a single authenticated image plus a verifier (oracle) is enough to successfully mount an oracle attack on the Y-M scheme and some of its variations. The proposed oracle attack does not need any knowledge of the used logo either.

In [F02], Fridrich presented an overall study of the security of fragile image authentication watermarks that have tamper localization capability. After investigating the possible attacks and the vulnerabilities of some existing schemes, the authors concluded that the inherent sequential character of the embedding in the pixel-wise watermarks was the reason that caused the security vulnerability against oracle attacks. Therefore, they turned their focus to block-based schemes and proposed a new block-based fragile watermark. The proposed scheme is a variation of the Wong scheme in [WM00]. In the new scheme, the authentication of the content and its origin are separate in order to identify the swapped blocks. A special symmetry structure is used to compose the binary logo that is used to authenticate each image block. The logo consists of the information about the image and the block origin, like image index, the block position, the camera serial number, etc. Although the proposed scheme is secure to all the known attacks that are addressed in the paper, such as VQ attacks and oracle

attacks, it reduces the tampering localization capability significantly as it is essentially block-based.

To thwart the VQ attacks, another fragile watermark was proposed in [CSST01], in which the watermark has a hierarchical structure. The image is divided into blocks in a multi-level hierarchy and the signatures for each block are inserted in the LSB plane. Signatures of the small blocks on the lowest level of the hierarchy ensure the accuracy of tamper localization and the higher level blocks signatures provide resistance to VQ attacks. This method achieves the superior localization property as a block-based scheme, but it is more complex than the Fridrich scheme.

To further improve the accuracy of tamper localization, recently a new statistical fragile watermarking scheme has been proposed in [ZW07]. In this scheme, the tailor-made authentication data consists of two parts. One part is a set of tailor-made authentication data calculated from the five most significant bits (MSB) of each pixel. The other part is a set of randomly generated test bits. The combination of these two parts replaces the three least significant bits (LSB) of each pixel to complete the embedding. In the authentication process, a statistical method is used to examine whether the five MSBs of each pixel are altered or not. This scheme can achieve a pixel-wise accuracy in locating the tampered pixels when the tampered area is not too extensive. However, it can not detect the alteration of the three least significant bits of each pixel. In addition, because the three LSB planes are completely replaced by the watermark, the quality of the watermarked image by this scheme is limited.

Besides the fragile watermarking algorithms in the spatial domain, some transform domain fragile watermarking schemes have also been proposed, for example, in the Discrete Cosine Transform (DCT) domain [WL98] or in the Discrete Wavelet Transform (DWT) domain [XA98][SL04]. The advantages of using the transform domains mainly lie in the following aspects. One of them is that the watermarking system can get more compatible with the popular image compression standards, e.g. JPEG. The embedding can be integrated into the compression process or completed directly in the compressed representation of the image. Another advantage is that the

perceptual distortion caused by the watermark can be better controlled in the frequency domain than in the spatial domain. Therefore, the watermarked image quality could be improved. In addition, since the frequency components are taken into account in the watermarking process, it becomes possible for the tamper detection to be localized in both spatial and frequency regions. Nevertheless, because the watermark is embedded in the frequency domain instead of by directly modifying the pixels, some slight pixel modification may not be detected by the transform domain watermarking algorithms. Moreover, the tamper localization accuracy is also bounded by the size of the image unit that is used to calculate the frequency components, for example, the block size used in the block-based DCT schemes. Subsequently, the sensitivity and accuracy of tamper detection are both decreased. Therefore, the transform domain methods are more often used in the design of the semi-fragile watermarking schemes that we will introduce in the next section.

2.2.2 Semi-fragile Watermarks

Since fragile watermarks are easily corrupted by any image processing procedure, the incidental distortion by the common image post-processing will also impair the watermark and render the image inauthentic. Obviously, it is very desired that the authenticator can distinguish incidental and malicious manipulations. To fulfill this requirement, semi-fragile watermarking techniques were proposed. In contrast to the exact/hard authentication by fragile watermarks, semi-fragile watermarks provide a selective/soft authentication. Semi-fragile watermarks monitor the image content instead of its digital representation. They allow slight or moderate modifications caused by common image processing like mild lossy JPEG compression, filtering and contrast enhancement, but will detect the malicious content-changing manipulations, like object addition, deletion and replacement. The extent of robustness of a semi-fragile watermark against incidental distortions is usually customizable according to the particular application requirement.

Semi-fragile watermarks are usually embedded in transform domains instead of the spatial domain in order to achieve moderate robustness, good imperceptibility and compatibility with compression standards. DCT and DWT domains are the most often used transform domains in semi-fragile watermarking. Since DCT and DWT are used in the popular image compression standards JPEG and JPEG2000, embedding techniques in DCT and DWT domains can be easily designed to be resistant to JPEG and JPEG2000 compression to some customized extent. Furthermore, the previous studies on human visual models in these domains can be directly reused in adaptively controlling the watermark embedding strength to improve the watermark imperceptibility. In addition, the spatial-frequency property of the wavelet transform enables good tamper localization capability in the authentication process.

Two embedding techniques are mainly used in the semi-fragile watermarking schemes. One is the spread spectrum method [F98a][LPD00], which was firstly proposed by Cox in [CKLS97]. The watermark message is first turned from a narrow band signal to a wide band signal and then embedded into the cover image additively or multiplicatively. The detection of a spread spectrum watermark is done by checking the correlation of the watermark signal and the watermarked image. Because a large amount of signal samples are necessary for good performance of the correlation detection, it is difficult for this embedding method to achieve a sufficient watermark payload in order to allow the tamper localization to fine scale. The other popular embedding method is the so-called quantization index modulation (QIM) method [CW01][LC00]. The watermark information is embedded by quantizing the selected frequency coefficients or some particular feature values to some pre-determined scales according to a look-up table or the simple odd-even mapping rule. By the QIM embedding, the embedding strength can be well controlled by the used quantization step, so that the watermark robustness can be customized quantitatively.

In the literature, a variety of semi-fragile watermarking algorithms have been proposed in the last decade. We only focus on reviewing some representative semi-fragile watermarking techniques in the following. In [F98a][F98b], Fridrich proposed a

technique in which the image is divided into medium-size blocks and in each block a spread spectrum watermark is embedded into the middle 30% of DCT coefficients additively. To verify the image integrity, the receiver tries to detect the embedded watermark in every block. If watermarks are detected in all the blocks with high detector responses, one can be fairly confident that the image has not been significantly manipulated. If the detector responses become overall lower over all the blocks, it is very likely that some kind of image processing operation has been applied. If only in a few blocks the detector responses are fairly lower than those in other blocks, one can estimate the probability that a block has been tampered based on the detector response. Since a medium-size block, e.g. 64×64, is needed to embed the spread spectrum watermark, this method can not achieve good tamper localization accuracy but only can provide an estimation of the undergone manipulations. If a smaller block size is used, the performance of the spread spectrum watermark will be significantly decreased. Furthermore, because robust watermarking technique is used in this scheme, the authenticator can not be very sensitivity to some elaborate modifications while it is fairly robust to common image processing like brightness/contrast adjustment and sharpening.

In [LC97][LC00][LC01], Lin et al. proposed a semi-fragile watermarking algorithm in DCT domain using the QIM embedding method. The proposed watermarking algorithm tolerates JPEG lossy compression to a pre-determined quality factor but is able to detect malicious manipulations. Two properties of DCT coefficients are used in the proposed authentication scheme. One is coefficient invariance that after quantizing a DCT coefficient to an integral multiple of the used quantization step, its value can be exactly recovered after JPEG compression with a smaller quantization step size. The other property is that the order relationship of DCT coefficient pair remains unchanged before and after JPEG compression. The second property is used to generate the authentication message and the first one to embed the message robustly against acceptable JPEG compression. In Lin's scheme, the authentication message generation and embedding process are performed on a basis of non-overlapping 8×8 blocks, similar to the JPEG compression process. In the authentication process, the extracted

authentication bits are compared with the regenerated ones. The proposed authenticator can localize the tempered blocks and recover the corrupted blocks approximately, if the recovery bits are also embedded. Similar to the Friedrich's scheme, the tamper localization accuracy of this method is also bounded by the used block size.

Eggers et al. proposed a watermarking technique for image authentication in [EG01]. The scheme is based on their previous work, the so-called SCS (Scalar Costa Scheme) watermarking technique in [ESG00]. A random binary sequence is embedded with a secret dither sequence into the DCT coefficients of 8×8 blocks. A likelihood test is used to determine whether the correct watermark is embedded with the specific key so as to examine if the image has been severely manipulated or not. The authors pointed out a fundamental problem of image authentication by semi-fragile watermarks that it is very difficult to embed watermarks in the flat image regions with moderate robustness. This problem will lead to false watermark detection in such regions.

In addition to the above-mentioned DCT domain techniques, some wavelet-based watermarking methods have also been proposed. In [KH99], Kundur et al. proposed a so-called telltale watermarking method, which embeds a random sequence independent of the image content into the wavelet coefficients. The image is first decomposed by a four-level wavelet transform using Haar bases. Then the watermark bits are embedded into the subbands of the four levels by the odd-even QIM embedding method. The decision to map the wavelet coefficients to odd or even is randomized by a secret key. The proposed authentication method is capable of characterizing the type of distortions based on the four levels of watermark extraction and verification. A similar wavelet-based approach was proposed in [YLL01]. After a four-level DWT of the image is taken, the mean value of a set of wavelet coefficients, instead of a single coefficient in Kundur's scheme, is used to embed a random sequence as the authentication data. The tampered area is estimated by using an information fusion procedure, which integrates the detection results obtained at multiple scales.

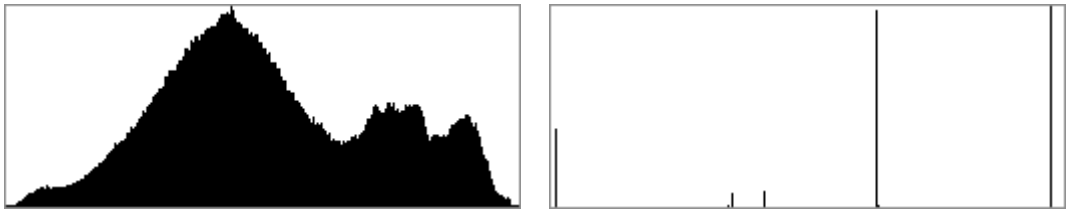
In [WKBC02], Winne et al. proposed a wavelet domain watermarking algorithm for image authentication by modifying a robust watermark algorithm. After the wavelet

transform of the image, the watermark data that is a random sequence is embedded into the first wavelet level. A vector is constructed from the three coefficients that are at the same frequency location but in the three different orientations, i.e. LH, HL and HH. The value of the median coefficient is quantized based on the watermark bit by an adaptive quantization step. A pre-distortion step is used to improve the performance and efficiency of the proposed algorithm. In the image authentication process, high tamper localization accuracy is achieved, which can deliver information about the shape of the modified object.

A new semi-fragile image authentication watermarking technique was proposed in [MSCS06], which improves the performance of the Lin's methods proposed in [LC97] [LC00] and [LC01]. This technique is essentially a modified version of Lin's scheme. Two possible solutions were presented to improve the tampering detection sensitivity: the random bias method and the non-uniform quantization method. Both methods reduce the probability that some types of manipulations remain undetectable, which cause only moderate changes of the feature values. In addition, the modified scheme extends the DCT-based watermarking technique to the wavelet domain and extends the acceptable image compression from JPEG to JPEG2000.

2.2.3 Watermarking for Synthetic Image Authentication

In addition to the classification based on the watermark fragility and according to the type of the targeted images, the authentication watermark techniques can be classified into two categories in another way: watermarking techniques for natural images and watermarking techniques for synthetic images. In the literature, most of the existing watermarking algorithms are designed for the natural images, which are usually true color or grayscale images. For example, most of the watermarking techniques introduced in the previous subsections are only applicable to true color and grayscale images, including both the fragile and semi-fragile watermarking schemes. These schemes achieve good watermark invisibility by taking advantage of an important property of natural images that they have continuous tone and their pixel values vary in



(a) (b)

Figure 2-2 Comparison of the histograms of natural and synthetic images: (a) Histogram of a natural image, (b) Histogram of a synthetic image.

a wide range. Because of this property, slightly modifying a pixel value will not cause perceptible artifacts.

For synthetic images, however, this property is not always true. The pixels of most synthetic images usually only take on very limited number of values, such as in simple line drawings, digital maps, etc. In binary text images, there are even less pixel values: only black and white. Figure 2-2 shows the difference of the histograms of natural and synthetic images. In addition, in synthetic images, there are usually large homogenous regions, in which there is only one uniform gray level or color. Hence arbitrarily changing the pixel values on a synthetic image will cause very visible artifacts. Furthermore, in practical applications, synthetic images are stored in different standard formats than natural images, which are usually palette-based and can only handle a limited number of colors. Due to these special characteristics of synthetic images, most of the existing watermarking algorithms for natural images can not be applied to the synthetic images straightforwardly. More detailed discussion about the special requirements of synthetic image watermarking will be addressed in Section 4.1.

Due to the simplicity of synthetic images, invisibly embedding a certain amount of watermark information becomes a more challenging task. Comparing to the watermarking algorithms for natural images, only a limited number of watermarking schemes for synthetic images have been proposed in the literature

[CWMA01][WL04][YK07], including watermarking techniques for formatted text images, drawings, digital maps, halftone images, generic simple images and so forth. A comprehensive review of the existing watermarking schemes for the synthetic images will be given in Section 4.2.

2.3 Problems and Challenges

From what has been addressed in the previous sections, we can see that most of the existing watermarking schemes do not satisfy all the requirements of an effective authentication system listed in Section 2.1. Especially, because some requirements are mutually competitive, a reasonable compromise is usually required in designing the authentication system. Therefore, the performance on one or more aspects will be inevitably and undesirably reduced. Furthermore, compared with natural images, studies on the synthetic image watermarking and authentication are still far from mature and satisfactory. Last but not least, selective watermarking and authentication according to the image content and the application requirement is also neglected by most of the existing techniques.

Overall, the existing problems and challenges can be summarized as follows.

- Tamper localization accuracy and image quality. As mentioned in previous sections, localizing the tampered regions is a very desirable feature in applications of image content authentication. In order to monitor every part of the image, a high watermark payload is subsequently required. However, embedding high volume of watermark information will introduce more data modification and therefore degrade the image quality. Hence, a desirable solution is to increase the tamper localization accuracy without increasing the watermark payload nor degrading the image quality more.
- Security of the existing semi-fragile watermarking algorithms. In order to achieve the capability of tamper localization, many semi-fragile watermarks are embedded in a block-based way. Since the embedding is performed locally,

they are somehow vulnerable to some kinds of local attacks, such as the VQ attack, especially when the watermarking algorithm is known to the adversary. Another security problem lies in the embedding strategy. Quantization-based embedding method is most often used in the watermarking algorithms for authentication. When an adversary knows the embedding algorithm, he can change the embedded data at will though the modified data might not match the authentication code, which presents concerns of counterfeiting attacks [HM00] [W03].

- Synthetic image watermarking and authentication. Most of the existing watermarking techniques are designed for natural images and can not be directly applied to synthetic images. Few studies have been done with regard to synthetic image watermarking. Due to the simplicity of synthetic images, it is more challenging to watermark synthetic images with high transparency, especially for content authentication supporting the tamper localization feature, because in this case more watermark payload is usually required. Therefore, the challenge is how to achieve content authentication with the desirable tamper localization for synthetic images by efficiently making use of the limited watermark capacity.

- Non-ubiquitous watermarking but ubiquitous authentication. In most of the existing watermark schemes for content authentication, no underlying semantic content is taken into account in the watermarking and authentication processes. The watermark information is usually embedded in a ubiquitous way over the whole image. In some applications, however, the fidelity requirement on some important image regions is differently specified. No slight image modification inside these regions is allowed. The integrity of these regions, however, is still of special importance and must be protected. So a new solution is needed that can provide an overall protection of the image, while being able to avoid modifying important image regions during the watermark embedding process.

More detailed specifications of the problems and challenges of the existing authentication watermarks will be given in each of the following chapters. Following the identified problems and challenges, we will propose corresponding solutions thereafter.

Chapter 3 Semi-Fragile Watermarking for Image Authentication

3.1 Introduction

As mentioned in the previous chapter, compared with cryptography, one of the advantages of watermarking techniques in authentication applications is the tamper localization capability. For image content authentication, this property is a particularly desirable feature in most applications. Besides the general integrity examination of the image content, the position information where the tampering has occurred is also very useful in practice. With the help of this information, other untouched image parts can still remain trustworthy and useful when the original data is not available. It can also help to infer the attacker's motives in many applications, such as in the case of forensic evidences.

In order to achieve the capability of localizing the tampered regions, many existing watermarking schemes embed the watermark in a block-based way [ESA04][LC00] [WL98][WKBC02]. As illustrated in Figure 3-1, the image is divided into blocks and the watermarks are embedded into every block respectively. The authentication of each block is done by verifying whether the watermark can be successfully extracted from

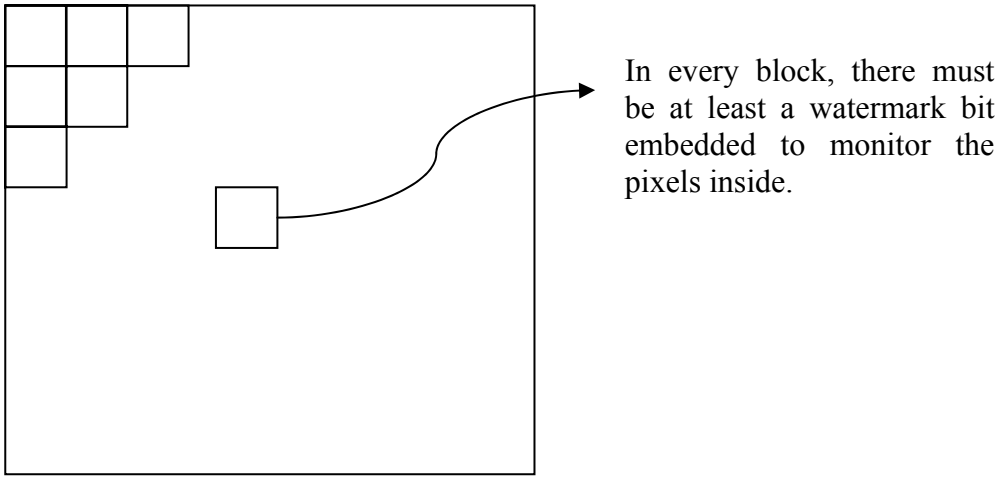


Figure 3-1 Illustration of the block-based watermarking for content authentication

the block. Hence, in the common block-based methods the maximal resolution of tampering detection is bounded to the block size that is used to embed the watermark. For example, in the algorithms proposed in [LC00] and [WL98], a block size of 8×8 is used and then the maximum detection resolution is limited to only 8×8 pixels. Moreover, because the block is the minimal unit that can contain at least one watermark bit, the maximal detection resolution is proportional to the watermark payload. In order to increase the detection resolution, a smaller block size must be used but this will lead to higher watermark payload. Subsequently, higher watermark payload will cause more artifacts and degrade the image quality. For example, in [WKBC02] the detection accuracy is improved to 2×2 pixels, but the watermark payload is also increased to 1 bit per 2×2 block. The challenge, therefore, is not only a good tradeoff between the two competitive factors, detection resolution and the watermark payload, but we also need to find a way to increase the detection resolution while embedding the same or even less watermark information.

Furthermore, in order to protect the whole image by block-based schemes, the authentication data, i.e. the watermark, has to be embedded locally all over the whole

image. However, as addressed in [EG01], it is very difficult to embed the data in smooth regions without causing noticeable artifacts [WL99], because the watermark capacity there is much lower than in other textured regions. Hence the watermark detection error rate will be significantly increased in such flat image regions. This problem will get even worse when embedding the watermarks into smaller blocks. In [WL99], the random shuffling is used to handle the uneven distribution of the watermark capacity in order to use the total watermark capacity of the image more efficiently. The goal in [WL99] is to utilize the available watermark capacity to hide as much as possible information. In this chapter, we will apply a similar idea not only to handle the uneven watermark capacity distribution, but more importantly, to enhance the tamper localization resolution with the same or less watermark payload.

In addition, another problem of the block-based methods is their security vulnerability against all kinds of local attacks. Because the block-based schemes embed the watermark locally, they show their weakness against such local attacks as block copy and paste, vector quantization (VQ) attacks and so forth. The VQ attacks are also known as collage attacks, which swap blocks in the same image or across different watermarked images [F02][OE05]. Almost all block-based watermarking methods are somehow vulnerable to such kinds of local attacks, particularly in case the authentication data and the embedding process are block independent. Not only block-DCT-based methods but also many DWT-based methods suffer from the VQ attacks due to the property of spatial-frequency localization of the wavelet transform. The threat of local attacks becomes even higher when the watermarking algorithm is known to the adversary.

Last but not least, the security of the embedding strategy itself is also one of our concerns. In the existing watermarking techniques for image authentication, quantization-based embedding methods are most often used. When the embedding algorithm is known to an adversary, he/she can modify the embedded data at will. This security problem can be alleviated in three ways: combining with cryptographic mechanisms, providing security to feature extraction and improving the embedding

mechanism itself. For example, traditional cryptographic techniques like the hash functions can be used in the watermarking systems to enhance the system security. These techniques, however, usually involve multiple pixel samples or coefficients. Hence, the cryptography-based watermarking algorithms can not always allow the localization of tampered regions to fine scale. Feature-based schemes have a similar problem since a feature is usually defined as a certain property of a set of image samples. With regard to this problem, an improvement using look-up table (LUT) embedding method was proposed in [W03], in which the maximal allowable run of "0" and "1" may be customized. For example, a maximal run of "0" and "1" can be increased to 2 comparing to the simple odd-even embedding (which is equivalent to the LUT embedding with the run of "0" and "1" always being 1). The LUT method, however, will degrade the image quality more, because more distortions are introduced when embedding the watermark with a larger run.

To solve the above-mentioned problems, in this chapter we propose a novel semi-fragile watermarking scheme for image authentication which allows to detect and localize tampered regions. We apply a random permutation process in the wavelet domain to build up a random map among the image locations. The randomly grouped wavelet coefficients refer mutually to each other. When any member of a group is manipulated, the whole group will be deemed as unverified. The final tamper detection and localization is based on the density of the unverified coefficient distribution. With a larger group size, we can reduce the necessary watermark payload while still keeping a high tamper localization resolution all over the whole image. The watermark can either be embedded into only the most suitable coefficients in each group or be proportionally distributed into all the coefficients. The coefficients whose modification causes less perceptual artifacts will take on a larger modification portion. In this way, we avoid embedding watermarks into the flat image regions but still have these region protected. Furthermore, the random permutation procedure enhances the security of the whole system against local attacks and it also improves the security of the embedding mechanism itself. Without the knowledge of the random permutation, even if the algorithm is publicly known, an adversary can not modify the embedded data.

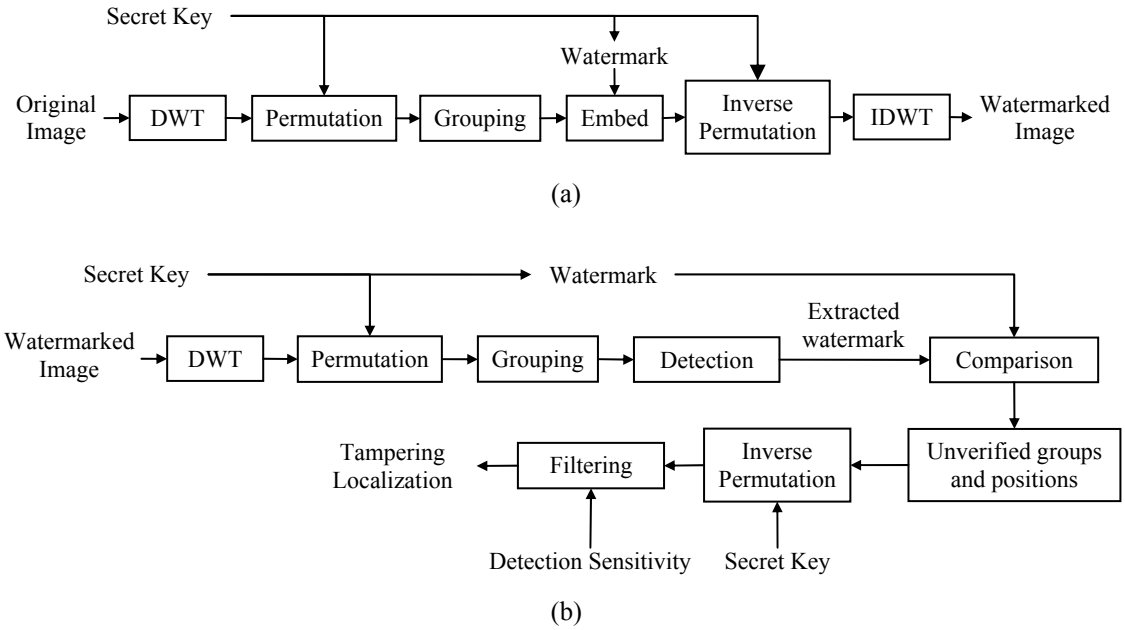


Figure 3-2 Block diagrams of the proposed watermarking scheme: (a) watermark embedding (b) watermark retrieval and image authentication.

The chapter is organized as follows. Firstly, in Section 3.2 we introduce the proposed watermarking scheme, including the watermark embedding and retrieval processes. Then, the image authentication process is presented in Section 3.3. Afterwards, we analyze the performance of the proposed scheme in Section 3.4 and discuss the extension of multi-resolution authentication in Section 3.5. The experimental results are given in Section 3.6. Finally, we conclude the chapter in Section 3.7.

3.2 Proposed Watermarking Scheme

The block diagram of the proposed authentication scheme is shown in Figure 3-2. It consists of two parts: the watermark embedding process, the watermark retrieval and image authentication process. We will introduce the watermark embedding and retrieval processes in this section. The image authentication part will be presented in the next section.

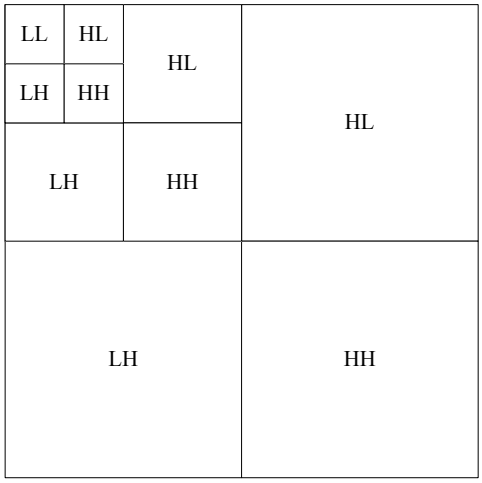


Figure 3-3 Syntax illustration of the wavelet decomposition

3.2.1 Watermark Embedding

The embedding process consists of three steps. The first step is to choose the decomposition level where the watermark will be embedded and to permute the coefficients. The second step groups the permuted coefficients and modifies the coefficients in every group if necessary. The last step inversely permutes the coefficients and performs the inverse wavelet transform to construct the watermarked image.

The proposed watermarking scheme can be applied to both gray level images and true color images. For simplicity, in the following sections, we introduce the proposed technique by embedding the watermark information in the luminance channel only. For true color images with red, green and blue channels, the luminance value of a pixel can be calculated by the formula I=0.299R+0.587G+0.114B, defined in the ITU-R BT.601 (formerly CCIR 601) standard, where R, G and B denote the red, green and blue values. When required, however, the embedding process can also be applied to R, G and B channels respectively.

3.2.1.1 Wavelet Coefficients Random Permutation

The proposed scheme performs watermark embedding in the Discrete Wavelet Transform (DWT) domain. The first level of wavelet decomposition produces four subbands, termed LL, LH, HL and HH. LL is a low resolution version of the original image and LH, HL and HH are the detail sub-images in horizontal, vertical and diagonal directions. The LL band is iteratively decomposed to obtain *R*-level wavelet transform as shown in Figure 3-3. More introduction on the wavelet transform can be found in [C96][D92]. The wavelet coefficients of different subbands are denoted as $f_{"level","subband"}$.

Depending on the application requirement, the subbands LH, HL and HH of one or more decomposition levels are used to embed the watermark. Embedding watermark in the high resolution level gives a higher capability of localizing the tampered regions but lowers the robustness to common image processing. On the contrary, embedding in the low resolution level will improve the watermark robustness while decreasing the accuracy of tempering localization. The performance of embedding in different levels will be discussed in the following sections. In the following part of this section, we suppose that the *r*th level is selected to embed the watermark and the resolution level variable *r* will be omitted from the text and equations because the embedding method is the same for other levels.

Before permuting the wavelet coefficients, all coefficients of the three subbands LH, HL and HH are firstly concatenated into a single sequence *S*. Three coefficients with the same coordinate of the three subbands, which correspond to the same spatial location, are continuously adjacent in the new sequence. Let $f_{HL}(m,n)$, $f_{LH}(m,n)$, $f_{HH}(m,n)$ denote respectively the coefficients of the different subbands, where (m,n) represents the position of the coefficient in the corresponding subband. The coefficients are rearranged in the following way:

$\{ f_{HL}(0,0), f_{LH}(0,0), f_{HH}(0,0), f_{HL}(0,1), f_{LH}(0,1), f_{HH}(0,1), ..., f_{HL}(M-1,N-1),$
$f_{LH}(M-1,N-1), f_{HH}(M-1,N-1)\},$

where M and N are the horizontal and vertical size of the subband, respectively. Figure 3-4 gives a visual illustration of the concatenation procedure when the first level decomposition is selected to embed the watermark.

Then the concatenated coefficients, i.e. the sequence S, are randomly permuted, as shown in Figure 3-4, controlled by a secret key k. In the permutation process, a minimal distance d between any adjacent members in S is required in the new permutated sequence in order to ensure adjacent coefficients in S are separately distributed after permutation. In this way, the coefficients corresponding to the same spatial location will be separated with a minimal distance d since they are continuously concatenated in S. The new sequence containing the randomly permuted coefficients is denoted as S'. Table 3-1 depicts a random permutation algorithm.

Table 3-1 Depiction of a random permutation algorithm

1. Initialization the variables.
   Set l the length of the coefficient sequence S.
   Set $index[l]$ the index of random permutation.
   Set i the index of $index[]$ and initialize i to 1.
   Set d the required minimal distance.
2. Generate a random number r_i between 1 and l.
3. If r_i already exists in $index[l]$, then go back to Step 2 to regenerate r_i.
3. If $|index[i-1]-r_i| < d$, then go back to Step 2 to regenerate r_i.
4. $index[i]= r_i$ and $i=i+1$, if $i \geq l$, then stop; otherwise, go back to Step 2.

After the random permutation, the sequence S' is divided into groups with a fixed group size g, as shown in Figure 3-4. In every group, one watermark bit will be

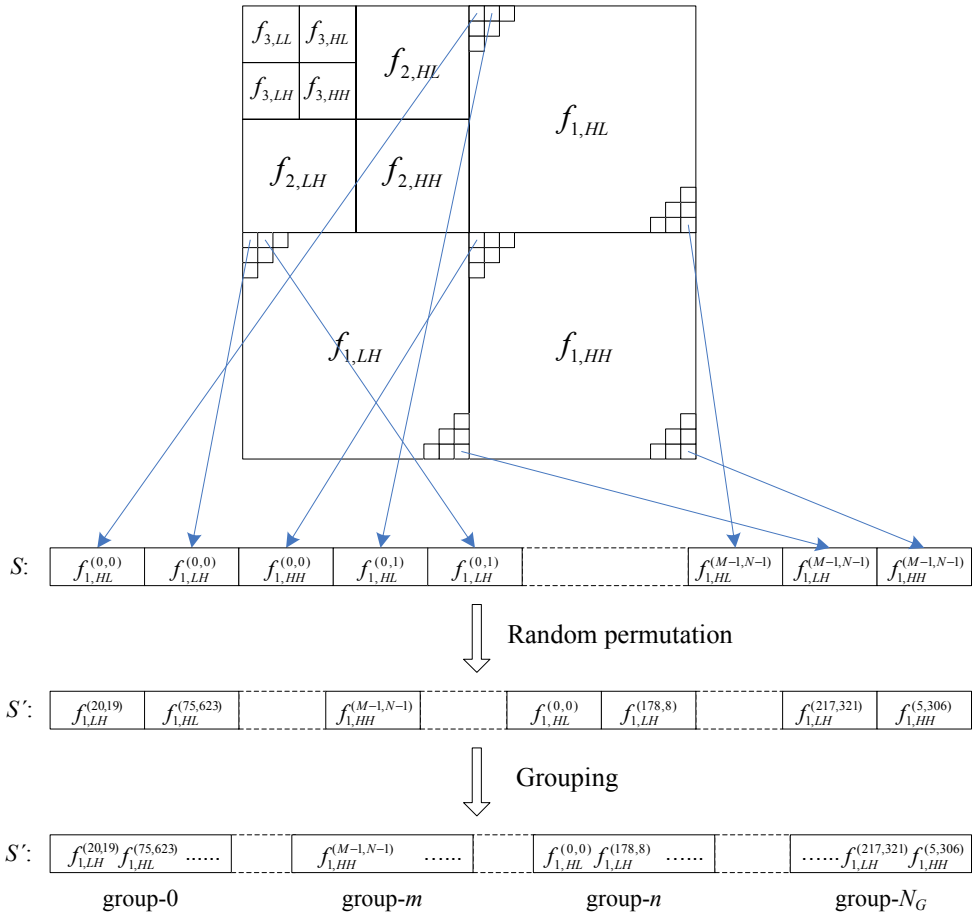


Figure 3-4 Wavelet coefficients concatenation and random permutation

embedded. The embedded bit will monitor all the members of this group. The members of each group correspond to various spatial locations, thanks to the random permutation process. The random permutation process also distributes the coefficients suitable for watermark embedding evenly over all the groups [WL99]. This property ensures that the embedding process makes no modifications in smooth image regions and therefore improves the fidelity of the watermark image.

The group size g determines the watermark payload and the number of the mutual referred coefficients. Thereby it will affect the watermarked image's quality and the maximal localizable tampered area. With a larger g, fewer watermark bits will be embedded, so a higher fidelity of the watermarked image will be achieved. But this will

not decrease the detection resolution, although it will reduce the size of the maximal localizable tampered area. We will discuss this aspect more in Section 3.3.

3.2.1.2 Wavelet Coefficient Modification

The watermark w consists of a binary random sequence, generated by the secret key, $w \in \{0,1\}$. The random sequence serves as an authentication code. This code is compared with the retrieved watermark in the authentication process. More details are given in Section 3.3.

The embedding process is performed by quantizing the weighted mean of the wavelet coefficients. In every group of the random sequence S', the weighted mean of all the wavelet coefficients is defined as

$$s_j = \sum_{i=1}^{g} p_i \cdot \left| f_j(i) \right|, \tag{3-1}$$

where $f_j(i)$ is the ith coefficient in the jth group and g is the group size. p is a bipolar random sequence with uniform distribution, $p_i \in \{-1,1\}$.

To embed the watermark, we then quantize s_j by a quantization step Q as shown in the following equations:

$$s_j = \lfloor s_j / Q \rfloor \cdot Q + \Delta_j, \tag{3-2}$$

$$Quan(s_j) = \begin{cases} 0 & if \ \ \lfloor s_j / Q \rfloor \ is \ even \\ 1 & if \ \ \lfloor s_j / Q \rfloor \ is \ odd \end{cases}, \tag{3-3}$$

where $\lfloor . \rfloor$ is the floor function and Δ_j is the quantization residue.

