Yun Qing Shi
Byeungwoo Jeon (Eds.)

LNCS 4283

Digital Watermarking

5th International Workshop, IWDW 2006
Jeju Island, Korea, November 2006
Proceedings

Springer

Lecture Notes in Computer Science 4283

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Yun Qing Shi Byeungwoo Jeon (Eds.)

Digital Watermarking

5th International Workshop, IWDW 2006
Jeju Island, Korea, November 8-10, 2006
Proceedings

Springer

Volume Editors

Yun Qing Shi
New Jersey Institute of Technology
Newark, New Jersey, USA
E-mail: shi@njit.edu

Byeungwoo Jeon
Sung Kyun Kwan University
300 Chunchun-dong
Jangan-gu, Suwon, Korea
E-mail: bjeon@yurim.skku.ac.kr

Library of Congress Control Number: 2006935426

CR Subject Classification (1998): K.4.1, K.6.5, H.5.1, D.4.6, E.3, E.4, F.2.2, H.3, I.4

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-48825-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-48825-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11922841 06/3142 5 4 3 2 1 0

Preface

Welcome to the proceedings of the Fifth International Workshop on Digital Watermarking (IWDW). Since the first IWDW held in Seoul, Korea in 2002, it has been a focal point for meeting in person and disseminating valuable scientific and technological developments in watermarking. IWDW 2006 was held on Jeju, the dream island in Korea. The main theme of the workshop was "*Meet the Challenges in this Digital World!*" As we all know, digital watermarking and its related technologies have emerged as the key ingredients of this digital world. We report on new developments and discuss how to best utilize the watermarking and its related new technologies to cope with many challenging issues in this digital world.

This year, we accepted 34 papers out of 76 highly qualified submissions from 14 different countries. Each paper was reviewed by three reviewers. The acceptance ratio of 44% indicates IWDW's continuing commitment to ensuring the quality of the workshop. In addition, we had three invited lectures and one panel discussion that shed invaluable insights to the watermarking community on new developments and future directions. The technical program featured such topics as steganography and steganalysis, data forensics, digital right management, secure watermarking, and their applications. The 34 accepted papers, three invited lectures, and the panel discussion covered both theoretical and practical issues that all of us can benefit from. Furthermore, 13 of the 34 papers were arranged in a poster session in order to facilitate more efficient and interactive information exchange.

Our deep appreciation goes to all of the authors who submitted papers to IWDW 2006, the invited lecturers, the panelists and the participants, who all contributed to IWDW 2006. We are grateful to the members of the Technical Program Committee and all the invited reviewers, since IWDW 2006 would not have been successful without their efforts and time — they finished their high-quality evaluation of the submitted papers in a professional and timely fashion. In addition, we are grateful to the kind sponsors of IWDW 2006, including Digital Times, Electronics and Telecommunications Research Institute (ETRI), Korea Advanced Institute of Science and Technology (KAIST), Korea University, and Sung Kyun Kwan University in Korea. Our appreciation goes to the General Chair, JooSeok Song, President of Korea Institute of Information Security and Cryptology, for his leadership, and to the Organizing Committee led by Jeho Nam for its excellent job in financing, publicity, publication, and registration. Last but not the least, our thanks go to Victoria Kim for her professional perfectionism in managing and assisting us as the Conference Secretary.

September 2006

Yun-Qing Shi
Byeungwoo Jeon

Organization

Committee List

Technical Program Committee

Mauro Barni (University of Siena, Italy)

Jeffrey Bloom (Thomson Corporate Research, USA)

Jana Dittmann (Otto-von-Guericke-University of Magdeburg, Germany)

Jean-Luc Dugelay (Institut EURECOM, France)

Teddy Furon (INRIA, France)

Miroslav Goljan (State University of New York, USA)

Jiwu Huang (Sun Yat-Sen University, China)

Mohan Kankanhalli (National University of Singapore, Singapore)

Stefan Katzenbeisser (Philips Research, Netherlands)

Hyoung-Joong Kim (Korea University, Korea)

C.-C. Jay Kuo (University of Southern California, USA)

Inald Lagendijk (Delft University of Technology, Netherlands)

Heung-Kyu Lee (Korea Advanced Institute of Science and Technology, Korea)

Zheming Lu (Harbin Institute of Technology, China)

Benoit Macq (Université Catholique de Louvain, Belgium)

Nasir Memon (Polytechnic University, USA)

M. Kivanc Mihcak (Bogazici University, Turkey)

Matt Miller (NEC, USA)

Hideki Noda (Kyushu Institute of Technology, Japan)

Jeng-Shyang Pan (National Kaohsiung University of Applied Sciences, Taiwan)

Fernando Perez-Gonzalez (University of Vigo, Spain)

Raphael C.-W. Phan (Swinburne University of Technology, Malaysia)

Ioannis Pitas (University of Thessaloniki, Greece)

Alessandro Piva (University of Florence, Italy)

Yong-Man Ro (Information and Communication University, Korea)

Ahmad-Reza Sadeghi (Ruhr University Bochum, Germany)

Kouichi Sakurai (Kyushu University, Japan)

Qibin Sun (Institute for Infocomm Research, Singapore)

Sviatoslav Voloshynovskiy (CUI-University of Geneva, Switzerland)

Chee Sun Won (Dongguk University, Korea)

Min Wu (University of Maryland, USA)

Additional Reviewer List

Roberto Caldelli (Swinburne University of Technology, Malaysia)

Chunhua Chen (New Jersey Institute of Technology, USA)

Wen Chen (New Jersey Institute of Technology, USA)

Dongdong Fu (New Jersey Institute of Technology, USA)

Hongmei Gou (University of Maryland, USA)

Anthony TS Ho (University of Surrey, UK)

Keiichi Iwamura (Science Univ. of Tokyo, Japan)

Xiangui Kang (Sun Yat-Sen University, China)

Oleksiy Koval (CUI-University of Geneva, Switzerland)

Minoru Kuribayashi (Swinburne University of Technology, Malaysia)

Hongmei Liu (Sun Yat-Sen University, China)

Yinian Mao (University of Maryland, USA)

Zhicheng Ni (World Gate Communications, USA)

Maria Paula Queluz (Swinburne University of Technology, Malaysia)

Shunquan Tan (Sun Yat-Sen University, China)

Francesca Uccheddu (Swinburne University of Technology, Malaysia)

Yoshifumi Ueshige (Institute of Systems and Information Technology/KYUSHU, Japan)

Avinash Varna (University of Maryland, USA)

Guorong Xuan (Tongji University, China)

Dekun Zou (Thomson Corporate Research, USA)

Table of Contents

Watermarking Is Not Cryptography

Ingemar J. Cox[1], Gwenaël Doërr[1], and Teddy Furon[2]

[1] University College London
Adastral Park, Ross Building 2
Martlesham IP5 3RE, United Kingdom
{i.cox, g.doerr}@adastral.ucl.ac.uk
http://www.adastral.ucl.ac.uk
[2] INRIA / TEMICS
Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
teddy.furon@irisa.fr
http://www.irisa.fr

Abstract. A number of analogies to cryptographic concepts have been made about watermarking. In this paper, we argue that these analogies are misleading or incorrect, and highlight several analogies to support our argument. We believe that the fundamental role of watermarking is the reliable embedding and detection of information and should therefore be considered a form of communications. We note that the fields of communications and cryptography are quite distinct and while communications *systems* often combine technologies from the two fields, a layered architecture is applied that requires no knowledge of the layers above. We discuss how this layered approach can be applied to watermarking applications.

1 Introduction

Digital watermarking has received considerable attention as a complement to cryptography for the protection of digital content such as music, video and images. Cryptography provides a means for secure delivery of content to the consumer. Legitimate consumers are explicitly or implicitly provided with a key to decrypt the content in order to view or listen to it. Unfortunately, not all legitimate consumers are trustworthy and an untrustworthy consumer may alter or copy the decrypted content in a manner that is not permitted by the content owner. However, cryptography provides no protection once the content is decrypted, which is required for human perception. Watermarking complements cryptography by embedding a message within the content. If properly designed, the message remains in the content after decryption and, more importantly, after digital-to-analog and analog-to-digital conversion. By so doing, watermarking can be used to close the 'analog hole'[1].

[1] Not only must the digital content be decrypted, but it must also be converted to an analog signal in order for a person to see or hear it. This gives rise to the 'analog hole', which refers to the fact that all digital protection is lost at the point of perception. And this analog signal may be re-digitized by an untrustworthy consumer in order to obtain an unprotected digital copy of the content.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 1–15, 2006.

Since the primary motivation for watermarking has been for security, numerous analogies have been made between watermarking and cryptography. In this paper, we argue that many of these analogies are for the moment misleading or incorrect. We argue that watermarking should only be viewed as a means for reliably embedding and decoding information hidden in a cover Work. As such, it is a communication system, often modeled as spread spectrum communications or communications with side information. A system incorporating watermarking may also use cryptography but we argue that, up to now, a layered model has been much more successful than intermingling the two concepts.

To support our argument, we first provide a brief introduction to key concepts in communications (Section 2) and cryptography (Section 3). We then discuss the security requirements associated with watermarking. Section 4 highlights a number of cryptographic analogies used within the watermarking community and discusses the weaknesses of these analogies. A contrario, Section 5 shows that the layered model offers much safer designs with the examples of watermarking-based content authentication and watermarking-based traitor tracing. The last section extends this discussion to signal processing other than watermarking.

2 Communications

Communications is concerned with *reliable* transmission of a message from Alice to Bob over an *unreliable* channel. A channel is considered unreliable if there is a finite probability that an error will occur between the points of transmission and reception, e.g. Alice sends a '0'-bit, but Bob decodes a '1'-bit. Reliable communications is concerned with bandwidth, power or signal-to-noise ratio (SNR), channel coding and bit error rate (BER).

It was, of course, Shannon [1] who showed that the maximum rate of error free transmission, i.e. the channel capacity (in bits per second), is given by:

$$C = 2B \log_2 \left(1 + \frac{s}{n}\right) \tag{1}$$

where B denotes bandwidth in hertz, and s and n the signal and white Gaussian noise powers respectively. In order to approach this limit, it is necessary to encode the message m, prior to transmission. This channel code provides a level of redundancy that is measured by the code rate, R. For example, if every k-bits of the message are represented by an n-bit code, then the rate is $R = k/n$, where $n > k$. Finally the BER is a direct measure of the error rate achieved by a particular code and is usually plotted as a function of the SNR.

The sources of bit errors are many. The most common error model is Gaussian noise, but there are many other error sources. However, all such sources are usually considered to be naturally occurring and not due to the effects of an adversary. In fact, it is very rare for a civilian communications system to consider a hostile channel. However, military communications must do so. In a hostile military environment, the two primary concerns are (i) jamming and (ii) detection. Jamming refers to attempts by an active adversary to prevent Bob receiving a signal. Detection refers to an adversary's efforts to detect (and localize)

enemy communications. If this is successfully achieved then military firepower may be used to destroy the communications. Note that at this level, the concern is with the delivery of bits, not with the security of the bits (which is discussed in the next section). Secure communication is irrelevant if Bob never receives the communications!

Spread spectrum (SS) communications was originally developed to protect military communications from detection and jamming [2], although it is now widely used in many civilian applications, e.g. mobile phones. The basic principle behind SS communications is that each message bit is multiplied by a (pseudo random) chip sequence that spreads the message bit over a much broader spectrum. For example, consider an implementation of SS communications based on frequency hopping. Here, the original message bit is transmitted as n lower power bits (the chip sequence), each of which is transmitted over a separate frequency band that is pseudo-randomly chosen. The receiver is synchronized with the transmitter and also has knowledge of the pseudo-random sequence of frequency bands being used. Thus, Bob is able to sum the lower energy in each of the individual bands to produce a good signal-to-noise ratio (SNR) at the receiver.

However, an adversary has much greater difficulty detecting the transmission, since Eve does not know the pseudo-random frequency hopping sequence. If Eve monitors just one frequency band, she cannot be confident that there is any communication, since the signal transmitted is very weak and only persists for a short time. Furthermore, Eve cannot jam the channel as a precaution against possible communications. This is because the power needed to confidently jam all the frequency channels would be impractically large.

Another communications model that has received recent interest is known as communications with side information. Here the channel has two noise sources, both of which are unknown to the receiver, but the first of which is entirely known to the transmitter. Under these circumstances, which arise in mobile telephony and digital watermarking, how much information can Alice reliably transmit to Bob? Costa [3] proved that the channel capacity is the same as if the first noise source is absent.

3 Cryptography

Cryptography is concerned with the *secure* transmission of a message from a sender, Alice, to a recipient, Bob, over an insecure channel. A channel is considered insecure if the bits sent by Alice may be read or altered by an adversary, Eve, prior to receipt by Bob. It is important to realize that an insecure channel is not an unreliable channel. In fact, cryptography often assumes reliable communications, i.e. Bob receives exactly the same bits sent by either Alice or Eve - there are no unintentional errors.

A secure transmission is concerned with (i) privacy, (ii) integrity and (iii) authentication. Privacy is concerned with ensuring that an adversary, Eve, can learn nothing about the message intended for Bob, by examining the encrypted

bits sent by Alice. Integrity is concerned with ensuring that Bob can be confident that the message has not been altered by Eve prior to receipt. And authentication is concerned with guaranteeing that the sender of the message is actually Alice and not an impostor.

To ensure privacy, cryptography assumes the existence of an encryption function , E(.), which takes a message, m, and a key, K, and outputs an encrypted message, c, i.e. $c = \mathrm{E}(m, K)$. It further assumes a decryption function, D(.) that takes an encrypted message, c and a key, K, and outputs a cleartext message, m, i.e. $m = \mathrm{D}(c, K) = \mathrm{D}(\mathrm{E}(m, K), K)$.

Shannon [4] defined perfect security as an encryption function in which an adversary, Eve, learns nothing about the message, m, by inspection of the ciphertext, c. Perfect security can be realized using a one-time pad. Unfortunately, a one-time pad is not practical in most situations. Consequently, modern cryptography is therefore concerned with the design of cryptographic algorithms which approximate perfect security while re-using a shared key, K. It is assumed that the encryption and decryption algorithms are known to all parties, including the adversary, Eve. This is known as Kerckhoffs' Principle [5] and reduces Eve's cryptanalysis problem to inferring the key, K.

If the length of the binary key is n-bits, the total number of keys is 2^n and is called the keyspace. For sufficiently large n, the keyspace is enormous and exhaustive enumeration or brute force search is infeasible. Note that cryptography assumes that Eve learns nothing about the true key, K, by trying a key, K', that is close to K in the sense of say Hamming distance. In other words, if Alice encodes a message twice, once using key, K and once using a key, K', that differs by only one bit from K, then the two encrypted ciphertexts will be completely different with no correlation between them. In reality, modern cryptographic algorithms only approximate these assumptions.

Cryptographic systems in which the encryption and decryption algorithm share the same key are known as symmetric key or private key systems. One problem with such is how to initiate the system, i.e. how do Alice and Bob agree on a key without sharing this knowledge with Eve? Public key or asymmetric key cryptography solves this problem by assigning two keys to each individual: a public one (PK) that is published on a database and a secret one (SK) which is never disclosed. Everybody knows the public key of everybody. The main feature of public key watermarking lies in the asymmetry of the keys used during encryption and decryption, namely $m = \mathrm{D}(\mathrm{E}(m, PK), SK)$. For instance, Alice can encrypt the message she wishes to transmit with Bob's public key (PK_B). The resulting ciphertext $c = \mathrm{E}(m, PK_B)$ can then only be decrypted with Bob's secret key (SK_B) i.e. by Bob himself. In other words, the message m has been sent securely without agreeing on a secret key beforehand[2].

Integrity is guaranteed through the use of another cryptographic primitive known as a one-way hash function. This is a function that takes an arbitrarily

[2] However, for practical reasons, public key cryptography is usually used to exchange a key at the beginning of a transmission. The subsequent messages are then encrypted/decrypted with a private key crypto-system using the agreed session key.

long bit sequence (the pre-image) and outputs a constant length bit sequence known as a hash or digest. The characteristics of a hash function are:

1. it is easy to compute the hash value given a pre-image,
2. it is computationally unfeasible to generate a pre-image that hashes to a particular value,
3. it is hard to generate two pre-images with the same hash value, and
4. a single bit change in the pre-image results in a major change of the hash value.

The properties of a hash make it well-suited for guaranteeing the integrity of a message and the authenticity of its sender. For instance, Alice computes a hash of her message concatenated with the shared secret key, K, and appends the hash to the end of the message. This is one way to make what cryptographers call a message authentication code (MAC). Note that the message need not be encrypted if privacy is not an issue. On receipt, Bob can take the received message and compute the hash of the received message concatenated with their shared secret key. If this recomputed hash is identical to the hash appended by Alice, then Bob can be confident that the message has not been tampered with. While Eve may alter the unencrypted message, she is unable to compute the associated hash since Eve does not know the key shared by Alice and Bob. Consequently, any alteration made by Eve will be detected by Bob. Digital signatures combine hashing and public key encryption to guarantee a better authenticity and non repudiation while easing the key management.

4 Digital Watermarking

The most basic requirement of a digital watermarking system is the ability to embed and decode a message hidden within a cover Work. Applications of digital watermarking may require very much more. However, all systems need a reliable mechanism for embedding and decoding message bits[3]. We therefore believe that digital watermarking is fundamentally a form of communications and, as such, is primarily concerned with the reliable transmission of a message over an unreliable channel.

Of course, applications of digital watermarking also have security concerns. Security threats depend on the watermark application. However, the categories of attacks that have been identified are:

1. Unauthorized embedding,
2. Unauthorized decoding, and
3. Unauthorized removal.

Since much of the motivation for watermarking is driven by security concerns, it is not surprising that analogies have been made between watermarking and

[3] Even fragile watermarks must provide a reliable communications channel in the absence of distortions.

cryptography. However, while there are superficial similarities, we believe that most of these analogies are flawed. As an example, let us consider keyspaces in watermarking and cryptography.

4.1 The Keyspace Analogy

Some articles have studied the security of watermarking schemes with information theoretical tools [6,7,8], and especially equivocation. Here we rephrase this analysis in a simpler manner, thanks to a keyspace analogy. This analogy assumes that a watermarking technique is reliable because it is keyed by a n-sample sequence just like a crypto-system keyed by a n-bit secret. This analysis is not generic, we only consider the example of SS.

The security of a crypto-system is usually assessed by a common feature: the keyspace. The key θ randomly chosen from a keyspace Θ is usually a binary string made of n bits. An adversary without any *a priori* knowledge of the secret key can simply exhaustively test all the elements of the keyspace. This strategy is referred to as a brute force attack. The size of the keyspace is $|\Theta| = 2^n$. For each tested element, the probability P that it is equal to secret key is $P = 2^{-n}$ or $\log_2(P) = -n$. In other words, the larger the number of bits in the key, the lower is this probability P and the more time will be required to disclose the secret θ. For large n, say $n = 256$, the probability is negligibly small.

The concept of a key has been adopted by the watermarking community. For example, in spread spectrum watermarking, the key is used as a seed to a pseudo-random number generator that creates a binary antipodal sequence used as a carrier or chip sequence. Alternatively, the key may directly refer to the chip sequence. Nevertheless the behaviour of these two keyspaces is very different!

First, the 2^n possible antipodal binary sequences are not all eligible to serve as spread spectrum chip sequences. For instance, zero-average chip sequences are preferred to avoid affecting the direct component (DC) of the host signal, e.g. the average brightness of an image. This constraint reduces the number of possible keys to (in terms of bits):

$$\log_2 |\Theta| = \log_2 \binom{n}{n/2} \simeq n - \frac{1}{2}\log_2(n). \tag{2}$$

The approximation in (2) shows that despite this constraint, the size of the set is almost exponential and thus not drastically reduced.

However, there is a second, more serious difference with cryptography: the secret carrier does not need to be exactly disclosed in order to break the watermarking system. An attacker simply needs a close enough estimate! The more correlated the attacker's estimate is to the true chip sequence (i.e. secret key), the less distortion is required to remove the watermark. Indeed, practical studies have shown that a normalised correlation greater or equal to $\rho_{\min} = 0.4$ between the attacker's estimate and the true key, is sufficient to remove a watermark while maintaining good perceptual quality [6].

In other words, if at least $k_{\min} = \lceil n(\rho_{\min} + 1)/2 \rceil$ samples of the estimated carrier match the ones of the secret carrier, the attack will be successful. Keeping

in mind that half these 'matching samples' need to be 1's to preserve the zero average, the probability P that a randomly picked eligible carrier leads to a successful attack is given by:

$$P = \sum_{\substack{k_{\min} \le k \le n \\ k \text{ even}}} \binom{n/2}{k/2}^2 \Bigg/ \binom{n}{n/2}. \tag{3}$$

Numerical computations show that $\log_2(P) \approx -0.12\,n$ bits for $\rho_{\min} = 0.4$.

This is a remarkable difference! In simple terms, the cryptanalyst looks for the one and only unique secret key among 2^n eligible elements, whereas the watermark hacker looks for one of the $2^{0.88n}/\sqrt{n}$ suitable carriers among a set of $2^n/\sqrt{n}$, i.e. the search space is only $2^{0.12n}$.

Furthermore, we note that it has been proven that information about the secret key leaks from watermarked content (at least for spread spectrum [6] and lattice quantization index modulation [9] watermarking schemes). Thus, observations of watermarked content give the pirate strong *a priori* knowledge with which to estimate the key.

The keyspace analogy shows that the belief (a n-sample watermarking carrier provides as much security as a n-bit cryptographic key) is clearly flawed and highly misleading.

4.2 The Public Key Analogy

As discussed in Section 3, public key cryptography provides a mechanism for Alice and Bob to initiate a secure communication without first having to share a secret key. In watermarking, we would like to permit an untrustworthy third party, Eve, to read a watermark embedded in a Work. However, if we grant this capability, we do not want Eve to be able to remove the watermark from the content. The capability to read-but-not-remove, is almost a holy grail of digital watermarking.

Unfortunately, as of the time of writing, there is neither a theoretical proof on the feasibility of this capability, nor a practical watermark algorithm, that we are aware of.

Indeed, a number of papers have been published [10,11,12] that seek to create read-but-not-remove watermarking systems based on an analogy to public key cryptography. The analogy is that the embedder will embed the watermark with one key, and the detector will detect the watermark with another. Since these two keys are different, perhaps this will prevent an adversary with a detector from removing the watermark.

This analogy is not just flawed, it is wrong [13]. It is true that these schemes provide a better robustness against average attack, Principal Component Analysis (PCA), and oracle attack. However, the disclosure of the detection key permits specialized closest-point attacks that prevent detection while maintaining a good perceptual quality [14]. Hence, they are bad candidates for providing the read-but-not-remove capability. This proves that asymmetry is not sufficient, and

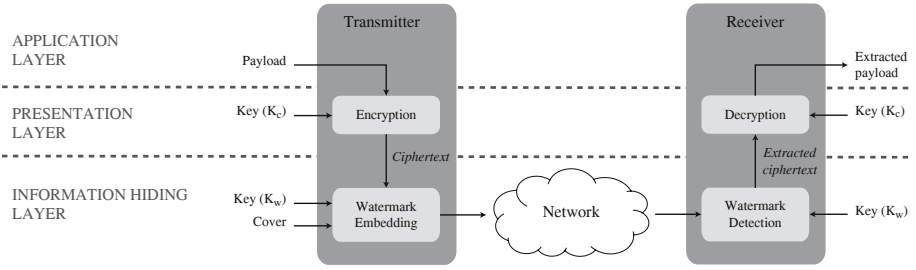


Fig. 1. Layered architecture for watermarking systems: a cryptographic primitive is simply added on top of the watermarking algorithm to provide security

indeed, it is not sure that it is even necessary to achieve the read-but-not-remove capability [15].

5 The Layered Approach

We believe that a layered approach to the design of secure watermarking systems, rather than an intermingling of the two fields of watermarking and cryptography, better ensures security for most applications.

We call a layered architecture a system where cryptography and watermarking primitives are well separated, as depicted in Figure 1, to implement a complete watermarking *system*. This approach is first motivated by its similarity with the popular open systems interconnection model (OSI model [16]). Using the OSI terminology, the information hiding layer is represented by the session layer (synchronization), the transport layer (error correction) and the physical layer (transmission). The presentation layer (encryption) is overlaid on top and the cleartext can be retrieved at the highest layer. Indeed, the 'encryption' box in Figure 1 has to be understood in a wide sense: the watermarking algorithm receives as input, the output of any cryptographic primitive, not only encryption.

These two layers have very different levels of security. The lower level, watermarking, has no security in the cryptographic sense. That is, it provides no protection with respect to privacy, authentication or integrity. It is only concerned with the reliably transmitting the (encrypted) bits. Since security for watermarking encompasses not just cryptographic security, (privacy, authentication, integrity), but also detection and removal, it is almost always the weakest link, as illustrated in Section 4.1 by the keyspace analogy.

Assume that cryptography is infinitely more secure than watermarking. Hiding a ciphertext, as depicted in Figure 1, forbids unauthorized embedding and decoding. However, it is absolutely useless against watermark removal or jamming. Hence, the layered approach does not necessarily bring a higher security level. However, it clearly separates the functionality of the two complementary technologies and reduces the risk of applying an inappropriate technique to solve a specific security issue.

5.1 Case Studies

Content Authentication. In this context, the goal is to ensure that the protected content has not been tampered with. Both cryptography and watermarking offer a technical solution to this challenge. Nevertheless, each one of them has its own shortcoming.

In cryptography, authentication is achieved by appending a digital signature or MAC to the content. However, 'appending' something to the content introduces an overhead i.e. more information has to be transmitted. And there is the risk that the MAC may be "lost" during format conversions.

Early proposals in watermarking suggested to embed a client-dependent watermark in the content. For example, the least significant bits (LSB) of an image are set to match a specific pseudo-random sequence. In this case, the drawback is that the watermark signal is not dependent of the protected content and can be copied to another one [17].

Combining both technologies immediately comes to mind as a means to avoid these shortcomings. One can indeed embed a watermark which encodes the digital signature or MAC of the content. When receiving a content, the user simply has to compare the digital signature of the content with the one stored by watermarking to validate the authenticity of the document. In this case, the watermark can no longer be copied from one content to the other because the watermark is now content-dependent. Moreover, no overhead is introduced. The only caution to be taken is that the watermarking process should not modify the bits of the content used to compute the digital signature of MAC. For instance, if the watermark is embedded in the LSB of an image, the digital signature should only be computed on the 7 most significant bits (MSB) of the pixels in the image.

A cryptographic hash provides exact authentication: a single bit change and the content is reported to be corrupted. In practice, a more flexible authentication of multimedia content may be desired to account for the various signal processing primitives (filtering, lossy compressions, etc) that do not modify the *semantic* meaning of the content. This has led to the introduction of 'robust hash functions' which will be further described in Section 6.1.

Traitor Tracing. In a typical fingerprinting scenario, Alice owns a few high valued multimedia items and wants to distribute these to a large number of customers. However, she is concerned that one or more of the customers may illegally redistribute her assets, thus inducing a loss of revenues. To address this issue, Alice introduces some customer-dependent modifications before distributing her assets. If a copy is found on an illegal distribution network, Alice looks for these modifications which serve as a fingerprint to trace back the 'traitor' who has broken his/her license agreement. Knowing Alice's strategy, a small set of users, usually referred to as a *collusion set*, may compare their individual copies, detect where they differ and create a new copy which potentially no longer contains a valid fingerprint. Alice's goal is then to design her system so that she can cope with such behaviours.

Traitor tracing has been studied by cryptographers where the problem is usually cast as the design of collusion-secure codes. A fundamental hypothesis is that, in combining their fingerprinted versions, a collusion inherently obeys a rule, called the marking assumption [18]. The most common marking assumption is that the set of colluders can only alter those bits of the codeword that differ between colluders. That is, those bits of each colluder's fingerprint (codeword) that are identical for all the colluders remain unchanged after a collusion attack. This assumption leads to the notion of a feasible set, also referred to as set of descendants, which is the collection of fingerprints that the collusion may produce. Traitor tracing needs collusion-secure codes designed so that each element in the feasible set can be linked back to at least one of the colluders as long as the number of colluders does not exceed a given limit.

Many recent fingerprinting solutions follow a layered architecture [19]. The cryptographic customer-dependent collusion-secure codeword is the payload embedded by the watermarking algorithm, and digital watermarking is simply exploited as a means to transmit the customer fingerprint, from one point to the other.

5.2 Knowledge of Lower Layers

In a layered model, the layers below do not need to know about the layers above. A function at layer i will accept inputs from layers above, but the function does not need to know how the inputs or outputs are interpreted by the layers above. However, the design and implementation of layers above may need a knowledge and understanding of the lower level protocols.

To illustrate this, let us re-examine the problem of traitor tracing[4]. The marking assumption on which collusion-secure codes are based is a model of the errors that can occur once the fingerprint is transmitted. These errors occur within the lower layers and can therefore be considered as a model of the 'noise' present in these levels. The most common marking assumption assumes that the set of colluders cannot alter those bits of the codewords that are common across all colluders.

We believe that this marking assumption is valid for watermarking algorithms that embed each bit of the fingerprint independently. For example, standard spread spectrum (SS) and quantization index modulation (QIM) techniques fit this model very well. However, more recent watermarking algorithms introduce dependency between successive embedded symbols in order to achieve higher embedding rates [23,24]. Thus, if one bit it altered, this may result in multiple successive bit errors at the decoder. Under these circumstances, the marking assumption would no longer be valid. For example, if two colluders are assigned two fingerprints that differ in only one bit, then the marking assumption states that all the remaining bits should be preserved. However, this one bit error may introduce a burst error at the decoder. Therefore, the extracted codeword may differ in bits that are common to both colluders.

[4] We will assume that the information hiding layer is secure enough to avoid estimation attacks [20,6]. In other words, a proper key scheduling policy [21,22] is enforced to prevent information leakage and subsequent jamming of the watermarking channel.

This example highlights the need for the upper levels to understand the workings of the lower levels. If a lower level watermarking algorithm is based on SS or QIM, then traditional collusion codes are applicable. However, if the lower level watermarking algorithm is based on dirty paper trellis coding, then a different type of collusion code must be designed and used.

6 Extension to Other Signal Processing

We would like to end this paper by considering the relationships between cryptography and signal processing other than watermarking.

6.1 Robust Hash

Section 5.1 has highlighted the need for 'flexible' hash functions which would output the same binary hash for perceptually similar contents. The quest for such a functionality has been previously explored in multimedia indexing and biometrics. This is usually referred to as 'robust hash', perceptual hash, soft hash or passive fingerprint [25,26].

Ideally, a robust hash would have the properties of:

1. it is easy to compute the hash value given a pre-image,
2. it is computationally unfeasible to generate a pre-image that hashes to a particular value,
3. it is hard to generate two perceptually different pre-images with the same hash value, and
4. only a perceptually significant change to the pre-image results in a change to the hash value.

Only the last two properties are different from the definition of a hash provided in Section 3. Nevertheless, the notions of "perceptually different" and "perceptually significant change" are very difficult to define.

Many researchers have attempted to realize a robust hash by designing completely new 'hash' functions. However, the design of hash functions is very difficult as revealed by incremental works [27,28]. Even cryptographic hash functions such as MD3 and SHA-1 have recently been shown to be partially flawed.

A layered approach would continue to use a cryptographic hash. The robust hash would be built on top of the cryptographic hash in a manner proposed by [29]. In such a design, the robust hash accepts the input content, e.g. an image, and extracts a robust *representation* of the content. It is this robust representation that is crytographically hashed. And it is this robust representation that is only altered if the input content is perceptually altered. The advantage of this layered solution is obvious. First, we can utilized well-know and trusted cryptographic tools. And second, we can utilize the considerable body of work regarding robust representations of signals. Finally, if the robust hash fails, we know it must be a failure of the robust representation. However, without a layered approach, errors are more likely and their causes more difficult to determine.

6.2 Signal Processing in the Encrypted Domain

Traditionally signal processing has been applied prior to encryption. In fact, for many encryption algorithms, it would make no sense to apply signal processing to the encrypted signal. The result would be nonsense. However, there is recent interest in developing encryption algorithms that permit signal processing of the encrypted signal. The motivation stems from the need to perform signal processing operations on machines that may not be trusted. For example, when streaming encrypted content over the Web, proxy servers may need to perform transcoding in order to reduce the bandwidth of the signal to match the recipient's (fluctuating) channel capacity. However, the proxy server is not trusted by the content owner who therefore does not wish the proxy server to decrypt the signal prior to transcoding[5].

We do not believe that this paradigm breaks the layered model. Rather, the processing of the encrypted signal should occur above the encryption layer. At the signal processing layer, there is a need to understand the nature of the encryption algorithm in order to design properly functioning algorithms. However, from the lower level encryption/decryption perspective, it is irrelevant what signal processing has occurred between encryption and decryption, provided the resulting signal can be correctly decrypted.

7 Conclusion

The interest in digital watermarking is strongly motivated by multimedia security issues. Consequently, it is not surprising that a number of cryptographic analogies have been applied to watermarking. However, we have argued that many of these cryptographic analogies are misleading or incorrect.

To support our argument we examined the concept of keys used in both cryptography and watermarking and showed that their properties are very different. In particular, for spread spectrum watermarking, an n-bit key has a keyspace of only $2^{0.12n}$ which is very much less than the equivalent keyspace in cryptography. We also discussed the concept of public key watermarking and argued that this concept, i.e. read-but-not-write, does not arise from using different keys for embedding and detection.

Fundamentally, watermarking is communications and is therefore concerned with the reliable delivery of bits over an unreliable channel. This is not a problem that is addressed by cryptography. However, cryptography does have a role to play in the development of applications of watermarking. Specifically, well-known cryptographic algorithms can be used to guarantee the privacy, authenticity and integrity of messages embedded in multimedia content. However watermark security must also consider the threat of unauthorized removal, for which there is no cryptographic solution.

[5] This problem has been considered in [30]. However, their proposed solution does not require signal processing of the encrypted stream. Rather, the stream is split into several layers that are independently encrypted. Then, if transcoding is required, the high-resolution stream can simply be deleted.

As in traditional communication systems, we recommend the use of a layered architecture. In such a design, watermarking is responsible for the synchronization and delivery of bits while cryptography is responsible for guaranteeing their privacy, integrity and authenticity. This separation simplifies analysis and modification of application systems.

To demonstrate this, we discussed two application areas: content authentication and traitor tracing. We explained how a layered design using cryptography and watermarking can be used to provide for both exact and approximate authentication (robust hash). Our discussion of traitor tracing served to highlight the point that in a layered architecture it may be necessary for the higher layers to understand the details of the lower levels. However, the lower level functions do not need to know how the upper layers will interpret signals.

We briefly commented on the recent interest in applying signal processing to encrypted signals and observed that such an approach can still be accommodated within a layered framework.

In summary, we hope this paper has clearly distinguished the roles of digital watermarking and cryptography and encouraged the use of a layered framework to the design of watermarking applications. We hope that these considerations will be taken into account in future proposals to combine cryptograpy and watermarking, together with other rules of good design [31].

References

1. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal **27** (1948) 379–423 & 623–656
2. Kahn, D.: Cryptology and the origins of spread spectrum. IEEE Spectrum **21** (1984) 70–80
3. Costa, M.: Writing on dirty paper. IEEE Transactions on Information Theory **29** (1983) 439–441
4. Shannon, C.: Communication theory of secrecy systems. Bell System Technical Journal **28** (1949) 656–715
5. Kerckhoffs, A.: La cryptographie militaire. Journal des sciences militaires **IX** (1883) 5–83
6. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: Theory and practice. IEEE Transactions on Signal Processing, Supplement on Secure Media **53** (2005) 3976–3987
7. Comesaña, P., Pérez-Freire, L., Pérez-Gonzàlez, F.: Fundamentals of data hiding security and their applications to spread spectrum analysis. In: Proceedings of the 7th International Workshop on Information Hiding. Volume 3727 of LNCS. (2005) 146–160
8. Pérez-Freire, L., Comesaña, P., Pérez-Gonzàlez, F.: Information-theoretic analysis of security in side-informed data hiding. In: Proceedings of the 7th International Workshop on Information Hiding. Volume 3727 of LNCS. (2005) 131–145
9. Pérez-Freire, L., Pérez-Gonzàlez, F., Comesaña, P.: Secret dither estimation in lattice-quantization data hiding: a set-membership approach. In: Security, Steganography, and Watermarking of Multimedia Contents VIII. Volume 6072 of Proceedings of SPIE. (2006) 6072–0W

10. Hartung, F., Girod, B.: Fast public-key watermarking of compressed video. In: IEEE International Conference on Image Processing. Volume I. (1997) 528–531
11. Furon, T., Duhamel, P.: An asymmetric public detection watermarking technique. In: Proceedings of the Third Information Hiding Workshop. Volume 1768 of Lecture Notes in Computer Science. (1999) 88–100
12. Eggers, J., Su, J., Girod, B.: Public key watermarking by eigenvectors of linear transforms. In: Proceedings of the European Signal Processing Conference. Volume III. (2000)
13. Furon, T., Duhamel, P.: An asymmetric watermarking method. IEEE Transactions on Signal Processing, Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery **51** (2003) 981–995
14. Furon, T., Venturini, I., Duhamel, P.: Unified approach of asymmetric watermarking schemes. In: Security and Watermarking of Multimedia Contents III. Volume 4314 of Proceedings of SPIE. (2001) 269–279
15. Miller, M.L.: Is asymmetry watermarking necessary or sufficient? In: Proceedings of European Signal Processing Conference. Volume I. (2002) 292–294
16. Zimmermann, H.: OSI reference model - the ISO model of architecture for open systems interconnection. IEEE Transactions on Communications **28** (1980) 425–432
17. Kutter, M., Voloshynovskiy, S., Herrigel, A.: Watermark copy attack. In: Security and Watermarking of Multimedia Contents II. Volume 3971 of Proceedings of SPIE. (2000) 371–380
18. Boneh, D., Shaw, J.: Collusion secure fingerprinting for digital data. IEEE Transaction on Information Theory **44** (1998) 1897–1905
19. Trappe, W., Wu, M., Wang, Z., Liu, K.: Anti-collusion fingerprinting for multimedia. IEEE Transaction on Signal Processing, Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery **51** (2003) 1069–1087
20. Doërr, G., Dugelay, J.-L.: Security pitfalls of frame-by-frame approaches to video watermarking. IEEE Transactions on Signal Processing, Supplement on Secure Media **52** (2004) 2955–2964
21. Lin, E., Delp, E.: Temporal synchronization in video watermarking. IEEE Transactions on Signal Processing, Supplement on Secure Media **52** (2004) 3007–3022
22. Harmancı, Ö., Kucukgoz, M., Mıhçak, K.: Temporal synchronization of watermarked video using image hashing. In: Security, Steganography and Watermarking of Multimedia Contents VII. Volume 5681 of Proceedings of SPIE. (2005) 370–380
23. Miller, M.L., Doërr, G., Cox, I.J.: Applying informed coding and informed embedding to design a robust, high capacity watermark. IEEE Transactions on Image Processing **13** (2004) 792–807
24. Abrardo, A., Barni, M., Pérez-Gonzàlez, F., Mosquera, C.: Trellis-coded rational dither modulation for digital watermarking. In: Proceedings of the 4th International Workshop on Digital Watermarking. Volume 3710 of LNCS. (2005) 351–360
25. Fridrich, J., Goljan, M.: Robust hash functions for digital watermarking. In: Proceedings of the International Conference on Information Technology: Coding and Computing. (2000) 178–183
26. De Roover, C., De Vleeschouwer, C., Lefèbvre, F., Macq, B.: Robust video hashing based on radial projections of key frames. IEEE Transactions on Signal Processing, Supplement on Secure Media **53** (2005) 4020–4037
27. Kozat, S., Venkatesan, R., Mıhçak, K.: Robust perceptual image hashing via matrix invariants. In: Proceedings of the IEEE International Conference on Image Processing. Volume V. (2004) 3443–3446

28. Monga, V., Mıhçak, K.: Robust image hashing via non-negative matrix factorizations. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. Volume II. (2006) 225–228
29. Sun, Q., Ye, S., Lin, C.-Y., Chang, S.-F.: A crypto signature scheme for image authentication over wireless channel. International Journal of Image and Graphics **5** (2005) 1–14
30. Wee, S., Apostolopoulos, J.: Secure scalable streaming enabling transcoding without decryption. In: Proceedings of the IEEE International Conference on Image Processing. Volume 1. (2001) 437–440
31. Katzenbeisser, S.: On the integration of watermarks and cryptography. In: Proceedings of the 2nd International Workshop on Digital Watermarking. Volume 2939 of Lecture Notes in Computer Science. (2003) 50–60

Secure Quantization Index Modulation Watermark Detection

Ton Kalker[1] and Mike Malkin[2]

[1] Hewlett Packard Laboratories, Palo Alto, USA
ton.kalker@hp.com
[2] Stanford University, USA
mikeym@cs.stanford.edu

Abstract. In this paper we introduce the problem of watermark security for systems in which an implementation of a watermark detector is available to an attacker. This paper serves as an introduction to a keynote talk at IWDW 2006. This talk will review two homomorphic encryption methods, viz. Paillier and Goldwasser & Micali, and their application to secure detection of Quantization Index Modulation (QIM) watermarks.

1 Watermarking Security

This paper addresses security aspects of watermark detection. Because there is not a single notion of security in the context of watermarking, we start this paper by expanding on the problem that this paper is discussing.

1.1 Setup

At the most abstract level, a watermarking scenario involves three parties, a message m and a digital object C, in this paper referred to as the *cover work*. One of the parties, in this paper referred to as Simon, takes the cover work C and modifies it to a *marked work* C_m such that the quality of C is close to that of C_m. Simon then sends this marked work to the recipient, referred to in this paper as Rob. The transmission may possibly be intercepted and the marked work be modified by a man in the middle (referred to as Evan). As Simon, Evan is constrained to maintain the *quality* of the work and therefore the quality of the *suspect work* Γ_m that Rob receives is close to that of C_m. In particular, Evan is not allowed to break the transmission by sending a different work or even nothing. Depending on the application, Rob then either attempts to establish the presence of message (in case Rob is not guaranteed that Γ_m contains a message) or to read the message m (in case Rob is guaranteed that a message m is present).

The intention of Simon and Rob is to set up a secret communication channel that cannot be *observed and/or created* by any third party. To this end Simon and Bob employ secret keys K_e and K_d for embedding and detection, respectively. Currently in all known watermarking systems the embedding and detection keys are essentially the same. In analogy with cryptographic terminology such watermarking systems are labeled *public*. Similarly, systems in which the keys are essentially different are called *asymmetric*.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 16–18, 2006.

1.2 Attacks

The goal of Evan is to break the secrecy communication channel in a number of possible ways. In the class of *blind attacks* Evan is not in any way able to observe (and replay) the actions of Simon and Rob. In particular, Evan has no access to secret keys and is not able to observe the outcome of watermark detections by Rob. The *jamming attack* is the most common blind attack. In this attack Evan tries to disrupt the communication channel between Simon and Rob without obtaining knowledge on the result of his efforts (at least, not directly). Evan may improve the efficiency of his attack by using several transmissions. This type of attack is typically relevant in forensic tracking scenarios.

In the class of *informed detection attacks*, Evan is able to observe the outcome of the watermark detections by Rob. In the *oracle attack* [2] Evan observes the results of watermark detections on many suitably modified versions of a single marked work in order to obtain knowledge on secret keys. It has been shown that oracle attacks are extremely powerful and that most spread-spectrum watermarking systems are easily broken by this attack.

In the oracle attack Evan has access to the watermark detector as a *black box* and is only able to observe input/output behavior. However, in many copy protection scenarios, an implementation of the watermark detector is available to Evan. This will allow Evan not only to observe input/output but also the operation of the watermark detector: the watermark detector is available to Evan as a *glass box*. In particular, without proper precautions, the secret detection keys will be visible to Evan. This is particularly worrisome for symmetric watermarking systems as this will reveal the embedding keys to Evan as well.

2 Homomorphic Watermark Detection

Currently there are two proposed solutions against the glass box attack. The first solution is the use of an asymmetric watermarking system. However, as these systems are not known to exist yet, this solution is not very practical yet. The alternative solution is to deploy homomorphic encryption combined with a trusted module.

2.1 Homomorpic Encryption

An encryption system is called homomorphic when the encryption operation commutes with a set of specified other operators. In our case, we are particularly interested in encryption systems that commute with addition and multiplication. Well-known examples are the Paillier [6] and Goldwasser & Micali [3] systems. The homomorphic property allows the computation of (semi-) linear functions in the encrypted domain.

2.2 Glass Box Resilience

Using homomorphic encryption, resistance against the glass box attack can be obtained by splitting the watermarking process into two modules. The public module performs the bulk of the watermark detection computations based upon homomorphically encrypted watermarking secrets. Evan is allowed to observe these computations, but assuming that Evan does not know the secret key K_h of the homomorphic system, he will

learn nothing. The output of the public module is an encrypted version of the watermarking message. This value is transmitted to a secure module that has access to the homomorphic key K_h and is able to decrypt the message and report it back to the public module. The secure module is a hardware platform that physically protects it internal secrets and computations and is therefore outside of the glass box. However, these secure platforms are typically resource limited (see for example the Trusted Platform Module by the Trusted Computing Group [7]). Splitting the watermark detection process in a bulk public but cryptographically protected part and a limited but physically secure part, emulates an overall opaque glass (i.e. black) box. Note that this homomorphic approach to the glass box attack does not provide resilience against the oracle attack!

This approach to secure watermark detection has successfully been applied to spread-spectrum watermark detection [4]. However, application of this approach to the non-linear Quantization Index Modulation method of Chen and Wornell [1] has proved to be more challenging.

2.3 Secure Quantization Index Modulation

Quantization Index Modulation (QIM) embeds a watermark into an signal by manipulating the signal so that transform coefficients are quantized in a specific manner. A watermark detector similarly transforms a suspect signal and checks to see if the transform coefficients are appropriately quantized. The quantization component of the watermarking system is in general difficult to integrate into a homomorphic encryption system. Using Goldwasser & Micali homomorphic encryption, a first successful approach has been described [5]. In IWDW 2006 keynote talk we will sketch the main ingredients of the approach as well as formal proofs of the security of the method.

References

1. B. Chen and G. W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423 – 1443, May 2001.
2. Ingemar Cox and Jean-Paul Linnartz. Public watermarks and resistance to tampering. In *Proceedings of the International Conference on Image Processing (ICIP'97)*, pages 26 – 29, 1997.
3. S. Goldwasser and S.Micali. Probabilistic encryption. *Journal of Computer and Systems Science*, 28:270 – 299, 1984.
4. Ton Kalker. Secure watermark detection. In *Proceedings of the 39th Allerton Conference On Communication, Control, And Computing*, 2005.
5. Mike Malkin and Ton Kalker. A cryptographic method for secure watermark detection. In *Proceedings of the Information Hiding Workshop*, 2006.
6. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Springer-Verlag, editor, *Proceedings of Eurocrypt '99*, volume 1592, page 223, 1999.
7. Trusted Computing Group. Trusted Platform Module, `https://www.trustedcomputinggroup.org/groups/tpm`.

Steganalysis in the Presence of Weak Cryptography and Encoding

Andreas Westfeld

Technische Universität Dresden
Institute for System Architecture
01062 Dresden, Germany
westfeld@inf.tu-dresden.de

Abstract. Steganography is often combined with cryptographic mechanisms. This enhances steganography by valuable properties that are originally left to cryptographic systems.

However, new problems for cryptographic mechanisms arise from the context of steganography. There are two sorts of steganographic tools: commercial tools with insecure or badly implemented cryptography and academic proof-of-concepts that abstain from the actual implementation of the cryptographic part.

Comparably to cryptography, steganography evolves in an iterative process of designing and breaking new methods. In this paper we examine the encoding properties and cryptographic functionality of steganographic tools to enable the detection of embedded information in steganograms even if the embedding part was otherwise secure.

1 Introduction

Digital steganography has evolved from an art in the early 1990's to a mature discipline in computer science. As a consequence, the gap between academic research and application has widened. The current situation is characterised by a huge and fast-growing number of publicly available tools offering steganographic functionality "out of the box." Apart from a poorly reflected use of deprecated and weak embedding functions, these tools also comprise all indispensable functions for pre- and post processing of message data and steganographic objects, respectively. These lateral processing steps are usually not covered in academic research. Since documentation is scarce and developers may tend to neglect possible security impacts, it was initially expected that a large number of tools would be vulnerable to simple (and thus avoidable) mistakes in the pre- and post-processing steps. We verified this assumption on a quantitative basis by scrutinising a number of arbitrarily chosen steganographic tools. This paper will give a survey on the security of current end-user steganography. Empirical evidence was gathered about weaknesses in steganographic tools, due to mistakes originating outside from the embedding function itself.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 19–34, 2006.

The combination of steganography with cryptography delivers important and desirable security properties:

- Encrypting messages before embedding identifies authorised recipients by the knowledge of a secret key, or, combined with other key-dependent transformations, becomes a second line of defence.
- Permuting data values pseudo-randomly before embedding reduces the local embedding density and dilutes statistical measurements.
- Encoding of message bits can be used to alter distributions for better camouflage, or add redundancy for error correction and robustness.

Hence, cryptographic primitives and encoding principles are constitutional elements for secure steganographic systems; as they offer their desirable properties in other applications as well.

This paper is organised as follows: Sect. 2 describes our approach to find the steganographic tools, the cryptographic functionality of which we survey in Sect. 3. We encountered different types of weaknesses in steganographic tools. These are described in Sect. 4. For each type of weakness we developed at least one prototype, which we present in Sect. 5. The paper is concluded in Sect. 6.