The watermark bit $w(j)$ is embedded by modifying the weighted mean s_j so that $Quan(s_j)$ is equal to $w(j)$. As shown in Figure 3-5, the weighted mean s_j is modified to the nearest 0 bin (the dashed line) or 1 bin (the solid line) according to $w(j)$. The

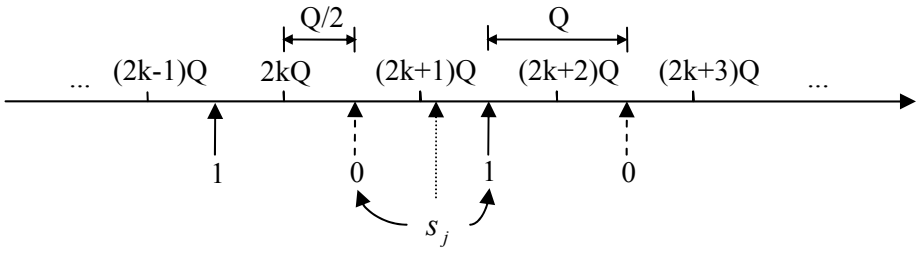


Figure 3-5 Illustration of the quantization process

bins of 1 and 0 are located in the middle of the quantization interval. Specifically, the modification of s_j is performed as

$$s_j^* = \begin{cases} \lfloor (s_j + Q/2)/Q \rfloor \cdot Q + Q/2, & if \quad Quan(s_j + Q/2) = w_j \\ \lfloor (s_j + Q/2)/Q \rfloor \cdot Q - Q/2, & if \quad Quan(s_j + Q/2) \neq w_j \end{cases}, \tag{3-4}$$

where s_j^* is the expected weighted mean value of the *j*th group.

The advantage of using weighted mean is that it preserves small variation when incidental distortions are encountered, because usually common image processing, e.g. JPEG compression, is applied to the entire image and causes similar distortions over the whole image area. For instance, if the magnitudes of all the $f_j(i)$ in the *j*th group are changed by Ω due to some distortions, then

$$s_j^{'} = \sum_{i=1}^{g} p_i \cdot (|f_j(i)| + \Omega) = \sum_{i=1}^{g} p_i \cdot |f_j(i)| + \sum_{i=1}^{g} p_i \cdot \Omega \,. \tag{3-5}$$

Since p is uniformly distributed, the second part of the above equation is approximately equal to zero. Then the weighted mean becomes

$$s_j' \approx \sum_{i=1}^{g} p_i \cdot |f_j(i)| = s_j \,. \tag{3-6}$$

For simplicity, the weighting coefficient p_i can be replaced by $(-1)^i$ and Equation (3-6) still holds. When the group size g is an even number, it becomes

$$s_j^{'} = \sum_{i=1}^{g} (-1)^i (\left| f_j(i) \right| + \Omega) = \sum_{i=1}^{g} (-1)^i \left| f_j(i) \right| + \sum_{i=1}^{g} (-1)^i \Omega = \sum_{i=1}^{g} (-1)^i \left| f_j(i) \right| = s_j . \quad (3\text{-}7)$$

On the contrary, malicious manipulations commonly occur in local image regions and will only distort the wavelet coefficients corresponding to the tampered regions. Because the random permutation process spreads all those local coefficients in different groups, in every affected group only one or few individual coefficients are distorted. Therefore, these distortions will change the weighted mean and cause a larger variation. Consequently, the quantization result may shift.

In order to modify the weighted mean s_j, we propose two methods to update the coefficients in every group as described respectively in the following.

***Method 1*:**

One way is to modify the coefficient with the maximal magnitude in the group, because the modification of such coefficients with large magnitude causes less noticeable artifacts than other coefficients with small magnitude. Here we define the coefficients whose absolute magnitudes are large as large coefficients independent of its sign. Large coefficients indicate that the image has more textured contents at the corresponding spatial location, while small coefficients correspond to the smooth regions. As mentioned in Section 3.2.1.1, the random permutation process distributes the large coefficients evenly across all groups, which ensures that the probability of every group having at least one large coefficient is quite high.

Let δ_j denote the difference between the expected weighted mean value and the original one:

$$\delta_j = s_j^* - s_j . \quad (3\text{-}8)$$

The largest coefficient is updated as follows:

$$f^{*}_{j,\max}(i) = f_{j,\max}(i) + p_i \cdot sign(f_{j,\max}(i)) \cdot \delta_j , \tag{3-9}$$

where $f_{j,\max}(i)$ is the coefficient in the jth group with the maximal magnitude, $sign(x)$=1 if $x \geq 0$ and $sign(x)$=-1 if x<0. If the sign of $f_{j,\max}(i)$ is changed after applying Equation (3-9), then $f_{j,\max}(i)$ is set to zero. In this case, the residue of the difference δ_j is $\delta_{j,residue} = sign(\delta_j)(|\delta_j| - |f_{j,\max}(i)|)$ after updating the largest coefficient. Then the second largest coefficient is updated by applying Equation (3-9) with $\delta_{j,residue}$. If there is still a residue $\delta_{j,residue} > 0$, other coefficients are updated accordingly until $\delta_{j,residue} = 0$. Table 3-2 depicts the coefficient-updating algorithm in a single group.

Table 3-2 Depiction of the coefficient-updating algorithm (Method 1)

1. Initialize the variables:

   Set g the group size.
   Set $sortf[g]$ the descending sorted (based on magnitude) coefficient sequence in group j.
   Set $\delta_{j,residue} = \delta_j$, the residue of the difference δ_j.
   Set $sign$ the sign of δ_j, if $\delta_j > 0$ $sign$=1, otherwise $sign$=-1.
   Set k the index of $sortf[]$ and initialize k to 0.
   Set $i[k]$ the original index of $sortf[k]$ in the group j.

2. Update the coefficients:

```
do{
    if( p_{i[k]} · δ_{j,residue} <0  &&  |sortf_j(k)| < |δ_{j,residue}| ){
        δ_{j,residue} =sign·( |δ_{j,residue}| - |sortf_j(k)| );
        sortf_j(k) =0;
        k++;
    }
    else{
        sortf_j(k) = sortf_j(k) + p_{i[k]} · δ_{j,residue} ·( sortf_j(k) > 0 ?1 : -1 );
        δ_{j,residue} =0;
    }
}while( |δ_{j,residue}| > 0  &&  k<g );
```

***Method 2*:**

The second method is to assign the watermark energy to all the members of the group by updating every coefficient in the group. The whole modification amount of the weighted mean s_j is not evenly distributed to every coefficient. The modification amount of every coefficient is determined by the proportion of its magnitude to the sum of the magnitudes of all the coefficients. Every coefficient is updated as

$$f_j^*(i) = f_j(i) + p_i \cdot sign(f_j(i)) \cdot \delta_{ji}, \tag{3-10}$$

where δ_{ji} is the update amount of the *i*th coefficient in the *j*th group that is calculated as

$$\delta_{ji} = \frac{|f_j(i)|}{\sum_{i=1}^{g} |f_j(i)|} \delta_j . \tag{3-11}$$

If the sign of $f_j(i)$ is changed after applying Equation (3-10), then $f_j^*(i)$ is set to zero. In this way, every coefficient will be modified and the larger coefficients will be changed more than the smaller ones, because the large coefficients can bear more distortion without causing perceptible artifacts. The correctness of the procedure can be seen as follows.

From Equation (3-8), the expected weighted mean value of the *j*th group is $s_j^* = s_j + \delta_j$, then

$$\begin{aligned} s_j^* &= \sum_{i=1}^{g} p_i \cdot |f_j(i)| + \sum_{i=1}^{g} \delta_{ji} = \sum_{i=1}^{g} p_i \cdot \left(|f_j(i)| + p_i \cdot \delta_{ji} \right) \\ &= \sum_{i=1}^{g} p_i \cdot sign(f_j(i)) \cdot \left(f_j(i) + p_i \cdot sign(f_j(i)) \cdot \delta_{ji} \right) \\ &= \sum_{i=1}^{g} p_i \cdot sign(f_j(i)) \cdot f_j^*(i). \end{aligned} \tag{3-12}$$

By imposing the constraint that if $sign(f_j^*(i)) \neq sign(f_j(i))$, then $f_j^*(i)$ =0, we obtain

$$s_j^* = \sum_{i=1}^{g} p_i \cdot sign(f_j^*(i)) \cdot f_j^*(i) = \sum_{i=1}^{g} p_i \cdot \left| f_j^*(i) \right| . \tag{3-13}$$

Thus, when $f_j(i)$ is updated to $f_j^*(i)$, the weighted mean value s_j is updated to s_j^* accordingly.

Theoretically, the two coefficient update methods will cause the same amount of image modification of δ_j. However, when applied to the image data, these two methods will achieve different image quality and detection performance. Some comparisons and discussions will be given in Section 3.6.

After all the watermark bits are embedded in all the groups of S', the sequence S' is inversely permutated to reconstruct the order of coefficients as in the original sequence *S*. The wavelet coefficients are then put back to their original positions in the subbands. Finally, the inverse wavelet transform is performed to obtain the watermarked image.

3.2.2 Watermark Retrieval

The secret key *k*, the selected level *r* of wavelet decomposition and the group size *g* are conveyed to the watermark detector as side information. The watermarked image is firstly decomposed by *r*-level wavelet transform. Then the coefficients in the three subbands HL, LH and HH of the *r*th level are concatenated in the same way as in the embedding process. Controlled by the secret key *k*, the same random permutation is performed to reconstruct the sequence S'.

The sequence S' is then divided into groups with the group size *g*. The weighted mean of all the coefficients in every group is recalculated and the watermark bit is extracted by quantizing the weighted mean

$$w_j^{'} = Quan(s_j^{'}) , \tag{3-14}$$

where $s_j^{'}$ is the recalculated weighted mean of the *j*th group and $w_j^{'}$ is the extracted watermark bit from the *j*th group.

3.3 Image Authentication Process

Every extracted watermark bit $w_j^{'}$ is compared with the embedded one that is generated by the secret key k. If all the extracted watermark bits match the original ones, the image is claimed authentic, otherwise it may be tampered. The following steps are taken to localize the tampered areas. A visual illustration of the authentication process is given in Figure 3-6.

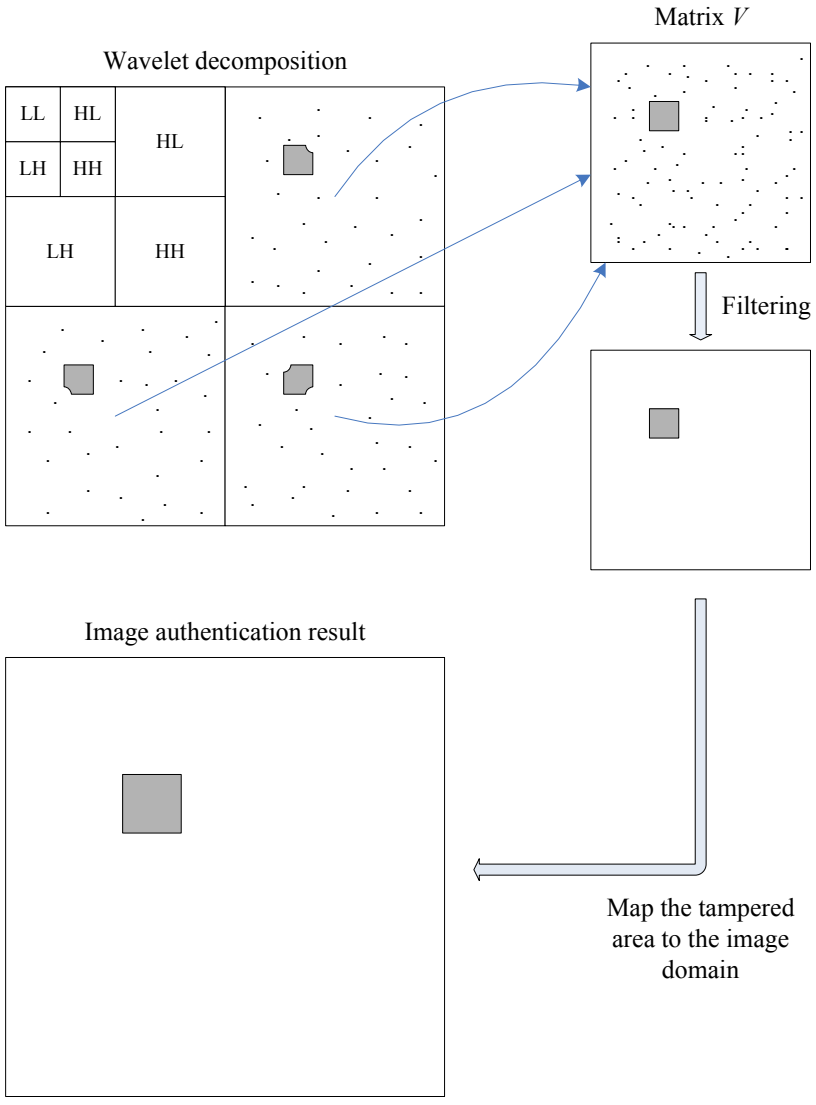


Figure 3-6 Illustration of the authentication process

Step 1:

For every group, if the extracted watermark bit does not match the embedded one, the whole group is considered as an unverified group. All the coefficients in the group are potentially tampered. Since we do not know exactly in this stage which coefficient is manipulated, all the members of the group are marked as unverified coefficients whether they are tampered or not. The actually tampered coefficients will be identified in Step 4.

Step 2:

All the coefficients in the sequence S' are mapped back to their original positions in the wavelet subbands by the inverse permutation. Since the coefficients in every unverified group come from various locations, all those unverified coefficients will randomly scatter over the three subbands as shown in Figure 3-6. At the location corresponding to the tampered region in every subband, the density of the unverified coefficients is distinctly much higher than in other portions, which is shown in Figure 3-6 as the gray areas. This is because all the unverified groups in S' contain either one or more coefficients coming from the tampered region. After mapped back to the wavelet subbands, those unverified coefficients are clustered together again. The other isolated unverified coefficients come from those groups that the tampered coefficients belong to. Thanks to the random permutation, they are distributed over the subbands sparsely like random noises.

Step 3:

We construct a matrix $\boldsymbol{V}$ of the same size of the subband at level-r wavelet decomposition, i.e. $1/4^r$ of the image size, in which every position corresponds to a $2^r \times 2^r$ pixel block of the image. For example, if r=1, namely, the watermark is embedded in the 1st level of the wavelet decomposition, the size of the matrix $\boldsymbol{V}$ is $(W/2)\times(H/2)$, where W and H denote the width and height of the image, respectively. We consider a position $V(m,n)$ in the matrix $\boldsymbol{V}$ as unverified when there

is an unverified coefficient at the same position in any subband of HL, LH, and HH. $V(m,n)$ is set to 1 if it is an unverified position, otherwise it is set to 0:

$$V(m,n)=\begin{cases}1, & \text{if any of } f_{r,HL}(m,n), f_{r,LH}(m,n), f_{r,HH}(m,n) \text{ is unverified}\\ 0, & \text{otherwise.}\end{cases} \tag{3-15}$$

As shown in Figure 3-6, the isolated unverified coefficients in the subbands are still randomly distributed over the matrix $\boldsymbol{V}$ like random noises, while in the tampered region the density of the unverified coefficients becomes even higher than in every individual subband.

Step 4:

Based on the fact that the tampering commonly occurs in a continuous area of the image in the practical applications, only the region with the high density of the unverified coefficients corresponds to the actual manipulation. In *Sept 1* we consider all the coefficients in an unverified group as unverified. Most coefficients, however, in an unverified group are in fact not manipulated. The isolated unverified coefficients in the subbands correspond to those unchanged coefficients in the unverified groups. We define those isolated unverified coefficients as *noise dots*. Other unverified coefficients, which are mapped back to adjacent positions, are identified as the actually tampered ones.

In order to refine the authentication result and give an accurate tampering localization, a noise filter is applied to the matrix $\boldsymbol{V}$ to remove the *noise dots*, i.e. the noise-like unverified positions, so that the tampered region can be easily picked out. The filtering process is performed as

$$V(m,n)=\begin{cases}1, & \text{if } \sum_{i=-d/2}^{d/2}\sum_{j=-d/2}^{d/2} V(m+i,n+j)F(i+\frac{d}{2},j+\frac{d}{2})>T\\ 0 & \text{otherwise,}\end{cases} \tag{3-16}$$

where d is the filter size, T is a preset threshold and F is (when the filter size d is set to 5)

$$F = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \tag{3-17}$$

The filter size d and the threshold T determine the sensitivity of the tampering detector and can be set according to the application requirement.

As shown in Figure 3-6, after filtering out the isolated unverified positions, the remaining unverified positions in $\boldsymbol{V}$ indicate the actually tampered location and are then mapped to the image spatial domain to locate the corresponding region. Since the size of the matrix $\boldsymbol{V}$ is $1/4^r$ of the image size, the authentication process provides a maximum detection resolution of $2^r \times 2^r$ blocks in the image domain. The localization capability of the proposed scheme will be discussed in Section 3.4.2.

3.4 Performance Analysis

In this section we will evaluate the performance of the proposed authentication scheme with respect to the quality of the watermarked image, the localization capability, the sensitivity of tampering detection and the security of the watermarking scheme.

3.4.1 Quality of Watermarked Image

The quality of the watermarked image is one of the most important concerns of watermarking applications. High quality and fidelity of the watermarked image is commonly desired. The Peak Signal-to-Noise Ratio (PSNR) is usually used to measure the quality of the watermarked image, which is defined as

$$PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right), \tag{3-18}$$

where MSE denotes the mean squared error (MSE)

$$MSE = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}\left\|I'(x,y) - I(x,y)\right\|^2, \tag{3-19}$$

I' is the watermarked image and I is the original one. W and H denote the width and height of the image respectively. Although it is known that PSNR can not always provide a good estimate of the true image fidelity [CMB01], the PSNR value gives an objective overall evaluation of the watermarked image quality and is the most often used measurement to evaluate the watermarking algorithms in related works.

In the proposed authentication scheme, the image distortion is caused by modifications of the wavelet coefficients in the watermark embedding process. Obviously, the quantization step Q used in Equation (3-4) will affect how much the quality of the watermarked image degrades. A larger quantization step will incur more modification to the wavelet coefficients, consequently resulting in more degradation of the watermarked image. The mean squared error incurred by the quantization can be derived as follows. In both the following two cases, we assume that the original wavelet coefficients are uniformly distributed over the range of $\psi = [kQ,(k+1)Q]$. First, we consider the case that the quantization result of the original coefficient weighted mean s_j matches the watermark bit $w(j)$. In this case, the weighted mean s_j in the range of ψ is rounded to $(k+1/2)Q$ and the mean squared error caused by the quantization is

$$MSE_1\big|_{\psi} = \frac{1}{Q}\int_{-\frac{Q}{2}}^{\frac{Q}{2}}\tau^2 d\tau = \frac{Q^2}{12}. \tag{3-20}$$

Next we consider the other case that quantization result does not match the watermark bit $w(j)$. The weighted mean s_j is modified to $(k-1/2)Q$ or $(k+3/2)Q$ determined by which one is nearest to s_j, incurring the mean squared error

$$MSE_2\big|_{\psi} = \frac{1}{Q}\left(\int_{\frac{-Q}{2}}^{0}(\tau+Q)^2 d\tau + \int_{0}^{\frac{Q}{2}}(\tau-Q)^2 d\tau\right) = \frac{7Q^2}{12}. \tag{3-21}$$

Assuming that with the probability of 1/2 the quantization result of s_j matches the watermark bit, the overall mean squared error caused by the quantization is

$$MSE\big|_{\psi} = MSE_1\big|_{\psi} + MSE_2\big|_{\psi} = \frac{1}{2}\cdot\frac{Q^2}{12} + \frac{1}{2}\cdot\frac{7Q^2}{12} = \frac{Q^2}{3}. \tag{3-22}$$

Equation (3-22) gives us the average distortion caused by embedding one watermark bit. If the watermark payload is *p*, namely, totally *p* watermark bits are embedded in the whole image, the mean squared error of the whole wavelet domain is

$$MSE\big|_{wavelet\ domain} = \frac{pQ^2}{3WH}. \tag{3-23}$$

In the proposed embedding scheme, one watermark bit is embedded in every group of the permuted sequence S', so the watermark payload is determined by the length of the sequence S' and the group size *g*. The length of the sequence S' depends on the image size and the selected wavelet decomposition level *r* used to embed the watermark. Therefore, for an image of fixed size, the group size *g* and the level *r* will determine the number of the groups, i.e., the number of watermark bits that will be embedded. So Equation (3-23) can be rewritten as the follows.

Given an image of the size $W \times H$, the length of the sequence S' is $l = \frac{3}{2^r \times 2^r}\cdot WH = \frac{3WH}{4^r}$, the watermark payload is $p = l/g$, and the mean squared error of the whole wavelet domain can be derived as

$$MSE\big|_{wavelet\ domain} = \frac{Q^2}{4^r g}. \tag{3-24}$$

According to the Parseval's Theorem, the mean squared error of the whole image is equal to its counterpart in the wavelet domain, $MSE\big|_{image\ domain} = MSE\big|_{wavelet\ domain}$. Therefore, the PSNR of the watermarked image is

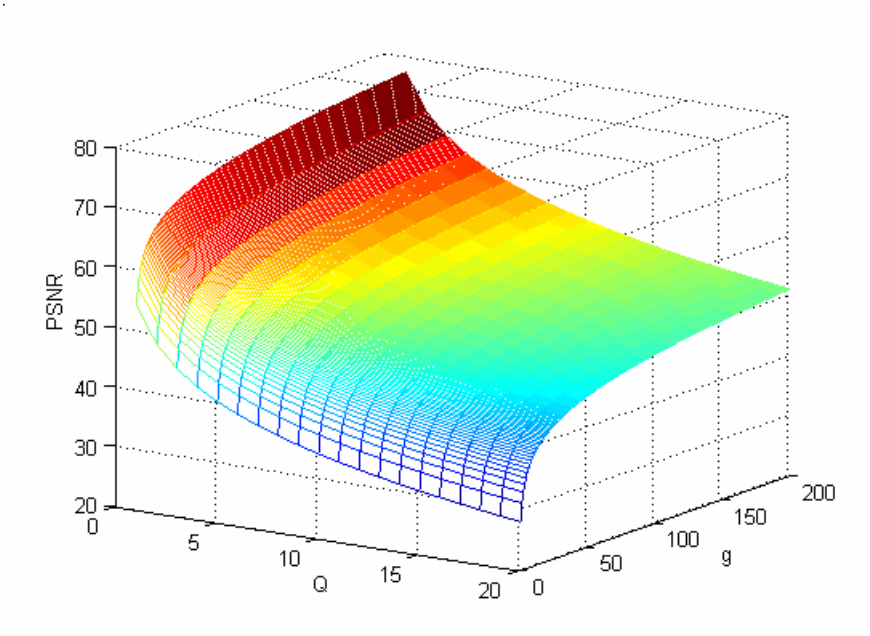


Figure 3-7 PSNR of the watermarked image for different quantization steps Q and group sizes g, where the watermark is embedded in the first wavelet level (r=1).

$$PSNR = 20\log_{10}(\frac{255}{Q}\sqrt{\frac{3WH}{p}}) = 20\log_{10}(\frac{255}{Q}\sqrt{4^r g}). \quad (3\text{-}25)$$

Equation (3-25) reveals that generally the quality of the watermarked image is determined by the quantization step and the total watermark payload. Specifically, in the proposed scheme the quality of the watermarked image is determined by the group size and selected wavelet level when the quantization step Q is fixed. With r=1, Figure 3-7 plots the PSNR of the watermarked image versus the quantization step Q and the group size g. We will demonstrate that the analytic results conform to the experimental results very well in Section 3.6.1.

It is also indicated in Equation (3-25) that embedding watermark in a higher decomposition level will improve the quality of the watermarked image. However, this

is achieved at the cost of decreasing the resolution of tampering localization. We will discuss this aspect in the next section.

3.4.2 Localization Capability and Probability of False Alarm

In the proposed scheme, the content of the image is monitored by the watermark embedded in the wavelet domain. Every position is verified by the watermark bit embedded in the corresponding wavelet coefficients, thanks to the spatial-frequency localization of the wavelet transform. Since every coefficient in the *r*th level wavelet decomposition corresponds to a $2^r \times 2^r$ block in the image domain, the maximal tampering detection resolution is limited to $2^r \times 2^r$. In the proposed watermarking scheme, changing any single coefficient may result in a mismatch between the detected watermark bit and the embedded one. Therefore, the proposed scheme can detect the change of a single coefficient and achieve the maximal tampering detection resolution of $2^r \times 2^r$ in the image domain. Obviously, embedding watermark in a higher wavelet decomposition level will decrease the tampering detection resolution.

Note that the maximal detection resolution is independent of the watermark payload. And as mentioned in the previous section, the watermark payload is determined by the group size *g*. This implies that increasing the group size will not decrease the tampering detection resolution while it decreases the watermark payload. In other words, we can embed less watermark bits while keeping the same tampering detection resolution. This advantage is achieved by the random permutation process. The inverse random permutation distributes the unverified coefficients sparsely over the subbands while the actually changed ones are clustered together at the tampered location. As long as the density of the unverified coefficients at the tampered location is distinguishable from the other regions, the tampered region can be localized. Compared with the traditional block-based algorithms, by using random permutation the proposed scheme significantly reduces the watermark payload while keeping the same detection resolution. Consequently it improves the quality of the watermarked image. For example, in order to achieve a maximal detection resolution of 4×4 pixels, the

traditional algorithms have to embed at least one watermark bit in every 4×4 blocks, namely, the embedding rate is 0.0625 bit/pixel. To achieve the same detection resolution, the proposed watermarking scheme, however, embeds only one bit in every group of the coefficient sequence obtained from the second level of wavelet decomposition. The embedding rate is decreased to $0.0625*3/g=0.1875/g$ bit/pixel. When the group size is set to 3, the watermark payload is equal to the block-based algorithms. Figure 3-8 plots the quality comparison when applying variable group sizes.

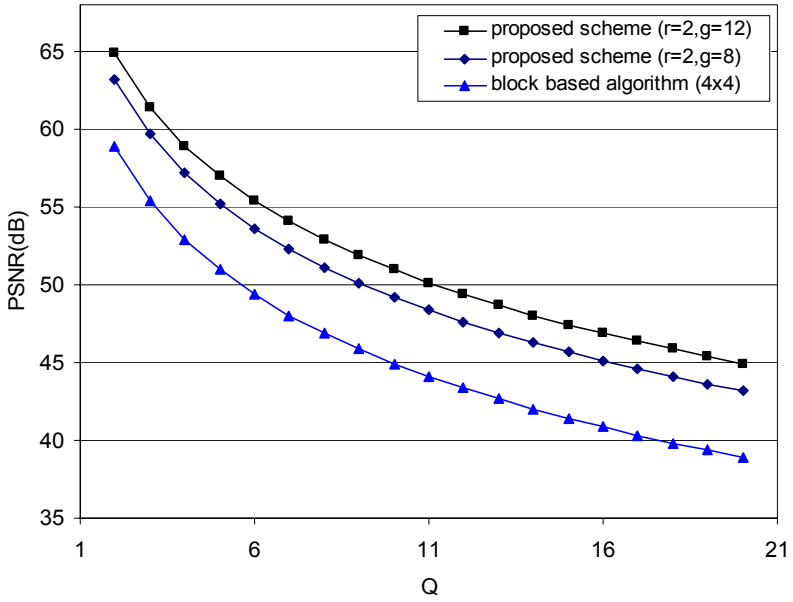


Figure 3-8 Watermarked image quality comparison between the classical block-based technique and the proposed scheme with tamper localization resolution of 4×4 pixels.

Nevertheless, increasing the group size will increase the false alarm rate and decrease the maximum tampered area that can be localized. Since changing one single coefficient may render a complete group unverified and all the coefficients in the group are deemed as potential unverified coefficients, when those potential unverified coefficients that come from unaltered area casually form a connected region of a certain size, a tampered area will be falsely detected, i.e. false alarm will be aroused. Let N_e

be the number of wrongly detected watermark bits, i.e. the number of unverified groups, then the total number of potential unverified coefficients will be $N_u = N_e g$. Thus, the bit error rate of watermark detection is

$$r_{BER} = \frac{N_e}{N_T} = \frac{N_e}{\frac{3WH}{(2^r \times 2^r)g}} = \frac{(2^r \times 2^r)N_e g}{3WH} = \frac{(2^r \times 2^r)N_u}{3WH}, \tag{3-26}$$

where N_T is the total number of groups.

When all unverified coefficient positions are mapped to the matrix $\boldsymbol{V}$, some of them might overlap each other. Let P_o be the probability of coefficient overlapping, then the probability that a position in matrix $\boldsymbol{V}$ is a potential unverified position will be

$$P_u = \frac{N_u(1-P_o)}{WH/(2^r \times 2^r)} = 3(1-P_o)r_{BER}. \tag{3-27}$$

Then the probability that there are at least t unverified positions in a 3×3 neighborhood is

$$P_{fa1} = \sum_{k=t}^{8} \binom{8}{k} P_u^{\,k} (1-P_u)^{8-k}, \tag{3-28}$$

where $t \in \{0,1,2,\cdots,8\}$. Figure 3-9 plots the probability P_{fa1} versus the number of unverified positions t for different P_u. When P_u=0.1, i.e. the number of unverified positions in matrix $\boldsymbol{V}$ takes on 10% of the total positions, the probability that there are 4 unverified positions in the 3×3 neighborhood is 5×10^{-3}. When the number of unverified positions go down to 5%, this probability drops to 3.7×10^{-4}. With totally 10% of unverified positions, i.e. P_u=0.1, the probability of casually forming a 3×3 unverified region is as low as 3.7×10^{-8}. With smaller P_u, this probability becomes even smaller. Figure 3-10 plots the probability P_{fa1} versus P_u for different t. When the proportion of unverified positions rises to 50%, the probability of at least 5 unverified positions in the 3×3 neighborhood increases to 0.36.

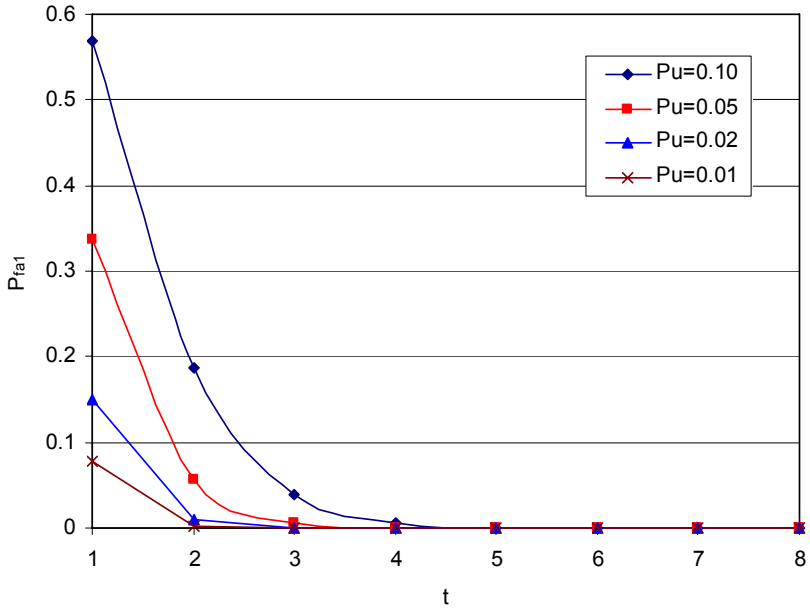


Figure 3-9 P_{fa1} versus the number of unverified positions t in the 3×3 neighborhood for different probabilities of being unverified position P_u.

Equation (3-28) reveals that high probability of being unverified position, i.e. large amount of unverified positions, will render a high probability of wrongly detecting unverified region. From Equation (3-26), we can see that with a certain amount of watermark detection errors the amount of unverified coefficients is determined by the group size. A larger group size g will cause more potential unverified coefficients outside the actually tampered region (defined as *noise dots* in Section 3.3), namely, it will increase the density of the *noise dots*. When a very large area is tampered, the density of the *noise dots* will become so high that it is very difficult or even impossible to distinguish the correct tampered region from the *noise dots* in the matrix $\boldsymbol{V}$. In this case, the tampered region can be localized and the whole image will be deemed as unverified.

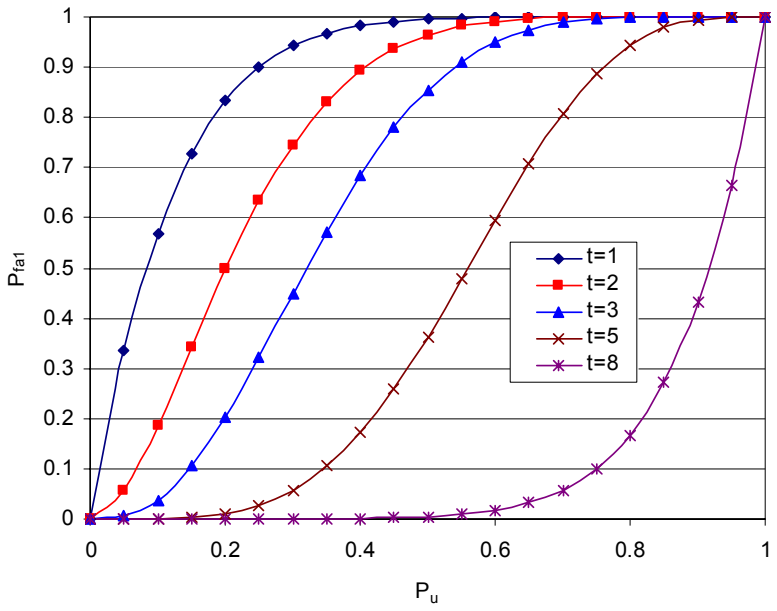


Figure 3-10 P_{fa1} versus the probability of being unverified position P_u for different numbers of unverified positions t in the 3×3 neighborhood.

3.4.3 Tampering Detection Sensitivity

The tampering detection sensitivity of the proposed scheme is basically determined by the quantization step Q in Equation (3-4). Based on the fact that image tampering usually occurs in local regions, such image manipulation can be considered as burst errors with large variance that will affect the local frequency property and render a set of frequency coefficients to be changed. These changed coefficients will be distributed into different groups. When the weighted mean of any group is shifted in any direction by the amount $\Delta \in (nQ - Q/2, nQ + Q/2]$ where n is odd, the quantization result will be changed and tampering alarm will be raised. When the tampered area is of a reasonable size, the number of changed coefficients will be large as well as the number of affected groups. So the probability of keeping the weighted means of all the affected groups will be very small.

There are two cases in which the tampering will not be detected. One is that in the practical applications the tampered area may contain some pixels that happen to be very similar to or even the same as the original ones. So the frequency at the corresponding location may remain unchanged. The other case is when the changed amount of the weighted mean of a group $\Delta \in (nQ - Q/2, nQ + Q/2]$ where n is even, the extracted watermark bit will still match the original one. This will also lead to the failure of detecting the tampering in the group. In these two cases, some positions inside the tampered region may remain verified. In other words, there will be some holes in the detection result. Those errors can be compensated by applying the noise filter in Equation (3-16). The holes will be filled by averaging with their neighbors.

Moreover, as the weighted mean is composed of frequencies in selected wavelet levels that correspond to certain frequency bands, if the image manipulation does not change the frequencies of that band, it can not be detected. For instance, manipulating a smooth region to another smooth one is difficult to detect by checking certain frequency change. For this reason, multi-level embedding can be used to increase the sensitivity to all kinds of manipulations. We shall introduce multi-level embedding and authentication in Section 3.5. Another solution to improve the sensitivity is to use content-based features, such as the most significant bit of the mean of a macro-block. Using the intensity-involved content features can monitor such kind of tampering that does not modify high and middle frequencies. In addition, if only the luminance value of pixels is used for watermark embedding, color manipulations can not be detected. Therefore, according to the application requirement, certain color channels can be used to embed the watermark respectively in order to ensure the protection of corresponding colors.

Instead of malicious tampering, the watermark detection may also be impaired by incidental distortions introduced by common image processing. However, comparing to malicious tampering, the variance of incidental distortions is much smaller. If we assume that the weighted mean is distorted by additive noise that follows i.i.d.