2 Tool Search

Our search is based on a comprehensive collection of steganographic tools, which grew over several years and is updated from time to time. We observe other collections (e. g., Johnson [1], [2], and Petitcolas [3]) to gain search terms apart from generic ones like "steganographic tools" and "download." According to our definition, steganographic tools are programs targeted to the end-user, which embed data into carrier with the aim of secrecy of the fact that hidden data exists. This implies that watermarking algorithms, steganalysis tools, and pure cryptographic tools are not included. If we count multiple releases of the same tool or ports to different platforms as one tool, the collection comprises 96 tools.

All tools in the collection were installed, and the files shipped with the install packages were examined. Some quick tests were performed for the tools with undocumented algorithms, steganograms were produced with these tools and compared with the original to determine the embedding function, if it uses encryption, or straddles the steganographic changes over the whole carrier medium.

3 Employment of Cryptography in Steganographic Tools

65 % of the tools provide encryption of the message before embedding. The rest neglects encryption or assumes that the message is encrypted using extra software (see Table 1). Although it is hard or impossible to read encrypted messages without the correct key, the steganographic embedding can be detected in most cases. In addition, many encryption applications add header information to the encrypted message. If such header information can be extracted, this proves the

Table 1. Availability of encryption option

Cryptographic availability	Number of tools	Percentage of all tools
implemented	62	65 %
not implemented	28	29 %
not indicated	6	6 %
total	*96*	*100 %*

presence of the embedded message. Once this header and the encrypted message is extracted, a cryptanalyst can try to find out the message content.

We can divide the cryptography integrated in steganographic tools into two categories:

1. tools that employ widely used and standardised algorithms and
2. tools that use improvised or non-public cryptography, which are mostly not published and not reviewable.

Table 2 lists the number of implementations by their particular standardised cryptographic algorithms. Because a single tool can offer more than one encryption scheme, we count implementations instead of tools. The particular algorithms appear only in the table if three or more implementations are known. Schemes provided by two or fewer tools have been added to the group "Other." The table shows that the most popular encryption algorithm in steganographic software is Blowfish, closely followed by DES.

Finally, most of the improvised algorithms break Kerckhoffs' principle [4] because they are kept secretly. In contrast to open source algorithms (e. g., Blowfish, DES, etc.) they are not reviewed and evaluated by a wide community and should not be trusted.

4 Weaknesses

This section gives an overview of the weaknesses detected in our survey of steganographic tools (see Table 3). We found weaknesses in 18 tools, with some seeming to be systematic while others are very specific. We explicitly do not count broken embedding function as a weakness, because they are not caused in the cryptographic or encoding part.

The indicated weaknesses lead to more or less strong attacks. Similar to cryptography, we can define classes of different strengths.

Total break: One can separate carriers and steganograms *almost* completely. (In cryptography: recovery of the secret key)

Existential break: One can tell for some files that they are *almost* certainly a steganogram. (In cryptography: one can forge the signature for some text, but not all texts.)

Table 2. Types of encryption schemes

Cryptographic category	Encryption scheme	Number of implementations	Percentage of all implementations
Standardised		*86*	*80 %*
	Blowfish	15	13 %
	DES	10	8 %
	RC4	5	5 %
	Twofish	5	5 %
	ICE	4	4 %
	IDEA	4	4 %
	AES	3	3 %
	GOST	3	3 %
	PGP	3	3 %
	RSA	3	3 %
	Other	31	29 %
Improvised		*22*	*20 %*

Exclusion of innocuous files: One can tell for sure which files (usually many) have not been modified by the tool under attack.[1]

The definitions use the word "almost" because, in contrast to cryptography, one not only has to distinguish between "able" and "unable" to read some information, but also have to resolve its origin: was it deliberately there or by chance. In most cases, there is a very small probability for the latter.

4.1 Magic Prefix

One of the most common mistakes is the inclusion of unencrypted status information at a deterministic place in the file (e. g., the beginning of the file). An adversary can access this information and use it to separate steganographic objects from plain carrier. This degrades the affected tool to a simple cryptographic tool with a very bad payload to data ratio. The presence of a magic prefix leads to total breaks. Examples: BlindSide [5], Stegano [6]

4.2 Specification of the Length of the Hidden Message

Some tools embed the length of the hidden message as status information with a fixed number of bits. While this is not as easy to detect as deterministic redundancy (cf. Sect. 4.1), comparing the length information with the maximum

[1] In some cases, one cannot exclude post-processing. Then "exclusion" means that this tool was not the last one to process the image.

Table 3. Selected steganographic tools and their weaknesses

Tool name	Attack prototype	Magic prefix	Fixed length	Varying length	Attachment	Small key	Weak straddling	Improper encoding	Break
AppendX ..	apdet	—	—	—	×	—	—	—	excl.
BlindSide ..	bsdet	×	(×)	—	—	×	—	—	total
Camera/Shy	—	—	—	—	—	—	(×?)	(×)	(exist.)
Contraband	codet	—	×	—	—	(×)	×	—	excl.
Cryptobola.	—	—	—	—	—	—	(×?)	—	(exist.)
Encrypt Pic	epdet	—	×	—	—	—	(×)	—	excl.
Gif it up!...	giudet	—	×	—	—	—	(×)	×	excl.
Gifshuffle...	—	—	—	—	—	—	(×?)	—	(excl.)
Hermetic...	—	—	(×?)	—	—	—	(×?)	—	(excl.)
Invisible....	rdjpegcom	—	—	—	×	—	—	—	excl.
Jsteg.....	jsteglen.R	—	—	×	—	—	(×)	—	excl.
Masker.....	apdet	—	—	—	×	—	—	—	excl.
MP3Stego..	—	—	(×?)	—	—	—	(×)	—	(?)
Pngstego...	—	—	—	—	—	(×?)	—	—	(excl.)
Stegano....	stdet	×	×	—	—	—	×	(×?)	total
Steganos...	—	—	—	—	—	—	(×?)	—	(excl.)
Steggy.....	apdet	—	(×)	—	×	—	—	—	excl.
Ump3c.....	updet	—	—	×	—	—	×	—	excl.

×—weakness,
excl.—exclusion of innocuous files,
total—total break,
exist.—existential break,
?—slightly better than random guessing,
(...)—attack not implemented

capacity for a given file size can easily identify a large number of plain carriers. This ratio increases for smaller carrier files. Examples: BlindSide [5], Contraband [7], EncryptPic [8], Gif it up! [9], Stegano [6], Steggy [10].

A similar comparison between length information and capacity can be made for those tools that avoid reserving a fixed number of bits, but store a prefix of the size of the length information before the actual information. Here, the adversary on average gains a little less information compared to the case with

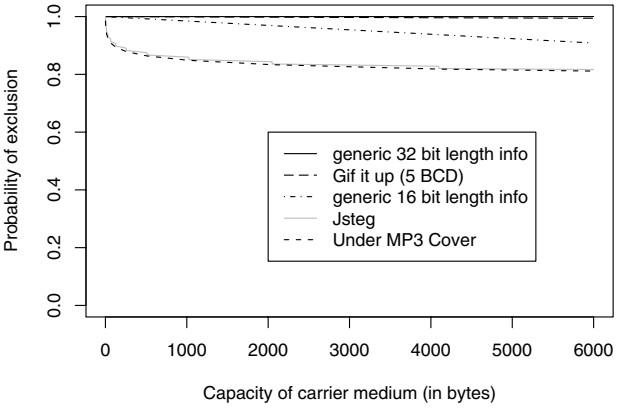


Fig. 1. Probability of exclusion by the attack under the assumption of uniformly distributed steganographic bits

fixed length information, but is still able to exclude innocuous files. Fig. 1 shows that the exclusion rate depends on the capacity of the carrier medium. Although the variable length field performs better, we can still exclude about 80 % of the carriers that provide less than 6 KB capacity. Examples: Jsteg [11], Under MP3 Cover [12].

4.3 Attached Message

We found some tools that do not actually alter the carrier data, but simply use special comment fields or undocumented areas at the file end to store the so-called "hidden" information. These tools provide almost no security. Surprisingly, quite a lot pursue this approach. However, because some commercial editors append their name to files or some executable files contain proprietary resources before file end, manual inspection is required for a total break. Machines can only exclude those files that don't have attachments. Examples: AppendX [13], Invisible [14], Masker [15], Steggy [10].

4.4 Small Keys

Another flaw is the insufficient length of the security parameter used for cryptographic functions. Some tools generate very short hashes from pass phrases to initialise random number generators. This enables an adversary to try the whole key space with brute force in reasonable time. Examples: BlindSide [5], Contraband [7], Pngstego [16].

4.5 Weak Straddling

A more specific vulnerability is the concentration of steganographic alterations in small parts of the carrier. Permutative straddling is a well-known and effective technique to avoid this problem. It is important to permute the alterations really

randomly, because an adversary can anticipate and exploit any deterministic step size (whether adaptive or not). This is not a severe weakness per se, but steganalytic attacks are much easier if the content is concentrated. Examples: Contraband [7], Encrypt Pic [8], Gif it up! [9], Jsteg [11], MP3Stego [17], Stegano [6], Under MP3 Cover [12].

4.6 Improper Encoding of the Embedded Message

Last but not least, a poor choice of source encoding—even after a cryptographic operation—introduces redundancies that can be detected by an adversary to identify steganograms, i. e., lead to existential breaks. Examples: Camera/Shy [18], Gif it up! [9].

5 Prototype Applications and Sample Sessions

All of the weaknesses classified in the previous sections have been successfully attacked with at least one prototype application. The prototypes are listed in Table 4. They are implemented in C/C++, and ready to download as source

Table 4. List of prototype applications

apdet	detects data appended to files
bsdet	determines a valid password for BlindSide
codet	excludes files not produced by Contraband
epdet	determines length information produced by EncryptPic
giudet	excludes files not produced by Gif it up!
stdet	detects data hidden with Stegano GIMP
updet	excludes files not produced by ump3c

code and Windows executables[19]. The usage is quite intuitive and does not require lengthy documentation: The programs are called from the command line with a filename of the medium under investigation as a parameter.

5.1 `apdet`—Detection of Appended Messages

The most primitive steganographic programs just append the message to the carrier medium. Although some of these programs are sold for a reasonable amount of money, their work can be done with simple commands that are already part of operating systems.

```
Unix:
    cat carrier.bmp readme.txt >steganogram.bmp
Windows:
    copy /b carrier.bmp+readme.txt steganogram.bmp
```

Such steganograms are more or less reliably detected by `apdet`. This tool currently supports the following carrier file formats: `.asf`, `.avi`, `.bmp`, `.dll`, `.exe`, `.gif`, `.jpg`, `.mid`, `.mov`, `.mp3`, `.mpg`, `.ocx`, `.snd`, `.tif`, and `.wav`. `apdet` needs a file name or directory that is automatically scanned (including subdirectories) for file names with the aforementioned suffixes. If additional bytes after the structurally expected end of the file are detected, the number of these bytes and the file name is written, otherwise `apdet` is silent.

```
$ apdet .
There are 333 bytes after the expected end of file!
... in file ./apstego.bmp
```

5.2 `bsdet`—Password Detection for BlindSide

`bsdet` detects steganograms created with BlindSide [5]. If the secret data are protected with a password, a string with four characters that is equivalent to the original password is derived and can be used to decrypt the embedded data. BlindSide hashes the password to a 16 bit key. Even worse: We don't have to search the whole key space. The structure of the embedded header (cf. Table 5) contains the known plaintext "OK," which can be used to determine the 16 bit key directly and convert it into an appropriate password. For instance, "cppe" is equivalent to the password "abc123," which was used to create the file `bsstego.encrypted.bmp`.

```
$ bsdet carrier.bmp
File does not contain any BlindSide hidden data.

$ bsdet bsstego.bmp
This file contains BlindSide hidden data.
Data is not encrypted, there is no password.

$ bsdet bsstego.encrypted.bmp
This file contains BlindSide hidden data.
Data is encrypted, a valid password is "cppe".
```

Table 5. Blindside message structure

Offset	Length (bytes)	unencrypted	encrypted
0	2	"BS"	"BE"
2	4	Length n	Encrypted length n
6	1	Version (0x01)	Encrypted version
7	2	"OK"	Encrypted "OK"
9	n	Message	Encrypted message

5.3 codet—Exclusion Despite Straddling

Contraband [7] uses a PIN[2] dependent straddling of the changes in the image. This PIN is needed to extract the length of the embedded message. This straddling is not very secure, and it is possible to extract the length with some errors. Nevertheless, `codet` is able to exclude a lot of carrier files from being steganographic. In the following example, we derive the number of leading zeros from the capacity. Since we know the earliest and latest bit positions of the (randomly straddled) length field, we can exclude files with less than 18 zeros in that part.

```
$ codet carrier.bmp
Analysing file carrier.bmp
capacity: 24000 bytes
--> 18 zeros expected in embedded length.
File does not contain any contraband hidden data.

$ codet costego.bmp
Analysing file costego.bmp
capacity: 24000 bytes
--> 18 zeros expected in embedded length.
min possible offset = 0
max possible offset = 3
There are 36 possible combinations for the first two digits of
the PIN.
possible sizes of embedded data: 1024, 2049, 4099, 8199 bytes

11100000000001000000000000000000001001000110100010100110101 <-- extracted bits
12345678901234567890123456789012345678901234567890123456789 <-- ruler
```

5.4 epdet—Exclusion by Length Information

Encrypt Pic [8] embeds the length of the message in the first 8 pixels. The capacity offered by this tool can be determined by the number of pixels divided by 2 (in bytes; one byte fits in two pixels). The image was not created by Encrypt Pic if the potential length specification exceeds the capacity−8. The length and other header information use 8 bytes of the capacity. The tool `epdet` compares the length and capacity to decide if a file is a potential steganogram.

```
$ epdet carrier.bmp
Analysing file carrier.bmp
capacity: 31992+8 bytes
File does not contain any encpic hidden data. (length
exceeds capacity: 118386139 > 31992)
11011011101101100111000011100000 <-- 32 length bits (LSB ...  MSB)
```

[2] Personal identification number, here: a four digit numeric password.

```
$ epdet epstego.bmp
Analysing file epstego.bmp
capacity: 31992+8 bytes
length of embedded data: 18713
10011000100100100000000000000000 <-- 32 length bits (LSB ...  MSB)
First two blocks of embedded data (2 x 128 bits)

01000001010011101110100110010011010011111010001010111011110010010000011100 0...
11000101101101110010111110011001010001000000011000110001000000101011001110110...
12345678901234567890123456789012345678901234567890123456789012345678901234567890 <-- ruler
```

Encrypt Pic can also encrypt the message with an unknown encryption algorithm. `epdet` dumps the first two 128 bit blocks of the ciphertext to the screen. If the plain text is periodically repeated (e. g., a series of zeros only), the ciphertext blocks are all the same. So at least the mode of operation used here is weak.

```
$ epdet epstego.periodic.bmp
Analysing file epstego.periodic.bmp
capacity: 290832+8 bytes
length of embedded data: 262144
00000000000000000010000000000000 <-- 32 length bits (LSB ...  MSB)
First two blocks of embedded data (2 x 128 bits)

00001101100100110010110111000000010111000111010010010101001001011111000011001...
00001101100100110010110111000000010111000111010010010101001001011111000011001...
12345678901234567890123456789012345678901234567890123456789012345678901234567890 <-- ruler
```

5.5 `giudet`—Exclusion by Length Information

Gif it up! [9] uses the first 20 pixels of a GIF file for storing the length of the embedded message. The size of the message is limited to 99,999 bytes. For each of the five digits, four bits are used to store the binary representation of the numbers 0 to 9, i. e., the length is encoded as binary coded digits (BCD). Consequently, media can be excluded from being steganographic in two ways. First one can exclude all length specifications that exceed the capacity of the current image as described in Section 4.2. Second, all digits > 9 (1010...1111) can be excluded because they are not BCD.

`giudet` uses this principle to detect valid length specifications embedded with "Gif it up!"

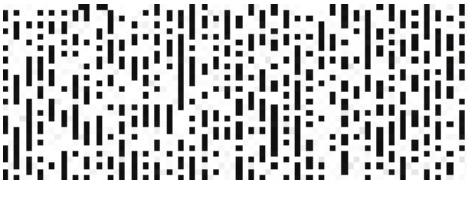


Fig. 2. Gif it up!: equidistant straddling (black: changed pixels)

```
$ giudet carrier.gif
Processing file carrier.gif
File does not contain any Gif-it-up hidden data.

$ giudet giustego.gif
Processing file giustego.gif
Possible length of embedded data: 2032
```

5.6 stdet—Detection by Magic Prefix

The embedding principle of Stegano [6] (a plug-in for GIMP) can be called LSB matching by increment (see Fig. 3). This is harder to detect than LSB replacement and the introduced error is smaller than with plus minus one steganography. Pixel values at the upper bound are only changed by decrement, because they cannot be incremented. Stegano is also the only known program that embeds in vertical direction, column by column from left to right (see Fig. 4).

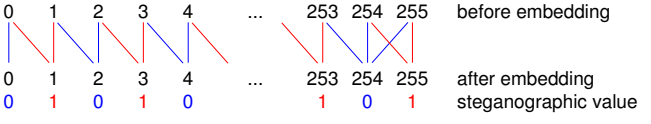


Fig. 3. Stegano: LSB matching by increment

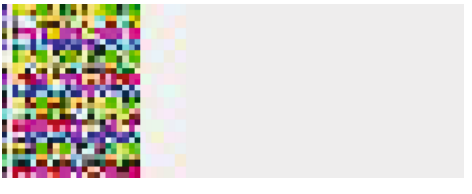


Fig. 4. Stegano: continuous embedding in vertical direction (coloured pixels are changed)

Stegano can be easily detected by the magic prefix `STG`, which is prepended to every embedded message. There is also information about the file name and length of the embedded message. Even if the embedded message is encrypted, which is possible only using an external tool, the prefix and header is still plaintext. This is exploited by `stdet`, which can reliably detect messages embedded with Stegano.

```
$ stdet carrier.bmp
Analysing file carrier.bmp
File does not contain any Stegano-GIMP hidden data.
```

```
$ stdet ststego.bmp
Analysing file ststego.bmp
Steganographic message detected.
embedded file name: /usr/users/mat02/s1924087/stegano/README
length of embedded data: 2140
```

5.7 updet—Exclusion by Length Information

Under MP3 Cover (ump3c [12]) is a tool that embeds one bit into every second block of an MP3 stream. As with some other tools, ump3c stores a length word in front of the actual data. Instead of reserving a fixed width, ump3c reserves 6 bits to specify the width. This makes the analysis interesting. Reasoning about the distributions of initial bits, under the assumption that normal MP3 files hold uniformly distributed bits in their first blocks[3], gives a theoretical ratio of arbitrary MP3 files that can be easily detected as non-steganographic.

```
$ updet carrier.mp3
Analysing file carrier.mp3
capacity: about 159 bytes
File does not contain any ump3c hidden data.

$ updet ump3c.stego.mp3
Analysing file ump3c.stego.mp3
capacity: about 159 bytes
Possible length specification found: 148
```

5.8 Analysis of Non-random Straddling Functions

MP3Stego [17] embeds into the parity of block lengths of MP3 streams. This tool is a good example of a specific straddling function: For each block a coin is tossed. On heads, the block is used for embedding, on tails not. As an embedding rate of 50 % was deemed too small by the author of the tool, MP3Stego ignores every third tail of the coin. This leads to an interesting probabilistic selection rule, which can be modelled as a finite state machine with deterministic initial state s_1 (see Fig. 5).

This finite state model is used to compute the expected embedding probabilities for the first n blocks of an MP3 stream and expected additional information for steganalysis. As shown in Fig. 5, the probability indeed vibrates around the steady state probability 0.6 in the first 8 blocks. Unfortunately, this convergence appears too fast, so that the additional information is marginal and does not lead to a significantly better ability to tell plain carriers apart. Nevertheless, this technique may lead to better results in cases where the parameters are chosen differently.

[3] Incidentally, we gain additional confidence from the fact that one of the leading bits is always zero, which seems to be a programming error.

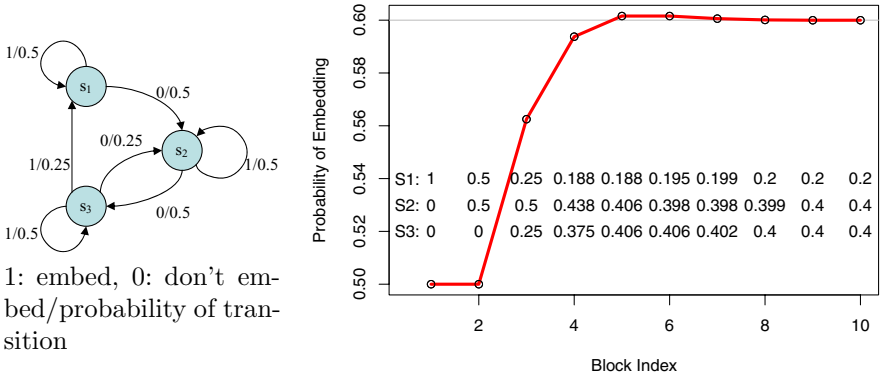


Fig. 5. Probability of embedding for the first 10 blocks with MP3Stego

5.9 Application of Jsteg Length Information to Jsteg Attacks

Jsteg length information can be used to reduce the detection power of targeted Jsteg attacks. We use two sets of 936 JPEG images to demonstrate—based on Jsteg length information—the generic improvement of the attacks of Zhang and Ping [20] and Lee et al. [21]. The first set C consists of the original carrier images from the CBIR database [22]. The second set S is derived from C with a Jsteg like algorithm (random embedding path) at 5 % embedding rate. Because of this small embedding rate, the attacks cannot perfectly separate both sets and will result in false positives or false negatives.

Jsteg embeds the length l of the embedded message (in bytes) in a variable length field at the beginning of the JPEG file. This is similar to ump3c (cf. Sect. 5.7). The first five bits tell the width of the length field ($\lceil \log_2 l \rceil = 0 \ldots 31$ bits). l is stored in these bits. Beside the length information one can determine also the capacity c of the JPEG image. This is the number of DCT coefficients that are neither 0 nor 1 minus the length of the length information ($5 + \lceil \log_2 l \rceil$). If

$$l = 0 \text{ or } l > \left\lfloor \frac{c - 5 - \lceil \log_2 l \rceil}{8} \right\rfloor,$$

Jsteg will not extract anything, and if the 6th bit in the extracted stream—the most significant bit of l—is 0 (in about 44 % of all cases), nothing has been embedded using Jsteg, because the length of the length field is always minimal. With these criteria we can find 717 (77 %) out of the 936 files in C that do not contain a proper Jsteg message. In 54 % the capacity was exceeded by the length specification. Fig. 6 shows the improved Receiver Operating Characteristic (ROC) of the attacks when they are combined with knowledge about the Jsteg length information. Table 6 shows the improved reliability $\rho = 2A - 1$ where A is the area under the ROC curve, the false positive rate (FPR) at 0.5 true positive rate (TPR) and the TPR for 1 % false positives.

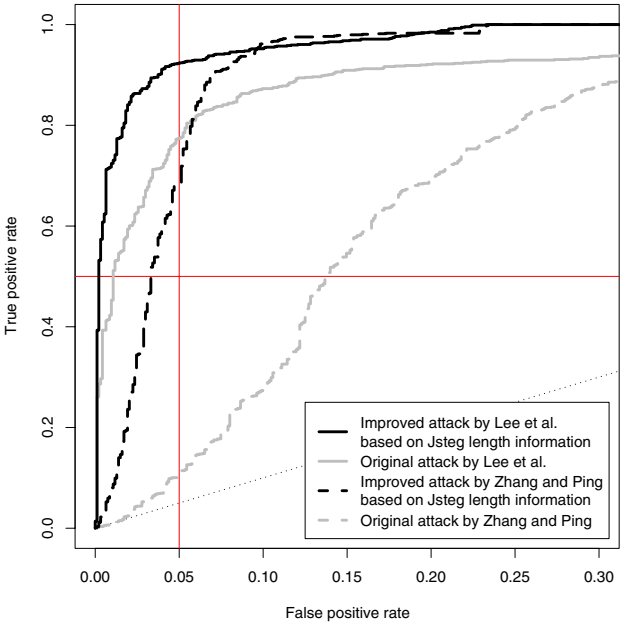


Fig. 6. ROC curves for generic improvement of targeted Jsteg attacks by exclusion of false positives based on Jsteg length information

Table 6. Reduced false positive rate and increased reliability ρ for targeted Jsteg attacks (5 % embedding rate)

Attack	ρ	FPR at 0.5 TPR	TPR at 0.01 FPR
Lee et al., original	0.8657	0.0107	0.4530
Lee et al., improved	0.9671	0.0021	0.7244
Zhang and Ping, original	0.6425	0.1389	0.0096
Zhang and Ping, improved	0.9163	0.0331	0.0801

6 Conclusion

The conclusions of this research are twofold. First, our prior belief that many end-user tools suffer from weaknesses in the encryption and encoding part has been confirmed. We presented a range of exemplary attacks against popular tools from the Internet, and the discovery of new weaknesses could be continued, although the academic outcome of analysing additional tools gets marginal.

Second, and more general, it remains the impression that the diversity of tools and carrier formats keep open the possibility for undetected steganographic communications, essentially because the resources to analyse every

tool and every exotic format[4] are limited. Even blind attacks cannot improve the situation substantially, because formats are not supported, features not developed, and training sets not available.

Acknowledgements

The author thanks Rainer Böhme for fruitful discussions, Benjamin Kellermann (né Scholz) for the implementation of `codet` and `apdet`, Kwangsoo Lee for providing his implementation of the category attack and 2 GB of test images, as well as Elke Franz for helpful comments on the paper. The work on this paper was supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under the research grant numbers FA8655-04-1-3036 and FA8655-06-1-3046. The U. S. Government is authorised to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on.

References

1. Johnson, N.F.: Steganography tools (2002) Online available at `http://www.jjtc.com/Security/stegtools.htm`
2. Johnson, N.F.: Steganography and digital watermarking tool table (2003) Online available at `http://www.jjtc.com/Steganography/toolmatrix.htm`
3. Petitcolas, F.A.P.: Steganographic software (2005) Online available at `http://www.petitcolas.net/fabien/steganography/stego_soft.html`
4. Kerckhoffs, A.: La cryptographie militaire. Journal des sciences militaires **IX** (1883) 5–38, 161–191 Online available at `http://www.petitcolas.net/fabien/kerckhoffs/crypto_militaire_1.pdf`
5. Collomosse, J.: Blindside (2000) Online available at `http://www.blindside.co.uk`
6. Cotting, D.: Stegano (2000) Online available at `http://registry.gimp.org/plugin?id=314`
7. Thijssen, J., Zimmerman, H.: Contraband (1998) Online available at `http://www.xs4all.nl/~jult/4u/contrabd.exe`
8. Collin, F.D.: EncryptPic (2000) Online available at `ftp://ftp.elet.polimi.it/mirror/Winsite/win95/miscutil/encpic13.exe`
9. Nelson, L.: Gif it up! (2004) Online available at `http://digitalforensics.champlain.edu/download/Gif-it-up.exe`
10. Iceman81: Steggy (2003) Online available at `http://mesh.dl.sourceforge.net/sourceforge/steggy/steggy0.1rc1.tar.gz`
11. Upham, D.: Jsteg (2000) Online available at `0urls.txt:http://munitions.vipul.net/software/steganography/jpeg-jsteg-v4.diff.gz`
12. Platt, C.: Ump3c (2004) Online available at `http://mesh.dl.sourceforge.net/sourceforge/ump3c/UnderMP3Cover-1.1.tar.gz`
13. Bauer, M.: AppendX (2003) Online available at `http://www.unet.univie.ac.at/~a9900470/appendX/apX`

[4] For example, Stegogo (http://sourceforge.net/projects/stegogo) embeds information into GNU Go Games.

14. Neobyte Solutions: Invisible (2000) Online available at http://www.neobytesolutions.com/downloads/invsecr.zip
15. Zaretskiy, E.: Masker (2001) Online available at http://www.softpuls.com/masker/
16. HappyCactus: Pngstego (2003) Online available at http://mesh.dl.sourceforge.net/sourceforge/pngstego/pngstego-0.3.2.tar.gz
17. Petitcolas, F.A.P.: MP3Stego (2001) Online available at http://packetstorm.trustica.cz/crypt/stego/SourceCode/MP3Stego_1.0.14b1_src.tar.gz
18. Hacktivismo: Camera/Shy (2002) Online available at http://mesh.dl.sourceforge.net/sourceforge/camerashy/CameraShy.0.2.23.1.exe
19. Westfeld, A., Kellermann, B.: Detection utilities (2005) Online available at http://dud.inf.tu-dresden.de/~westfeld/detectors/
20. Zhang, T., Ping, X.: A fast and effective steganalytic technique against JSteg-like algorithms. In: Proc. of the 2003 ACM Symposium on Applied Computing, Melbourne, Florida (2003) 307–311
21. Lee, K., Westfeld, A., Lee, S.: Category attack for LSB steganalysis of JPEG images (2006) In these proceedings.
22. University of Washington: CBIR image database (2004) Online available at http://www.cs.washington.edu/research/imagedatabase/groundtruth

Category Attack for LSB Steganalysis of JPEG Images

Kwangsoo Lee[1], Andreas Westfeld[2], and Sangjin Lee[1]

[1] Center for Information Security Technologies (CIST),
Korea University, Seoul, Korea
kslee@cist.korea.ac.kr, sangjin@korea.ac.kr
[2] Technische Universität Dresden,
Institute for System Architecture,
01062 Dresden, Germany
westfeld@inf.tu-dresden.de

Abstract. In this paper, we propose a new method for the detection of LSB embedding in JPEG images. We are motivated by a need to further research the idea of the chi-square attack. The new method simply use the first-order statistics of DCT coefficients, but is more powerful to detect the random embedding in JPEG images. For evaluation, we used versions of Jsteg and Jphide with randomized embedding path to generate stego images in our experiments. In results, the proposed method outperforms the method of Zhang and Ping and is applicable to Jphide. The detection power of both proposed methods is compared to the blind classifier by Fridrich that uses 23 DCT features.

1 Introduction

Steganography aims to hide the existence of secret messages by embedding them into ordinarily looking cover objects, while steganalysis aims to detect stego objects containing hidden messages [1]. Today, in digital era, the digital media such as image and audio are proliferated through the internet so that their transmissions are usual events in our daily life. Besides, many people have thought that digital media contain a lot of redundancies such as natural noises and quantized errors whose changes were expected to make no significant impacts on their perceptual and statistical properties. These have led people to research digital media for steganography, and in particular, digital images have been mostly researched for steganography.

The LSB embedding is a well-known steganographic method that is the way of replacing secret (usually encrypted) message bits with the least significant bits (LSBs) of sample values in digital media. It can be classified into two types according to the way of how to select media samples for carrying message bits. One is the sequential embedding in which message-carrying samples are selected in a fixed order that is publicly known, and the other is the random embedding in which message-carrying samples are randomly selected with a stego key that is shared by communication parties.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 35–48, 2006.

There are several attacks on LSB embedding. An earlier approach was proposed by Westfeld and Pfitzmann [2]. Their method, named the chi-square attack, exploits the first-order statistics (histogram) of samples in digital media. The chi-square attack works well for the sequential embedding, but not for the random embedding unless approximately all samples have been used for carrying message bits. Provos and Honeyman [3] presented an extended technique of the chi-square attack and tested it for JPEG based steganography such as Jsteg [11] and Jphide [12]. The extended chi-square attack works better for the random embedding such as OutGuess 0.13b [13] (OutGuess 0.13b can be viewed as a randomized version of the Jsteg algorithm). However, it seems that there still exists a limitation of the detection performance on the extended chi-square attack for the random embedding. Subsequent versions of OutGuess preserve the first-order statistics. Histogram-based attacks will fail to detect OutGuess, however, it is easily detected by comparing to calibrated statistics [4].

In this paper, we propose a new method for detection of LSB embedding in JPEG images. We are motivated by a need to further research the idea of the chi-square attack. The new method simply use the first-order statistics of DCT coefficients, but is more powerful to detect the random embedding in JPEG images. For evaluation, we used versions of Jsteg and Jphide with randomized embedding path to generate stego images in our experiments. In results, the proposed method outperforms another histogram-based detection by Zhang and Ping [7] and is applicable to Jphide. The detection power of both proposed methods is compared to the blind classifier by Fridrich [8] that uses 23 DCT features.

The paper organization is as follows: In the next section, we review the previous histogram-based attacks on LSB embedding. In Section 3, we describe the proposed approach towards an improvement of the idea of the chi-square attack, and some techniques to detect the Jsteg and Jphide embedding. In Section 4, we displays the experimental results of the proposed attack. In Section 5, we evaluate the detection reliability of the proposed attack and provide a fair comparison with previous methods to detect the randomized Jsteg and Jphide embedding. Finally, we conclude this paper in Section 6.

2 Histogram-Based Attacks on LSB Embedding

In this section, we briefly review the histogram-based attacks previously proposed for LSB steganalysis.

2.1 The Original Chi-Square Attack

Westfeld and Pfitzmann [2] proposed a categorical data analysis for detection of LSB embedding in digital images. LSB embedding induces categories of two sample values in which values only differ in the LSBs and so are possibly transformed into each other by LSB embedding operation. We will call them the induced categories throughout this paper, instead of the pairs of values (PoVs) named in their literature. To exemplify the induced categories, let us assume that the digital image is represented by a sequence of samples whose values are

integer numbers and all samples in the digital image are possibly used for carrying message bits. Then induced categories can be represented by the pairs of integer numbers, $(2m, 2m+1)$.

They discovered the fact that if a random message whose bits 0 and 1 are uniformly distributed is embedded in the LSBs of image data, the frequencies of sample values in each of PoVs are likely to be equal. This fact is generally untrue for cover images and was used for their categorical data analysis, named the chi-square attack. The chi-square attack is the way of measuring the degree of similarity between the observed sample distribution and the theoretically expected distribution in the induced categories, by means of a hypothesis test, the χ^2-test.

In order to give a formal description, let h_i denote the observed sample histogram. Then, for the induced categories $(2m, 2m+1)$, the observed distribution $\{o_m\}$ is given by

$$o_m = h_{2m} \ , \tag{1}$$

and the expected distribution $\{e_m\}$ is determined by

$$e_m = \frac{h_{2m} + h_{2m+1}}{2} \ . \tag{2}$$

The difference between the two distributions is measured by the following χ^2 statistics with $\nu - 1$ degrees of freedom,

$$\chi^2 = \sum_{e_m \neq 0} \frac{(o_m - e_m)^2}{e_m} = \frac{1}{2} \sum_{m \in Z} \frac{(h_{2m} - h_{2m+1})^2}{h_{2m} + h_{2m+1}} \ , \tag{3}$$

where ν is the number of different categories. The degree of similarity between the two distributions $\{o_m\}$ and $\{e_m\}$ is then calculated by the complement of the cumulative distribution function (CDF),

$$p = 1 - \int_0^{\chi^2} \frac{t^{(\nu-2)/2} e^{-t/2}}{2^{\nu/2} \Gamma(\nu/2)} dt \tag{4}$$

The p-value p is used for the decision of whether the image contains a secret message hidden with the LSB embedding or not.

To make an allowance for the detection of a sequential embedding, they implemented the χ^2-test on the recurrent samples of progressively increasing sizes, where the samples are selected in the same way of the sequential embedding. If the image contains a sequentially embedded secret message, the chi-square attack will show the result that the p-values are very close to 1 from the start of the test until rapidly fall down to 0 at the end of the hidden message (this can be additionally used to estimate the hidden message length). It is said that the chi-square attack is highly efficient for detecting the sequential embedding. It seems to be generally applicable to any types of digital images, and works very well for Jsteg [11] and Jphide [12] that are the sequential embedding for the JPEG image. However, it can hardly detect straddled messages unless approximately all samples have been used.

2.2 The Extended Chi-Square Attack

Provos and Honeyman [3] extended the chi-square attack by exploiting the sliding window of a fixed size to obtain the sample data, instead of increasing the window size. In this approach, it is important to find the appropriate window size for reliable detection. They implemented the χ^2-test for the shifted categories $(2m-1, 2m)$, $m \in Z$, and find the smallest window size that produces p-values bounded below a certain small threshold. This was formalized and adapted by Fridrich et al. [5]. The extended technique works better for the random embedding such as OutGuess 0.13b [13] that is a random LSB embedding for JPEG images. However, there still exists a limitation of the chi-square attack for the detection of small messages hidden with the random embedding [5].

2.3 Zhang and Ping Method

Another method to target the Jsteg-like algorithm was proposed by Zhang and Ping [7]. They considered the histogram shape of quantized DCT coefficients in JPEG images and assumed that the histogram has symmetry around zero. We will call their method the ZP attack. The following is a brief description of the ZP attack: Let h_i be the histogram of DCT coefficients in a JPEG image, where the indices i denote the values of the DCT coefficients. Let $f_0 = \sum_{i>0} h_{2i} + \sum_{i<0} h_{2i+1}$ and $f_1 = \sum_{i<0} h_{2i} + \sum_{i>0} h_{2i-1}$. In order to determine whether the JPEG image is stego or not, check that $f_1 > f_0$ and then calculate the statistics,

$$\chi^2 = \frac{(f_0 - f_1)^2}{f_0 + f_1} \ . \tag{5}$$

If χ^2 is greater than a certain small threshold, then the image will be determined as the stego image. As an additional information, the method can estimate the length of hidden message as the β value,

$$\beta = \frac{f_1 - f_0}{h_1} \ . \tag{6}$$

We have seen that ZP attack works better than the extended chi-square attack for the randomized Jsteg embedding, however, it does not work for the randomized Jphide embedding.

3 The Category Attack

In this section, we describe our approach to an improved histogram-based attack on LSB embedding. We name it the category attack.

3.1 Main Approach

In our development towards an improvement of the chi-square attack, we make a comparison of the induced categories and the shifted categories, while the authors of the extended chi-square attack used the shifted categories for the

appropriate window size. It is worth to mention that the p-value in Eqn. (4) is not a suitable measurement to discover small changes in sample distribution, that might happen when small messages are hidden with the LSB embedding. The reason is due to the fact that the CDF in Eqn. (4) is not activated unless the χ^2 value in Eqn. (3) decreases below a certain small quantity that depends on the degrees of freedom. In order to achieve an improvement of the chi-square attack, we discard the CDF, and instead use the χ^2 statistics like Eqn. (3) in the comparison of the induced categories and the shifted categories.

3.2 Basic Setting

Without loss of generality, let us assume that the digital image is represented by a sequence of samples whose values are integer numbers. Let X be the random variable of samples in a cover image, and let f_x be the probability distribution of X. Let us define the two statistics χ^2_{ind} and χ^2_{shi} as follows:

$$\chi^2_{\text{ind}} = \frac{1}{2} \sum_{m \in Z} \frac{(f_{2m} - f_{2m+1})^2}{f_{2m} + f_{2m+1}} \text{ , and}$$
$$\chi^2_{\text{shi}} = \frac{1}{2} \sum_{m \in Z} \frac{(f_{2m-1} - f_{2m})^2}{f_{2m-1} + f_{2m}} \text{ .} \tag{7}$$

χ^2_{ind} and χ^2_{shi} will be used as the overall statistics for the degree of difference between sample frequencies in the induced category and in the shifted category respectively. Let X' be the random variable of samples in the stego image which is generated by the LSB embedding with randomized embedding path in the cover image, and let $f'_{x'}$ be the probability distribution of X'. Let $\widetilde{\chi}^2_{\text{ind}}$ and $\widetilde{\chi}^2_{\text{shi}}$ be defined with the stego distribution f' in similar ways of Eqn. (7).

Let ℓ, $0 < \ell < 1$, be the length of hidden message relative to the number of usable samples for carrying message bits. We call ℓ the embedding rate. For example, Jsteg does not modify the quantized DCT coefficients of values 0 and 1, and thus the embedding rate ℓ for Jsteg is the relative length of hidden message in comparison with the number of quantized DCT coefficients unequal to 0 and 1. For Jphide, if we pretend that Jphide does not modify the quantized DCT coefficients of values -1, 0 and 1, the embedding rate ℓ is the relative length of hidden message in comparison with the number of quantized DCT coefficients unequal to -1, 0 and 1. In the subsequent development, for the ease of description, we will assume that all samples in the digital image are possibly used.

3.3 Effect on Induced Categories

Since the random embedding is considered, $\ell/2$ is then the probability that the LSB of a sample could be flipped by LSB embedding. So, we can establish the basic relation between the two distributions f and f' as follows: for $m \in Z$,

$$f'_{2m} = f_{2m} - \frac{\ell}{2}(f_{2m} - f_{2m+1}) \text{ , and}$$
$$f'_{2m+1} = f_{2m+1} + \frac{\ell}{2}(f_{2m} - f_{2m+1}) \text{ .} \tag{8}$$

It is clear that

$$f'_{2m} + f'_{2m+1} = f_{2m} + f_{2m+1} \text{ , and}$$
$$f'_{2m} - f'_{2m+1} = (1-\ell)\,(f_{2m} - f_{2m+1}) \text{ .} \tag{9}$$

This means that, after the LSB embedding, the category frequency (the sum of sample frequencies in the category) of the induced category $(2m, 2m+1)$ is not changed, but the difference between sample frequencies in the category linearly decreases as the embedding rate ℓ increases. It follows that

$$\widetilde{\chi}^2_{\text{ind}} = (1-\ell)^2 \chi^2_{\text{ind}} \text{ ,} \tag{10}$$

and therefore, the LSB embedding causes a decrease in the quantity of χ^2_{ind} for the induced categories:

$$\widetilde{\chi}^2_{\text{ind}} \ll \chi^2_{\text{ind}} \text{ .} \tag{11}$$

3.4 Effect on Shifted Categories

However, the above argument is not true for the shifted categories. From Eqn. (8), we can deduce that

$$f'_{2m-1} + f'_{2m} = f_{2m-1} + f_{2m} + \frac{\ell}{2}(f_{2m-2} - f_{2m-1} - f_{2m} + f_{2m+1}) \text{ ,}$$
$$f'_{2m-1} - f'_{2m} = f_{2m-1} - f_{2m} + \frac{\ell}{2}(f_{2m-2} - f_{2m-1} + f_{2m} - f_{2m+1}) \text{ .} \tag{12}$$

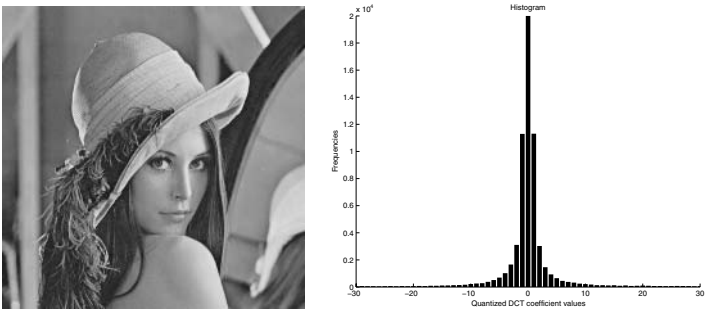
One can see that, for the shifted category $(2m-1, 2m)$, both changes of the category frequency and the difference between sample frequencies in the category are controlled by the frequencies of consecutive four sample values, where the two values are contained in the category and the other two values are externally adjacent to the category. This will lead to a contrast between the induced categories and the shifted categories under the LSB embedding.

In order to analyze the effect of LSB embedding on the difference between sample frequencies in the shifted category, we should define the relation among the frequencies of consecutive four sample values. To exemplify the relation, it is nice to consider the histogram of DCT coefficients in the JPEG image. Fig. 1 shows a part of the histogram of the well-known Lena image transformed in JPEG format with 75 % quality factor (we set the frequency of the coefficient value 0 to $2 \cdot 10^4$). There are a peak at the center of mass, slopes in both sides of the center, and tails at the edges with negligible probabilities. We note that the slopes which are monotonically increasing or decreasing appear in most of small intervals with a significant portion of the distribution, except for the interval containing the value 0 as an internal value.

Let us assume that the histogram is monotonically increasing or decreasing on consecutive four values for a shifted category $(2m-1, 2m)$[1]. That is,

$$f_{2m-2} < f_{2m-1} < f_{2m} < f_{2m+1} \text{ , or}$$
$$f_{2m-2} > f_{2m-1} > f_{2m} > f_{2m+1} \text{ .} \tag{13}$$

[1] It is reasonable when considering JPEG images, but generally untrue for other types of digital images.



(a) Lena (b) Histogram

Fig. 1. Histogram of quantized DCT coefficients for Lena image in JPEG format

From Eqn. (12), we can deduce that

$$f'_{2m-1} + f'_{2m} = f_{2m-1} + f_{2m} \pm \frac{\ell}{2}(|f_{2m-2} - f_{2m-1}| - |f_{2m} - f_{2m+1}|) \ ,$$
$$|f'_{2m-1} - f'_{2m}| = |f_{2m-1} - f_{2m}| + \frac{\ell}{2}(|f_{2m-2} - f_{2m-1}| + |f_{2m} - f_{2m+1}|). \quad (14)$$

After the LSB embedding, the difference between sample frequencies in the shifted category $(2m-1, 2m)$ linearly increases as the embedding rate ℓ increases. The category frequency in the shifted category is also altered, but the change is relatively small in comparison with the change of the frequency difference. Therefore, the LSB embedding causes an increase in the quantity of χ^2_{shi}:

$$\widetilde{\chi}^2_{\text{shi}} \gg \chi^2_{\text{shi}} \ . \quad (15)$$

3.5 Statistical Measurement

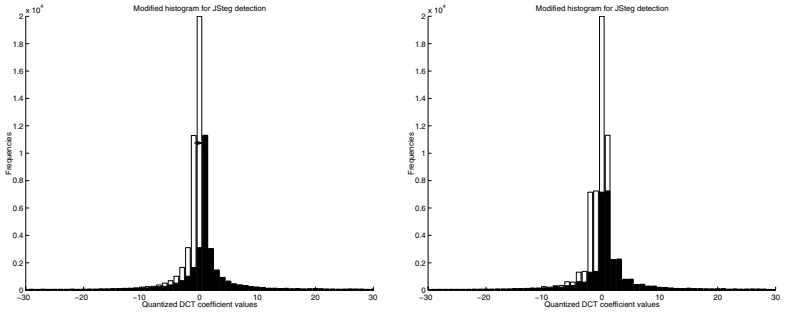
In summary, after the LSB embedding, the quantity of the statistics χ^2_{ind} for induced categories decreases, but the quantity of the statistics χ^2_{shi} for shifted categories increases. This will result in a great difference between the induced categories and the shifted categories under the LSB embedding. For the detection of LSB steganography, we decide to simply use the relative difference of the two statistics defined as follows:

$$R = \frac{\chi^2_{\text{shi}} - \chi^2_{\text{ind}}}{\chi^2_{\text{shi}} + \chi^2_{\text{ind}}} \ . \quad (16)$$

If there are some patterns in the value of R for a certain type of cover histogram, we can use them for LSB steganalysis of the digital image. And we have observed that the R statistics well discriminated between cover images and stego images in JPEG format, where stego images are generated by the Jsteg and Jphide with randomized embedding path.

3.6 Technique for Jsteg Detection

Jsteg [11] can be viewed as the LSB embedding with an exception for usable DCT coefficients; It does not modify the DCT coefficients of the values, 0 and 1. So, for detection of the Jsteg-like algorithm, we ignore them and increase the DCT coefficients of negative values by 2. Fig. 2 displays the modification of the histogram by the preprocessing of the Lena image in JPEG format before and after the Jsteg embedding with full capacity. The white bars are the histogram of all DCT coefficients. We strip two of them that are not used for steganography, namely h_0 and h_1, and regard only the resulting histogram with the black bars. One can see that the modified histogram of the cover image still maintains the symmetry around one value which can deduce that $\chi^2_{\text{ind}} = \chi^2_{\text{shi}}$. This is common for JPEG images and therefore R statistics can be used for the detection.

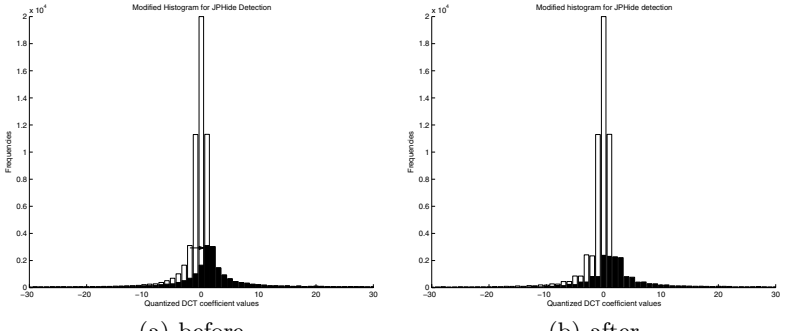


(a) before (b) after

Fig. 2. Histogram of Lena image in JPEG format before and after Jsteg embedding

3.7 Technique for Jphide Detection

Jphide [12] can be viewed as the LSB embedding in a sense that message bits are encoded in the LSBs of the absolute values of DCT coefficients; although



(a) before (b) after

Fig. 3. Histogram of Lena image in JPEG format before and after Jphide embedding

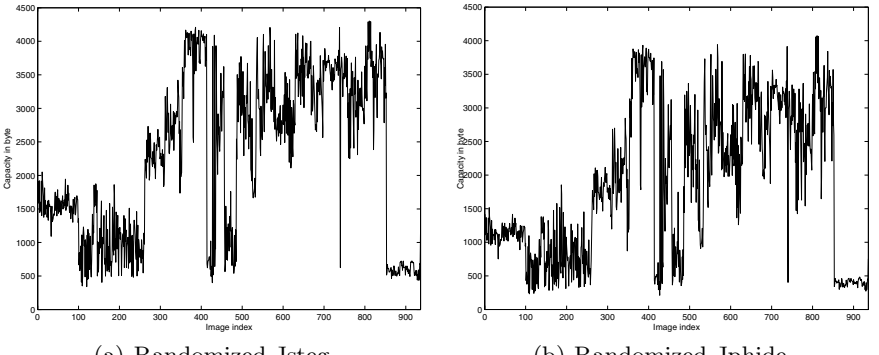
Jphide occasionally modifies the second least significant bits, but these are not frequent and the effect on the statistics is negligible. Jphide also modifies the DCT coefficients of values -1, 0 and 1 in a special way. So, we ignore them and increase the DCT coefficients of negative values by 3. This technique still allows us to detect the Jphide embedding. Fig. 3 displays the modified histogram by the preprocessing of the Lena image in JPEG format before and after the modified Jphide embedding with full capacity. Again, the steganographically unused values, namely h_{-1}, h_0 and h_1, are stripped from the histogram. In this case, however, the modified histogram of the cover image results in $\chi^2_{\text{ind}} > \chi^2_{\text{shi}}$.