Gaussian distribution with zero mean and variance σ^2, the probability of false alarm in a group can be calculated as follows:

$$P_{fa2} = 2\times\sum_{i=0}^{\infty}(-1)^i\frac{1}{2}erfc(\frac{iQ+Q/2}{\sqrt{2}\sigma}) = \sum_{i=0}^{\infty}(-1)^i erfc(\frac{iQ+Q/2}{\sqrt{2}\sigma}), \tag{3-29}$$

where erfc(·) is the traditional complementary error function given by [AS72]

$$erfc(x) = 1 - erf(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}dt. \tag{3-30}$$

Because the tail of Gaussian distribution decays quickly, the probability that the weighted mean is shifted by the noise far away from the original value is small. Hence the probability of false alarm can be approximated by the first several *i*'s in Equation (3-29) as follows:

$$P_{fa2} \approx erfc(\frac{Q}{2\sqrt{2}\sigma}) - erfc(\frac{3Q}{2\sqrt{2}\sigma}) + erfc(\frac{5Q}{2\sqrt{2}\sigma}). \tag{3-31}$$

Finally, the sensitivity of tampering detection can be adjusted by choosing different filtering sizes and the threshold in Equation (3-16). If the threshold is preset to *d*/2, i.e. the half filter size, the larger is the filter size, the lower the sensitivity will be. Based on different application requirements, by presetting the filter dimension, the authenticator can identify tampering of various sizes, bypassing the smaller incidental distortions but detecting the bigger ones. When a high level of security is specified, no filtering is applied and this will provide the highest sensitivity to tampering. In this case, any pixel change that affects the quantization result will be identified as tampering and render the image unauthenticated.

3.4.4 Security

In this section, we discuss some security considerations of the proposed scheme. First, we focus on the security of the authentication system, especially in the case that the

authenticator is publicly available. Second, the security of the proposed scheme against local attacks is addressed.

3.4.4.1 Security of the Authentication System

Security of the proposed authentication system is ensured by using a secret key that controls the way of random permutation and authentication code generation. Even if the algorithm is publicly available, without the knowledge of the secret key, it is very difficult for an adversary to construct the correct coefficient mapping or get the authentication code when the image size and the code length are reasonably large. Therefore, the probability of successfully forging an authentic image is very small.

However, it should be noted that if the authenticator is publicly available, i.e. an adversary has unlimited access to the authenticator, the noise-dots filtering is not an optional step any more. It becomes mandatory for security reason. If the filter can be set off, an adversary can use the authenticator output to estimate the random permutation by manipulating the image pixel by pixel. Every time one single image location is tampered, the authenticator will output isolated unverified positions that come from the corresponding coefficient group. By recording and comparing these outputs, after sufficient attempts the random permutation may be reconstructed. After knowing the random mapping of locations, an adversary can design an affective attack system that manipulates the image by adjusting all the corresponding positions in a permutated group so that the weighted mean of the group keeps unchanged or remains mapping to the embedded watermark bit value. If the used parameters, such as the quantization step Q, are known as well, the used authentication code may also be revealed. Consequently, the attacker can either tamper a protected image in an undetectable way or embed a valid authentication code into other images. Therefore, when the authenticator can be accessed unlimitedly by an adversary, in order to ensure the system security a noise-dots filter with reasonable size must be always present.

3.4.4.2 Security against Local Attacks

Since the watermark is not embedded in a local way, the proposed scheme is intrinsically secure to local attacks that are usually mounted in a block-based way, such as copy and paste and block swapping. The most successful local attack is the vector quantization (VQ) attack, also known as collage attack [CMB01], which is to construct a counterfeit image by choosing authentic blocks from different watermarked images. The premise of successfully mounting a VQ attack is that an adversary has access to a set of images that are watermarked using the same secret key. The adversary replaces each block of the unwatermarked image with the most similar block selected from the set of watermarked images. In this way, a counterfeit of authentic image can be constructed. VQ attack is particularly applicable to block-wise independent embedding approaches [HM00], in which the authentication of each block depends only on the content of the block itself. The proposed scheme in this chapter is immune to VQ attacks, because the coefficients of a group are selected randomly over the whole image. Without the knowledge of the way of random permutation, it is very difficult for the adversary to reveal the relationship among the coefficients and therefore the replacement of similar pixels/blocks can not be performed. Any local pixel replacement will cause the weighted means of a number of groups changed, which will render some of these affected groups unverified. Due to the variation among these groups, it's very hard to find a way to perform a local manipulation with the caused changes in all the affected groups undetected.

3.5 Multi-Resolution Authentication

As mentioned in Section 3.4.3, various modifications will cause changes of different frequency bands. For example, some edge manipulations will change high frequencies of the image, while other kinds of tampering may modify only middle or low frequencies. Hence, checking all the frequency bands will certainly decrease the miss probability of tamper detection.

Based on the proposed watermarking scheme, a multi-resolution authentication scheme can be designed. Independent authentication codes can be embedded in each wavelet decomposition level of the image. Each embedded code can achieve integrity verification with the corresponding resolution. The watermark in the high decomposition has better robustness against image distortions but lower localization capability of content tampering. On the contrary, the watermark in the low decomposition level is able to localize the tampering with higher resolution but easier to be impaired by the incidental distortions caused by common image processing. Note that by embedding multiple watermarks in all the wavelet levels, the watermarked image's quality will be more degraded because more modifications are introduced. The final image quality can be evaluated by the total modification amount that can be obtained by applying Equation (3-24) to every wavelet level embedding respectively.

3.6 Experimental Results

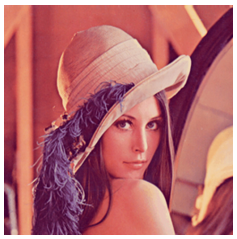
To evaluate the proposed watermarking scheme, 1086 images are used in our tests that are of various sizes and most of them are taken by three different digital cameras. Some standard test images that are commonly used in image processing, such as Gold Hill, Lena and Peppers, are also included in the test set. The used test images have different content properties such as high textured, few textured and mixed. A variety of sample images are shown in Figure 3-11. We evaluate the proposed scheme in aspects of image quality after watermarking embedding, tamper localization capability and robustness against incidental distortions. We shall represent these experimental results in the following subsections respectively.

3.6.1 Image Quality Test

First, we use the image "Gold Hill" of 720×576 as an example to test the watermarked image quality. The original and the watermarked images are shown in Figure 3-12. The watermark is embedded in the first level of wavelet decomposition and we set the quantization step Q=6 and group size g=12. In the embedding process, Equation (3-9),

Gold Hill

Peppers



Lena

Airplane

Bonn

Benz

Test Box

Figure 3-11 Sample images from the image test set

i.e. the first coefficient-updating method, is used to update the wavelet coefficients. From Figure 3-12 (b), we can see the embedded watermark is completely imperceptible. The PSNR of the watermarked image is as high as 49.05dB. This value is very close to the expected theoretical value 49.38dB calculated by Equation (3-25).

Second, we test the image quality with all the images in the test set. In the following tests, if not specified, Equation (3-9) is used to update the wavelet coefficients in the embedding process. Table 3-3 lists the PSNR values of the sample test images shown in Figure 3-11 for different embedding parameters. As shown in Table 3-3, with the same embedding parameters the quality results of different images are very close to the image "Gold Hill". For most embedding parameter sets, the watermarks embedded in all the images are imperceptible to human observers under normal viewing condition, except "r=1, Q=16, g=3" and "r=2, Q=16, g=3". In these two cases, the watermark embedded in less textured images like "Benz" becomes slightly noticeable.

(a)

(b)

Figure 3-12 Original and watermarked images: (a) Original image of "Gold Hill" of size 720×576, (b) Watermarked image with r=1, Q=6 and g=12, PSNR=49.05dB.

Table 3-3 Quality of different watermarked images in PSNR (dB)

Image Name	Gold Hill	Peppers	Lena	Airplane	Bonn	Benz	Test Box
Image Size	720×576	512×512	512×512	512×512	720×576	640×480	1024×1024
r=1,Q=6,g=3	43.52	43.57	44.78	44.02	43.80	43.95	44.02
r=1,Q=6,g=12	49.05	49.06	50.07	49.22	49.20	49.22	49.18
r=1,Q=16,g=3	36.19	36.45	36.38	36.41	35.95	36.04	36.26
r=1,Q=16,g=12	41.18	41.44	42.74	41.60	41.09	41.43	41.36
r=2,Q=6,g=3	49.13	49.36	49.51	49.38	49.21	49.51	49.33
r=2,Q=6,g=12	54.97	55.03	54.89	54.92	54.95	54.72	54.97
r=2,Q=16,g=3	41.45	42.01	42.00	41.67	41.39	41.83	41.60
r=2,Q=16,g=12	46.99	47.17	47.15	46.90	46.89	47.02	46.98

To evaluate the analysis of image quality, we compare the experimental results with the analytic results calculated by Equation (3-25) for different embedding parameters. Figure 3-13 plots both the analytic PSNR values of watermarked images and the experimental values. Each experimental value is the average of the PSNR values of the total 1086 test images. The watermark is respectively embedded in the first (r=1) and second (r=2) wavelet levels with different quantization steps Q. As can be seen from Figure 3-13, the analytic and experimental results conform very well. Figure 3-13 also indicates that a larger quantization step Q will degrade the watermarked image quality more. In addition, with the same g and Q embedding the watermark into the higher wavelet level will render better quality of watermarked images.

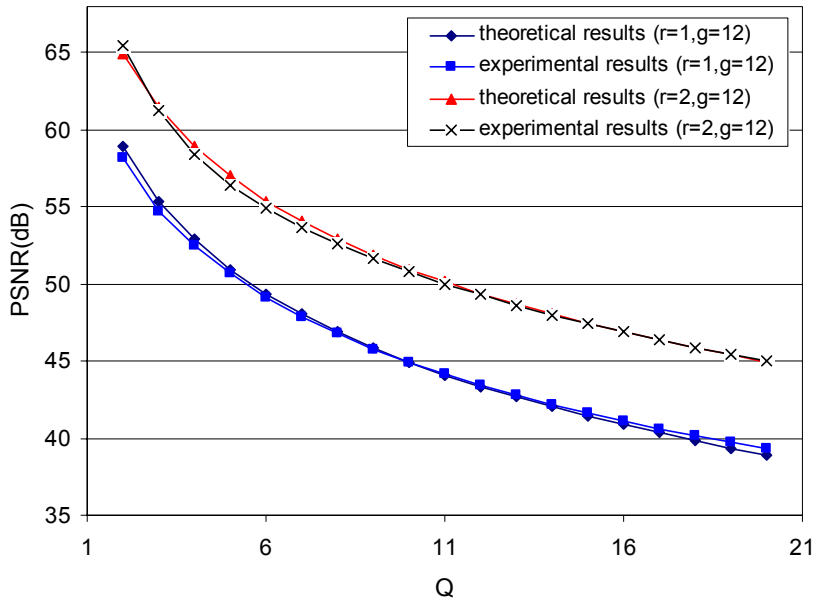


Figure 3-13 Theoretical and experimental results of watermarked image quality for different embedding parameters. The experimental PSNR values are the average of the 1086 test images.

As mentioned in Section 3.4.1, the group size g will also affect the quality of the watermarked image since it determines the watermark payload. To test the effect of group size, we embed the watermark in the same wavelet level with different group sizes. Figure 3-14 plots the PSNR values of watermarked images, in which the watermark is embedded in the first wavelet level (r=1) with various group sizes. Each value is the average of the PSNR values of the total 1086 test images. The test results demonstrate the analytic conclusion that a larger g renders better image quality because it decreases the watermark payload.

Finally, we compare the quality of watermarked images by different coefficient update methods by embedding the same watermark into all the test images using Method 1 and Method 2 respectively. Table 3-4 lists the results of PSNR and BER (Bit Error Rate) of watermark detection of two less textured images “Peppers” and “Benz” for different embedding parameters. As can be seen, the quality of watermarked images by Method2 is overall better than that by Method1. This quality improvement is obtained by more

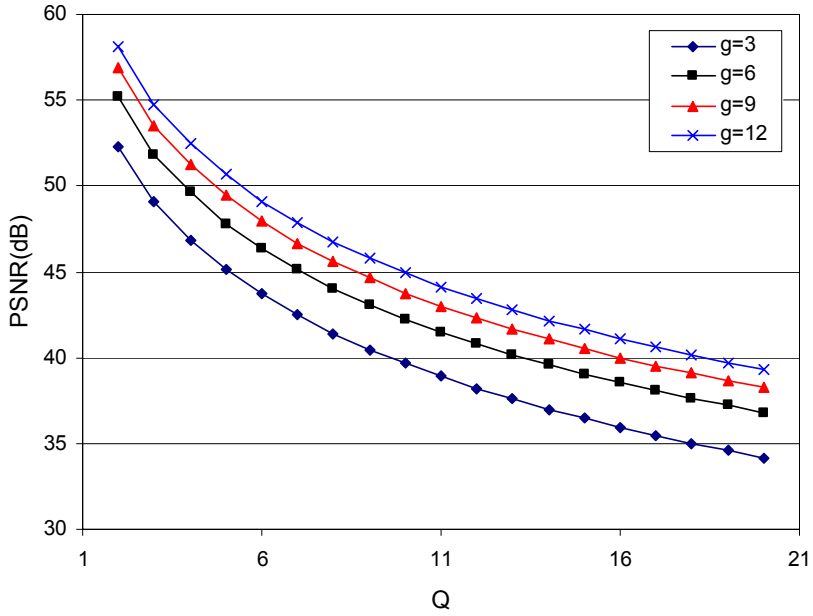


Figure 3-14 Watermarked image quality versus quantization step Q for different group size *g*. The watermark is embedded in the first wavelet level (*r*=1). The PSNR values are the average of the 1086 test images.

unsuccessful or incomplete embeddings, in which case the magnitudes of some small coefficients can not bear the assigned portion of the modification amount. Therefore, in these cases the corresponding weighted mean values are not updated to the expected value. Such incomplete embedding happens more often in less textured images, because less textured images contain less middle and high frequency components. This also explains why the PSNRs of watermarked images by Method 2 are larger than the analytic values. As a result, due to the unsuccessful embeddings, more watermark detection errors occur in the detection process. As shown in Table 3-4, the BERs of Method2 are overall higher than those of Method1.

In addition to the results listed in Table 3-4, we also compare the PSNR and BER results of the two methods by using the total 1086 test images. Figure 3-15 plots the PSNR values of the images that are watermarked by using Method 1 and Method 2 respectively for different embedding parameters. All the PSNR values are the average

of the results of all the 1086 test images. As can be seen, the PSNR values of the images watermarked by Method 2 are overall higher than both Method 1 and the analytic results. Figure 3-16 plots the corresponding BER distribution of the two methods over the whole test image set. For Method 2, when the quantization step is reduced to 6, the BER will be significantly increased, which reveals that small embedding strength will cause more unsuccessful embeddings. Therefore, when a small quantization step is specified, Method 1 is more suitable for the proposed watermarking scheme.

Table 3-4 Comparison of image quality and BER for the different wavelet coefficient update methods in the embedding process

Image Name	Peppers				Benz			
	Mehod1	BER	Method2	BER	Mehod1	BER	Method2	BER
r=1,Q=6,g=3	43.57	2.9×10^{-4}	46.32	3.8×10^{-3}	43.95	1.6×10^{-3}	45.83	0.14
r=1,Q=6,g=12	49.06	1.0×10^{-3}	56.75	1.5×10^{-2}	49.22	4.0×10^{-3}	52.54	7.9×10^{-3}
r=1,Q=16,g=3	36.45	4.6×10^{-5}	39.36	1.1×10^{-2}	36.04	1.1×10^{-3}	38.66	0.16
r=1,Q=16,g=12	41.44	6.1×10^{-5}	48.10	0	41.43	2.4×10^{-3}	44.83	1.1×10^{-2}
r=2,Q=6,g=3	49.36	2.4×10^{-4}	51.77	2.7×10^{-3}	49.51	9.9×10^{-4}	51.00	8.8×10^{-2}
r=2,Q=6,g=12	55.03	1.5×10^{-3}	62.56	0.17	54.72	2.7×10^{-3}	58.56	9.0×10^{-3}
r=2,Q=16,g=3	42.01	0	44.42	5.7×10^{-3}	41.83	6.3×10^{-4}	43.76	0.10
r=2,Q=16,g=12	47.17	2.4×10^{-4}	53.31	0	47.02	1.9×10^{-3}	50.66	2.1×10^{-3}

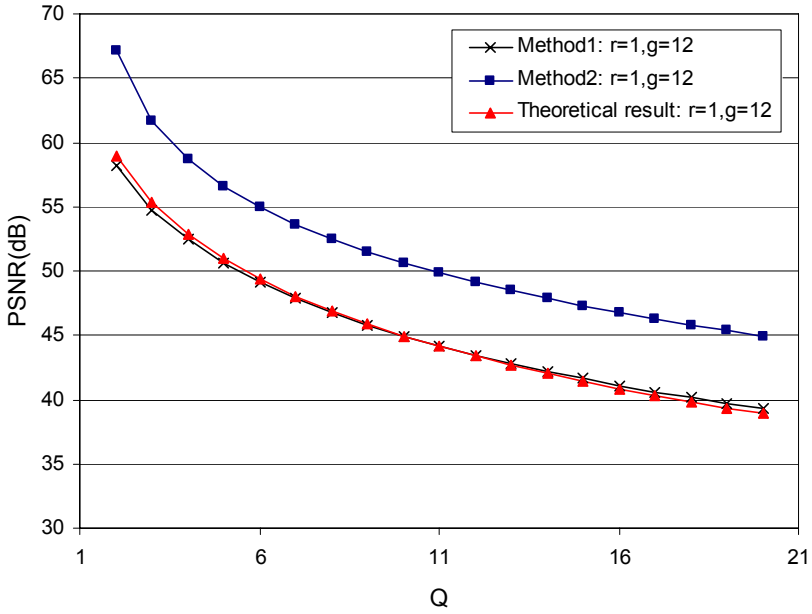


Figure 3-15 Comparison of PSNR of watermarked images by two coefficient update methods for different embedding parameters. The PSNR values are the average of the 1086 test images.

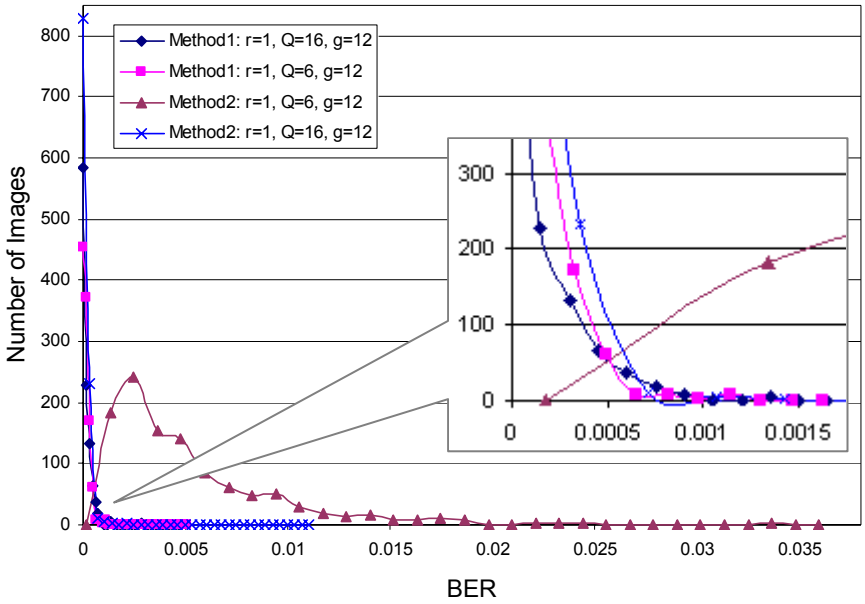


Figure 3-16 Comparison of BER distribution of two coefficient update methods for different embedding parameters.

3.6.2 Tamper Localization Capability Test

In the following experiments, we test the localization capability of the proposed watermarking scheme. First, we set r=1, Q=6 and g=12 in the watermark embedding process. Figure 3-17 shows a tampered image of "Gold Hill", in which the man on the street is removed. The yellow ellipse indicates the tampering position. Figure 3-18 gives the authentication results. In Figure 3-18 (a), the authentication result is depicted by white dots. Every dot corresponds to a potential unverified position in the matrix $\boldsymbol{V}$, which are from the unverified groups in the randomly permutated sequence S'. Though those white dots randomly scatter over the whole image, we can clearly see that the density of the white dots in the tampered region is much higher than that in other regions where the dots are actually isolated. Therefore, the tampered region can be easily recognized. The authentication result not only localizes the position of the tampered region but also depicts the tampered region's size and shape. Since the watermark is embedded in the first wavelet level (r=1), the size of a white dot is 2×2 pixels, namely, the maximal detection resolution is 2×2 pixels.

Then we refine the authentication result by filtering out the *noise dots* using Equation (3-16). The filter size d and the threshold T are set to 5 and 4 respectively. The refined authentication result is shown in Figure 3-18 (b). The localized tampered regions are depicted in white color. In the refined result, all the *noise dots* are removed and the holes in the tampered regions are also filled by the filtering process. As can be seen in Figure 3-18 (b), the localized tampered region clearly reveals the size and shape of the tampering.

Second, we set r=2, Q=6 and g=12 and embed the watermark into the original image again with the new parameters. Then the same manipulation is done on the watermarked image as in the first test. The authentication result is shown in Figure 3-19, in which the tampered region is correctly localized, depicted in white color. Since in this case the watermark is embedded in the second wavelet level (r=2), as shown in Figure 3-19 (a), the size of a white dot is 4×4 pixels. Therefore, the maximal detection resolution decreases to 4×4 pixels accordingly. The refined result is shown in Figure

3-19 (b). As can be seen, the tampering position and shape are depicted by the white dots in a coarser way than that in the first test. This result demonstrates what we discussed in the previous sections, i.e. embedding in the higher wavelet levels will lower the detection resolution. However, by sacrificing the accuracy of the tamper localization, we can obtain better image quality as already demonstrated in the previous test results, and more robustness against incidental distortions like lossy compression, which we will present in the following tests.

Figure 3-17 A tampered image: the man on the street is removed. The altered position is indicated by an ellipse.

(a)

(b)

Figure 3-18 Image authentication results with r=1: (a) Authentication result with the noise dots (in white), (b) Refined authentication result (after filtering out the noise-dots).

(a)

(b)

Figure 3-19 Image authentication results with r=2: (a) Authentication result with the noise dots (in white), (b) Refined authentication result (after filtering out the noise-dots).

3.6.3 Robustness against JPEG Compression

JPEG compression is the most frequent processing in many applications due to the popularity of the JPEG storage format. In this subsection, we test the watermark robustness against incidental distortions caused by the JPEG compression. The robustness of the embedded watermark depends on the three embedding parameters: the wavelet level r, the group size g and the quantization step Q. For different embedding parameters, the performance of the watermark detector after JPEG compression is shown in Figure 3-20, Figure 3-21 and Figure 3-22. The watermark is embedded into the original image "Gold Hill" with different parameters respectively. And then the watermarked images are compressed with different JPEG quality factors, varying from 10 to 100. Since JPEG is performs lossy compression, even with the quality factor 100 it still causes image quality degradation.

As expected, embedding the watermark in a higher wavelet level achieves higher robustness against JPEG compression than in lower levels, because higher wavelet levels contain the middle and low frequency parts of the image which are less influenced by the JPEG compression. As shown in Figure 3-20, the watermark embedded in the first wavelet level is only robust to JPEG compression with quality factor 100. When embedded in the second or third level, the watermark becomes much more robust as can be seen in Figure 3-21 and Figure 3-22, which can be robust against JPEG compression with quality factor 80 or even as low as 50.

In addition, the quantization step determines the watermark embedding strength and a larger Q makes the watermark more tolerant to distortions. As can be seen from the figures, for every wavelet level, a larger Q achieves a lower watermark detector error rate. However, a larger Q will also introduce more image modifications in the embedding. Furthermore, a small group size g also increases the watermark robustness. This is because the JPEG compression introduces distortions to every wavelet coefficient; fewer group members accumulate fewer errors. However, this robustness improvement is achieved at the cost of a higher watermark payload, which will lower the watermarked image's quality.

Overall, as can be seen from Figure 3-20, Figure 3-21 and Figure 3-22, higher wavelet level *r*, larger quantization step *Q* and smaller group size *g* will enhance the watermark robustness against the distortions caused by JPEG compression while they will decrease the resolution of tampering localization or degrade the watermarked image's quality.

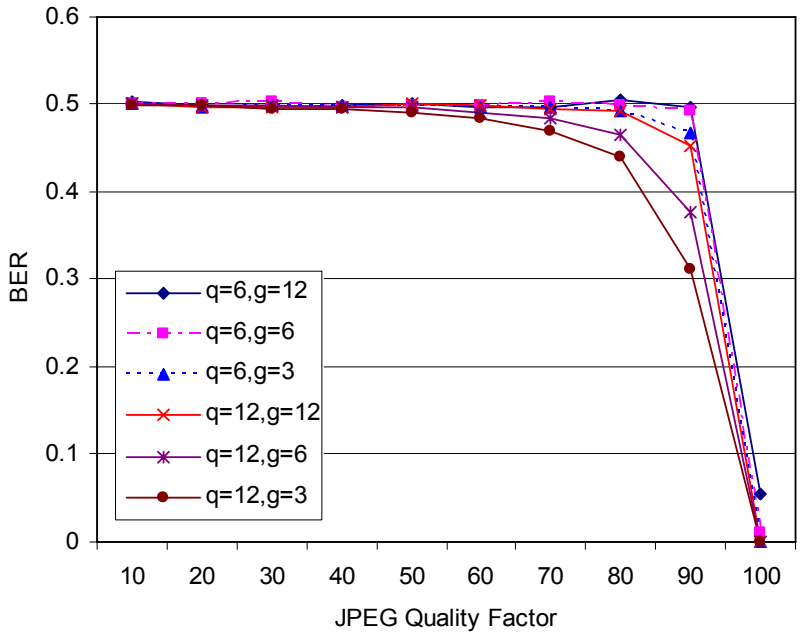


Figure 3-20 Watermark detector performance after JPEG compression for different quantization steps and group sizes. The watermark is embedded in the first wavelet level (r=1).

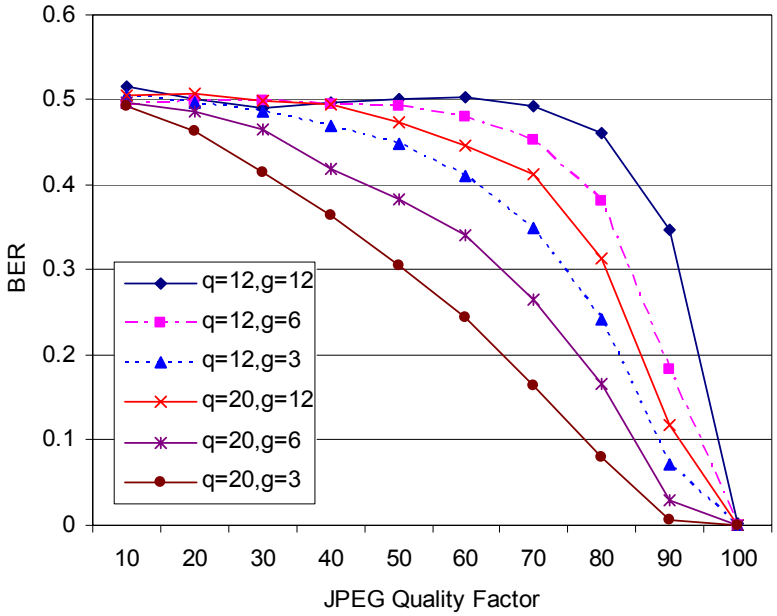


Figure 3-21 Watermark detector performance after JPEG compression for different quantization steps and group sizes. The watermark is embedded in the second wavelet level (r=2).

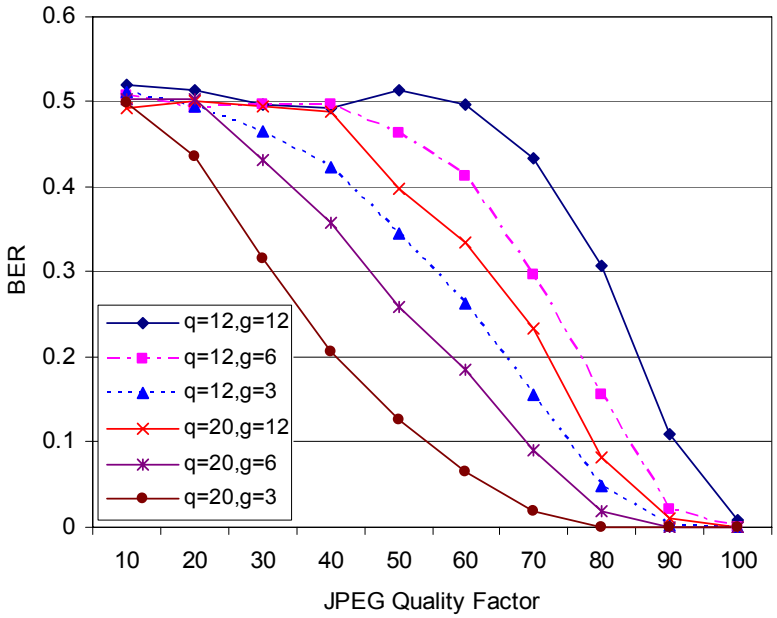


Figure 3-22 Watermark detector performance after JPEG compression for different quantization steps and group sizes. The watermark is embedded in the third wavelet level (r=3).

3.7 Conclusion

In this chapter, we first addressed the existing problems and challenges in digital watermarking for image authentication, and then we proposed a novel semi-fragile watermarking scheme which can detect and localize the tampered regions in the image. The proposed scheme significantly improves the resolution of tampering detection with low watermark payload by introducing a random permutation procedure in the wavelet domain. The proposed watermarking scheme utilizes every watermark bit to monitor random image positions instead of a local image block. Thanks to the random wavelet coefficient grouping, the scheme is intrinsically secure to local attacks. Because the embedded watermark is randomly distributed into the suitable embedding coefficients, the security of the embedding mechanism is also improved. With different embedding parameters, the embedded watermarks can survive moderate JPEG compression to some extent. Scalable sensitivity of tampering detection is enabled in the authentication process by presetting the noise filter size.

Chapter 4 Synthetic Image Authentication

4.1 Introduction

Besides images taken from the natural world, there are also lots of synthetic images widely used in various applications, such as digital maps, document images, engineering drawings, computer generated graphics, scanned documents and handwritten signatures, and so forth. For example, digital maps are now widely used in different Geographic Information Systems (GIS) on the Internet and handheld devices. In addition, all kinds of important documents, such as legal documents, financial instruments, certificates and insurance information, have been digitalized and stored. Due to the wide popularity, the authentication of synthetic images against tampering and forgery is becoming a great concern.

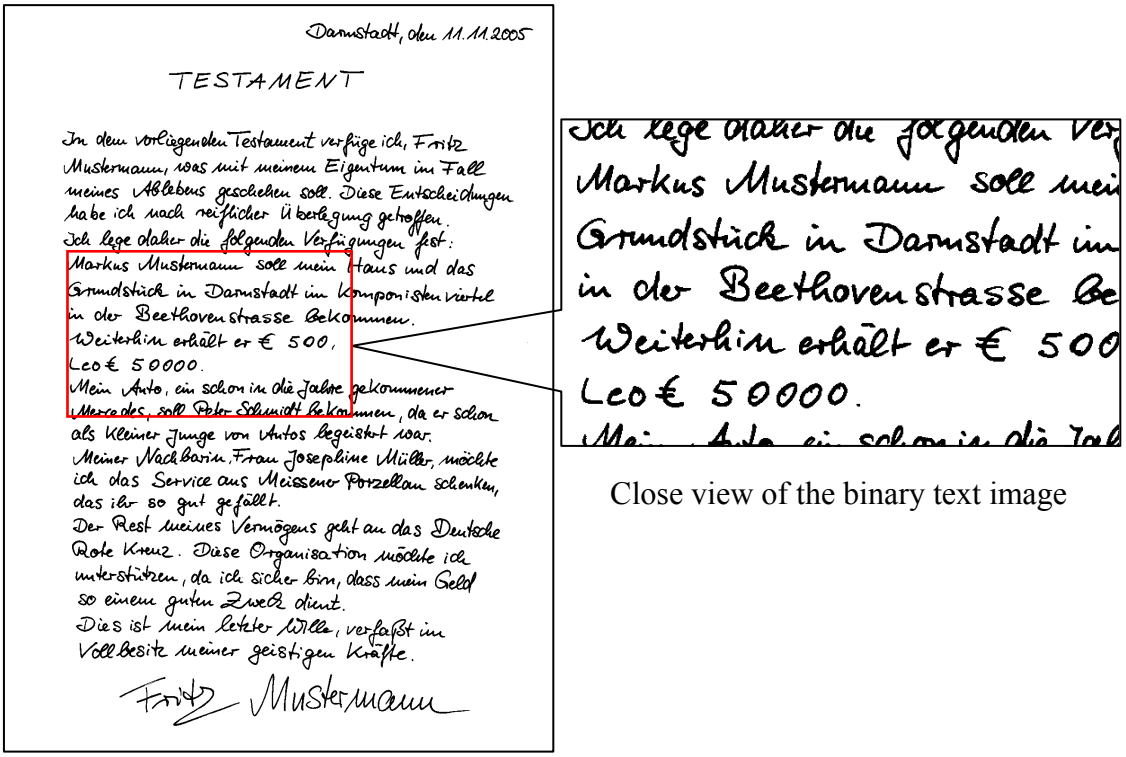
In order to differentiate from graphics, the term *synthetic image* in this thesis refers to all the simple images that are represented by a few number of color/gray values. The extreme case is binary images that contain only two colors: black and white. Compared with continuous-tone natural images, synthetic images have much fewer colors and no complex texture variation. Unlike natural images, in which pixel values vary in a wide range, the pixels in the synthetic images only take on a limited number of values. Moreover, in synthetic images, the color and brightness usually change abruptly from

one value to another without any transition, which results in sharp edges. In addition, there are usually large homogenous regions in synthetic images. In these regions, there is only one uniform gray level or color. Hence arbitrarily changing pixels on a synthetic image will cause very visible artifacts. Table 4-1 lists major differences between natural images and synthetic images.

Table 4-1 Comparison of the natural and synthetic images

Characteristics	**Natural Images**	**Synthetic Images**
Color	Continuous-tone, plenty of colors (True-color)	A few of colors, limited number of pixel values
Texture	Complex texture variation	Simple texture, plenty of flat regions
Edge	Mild edges with gradual brightness and color transition	Sharp edges with abrupt brightness and color change
Storage Format	True-color formats, e.g. JPEG, TIFF, BMP	Indexed color or binary formats, e.g. GIF, PNG, TIFF

Figure 4-1 and Figure 4-2 give two examples of synthetic images. Figure 4-1 shows a text image that is a typical kind of synthetic image. The sample image is a binary bitmap that contains only black and white colors and has large blank margins. When stored in an indexed color format, the color palette will contain only two entries. In addition, because the pixels take on only two possible values, a binary image is also commonly stored as a bitmap with a color depth of 2. Figure 4-2 presents a digital map as an example of a color synthetic image. The sample map does not look like a very simple image as it contains many lines, curves and symbols in different colors. In this map, however, there are actually only seven distinct colors, as shown in the color palette on the right. Furthermore, lots of regions in the map are filled with uniform colors and contain no texture at all. One magnified smooth part is shown below in Figure 4-2, in which most of the area is filled with pure white color. Note that synthetic



Close view of the binary text image

Figure 4-1 Synthetic image example A: binary text image

images, such as digital maps, can also be stored in vector image formats. In this chapter, we only consider synthetic images stored as bitmap images.

Because of its unique property, invisibly embedding data in a synthetic image becomes a more challenging task. On the other hand, unfortunately, due to the simplicity of the content, it is much easier for an adversary to manipulate a synthetic image than a continuous-tone natural image. An adversary even does not need any powerful image-editing software like Adobe Photoshop because a simple image modification tool is enough to make a perfect forgery without leaving any noticeable traces on the original synthetic image.