4 Experimental Results

4.1 Image Sets

We used 936 JPEG images of the CBIR image database from Washington University [14]. We only used Y channel data of JPEG images in the test. The testing was done for 6 embedding rates, 5 %, 10 %, 20 %, 30 %, 40 % and 50 %, in bits per a usable coefficient (bpc) for each steganography algorithm. We embedded random messages in the LSBs of usable coefficients that are randomly selected from Y channel data. The random message here was newly generated for each stego image. In the simulation, we implemented randomized versions of Jsteg and Jphide algorithms; for the randomized Jphide algorithm, we did not use the DCT coefficients of values -1, 0 and 1, and embedded message bits in the LSBs of absolute values.

Fig. 4 shows message lengths (in bytes) hidden in the stego images for an embedding rate of 10 %. Fig. 4 (a) shows the message lengths for the randomized Jsteg embedding. The maximal capacity is 4302 bytes, and the minimal capacity is 340 bytes, and the average is 2267 bytes. Fig. 4 (b) shows the same for the randomized Jphide embedding. The maximal capacity is 4078 bytes, the minimal capacity is 211 bytes, and the average is 1846 bytes.

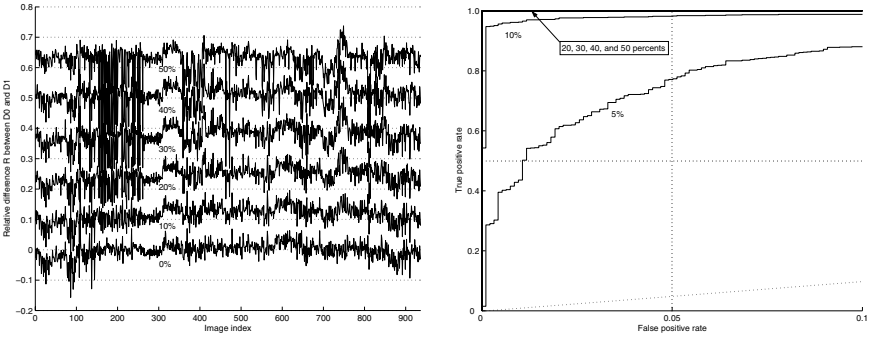


(a) Randomized Jsteg (b) Randomized Jphide

Fig. 4. Embedding capacities for stego images in case of the 10 % embedding rate

4.2 On Randomized Jsteg Algorithm

Fig. 5 displays the results of the proposed attack on the randomized Jsteg algorithm. In (a), one can see that the R statistics is highly sensitive to the low-rate embedding. Even for the 10 % embedding rate, most of stego images seems to be distinguished from the cover images. This well explains the ROC curves for the category attack on the randomized Jsteg algorithm in (b). The ROC curves show that, when the embedding rates are greater than or equal to 20 %, all stego images were perfectly separated from the cover image set.

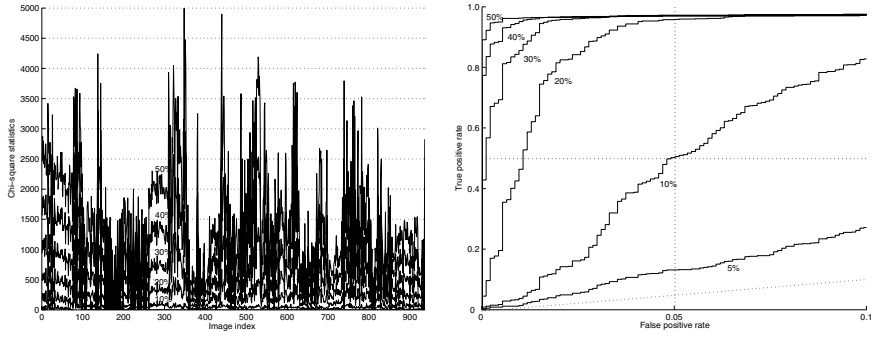


(a) Test Results for R Statistics (b) ROC curves for embedding rates

Fig. 5. Results for the category attack on the randomized Jsteg algorithm

4.3 Vs. ZP Attacks

Fig. 6 displays the results of the ZP attack on the randomized Jsteg algorithm. Here the χ^2 and the relation $f_1 > f_0$ were used. At a glance, the category attack outperforms the ZP attack.

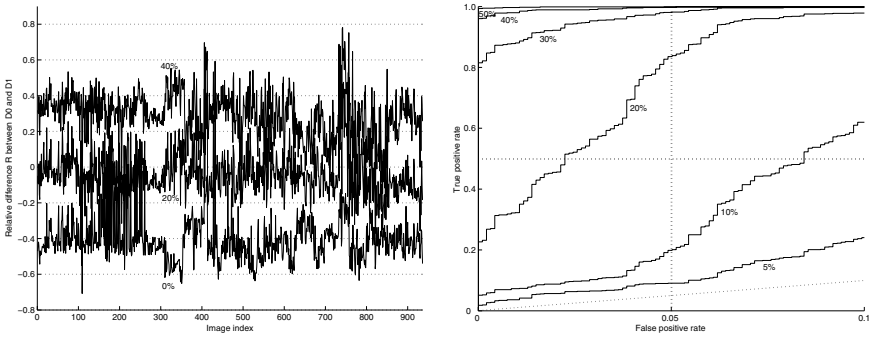


(a) Test Results for χ^2 statistics (b) ROC curves for embedding rates

Fig. 6. Results for the ZP attack on the randomized Jsteg algorithm

4.4 On Randomized Jphide Algorithm

Fig. 7 displays the results of the proposed attack on the randomized Jphide algorithm. The category attack works for the randomized Jphide embedding. However, the initial stats of R statistics for cover images are varied and these will disturb the correct decision for the low-rate embedding.



(a) Test Results for R Statistics (b) ROC curves for embedding rates

Fig. 7. Results for the category attack on the randomized Jphide algorithm

5 Evaluation and Comparison of Detection Reliability

Table 1 and 2 give different quality measures that are used in the literature. Fridrich measures the reliability ρ, defined by twice the area between the ROC curve and the diagonal ($\rho = 1$ means perfect separation, $\rho = 0$ equals random guessing) [8]. Lyu and Farid (LF) measures the true positive rate (TPR) at 1 % false positive rate (FPR) [6]. Ker requires the FPR to be less than 5 % at 50 % TPR [9]. We give the FPR for 50 % TPR in the table. For the LF and Ker values, we give also the separating thresholds. The "Mean" and "Var" columns contain the mean and variance values of the attack results for the respective set steganograms.

The results of the ZP algorithm presented here are based on the β value determined according to Eqn. (6). The original ZP algorithm decides using the χ^2 and the relation $f_1 > f_0$. This relation is also expressed by the sign of β and we found that the degree of negativity can slightly improve the detection reliability.

We implemented the 23 DCT features by Fridrich [8], extracted them from the 12168 files and trained a support vector machine[2] on 2×690 files for the 6 embedding rates and the 2 algorithms. We classified the remaining 2×246. Compared to targeted attacks, the result is rather poor for low embedding rates (this can be also assured at Fig. 8). However, the blind attack is universal and

[2] We use the SVM implementation from the e1071 package of the R software [15].

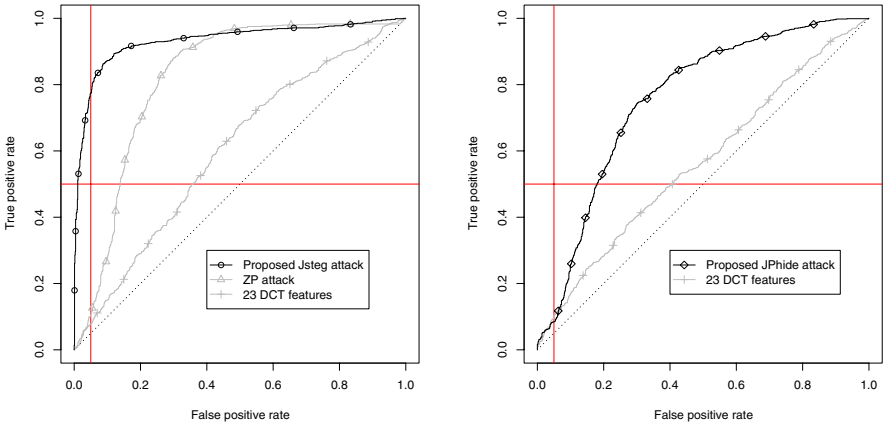
Table 1. Evaluation of randomized Jsteg detection power

Attack	Image set	ρ	Ker	Ker.thr	LF	LF.thr	Mean	Var
CA Jsteg	05	0.8657	0.0107	0.0568	0.4530	0.0600	0.0547	0.0008
	10	0.9897	0.0000	0.1178	0.9605	0.0605	0.1155	0.0009
	20	1.0000	0.0000	0.2466	1.0000	0.0613	0.2432	0.0012
	30	1.0000	0.0000	0.3802	1.0000	0.0613	0.3755	0.0015
	40	1.0000	0.0000	0.5132	1.0000	0.0613	0.5055	0.0016
	50	1.0000	0.0000	0.6361	1.0000	0.0613	0.6268	0.0017
23dctf	05	0.2078	0.3659	0.4011	0.0285	0.8990	0.4644	0.0380
	10	0.5600	0.1057	0.5626	0.0569	0.8997	0.5703	0.0444
	20	0.8469	0.0325	0.7158	0.2398	0.8643	0.7049	0.0602
	30	0.9387	0.0000	0.8074	0.5935	0.7600	0.7731	0.0574
	40	0.9775	0.0000	0.8616	0.7967	0.6505	0.8230	0.0472
	50	0.9935	0.0000	0.8947	0.8699	0.6137	0.8610	0.0381
ZP beta	05	0.6425	0.1389	0.0430	0.0096	2.1429	0.0767	0.6344
	10	0.8414	0.0577	0.0928	0.0096	2.1970	0.1246	0.5871
	20	0.9286	0.0203	0.1934	0.0085	2.1970	0.2204	0.4291
	30	0.9389	0.0171	0.2950	0.0075	2.1970	0.3203	0.3437
	40	0.9419	0.0171	0.3959	0.0075	2.1970	0.4183	0.2356
	50	0.9460	0.0160	0.4959	0.0064	2.1970	0.5140	0.1788

Table 2. Evaluation of randomized Jphide detection power

Attack	Image set	ρ	Ker	Ker.thr	LF	LF.thr	Mean	Var
CA Jphide	05	0.5133	0.1795	-0.3369	0.0331	-0.0051	-0.3176	0.0138
	10	0.7886	0.0812	-0.2396	0.0673	-0.0054	-0.2235	0.0139
	20	0.9392	0.0267	-0.0635	0.2949	-0.0057	-0.0453	0.0164
	30	0.9902	0.0000	0.1209	0.8643	-0.0051	0.1272	0.0178
	40	0.9981	0.0000	0.2983	0.9754	-0.0051	0.2871	0.0193
	50	0.9997	0.0000	0.4560	0.9957	-0.0051	0.4298	0.0204
23dctf	05	0.1393	0.4024	0.6847	0.0244	0.9398	0.6406	0.0405
	10	0.2716	0.2927	0.6204	0.0772	0.8828	0.6006	0.0417
	20	0.5099	0.1504	0.5804	0.1789	0.7938	0.5900	0.0473
	30	0.6661	0.0691	0.6191	0.2602	0.7854	0.6405	0.0494
	40	0.8028	0.0366	0.6742	0.3618	0.7592	0.6827	0.0454
	50	0.8907	0.0081	0.7169	0.5000	0.7169	0.7249	0.0404

detect algorithms like Outguess that preserve first-order statistics. Because the proposed attack decides based on histograms, it is unable to detect algorithms like Outguess.



(a) Jsteg Attacks (b) Jphide Attacks

Fig. 8. ROC curves showing the improved reliability of the category attack on randomized Jsteg and Jphide algorithms. Here, 5 % embedding rate is used for stego images.

6 Conclusion

In this paper, we proposed the category attack for LSB steganalysis of JPEG images. The category attack exploits simply the histogram of DCT coefficients, but is more powerful to detect the randomized Jsteg embedding as well as the randomized Jphide embedding. The proposed method outperformed the Jsteg detection by Zhang and Ping. The detection power of both proposed methods were compared to the blind classifier by Fridrich that uses 23 DCT features.

There seems to exist a relation between the R statistics used in the category attack and the length of hidden messages. The exact formula to estimate the hidden message length will be further researched.

Acknowledgements

This work was supported by grant No. M10640010005-06N4001-00500 from the national R&D Program of MOST and KOSEF.

References

1. S. Katzenbeisser and F.A.P. Petitcolas, *Information Hiding - techniques for steganography and digital watermarking*, Artech House Books, 1999.
2. A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in Information Hiding: 3rd International Workshop, IH'99 Dresden, Germany, September 29 – October 1, 1999 A. Pfitzmann, ed., LNCS 1768, pp. 61–76, Springer-Verlag, Berlin Heidelberg, 2000.

3. N. Provos and P. Honeyman, "Detecting Steganographic Content on the Internet," CITI Technical Report 03–11, 2001.
4. J. Fridrich, M. Goljan and D. Hogea, "Attacking the OutGuess," in Proc. of the ACM Workshop on Multimedia and Security 2002, Juan-les-Pins, France, December 6, 2002.
5. J. Fridrich, M. Goljan, and D. Soukal, "Higher-Order Statistical Steganalysis of Palette Images," in Proc. of EI SPIE, Santa Clara, CA, Jan 2003, pp. 178-190.
6. S. Lyu and H. Farid, "Detecting Hidden Messages Using Higher-Oder Statistics and Support Vector Machines," in Information Hiding: 5th International Workshop, IH2002, Noordwijkerhout, The Netherlands, October 7–9, 2002, F.A.P. Petitcolas, ed., LNCS 2578, pp 340–354, Springer–Verlag, Berlin Heidelberg, 2003.
7. T. Zhang and X. Ping, "A Fast and Effective Steganalytic Technique against Jsteg-like Algorithms," in ACM Symposium on Applied Computing, March 9–12, 2003, Florida, USA, 2003.
8. J. Fridrich, "Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes," in Information Hiding: 6th International Workshop, IH2004, Toronto, Canada, May 23–25, 2004, Revised Selected Papers, J. Fridrich, ed., LNCS 3200, pp. 97–115, Springer–Verlag, Berlin Heidelberg, 2004.
9. A. D. Ker, "Improved Detection of LSB Steganography in Grayscale Images," in Information Hiding: 6th International Workshop, IH2004, Toronto, Canada, May 23–25, 2004, Revised Selected Papers, J. Fridrich, ed., LNCS 3200, pp. 97–115, Springer–Verlag, Berlin Heidelberg, 2004.

Internet Sources

10. Steganographic Tool Lists, `http://www.stegoarchive.com`.
11. D. Upham, Jpeg-Jsteg, `http://www.funet.fi/pub/crypt/stegangraphy/jpeg-Jsteg-v4.diff.gz`.
12. Allan Latham, Jphide and JPSeek, `http://linux01.gwdg.de/$\sim$alatham/stego.html`.
13. Niels Provos, OutGuess – Universal Steganography, `http://www.outguess.org`.
14. CBIR Image Database, University of Washington, `http://www.cs.washington.edu/research/imagedatabase/groundtruth`.
15. R: A Language and Environment for Statistical Computing, R Development Core Team, R Foundation for Statistical Computing, Vienna, Austria, 2006, http://www.r-project.org,

Steganalysis Using High-Dimensional Features Derived from Co-occurrence Matrix and Class-Wise Non-Principal Components Analysis (CNPCA)

Guorong Xuan[1], Yun Q. Shi[2], Cong Huang[1], Dongdong Fu[2], Xiuming Zhu[1], Peiqi Chai[1], and Jianjiong Gao[1]

[1] Dept. of Computer Science, Tongji University, Shanghai, P.R. China
grxuan@public1.sta.net.cn
[2] Dept. of Electrical & Computer Engineering, New Jersey Institute of Technology
Newark, New Jersey, USA
shi@njit.edu

Abstract. This paper presents a novel steganalysis scheme with high-dimensional feature vectors derived from co-occurrence matrix in either spatial domain or JPEG coefficient domain, which is sensitive to data embedding process. The class-wise non-principal components analysis (CNPCA) is proposed to solve the problem of the classification in the high-dimensional feature vector space. The experimental results have demonstrated that the proposed scheme outperforms the existing steganalysis techniques in attacking the commonly used steganographic schemes applied to spatial domain (Spread-Spectrum, LSB, QIM) or JPEG domain (OutGuess, F5, Model-Based).

Keywords: steganalysis, co-occurrence matrix, class-wise non-principal components analysis (CNPCA).

1 Introduction

This paper[1] addresses universal image steganalysis under the framework of pattern recognition. The steganalysis is the counterpart of steganography. The purpose of the steganalysis is to detect the hidden message, equivalently, to discriminate the stego-object from the non-stego-object. The steganalysis techniques proposed in the literature can be classified into two categories: the universal steganalysis which is designed to detect the hidden message embedded with various data embedding algorithms, and the specific steganalyis which is designed to attack a specific steganography technique.

Farid [1] proposed a universal steganalysis algorithm based on high-order statistical moments derived from high-frequency wavelet subbands. These statistics are based on decomposition of images with separable quadrature mirror filters. The high-frequency subbands' statistical moments are obtained as features for

[1] This research is supported partly by National Natural Science Foundation of China (NSFC) on the project (90304017).

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 49 – 60, 2006.

steganalysis. It can differentiate stego-images from non-stego (also referred to as cover) images with a certain success rate. In [2], Xuan et al. proposed a universal steganalysis approach, which selects statistical moments of characteristic functions of the test image, and all of their wavelet subbands as features. This steganalyzer outperforms [1] in general. In [3], Fridrich developed a steganalysis scheme specifically designed for attacking JPEG steganography. A set of well-selected features for steganalysis are generated from the statistics of the JPEG image and its calibrated version. This scheme outperforms [1] and [2] in attacking the JPEG steganography, such as OutGuess [4], F5 [5], and Model-based (MB) [6]. The feature extraction algorithm of the steganalyzer [3], however, is complicated and takes time.

The above mentioned steganalysis schemes [1, 2] are both based on the statistics of the histogram of wavelet subbands. (Note that the scheme [2] is partially based on histogram of given test image as well.) Histogram itself is known as the first order statistics. In [3], in addition to the first order statistics, histogram, the second order statistics such as co-occurrence in the JPEG coefficient domain are also used to extract features. However, only the co-occurrence counted from some modes of JPEG coefficients between neighboring blocks has been used. It is noted that the Markov chain was firstly used for steganalysis by Sullivan et al. [7]. There, they scan the whole image horizontally row-by-row and then calculate the empirical transition matrix, which is essentially something similar to the co-occurrence matrix. Since the dimensionality is extremely high (e.g., $256 \times 256 = 65,536$ for an 8-bit gray-level image), not all of elements of the matrix can possibly be used as features. Only some elements are selected. The authors of [7] select several largest probabilities along the main diagonal together with their neighbors, and then randomly select some other probabilities along the main diagonal as features, resulting in a 129-dimensional feature vector. Finally, supporting vector machine (SVM) is adopted in their scheme for classification. This technique, though successful to some extent for the detection of spread spectrum (SS) data hiding, does not perform well for attacking other steganography methods, in particular, for those JPEG steganographic methods. One of reasons is it has abandoned some useful information due to the random fashion of some features' selection.

Inspired by [7], this paper presents a new steganalysis scheme based on the high-dimensional features generated from the co-occurrence matrix. In this scheme, we propose to adopt high-dimensional features, hence using the adequate information of co-occurrence matrix, to capture the changes before and after the data embedding. The class-wise non-principal component analysis (CNPCA) is proposed to solve the classification problem in high-dimensional space. In addition to working on the gray-level co-occurrence matrixes for attacking steganographic methods in spatial domain, we also work on the co-occurrence matrixes associated with JPEG coefficient domain to attacking modern JPEG steganographic techniques, such as OutGuess [4], F5 [5], and MB [6]. Considering the 2-D nature of images, in either case, we consider the vertical, main-diagonal and minor-diagonal directions in addition to the horizontal direction when generating co-occurrence matrixes. Our extensive experiments have demonstrated that the proposed scheme performs better than the existing steganalysis schemes.

The rest of the paper is organized as follows. Section 2 describes the feature extraction from gray-level co-occurrence matrix. The CNPCA classification method

is introduced in Section 3. The scheme to attack JPEG steganography is described in Section 4. The experimental results are given in Section 5 and the paper is concluded in Section 6.

2 Feature Extraction from Gray-Level Co-occurrence Matrix

Gray-level co-occurrence matrix describes the co-occurrence of the various gray-levels at some specified spatial positions in an image. The gray-level co-occurrence matrix of the natural image tends to be diagonally distributed because the gray-levels of the neighbor pixels in natural images are often highly correlated. After the data embedding, however, the high-concentration along the main diagonal of gray-level co-occurrence matrix spreads because the high-correlations between the pixels in the original image have been reduced. This has been shown in [7].

The parameters of gray-level co-occurrence matrix in our scheme are chosen as follows. The gray-levels are 0-255 for 8-bits gray-level images. The gray-level co-occurrence matrix offset parameter d [8] is set to 1, namely, only the nearest neighborhoods are considered in our method. Four different directions are selected for gray-level co-occurrence matrix calculation [8], i.e., θ = 0°, 45°, 90° and 135°, respectively. We thus obtain four gray-level co-occurrence matrixes: G1, G2, G3, G4 from these four different directions, respectively. From these four matrixes, we generated the following resultant co-occurrence matrix, i.e.,

$$G = normal\left(G_1 + G_2 + G_3 + G_4\right) \tag{1}$$

where the operator *normal* represents average and normalization.

	-1	**0**	**1**
-1		1	
0	1		3
1		3	

Fig. 1. The co-occurrence matrix of the 1-D sequence [-1, 0, 1, 0, 1]

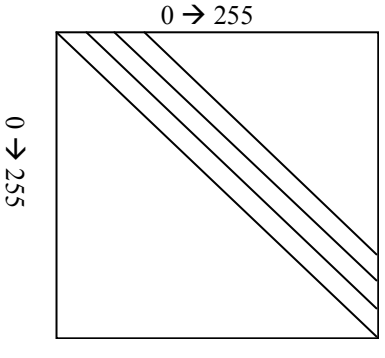


Fig. 2. Gray-level co-occurrence matrix (256×256)

According to [8], the co-occurrence matrix we generated is a symmetric matrix. That is, we not only count the occurrence of *a* with *b*, but also the occurrence of *b* with *a*. As a simple example, consider a 1-D signal [-1, 0, 1, 0, 1]. Its co-occurrence matrix is symmetric as shown in Fig. 1.

Considering the symmetry of the co-occurrence matrix, we adopt the elements of the main diagonal and a part of the upper triangle of the matrix, as shown in Fig. 2, to construct the feature vector in our proposed scheme. In our experimental work, as shown in Fig. 2, we use 1018-dmensional (256×4-6=1018□feature vectors. Statistically, the energy of the selected elements is about 50-70% of the whole upper triangle of the gray-level co-occurrence matrix. The selected feature vector, therefore, keeps most information of the gray-level co-occurrence matrix and is expected to be able to capture the changes caused by the data embedding process.

Let G_{ori} denote the gray-level co-occurrence matrix of the original cover image and G_{steg} the gray-level co-occurrence matrix of the stego image. Thus, $(G_{ori}-G_{steg})^2$ describes the energy differences between them, which is shown in Fig. 3. It is observed from Fig. 3 that the energy difference concentrates around the main diagonal of gray-level co-occurrence matrix. This observation together with the symmetry of co-occurrence matrix justifies our feature selection of the elements of gray-co-occurrence matrix as shown in Fig. 2.

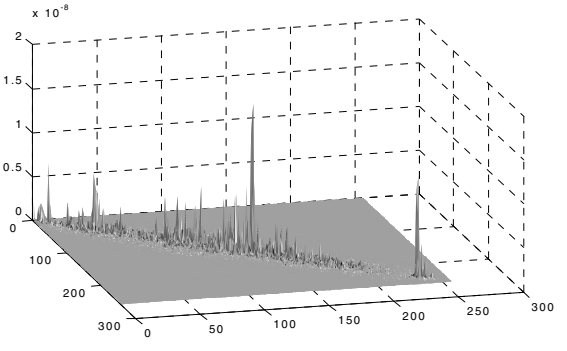


Fig. 3. The distribution of energy difference (refer to text)

If we adopt the Euclidean distance based Bayes classifier to classify the 1018-dimensional feature vectors, it would have been very hard to calculate the inverse covariance matrix because of the high dimensionality. To solve this problem, we propose the class-wise non-principal components analysis method.