Although a variety of watermarking schemes for image authentication have been proposed, most of them are developed for color and grayscale natural images and can

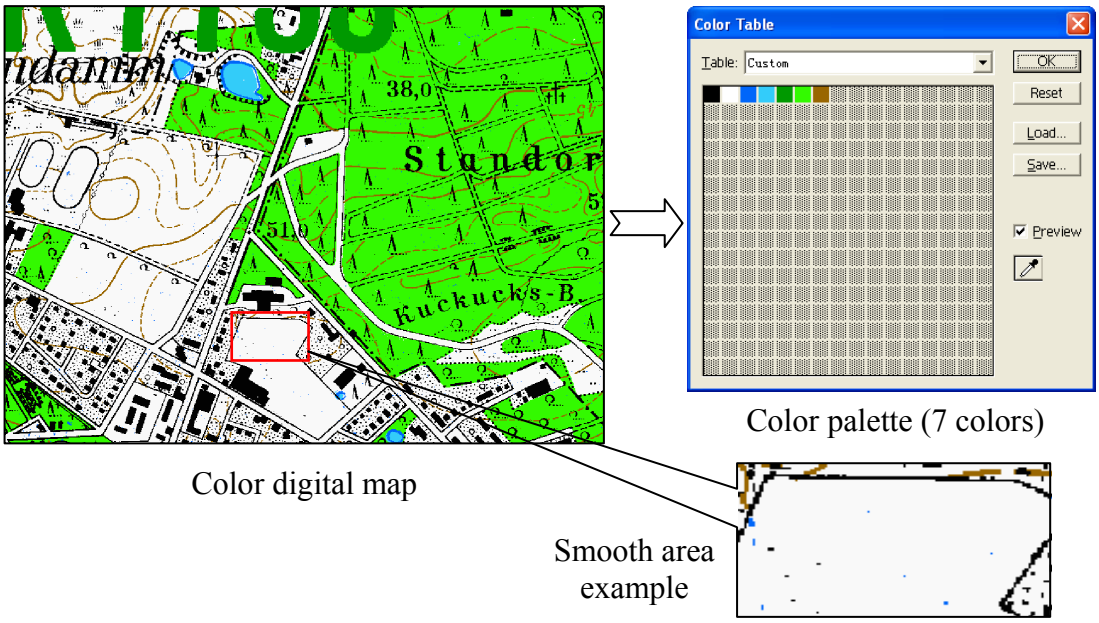


Figure 4-2 Synthetic image example B: color digital map

not be applied to the synthetic images directly. In those schemes, the watermark information is commonly embedded by changing the least significant bits (LSB) of the pixel values [W98][FGB00][F02] or slightly modifying the transform coefficients [WL98][F99][WKBC02]. Since synthetic images contain plenty of sharp edges and smooth areas, such kinds of modifications will either introduce visible artifacts or significantly decrease the reliability of the embedded watermark because of the weak embedding strength.

Another problem caused by common watermarking schemes for natural images is that new colors will be introduced into the cover image. As synthetic images only contain a limited number of colors, they are usually stored in indexed color formats (e.g. GIF, Graphics Interchange Format) instead of true the color formats (e.g. JPEG). The indexed color formats use a color palette to indicate different used colors. The number of colors that an indexed color format can store is usually limited and depends on the size of the used color palette. For example, the GIF format uses a color palette that can

contain 256 distinct colors. The classical watermarking approaches for natural images embed the watermark by slightly changing the pixel values, which will inevitably introduce new pixel values, i.e. new colors. It becomes even worse for the approaches that embed data in the transform domains by modifying the frequency coefficients, e.g. in DCT or DWT domains, because in this case it is very hard to predict and control the number of the introduced new colors. Because the introduction of the new colors will change the entries of the original image palette, from the compatibility point of view, it is very undesirable to introduce additional pixel values to the synthetic images in most of the applications. Moreover, when so many additional colors are introduced that the total number of the colors exceeds the palette's capacity, it becomes impossible to store the watermarked image in the original format.

Therefore, for synthetic image authentication, specific watermarking schemes must be designed to handle the above-mentioned requirements. Generally speaking, a watermarking scheme for synthetic image authentication should satisfy the following listed requirements:

1. Watermark transparency: no noticeable artifacts should be introduced, i.e. high image quality must be achieved after watermark embedding.
2. Format compatibility: no additional color should be introduced in the watermarked image in order to keep the color palette intact.
3. Tamper localization capability: the authenticator should be able to localize the tampered region with a resolution as high as possible.
4. Recovery capability: it is a very desirable feature that the authenticator is able to recover the original content in tampered regions.
5. Blind detection: the embedded watermark can be extracted without referring to the original image.

In this chapter, we propose a novel watermarking scheme to authenticate the content integrity of synthetic images. In the proposed scheme, stricter rules of modifying pixels

are specified compared to the other existing schemes so that the quality of the watermark image gets improved. Moreover, in the embedding process, no additional pixel value will be introduced. A random permutation process is applied to the whole image before embedding the watermark bits. The watermark information is embedded in the permuted image domain and every embedded watermark bit is utilized to monitor a group of pixels so that all pixels of the image instead of blocks are identified by much less watermark bits. Combining random permutation and statistical tamper detection, the proposed scheme achieves pixel-wise tamper localization capability. We present a new embedding strategy that enables the recovery capability of the authentication system. Hence, in the authentication process, not only can the proposed scheme localize the tampered area but it also can recover the removed content and identify the forged parts. Experimental results demonstrate the capability of the proposed scheme to localize and recover tampered areas in watermarked images. The proposed scheme can be applied to various kinds of synthetic images, including binary images or images with few colors.

The organization of this chapter is as follows. Firstly, in Section 4.2, we retrospect the previous work related to the authentication and data hiding for the synthetic images and address the unsolved problems and challenges of synthetic image authentication. Then, in Section 4.3, we introduce the proposed watermarking scheme, including the watermark embedding and retrieval processes. The authentication process is presented in Section 4.4. Afterwards, we analyze the proposed scheme's performance and security issues in Section 4.5. Experimental results are given in Section 4.6. In Section 4.7, we discuss the possible extension of the proposed embedding strategy. Finally, we conclude the chapter in Section 4.8.

4.2 Previous Work

In the literature, a variety of digital watermarking and data hiding techniques have been proposed for embedding data in synthetic images, especially in binary images [CWMA01]. Basically, most of the proposed schemes can be classified into two

categories according to the way of modifying the cover image to embed the watermark information.

1. The first category includes algorithms that embed the watermark information by modifying certain characteristics of some pixel groups, such as the position of the text line or word, the spacing of words or characters, the thickness of stokes, etc.

2. The second category contains techniques that embed the watermark information by modifying individual pixels to certain desired values according to the data to be embedded. Those pixels can be chosen either randomly or according to some visual impact measures.

Some early works of text document watermarking fall into the first category. In [ML97][LMB95][LM98][LML98], the watermark bit is embedded by slightly shifting the relative position of the text lines or word spacing or the combination. For example, a text line can be moved up to encode a "1" or down to encode a "0". Similarly a word can be moved horizontally to change the spacing to encode the watermark bit. Some varieties of this embedding method were also proposed. In [BG96], Brassil proposed to embed data by modifying the height of a bounding box that encloses a group of words. This approach achieves a higher watermark capacity than the text line or word shifting method. Instead of using inter-word spacing, Chotikakamthorn [C98][C99] employed character-to-character spacing to embed data in, so that it can be applied to text documents in non-English languages that do not have spaces to separate words such as Chinese, Japanese and Thai. In [HY01], the inter-word spaces are slightly modified so that the spaces across different lines of a text act as sampling points of sine waves. The watermark is then embedded into these sine waves. By embedding the watermark information in both horizontal and vertical directions, the proposed approach achieves high robustness against interference. Another embedding method in this category is to modify the thickness of character stokes. In [AM99], the average width of the horizontal stokes of characters is used as a feature to embed the information. Two operations "make fat" and "make thin" are applied according to the desired watermark

bit, which increases or decreases the selected stoke widths. In general, all of the above-mentioned algorithms are only applicable to document images with formatted text, and are not suited for other generic binary or synthetic images such as drawings, maps, etc. Furthermore, all of these approaches were proposed for data hiding or copyright protection and can not be directly applied for authentication purposes.

In the second category, some watermarking algorithms modify individual pixels randomly without taking into account the visual impact of such modification. In [FA00a], Fu et al. proposed a simple embedding method called DHST (Data Hiding by Self-Toggling). A set of random locations in the image are selected to embed the data. At one of the selected locations, the pixel is forced to be black or white according to the data to be embedded. The basic DHST technique was subsequently improved by the techniques DHPT (Data Hiding by Pair Toggling) and DHSPT (Data Hiding by Smart Pair Toggling) proposed by the same author in [FA00a] and [FA00b]. In [FA01], an algorithm called IS (Intensity Selection) was proposed to select the best embedding locations so that the visual quality of the watermarked image was significantly enhanced. In [KA03] and [KA04], an authentication watermarking scheme was proposed by using the DHST embedding technique. The scheme was claimed to be also applicable to generic binary images, provided that the number of embedded bits is far fewer than the number of pixels in the host image. However, when applied to text documents, annoying salt-and-pepper noise will be introduced, which is not acceptable in most of the applications. In [PCT00] and [TCP02], Tseng et al. proposed a block-wise data hiding technique that modifies at most two pixels in a block with m pixels to embed $\lfloor \log_2(m+1) \rfloor$ bits. This technique was improved by Chang et al. in [CTL05] to embed the same amount of bits by modifying only one pixel at most. In these techniques, although the number of the pixels to be modified is constrained in each block, there is no control on the quality of the image after the modification because the pixels to be modified are selected randomly. In summary, since the above-mentioned techniques do not take into account the visual impact of pixel toggling, visible

distortions will appear when they are directly applied to generic binary or other synthetic images.

In order to improve the quality of watermarked images, some other algorithms have been proposed to embed the data by selectively modifying pixels according to the visual impact instead of randomly toggling pixels. The above-mentioned embedding techniques proposed in [PCT00][TCP02] were improved with respect to visual quality in [TP01] by imposing a constraint that every pixel to be modified must be on the boundary. With the improved visual quality, the data hiding capacity was decreased to embed $\lfloor \log_2(m+1) \rfloor - 1$ bits in a block with m pixels. This improvement, however, did not give further analysis and comparison on the various visual impacts among different types of boundary pixels. In order to identify the pixels that cause the least noticeable artifacts, template ranking has been widely used in many papers. The method proposed in [MWM01] uses the outer boundary of a character to embed data. A set of pairs of five-pixel long boundary patterns are identified and used to embed data. Every pair of patterns has the duality property that changing the center pixel of one pattern would result in the other pattern. This property allows easy extraction of the hidden data without referring to the original image. In [PWP02], Pan et al. proposed a data hiding method for few-color images using prioritized pattern matching. In order to increase the data hiding capacity, the concept of "Supblock", which can be decomposed into several overlapping subblocks, was used to increase the number of the embeddable blocks. However, flipping the central pixels in some used patterns may cause visible distortions such as a hole in a straight line. Kim et al. [KQ04] proposed a cryptography-based authentication technique for binary images, which can be applied to generic binary images with good visual quality and can be used in conjunction with secret-key or public/private-key ciphers. A variation of the proposed algorithm was also presented that can locate the tampering even with image cropping. However, the public-key scheme is not secure against the parity attack [K05]. This problem was solved by the technique proposed in [K05], named AWTC (Authentication Watermarking by Template ranking with symmetrical Central pixel) that is immune to the parity attack.

In [WTL00] and [WL04], Wu et al. proposed a quantitative analysis method of visual distortion. The term "flippability score" is used to indicate the visual impact of flipping a pixel, which is computed by analyzing the smoothness and connectivity of a 3×3 block centered at the pixel. Pixels with large scores will be flipped with high priority in the embedding process because they will cause less noticeable artifacts. In Wu's approach, odd-even embedding method was applied. The cover image is divided into blocks and one bit is inserted in every block by forcing the number of the black pixels in the block to be odd or even. If the block has the desired parity, it is left intact. Otherwise, the pixel with the highest "flippability score" will be flipped. Random shuffling technique is employed to equalize the uneven watermark capacity over the image. In the recent work [YK07], Yang et al. proposed another criterion to assess the pixel flippability called "connectivity-preserving criterion". Based on this criterion, the center pixel in a 3×3 block is considered as flippable if the connectivity between pixels in the block will not change before and after flipping. Moreover, the flippability of a pixel will not be changed by the embedding process, so it can be identified again in the detection process without referring to the original image. To increase the watermark capacity, interlaced block partitions may be used and the uneven watermark capacity problem is handled by embedding the data only in those "embeddable" blocks.

In addition, a few other methods have also been proposed to hide data in simple images while keeping good image quality. In [LWKS02], Lu et al. proposed a Distance-Reciprocal Distortion Measure (DRDM) to assess the quality of binary document images, which has much better correlation with the human visual perception than PSNR (Peak Signal-to-Noise Ratio). Subsequently, the DRDM technique was applied to data hiding and the authentication of binary document images in [LKC03]. When necessary, the pixels with lowest DRDM value in every block will be flipped to embed the desired bit. 2-D shifting is employed before odd-even embedding process to provide security against tampering. A denoise-pattern based embedding method was proposed in [YK04], in which eight denoise-patterns were identified to select suitable pixels for flipping. The embedding process will smooth out the original image, so the quality of watermarked image may be enhanced. One main drawback of this method is that the

watermark detector needs the location map of the embedding to extract the embedded bits.

Among the above-mentioned schemes, most of them are targeted only for data hiding instead of authentication except those proposed in [KA03][KA04][K05][KQ04] [LKC03][WL04] and [YK07]. The primary concern of data hiding is to increase the embedding capacity, while for image authentication tamper detection and security are of most concerns. On the contrary, for image authentication, when the image integrity gets already ensured, low embedding capacity is more desirable because embedding more data will degrade the image fidelity. In addition, tamper localization and recovery capability are very important features in the applications of image content authentication. However, most of the proposed authentication schemes can neither localize the tampered area nor recover the original image content. In the schemes proposed in [KA03][KA04][WL04] and [YK07], a binary logo is used as a visual authentication watermark. The image authenticity is determined based on the integrity of the extracted logo image. If the extracted logo is identical to the original version, the image is considered as authentic. Once the image is manipulated, the extracted logo will be destroyed and become a random pattern. In all of these schemes, only a binary output can be provided and none of them can localize the position of tampered regions. The secret-key authentication scheme proposed in [KQ04] is designed to be able to localize the tampered area, but the localization resolution is bounded by the sub-image size and therefore the localization result is quite rough. It can identify the unverified regions only with a resolution as low as 128×128. Hence, no accurate tampered position can be provided. The scheme proposed in [LKC03] may achieve more accurate tamper localization results, but it depends on the way of 2-D shifting and the localization capability was not explicitly addressed in the paper. Furthermore, none of the previously proposed schemes in the literature has achieved the capability of recovering the original content in synthetic image authentication. In other words, the authenticators of those schemes can only verify whether (and where in a few cases) the image content has been manipulated or not but are not able to estimate or even recover the original version of the manipulated parts.

4.3 Proposed Scheme

In our proposed scheme, we follow the approach of the second category to embed the watermark. Suitable individual pixels are selected to be modified to embed the desired watermark bits. Stricter rules are imposed on choosing flippable pixels in order to improve the quality of watermarked image. In our scheme, we focus on solving three main issues for synthetic image authentication: (1) how to achieve high fidelity of the watermarked image, (2) how to achieve a high resolution tamper localization while keeping the watermark payload reasonably low, i.e. how to efficiently use every embedded watermark bit to monitor as many as possible pixels, (3) how to recover the original pixel value from the tampered image regions.

As illustrated in Figure 4-3, the proposed watermarking system consists of two parts. The first part includes the pixel classification, random permutation and the watermark embedding process that embeds the authentication information into the image to identify all the pixels. The second part consists of the watermark retrieval and the image authentication processes. A secret key is applied in both the watermark embedding and retrieval processes to ensure the security of the whole system. If the image is tampered and becomes unverified, the authenticator will localize the tempered positions and recover the original image content.

4.3.1 Pixel Classification

As mentioned in the previous section, due to the simplicity of a synthetic image, most pixels in such an image can not be changed; otherwise visible artifacts can easily be introduced. To guarantee the watermark's transparency, the watermark embedding positions must be carefully chosen. Therefore, before embedding the watermark information into a synthetic image, it should be determined which pixels can be changed causing least noticeable artifacts. We classify all the pixels in an image into two categories. One category contains the so-called flippable pixels, and the other non-

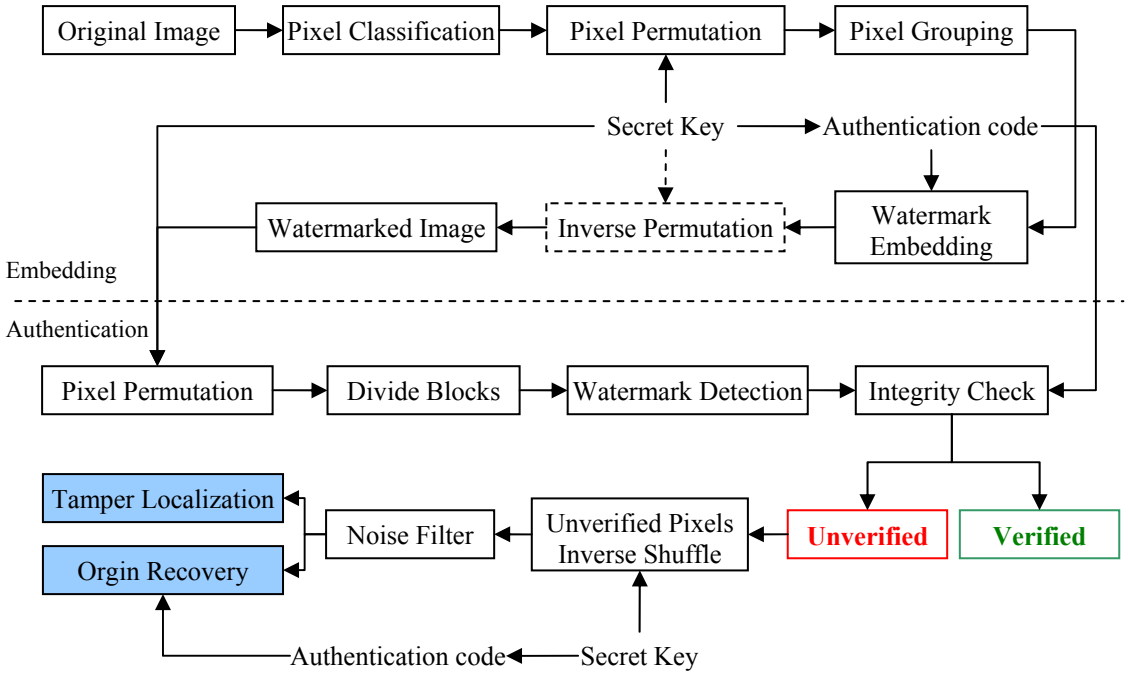


Figure 4-3 Diagram of the proposed authentication watermark system

flippable pixels. Flippable pixels can be changed to embed the watermark without causing noticeable artifacts.

For simplicity, we start with binary images to identify the flippable pixels. For instance, Figure 4-3 shows a part of the binary document image of Figure 4-1 and gives examples of flippable pixels and non-flippable pixels. The dotted lines indicate the flipping of the blue pixels and the dashed lines indicate the red ones. It can be easily observed that after flipping the blue and red pixels in the left image the caused artifacts in the right image are quite different. The flipping of the two red pixels causes distinct noticeable discontinuousness while the flipping of blue pixels doesn't. So the blue pixels are identified as flippable pixels whose change will cause less noticeable artifacts and we should avoid changing the red pixels in order to keep the image quality.

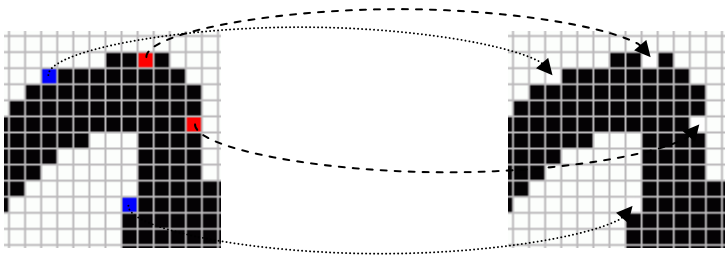


Figure 4-4 Examples of flippable and non-flippable pixels

In a binary image, the flippability of a pixel can be determined by analyzing the local property in its neighborhood, for example in an $n \times n$ block. This is not a complex problem if the used block is small, such as in a 3×3 block. In [WL04], a flippability score, which indicates the pixel's flippability, is given to each pixel by analyzing the smoothness and connectivity of the neighborhood around it in a 3×3 window. The smoothness is measured by the total number of the pixel transitions in horizontal, vertical, diagonal and anti-diagonal directions and the connectivity by the number of the black and white pixel clusters. A higher flippability score indicates that changing the pixel will cause less noticeable artifacts. Based on the similar idea, Yang and Kot proposed a new flippability criterion in [YK07]. According to this criterion, the flipping of a pixel should preserve the pixel connectivity in its 3×3 neighborhood. These two methods achieve similar results. Figure 4-5 lists the 3×3 patterns in which the change of the center pixel is less noticeable. All of the listed patterns have a flippability score larger than 0.25. The patterns (a)-(b) and (i)-(l) comply with the connectivity-preserving criterion in [YK07].

However, although the patterns listed in Figure 4-5 are relatively suitable for changing among all the possible patterns, some of them will still cause noticeable artifacts to some extent. In our watermarking scheme, we propose a statistical detection strategy that allows watermark embedding and detection errors to exist up to a reasonable rate. Thus, we can reduce the watermark payload by bearing some unsuccessful embeddings when the watermark capacity of the whole image is less than what is required. In

addition, this strategy enables us to utilize every embedded watermark bit to monitor more image pixels, which will further reduce the watermark payload. Therefore, in the proposed watermarking scheme, the required number of flippable pixels is reduced. To further reduce the impact of the watermark embedding on the image fidelity, we formulate a new set of rules to determine the flippable pixels as follows, which is simpler but much stricter. Because the rules to determine the flippable pixels are simplified, the computation cost is consequently reduced. Hence in our scheme neither offline computation nor storage of a look-up table of flippable patterns is needed. All flippable pixels can be identified online quickly.

In a 3×3 neighborhood, the center pixel will be considered as flippable when the following three rules hold.

1. Both the horizontal and the vertical transition of the center pixel must be equal to one,
2. Both the diagonal and the anti-diagonal transition of the center pixel must be equal to one and
3. There must be at least one row or column whose transition is equal to zero.

The transition of the center pixel $p(i,j)$ is calculated as follows:

$$\begin{aligned}
&horizontal: && t_h = \sum_{k=-1}^{0} D[p(i,j+k), p(i,j+k+1)], \\
&vertical: && t_v = \sum_{k=-1}^{0} D[p(i+k,j), p(i+k+1,j)], \\
&diagnol: && t_d = \sum_{k=-1}^{0} D[p(i+k,j+k), p(i+k+1,j+k+1)], \\
&antidiagnol: && t_a = \sum_{k=-1}^{0} D[p(i-k,j+k), p(i-k-1,j+k+1)],
\end{aligned} \tag{4-1}$$

where $D[\cdot]$ is a differential operator. D[a,b]=1 if a$\neq$ b and D[a,b]=0 if a=b.

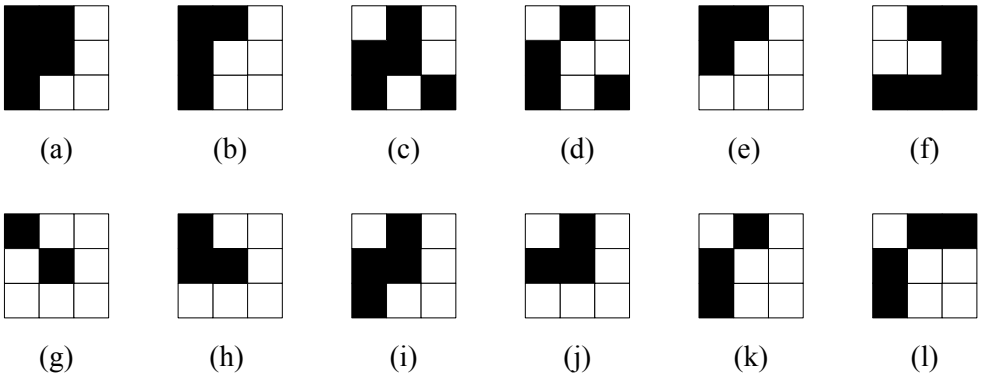


Figure 4-5 List of the patterns in which the center pixel is suitable for flipping, excluding the symmetric cases of rotation, mirroring and complement.

Rule 1 ensures that the center pixel is not in a straight line and there is one and only one pixel in both directions having the same color as the center pixel, which means the center pixel is an edge pixel. Rule 2 ensures that the center pixel is not a corner pixel. Rule 3 guarantees that the center pixel is along a straight line and not surrounded by pixels with different colors. According to these three rules, among the patterns listed in Figure 4-5, only the patterns (a) and (b) can be considered as flippable. In [WL04] these two patterns have the largest flippability score of 0.625. They also comply with the connectivity-preserving criterion proposed in [YK07]. If we unleash Rule 3, the patterns (c) and (d) will become flippable while other patterns (e)-(l) still remain non-flippable. Therefore, Rule 3 can be deemed as optional when a higher watermark capacity is required.

If the targeted synthetic image is not binary but has a limited number of colors, a pixel classification process is performed before identifying the flippable pixels. All pixels are classified into two sets of colors, $c1$ and $c2$, corresponding to the black and white color in binary images. Every set contains one or more colors. This pixel classification is similar to the image binarization problem, but because we consider only the synthetic images with a limited number of colors, the problem becomes much simpler. Such

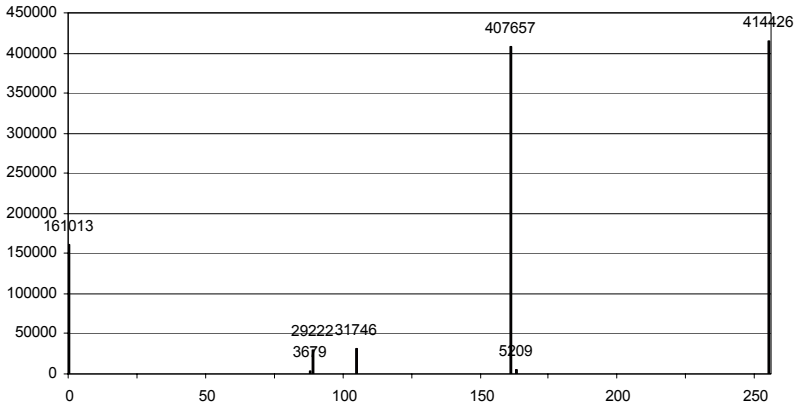


Figure 4-6 Histogram of the color map image

synthetic color images usually have a discrete histogram distribution, so it is easy to find a suitable threshold to classify the pixels into two sets.

The threshold can be fixed or be chosen adaptively depending on the image content. The middle value of all the possible pixel values, i.e. 128, is a suitable threshold for most images with moderate brightness/color distribution. For example, Figure 4-6 plots the luminance histogram of the color map image shown in Figure 4-2. The two highest peaks are the two background colors, green and white. Therefore, a threshold of 128 can separate the foreground pixels from the background. The background pixels are classified to set $c2$ and the foreground pixels to $c1$.

Nevertheless, if most pixels of the whole image are very light or dark, with this middle value we can not get enough distinct pixels to embed the watermark. In this case, the mean value can be used as the threshold to classify the pixels. Furthermore, according to the cover image's property, this classification may be determined by the pixel luminance or hue or both. In most cases, the pixels can be classified based on the difference of the luminance. However, for special images in which the pixels have the same luminance and differ only in colors, the classification should depend on the histogram of the hue.

4.3.2 Pixel Permutation

Across an image of any kind, the watermark capacity always differs from one region to another. For synthetic images, this problem becomes more obvious due to the fewer colors and more large plain areas. In the smooth areas of a single color, no watermark can be embedded without introducing visible artifacts. All the flippable pixels appear on the boundary of text and drawing. Even in the non-smooth area, however, the watermark capacity is also quite limited. Therefore, compared with a true-color natural image, the total watermark capacity in a synthetic image is much lower. However, as mentioned in the previous section the requirement for synthetic image authentication is the same or even higher as it is much easier to manipulate. Every part of the image must be protected. Therefore, the limited watermark capacity has to be utilized more efficiently. Every embedded watermark bit must be used to protect as many pixels as possible while keeping a high tamper detection resolution.

Because no watermark can be embedded in the plain areas of a single color, the pixels in such areas have to be protected by the watermark embedded in other image parts. In order to achieve this goal, a reference relationship among these pixels is required. Therefore, before embedding the watermark, we perform a random permutation of the whole image as follows:

$$\begin{aligned} I_o &\xrightarrow{permute} I_p, \\ (x_p, y_p) &= Permute((x, y), K), \end{aligned} \tag{4-2}$$

where (x, y) is the pixel coordinate in the original image I_o and (x_p, y_p) the coordinate in the randomly permuted image I_p. K is a secret key that controls the permutation process. Similarly as in Section 3.2.1.1, to guarantee adjacent pixels in the original image I_o to be distributed separately in I_p after permutation, the distance between any adjacent pixels in the original image must be larger than a minimum d in the permuted image.

The random permutation distributes the flippable pixels evenly, i.e. it equalizes the uneven watermark capacity [WTL00]. It also enhances security since the secret key K is needed to recover the permutation in order to embed/extract the watermark correctly. More importantly, the random mapping obtained by the permutation process enables statistical tamper detection that achieves a pixel-wise localization resolution. We will present the tamper localization capability in the following sections. In a practical implementation, the image will not be actually permutated while only the permutation indices are stored. All the embedding operations are done directly on the original image. Hence there is no need to do a reverse permutation after embedding.

4.3.3 Watermark Embedding

After permuting the pixels, we group them in the permuted image. One possible way to group pixels is to divide the permuted image into blocks of size b. In every block, we shall embed one watermark bit by enforcing a certain feature of the block to conform to a pre-defined relationship with the watermark bit. An essential requirement of the feature is that any single pixel manipulation in the block will change the feature of the block. One possible feature is the total number of the black pixels in one block. Any pixel flipping, whether from white to black or inversely, will change this number. For color synthetic images, the total number of the pixels belonging to category $c1$ can be used accordingly. Thus, a mutual reference relationship is established among all the pixels in the same block. Every embedded watermark bit is used to monitor a set of pixels in one block. Note that the unit that is used to embed one watermark bit does not have to be square block. The permuted image can be divided into groups in any way.

One possible way to embed the watermark bit is to apply the odd-even embedding method [WL03], also known as dither modulation embedding [CW01]. This method embeds a “0” by quantizing the total number of the black pixels in a block to 2kQ, and to (2k+1)Q to embed a “1”, where Q is a quantization step and $k \in \mathrm{Z}$. The modification is achieved by flipping the flippable pixels in the block when the quantization result of the original number of black pixels does not map to the desired bit value. Actually, the

odd-even embedding method is a special case of the look-up table approach with both the maximal runs of "0" and "1" equal to one [YM97][WL98][W03]. Both methods are illustrated in Figure 4-7. The look-up table embedding improves the security and watermark robustness when the maximal runs of "0" and "1" are bigger than one. When the quantization result is changed to an adjacent entry of the original one by noise or attacks, the extracted watermark may not be affected if this adjacent entry has the same mapping bit value. For example, as shown in Figure 4-7, when the quantization result is changed from 2kQ to (2k+1)Q, the mapping bit value is changed from "0" to "1" in the odd-even embedding case but keeping unchanged in the lookup table embedding case. The enhanced robustness, however, might conceal some pixel manipulations and render tampering of these pixels undetectable.

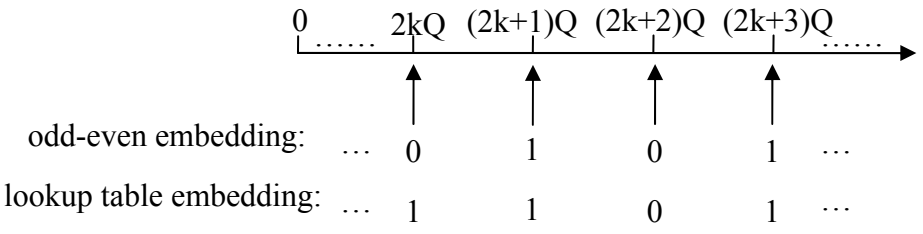


Figure 4-7 Illustration of different watermark embedding approaches

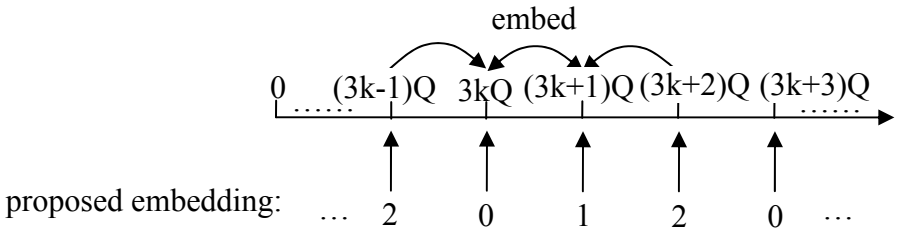


Figure 4-8 Illustration of the proposed embedding approach

Furthermore, both the odd-even embedding and lookup table embedding approaches can only give a binary result whether the extracted watermark bit matches the original one or not, but can not provide accurate information about how the feature value is changed. In other words, no original feature value can be estimated. As shown in Figure 4-7, these approaches can not determine whether the changed quantization result is dragged from the left entry to the right one or from the right one to the left one. For example, when the detected watermark bit is "0" and the original bit is "1", both approaches can not tell if the original bit corresponds to the left 1-entry of the detected 0-entry or the right 1-entry, namely, if the number of the black pixels in a block is increased by Q or decreased by Q. Nevertheless, the capability of estimating the original feature value is very desirable in the authentication applications because it can help to recover the original image content. Therefore, we propose a new embedding strategy to embed the watermark.

The proposed embedding approach is illustrated in Figure 4-8. A dummy quantization entry, the entry "2" as shown in Figure 4-8, is introduced. We name the entry "2" a dummy entry because none of the feature values will be enforced to the values corresponding to this dummy entry during the embedding process. The watermark bit only takes on the values "0" and "1". The watermark sequence $w(n)$, used as an authentication code, is generated under the control of the secret key K, $w(n)=G(n, K)$, where $w(n) \in \{0,1\}$. Nevertheless, the introduction of the dummy entry makes it possible to estimate how the feature value is changed. Thereby it enables a recovery capability in the authentication process. A binary version of the tampered region can be recovered based on a statistical clustering of the potential tampered pixels. The recovery capability will be addressed in detail in Section 4.4.

For every block, the embedding mechanism is as follows. Let M_j be the number of the black pixels in the jth block of the permuted binary image. In the color image, M_j is the number of the pixels in the jth block that belong to the set $c1$ accordingly. M_j is then quantized by the quantization step Q:

$$M_j = \left\lfloor \frac{M_j}{Q} \right\rfloor \cdot Q + \Delta_j, \tag{4-3}$$

$$Quan(M_j) = \begin{cases} 0 & if \quad \lfloor M_j / Q \rfloor = 3k \\ 1 & if \quad \lfloor M_j / Q \rfloor = 3k+1, \quad k \in \mathbf{Z}, \\ 2 & if \quad \lfloor M_j / Q \rfloor = 3k+2 \end{cases} \tag{4-4}$$

where $\lfloor . \rfloor$ is the floor function and Δ_j is the quantization residue.