3 Class-Wise Non-Principal Component Analysis (CNPCA)

The class-wise non-principal component analysis (referred to as CNPCA for short) [9] is to classify the samples based on the distances between the samples and the mean vectors of each class in the space spanned by the eigenvectors associated with the smallest eigenvalues of each class.

3.1 Definitions

Let $\boldsymbol{x}$ denote the n-dimensional random vectors in the k^{th} class, and assume that there are in total K different classes. When the eigenvalues of the covariance matrix generated from all of $\boldsymbol{x}$ are ranked from the largest to the smallest in an non-increasing order, the corresponding eigenvector matrix can be expressed as:

$$\Phi_k = \left(\Phi_k\right)_{n\times n} = \left[\Phi_{rk}, \Psi_{rk}\right]_{n\times n} \tag{2}$$

where the n is the dimensionality; the r, $(r \le n)$, is the number of eigenvectors associated with the largest eigenvalues; the $(n\text{-}r)$ is the number of eigenvectors associated with the smallest eigenvalues; the $\boldsymbol{\Phi_k} = \left(\Phi_k\right)_{n\times n}$ is the eigenvector matrix with all eigenvectors of the k^{th} class; the $\boldsymbol{\Phi_{rk}} = \left(\Phi_{rk}\right)_{n\times r}$ is the principal components matrix with all the r eigenvectors of the k^{th} class; the $\Psi_{rk} = \left(\Psi_{rk}\right)_{n\times(n-r)}$ is the non-principal components matrix with all, $(n\text{-}r)$, remaining eigenvectors of the k^{th} class; the k^{th} class' non-principal components $\boldsymbol{\Psi_{rk}}$ and principal components $\boldsymbol{\Phi_{rk}}$ are complementary to each other.

In CNPCA classification, given a test sample vector $\boldsymbol{y}$, its Euclidean distance to the mean vector of the k^{th} class in the subspace spanned by the $(n\text{-}r)$ class non-principal components is adopted as the classification criterion, referred to as CNPCA distance. The CNPCA distance of the vector $\boldsymbol{y}$ to the k^{th} class is defined as:

$$D_{rk} = \left\| \Psi_{rk}^{t}\left(y - M_k\right)\right\| \tag{3}$$

where D_{rk} stands for the Euclidean distances between the sample $\boldsymbol{y}$ and the mean of the k^{th} class, M_k, in the $(n\text{-}r)$ dimensional CNPCA space, $\boldsymbol{D_{rk}}$ can be represented by the class-wise non-principal components matrix $\boldsymbol{\Psi_{rk}}$. Obviously, there are two special cases. When $r=0$, CNPCA distance becomes the conventional Euclidean distance while when $r=n$, CNPCA distance equals to 0. Hence the case of $r>0$ and $r<n$ is usually used in CNPCA.

In summary, during the CNPCA classification, a given test sample $\boldsymbol{y}$ is firstly mapped into the $(n\text{-}r)$ non-principal components subspace of each class. The distances in these subspaces between $\boldsymbol{y}$ and the mean of each class are then calculated. Finally the $\boldsymbol{y}$ is classified to the k^{th} to which the CNPCA distance is the minimum, i.e.,

$$\hat{k} = \arg\min_{k}\left\{\ D_{rk}\ \right\}.$$

The number of r is an important parameter for CNPCA. It can be estimated by minimizing the classification error rate $\boldsymbol{\varepsilon}$:

$$\hat{r} = \arg\min_{r}\left\{\ \varepsilon\left(D_{rk}\right)\ \right\}$$

3.2 Classification Procedure

Step 1: We first apply the K-L transform to the training samples of each class. The $(n\text{-}r)$ eigenvectors associated with the $(n\text{-}r)$ smallest eigenvalues are selected as the

dimension reduction matrix, Ψ_{rk} , for each class. The mean vector of each class, $\boldsymbol{M_k}$, is also calculated in this step.

Step 2: For testing sample $\boldsymbol{y}$, the CNPCA distances between the sample and each class, $\boldsymbol{D_{rk}}$, are calculated according to the following formula,.

$$D_{rk} = \left\| \Psi_{rk}^{t} \left(y - M_k \right) \right\| = \left(y - M_k \right) \Psi_{rk} \Psi_{rk}^{t} \left(y - M_k \right) \tag{4}$$

Step 3: The testing sample $\boldsymbol{y}$ is classified to the class to which the $\boldsymbol{y}$ has the minimum CNPCA distance $\boldsymbol{D_{rk}}$. In other words, the classification decision is made by:

$$\hat{k} = \arg\min_{k} \left\{ D_{rk} \right\} \tag{5}$$

3.3 CNPCA vs. PCA

The concept of CNPCA classification is quite different from that of the conventional PCA classification. While the CNPCA method utilizes the within-class information in each class effectively, the PCA (Principal Component Analysis) is a dimension reduction method for the whole set which averages the within-class distribution of each class. When the samples scatter within each class and cluster between classes, the PCA does not perform well. On the contrary, CNPCA describes the minimum variance directions for each class. It is very suitable to solve the problem of the scattering within each class and clustering between classes. Actually, image steganalysis is a typical two-class ("stego-image" and "non-stego-image") classification problem in which the samples scatter within class and cluster between classes. The content of the image database is very diverse. The samples, therefore, scatter within each class. On the other hand, because the embedding process must be invisible, the data embedding strength has to be small enough, which makes the samples cluster between classes. The CNPCA removes the principal components while keeps the non-principal components. The main purpose of doing so is to select features which are sensitive to "embedding" instead of "the image content itself" from the high dimensional space of gray-level co-occurrence matrix. For more detail, readers are referred to [9].

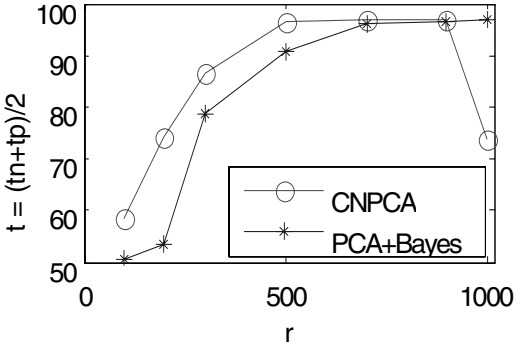


Fig. 4. Comparison of CNPCA and PCA+Bayes classifier (for image database refer to Section 5.1), embedding method is LSB with 0.1bpp), tn: true negative rate, tp: true positive rate

To verify this idea, we compare the performance of the proposed CNPCA classifier, and the PCA dimension reduction followed by a Bayes classifier in Figure 4.

4 Attacking JPEG Steganography

Since the JPEG (Joint Photographic Experts Group) format is the most dominant image format for image storage and exchange these days, the JPEG steganographic techniques have attracted more and more attentions. In JPEG steganographic techniques, secret message are embedded by modifying the quantized block-DCT coefficients of the cover image. Steganalyzing the JPEG steganography directly in JPEG block-DCT coefficient domain is more effective than doing it in image pixel domain. Therefore, we calculate the co-occurrence matrix in the block-DCT domain when attacking JPEG steganography.

The procedure of the feature extraction is as follows:

(1) Read in the quantized block-DCT coefficients from a given JPEG file.
(2) Expand the block-DCT coefficients of each 8×8 block into a 1-D vector $V_i(0,1,2,\ldots,63)$ in the zig-zag order [10], where i is the block index.
(3) Only keep the low frequency part of the 1-D vector, i.e., $V_i(1,2,\ldots,20)$. We do it in this way because most of the high frequency coefficients are quantized to zero (thus not much information will be lost), and few modern steganographic methods touch DCT DC coefficients. Since the magnitude of the block-DCT coefficients has a quite large dynamic range, we further clip the values of V_i to the range of $[-T, T]$, where T is a predefined threshold. Properly setting a threshold will not lose much information because the block DCT coefficients follow the generalized Laplacian distribution which has a very large peak around zero, and most of DCT AC coefficients are small. As an example, we have calculated the percentage of the block DCT AC coefficients which are below a given threshold T for the popularly used Lena image with Q-factor 80 in JPEG compression, and found that for $T=7$, 96.59% of block DCT AC coefficients fall into the interval of $[-T, T]$. Therefore, using an appropriate threshold only lose trivial information while reduce the computational complexity dramatically.
(4) Calculate the co-occurrence matrix G_i for each 1-D vector V_i.
(5) Calculate the global average co-occurrence matrix by using $G=\frac{1}{N}(\sum_{i=1}^{n}G_i)$, where N is the total number of the blocks in a JPEG image. We use the whole upper triangle of G as feature for steganalysis. Finally, CNPCA is employed as classifier for classification.

5 Experiments in Steganalysis

5.1 Attacking Steganography in Spatial Domain

We generate a hybrid image database composed of totally 3908 images, in which 1096 images are from CorelDraw [11] and the other 2812 images are from the UCSB web site [12]. In the experiments, we randomly select half of the images (1954

images） as the training samples and the other half as the testing samples. Three embedding methods: Cox et al.'s spread spectrum (SS), QIM, and LSB are used in the experiments and the embedding rate are set to be 0.3 bpp (bits per pixel), 0.1 bpp and 0.02 bpp, respectively. The experimental results are shown in Table 1 and Fig. 5.

Table 1 illustrates the performance comparison between the proposed scheme (from *r=500* to *r=900*) and the steganalysis methods proposed by Farid [1] and Sullivan et al. [2]. The detection rates shown in Table 1 are actually the average results of 10 times tests（the training and testing samples are randomly selected each time). As can be seen in Table 1, our proposed method outperforms Farid's and Sullivan's method for all the embedding methods (except QIM) at all the embedding rates. This superiority is obvious especially for the low embedding rate LSB0. (1 and 0.02 bpp).

Table 1. Detection accuracy comparison (tn: true negative; tp: true positive; t=(tn+tp)/2)

	Farid [1]			Sullivan et al. [7]			Proposed		
	tn	tp	t	tn	tp	t	tn	tp	t
SS	23	89	56	86	64	75	76	82	79
Qim	66	92	79	91	90	91	78	97	87
LSB(0.3bpp)	37	91	64	56	74	65	99	99	99
LSB(0.1bpp)	23	89	56	45	62	53	96	98	97
LSB(0.02bpp)	8	92	50	39	57	48	73	79	76

5.2 Selection of Dimensionality *r*

According to [13], in our proposed method, the detection rate *t* is defined as the arithmetic average of true positive rate and true negative rate. It is also referred to as

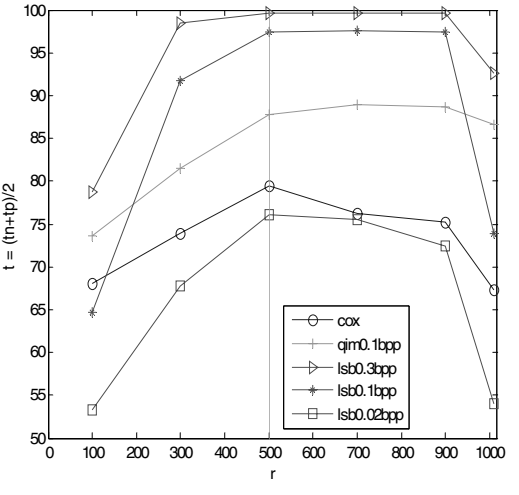


Fig. 5. Average detection rate as a function of the non-principal components dimensionality *n-r*

accuracy. It is a function of the non-principal components dimensionality (n-r). Generally speaking, it can keep high detection rate in a relatively wide range around the peak value. As shown in Fig. 5, the detection rates achieve their peak values when r is around 500 (at this point, the dimensionality of the non-principal components is 1018-500=518). The peak values keep almost constant till r=*900* (at this point, the dimensionality of the non-principal components is 1018-900=118).

5.3 Stability Study

To verify the stability of the detection rate, we repeat the test 10 times in this experiment. Each time, we randomly select half of the 3908 images as the training set and the other half as the testing set. The results (r is set to 500) are recorded and shown in Figure 6.

As we can see in Figure 6, the detection rates are quit stable for different training and testing sets.

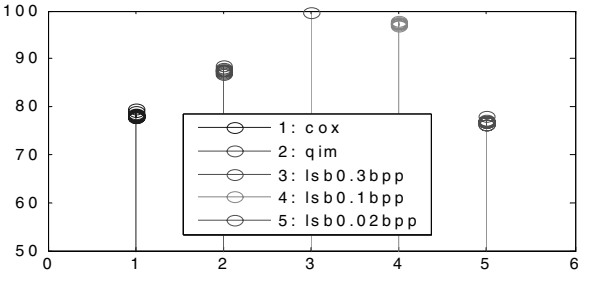


Fig. 6. Performance of the stability experiments (The numbers in horizontal axis stand for different embedding scheme and the vertical axis denotes the detection rate. Tests on each embedding method are repeated 10 times).

5.4 Attacking JPEG Steganography

In JPEG steganalysis, to avoid the influence of JPEG double compression on steganalysis performance, we use the 1096 uncompressed images from the CorelDraw database [11]. We firstly JPEG [14] compress all the images using Quality-factor 75 as the cover images. All the images are also embedded by separately using OutGuess [15], F5 [16], MB1 [17] (MB without deblocking operation) and MB2 (MB with deblocking operation) with the amount of embedded message as 1kB (i.e., 1024 bytes), 2kB and 4kB, respectively. The images have been cut to keep the central portion having a size 768x512 or 512x768. The central cut was conducted in the JPEG coefficient domain in order not to involve additional JPEG compression.

In the classification process, we randomly selected 896 (about 5/6 of) original images and the corresponding 896 stego-images for training and the remaining 200 pairs (about 1/6) of stego images and non-stego images for testing. For comparison purpose, we have also implemented the steganalysis schemes proposed by Farid [1], Xuan et al. [2] and Fridrich [3]. Then we apply them to the same set of images and with the same steganographic methods. The same training and testing procedures are used. All the

results are listed in Table 2. The experimental results reported here are the averages of the 10 times of random tests. The r parameter in CNPCA classification is selected as follows. For F5, the detection rate peak value appears at $r = \{3, 4, 5, 6, 7\}$. We select $r = 5$ in Table 2. For OutGuess, MB1 and MB2, the detection rate peak value appears at $r = \{10, 11, 12, 13, 14, 15]$. We select $r = 12$ in Table 2.

As can be seen in Table 2, the proposed scheme outperforms all the other steganalysis schemes in detecting these four JPEG steganographic techniques at all of these three different data embedding rate. The exceptions are in detecting F5 at embedding rates of 1kB and 2kB, i.e., the detection rates achieved by our proposed method is 1% and 2%, respectively, less than that achieved by [3]. In general, the detection rates achieved by the proposed scheme are comparable to or higher than that by Fridrich's method [3] while outperforms that by Farid's [1] and Xuan et al.'s [2] by a significant margin. As a whole, therefore, the proposed scheme outperforms the existing steganalysis methods.

Table 2. Performance comparison in JPEG steganalysis

		Farid			Xuan et al.		
		tn	tp	t	tn	tp	t
F5	1kB	51	54	53	62	58	60
	2kB	56	56	56	64	65	65
	4kB	68	53	60	85	77	81
Outguess	1kB	58	38	48	61	41	51
	2kB	61	39	50	67	61	64
	4kB	59	48	54	73	79	76
MB1	1kB	48	55	52	59	60	59
	2kB	52	53	53	71	69	70
	4kB	53	58	55	83	81	82
MB2	1kB	55	47	51	65	55	60
	2kB	49	58	53	75	64	69
	4kB	59	52	55	85	83	84
		Fridrich			Our proposed		
		tn	tp	t	tn	tp	t
F5	1kB	76	72	74	78	68	73
	2kB	86	87	87	84	85	85
	4kB	94	98	96	97	98	98
Outguess	1kB	91	87	89	98	98	98
	2kB	97	96	97	100	100	100
	4kB	98	97	98	100	100	100
MB1	1kB	66	65	66	90	84	87
	2kB	88	83	86	98	88	93
	4kB	91	88	90	99	99	99
MB2	1kB	64	61	62	89	82	86
	2kB	76	77	76	98	92	95
	4kB	88	80	84	100	99	99

5.5 Computational Complexity

Computational complexity is important to estimate a system's potential for real-time application. In this section, time cost of different steganalysis schemes is used as a main criterion for measuring computational complexity. In this experiment, we only test the time spent for feature extraction, which consumes most of the time, and classifier's parameters can be trained before actually use. Randomly select 100 JPEG images form the database described in 5.4 for testing the time cost, the result is show in Table 3. The experimental environment is: Intel Celeron(R) CPU 1.70GHz, memory 256MB, and Matlab 7.1 version.

Table 3. Time cost of features extraction

Steganalysis	Featrues dimention	Seconds of 100 images	Seconds per image
Farid[1]	72	657.85	6.58
Xuan[2]	39	406.46	4.06
Fridrich[3]	23	1159.20	11.59
Sullivan[7]	130	130.51	1.30
Proposed	120	130.30	1.30

As we can see from Table 3 that though more dimensions are used in our proposed scheme, the time cost is the least, while the scheme proposed in [3] has highest computational complexity, even though it uses only 23 features. From these experiments, one can observe that the proposed steganalysis method is effective both in high detection rate and low time cost, which lends itself for potential of real-time usage in practice.

6 Conclusions

1) We have proposed to use the high dimensional features generated from co-occurrence matrix in image pixel domain and in JPEG coefficient domain to capture the changes occurring to images before and after the data embedding.
2) Class-wise non-principal component analysis (CNPCA) is proposed to be utilized as classifier for steganalysis. The CNPCA classification overcomes the problems where the inverse of covariance matrix does not exist in pattern classification. It demonstrates good performance in tackling the problem caused by high dimensionality of feature vectors and/or the problem where between-class feature vectors cluster while within-class vectors scatter.
3) The experimental works have demonstrated improved performance in steganalysis, compared with the existing steganalyzers.

References

1. Farid H.: Detecting hidden messages using higher-order statistical models. Proceeding of the IEEE International Conference on Image Processing, Vol. II, New York, (2002) 905 - 908
2. Xuan, G., Shi, Y.Q., Gao, J., Zou, D., Yang, C., Zhang, Z., Chai, P., Chen, C., Chen, W.: Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions. Information Hiding Workshop, Barcelona, Spain, June (2005) 262 – 277

3. Fridrich, J.: Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. 6th Information Hiding Workshop, Toronto, Canada (2004)
4. Provos, N.: Defending against statistical steganalysis. 10th USENIX Security Symposium, Washington DC, USA (2001)
5. Westfeld, A.: F5 a steganographic algorithm: High capacity despite better steganalysis. 4th International Workshop on Information Hiding, Pittsburgh, PA, USA (2001)
6. Sallee, P.: Model-based methods for steganography and steganalysis. International Journal of Image and Graphics, 5(1) (2005) 167-190
7. Sullivan, K., Madhow, U., Chandrasekaran S., Manjunath, B.S.: Steganalysis of spread spectrum data hiding exploiting cover memory. SPIE 2005, vol. 5681, (2005) 38 - 46
8. Haralick, R.M.: Textural features for image classification. IEEE Trans. Systems Man Cybernetics. SMC-3 (1973)
9. Xuan, G., Chai, P., Zhu, X., Yao, Q., Huang, C., Shi, Y.Q., Fu, D.: A novel pattern classification scheme: Classwise non-principal component analysis (CNPCA). International Conference on Pattern Recognition (ICPR), Hong Kong, August (2006)
10. Shi, Y.Q., Sun, H.: Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards. CRC Press, Boca Raton, FL. (1999)
11. http://www.corel.com
12. http://vision.ece.ucsb.edu/~Sullivan/Research_imgs/
13. Fawcett, T.:"ROC Graphs: Notes and Practical Considerations for Researchers", Tech Report HPL-2003-4, HP Laboratories. (2003)
(http:// home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf)
14. http://www.ijg.org/
15. http://www.outguess.org/
16. http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html
17. http://redwood.ucdavis.edu/phil/papers/iwdw03.htm

Multi Bit Plane Image Steganography

Bui Cong Nguyen, Sang Moon Yoon, and Heung-Kyu Lee

Department of EECS, Korea Advanced Institute of Science and Technology,
Guseong-dong, Yuseong-gu, Daejeon, Republic of Korea
{nguyenbc, pisces}@mmc.kaist.ac.kr

Abstract. This paper addresses a novel steganography method for images. Most statistical steganalysis algorithms are strong to defeat previous steganography algorithms. RS steganalysis and pixel difference histogram analysis are two well-known statistical steganalysis algorithms which detect non-random changes caused by embedding a secret message into cover image. In this paper, we first explain how two steganalysis algorithms exploit the effect of the non-random changes and then propose a new steganography method that avoids the non-random changes to evade statistical analysis methods. For this purpose, we adjust the embedding process to be more adaptive to cover image by considering embedding in Gray code bit planes, not natural binary bit planes, of cover images, and two parameters: (1) similarity threshold for selecting non-flat area in lower bit planes, and (2) size of flat blocks $n \times n$ in embedding bit planes. Experimental results show that the secret messages embedded by our method are undetectable under RS steganalysis and pixel difference histogram analysis.

1 Introduction

Nowadays, the digital communication becomes widespread through many environments such as the Internet and mobile networks, etc. Many types of media: text, audio, and video are transferred. Cryptography has been the most reliable mechanism to make a secure communication between two parties for many decades. However, the surveillance technology in the digital world may help the third party to identify, capture, and disrupt the encrypted traffic. Steganography in Greek means covert writing. It even hides the existence of the communication by sending to the other party the look-like innocent cover which has an embedded message. Therefore, steganography provides us a secret communication in open environment like Internet.

At the beginning phase of applying steganography technique for secure digital communication, steganography researchers have developed many embedding technologies for various types of media. They mainly concentrated on embedding capacity, not on the secure under attacks. Later, the steganalysis researchers proposed a large number of analysis algorithms to detect the existence of steganography work in digital media.

Steganography researchers used to believe that embedding a message into only the least significant bit (LSB) of the cover is secure enough since embedding

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 61–70, 2006.

process makes no changes or only small changes in each pixel value. However, steganalysis researchers gradually found that some characteristics of the cover are changed even if we embed a small message to the LSB of the cover. Westfeld and Pfitzmann [1] designed a technique to successfully identify sequential LSB embedding method, which based on pair-of-value distributions called χ^2 - statistical analysis. Provos [2] extended this method by re-sampling test intervals and re-pairing values. Fridrich, Goljan and Du [3] proposed the RS steganalysis method to detect LSB steganography in gray (8-bit) and color (24-bit) images. Dumitrescu, Wu and Wang [4] presented a similar technique to the RS steganalysis technique for the LSB-based steganalysis by analyzing pixel subsets whose characteristics are changed by message embedding. Their methods work reliably even for relatively small messages randomly scattered in a digital image.

In this paper, we introduce an embedding method that can evade the RS steganalysis and histogram analysis by embedding a message in multi CGC bit planes of the cover image. The remaining parts of this paper are organized as follows. Section 2 summarizes the existing steganography techniques like the least significant bit (LSB) embedding and pixel-value differencing (PVD), and statistical steganalysis techniques like RS steganalysis and histogram analysis. In Section 3, we present our method for escaping the detection of the steganalysis methods. Our experimental results are in Section 4. Finally, the conclusions and discussion about future work of our research are in the last section.

2 Scenario of Steganography and Steganalysis

2.1 RS Steganalysis

As we mentioned above, the RS steganalysis is an effective method to detect several steganography techniques. Recently, there are some papers[5],[6] derived from it with a little improvement on speed of calculation or exactness, however, the main idea is not changed much. At first, the RS steganalysis method divides the received image into same small sized blocks. The method defines two functions: F_1, which changes a pixel value $0 \leftrightarrow 1$, $2 \leftrightarrow 3$, $4 \leftrightarrow 5$, ...,$254 \leftrightarrow 255$ and F_{-1}, which changes a pixel value $-1 \leftrightarrow 0$, $1 \leftrightarrow 2$, $3 \leftrightarrow 4$, ...,$255 \leftrightarrow 256$. R_M is the ratio of the blocks in which the total of fluctuations increases when F_1 is applied to the blocks with mask M. S_M is the ratio of blocks in which the total of fluctuations decreases when F_1 is applied to the blocks with mask M. Also, R_{-M} and S_{-M} are defined with F_{-1} instead of F_1. Fridrich found that the RS ratio of a typical image should satisfy the rule: $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ through large amount of experiments. When only LSB of the original cover is changed, the difference between R_M and R_{-M} and the difference between S_M and S_{-M} increase. Then, the rule is violated, therefore, one could conclude that the tested image has a hidden message.

We explain the weakness of LSB embedding method by looking at the pixel values of the cover under the LSB embedding process. In the process, the replacement of the LSB in a pixel by a message bit (0 or 1) changes the old pixel

value p_c to the new value p_s. There are 3 cases to change from p_c to p_s like below:

$$p_s = \begin{cases} p_c \; if \; LSB \; of \; p_c = message \; bit \\ p_c + 1 \; if (p_c \bmod 2) = 0 \; and \; LSB \; of \; p_c \neq message \; bit \\ p_c - 1 \; if (p_c \bmod 2) = 1 \; and \; LSB \; of \; p_c \neq message \; bit \end{cases} \quad (1)$$

However, when we randomly change a pixel value with three values $[-1, 0, 1]$, there are 5 cases like below:

$$p_s = \begin{cases} p_c + 0 = p_c \\ p_c \pm 1 \; if (p_c \bmod 2) = 0 \\ p_c \pm 1 \; if (p_c \bmod 2) = 1 \end{cases} \quad (2)$$

Hence, in LSB embedding method, the number of change cases of a pixel value with the range [-1, 0, 1] is three. It is smaller than the number of change cases (5 cases) in random change. For example, in LSB embedding method, there is change case from p_c to p_s as $0 \leftrightarrow 1$ but there is no change case as $1 \leftrightarrow 2$. In Dumitrescu, Wu and Wang's paper [4], the authors consider this characteristic as the relationship between LSB and the remain bit planes. The attacker can exploit this characteristic to detect the existence of the message. To make the random change in a pixel value, we should embed not only in LSB but also in higher bit planes in a random way.

2.2 Pixel Difference Histogram Analysis

The method PVD [7] uses differences of pair neighbor pixels to hide message words. At first, cover image is divided to non-overlapping blocks of two neighboring pixels. The scanning cover pixels for partitioning the cover image is a zigzag line through all row or column. A difference between two neighbor pixels of a block is calculated: $d = p_{i+1} - p_i$ which p_i and p_{i+1} are the two pixel values. We have $|d| \in [0, 255]$. Classify $|d|$ into a number of contiguous ranges, $R_k (k = 0, 1, ..., K-1)$ where the width R_k is a power of 2. A practical set of R_k is [0 7], [8 15], [16 31], [32 63], [64 127], [128 255]. The author denoted l_k, u_k, w_k as the lower bound, upper bound, and width of R_k respectively. If $|d|$ is in R_k, a message word with $\log_2(w_k)$ bits is embedded in the corresponding two-pixel block. A $\log_2(w_k)$ message word has a decimal value b. We can calculate b as below:

$$d' = \begin{cases} l_k + b \; if \; d \geq 0 \\ -(l_k + b) \; if \; d < 0 \end{cases} \quad (3)$$

The new difference $|d'|$ will be in the same range R_k with $|d|$. The embedding procedure is described with the formula below:

$$(p'_i, p'_{i+1}) = f[(p_i, p_{i+1}), d']$$

$$= \begin{cases} (p_i - r_c, p_{i+1} + r_f), \text{ if d is odd} \\ (p_i - r_f, p_{i+1} + r_c), \text{ if d is even} \end{cases} \quad (4)$$

where

$$r_c = \left\lceil \frac{d' - d}{2} \right\rceil,\ r_f = \left\lfloor \frac{d' - d}{2} \right\rfloor \tag{5}$$

In the actual embedding process, a block is labeled as *unusable* and excluded if the embedding process makes the value of p_i or p_{i+1} out of the range [0 255].If the portion of *unusable* block is large, the embedding capacity will be reduced significantly.

In this method, the changes in pixel values become more random because the message words will change several bits in a pixel. Hence, it can evade the RS steganalysis. However, X. Zhang and S. Wang [8] found the unusual steps in the histogram of pixel differences. Based on that problem, they declared the presence of secret messages. They also introduced a way to make the changes of the pixel value more random, therefore, the method can avoid the step effect. However, their method still has unusable blocks, so, the embedding capacity is reduced. They also embed message even in smooth regions of cover image where pixel values are not fluctuated.

3 Proposed Steganography Method

3.1 Related Factors

As we mentioned in the previous section, our method is designed to avoid the detection of several steganalysis methods. The first goal of the embedding method is to avoid the human visual system analysis. The second goal is to avoid the non-random changes of pixel values. Therefore, we use several factors to adjust the embedding process when we embed message in non-noisy area of bit planes. For this purpose, we embed in Canonical Gray Coding (CGC) (also named as Gray code) not natural binary bit planes, of the cover image, and use two parameters: (1) size $n \times n$ of similarity blocks, (2) threshold t for selecting flat areas in each bit plane.

We use multi bit planes in CGC code instead of PBC code for embedding, which results in more random change in pixel value. The formula to transfer from a N-bit pixel value in PBC to CGC is given below:

N-bit pixel value in PBC system: $b_N b_{N-1} b_{N-2} \dots b_i \dots b_1$ (b_1 is LSB), $b_i = (0, 1)$. is converted to N-bit pixel value in CGC system: $g_N g_{N-1} g_{N-2} \dots g_i \dots g_1$ as follows:

$$\begin{cases} b_N = g_N \\ g_i = b_i \oplus b_{i+1}\ 1 \le i \le N-1 \end{cases} \tag{6}$$

For transfer back from CGC code to PBC code, we use:

$$\begin{cases} b_N = g_N \\ b_i = g_i \oplus b_{i+1}\ 1 \le i \le N-1 \end{cases} \tag{7}$$

Changing bit b_i of a pixel in PBC system causes a step $\pm 2^{i-1}$ change in its value. Changing bit b_i of a pixel in CGC system results in a change of its value in the range: $[1, 2^i - 1]$. Therefore, if we change several bits of a CGC pixel , the change of the pixel value is scattered in a non-step range unlike PBC case.

First, the image is decomposed into N CGC bit planes $B_N B_{N-1} B_{N-2} \dots B_i \dots B_1$ (B_1 is LSB bit planes). Based on the required security, we choose the number of bit planes (BP) to embed. With a gray 8-bit image, modification in bit plane number five or higher, the degradation of the image will be detectable by human visual system. For actual experiments, we choose the number of bit planes (BP) is four or three. We embed messages in the cover with the descending order of bit plane (from BP to B_1).

In smooth regions of cover image, pixels have similar values. If we embed a message in these regions, we may add noise to non-noisy (flat) areas and steganalysis algorithms can detect abnormal noise in these regions. Accordingly, we find the smooth regions and mark them in order to avoid modifying them in the embedding process. The smooth regions are obtained by combining smaller smooth areas (flat areas) which size $n \times n$ where $1 \leq n \leq (height\ or\ width\ of\ image)$.

The embedding process for bit plane number i (B_i) with $1 \leq i \leq BP$ is as follow. Before embedding message in bit plane B_i, we first find the flat areas in bit plane B_i. We scan all image pixels with a window size $n \times n$. In this window, we calculate the differences of the top left pixel (pivot) with the remaining pixels. If all the absolute differences are smaller than threshold t, the area size $n \times n$ in bit plane B_i corresponding to the current window position is flat. In practical experiment, we choose the threshold $t = 2^i$ or 2^{i+1}. Thus, we can easily calculate the differences of the pivot with the remain pixels by shifting all four pixels p_1, p_2, p_3, p_4 i bit to the right. Then the new threshold t' will be 0 or 1 corresponding to 2^i or 2^{i+1} in old t. For instance, window size $= 2 \times 2$ with four pixels p_1, p_2, p_3, p_4 as $\begin{bmatrix} p_1\ p_2 \\ p_3\ p_4 \end{bmatrix}$. We get p'_1, p'_2, p'_3, p'_4 when we shift all four pixels p_1, p_2, p_3, p_4 to right i bit. The differences of the pivot with the remaining pixels in new window value will be $\begin{bmatrix} p'_1 - p'_1\ p'_1 - p'_2 \\ p'_1 - p'_3\ p'_1 - p'_4 \end{bmatrix}$. The area size $n \times n$ in bit plane B_i corresponding to the current window position is flat if all of following conditions: $|p'_1 - p'_1| \leq t'$, $|p'_1 - p'_2| \leq t'$, $|p'_1 - p'_3| \leq t'$, and $|p'_1 - p'_4| \leq t'$ are satisfied.

The secret message is embedded in the non-flat areas of bit plane B_i with a pseudo random sequence. After embedding message in bit plane B_i, we find the flat areas for bit plane B_{i-1} and embed remaining message in its noisy areas. This process is repeated until we finish the work in the LSB bit plane.

We make the pseudo-random sequences dependent on secret key for shuffling the index of non-flat areas in the bit planes where we embed. Hence, the message is scattered to the non-flat areas of the bit plane. There is no need for header data for locating the embedded areas because we can calculate the non-flat areas of a bit plane from higher bit planes then extract the hidden message with the given secret key. We show the embedding procedure and extracting procedure below.

3.2 Embedding Procedure

The embedding procedure for sender side:

1. Transform the cover image from PBC to CGC system.
2. Decompose the Image into N-bit planes by bit- slicing operation. Bit plane 1 is LSB bit plane.
3. Compress and encrypt the message if needed.
4. Do embed the message in bit planes from highest BP to 1:
   Embedding in bit plane k ($1 \leq k \leq BP$):
   - Find all $n \times n$ flat areas in bit plane B_i with threshold t.
   - Make pseudo-random sequence according to the secret key for every pixel in the non-flat areas.
   - Embed message bits into the non-flat areas of the bit plane k with the pseudo-random sequence.
   - Repeat with bit plane $k-1$ until finish the step $k = 1$ (LSB).
5. Transform the cover image from CGC to PBC system.

3.3 Extracting Procedure

The extracting procedure for the receiver side:

1. Transform the cover image from PBC to CGC system. Decompose the Image into N-bit planes by bit- slicing operation.
2. Scan the bit planes to find all flat areas size $n \times n$.
3. Generate the pseudo-random sequence for non-flat areas in each bit plane dependent on the secret key (password).
4. Extract the message according to the pseudo-random sequence in highest bit plane to lowest bit plane. In the embedding process, when we embed in the higher bit plane, the flat areas for the lower bit plane are changed. Thus, we have to extract from the higher bit plane to lower bit plane because the extraction from lower bit plane depend on the higher bit plane.
5. Uncompress and decrypt the message if needed.

4 Experimental Results

In this section, we show the experimental results of our embedding method in 8 color bitmap images. We tested with $n \times n$ size $= 2 \times 2$, 2×3, 3×3, 3×4, 4×4. Several values of threshold $t' = 0, 1, 2, 3$ are tested. We chose the practical cases for our test: $n \times n$ size $=2 \times 2$ and $t' = 0$ or 1, $BP = 4$. We calculated the flat areas for bit plane B_2 and set the flat areas for bit plane B_1 similar to bit plane B_2. Figure 1 shows the images in the experiment.

In our experiment, we applied some cryptography techniques. The Matrix Encoding [9] method makes the number of modified bits smaller when we embed a small message in the cover. We use the pseudo-random sequence generation method ISAAC (Indirection, Shift, Accumulate, Add, and Count, by R. Jenkins Jr.) for scattering the message according to a given key. ISAAC is a fast cryptographic random number and the generated values are uniformly distributed, unbiased, and unpredictable if you do not know the key (the seed).

Fig. 1. Collection of tested images

4.1 Experiment with RS Steganalysis

The output images of our embedding method and LSB embedding method are tested under the RS steganalysis. The figure 2 shows the RS steganalysis results for embedded cover using the LSB embedding method. In this figure, we can observe that LSB embedding method makes R_M and R_{-M}, as well as S_M and S_{-M}, separated significantly when the embedding capacity is higher than 1%. The figure 3 sketches the RS steganalysis results for embedded cover using our method. In this case, R_M, R_{-M}, S_M, and S_{-M} are almost unchanged. The rule $R_M \cong R_{-M}$and $S_M \cong S_{-M}$ is kept in our output image. Therefore, LSB embedding method is detectable under RS steganalysis but our method is secure under this analysis.

4.2 Experiment with Pixel Difference Histogram

We also tested our embedded images with pixel difference histogram analysis. In this case, we used baboon.bmp image with size 256×256 for the test. In the figure 4b and figure 4c, the PVD embedded image has the step effect in pixel difference histogram. In our case, the histogram, in figure 4d, looks smooth and similar to the histogram of the original cover. Therefore, our method is secure under the pixel difference histogram analysis. Our method has better results than PVD method under this analysis because we make the changes in pixel values more random. In addition, our method does not have unusable blocks as PVD method and we avoid to embed message in smooth areas. In the safe range for embedding, our method has higher PSNR than PVD method with the same embedding capacity. For example, embedding rate is equal to 14.5% of the size of Baboon image, PVD method output has $PSNR = 32.17dB$ and our method output has $PSNR = 37.73dB$.

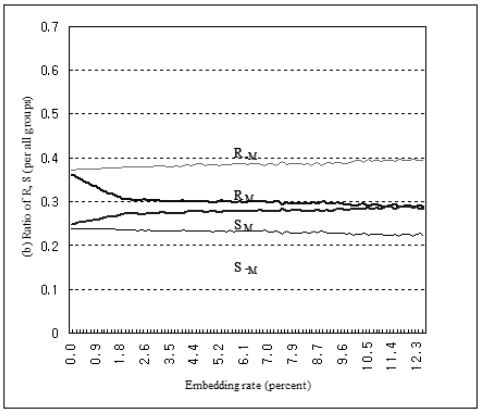


Fig. 2. LSB embedding method under RS steganalysis

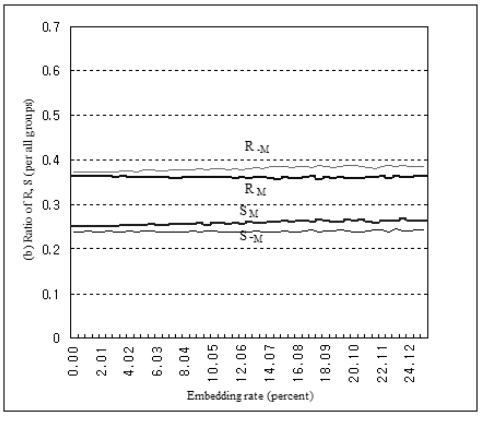


Fig. 3. Our embedding method under RS steganalysis

5 Conclusion

We built a steganography algorithm based on embedding into non-flat areas of multi CGC bit planes of the cover image. We have tested our embedding method under two analysis techniques RS steganalysis algorithm and pixel deference histogram analysis. The experimental results show that our embedded images is secure under the two steganalysis techniques. In the future work, we will have more tests under the others steganalysis techniques in order to show that our method is secure under many steganalysis techniques. We will also extend our algorithm to the other image types such as embedding in transform domain image with some modification to make it adaptive to those image types.

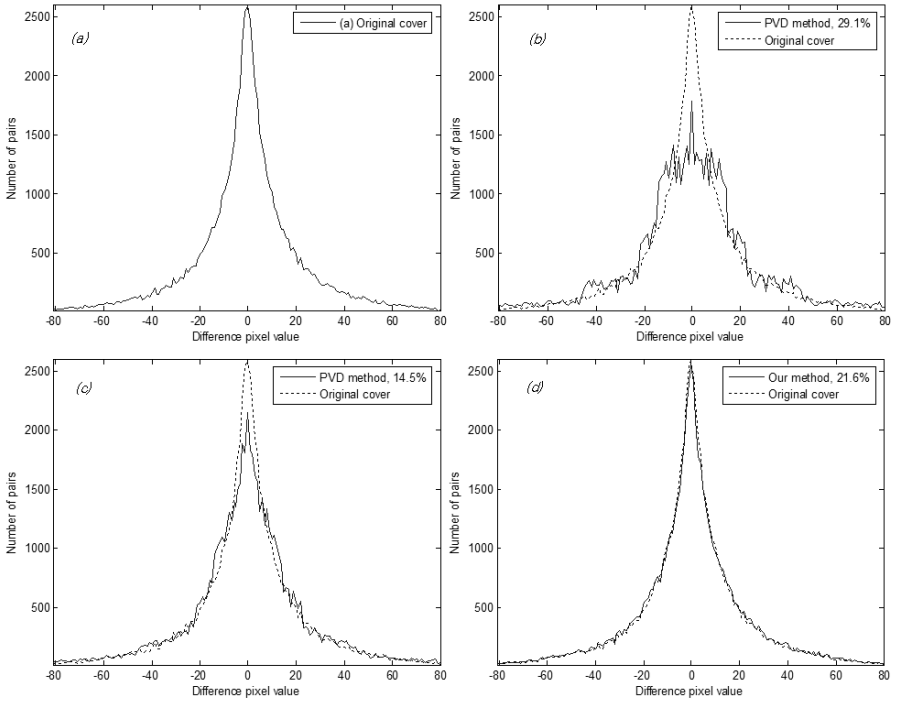


Fig. 4. Pixel difference histograms: (a) Pixel difference histogram of original Baboon (b) Pixel difference histogram of embedded cover with PVD method, embedding capacity = 29.1%, (c) Pixel difference histogram of embedded cover with PVD method, embedding capacity = 14.5%, (d) Pixel difference histogram of embedded cover with Our embedding method, embedding capacity = 21.6%

Acknowledgement

This work is partially supported by the Ministry of Education and Human Resources Development(MOE), the Ministry of Commerce, Industry and Energy(MOCIE) and the Ministry of Labor(MOLAB) through the fostering project of the Lab of Excellency.

References

[1] A. Westfeld, and A.Pfitzmann : Attacks on Steganographic Systems, in Proc. 3rd Information Hiding Workshop, LNCS 1768, Springer-Verlag, pp.61-76, 1999.
[2] N. Provos : Defending Against Statistical Steganalysis, in Proceedings of the 10 USENIX Security Symposium, pp. 323-335, 2001.
[3] J. Fridrich, M. Goljan, and R. Du : Reliable Detection of LSB Steganography in Color and Grayscale Images, in Proc. of ACM Workshop on Multimedia and Security, Ottawa, CA, pp. 27-30, October 2001.

[4] S. Dumitrescu, X. Wu, and Z. Wang : Detection of LSB Steganography via Sample Pair Analysis, in 5th Information Hiding Workshop, Noordwijkerhout, Netherlands, October 2002, LNCS 2578, Springer- Verlag, 2003.
[5] P. Lu, X. Luo, Q. Tang, and L. Shen : An Improved Sample Pairs Method for Detection of LSB Embedding, in 6th Information Hiding International Workshop, Toronto, Canada, May 23-25, 2004, LNCS 2578, Springer- Verlag, 2004.
[6] X. Luo, B. Liu, F. Liu: Detecting LSB Steganography Based on Dynamic Masks, in 5th International Conference on Intelligent Systems Design and Applications pp. 251-255, 2005.
[7] Wu, D.C., Tsai, W.H. : A steganographic method for images by pixel-value differencing. Pattern Recognition Letter. 24, pp. 1613-1626, 2003.
[8] X. Zhang and S. Wang : Vulnerability of pixel-value differncing steganography to histogram analysis and modification for enhanced security, in Pattern Recognition Letter 25, pp. 331-339, 2004.
[9] Westfeld, A.: High Capacity Despite Better Steganalysis (F5.A Steganographic Algorithm). In: Moskowitz, I.S. (eds.): Information Hiding. 4th International Workshop. Lecture Notes in Computer Science, Vol.2137. Springer-Verlag, Berlin Heidelberg New York, pp. 289. 302, 2001.

Reversible Watermarking for Error Diffused Halftone Images Using Statistical Features

Zhe-Ming Lu[1], Hao Luo[1,2], and Jeng-Shyang Pan[2]

[1] Visual Information Analysis and Processing Research Center, Harbin Institute of Technology Shenzhen Graduate School
518055 Shenzhen, P.R. China
zhemingl@yahoo.com, luohao723@126.com
[2] Department of Electronic Engineering, Kaohsiung University of Applied Sciences
807 Kaohsiung, Taiwan, ROC
hluo@bit.kuas.edu.tw, jspan@cc.kuas.edu.tw

Abstract. This paper proposes a reversible watermarking scheme for error diffused halftone images. It exploits statistical features of 2×2 binary patterns in halftone images to embed data. According to a small look-up table constructed in advance, a state sequence is extracted and losslessly compressed, and the saved space is filled up with the watermark and some side information. We modulate the extracted state sequence into a new concatenated sequence by similar pair toggling, and meanwhile the watermark and the LUT are embedded. The proposed scheme can provide a considerable capacity and the original image can be recovered if its watermarked version is intact.

Keywords: reversible watermarking, halftone image, statistical features.

1 Introduction

Digital halftoning is a process to transform continuous-tone images into two-tone images, e.g. from 8-bit gray level images to 1-bit binary images. Halftone images can resemble their original versions when viewing from distance by the low-pass filtering of the human eyes. Popular halftoning techniques can be divided into three categories: ordered dithering [1] based, error diffusion [2] based and direct binary search [3] based. Among these, error diffusion based techniques achieve a preferable tradeoff between the high visual quality and the reasonable computational complexity. With halftone images widely used in books, magazines, printer outputs and fax documents, it is desirable to embed data in this kind of images for copyright protection, content authentication and tamper detection. Up to now, many watermarking techniques have been developed for still image, audio, video, etc. [4]. Different from gray level or color images, there are mainly three challenges to embed data in halftone images. The first one is the less information redundancy for each pixel value is either black or white. Because of this, many data hiding approaches such as some transform domain based techniques cannot be directly transplanted to halftone images. Another challenge is the visual quality degradation. To insert data in halftone images, the change of the pixel value is either from black to white or vice versa. According to the

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 71 – 81, 2006.

study of the human visual system (HVS), human eyes are sensitive to the abrupt change aroused by watermarking, e.g., new appearances of the white cross and the black cross. The third one is the lower capacity compared with continuous-tone images. High capacity is one of the key factors to evaluate the performance of watermarking techniques. In fact, for halftone images, this challenge is closely related to the former two challenges. It is expected that a large quantity of data are difficult to be embedded into halftone images considering high distortion, for less information redundancy can be employed.

In recent years, a set of watermarking methods for halftone images are reported. These approaches can be divided into three classes: (1) pixel-based: to change the values of individual pixels usually randomly selected [5]. (2) block-based: to partition the original image into pixel blocks and modify the characteristics of some blocks [6] [7] [8]. (3) hybrid-based: to insert data by combining the characteristics of pixel-based and block-based [9] [10]. However, most existing watermarking methods cannot recover the original image because of the irreversible distortion introduced. Although the distortion is slight, it may not satisfy the requirement of some specific applications, where content accuracy of the host image must be guaranteed, e.g., military maps, medical images, great works of art, etc. Therefore, it is quite necessary to develop a reversible watermarking method for halftone images. Nevertheless, up to now, there has been little attention paid to this research theme.

This paper presents a hybrid-based reversible method whose original idea is motivated from the R-S algorithm developed by Fridrich et al. [11]. Firstly, we investigate the statistical features of the 2×2 binary patterns in an error diffused halftone image. Based on this, a look-up table (LUT) is constructed. It consists of two pairs of similar block patterns. We use 0 and 1 to denote the states of two groups of patterns respectively. Secondly, searching all blocks in the image: if the current block is the same as some patterns in the LUT, record its state. Thus a state sequence can be obtained. Thirdly, the state sequence is losslessly compressed and the saved space is filled up with the watermark and some side information. In our context the side information refers to some extra data aroused by the LUT embedding. Next, the watermark is embedded by similar pair toggling with reference to the new state sequence. The last step is to insert the LUT with a secret key, and meanwhile the watermarked halftone image is obtained. In the watermark extraction stage, the LUT must be retrieved first and other procedures are just the inverse process of the embedding. As a reversible technique, the original image can be perfectly recovered if its watermarked version is intact. Moreover, our approach can be easily extended for halftone image lossless authentication, e.g. hiding an image hash.

The rest of this paper is organized as follows. Section 2 briefly reviews the error diffusion halftoning process. Section 3 describes the proposed method. In Section 4, experimental results are presented and discussed. Section 5 concludes the whole paper.

2 Error Diffusion Halftoning

Error diffusion is a halftoning technique which can produce high quality halftone images. The flow chart of the error diffused halftoning process is shown in Fig. 1.