The watermark bit $w(j)$ is embedded by enforcing M_j to be [3k+ $w(j)$]Q. To minimize the modification, M_j is enforced to the nearest entry that maps to $w(j)$. As shown in Figure 4-8, the M_j mapping to "2" is forced to its neighbors of the entry "0" or "1" and the M_j mapping to "0" or "1" is switched to each other according the desired watermark bit value $w(j)$. The following equation describes this process.

$$M_j^* = \begin{cases} \lfloor (M_j + Q/2)/Q \rfloor \cdot Q & if \quad Quan(M_j + Q/2) = w(j) \\ \lfloor (M_j + Q/2)/Q \rfloor \cdot Q \pm Q & if \quad Quan(M_j + Q/2) \neq w(j) \end{cases} \tag{4-5}$$

where M_j^* is the number of the black pixel in the jth block after watermark embedding. The update from M_j to M_j^* is achieved by modifying the flippable pixels in the jth block. If the cover image is a color image, the feature value update is performed by modifying the proper flippable pixels to the nearest colors in the neighborhood that belong to the different category. In this way, the watermark visibility is minimized.

Larger Q will increase the robustness of the watermark because any perturbation smaller than Q/2 caused by noise will not affect the accuracy of watermark detection, which is a desirable feature in data hiding or robust watermarking applications. In authentication applications, larger Q can provide more accuracy of the estimation of pixel manipulations. As all the feature values are enforced to kQ in the embedding process, any mismatch can be detected even if the extracted watermark bit might still be correct. In the particular case of synthetic images where the number of pixels is used

as feature value, when the number of the tampered pixels in a block is larger than one, larger Q can provide better estimation of the number of the modified pixel than smaller Q by detecting how far the feature value is away from the previously enforced value. However, using a larger Q will lead to more pixel modification during watermark embedding and this will subsequently degrade the image quality.

4.3.4 Watermark Retrieval

Watermark retrieval is done by performing the same permutation and quantization processes in the targeted image as in watermark embedding. The secret K and the block size b are conveyed to the watermark detector as side information. The targeted image $\hat{I}_o^*$ is first randomly permuted under the control of the secret key K. Then the permuted image $\hat{I}_s^*$ is divided into blocks with the size b. Let $\hat{M}_j^*$ be the number of the black pixels in the jth block of the permuted binary image. Every watermark bit value is retrieved as follows:

$$\hat{w}(j) = Quan(\hat{M}_j^* + Q/2), \tag{4-6}$$

where $\hat{w}(j)$ is the extracted watermark from the jth block. Note that the extracted watermark $\hat{w}(j)$ may take on three values of "0", "1" and "2" unlike the original watermark sequence $w(j) \in \{0,1\}$. Obviously, all the extracted watermarks with the value "2" do not match the corresponding original watermark bits and therefore are detection errors. In the following authentication process, these detection errors will be used to estimate the original pixel value.

4.4 Authentication and Pixel Recovery

In this section we present the image authentication, tamper localization and pixel recovery processes. For clear description, we first introduce the authentication and tamper localization processes in Section 4.4.1 without taking into account pixel

recovery. Afterwards, we present the strategy of recovering tampered pixels in Section 4.4.2.

4.4.1 Image Authentication and Tamper Localization

The authentication process is similar to the one proposed in Section 3.3, which consists of three steps as illustrated in Figure 4-9. First, with the knowledge of the secret key K, the original watermark sequence w can be regenerated. After obtaining the extracted watermark $\hat{w}(j)$ for the jth block, the verification can be done by comparing it with the original watermark bit $w(j)$. For every block, if the extracted watermark bit does not match the original one, the block will be considered as an unverified block. All the pixels in an unverified block are marked as potential unverified pixels. As shown in Figure 4-9 (a), the gray block indicates the unverified blocks in the permuted image.

Second, the image is inversely permuted and all the pixels are moved back to their original position in I_o. The potential unverified pixels marked in step 1 will be randomly distributed over the whole image. Because the mismatches of extracted watermark bits are caused by the pixel manipulation, every unverified block must contain at least one altered pixel from the tampered regions. Therefore, after the inverse permutation, all the altered pixels in every unverified block will be moved back to the tampered region. Consequently, in the tampered region, the unverified pixels will be clustered together and have a much higher density than other areas. Furthermore, the clusters of the unverified pixels will form the shape of the original content in the tampered region in a pixel-wise resolution. Outside the tampered regions, the other potentially unverified pixels will scatter over the whole image sparsely. All these isolated unverified pixels will be considered as noise dots and be filtered out in the next step. As shown in Figure 4-9 (b), the gray dots and areas indicate potential unverified pixels in the original image domain. The gray rectangle block denotes the actually tampered region with high density of unverified pixels.

Finally, the authentication result is refined by a noise filter, e.g. a median filter, to remove the noise-like isolated unverified pixels. The filter in described in Equation (3-16) can also be applied here. Thus, the tampered region will be easily picked out, as shown in Figure 4-9 (c).

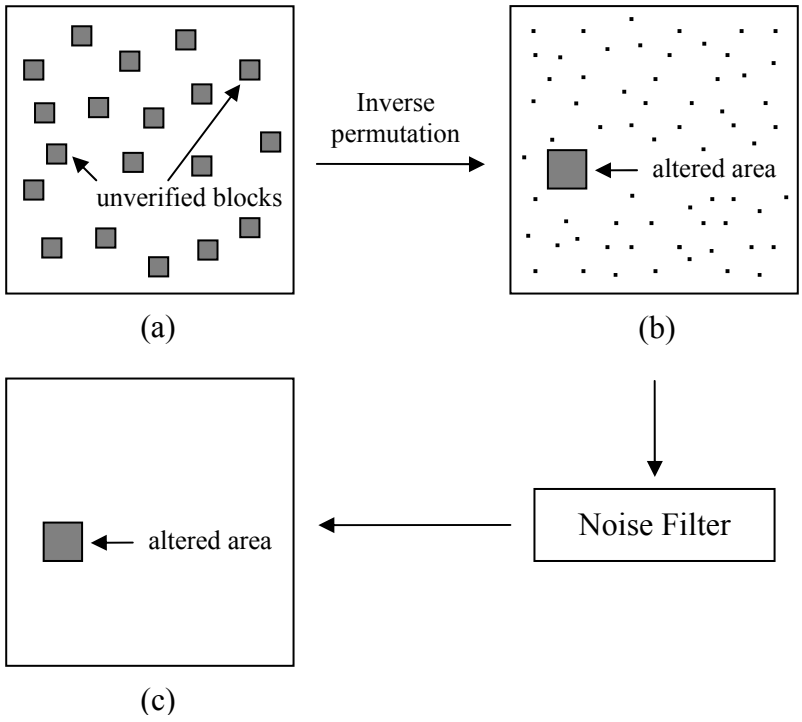


Figure 4-9 Illustration of the image authentication process: (a) unverified blocks in the permuted image, (b) potential unverified pixels (scattering over the image like noises) and the altered area (with high density of unverified pixels), (c) altered area after noise filter (isolated noise-like unverified pixels removed).

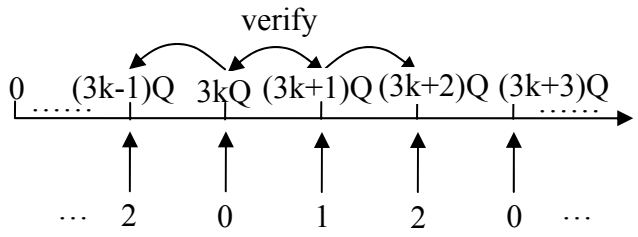


Figure 4-10 Illustration of watermark verification process

Table 4-2 Possible mismatch cases between the extracted and the original watermarks

Case	$w(j)$	$\hat{w}(j)$
1	0	1
2	0	2
3	1	0
4	1	2

4.4.2 Recovery of Tampered Pixels

In this section, we present how the altered pixels in the tampered regions can be recovered. As illustrated in Figure 4-10, if the extracted watermark $\hat{w}(j)$ does not match the original watermark bit $w(j)$, there are four different cases how the watermark might be changed. Table 4-2 lists the four possible mismatch cases between the extracted and the original watermark bits.

The watermark mismatch is incurred by the change of the feature value, either by increasing or decreasing. Due to the introduction of the dummy entries "2", in order to get the watermark change in each case listed in Table 4-2, the amount of increasing or decreasing the feature value is different. For example, Case 2 where the watermark is changed from "0" to "2" might be caused by a decrease of the feature value of [Q/2, 3Q/2] or an increase of [3/2Q, 5/2Q]. If we assume that the modification of the

feature value of a block is smaller than 3/2Q, the feature value can only be moved to its neighborhood. In practice, this assumption is reasonable because when the size of manipulated regions is far smaller than the image size, the total amount of pixel manipulation will be evenly distributed into quite a few blocks in the permuted image. Hence, the amount of pixel modification in every single block will be quite limited. If the size of manipulated regions is relatively very large, the tampered area can not be localized any more due to the overwhelming noise-like potential unverified pixels in the authentication process. The limit of the manipulated region size that can be localized will be discussed in the next section.

Under this assumption, we can determine whether the feature value is shifted to the left or the right entry by the pixel manipulation, namely, whether it is increased or decreased. When Q=1, this means that only one pixel in a block is altered. Based on the shift direction of the feature value, we can determine what kind of pixel is changed in the block. In case the feature value is the number of black pixels or the pixels belonging to category $c1$ in color image, the original pixel can be recovered as listed in Table 4-3. For example, in Case 2 where the extracted watermark $\hat{w}(j)$ is "2" and the original one $w(j)$ is "0", it means the number of the black value is increased by one, namely, one of the white pixels in the *j*th block is modified to black. Therefore, the original pixel $I_o(x,y)$ should be white. At this step, we can not exactly determine which white pixel in the block is changed, so we consider all the white pixels in the block as potential unverified pixels. Note that we can also mark all the pixels in the unverified blocks as potential unverified pixels instead of only white pixels. It will compensate the errors caused by the blocks in which not only one white pixel is changed and will therefore increase the density of the unverified pixels in the tampered region after the inverse permutation. In the other 3 cases, the original pixel color can be similarly deduced.

Based on the mismatch of the extracted watermark and the original one, we classify the unverified pixels into two categories. If the watermark change falls into Case 1 and 3, all the unverified pixels in the *j*th block are classified into category $u1$. If the watermark change falls into Case 2 and 4, all the unverified pixels in the *j*th block are classified

into category $u2$. Thus, we obtain two different kinds of unverified pixels. After the inverse permutation in the second step of authentication process, the two categories of unverified pixels will be clustered in separate tampered regions according to the way of the pixel manipulation and form the shape of the removed original content or the added parts. When all pixels in unverified blocks are considered as unverified, there will be some mixture of unverified pixels of two categories in some tampered regions. In each tampered region, however, the unverified pixels of one category will be overwhelming majority. Therefore, the original pixel values of each tampered region can be easily estimated and recovered. After filtering the isolated noise-like unverified pixels in the third step of the authentication process, together with the reconstruction of the shape of the original content and the added part, the manipulated image content can be recovered accordingly. Note that for color synthetic images only the original category of the pixel can be recovered instead of the actual pixel color in binary images.

Since there are two categories of unverified pixels, the noise filter can be applied to every category separately. A properly designed noise filter can not only filter out the noise-like pixels, but also can compensate for detection errors. If the number of altered pixels in one block is larger than one, the filter can be used to smooth out the wrong points. For example, if only white or black pixels are modified in a block and the number of the modified pixels is a multiple of 3, the output of the quantization function will change from kQ to (k+3), which have the same mapping value and therefore the pixel modification will not be detected. Another possible case occurs if a certain number of black pixels are modified to white in a block, while the same amount of white pixels are modified to black in the same block: the feature value, i.e. the number of the black pixels, will keep the same and such modification can not be detected. Such tamper detection errors will result in some holes in the area of converged unverified pixels, the noise filter will fill these holes by checking their neighbor pixels. Furthermore, the size of the applied noise filter can be scaled differently to detect manipulations at various scales according to the requirement of particular applications.

Table 4-3 Original pixel recovery based on the watermark comparison

Case	$w(j)$	$\hat{w}(j)$	$I_o(x,y)$
1	0	1	white/$c2$
2	0	2	black/$c1$
3	1	2	white/$c2$
4	1	0	black/$c1$

4.5 Analysis and Discussion

4.5.1 Quality of Watermarked Image

In this section we shall discuss the quality of the watermarked image of the proposed embedding method and compare it with the odd-even embedding and look-up table embedding approaches. In the odd-even embedding method, every entry has two neighbors with the different mapping value. Hence, when the feature value needs to be forced to a desired mapping bit value, it can be modified in both directions by choosing a nearer one to minimize the artifact caused by embedding. For example, when the quantization step Q=1, in the odd-even embedding case the maximal number of pixels that need to be flipped in a block is bounded to one. In the look-up table embedding case, however, with the maximum run of "0" and "1" larger than one, some mapping entries have one or both neighbors with the same mapping value. So if the original feature value does not map to the desired watermark bit value, in order to minimize the amount of modification, the update of the feature value can only be made in the direction that has a shorter distance to the next entry with different mapping value. When Q=1, the maximal flipping number may be larger than one when the maximal run of "0" or "1" exceeds two. In this case, more flippable pixels are required to fulfill the watermark embedding and the image quality will be degraded due to more pixel modifications.

The proposed embedding method resembles the look-up table embedding case with maximum run of 2. When the quantization step Q=1, the maximal number of pixels that need to be modified in a block is still bounded to one. When the desired watermark bit value is “0” or “1”, as illustrated in Figure 4-8, whether the original feature value maps to “0”, “1” or “2”, only one step change is necessary to map the feature value to the desired bit value. However, compared with the odd-even embedding, the direction of the one-step change is constrained. For example, when the feature value maps to “0” while the watermark bit is “1”, the feature value must be increased in order to fulfill the embedding by only modifying the feature value by Q. Otherwise, a 2Q decrease has to be made to map the feature value to “1”. In this case, therefore, a white flippable pixel, or a flippable pixel belonging to color set *c*2 in the case of color images, is required to be modified. If there is no white flippable pixels in the block, two black flippable pixels have to be changed to get the same mapping value. As we mentioned in the previous section, in the proposed watermarking scheme, thanks to the statistical tampering detection mechanics, embedding errors are allowed to some extent. Hence, in this example if there is no white flippable pixel in the block, all the pixels in the block will be kept intact in order to keep the image fidelity high. As a tradeoff, this will result in a detection error. When the image quality is not a key requirement in some particular applications, this constraint can be released.

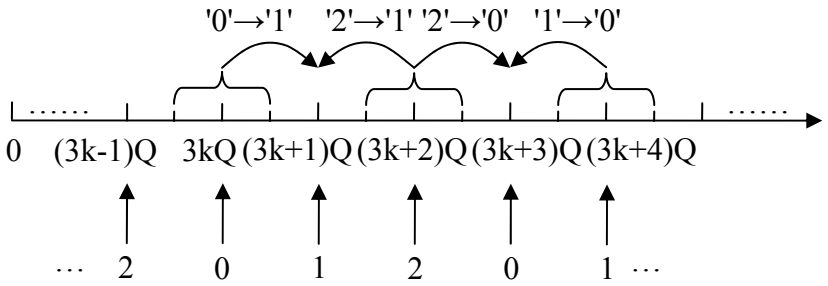


Figure 4-11 Illustration of the modification of the feature value in the case that the quantized feature value does not map to the desired watermark bit value.

Although the maximal number of pixels that need to be modified can be bounded by one in every block, the proposed embedding will introduce more modifications in the whole image than the look-up table embedding with maximum run of 2. The reason lies in two aspects. First, in the look-up table embedding case there is still the possibility that some entries have both neighbors mapping to the other different bit value like in the odd-even embedding case. But in the proposed embedding method every entry has two neighbor entries mapping to the other two different bit values. Second, due to the existence of the dummy entry, in the proposed embedding method there are three different mapping values, "0", "1" and "2", instead of two values in the look-up table embedding case. Therefore, the probability that the mapping value of the quantized feature value kQ matches the desired watermark bit decreases from 1/2 to 1/3. In other words, with probability of 2/3 the feature value has to be shifted to its neighbor entry that maps to the desired watermark bit.

In the following we shall calculate the introduced distortion by the proposed embedding method quantitatively. If we assume that the original feature value is uniformly distributed over the range of $\psi = [(k-1/2)Q,(k+1/2)Q]$, in the case that the mapping value corresponding to the quantized feature value matches the watermark bit, the feature value is rounded to kQ and the mean squared error caused by the embedding is

$$MSE_1\big|_{\psi} = \frac{1}{Q}\int_{-\frac{Q}{2}}^{\frac{Q}{2}} \tau^2 d\tau = \frac{Q^2}{12}. \tag{4-7}$$

Note that Equation (4-7) is the same as Equation (3-20) that describes the case of odd-even embedding. This means that in this case the MSE caused by the proposed method is the same as the odd-even embedding.

Now we consider the case that the quantized feature value does not map to the desired watermark bit value. In this situation, the feature value has to be modified to the nearest neighbor entry (k-1)Q or (k+1)Q that maps to the desired bit value in order to embed the watermark bit. As mentioned above, in the proposed method, only one direction can

be chosen to shift the feature value in order to minimize the amount of modification. Figure 4-11 illustrates the process of modifying the feature value in this case. As shown in Figure 4-11, for all the four cases of shifting, i.e. "0"→ "1", "1" → "0", "2" → "0" and "2" → "1", the minimal and the maximal shift distances are 1/2Q and 3/2Q respectively. Thereby the mean squared error incurred by the embedding in this case is

$$MSE_2\big|_{\psi} = \frac{1}{Q}\int_{-\frac{Q}{2}}^{\frac{Q}{2}} (\tau \pm Q)^2 d\tau = \frac{13}{12}Q^2. \tag{4-8}$$

Thus, the overall MSE incurred by watermark embedding is

$$MSE\big|_{\psi} = MSE_1\big|_{\psi} + MSE_2\big|_{\psi} = \frac{1}{3}\cdot\frac{Q^2}{12} + \frac{2}{3}\cdot\frac{13Q^2}{12} = \frac{3}{4}Q^2. \tag{4-9}$$

Compared with Equation (3-22), we can see that using the same quantization step Q the proposed embedding method introduces MSE distortion of $3/4\,Q^2$, which is larger than MSE distortion of $1/3\,Q^2$ caused by the odd-even embedding. It is also larger than the MSE distortion of $1/2\,Q^2$ caused by the look-up table embedding [W03]. Therefore, the quantitative analysis result conforms to what we analyzed above. The degradation of the image quality is the cost of the capability of estimation of the original pixel value, which enables the tamper recovery.

4.5.2 Sensitivity to Pixel Manipulations

For binary images, by using the number of black pixels in a block as the feature value, the watermark detector is very sensitive to any single pixel manipulation because any pixel flipping may cause the feature value of the block changed. Even one single pixel modification will make the extracted bit from the corresponding block mismatch the original one and raise tampering alarm. Although one single unverified block will only cause a few unverified pixels (coming from the unverified block) that scatter over the whole image and the actually manipulated pixel is indistinguishable, a tampering as small as 2 pixels or bigger will become distinguishable and can be easily picked out

from other isolated unverified pixels. Thus, the position of tampering can be precisely localized.

Nevertheless, when more than one pixel is subject to be modified in one single block, it might occur that the quantization result of modified feature value maps to the same value as the original bit. Subsequently, such false bit-matching will result in undetected manipulated pixels. Let n_b be the number of the changed black pixels and n_w the number of the changed white pixels and Q=1. In the case of the odd-even embedding, when the total number of the changed pixels, $k=n_b+n_w$, is even, the watermark bit extracted from this block will not be changed, namely, still matching to the original bit. If there are totally n pixels in one block and each pixel change is independent from other pixels with probability p, the probability that the embedded bit can still be successfully extracted from a modified block is

$$P_m = \sum_{\substack{k=2,\\ k \text{ is even}}}^{n} \binom{n}{k} p^k (1-p)^{n-k} = \frac{1+(1-2p)^n - 2(1-p)^n}{2}. \tag{4-10}$$

In the case of the proposed embedding method, whether the extracted bit matches the original one or not depends on the number of the changed pixels. The extracted bit will match the original one only when the difference of the numbers of the changed black pixels and white pixels is equal to the multiples of three, i.e. $|n_b - n_w| = 3i$, where $i \in N^0$. If we assume that in the block there is the same number of black pixels as white pixels, the probability in Equation (4-10) becomes

$$P_m = \sum_{k=2}^{n} \left(\sum_{\substack{n_b+n_w=k,\\ |n_b-n_w|=3i}} \binom{n/2}{n_b} \binom{n/2}{n_w} \right) p^k (1-p)^{n-k}. \tag{4-11}$$

Since the number of the changed pixels is constrained by the condition $|n_b - n_w| = 3i$, the binomial coefficient in Equation (4-11) becomes much smaller than that in

Equation (4-10). Therefore, with the proposed embedding approach, the miss probability of pixel manipulation is significantly decreased.

For color synthetic images, the sensitivity to pixel manipulation depends on the used feature value. For example, in the case that all the pixels are divided into two categories as introduced in Section 4.3.1 and the feature value is defined as the number of pixels that belongs to color set $c1$, if the pixel manipulation does not change the category that the pixel belongs to, it can not be detected as it will not affect the used feature value. This problem can be compensated by utilizing a more complicated feature value that involves all the possible pixel colors explicitly, which will be changed by any kind of pixel value modification. In Section 4.7, Equation (4-17) gives an example of feature value that can identify and recover three kinds of colors. If the recovery capacity is not of concern, it becomes easier to design a feature value to identify multiple pixel values.

Similarly as discussed in Section 3.4.3, the final sensitivity of the proposed authentication system can be adjusted by the noise filter size and threshold in the authentication process. By applying noise filters of various sizes and different thresholds, the authenticator can bypass possible incidental noises that may be introduced in the targeted application so that the false alarm rate can be decreased. When no noise filter is applied in the authentication process, the highest sensitivity of the scheme will be achieved.

4.5.3 Localization and Recovery Capabilities

As we presented in Section 4.4, the tamper regions are localized by the clustering of identified unverified pixels after mapping back to the original image domain. According to the sensitivity analysis in the previous section, as long as the number of modified pixels in one block doesn't conform to the condition $|n_b - n_w| = 3i$, a watermark bit mismatch will occur and this will render the block unverified. All the pixels in unverified blocks, including both the actually altered or unaltered pixels, will subsequently be identified as potential unverified pixels. In this way, almost every

single altered pixel will be identified. Therefore, after mapping all the potential unverified pixels back to the original image domain, the authenticator achieves a pixel-wise resolution of tamper detection and localization based on the distribution density of unverified pixels. Under the assumption that one single block in the permuted domain will contain at most one modified pixel, the original values of altered pixel(s) can be estimated. After all altered pixel values are estimated, the estimation result is refined based on the distribution density of each kind of unverified pixels.

Because the same random permutation technique is applied as in Chapter 3, the proposed scheme has the similar advantage and limitation with respect to tamper localization. As discussed in Section 3.4.2, the tamper localization resolution is independent of the watermark payload that is determined by the used block size. This implies that a reasonably large block size can be used to reduce the watermark payload without decreasing the resolution of tamper localization. Larger block size, however, will increase the probability of false alarm and decrease the maximal size of altered area that can be accurately localized. The analysis of the probability that the potentially unverified pixels casually form a connected area can be similarly deduced as in Section 3.4.2. In the case of synthetic image authentication, Equation (3-26) becomes

$$r_{BER} = \frac{N_e}{N_T} = \frac{N_e}{\frac{WH}{b \times b}} = \frac{N_e b^2}{WH} = \frac{N_u}{WH}. \tag{4-12}$$

In this case the probability that a pixel is a potentially unverified pixel is equal to the watermark detection error rate

$$P_u = \frac{N_u}{WH} = \frac{N_e b^2}{WH} = r_{BER}. \tag{4-13}$$

Then the probability that there are at least t unverified pixels in the 3×3 neighborhood of an unverified pixel can be obtained by Equation (3-28) as follows:

$$P_{fa} = \sum_{k=t}^{8} \binom{8}{k} P_u^{\,k} (1 - P_u)^{8-k}. \tag{4-14}$$

Similarly as discussed in Section 3.4.2, given a block size *b*, the authenticator can only provide accurate tamper localization and recover the altered pixels when the amount of altered pixels is bound to a certain limit. If a large amount of pixels is altered, the tampered area may not be localized because too many blocks are marked as unverified and then the unverified pixels become overwhelming after mapping back to their original positions. Hence, it is very difficult or even impossible to distinguish the actually tampered area from other noise-like unverified pixels. In such a case, the manipulation will render the whole image unauthentic.

If the random permutation distributes all the pixels evenly over the image, when the number of altered pixels reaches

$$L_w = \frac{H \times W}{b^2}, \tag{4-15}$$

where L_w is the length of the watermark sequence, all the blocks will contain one altered pixel. Therefore, all the pixels will be marked as unverified. After mapping back to the original image domain, the image will be full of randomly distributed unverified pixels. In this case, neither tamper localization nor pixel recovery is possible any more.

Equation (4-15) reveals that smaller block sizes will increase the limit of maximal localizable area, because every embedded watermark bit monitors fewer pixels and the number of potential unverified pixels introduced by every altered pixel will decrease. However, using a smaller block size will increase the length of watermark sequence, i.e. higher watermark payload. Embedding more watermark bits will cause more pixel modifications and therefore degrade the image quality. On the other hand, the number of embeddable watermark bits is bounded by the number of flippable pixels, i.e. the total watermark capacity. Therefore, the minimal block size depends on the number of flippable pixels and the cover image size. Given a quantization step Q, according to Equation (4-9), the average amount of pixel modification is $\sqrt{3}/2Q$. Thus, for a cover

image of size $H \times W$, in order to ensure successful embedding in each block, the minimal block size is limited to

$$b_{\min} = \sqrt{\frac{H \times W}{N_f} \frac{\sqrt{3}}{2} Q}, \tag{4-16}$$

where N_f is the total number of flippable pixels in the cover image.

In addition, the recovery capability of the proposed scheme is limited to a binary pixel recovery, which can only recover two kinds of pixel values. In the proposed embedding method, there are three different mapping entries instead of two entries in the classical look-up table embedding. Thanks to the dummy entry, we can estimate the original feature value when the extracted watermark does not match the original one. In the pixel recovery process, we make an important assumption that the amount of modification of the feature value in one single block is smaller than 3/2Q, namely, one of its neighbor entries that maps to the original watermark bit will be considered as the original enforced feature value in the watermark embedding. Since there exists only one dummy entry between "0" and "1", we can only distinguish the modification with one hop from the original mapping entry to its neighbor entries, i.e. whether the quantized feature value is increased or decreased by Q. Any modification that causes two or more hops will cause an estimation error. This means only one kind of modification amount Δ_f can be distinguished. Since the pixel recovery process is based on the estimation of the change of feature value, we can only recover the modified pixels into two categories as we mentioned in Section 4.4. Therefore, for color images, the original brightness and color information may be lost. We shall present a possible extension of the proposed scheme in Section 4.7 that can recover three different pixel values.

4.5.4 Security

The security issues can be analyzed similarly as in Section 3.4.4. In the applications of image content authentication, the main objective of an adversary is to forge an authentic image that can pass the authenticator test. Therefore, it is important to study the following two kinds of problems [CMB01][WL04][YK07]: (1) the possibility of manipulating the image content without changing the embedded authentication code, and (2) the possibility for an adversary to embed a valid authentication code into an image. In both problems, we assume that an adversary knows the algorithm but has no knowledge of the secret key. For the first problem, according to the sensitivity analysis in Section 4.5.2, whether the number of altered pixels is small or large, the probability of missing to detect the alteration is always very small as long as the number of blocks, i.e. the authentication code length, is reasonably large.

For the second problem, when multiple watermarked image copies of the same cover image with different data embedded by the same key are available to an adversary, it may be possible for an adversary to estimate which pixel conveys what data by comparing those copies, so that he can compose an image with his own data embedded [WL04][YK07]. In the proposed scheme, since the embedded watermark, i.e. the authentication code, is generated by the secret key, with the same key the same code will always be embedded into all the cover images. Hence, there will never be multiple copies in the proposed scheme. Without the availability of multiple copies, it is extremely hard for an adversary to embed specific data into the image due to the secrecy of random permutation, even if he knows the algorithm. Furthermore, without the knowledge of the key it is also very hard for an adversary to get the correct authentication code. The length of the code L_w is equal to the number of the blocks, so the probability of getting a correct code is 2^{-L_w}. When L_w is reasonably large, this probability will be very small. For example, given an image of 512×512 pixels, when the block size is set to 16, this probability is as small as 2^{-1024}.

If the authenticator is unlimitedly accessible to an adversary, the security problem discussed in Section 3.4.4 also exists in the proposed scheme in this chapter. In this case, filtering of isolated unverified pixels must be a mandatory step. Otherwise, since the proposed embedding method in this chapter is based on pixel-wise features and flipping, every time an adversary alters an individual pixel, the pixels belonging to the same block will be output as unverified pixels by the authenticator. Thus, the composition of one block is exactly revealed. In this way, an adversary can alter the image pixel by pixel and record the authentication outputs. After sufficient attempts, the whole pixel mapping built by random permutation will be precisely discovered. Once the pixel permutation is known, the adversary can easily mount a successful attack by manipulating a preferable image part at will and then altering the other corresponding pixels to keep feature values of each affected blocks unchanged. Therefore, in order to counteract such kind of attacks, the authentication result must be refined by a noise filter with reasonable size so that the pixel alteration of small size such as pepper-and-salt noises will be neglected. The smallest detectable alteration size depends on the applied filter size and the threshold.

4.6 Experimental Results

To evaluate the proposed scheme, we use an image set that includes 508 synthetic images of different kinds and different sizes. The test image set consists of binary text images, line drawings (e.g. comic images) and color digital maps. The text images are obtained from different types of scanners and screenshots, including both typed and handwritten documents. These documents are not only written in Latin letters (in English and German) but also in Chinese characters. The used test maps consist of city maps and topographic charts. Figure 4-12 shows some examples from the synthetic image test set.

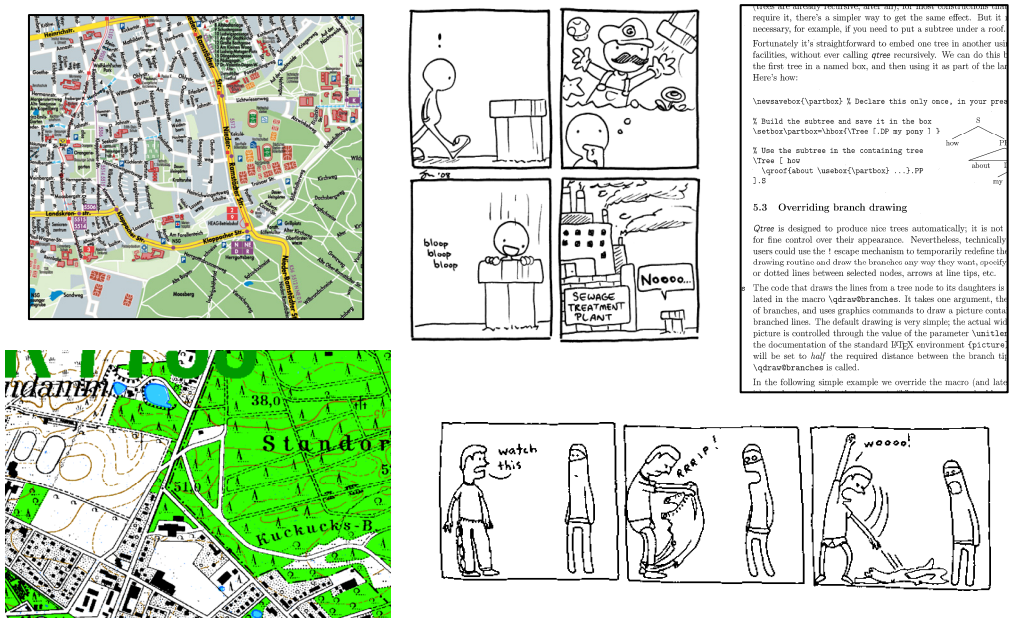


Figure 4-12 Sample images from the synthetic image test set

In the following tests, if not specified, the parameters are set as below:

1. Quantization step Q=1 (i.e. no quantization),
2. Block size b=16,
3. Only patterns (a) and (b) in Figure 4-5 are considered as flippable pixels,
4. At most one pixel is flipped in each block.
5. Noise filter size d=5 and the threshold T=4.

For color images, the mean luminance value of the whole image is used as the threshold to classify the two color sets. Figure 4-13 plots the histogram of the detection error rate (BER) of all the images in the test set.

Because of the condition 4 above, in some blocks the desirable watermark bit can not be successfully embedded by only modifying one flippable pixel. This constraint ensures the high fidelity of the watermarked image but it also introduces detection errors even if the image is not tampered. From Figure 4-13, we can see that all images

have a detection error rate lower than 0.167 and for most test images the detection error rate is lower than 0.10. Higher detection rates occur in such images that contain plenty of white plain regions or straight drawing lines such as tables. In these image parts, there are no or only very few flippable pixels. Therefore, the whole watermark capacity of those images is significantly decreased. Nevertheless, these errors are allowable since they introduce only isolated detection errors that can be easily distinguished from the actually tampered areas. Figure 4-13 also plots the corresponding probability P_{fa} that there are at least five unverified pixels in a 3×3 neighborhood. We can see that this probability is always kept very low. Even with the highest BER of 0.167 in the test set, the corresponding probability is only 4.53×10^{-3}. Therefore, with a reasonable detection error rate, the embedded watermark can still achieve a good tampering detection, localization and pixel recovery performance, which will be presented in the following tests.

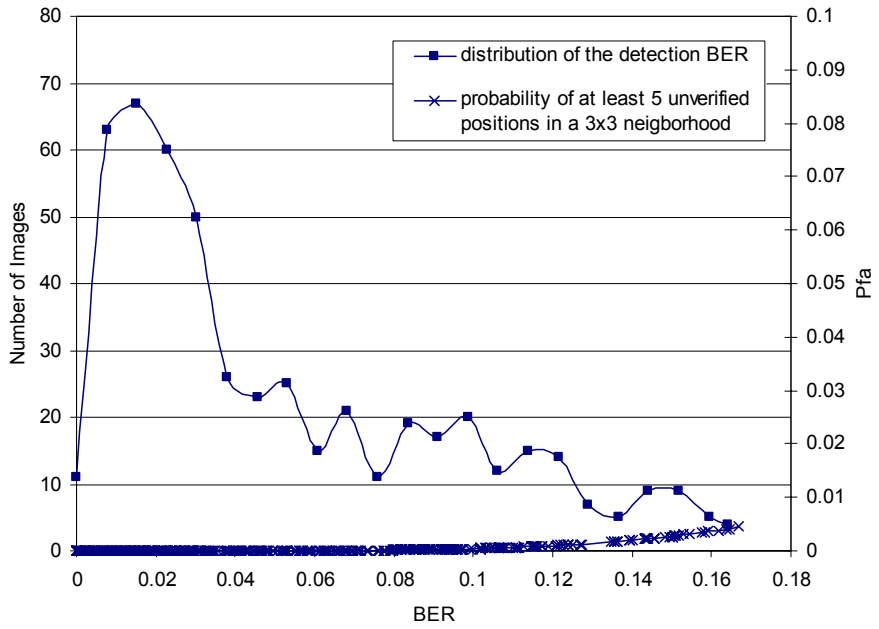


Figure 4-13 Distribution of the detection BER of the whole synthetic image test set (508 images in total) and the corresponding probability that there are at least five unverified positions in a 3×3 neighborhood, i.e. t=5.

To compare the probability P_{fa} with different parameters and evaluate the analysis result in Section 4.5.3, more test results are presented in Figure 4-14. The experimental results in Figure 4-14 are obtained from the whole synthetic image test set (508 images in total). Every 3×3 pixel neighborhood in each test image is involved to calculate the probability P_{fa} corresponding to the obtained BER value from the image. The analytical results are calculated by Equation (4-14) using the BER values obtained from all the test images in the experiments. Figure 4-14 plots three groups of curves for different parameter *t*, i.e. for t=4, t=5 and t=7 respectively. From these groups of curves in Figure 4-14, we can see that the experimental results and the analytical results conform to each other very well for all the parameter *t* values.

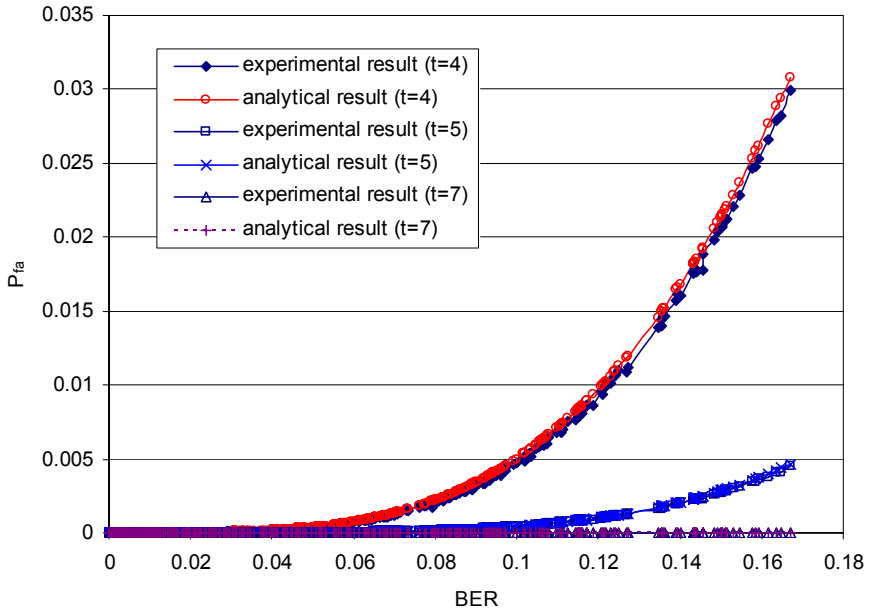


Figure 4-14 Analytical and experimental results of the probability that there are at least *t* unverified pixels in a 3×3 neighborhood. The experimental results are obtained from the whole synthetic image test set (508 images in total).

Table 4-4 Watermark Embedding Information of Various Sample Images

Test Image	Image Size	Block #	Non-matched Percentage	Flipped Pixels #	Unsuccessful Embedding #
Typed Text	856×575	1890	64.3%	1176	39
Handwritten Text	1755×1275	8611	66.3%	5026	687
Digital Map	1202×876	4050	66.5%	2561	133

In the rest of this section, we present more test results to evaluate the tampering detection and recovery capabilities of the proposed scheme by taking a variety of images from the test set as examples. These example images include a binary typed text image, a binary handwritten text image and a color digital map, as shown in Figure 4-16, Figure 4-19 and Figure 4-20. The original handwritten text image is shown in Figure 4-1. Table 4-4 lists the watermark embedding information of these three different kinds of images. Note that for the handwritten text image, due to the large white margins and plain areas, the unsuccessful embedding rate reaches nearly 8%.

4.6.1 Binary Text Images

Two kinds of binary text images are taken as examples, one handwritten text image and one typed text image, which are both typical synthetic images and widely used in many applications. Figure 4-15 gives a closer view of a part of the original and the watermarked handwritten text image. As shown in Figure 4-15 (a) and (b), the original and watermarked images look nearly identical to human observers, i.e. the watermark embedding process does not introduce noticeable artifacts. The difference between the original and the watermarked images is shown in Figure 4-15 (c). The flipped pixels are shown in black.

Figure 4-16 and Figure 4-17 present different kinds of manipulation tests and authentication results of a handwritten text image. In Figure 4-16, two kinds of manipulations, content addition and content deletion, were made on the watermarked

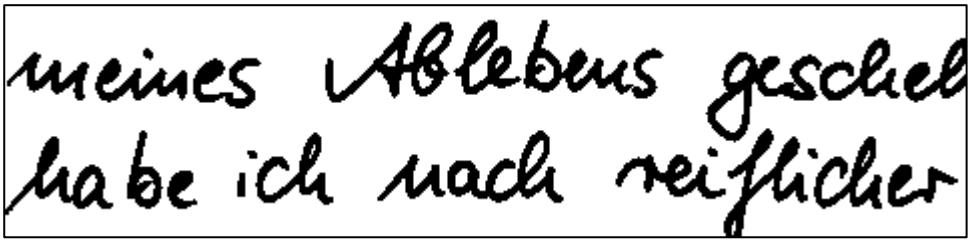
image respectively. First, the watermarked image was tampered by adding two zeroes at the end of the number "€500" to change it to "€50000". Second, the text "Leo €50000" was erased from the watermarked image. The third test is to combine the first and second alterations together. All these manipulations were done in Photoshop by simple copy, cut and paste operations. The altered versions look perfect and no noticeable trace is left on the image. The authentication and recovery result of every test is shown below the altered version. Two different kinds of manipulations are indicated in different colors: the deleted parts are indicated in red and the added ones in blue. From the result images, we can see that all the alterations are successfully detected and precisely localized. The deleted content is correctly recovered. Note that there are some blue dots in the red deleted parts and vice versa. These wrong color dots are caused by the unverified blocks that contain more than one altered pixels as we discussed in Section 4.4.

Another kind of manipulation, content replacement, is shown in Figure 4-17. The name "Markus" was removed and replaced by "Stefan". In this case, the added and deleted content are partly overlapped. The authentication and recovery result distinguishes the deleted name and the forged name successfully. The deleted name "Markus" is recovered in red color and the forged name "Stefan" is indicated in blue. It can be seen that in other areas there are still some noise-like red and blue dots that are not removed by the noise filter, but these noise dots do not affect the authentication result.

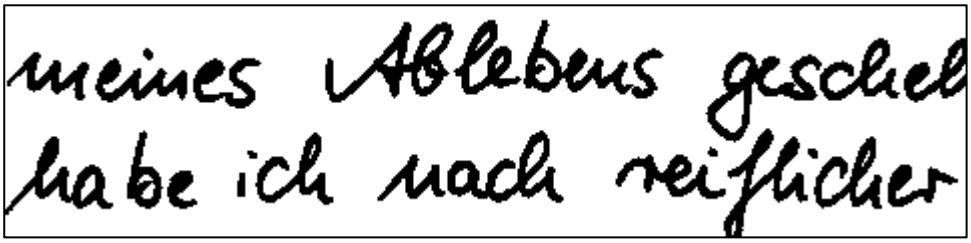
As we discussed in the previous sections, without noise filter the authenticator will give the highest sensitivity to any pixel manipulation. Figure 4-18 shows the above-mentioned four authentication results without noise filtering. All the red unverified pixels denote possible modifications from black to white, i.e. content deletion, while the blue unverified pixels denote possible modifications from white to black, i.e. content addition. As shown in Figure 4-18, without any noise filtering, it is still very easy to distinguish different kinds of content manipulation. The altered content parts are successfully localized and recovered with higher sensitivity. For example, in Figure 4-18 (a), the deleted comma is partly overlapped with the added zero, so the actually

changed part is very small because the overlapped part remains in black. Despite the overlapping and small alternation, the authenticator gives a very accurate localization and recovery result as shown in Figure 4-18 (a). The overlapping part is shown in black, which means no modification, while the protruding parts of the deleted comma are shown in red to reveal the deletion. The added comma can also be recognized in blue color.

Besides the handwritten text image, Figure 4-19 presents an example of the typed text images, which is a part of typical electronic archive copy from the library. As shown in Figure 4-19 (a) and (b), the watermarked image achieves a very good visual quality and there is no noticeable difference between the original text image and the watermarked one. Figure 4-19 (c) gives a close view of a part of the watermarked image. In Figure 4-19 (d), the last word “way” is deleted and the period position is also shifted to left accordingly. The manipulation can be easily distinguished by comparing Figure 4-19 (c) and (d). However, without knowledge of the original version and only from the tempered version Figure 4-19 (d), the manipulation is very successful and completely unnoticeable. The authentication and recovery result is shown in Figure 4-19 (e), the deleted word “way” and the period are recovered in red color, while the faked period is also marked out in blue color.



(a)



(b)

(c)

Figure 4-15 Close view of a part of watermarked image: (a) Original image, (b) Watermarked image, (c) Difference image, flipped pixels are shown in black.

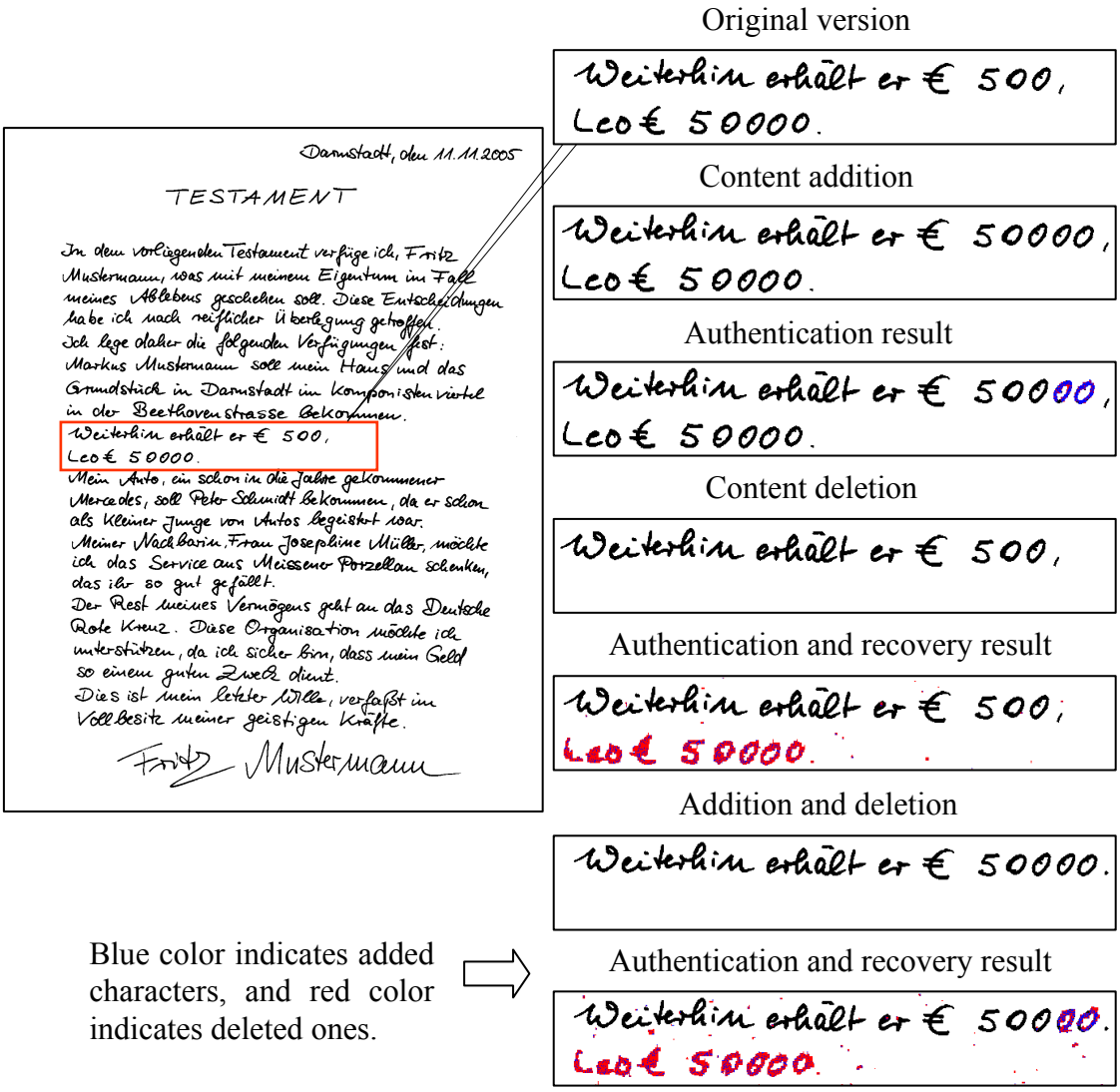


Figure 4-16 Handwritten text image test: content removal and addition. Left: watermarked image; Right: original version, different tampered versions and authentication results. Two zeros “00” is added at the end of “€500” and the text “Leo €5000” is deleted respectively. The detected result indicates the manipulations in different color: blue for addition and red for removal.

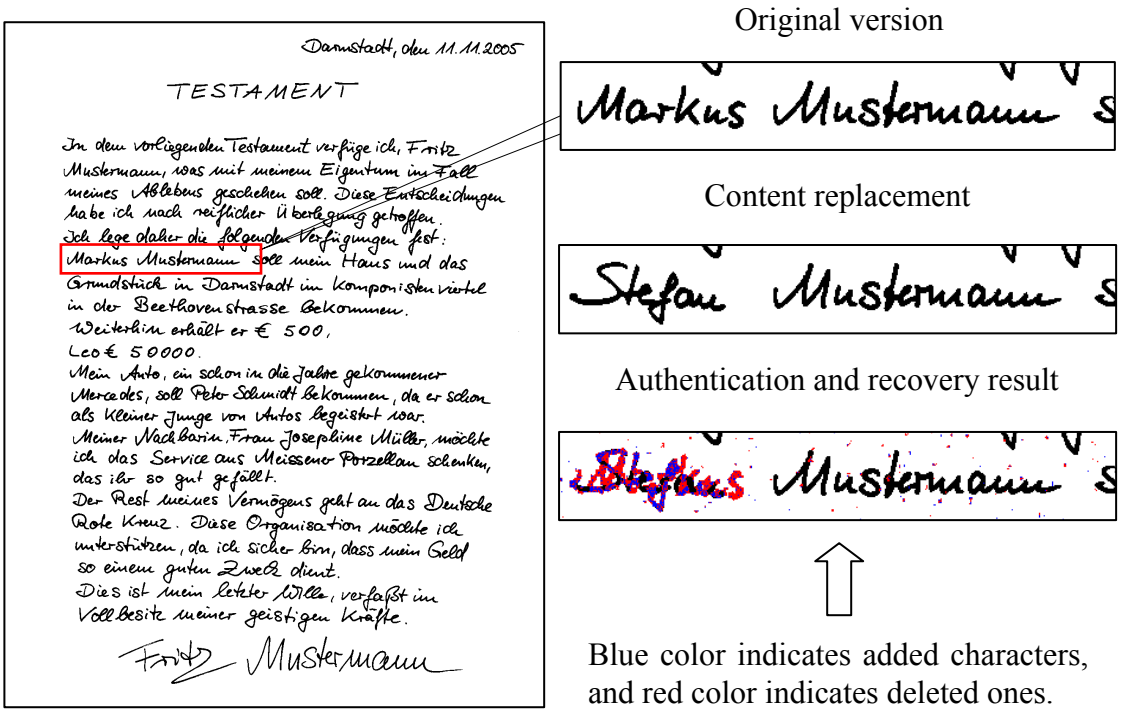
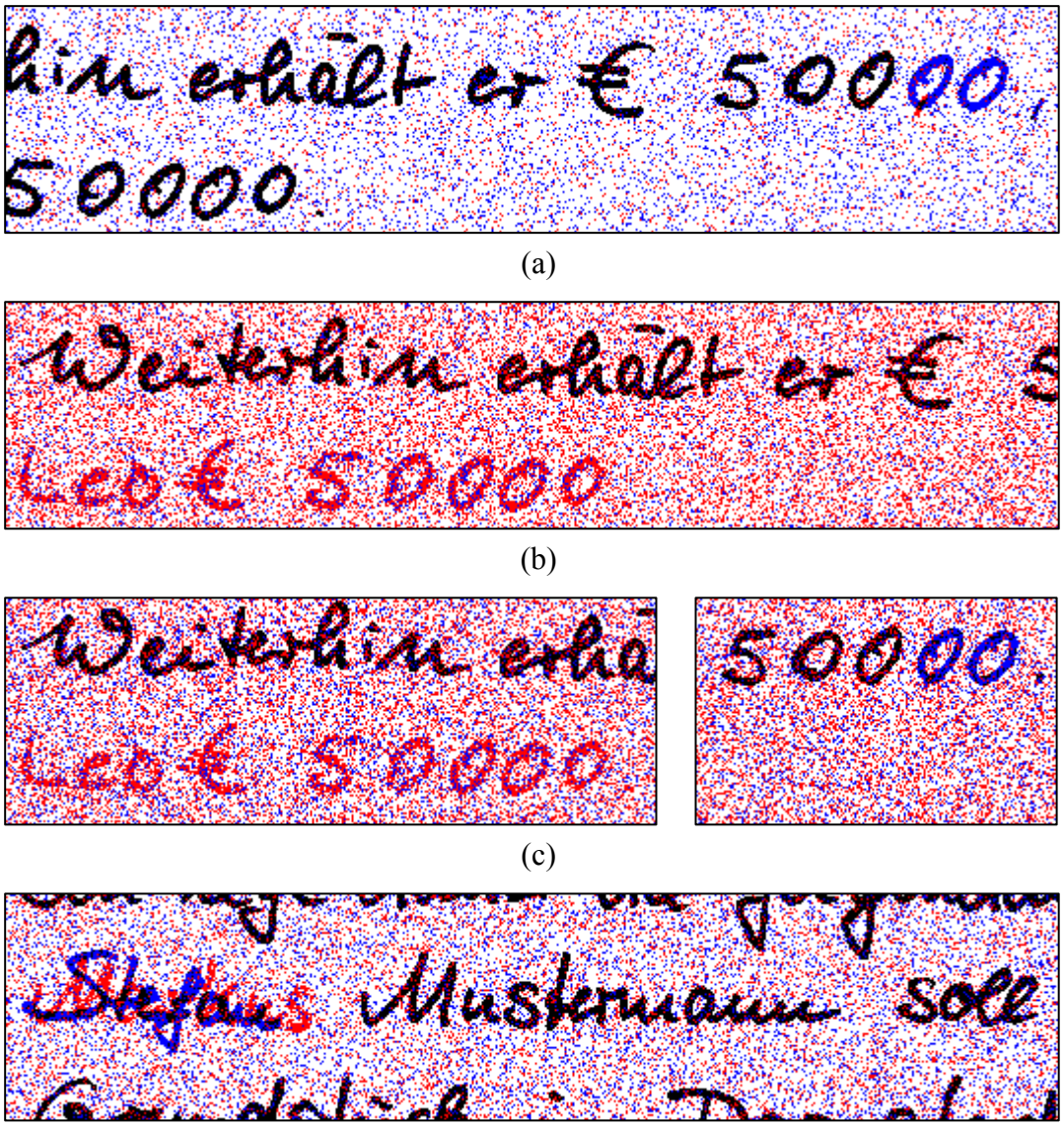


Figure 4-17 Handwritten text image test: content replacement. Left: watermarked image; Right: original version, tampered version and authentication result. The name "Markus" is replaced by "Stefan". The authentication result indicates the deleted "Markus" in red color while the forged "Stefan" in blue color.



(a)

(b)

(c)

(d)

Figure 4-18 Authentication results without noise filter. (a) Content addition, (b) Content deletion, (c) Combination of content addition and deletion, (d) Content replacement. The red unverified pixels denote possible modifications from black to white, i.e. content deletion, and the blue pixels denote possible modifications from white to black, i.e. content addition.

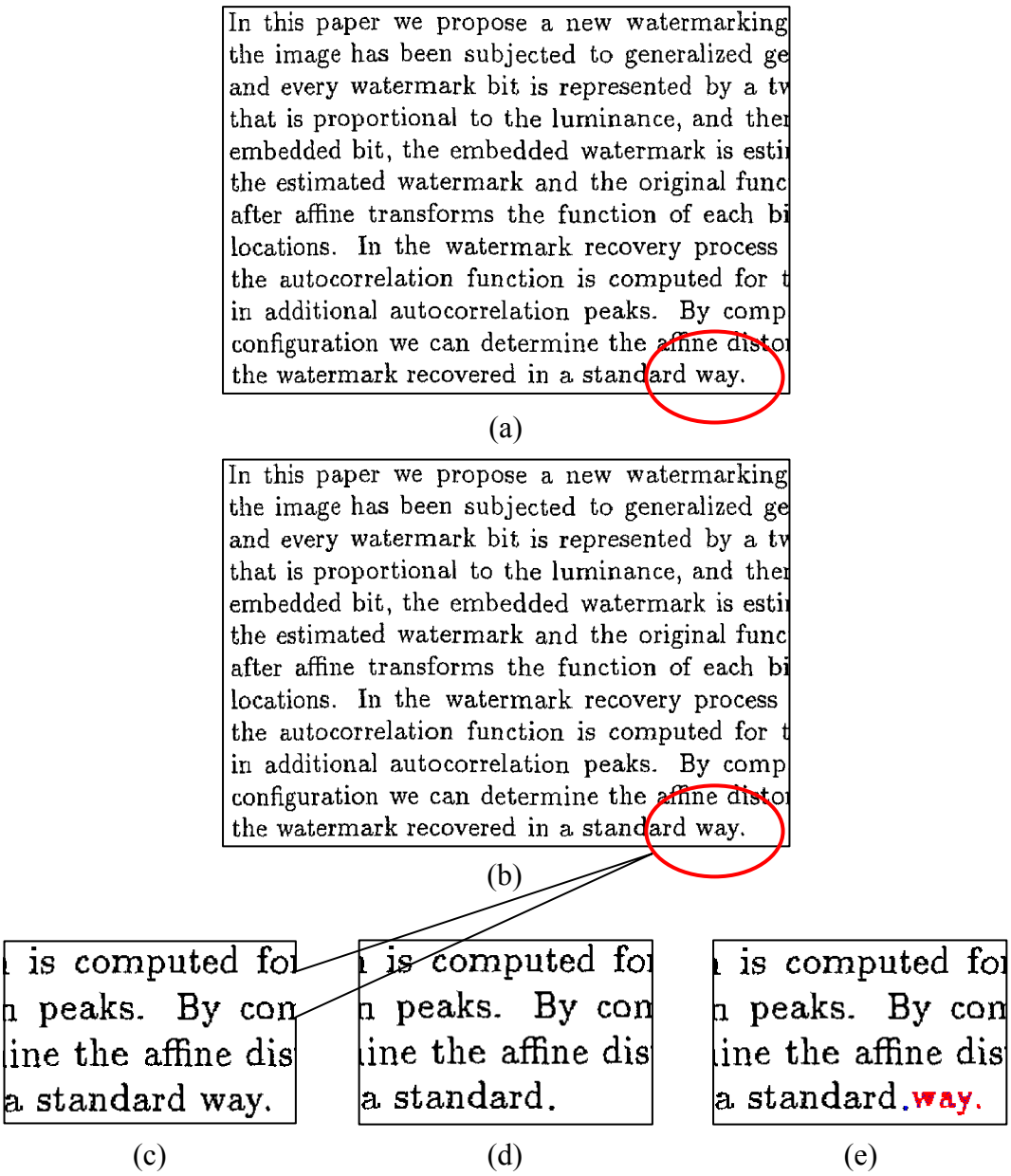


Figure 4-19 Binary typed text image test: content removal and addition. (a) Original text image, (b) Watermarked text image, (c) Part of the watermarked map, (d) Tampered version: the last word “way” is deleted and the period is then shifted to left. (e) Authentication and recovery result: the deleted word “way” and period are indicated in red color and the added period in blue color.

4.6.2 Color Digital Map

For color synthetic images, we take a digital map from the Media@Komm[1] project as example to present the experimental results. As shown in Figure 4-2, the example map contains seven distinct colors in total and is stored in palette image format. During watermark embedding, two background colors, white and green, are classified into the set $c2$, while the other five foreground colors are classified into the set $c1$. The number of foreground pixels in each block is used as the feature value in the watermark embedding process. Figure 4-20 (a) and (b) show the original digital map and the watermarked version respectively. Visually comparing these two maps, we can see that no annoying artifact is introduced by the embedding process. Figure 4-20 (c) shows a close view of part of the watermarked image. Deletion and addition manipulations are then made on the watermarked map. As shown in Figure 4-20 (d), one curve is deleted from the watermarked map, which is a modification from foreground color to background color, while another forged curve is added on the opposite side, which is a modification from background color to foreground color. After the image authentication process, both manipulations are precisely localized as shown in Figure 4-20 (e). The deleted curve is successfully recovered and is indicated in red color, and the forged curve is also revealed in blue color. As mentioned in Section 4.5.3, for color images, the authenticator can only recover the altered pixels to their original sets, but is not able to recover their original colors. In Figure 4-20 (e), red pixels indicate that their original color(s) belongs to the set $c1$, while blue pixels denote that their original color(s) belong to the set $c2$.

[1] http://www.mediakomm.net/.

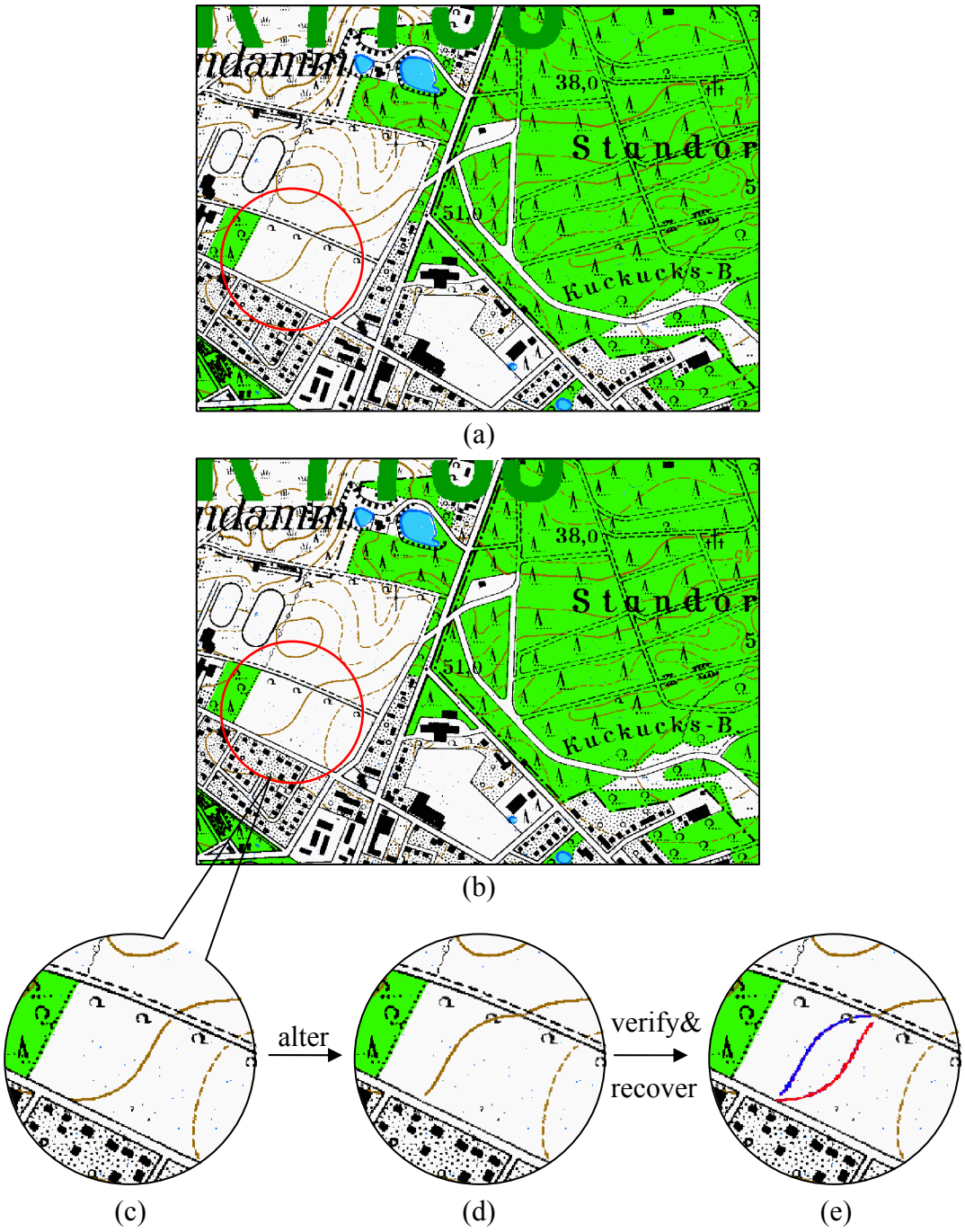


Figure 4-20 Color digital map test. (a) Original map, (b) Watermarked map, (c) Part of watermarked map, (d) Altered version: one curve was deleted and another one was added in a mirrored way, (e) Authentication and recovery result: the deleted curve is indicated in red color and the forged one in blue color.

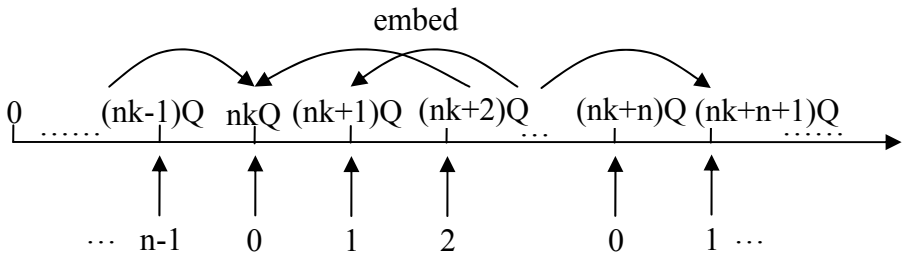


Figure 4-21 Illustration of the generalization of the proposed embedding method

4.7 Extension of the Proposed Embedding Method

In order to increase the pixel categories that can be distinguished in the recovery process, more dummy entries should be inserted between the possible watermark bit values "0" and "1". As illustrated in Figure 4-21, the proposed embedding method is generalized by modifying the period of the quantization entry from 3 to *n*. Thus, between the two "0" entries *n*kQ and (*n*k+*n*)Q there are (*n*-2) dummy entries. Hence, we can identify (n-1) kinds of Δ_f under the assumption that the mapping value is only changed by the minimum possible pixel modification. Note that if *n* is an even number, the entry "0" or "1" has the same distance to every dummy entry in both directions. In this case, we can not distinguish the direction of the modification. Hence, *n* has to be an odd number. Furthermore, in order to identify *m* categories of pixels in the recovery process, the feature value is required to be a certain characteristic of the block that takes into account all the *m* pixel categories. Modifying a pixel of any category should render the feature value changed. In addition, modifying pixels in different categories must cause different kinds of change of the feature value.

We take the case of *m*=3 as an example, in which three different pixel categories need to be identified. We denote these three category as *a*, *b* and *c*. Between the categories *a*, *b* and *c*, there are totally 6 possible kinds of pixel modifications: $a \rightarrow b$ and $b \rightarrow a$, $a \rightarrow c$ and $c \rightarrow a$, $b \rightarrow c$ and $c \rightarrow b$. We therefore need (*n*-1)=6 kinds of Δ_f to identify all these

possible pixel modifications, i.e. n=7. Based on the above-mentioned requirements, with n=7 and Q=1, a possible feature value can be defined as

$$f_j = Na_j + 2Nb_j + 4Nc_j \tag{4-17}$$

where f_j is the feature value in the jth block and Na_j, Nb_j, Nc_j denotes the number of pixels that belong to the category a, b and c in the jth block respectively. With the feature value f_j, any kind of pixel modification among the category a, b and c will cause different Δ_f. Under the assumption that only one pixel at most may be modified in a block, all the possible cases are listed in Table 4-5.

Table 4-5 List of the possible feature value changes and pixel modification

Case	$w(j)$	$\hat{w}(j)$	Δ_f	Pixel change	$I_o(x,y)$
1	0	1	+1	$a \rightarrow b$	a
2	0	2	+2	$b \rightarrow c$	b
3	0	3	+3	$a \rightarrow c$	a
4	0	4	-3	$c \rightarrow a$	c
5	0	5	-2	$c \rightarrow b$	c
6	0	6	-1	$b \rightarrow a$	b
7	1	0	-1	$b \rightarrow a$	b
8	1	6	-2	$c \rightarrow b$	c
9	1	5	-3	$c \rightarrow a$	c
10	1	2	+1	$a \rightarrow b$	a
11	1	3	+2	$b \rightarrow c$	b
12	1	4	+3	$a \rightarrow c$	a

It should be noted that using a bigger n will cause more pixel modifications during the watermark embedding process. In every single block, one proper pixel modification can enforce any feature value to the desired mapping value. With n=7, for example, the

maximal possible amount of feature value modification is 3 and a pixel flipping from category *c* to *a* can fulfill this maximal modification. Hence, the maximal number of pixels that need to be modified in a block is still bounded to one. Nevertheless, the total amount of pixel modification in the whole image will still be raised because the probability that the original feature value f_j maps to the desired watermark bit value is decreased from 1/3 to 1/*n*, even though we impose the constraint that only one pixel can be flipped in a block. Furthermore, since only one pixel is allowed to be modified in a block, in order to fulfill a successful embedding different kinds of flippable pixels are needed. For example, if the original feature value maps to "3" while the desired watermark bit is "1", a flippable pixel that can be flipped from category *c* to *a* is required. Therefore, the requirement of different kinds of flippable pixels is significantly raised.

4.8 Conclusion

Due to the simplicity of the content, it is more challenging to hide data invisibly in synthetic images. However, the same reason also renders them much easier to be manipulated. Most of the existing watermarking algorithms are only targeted for color or grayscale natural images and are not applicable for synthetic images. The capability of existing watermark techniques for synthetic image authentication is limited in tamper localization and none of them has the capability of recovering the altered original content.

In this chapter, after addressing the necessity and challenges of synthetic image authentication and reviewing the existing watermarking techniques for simple images, we propose a novel watermarking scheme to solve the problem. Random permutation is applied to establish random reference relationships among image pixels and it also equalizes the uneven watermark capacity over the whole image. A new embedding strategy is proposed to replace the classical odd-even embedding or look-up table embedding approaches. The new embedding method enables the recovery capability of original pixels in the image authentication process. Based on the distribution and

density of potential unverified pixels, the proposed scheme achieves pixel-wise tamper localization capability. Furthermore, the altered pixels can be recovered to their original values or color sets. A variety of experimental results demonstrate the effectiveness of the proposed scheme for synthetic image authentication. An extension of the proposed embedding method is also discussed.

Chapter 5 Image Authentication with Region of Interest (ROI)

5.1 Introduction and Prior Work

Digital watermarking verifies the image content by embedding additional information, referred to as the watermark, into the host image data. The watermark is embedded by slightly modifying the original data. Therefore, it is inevitable to change the host data in some way. Although the modification of the original content is strictly controlled to be so slight that it is commonly imperceptible to human eyes, most of the watermarking algorithms will still cause a certain amount of permanent loss of content fidelity during the embedding process. The quality loss is usually proportional to the amount of the embedded watermark information. As mentioned in the prior chapters, a high watermark payload is usually required for content authentication, so the quality degradation caused by the authentication watermark is consequently increased. In addition, as the integrity of every image part needs to be ensured, most of the existing watermarking schemes embed the watermark ubiquitously over the entire image area. As a result, the quality degradation also exists ubiquitously over the whole image. Since such degradation for the human observer is masked and minimized by using perceptual models, it may be accepted in many applications.

In some applications, however, the fidelity of the original image is of special importance, such as medical images, satellite images and military images. In these applications, even slight modifications are not acceptable, especially in some important image regions. The slight quality degradation caused by the watermark embedding becomes intolerable. For example, the reliability of the data, i.e. the integrity of the records, is an important issue for medical images, because any manipulation or quality compromise could result in serious misdiagnosis of the patient's disease [PM05][GPK06]. Thereby, in the most important parts of medical images, any slight modification is not allowed. In addition, satellite images also require high image fidelity. Slight quality loss might result in the deterioration of their commercial value, rendering it unfit for reuse or further distribution [CGM02].

On the other hand, in some other applications, only one or more particular regions in the image are suitable for watermark embedding, while the remaining parts have little or no watermark capacity, in which if a watermark is enforced to embed, severe quality loss will be rendered. For example, in the applications of identification, a Photo-ID card as a whole is considered as one picture. When there is no background image on the card, an ID card only includes a photo and a few lines of text and many margins. The text is printed on the card with very high resolution and can hardly be modified without introducing any artifacts. Therefore, only the region of the photo is suitable for watermark embedding. Technically, the above-mentioned two cases are the same, which are just inverse definition of special regions in different applications. Thereby they can be handled by the same solution.

Obviously, the common watermarking strategies, which embed the watermark ubiquitously in the whole image, can not satisfy such special applications. They can not process special image regions separately. One solution to satisfy the aforementioned special requirements is to use the so-called invertible watermarking technique. An invertible watermark can be removed from the image content after the extraction so that the original image data can be precisely recovered [FGD01][NSAS06][A03]. In invertible watermark techniques, some portions of the host signals, e.g. some pixels or

frequency coefficients, are compressed to provide additional space to store the net watermark payload and the original signal information [AK03][CSTS02]. Hence the total payload is usually high. Due to the high required payload, the drawback of many invertible watermarking algorithms is that the quality in the marked state is lower than that in most traditional watermarking algorithms. This means that only the owners of the fitting key can benefit from the invertible strategy, the rest of the users will suffer even more quality loss. Furthermore, the invertible watermarking solution can only be used in cases in which the image can be converted back to the original state. Usually this condition is true only in the digital world. For example, the invertible watermark is not feasible for the above-mentioned ID card application, because the picture of an ID card is rendered in an analog way, i.e. printed on the card. The quality loss will remain on the card permanently and the watermark can not be removed thereafter.

The other alternative solution is a watermarking technique that supports regions of interest (ROI). In the literature, some watermarking algorithms have been proposed combined with the concept of region of interest (ROI). In [LHLH03], Lie proposed a dual watermarking scheme for JPEG2000 images. One fragile watermark is embedded into the first wavelet level of the ROI and the other robust watermark is embedded into the third wavelet level of the ROB (region of backgrounds). By combining the dual watermarks, the scheme can distinguish malicious attacks from allowable image processing. In [CGM02], Chauhan et al. proposed a pixel-domain watermarking algorithm based on a look-up table method. A visually meaningful binary logo is embedded in original satellite images as the watermark while avoiding distorting certain vital regions. Two spatial-domain watermarking schemes for medical images were proposed in [W02] and [CWC05], in which the proposed schemes embed the signature information of the ROI into other non-ROI image parts so as to avoid distorting the image data inside the ROI. In [CWC05], the same watermarking technique was applied in the wavelet-domain and the watermark was embedded only into the non-ROI wavelet coefficients. Nevertheless, all these above-mentioned algorithms are either limited to a specific image format [LHLH03], or they need precise location information of the ROI in order to successfully extract the embedded

watermark [CGM02][W02][CWC05]. These requirements significantly decrease the practicability and the portability of these ROI-based watermarking schemes, because in the practical applications the ROI information may be often unavailable at the watermark detector side.

In this chapter, we first propose a framework for ROI-supporting watermarking systems. The framework extends the watermarking schemes proposed in Chapter 3 and Chapter 4 by introducing the concept of a Region of Interest. The proposed framework can also be applied to other different watermark embedding schemes as long as the watermark is embedded into the subsets of the image separately. Based on the framework, we modify our wavelet-based watermarking scheme in Chapter 3 so as to support regions of interest masking. The content inside the preferred ROI(s) is kept intact during the watermark embedding process, while its integrity is still ensured by the embedded watermark in the other parts of the image. No ROI information is required in the watermark extraction and image authentication processes. Experimental results demonstrate that the proposed solution can detect and localize the manipulations both inside and outside the ROI(s) with the same resolution. We also evaluate the effectiveness of the proposed strategy for extending the synthetic image watermarking scheme in Chapter 4 to support the ROI concept.

This chapter is organized as follows. Firstly, in Section 5.2, we give out the detailed definition of a region of interest. Then in Section 5.3, we introduce the proposed framework for watermarking with Region of Interest masking. The ROI-based watermarking scheme and the authentication processes are presented in Section 5.4. The performance of proposed scheme is discussed in Section 5.5 and experimental results are given in Section 5.6. In Section 5.7, we evaluate the effectiveness of the proposed framework for the watermarking scheme for synthetic image authentication. Finally, we conclude the chapter in Section 5.8.

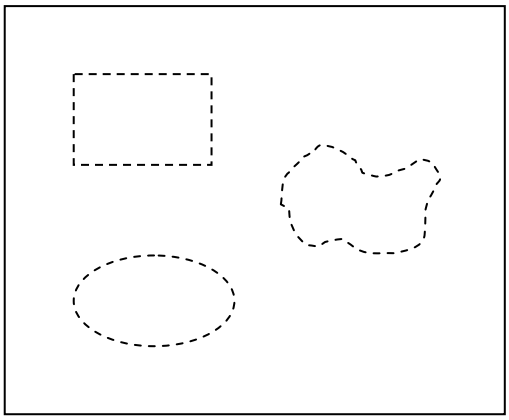


Figure 5-1 Examples of Region of Interest

5.2 Definition of Region of Interest (ROI)

The concept of Region of Interest (ROI) is widely used in various application fields, such as medical and biomedical applications, visual communication applications with limited available bandwidth, client/server communication for image browsing with preview functionality, camera-based applications for automatic object capture (e.g. in a seminar, class-room, or in a teleconference), and content-based image retrieval (CBIR). However, in different applications, ROI has slightly different meanings and definitions. Depending on the context, a ROI can be defined either simply as a rectangular subset of data, or as any combination of irregular shapes as shown in Figure 5-1, which contains the important parts of the data based on the targeted application. The common purpose of defining regions of interest is to enable special processing in these sub-regions that is not applied to the whole image area. For example, a formalization of the ROI concept is included in the JPEG2000 standard [J00], which represents the state of the art of the still image coding standards. JPEG2000 supports different progressive decoding modes, one of which is related to the ROI functionality of this format. In this context, a ROI is a part of the image with arbitrary shape, which is supposed to have a better quality at any decoding bit-rate than the rest of the image.

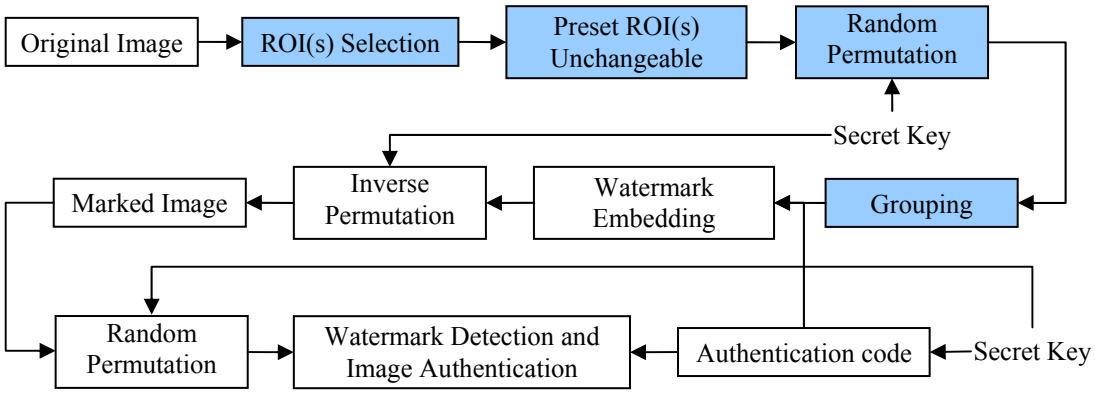


Figure 5-2 General Framework of ROI-supporting Image Watermarking

In our work the ROI concept is not limited to that in JPEG2000 images or any other specific application, so our proposed scheme can be applied to different kinds of applications. We use the general definition of the region of interest (ROI), by which a ROI is a part of the image with arbitrary shape and size as shown in Figure 5-1, which has special requirements in the specific application. In our work, no additional information of the defined ROI(s) needs to be stored along with the image, such as region boundary chain code or binary mask used in JPEG2000. Furthermore, specifying multiple regions of interest in one image is allowed. The defined ROI(s) will be treated differently during the watermarking process. For example, the data inside the ROI(s) will not be modified during watermark embedding. Nonetheless, the content of the ROI(s) is still protected like other image regions. In practical applications, the ROI(s) can be either predefined kind of objects like chromosomes in cytogenetic images, or areas with arbitrary shape that are interactively specified by the user before the watermark embedding.

5.3 Proposed Framework of Watermarking with Region of Interest

Figure 5-2 shows the proposed framework for ROI-supporting image watermarking. This framework shares the same idea of random permutation that is used in the

proposed watermarking schemes in Chapter 3 and Chapter 4. Nevertheless, this framework can be applied not only to the watermarking schemes that we proposed in the previous chapters, but also to other watermarking algorithms designed in both the spatial and transform domains as long as they embed the watermarks into separable image units, such as in a block-based way. Compared with the classical watermarking model, the proposed framework includes three additional steps before the embedding process: ROI(s) selection and presetting, random permutation and grouping.

5.3.1 ROI Selection and Presetting

Before performing the watermark embedding process, one or multiple preferred regions of interest are specified on the image. We denote the set of ROI(s) as R_{roi} and preset it as unchangeable so that the data inside R_{roi}, for example the pixels or the corresponding transform coefficients, can not be modified during the watermark embedding process. The locations of the ROI(s) are input to the watermark embedder as parameters. Then the classical watermark embedder is modified to

$$I_i^{'} = \begin{cases} I_i + w_i, & if\ I_i \notin R_{roi}, \\ I_i, & otherwise. \end{cases} \tag{5-1}$$

5.3.2 Random Permutation and Grouping

These two steps, random permutation and grouping, are used to establish a mutual reference relationship among all the locations of the image. Because the ROI is defined unchangeable, no watermark can be embedded in the ROI. The authentication of the ROI(s) must be achieved by the embedded watermarks in the others parts of the image. Therefore, how the ROI(s) and other image parts are linked together becomes an essential issue for the authentication process.

In the proposed framework, we first utilize a random permutation process to obtain a random mapping of all the image locations. The random permutation is performed before the watermark embedding in the corresponding domain where the watermark

will be embedded. The units that are used to embed the watermark, e.g. single pixel, transform coefficient or image block, are randomly permuted under the control of a secret key.

The permutated units are then divided into groups with a group size *g*. A group can be defined as any kind of combination of image locations, such as rectangle blocks, 1-D row or column vector, etc. This grouping step determines the subsets in which the locations refer to each other. Due to the random permutation process, the members of every group come from different random locations. Likewise, all the locations inside ROI(s) are also randomly distributed into different groups, mutually referred to other locations in the same group. During the watermark detection and image authentication procedure, the reference relationship is recovered by applying the same random permutation using the same secret key. Therefore, no location information of the ROI(s) needs to be stored or transfered.

5.4 Watermarking and Authentication Processes

5.4.1 Watermark Embedding and Detection

We first adopt the watermarking scheme of Chapter 3 to the proposed framework. Since the watermark is embedded in the wavelet domain, the specified ROI(s) locations have to be first mapped from the spatial domain to the wavelet domain. Thanks to the spatial-frequency localization of the wavelet transform, this mapping can be easily accomplished. Let (x_l, y_t) and (x_r, y_b) be the coordinates of the top-left and bottom-right corners of a ROI, then the corresponding coordinates of the ROI's top-left corner in each wavelet decomposition level can be calculated as

$$\begin{cases} (x_l / 2^r + W / 2^r, y_t / 2^r) & \textit{in horizontal subband}, \\ (x_l / 2^r, y_t / 2^r + H / 2^r) & \textit{in vertical subband}, \\ (x_l / 2^r + W / 2^r, y_t / 2^r + H / 2^r) & \textit{in diagonal subband}, \end{cases} \tag{5-2}$$

where r is the wavelet decomposition level. W and H denote the image width and height respectively. The coordinates of the right-bottom corner of the ROI can be calculated likewise. All the wavelet coefficients inside the ROI(s) are set to be unchangeable and will be skipped by the embedder later.

All the wavelet coefficients in the selected decomposition level r, including the ROI(s) coefficients, are firstly concatenated into a single string and then randomly permuted in the same way as described in Chapter 3, controlled by the secret key. Finally the rearranged coefficients are divided into groups of size g. Since the ROI coefficients are also permuted together with other non-ROI coefficients, they will be evenly distributed into all the groups. As a result, some groups will contain a mixture of both kinds of coefficients. We name the groups containing at least one ROI coefficient *ROI-group* and others *non-ROI-group*. The weighted mean value s_j of every group is obtained and quantized according to Equation (3-1)-(3-4) to embed the corresponding watermark bit.

In the coefficient update process, Equation (3-9) is modified to

$$f^*_{j,\max}(i) = \begin{cases} f_{j,\max}(i) + p_i \cdot sign(f_{j,\max}(i)) \cdot \delta_j, & if\ f_{j,\max}(i) \notin R_{roi}, \\ f_{j,\max}(i), & otherwise. \end{cases} \tag{5-3}$$

Equation (5-3) ensures that the coefficients inside ROI(s) will not be changed. All the necessary modifications for watermark embedding are made on the non-ROI coefficients in the group.

Likewise, if the second coefficient update method is adopted, Equation (3-10) should be modified similarly as follows:

$$f^*_j(i) = \begin{cases} f_j(i) + p_i \cdot sign(f_j(i)) \cdot \dfrac{\left|f_j(i)\right|}{\sum_{f_j(i) \notin R_{roi}} \left|f_j(i)\right|} \delta_j, & if\ f_j(i) \notin R_{roi}, \\ f_j(i), & otherwise. \end{cases} \tag{5-4}$$

By applying either Equation (5-3) or (5-4), if the sign of $f_j(i)$ is changed after the update, $f_j(i)$ is set to zero in order to avoid severe image distortions.

The watermark detection process is performed in the same way as non-ROI watermarking. The ROI information is not required by the watermark detector. After the recovery of the random permutation of the selected wavelet coefficients, the watermark bit in every group is extracted by Equation (3-14). Note that the ROI(s) coefficients contribute to the weighted mean in both the embedding and detection processes, although they are not modified to embed the watermark. Therefore, tampering the ROI(s) will cause the weighted mean to change and subsequently change the quantization result, leading to a mismatch between the extracted watermark bit and the embedded one. Therefore, the tampering inside the ROI(s) can be detected by the watermark.

5.4.2 Image Authentication

The image authentication process for the non-ROI watermarking in Chapter 3 can still be used here. By comparing the extracted watermark bit with the original one, we find out all the unverified groups where mismatches occur. All the coefficients in the unverified group are marked as unverified coefficients regardless of whether they are from inside or outside ROI. If the tampering occurs inside the ROI(s), as we discussed in the previous section the corresponding ROI-groups will be detected as unverified. Consequently, the ROI coefficients belonging to these groups are identified as unverified as well as the non-ROI coefficients. After all these coefficients are mapped back to their original positions, the unverified ROI coefficients will be clustered in the tampered area inside the ROI, while the other non-ROI coefficients will scatter over the other regions sparsely. Thus, the tampering can be localized based on the density of the unverified coefficients and can be easily filtered out in the same way as we discussed in Chapter 3. On the contrary, if the tampering occurs outside the ROI(s), the involvement of ROI(s) will not affect the authentication process and it will be the same as the non-ROI watermarking case we discussed in Chapter 3. Note that the same localization

resolution can be achieved for both the tampering inside and outside the ROI(s). For example, if the watermark is embedded in the *r*th decomposition level, the localization resolution will be $2^r \times 2^r$ regardless of where the tampering occurs.

Note that the authenticator does not need to know the existence of the ROI(s). The authentication process can be performed in the same way as the non-ROI watermarking case without any knowledge of ROI(s). The involvement of ROI(s) will not affect the tamper localization resolution for the whole image.

5.5 Performance Analysis

5.5.1 Quality of Watermarked Image

From Equation (5-3) and (5-4), we can see that the maximal modification of the wavelet coefficients in one group is still bounded to δ_j, the same as in Equation (3-9) and (3-10). This reveals that, compared to the non-ROI watermarking case, the involvement of ROI(s) does not introduce more modifications during the watermark embedding. Therefore, the overall image quality, measured by PSNR, will not be more degraded than that with non-ROI watermarking.

The involvement of the ROI(s), however, will decrease the reliability of the embedded watermark. Because all the data inside the ROI(s) are enforced to be unchangeable, the total watermark capacity of the image is decreased accordingly, while the required watermark payload still remains the same when applying the same group size. Therefore, when the remaining watermark capacity is lower than the required watermark payload, some watermark bits can not be embedded completely as required by Equation (3-4). This problem will happen when all the embeddable coefficients in a group can not bear the required modifications of the weighted mean s_j. As a result, the value of the weighted mean s_j can not be moved to the corresponding middle position of the quantization intervals as shown in Figure 3-5.

In Equation (5-3) or (5-4), due to the constraint that if the sign of $f_j(i)$ is changed after the modification $f_j(i)$ is set to zero, the actual modification amount applied to a non-ROI coefficient $f_j(i)$ is bounded to

$$M_j(i) = \begin{cases} \min(|f_j(i)|, |\delta_j(i)|), & \text{if } p_i \cdot sign(\delta_j) < 0, \\ |\delta_j(i)|, & \text{if } p_i \cdot sign(\delta_j) > 0, \end{cases} \tag{5-5}$$

where $\delta_j(i)$ denotes the proportion of the total modification that is assigned to the coefficient $f_j(i)$. Therefore, in the case of $p_i \cdot sign(\delta_j) < 0$, if $|\delta_j(i)| > |f_j(i)|$, then the watermark can not be completely embedded. Such an incomplete embedding will lower the watermark robustness against minor distortions and even lead to a detection failure. When an incomplete watermark embedding occurs, the overall image quality might become better in that fewer modifications are made to the image.

Figure 5-3 plots the simulation result of the watermarked image quality for different ROI sizes. The test image set of Section 3.6, including 1086 images of various kinds, is used in the simulation. The PSNR values plotted in Figure 5-3 are the average results of all these 1086 watermarked images for different ROI sizes. In the simulation, the first wavelet decomposition level is selected to embed the watermark and the quantization step Q and the group size g is set to 6 and 12 respectively. The specified ROI size varies from 0% to 92% of the original image size. The ROI area is always selected in the center of the image. From Figure 5-3, we can see that the ROI size has little effect on the PSNR of the watermarked image. The PSNR becomes a little bit higher when the ROI takes up 60% of the image size or more, especially after the ROI size is bigger than 80%. As analyzed above, this is because with increasing the ROI size the remaining area at the sides of the image becomes smaller and smaller and accordingly fewer and fewer modifications can be made to the image, namely, the incomplete embedding occurs more and more frequently. With the incomplete embedding increasing, the watermark reliability will accordingly decrease.

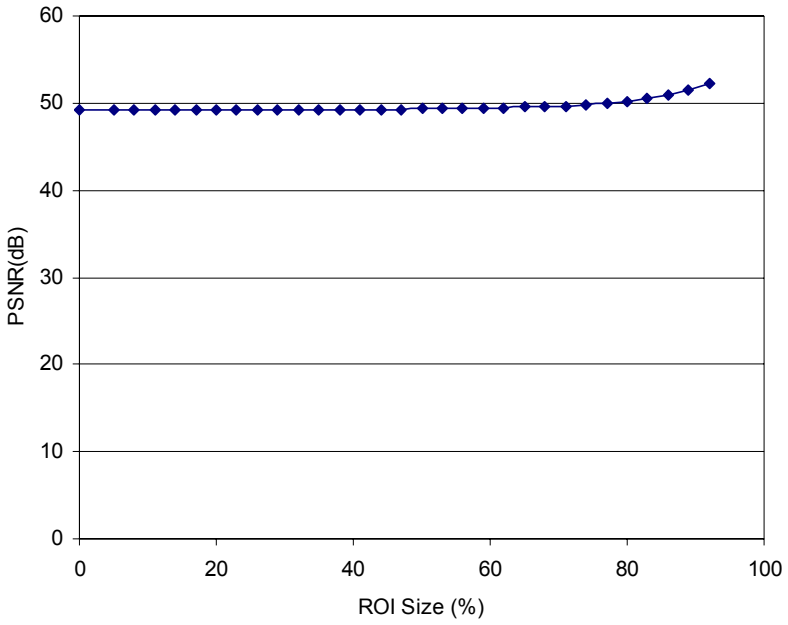


Figure 5-3 Watermarked image quality for different ROI sizes. The PSNR values are the average of the 1086 test images.

To illustrate the effect of incomplete and unsuccessful embeddings, Figure 5-4 plots the curve of the bit error rate (BER) of the watermark detection versus different ROI sizes. All the BER values in Figure 5-4 are the average results of all the 1086 test images for different ROI sizes. From Figure 5-4 we can see that when the ROI(s) area takes up a percentage of the image size as high as 60% and more, the watermark detection error rate rises significantly. When the ROI size is larger than 80%, the slope of the curve becomes bigger, i.e. the BER rises even more rapidly. These results conform to the image quality discussion above very well and explain the image quality improvement in Figure 5-3 for large ROI sizes. They also reveal that, due to the limit of the watermark capacity of the image, the ROI size must be restricted in order to ensure a specific rate of successful embedding. In the next section, we will discuss the maximal size of the ROI(s) area with a given BER threshold.

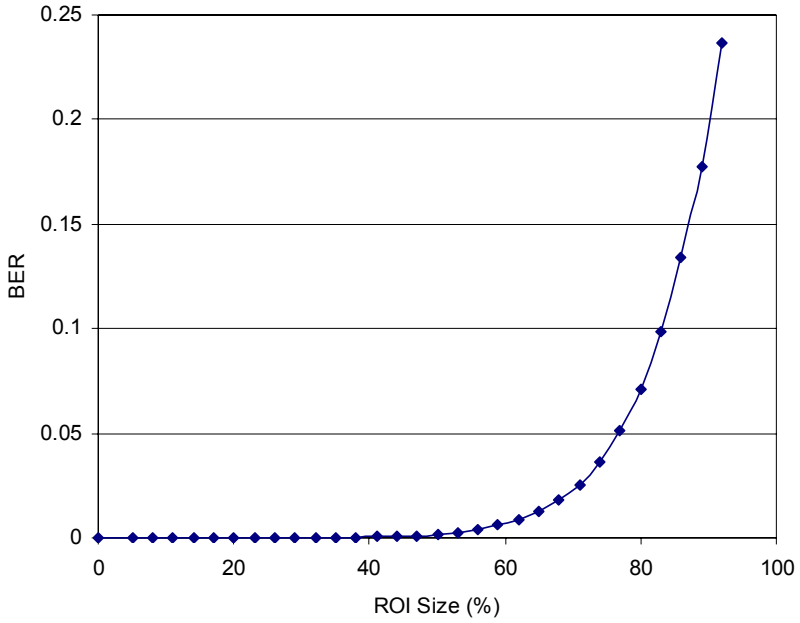


Figure 5-4 Watermark detection error rate for different ROI sizes. The BER values are the average of the 1086 test images.

Note that the visual quality of the non-ROI regions might be degraded, though the total modification of the cover image doesn't change. This is because the introduction of ROI(s) decreases the selectivity of the coefficients to be modified. As we mentioned in Chapter 3, thanks to the random permutation, the coefficients that are most suitable for embedding watermark is distributed evenly into all the groups. The modification of these coefficients will cause less visible artifacts on the image. In the ROI-groups, however, the most suitable coefficients of the group for modification might belong to the selected ROI(s) area and have been preset as unchangeable. In this case, other coefficients in the group have to be used to carry the modification. As a result of such suboptimal embedding, the visibility of the watermark will be increased.

5.5.2 Limit of ROI Size

As discussed in the previous section, with increasing the size of ROI(s), incomplete embeddings will happen more frequently and the watermark detection error rate will subsequently increase. In the extreme case, the ROI(s) takes up such a large image area that the remaining region has a too low watermark capacity to perform the embedding successfully. In other words, the total amount of the modifications applied to the non-ROI coefficient in a group is still not enough to move the weighted mean s_j to the correct quantization interval as depicted in Figure 3-5, so the watermark embedding fails. The same problem will also happen when the ROI(s) area takes up most of the regions that contribute the most watermark capacity of the image, such as the textured areas, though its area is not that large. It is well known that the textured image area has much higher watermark capacity than the smooth area with the same watermark invisibility and robustness. Hence if the remaining regions include only smooth areas it also might have no enough capacity to carry the necessary watermark payload. According to what we discussed above, the remaining non-ROI must have enough watermark capacity in order to ensure the watermark to be embedded successfully.

Theoretically, at least one non-ROI coefficient in every group must be guaranteed in order to have the least space to carry the modification for the watermark bit embedding. If we assume that the random permutation ideally distributes the non-ROI coefficient evenly into all the groups and every non-ROI coefficient is large enough to carry the necessary modification for embedding, the theoretical maximal ROI(s) area, which can be specified on an image of size $W \times H$, can be calculated as

$$L_{roi} = \frac{g-1}{g} WH \,. \tag{5-6}$$

Equation (5-6) reveals that the group size g determines maximal limit of the ROI size. A larger ROI area can be specified with a bigger group size. Nevertheless, Equation (5-6) only describes the ideal case. With a ROI area of size L_{roi} specified, every group is assigned just one non-ROI coefficient and (g-1) ROI coefficients. In this ideal case,

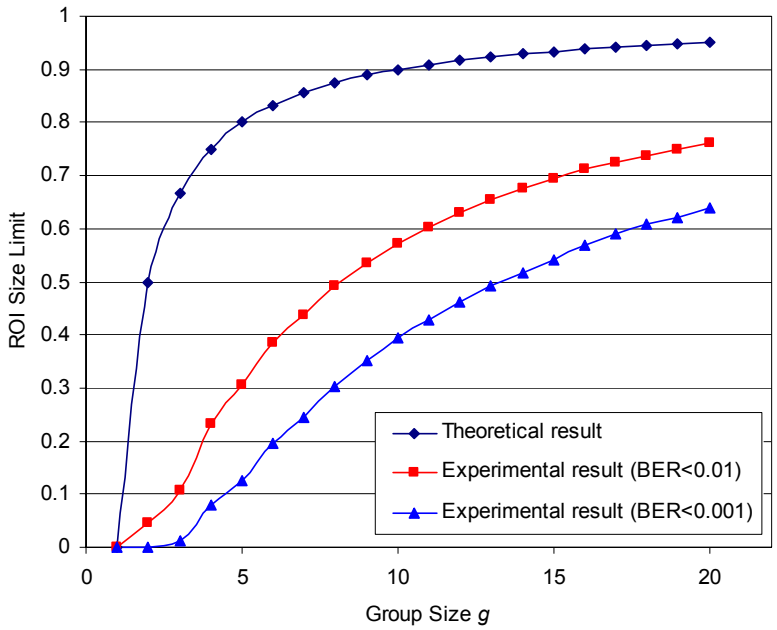


Figure 5-5 Comparison of theoretical and experimental results of ROI size limit. The experimental values are the average of the 1086 test images.

we assume not only the random permutation can ideally distribute the non-ROI coefficients evenly into every group but also every single non-ROI coefficient can bear the total modification that the watermark bit of the group requires in Equation (5-3) or Equation (5-4). Only when these two assumptions hold at the same time, can we ensure that all the watermark bits can be successfully embedded. The second assumption depends strongly on the characteristics of the image content and is rarely true in most images. Even if the non-ROI area is full of textures, it still can not be guaranteed that the magnitude of every non-ROI coefficient is large enough to take the total modification. Therefore, the actual maximal size limit of ROI(s) is usually quite smaller than the value L_{roi} obtained from Equation (5-6).

In Figure 5-5 we plot the theoretical ROI size limits obtained by Equation (5-6) and the simulation results from the image test set. All the experimental values in Figure 5-5 are the average results of all the 1086 test images for different group size. The same parameters are used as in the simulation of image quality. The first wavelet level is

selected to embed the watermark and the quantization step is set to 6. In the simulation, we obtain the maximal ROI sizes with the watermark BER below 0.01 and 0.001 respectively. The group sizes vary from 1 to 20. When the group size is 1, every wavelet coefficient has to be modified to carry one watermark bit. Therefore no ROI can be specified in this case. The maximal ROI size increases rapidly when the group size increases from 1 to 10. For example, it rises from 0 to 61% when the BER is lower than 0.01. And thereafter the slope of the curve becomes smaller and the ROI size limit rises slowly with the increase of the group size. This is because for relatively large ROI sizes the watermark capacity of the remaining area is rather limited and has fewer capacity surpluses for more ROI area. Figure 5-5 shows that with the group size of 12 that we apply in the previous simulations, we can specify ROI(s) areas nearly as large as the half of the image size while keeping the BER below 0.001. From Figure 5-5, we can see that the curves of the simulation results are always below the theoretical one, coinciding with the analysis conclusion above. With higher watermark reliability (BER<0.001), the maximal ROI size is further decreased. As shown in Figure 5-4, when the ROI size takes up 92% of the image size, the theoretical maximal limit according to Equation (5-6), the watermark detection error rate rises to 0.2369. Such a high error rate is not acceptable in practice because it will cause a too high false alarm rate of tampering detection.

5.6 Experimental Results

In this section, we first evaluate the watermark embedding performance with ROI masking. Second, the tamper localization capability of the proposed ROI watermarking scheme is tested. In the following experiments, the same test image set as in Section 3.6, which includes 1086 images of various kinds, is used. The watermark is embedded in the first level of wavelet decomposition, i.e. r=1, and the quantization step Q and the group size g is set to 6 and 12 respectively. In the watermark embedding process, Equation (5-3) is adopted to update the wavelet coefficients.

First, we embed the watermarks into all the 1086 images in the test set respectively, with a ROI that takes 20% of the cover image size. The ROI areas are always specified in the center of the images. Figure 5-6 plots the distribution of the quality of the 1086 watermarked test images in PSNR values and Figure 5-7 plots the distribution of the BER of the watermark detection. As can be seen from Figure 5-6, the PSNR values of the watermarked images distribute tightly around the peak that corresponds to the PSNR value of 49.13dB, very near to the average value of all the images 49.17dB. Similarly, in Figure 5-7 most of the BER values distribute near to zero and the number of images sharply decreases near to zero when the BER is higher than 1.0×10^{-3}. In total, in 16 images the BER values are higher than 1.0×10^{-3} and only in 2 images the BER values are higher than 2.35×10^{-3}. The average BER of all test images is 1.20×10^{-4}.

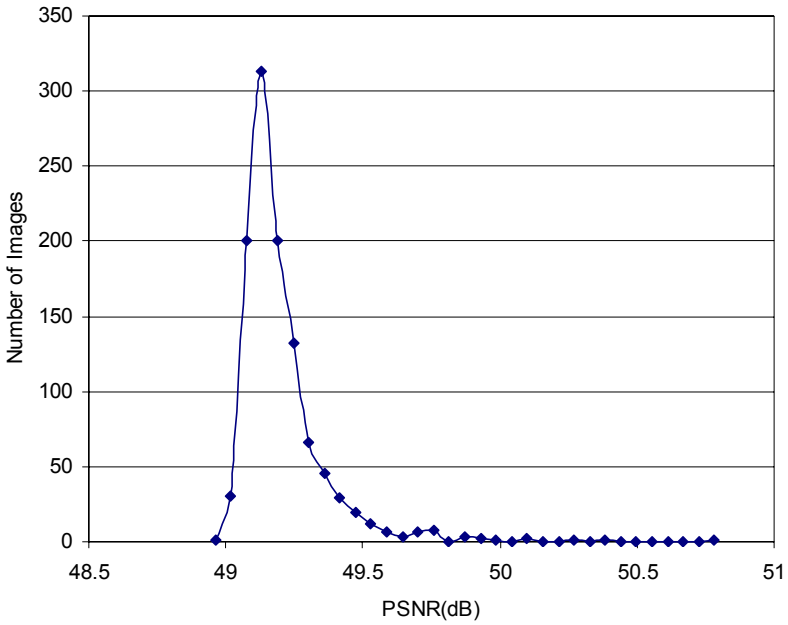


Figure 5-6 Image quality distribution (PSNR values) of the 1086 watermarked test images with ROI masking of 20% of the cover image size.

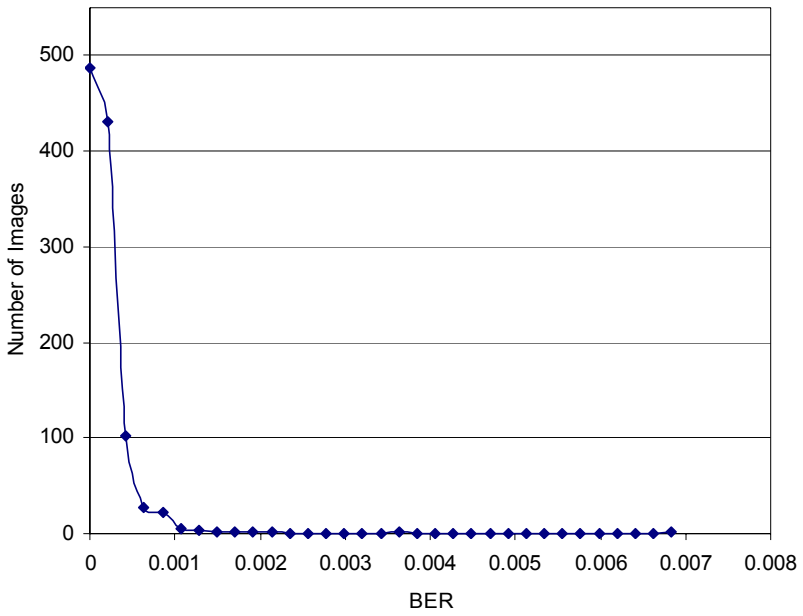
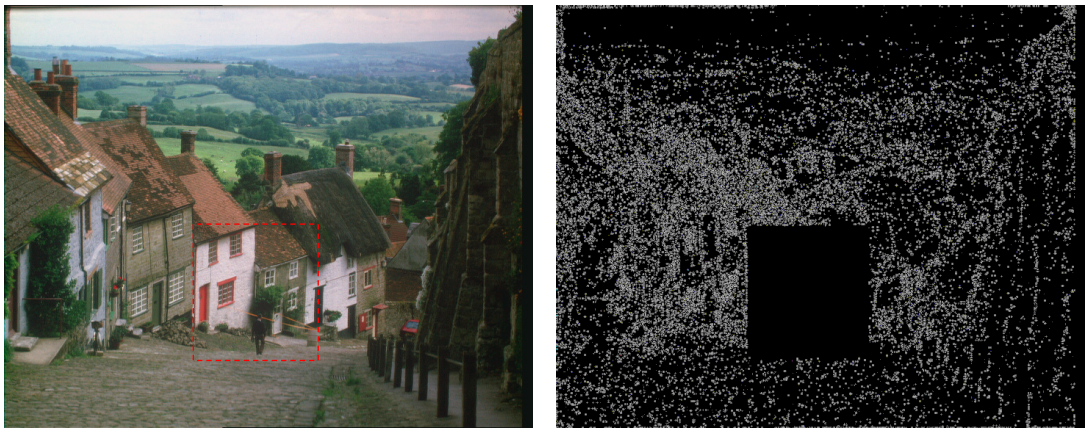


Figure 5-7 BER Distribution of the watermark detection from the 1086 watermarked test images with ROI masking of 20% of the cover image size.

Second, to evaluate the tamper localization capability of the proposed ROI watermarking scheme, we take the image "Gold hill" of 576x720 pixels as an example as shown in Figure 5-8 (a) to give the detailed experimental results. We specify one ROI of size 165×180 pixels starting from the coordinate of (260,300) and ending at the coordinate of (425,480), indicated by a red dashed rectangle in Figure 5-8 (a). The PSNR of the watermarked image is as high as 49.1dB. The embedded watermark is completely imperceptible under the normal viewing condition. The difference of the original and watermarked images is shown in Figure 5-8 (b). For the display purpose, the difference is magnified 30 times and the contrast is enhanced. It can be clearly seen that in the corresponding ROI area there is no modification caused by the watermark embedding.

After obtaining the watermarked image, we test the proposed scheme's tamper localization capability by deleting two objects inside and outside the ROI respectively. We first remove the person on the street that locates inside the ROI, as shown in Figure

5-9 (a). The yellow ellipse indicates the tampered position. The authentication result is shown in Figure 5-9 (b). The removed person is successfully detected and precisely localized, depicted in white color. Then we manipulate a non-ROI area on the watermarked image. As shown in Figure 5-10 (a), one window of the house, which is outside the ROI, is replaced by the wall around. Figure 5-10 (b) shows the authentication result. The tampered part is also correctly detected and localized. Both tampered areas are localized with the same resolution of 2×2 pixels because the watermark is embedded in the first level of the wavelet decomposition. We also apply these two manipulations on the image at the same time. Figure 5-11 shows the tampered image and the authentication result. It demonstrates when both the ROI and non-ROI areas are simultaneously tampered the proposed scheme can still correctly detect and localize the manipulations.



(a) (b)

Figure 5-8 ROI-based watermarking result. (a) Watermarked image (PSNR=49.1dB, the red dashed rectangle indicates the specified ROI position), (b) Different image (magnified 30 times and contrast enhanced for the display purpose).

(a) (b)

Figure 5-9 Authentication result when the ROI area is tampered. (a) Tampered image. Inside the ROI, the person on the street is deleted, indicated by the yellow ellipse, (b) Authentication result of (a), the localized tampered area is indicated in white color.

(a) (b)

Figure 5-10 Authentication result when the non-ROI area is tampered. (a) Tampered image. Outside the ROI, one window of the house is removed, (b) Authentication result of (c), the localized tampered area is indicated in white color.

(a) (b)

Figure 5-11 Authentication result when both the ROI and non-ROI areas are simultaneously tampered. (a) Manipulations both inside and outside the ROI, (b) authentication result of (a), the localized tampered area is indicated in white color.

5.7 Synthetic Image Authentication with ROI

As mentioned in Section 5.3, the proposed framework can be applied to all watermarking schemes, as long as they embed the watermark into separable units. Hence, besides the wavelet-based algorithm we introduced above, the proposed framework can also be applied to the watermark scheme for synthetic image authentication that we proposed in Chapter 4. In this section, we introduce and evaluate the synthetic image watermarking scheme with ROI support.

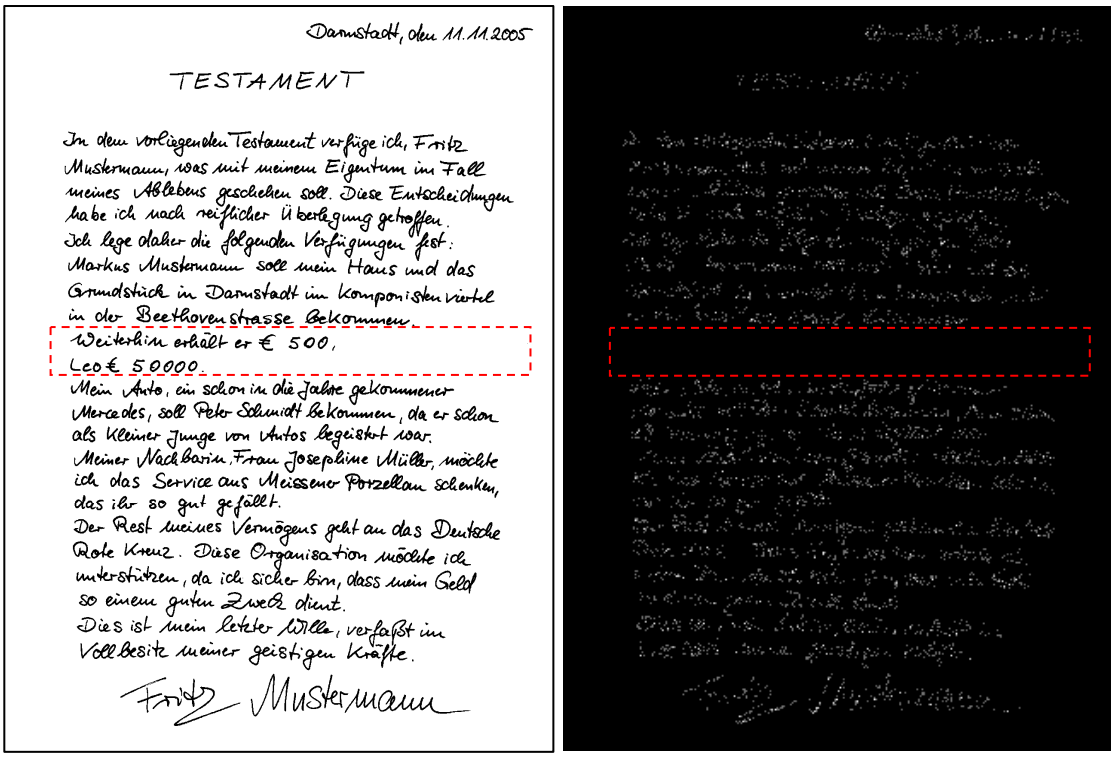
Extending the watermarking algorithm in Chapter 4 to support ROI is straightforward as follows. In the step of pixel classification, all the pixels inside the specified ROI areas are deemed as non-flippable pixels regardless of their properties of smoothness and connectivity in local neighborhood. This step ensures that all the ROI pixels will not be modified in the embedding process. After randomly permuting all the pixels and dividing them into blocks, we name those blocks that contain at least one ROI pixel *ROI-block* and others *non-ROI-block*. In each *ROI-block*, the block feature, i.e. the

number of black pixels or pixels belong to color set $c1$ for color images, is obtained by counting all the pixels in the block, including both the ROI pixels and non-ROI pixels. Thus, any manipulation of either ROI pixels or non-ROI pixels will change the block feature.

The watermark detector does not require any ROI information. The retrieval process remains the same as in Section 4.3.4 and the watermark bits can be extracted from each block by applying Equation (4-6). After extracting all the watermark bits, the image authentication and pixel recovery processes in Section 4.4 can be applied to localize the tampered regions and recover the manipulated pixels, which do not need any knowledge of ROI(s) either. Both tampering inside and outside ROI(s) can be detected and localized with the same resolution as in non-ROI watermarking.

Similarly as we discussed in Section 5.5, although the involvement of ROI(s) does not affect the image quality, the size of ROI(s) is limited by the watermark capacity of the non-ROI image portion. In the case of synthetic image watermarking, this capacity refers to the total number of flippable pixels in the non-ROI region. When the ROI area becomes larger, the number of flippable pixels will decrease so that the bit error rate (BER) of watermark detection will rise due to the increasing unsuccessful embedding rate.

Figure 5-12 shows the experimental results of ROI-based watermarking and authentication results for synthetic images. As shown in Figure 5-12 (a) a ROI is specified in the middle of the text image, which is indicated by the red dashed rectangle. Two lines of text are inside the ROI. Figure 5-12 (b) displays the difference image between the original image and the watermarked image. For display purposes, the different is magnified 30 times and the contrast is enhanced. It can be clearly seen that the ROI area keeps intact after watermark embedding. Text manipulations inside and outside the ROI area are shown in Figure 5-13 (a) and (c) respectively. Inside the ROI the text “Leo €5000” is deleted and outside the ROI the name “Markus” is replaced by “Stefan”. Authentication results are given in Figure 5-13 (b) and (d). As can be seen, the authentication results are as good as in non-ROI watermarking case.



(a) (b)

Figure 5-12 ROI-based watermarking result of the text image. (a) Watermarked image with ROI, the red dashed rectangle indicates the specified ROI position, (b) Different image (magnified 30 times and contrast enhanced for the display purpose).

Both the text deletion and replacement are correctly localized and recovered. In the authentication results, the deleted text is indicated in red color and the forged text in blue color.

5.8 Conclusion

Image fidelity is of essential importance in some special applications, so the slight modification caused by watermark embedding becomes not desired or even unacceptable, especially in some important image regions. Therefore, a non-ubiquitous watermarking solution is required to preserve the image quality in important regions. In this chapter, we proposed a watermarking framework for image authentication with

region of interest (ROI) masking. The watermark embedding is accomplished by only modifying the image content outside ROI(s). Therefore, the selected ROI(s) area is kept intact during the watermark embedding process so that it can satisfy the requirement of high fidelity of these important image areas in some special applications like medical and satellite images. Although no watermark is embedded inside, the integrity of ROI(s) is still protected. The proposed framework is evaluated by the watermarking scheme proposed in the previous chapters. Experimental results demonstrate that the proposed image authentication scheme with ROI is able to detect and localize the manipulations both inside and outside the ROI areas with the same resolution. Besides the presented watermarking schemes, the framework can also be applied to other watermarking schemes that embed the watermark bits in separate units, such as in a block-based way.

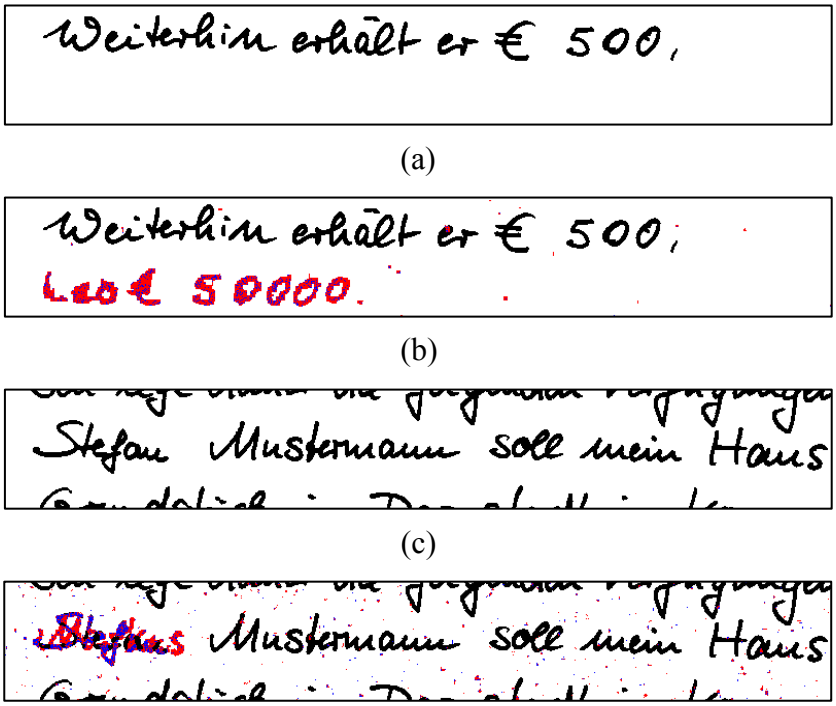


Figure 5-13 Authentication results for manipulations inside and outside the ROI area in the text image. (a) Manipulation inside the ROI, the text "Leo €5000" is deleted, (b) Authentication result of (a), (c) Manipulation outside the ROI, the name "Markus" is replaced by "Stefan", (d) Authentication result of (c). The authentication results indicate the deleted text in red color and the forged text in blue.

Chapter 6 Final Remarks

6.1 Conclusion

In this thesis, we have addressed the challenges of the digital watermarking techniques for image content authentication and successfully developed several novel watermarking solutions. In our solutions, we have concentrated on reducing the necessary watermark payload to improve the image quality and also enhancing the tamper detection capability of the authentication system.

For natural images, we have proposed a novel semi-fragile watermarking scheme for content authentication. The proposed scheme embeds the watermark in the wavelet domain. By introducing the random permutation strategy, the required watermark payload is reduced and the tamper localization accuracy is significantly increased. Because less image modifications are needed during the embedding, the image quality also gets improved. Furthermore, thanks to the embedding among the random distributed wavelet coefficients, the proposed algorithm is intrinsically secure to local attacks. The statistical tamper detection method enables scalable levels of detection sensitivity.

For synthetic images, we have discussed their unique characteristics and special requirements for the watermarking algorithm. Due to the simplicity of the content, the

common watermarking algorithms designed for natural images are not applicable to synthetic images. Based on our study, we have developed a novel watermarking scheme for synthetic image authentication. A new embedding strategy is introduced to replace the classical odd-even embedding method, which enables the capability of recovering the altered pixels. Based on a statistical analysis of the potential unverified pixels, the proposed scheme can achieve a pixel-wise tamper localization resolution. The proposed algorithm can be applied to different kinds of synthetic images, like binary documents, digital maps, line drawings, and so on.

In addition, we have also addressed the challenges that arise in some special applications, where the image fidelity is of essential importance, especially in the important regions where no modification is allowed. In this case, the common watermarking schemes can not fulfill the requirements. To solve this problem, we have proposed a non-ubiquitous watermarking framework for the image content authentication by introducing the concept of region of interest (ROI) masking. Under this framework, we have modified the proposed watermarking algorithms for natural and synthetic images to embed the watermark only by modifying the image content outside the specified regions of interest. Although no watermark is embedded in these ROI regions, their integrity is still ensured by the watermark information embedded in the other regions. The same tamper localization resolution is achieved both inside and outside the unwatermarked regions. Moreover, we have also discussed that the proposed framework can be applied to other watermarking schemes that embed the watermark bits in separate image units, such as in a block-based way.

6.2 Future Work

In both the watermarking algorithms for natural images and synthetic images, we have used an authentication code as the watermark that is generated under the control of the secret key. The authentication code is independent of the image content. As the next step, we plan to use some content-based features to substitute the authentication code. The properly selected features can enhance the sensitivity of the watermark to

intentional/malicious content manipulations and simultaneously reduce the false alarm rate caused by incidental distortions from the common image processing.

In the proposed watermarking scheme for synthetic image authentication, we have classified the existing colors to binary sets and therefore it can only identify two kinds of pixel categories in the tamper recovery process. As discussed in Section 4.7, we plan to extend the proposed embedding methods by introducing more dummy mapping entries. This extension will consequently require more kinds of flippable pixels to fulfill the pixel modifications in every single block. In future work, to enlarge the watermark capacity for synthetic images, we intend to use more available entries in the palette for the watermarking embedding of the color synthetic images. In addition, more complex pixel flipping rules can be defined to identify more flippable pixels. For example, the current rules are only based on the single pixel flipping. By taking into account flipping a group of pixels together, the number of flippable pixels can be further increased.

In this thesis, we have introduced the concept of non-ubiquitous watermarking by developing a watermarking framework with region of interest masking. Under this framework, high image fidelity in the user-defined regions is achieved. However, although the fidelity of the specified regions can be perfectly preserved, the authenticator still performs a holistic content verification. Any content tampering either inside or outside the ROI regions will render the whole image unauthentic. Unfortunately, in many practical applications, this is not the most desirable way of verifying the image content. In these applications, not all kinds of content-changing manipulations are deemed as malicious attacks. For instance, visual annotation, which adds additional visual content on the images like a logo or time stamps, is usually an acceptable manipulation in most cases. Moreover, the importance of different regions in an image is also different. The content of regions of interest, which is usually determined by the specific application, is of most interest and importance and therefore requires more or higher level of protection. To tackle these requirements, semantic

authentication mechanisms are needed to be developed based on the content analysis and image understanding techniques.

In multimedia authentication research, content-based authentication is still an open issue. To accomplish semantic image authentication, not only the syntactic features need to be utilized based on visual models, but also the underlying semantic content and features have to be defined and applied in the watermarking process. We have done some preliminary studies with regard to the semantic image authentication. We have developed a semantically extended watermarking model by illustrating the relationship of digital watermarking and content understanding. Both the semantic and syntactic image features are considered in the proposed watermarking model. This model provides a framework solution for the semantic watermarking. Based on the proposed model, we have then developed a semantic watermarking scheme that enables semantic image content authentication with multiple security levels. In this scheme, we consider the human faces as the particular regions of interest because they are increasingly important for security issues and massively present in different visual contents. Multiple watermarks are embedded in different image regions so that our approach is able to trace the type of the manipulation and identify the attacks among the face regions, such as face adding, moving, deletion, and so forth. This information can help us to infer the attacker's motives. More detailed introduction can be referred to [LSFS04][LSFS05a][LSFS05b]. In future work, we plan to improve the proposed semantic watermarking model by integrating the semi-fragile watermarks to provide more effective integrity protection against slight tampering inside face regions. Another open issue is to achieve moderate robustness against geometric distortions, such as scaling and rotation, because in some cases slight geometric transformations are also acceptable manipulations as long as they do not change the image meaning. In addition, we also intend to test the semantic watermarking model with other popular regions of interest, such as cars, people, trees and so forth.

References

[A03] M. Awrangjeb, “An Overview of Reversible Data Hiding”, in 6th International Conference on Computer and Information Technology (ICCIT 2003), pp. 75-79, Bangladesh, 19-21 Dec, 2003.

[AK03] M. Awrangjeb, M. S. Kankanhalli, “Lossless watermarking considering the human visual system”, Int. Workshop on Digital Watermarking 2003, Lecture Notes in Computer Science 2939, pp. 581-592, 2003.

[AM99] T. Amamo, D. Misaki, “Feature calibration method for watermarking of document images”, in Proc. 5th Int. Conf. Document Analysis and Recognition, pp. 91–94, Bangalore, India, 1999.

[AS72] M. Abramowitz, I. A. Stegun, eds., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover Publications, 1972.

[BG96] J. Brassil, L. O’Gorman, “Watermarking document images with bounding box expansion”, Proc. 1st Int. Workshop on Information Hiding, Newton Institute, Cambridge, UK, pp. 227-235, May 1996.

[C96] K. R. Castleman, *Digital Image Processing*. Prentice Hall, ISBN: 0132114674, 1996.

[C98] N. Chotikakamthorn, "Electronic document data hiding technique using inter-character space," in Proc. IEEE Asia-Pacific Conf. on Circuits and Systems, IEEE'APCCNS 1998, pp. 419–422, November 1998.

[C99] N. Chotikakamthorn, "Document image data hiding techniques using character spacing width sequence coding", Proc. IEEE Intl. Conf. Image Processing, Japan, 1999.

[CGM02] Y. Chauhan, P. Gupta, K. L. Majumder, "Digital Watermarking of Satellite Images", Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP2002, December, 2002.

[CKLS97] I. J. Cox, J. Kilian, T. Leighton, T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Trans. on Image Processing, vol. 6, no. 12, pp. 1673-1687, 1997.

[CMB01] I. J. Cox, M. L. Miller, J. A. Bloom, *Digital watermarking*. San Mateo, CA: Morgan Kaufmann, ISBN: 1-55860-714-5, 2001.

[CMTWY99] D. Coppersmith, F. Mintzer, C. Tresser, C. Wu, C. Yeung, "Fragile imperceptible digital watermark with privacy control", in Proceedings of SPIE, Security and Watermarking of Multimedia Contents, pp. 79-84, San Jose, USA, January, 1999.

[CS07] N. Cvejic , T. Seppänen, *Digital Audio Watermarking Techniques and Technologies: Application and Benchmarks*. Idea Group Inc (IGI), ISBN: 159904515X, 2007.

[CSST01] M. U. Celik, G. Sharma, E. Saber, A. M. Tekalp, "A hierarchical image authentication watermark with improved localization and security", Proceedings of IEEE International Conference on Image Processing (ICIP2001), vol. 2, pp. 502-506, October 2001.

[CSTS02] M. U. Celik, G. Sharma, A. M. Tekalp, E. Saber, "Reversible Data Hiding", IEEE Int. Conf. on Image Processing, vol. 2, pp. 157-160, 2002.

[CTL05] C. C. Chang, C. S. Tseng, C. C. Lin, “Hiding data in binary images”, ISPEC 2005, Lecture Notes in Computer Science 3439, pp. 338-349, 2005.

[CW01] B. Chen, G. W. Wornell, “Quantization Index Modulation: a Class of Provably good Methods for Digital Watermarking and Information Embedding”, IEEE Trans. Information Theory, vol. 47, no. 4, pp. 1423-1443, May 2001.

[CWC05] S. Cheng, Q. Wu, K. R. Castleman, “Non-ubiquitous Digital Watermarking for Record Indexing and Integrity Protection of Medical Images”, In Proc. of IEEE International Conference on Image Processing (ICIP2005), vol. 2, pp. 1602-1605, 2005.

[CWMA01] M. Chen, E. K.Wong, N. Memon, S. Adams, “Recent development in document image watermarking and data hiding”, Proc. SPIE, vol. 4518, pp. 166–176, Aug. 2001.

[D92] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia: SIAM, ISBN: 0898712742, 1992.

[EG01] J. J. Eggers, B. Girod, “Blind watermarking applied to image authentication”, International Conference on Acoustics, Speech and Signal Processing, ICASSP’2001, vol. 3, pp. 1977-1980, Salt Lake City, USA, May 2001.

[ESA04] Ö. Ekici, B. Sankur and M. Akçay, “Comparative evaluation of semifragile watermarking algorithms”, Journal of Electronic Imaging, 13(1), 209-216, January 2004.

[ESG00] J. J. Eggers, J. K. Su, B. Girod, “A blind watermarking scheme based on structured codebooks”, in Secure Image and Image Authentication, Proc. IEE Colloquium, pp. 4/1-4/6, London, UK, April 2000.

[F02] J. Fridrich, “Security of Fragile Authentication Watermarks with Localization”, Proc. of SPIE Security and Watermarking of Multimedia Contents IV, vol. 4675, pp. 691-700, San Jose, California, January, 2002.

[F98a] J. Fridrich, “Image watermarking for tamper detection”, Proc. of International Conference of Image Processing, ICIP’98, Chicago, Oct 1998.

[F98b] J. Fridrich, “Methods for detecting changes in digital images”, Proc. of the 6th IEEE International Workshop on Intelligent Signal Processing and Communication Systems, ISPACS’99, pp. 173-177, Melbourne, November, 1998.

[F99] J. Fridrich, “A hybrid watermark for tamper detection in digital images”, Proc. of the Fifth International Symposium on Signal Processing and Its Applications (ISSPA'99), vol. 1, pp. 301–304, Brisbane, Australia, August 22-25, 1999.

[FA00a] M. S. Fu, O. C. Au, “Data hiding for halftone images”, Proc of SPIE Conf. On Security and Watermarking of Multimedia Contents II, vol. 3971, pp. 228-236, Jan. 2000.

[FA00b] M. S. Fu, O. C. Au, “Data hiding by smart pair toggling for halftone images”, Proc. of IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol. 4, pp. 2318-2321, June 2000.

[FA01] M. S. Fu, O. C. Au, “Improved halftone image data hiding with intensity selection”, Proc. IEEE International Symposium on Circuits and Systems, vol. 5, pp. 243-246, 2001.

[FGB00] J. Fridrich, M. Goljan, A.C. Baldoza, “New Fragile Authentication Watermark for Images”, ICIP'2000, Vancouver, Canada, September, 2000.

[FGD01] J. Fridrich, M. Goljan, R. Du., “Invertible Authentication”, In Proc. SPIE, Security and Watermarking of Multimedia Contents III, vol. 3971, pp. 197-208, San Jose, USA, 2001.

[FGM00] J. Fridrich, M. Goljan, N. Memon, “Further Attacks on Yeung-Mintzer Fragile Watermarking Scheme,” Electronic Imaging 2000, Security and Watermarking of Multimedia Contents II, Proc. of SPIE, vol. 3971, pp.428–437, San Jose, California, January, 2000.

[FGM02] J. Fridrich, M. Goljan, N. Memon, "Cryptanalysis of the Yeung-Mintzer Fragile Watermarking Technique", Journal of Electronic Imaging, vol. 11, no. 2, pp. 262-274, May, 2002.

[G04] O. Goldreich, *Foundations of Cryptography – Volume 2*: *Basic Applications*. Cambridge University Press, ISBN: 0-521-83084-2, 2004.

[GPK06] A. Giakoumaki, S. Pavlopoulos, D.Koutsouris, "Multiple Image Watermarking Applied to Health Information Management", IEEE Transactions on Information Technology in Biomedicine, vol. 10, Issue 4, pp. 722 – 732, Oct. 2006.

[HM00] M. Holliman, N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," IEEE Trans. Image Process., vol. 9, no. 3, pp. 432–441, 2000.

[HY01] D. Huang, H. Yan, "Interword distance changes represented by sine waves for watermarking text images", IEEE Trans. Circuits Syst. Video Technology, vol. 11, no. 12, pp. 1237–1245, December 2001.

[J00] JPEG 2000 part 2 Final Committee Draft, ISO/IEC JTC1/SC20 WG1 N2000, December 2000.

[K05] H.Y. Kim, "A new public-key authentication watermarking for binary document images resistant to parity attacks", in Proc. IEEE Int. Conf. on Image Processing (ICIP2005), vol. 2, pp. 1074-1077, 2005.

[KA03] H. Y. Kim, A. Afif, "Secure authentication watermarking for binary images", Proc. of XVI Brazilian Symposium on Computer Graphics and Image Processing, pp. 199-206, Oct. 2003.

[KA04] H. Y. Kim, A. Afif, "A Secure Authentication Watermarking for Halftone and Binary Images," Int. J. Imaging Systems and Technology, vol. 14, no. 4, pp. 147-152, 2004.

[KH99] D. Kundur, D. Hatzinakos, “Digital watermarking for telltale tamper proofing and authentication”, Proceedings of IEEE, vol. 87, no. 7, pp. 1167-1180, July 1999.

[KP00] S. Katzenbeisser, F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, ISBN: 1-58053-035-4, 2000.

[KQ04] H. Y. Kim, R. L. de Queiroz, “Alteration-Locating authentication watermarking for binary images”, Int. Workshop on Digital Watermarking 2004, Lecture Notes in Computer Science 3304, pp. 125-136, 2004.

[LC00] C. Y. Lin, S. F. Chang, “Semi-fragile watermarking for authentication JPEG visual content”, Proc. of SPIE, Security and Watermarking of Multimedia Contents II, vol. 3971, pp. 140-151, 2000.

[LC01] C. Y. Lin, S. F. Chang, “A robust image authentication method distinguishing JPEG compression from malicious manipulation”, IEEE Trans. Circuits and System for Video Technology, vol. 11, no. 2, pp. 153-168, 2001.

[LC97] C. Y. Lin, S. F. Chang, “A robust image authentication algorithm surviving JPEG compression”, Proc. of SPIE, Storage and Retrieval of Image/Video Database, EI’98, vol. 3312, pp. 296-307, San Jose, Jan 1998.

[LHLH03] W. Lie, T. Hsu, G. Lin, W. Ho, “Fragile watermarking for JPEG-2000 images”, 16th IPPR Conf. on Computer Vision, Graphics and Image Processing (CVGIP 2003), pp. 823 – 826, August, 2003.

[LKC03] H. Lu, A. C. Kot, and J. Cheng, “Secure data hiding in binary images for authentication”, in Proc. 2003 Int. Symp. Circuits and Systems, ISCAS’03, vol. 3, pp. 806-809, May 25–28, 2003.

[LM98] S. H. Low, and N. F. Maxemchuk, “Performance comparison of two text marking methods”, IEEE Journal on Selected Areas in Communications, vol. 16, no. 4, May 1998.

[LMB95] S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O'Gorman, "Document marking and identification using both line and word shifting", INFOCOM 95, IEEE Computer Society Press, Los Alamitos, California, 1995.

[LML98] S. H. Low, N. F. Maxemchuk, A. M. Lapone, "Document identification for copyright protection using centroid detection", IEEE Trans. on Comm., vol. 46, no. 3, pp. 372-383, Mar 1998.

[LPD00] E. T. Lin, C. I. Podilchuk, E. J. Delp, "Detection of image alterations using semi-fragile watermarks," Proc. of SPIE, Security and Watermarking of Multimedia Contents II, vol. 3971, pp. 152–163, January 2000.

[LSFS04] H. Liu, H. Sahbi, L. C. Ferri, M. Steinebach, "Image authentication using automatic detected ROIs", 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2004, ISBN 972-98115-7-1, Lisbon, Portugal, April 21-23, 2004.

[LSFS05a] H. Liu, H. Sahbi, L. C. Ferri, M. Steinebach, "Advanced semantic authentication of face images", 6th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS05, Montreux, Switzerland, April 13-15, 2005.

[LSFS05b] H. Liu, H. Sahbi, L. C. Ferri, M. Steinebach, "Semantically extended digital watermarking model for multimedia content", 9th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS2005), Salzburg, Austria, September 19-21, 2005.

[LWKS02] H. Lu, J. Wang, A. C. Kot, Y. Q. Shi, "An objective distortion measure for binary document images based on human visual perception," in Proc. Int. Conf. Pattern Recognition, vol. 4, pp. 239–242, Quebec, Canada, Aug. 2002.

[ML97] N. F. Maxemchuk, S. H. Low, "Marking text documents," Proc. IEEE Int. Conf. on Image Processing (ICIP'97), October 1997.

[MOV96] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, ISBN: 0-8493-8523-7, 1996.

[MSCS06] K. Maeno, Q. Sun, S. Chang, M. Suto, "New semi-fragile image authentication watermarking techniques using random bias and non-uniform quantization", IEEE Transactions on Multimedia, vol. 8, no. 1, pp. 32-45, February 2006.

[MWM01] Q. Mei, E. K. Wong, N. Memon, "Data hiding in binary text document", Proc. SPIE, vol. 4314, pp. 369-375, 2001.

[NSAS06] Z. Ni, Y. Shi, N. Ansari, W. Su, "Reversible data hiding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, No. 3, pp. 354-362, March 2006.

[OE05] A. H. Ouda, M. R. El-Sakka, "Localization and security enhancement of block-based image authentication," Proc. of IEEE International Conference on Image Processing (ICIP 2005) vol.1, pp.673-676, Sept. 2005.

[PCT00] H. K. Pan, Y. Y. Chen, Yu-Chee Tseng, "A Secure Data Hiding Scheme for Two-Color Images," in Proc. IEEE 5th Symp. on Computers and Communications, pp. 750-755, July 2000.

[PM05] B. Planitz, A. Maeder, "Medical Image Watermarking: A Study on Image Degradation", in Proc. Australian Pattern Recognition Society Workshop on Digital Image Computing, WDIC 2005, Brisbane, Australia, 2005.

[PWP02] G. Pan, Z. Wu, Y. Pan, "A data hiding method for few-color images," in Proc. IEEE ICASSP 2002, vol. 4, pp. 3469–3472, May 13–17, 2002.

[PZ98] C. I. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models", IEEE Journal on Selected Areas in Communications, vol. 16(4), pp.525--539, May 1998.

[S95] D. R. Stinson, *Cryptography*: *Theory and Practice*. Boca Raton, FL: CRC Press, 1995.

[SL04] H. Si, C.-T. Li, "Fragile watermarking scheme based on the block-wise dependency in the wavelet domain," in Proc. ACM Multimedia and Security Workshop, pp. 214-219, Magdeburg, Germany, September 2004.

[SMW99] N. Memon, S. Shende, P. Wong, "On the security of the Yeung-Mintzer authentication watermark", Proceedings of the Conference on Image Processing, Image Quality and Image Capture Systems (PICS-99), pp. 301-306, Savanna, Georgia, April 1999.

[SO05] "Fototricks im Geschäftsbericht: Deutsche Bank montiert ihren Vorstand neu", Spiegel Online, available at http://www.spiegel.de/wirtschaft/0,1518,349762,00.html, April 5, 2005.

[TCP02] Y. C. Tseng, Y. Y. Chen, H. K. Pan, "A Secure Data Hiding Scheme for Binary Images," IEEE Transactions on Communications, vol. 50, no. 8, pp.1227-1231, August 2002.

[TP01] Y. C. Tseng, H. K. Pan, "Secure and invisible data hiding in 2-color images", IEEE INFOCOM 2001, Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 887-896, 2001.

[W01] M. Wu, "Multimedia Data Hiding", Ph.D. dissertation, Princeton University, available at http://www.ece.umd.edu/~minwu/research/phd_thesis.html, Princeton, NJ, USA, June 2001.

[W02] A. Wakatani: "Digital Watermarking for ROI Medical Images by Using Compressed Signature Image", 35th Hawaii International Conference on System Sciences (HICSS-35), Island of Hawaii, January 7-10, 2002.

[W03] M. Wu, "Joint Security and Robustness Enhancement for Quantization Based Data Embedding", IEEE Trans. On Circuits and Systems for Video Technology, vol. 13, no. 8, August 2003.

[W98] P. W. Wong, “A Public Key Watermark for Image Verification and Authentication”, Proceedings of IEEE International Conference on Image Processing, Chicago, USA, pp. 425-429, October, 1998.

[WKBC02] D. A. Winne, H. D. Knowles, D. R. Bull, C. N. Canagarajah, “Digital Watermarking in Wavelet Domain with Predistortion for Authenticity Verification and Localization”, Proc. of SPIE Security and Watermarking of Multimedia Contents IV, vol. 4675, San Jose, California, January, 2002.

[WL04] M. Wu, B. L, “Data Hiding in Binary Image for Authentication and Annotation”, IEEE Trans. Multimedia, vol. 6, no. 4, pp. 528-538, August 2004.

[WL98] M. Wu, B. Liu, “Watermarking for Image Authentication”, in Proc. IEEE Int. Conf. on Image Processiong (ICIP’98), Chicao, IL, 1998.

[WL99] M. Wu and B. Liu, “Digital watermarking using shuffling”, Proc. of IEEE International Conference on Image Processing (ICIP'99), Kobe, Japan, vol.1, pp.291-295, Oct. 1999.

[WM00] P. W. Wong, N. Memon, “Secret and public key authentication watermarking schemes that resist vector quantization attack”, Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents II,vol. 3971, pp. 417-427, San Jose, USA, 2000.

[WTL00] M. Wu, E. Tang, B. Liu, “Data Hiding in Digital Binary Image”, in Proc. IEEE Int. Conf. Multimedia and Expo., pp. 393-396, 2000.

[WZLL04] J. Wu, B. B. Zhu, S. Li, F. Lin, “Efficient oracle attacks on Yeung-Mintzer and variant authentication schemes”, IEEE International Conference on Multimedia and Expo (ICME '04), vol.2, pp. 931-934, 27-30 June 2004.

[YK04] H. Yang, A. C. Kot, “Data hiding for bi-level documents using smoothing techniques,” in Proc. IEEE Int. Symp. Circuits Systems (ISCAS’04), vol. 5, pp. 692–695, May 2004.

[YK07] H. Yang, A. C. Kot, “Pattern-Based Data Hiding for Binary Image Authentication by Connectivity-Preserving”, IEEE Trans. On Multimedia, vol. 9, no. 3, April 2007.

[YLL01] G.Yu, C. Lu, H. M. Liao, “Mean quantization-based fragile watermarking for image authentication”, Optical Engineering, vol.40, no.7, pp.1396-1408, July 2001.

[YM97] M. M. Yeung, F. Mintzer, “An Invisible Watermarking Technique for Image Verification”, in Proc. IEEE Int. Conf. on Image Processing (ICIP’97), vol. 2, pp. 680-683, Santa Barbara, 1997.

[ZS03] B. B. Zhu, M. D. Swanson, “Multimedia authentication and watermarking,” Multimedia Information Retrieval and Management, D. Feng, W. C. Siu, and H. J. Zhang, Eds. Springer-Verlag, ISBN 3540002448, Ch. 7, pp. 148–177, 2003.

[ZST04] B. B. Zhu, M. D. Swanson, A. H. Tewfik, “When Seeing Isn't Believing,” IEEE Signal Processing Magazine, vol. 21, no. 2, pp. 40-49, March 2004.

[ZW07] X. Zhang, S. Wang, “Statistical fragile watermarking capable of locating individual tampered pixels”, IEEE Signal Processing Letters, vol. 14, no. 10, pp. 727-730, October 2007.

Curriculum Vitae

Huajian Liu

Education

2002-2008	Ph.D. candidate in Computer Science, Department of Computer Science (Fachbereich Informatik), Darmstadt University of Technology (Technische Universität Darmstadt), Darmstadt, Germany
1999-2002	Master of Engineering in Signal and Information Processing, Department of Electronic Engineering, Dalian University of Technology, Dalian, China
1995-1999	Bachelor of Engineering in Electronic Engineering, Department of Electronic Engineering, Dalian University of Technology, Dalian, China
1992-1995	Weifang No. 1 Senior High School, Weifang, China
1989-1992	Weifang No. 2 Junior High School, Weifang, China

Professional Experience

2007-present	Fraunhofer Institute SIT, Research Associate, Division of Transaction and Document Security (TAD), Darmstadt, Germany
2002-2006	Fraunhofer Institute IPSI, Research Associate, Division of Media Security in IT (MERIT), Darmstadt, Germany