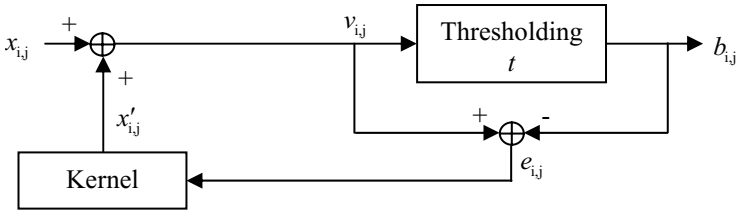


Fig. 1. Flow chart of the error diffusion halftoning

When halftoning a continuous-tone image line by line sequentially, the past error is diffused to the current pixel. Suppose 0 and 255 denote a black pixel and a white pixel respectively. $x_{i,j}$ is the current processing pixel and $x'_{i,j}$ is the diffused error sum added up from the neighboring processed pixels. $b_{i,j}$ represents the binary output at Position (i,j). $v_{i,j}$ is the modified gray output and $e_{i,j}$ is the difference between the $v_{i,j}$ and the $b_{i,j}$. The relationships of these variables are described below.

$$v_{i,j} = x_{i,j} + x'_{i,j} \ . \tag{1}$$

$$x'_{i,j} = \sum_{m=0}^{1} \sum_{n=-1}^{1} e_{i+m,j+n} \times k_{m,n} \ . \tag{2}$$

$$e_{i,j} = v_{i,j} - b_{i,j} \ . \tag{3}$$

$$b_{i,j} = \begin{cases} 0 & \text{if } v_{i,j} < t \\ 1 & \text{if } v_{i,j} \geq t \end{cases} \ . \tag{4}$$

where the parameter t in the equation (4) is a threshold usually set as 128. From the Fig. 1, we can see that different kernels $k_{m,n}$ correspond to different visual quality of halftone images. Nowadays, several available the error diffusion kernels are good choices. For example, Floyed-Steinberg [2], Jarvis [12] and Stucki [13] are three popular error diffused kernels which applied to transform gray level images into halftone images. These kernels are shown in Fig. 2, Fig. 3 and Fig. 4 respectively.

	X	7/16
3/16	5/16	1/16

Fig. 2. Floyed-Steinberg error diffusion kernel (X is the current pixel)

		X	7/48	5/48
3/48	5/48	7/48	5/48	3/48
1/48	3/48	5/48	3/48	1/48

Fig. 3. Jarvis error diffusion kernel (X is the current pixel)

		X	7/42	5/42
2/42	4/42	8/42	4/42	2/42
1/42	2/42	4/42	2/42	1/42

Fig. 4. Stucki error diffusion kernel (X is the current pixel)

3 Proposed Method

3.1 Statistical Features

The original image is partitioned into non-overlapping 2×2 blocks. Obviously there are 16 patterns P_1, P_2…, P_{16} of a 2×2 binary block, as shown in Fig. 5. We investigate their appearance frequencies in halftone images. Through numerous heuristic experiments, it is amazing to find that in most cases the two patterns P_4 and P_{13} rarely appear compared with the other two P_7 and P_{10}. If we consider a 2×2 binary pattern as a fine texture in halftone images, it is easy to understand the error diffusion halftoning process produces a large quantity of P_7 and P_{10}, in contrast much fewer P_4 and P_{13}. Although Floyd-Steinberg, Jarvis and Stucki kernels are various in sizes or weight values at different locations, similar visual quality of the halftone images can be obtained by performing these kernels error diffused filtering on corresponding 8-bit gray level images. Consequently, the statistical features widely exist in these halftone images.

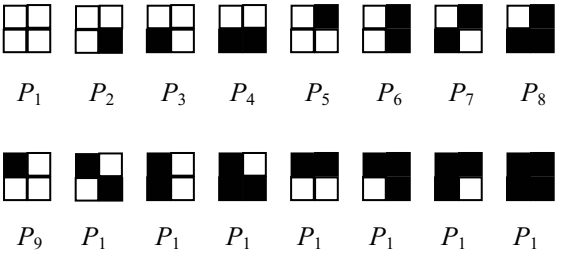


Fig. 5. 2×2 binary patterns

For simplicity, in the following sections, we only focus on the images obtained by Floyd-Steinberg kernel halftoning. To illustrate the statistical features, an example is shown in Fig. 6, where six 512×512 halftone images (see Fig. 13), Lena, F16, Baboon, Pepper, Boat and Barbara are examined. The statistical features can be exploited to embed data.

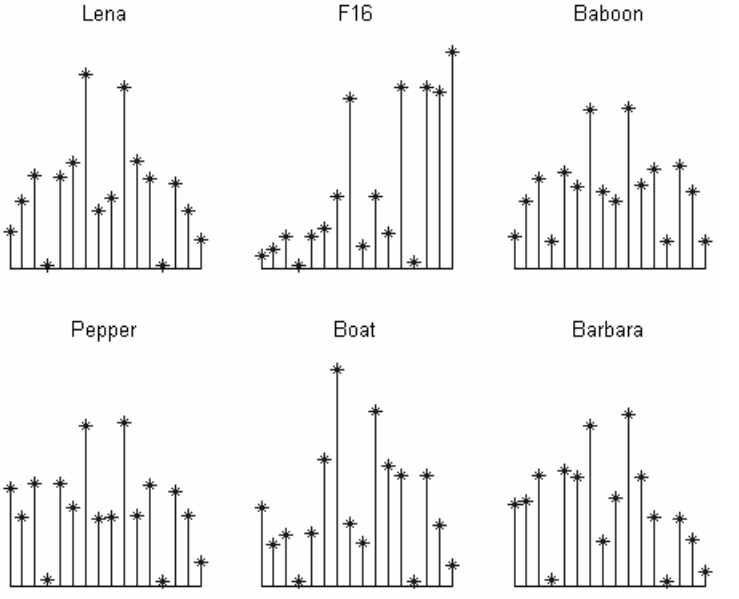


Fig. 6. 16 2×2 patterns' appearance times in six 512×512 halftone images

3.2 Watermark Embedding

Suppose I and I_W denote the original image and the watermarked image respectively, and W denotes the original watermark. The block diagram of the watermark embedding is shown in Fig. 7, with steps given as follows.

(1) Image partition. We partition I into non-overlapping 2×2 pixel blocks.

(2) LUT construction. Before embedding data, we need to construct a LUT. The LUT refers to two pairs of 2×2 patterns $\{H_1, L_1\}$, $\{H_2, L_2\}$, where H_k (k=1,2) is defined as the similar pattern of L_k and vice versa. Suppose N_i $(1 \le i \le 16)$ denotes the number of appearance times of P_i. According to the statistical features, L_1 and L_2 are

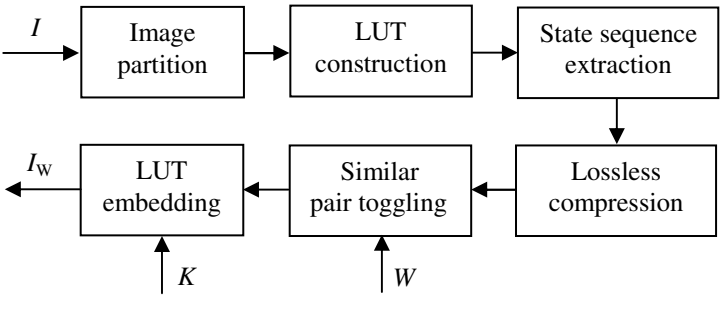


Fig. 7. Block diagram of watermark embedding

fixed as P_4 and P_{13} respectively, and H_1 and H_2 are selected as P_7 and P_{10}. After counting N_4, N_{13}, N_7 and N_{10} in I, a LUT is constructed. An example LUT is shown in Fig. 8.

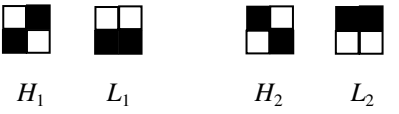


Fig. 8. An example LUT

(3) State sequence extraction. We use 0 and 1 denote the state of H_k and L_k respectively. Searching all blocks in I, if we come across H_k or L_k, we record their corresponding state. Hence a binary state sequence $S = \{s_1, s_2, ..., s_r\}$ is obtained, where r is the length of S. Obviously, r is equal to the sum of N_4, N_{13}, N_7 and N_{10}.

$$r = N_4 + N_7 + N_{10} + N_{13} . \tag{5}$$

(4) Lossless compression. In this step, S is losslessly compressed into S_C, in our case the arithmetic coding is used. The saved space S-S_C is filled up with W and side information SI, as shown in Fig. 9. In other words, the new state sequence $S' = \{s'_1, s'_2, ..., s'_r\}$ is the concatenation of S_C, W and SI, whose length is equal to that of S.

(5) Similar pair toggling. This operation aims to embed the watermark. Searching all blocks in I again, if the current block CB is the same as H_k and the corresponding S' bit s'_d ($1 \le d \le r$) is 1, we replace H_k with L_k; if CB is equal to L_k and the corresponding S' bit is 0, L_k is replaced with H_k. In the other two cases, no replacement is done. In a word, states of H_k and L_k are modulated into S' as

$$\begin{cases} H_k \leftarrow L_k & \text{if } CB = H_k, s'_d = 1 \\ L_k \leftarrow L_k & \text{if } CB = L_k, s'_d = 1 \\ H_k \leftarrow H_k & \text{if } CB = H_k, s'_d = 0 \\ L_k \leftarrow H_k & \text{if } CB = L_k, s'_d = 0 \end{cases} . \tag{6}$$

where $\leftarrow$ means replacing the left pattern with the right one. In this way, the watermark is embedded.

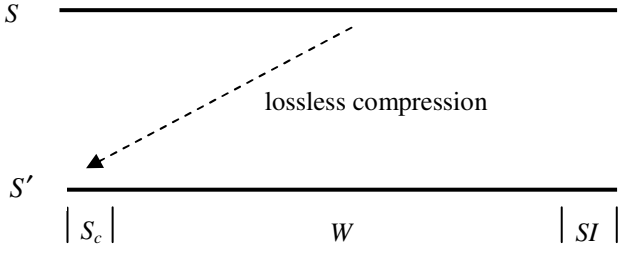


Fig. 9. State sequence lossless compression and the saved space allocation

(6) LUT embedding. It is necessary to protect the LUT. As its size is 16 bits, we randomly choose 16 pixels locations using a pseudo-random number generator with a key K. Note these pixels must not fall into H_k or L_k blocks. As side information $SI=\{si_1, si_2, \ldots, si_{16}\}$, the 16 pixels values are retrieved into a binary sequence, and concatenated with S_C and W, then embedded based on aforementioned similar pair toggling. After that, the LUT is inserted into these 16 pixels locations by directly replacing. Thus the watermarked image I_W is obtained.

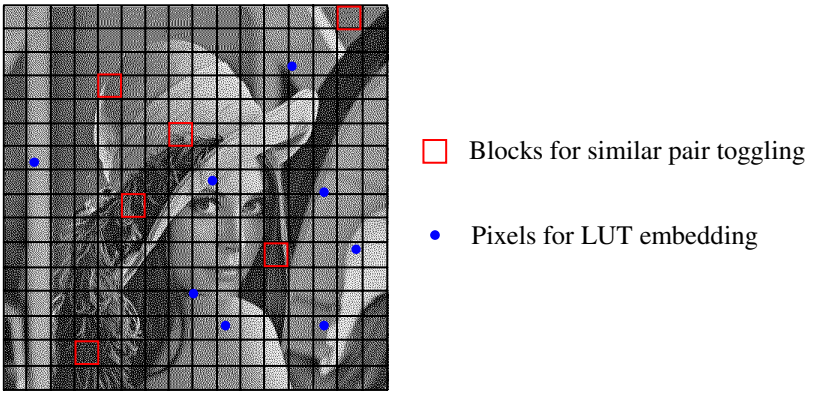


Fig. 10. LUT embedding

The schematic diagram of LUT embedding is shown in Fig. 10, where the red blocks represent patterns H_k or L_k used to extract the state and simialr pair toggling, and the blue points represent the pixels selected with K, these pixels values are extracted as SI, and then the rearranged LUT is inserted into these locations.

3.3 Watermark Extraction and Lossless Recovery

The watermark extraction and lossless recovery is the inverse process of watermark embedding. The block diagram of the watermark extraction and lossless recovery is shown in Fig. 11, with steps given as follows.

(1) Image partition. I_W is also partitioned into non-overlapping 2×2 pixel blocks.

(2) LUT reconstruction. The same key K is used to find the 16 pixels locations and the retrieved pixel values can be rearranged into the LUT.

(3) State sequence extraction. According to the LUT, in I_W we search blocks except those where $si_1, si_2, \ldots, si_{16}$ lies in and a state sequence also can be extracted. If I_W suffers no alteration, this sequence is exactly the same as S'. The middle part is extracted as the watermark, as shown in Fig. 12.

(4) Lossless decompression. The first part S_C is losslesly decompressed into S using arithmetic decoding.

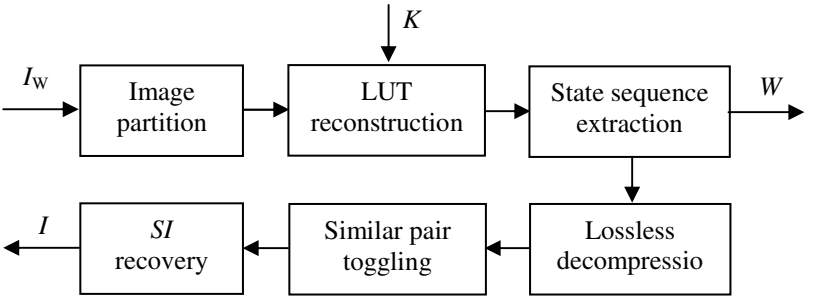


Fig. 11. Block diagram of watermark extraction and lossless recovery

(5) Similar pair toggling. Searching H_k and L_k in I_W and modulate S' into S based on the following rule: if we come across H_k and the corresponding S bit is 1, we replace H_k with L_k; if we come across L_k and the corresponding S bit is 0, L_k is replaced with H_k. In the other two cases, there is no replacement. The operation is as follows

$$\begin{cases} H_k \leftarrow L_k & \text{if } CB = H_k, s_d = 1 \\ L_k \leftarrow L_k & \text{if } CB = L_k, s_d = 1 \\ H_k \leftarrow H_k & \text{if } CB = H_k, s_d = 0 \\ L_k \leftarrow H_k & \text{if } CB = L_k, s_d = 0 \end{cases}. \tag{7}$$

(6) Side information recovery. The last 16 bits of S' is recovered as *SI*, and they are used to replace the pixels localized by *K*.

Thus, after modulating S' into S by similar pair toggling and recovering side information, the original image is perfectly recovered.

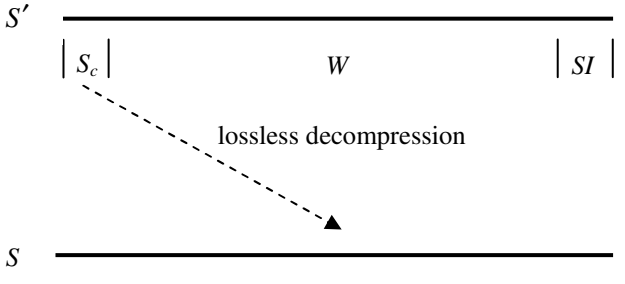
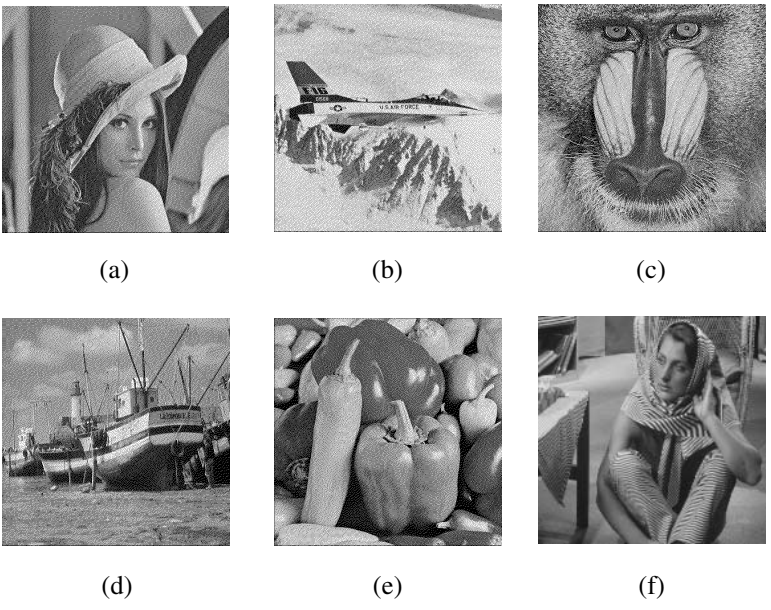


Fig. 12. State sequence lossless decompression for watermark extraction and side information recovery

4 Experimental Results

Six 512×512 halftone images as shown in Fig. 13 are selected to test the performance of our scheme. Capacities are listed in Table 1. As a reversible technique, these



(a) (b) (c)

(d) (e) (f)

Fig. 13. Test 512×512 halftone images, (a) Lena, (b) F16, (c) Baboon, (d) Boat, (e) Pepper, (f) Barbara

Table 1. Capacities of test images

Halftone image	Capacity (bit)
Lena	17550
F16	4903
Baboon	1806
Pepper	13588
Boat	17460
Barbara	14142

capacities are considerable for the halftone image is 1 bit deep. The average luminance of the watermarked image is approximately equal to that of the original image. This is because all of the average gray levels of P_4, P_{13}, P_7 and P_{10} are equal and thus similar pair toggling among them can not result in the sharp luminance change. Furthermore, the capacity is determined by the number of these four patterns' appearance times. In other words, it is content-dependent, and thus high distortion can be avoided. For example, in Table 1, we can see Lena can be hidden much more information than Baboon. Therefore, usually the introduced distortion can be accepted. Fig. 14 lists the experiment results on Lena and Baboon. The watermarks are a 17550 bits and 1806 bits binary sequence created by a pseudo-random number generator. The host images are recovered since the normalized cross-correlation values between the original and the recovered images is 1.

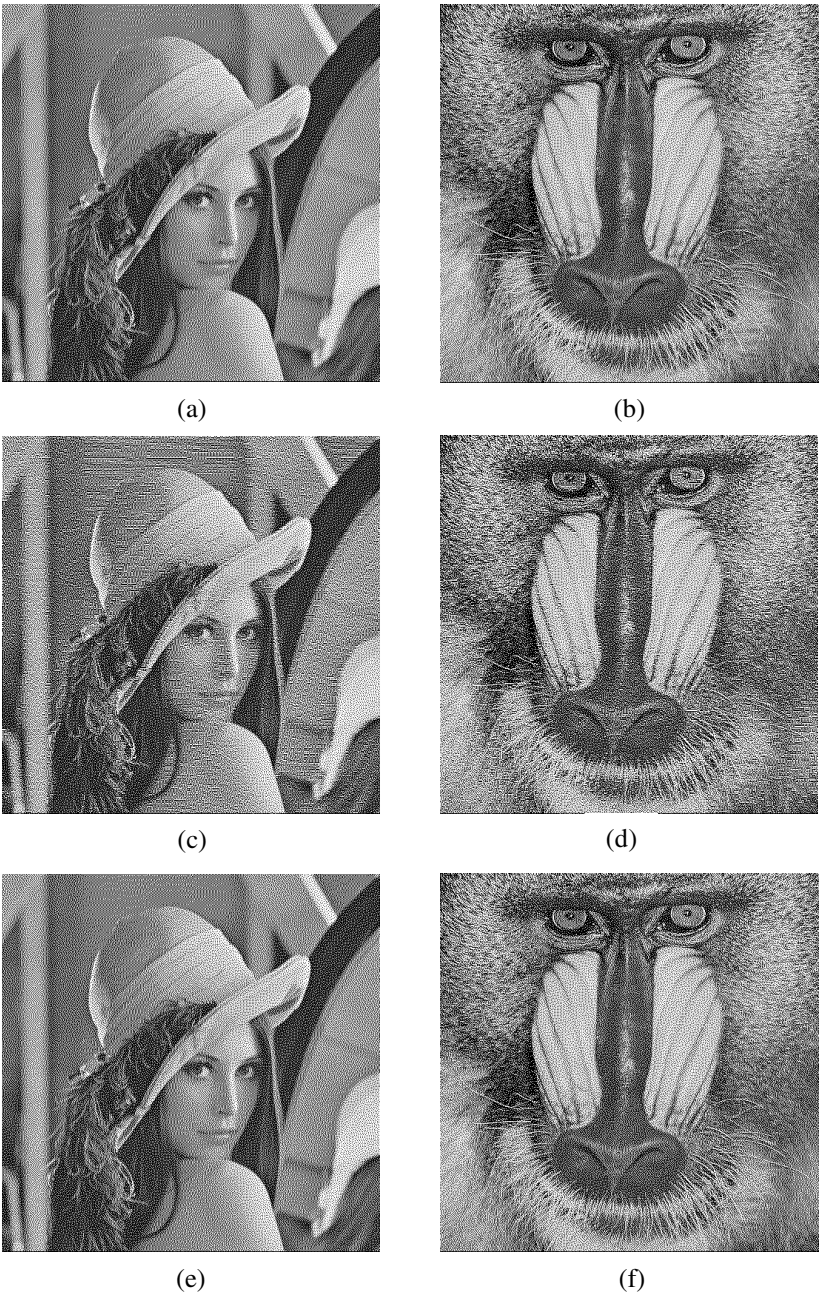


Fig. 14. Reversible watermarking on the halftone Lena and Baboon, (a) the original Lena, (b) the original Baboon, (c) the watermarked Lena, (d) the watermarked Baboon, (e) the recovered Lena, (f) the recovered Baboon

5 Conclusions

In this paper, we propose a reversible watermarking scheme for error diffused halftone images. It employs specific statistical features of 2×2 patterns appearance in this kind of images to embed data. The original image can be recovered if the watermarked image is intact. Our scheme can be used in some applications where content accuracy of the original image must be guaranteed. Besides, it is easily extended to halftone image lossless authentication, e.g. hiding an image hash.

Acknowledgments

This work was supported by the Program for New Century Excellent Talents in University of China under grant NCET-04-0329 and Foundation for the Author of National Excellent Doctoral Dissertation of P. R China (No. 2003027).

References

1. Ulichney, R.: Digital Halftoning. Cambridge, MA: MIT Press (1987)
2. Floyd, R. and Steinberg, L.: An adaptive algorithm for spatial gray scale. SID. Int. Symp. Dig. Tech. Papers. (1975), 36–37
3. D. Lieberman, J. Allebach.: Digital Halftoning Using the Direct Binary Search Algorithm. Proceedings of the 1996 IST International Conference on High Technology, Japan (1996) 114–124
4. Pan, J. S., Huang, H. C. and Jain, L. C. (eds.): Intelligent Watermarking Techniques. World Scientific Publishing Company. Singapore. (2004)
5. Fu, M. S., Au, O. C.: Data Hiding Watermarking for Halftone Images. IEEE. Transaction on Image Processing. vol. 11, no. 4, (2002) 477–484
6. Baharav, Z., Shaked, D.: Watermarking of Dither Halftone Images. Hewlett-Packard Labs Tech Rep, HPL-98-32, (1998)
7. Hel-Or, H. Z..: Watermarking and Copyright Labeling of Printed Images. Journal of Electronic Imaging, (2001) 794–803
8. Liao, P. S., Pan, J. S., Chen, Y. H., Liao, B. Y.: A Lossless Watermarking Technique for Halftone Images. KES (2) (2005) 593–599
9. Pan, J. S., Luo, H., and Lu, Z. M.: A Lossless Watermarking Scheme for Halftone Image Authentication. 6(2B) (2006) 147–151
10. Pei, S.C., Guo, J. M.: Hybrid Pixel-Based Data Hiding and Block-Based Watermarking for Error-Diffused Halftone Images. IEEE Trans Circuits and Systems for Video Technology, (2003) 867–884
11. Fridrich, J., Goljan, M., and Du, R.: Lossless Data Embedding New Paradigm in Digital Watermarking, Special Issue on Emerging Applications of Multimedia Data Hiding, Vol. 2002, No.2 (2002) 185–196
12. Jarvis, J. F., Judice, C. N. and Ninke, W. H.: A survey of techniques for the display of continuous-tone pictures on bilevel displays. Computer Graphics Image Process. vol. 5, (1976) 13–40
13. Stucki, P.: MECCA – A multiple error correcting computation algorithm for bilevel image hardcopy reproduction, Research Report RZ1060, IBM Res. Lab., Zurich, Switzerland, (1981)

Wavelet Domain Print-Scan and JPEG Resilient Data Hiding Method

Anja Keskinarkaus, Anu Pramila, Tapio Seppänen,
and Jaakko Sauvola

MediaTeam, Oulu
University of Oulu
FINLAND
anja.keskinarkaus@ee.oulu.fi

Abstract. In this paper we present a print-scan resilient method to embed multibit messages into images. Multilevel watermarking principles are applied in order to embed a reference watermark and the message bits. Methods to embed a robust spatial domain reference watermark, utilizing HVS are proposed and methods to improve the estimation of affine transformation parameters from a periodic reference watermark are considered. The multibit message is embedded on the approximation coefficients of the wavelet transform, utilizing JND profile estimation and additive spread spectrum techniques. A blind correlation based detection method is made use of to recover the message. In the experiments the imperceptuality issues related to multilevel watermarking are considered with different parameter settings and the robustness against print-scan attack and JPEG compression is measured and the results are shown.

Keywords: Watermarking, JND profile estimation, periodic reference watermark, spread spectrum technique, multilevel technique, affine transformation.

1 Introduction

Development of Internet and numerous hardware and software applications have brought abundance of possibilities to use and distribute multimedia content. From vendors point of view, this development has created many opportunities but also posed threats in the form of copyright infringements. The majority of the watermarking research has focused on threats and on developing methods to hide information about the copyright owner to the piece of media content and later to trace copyright violations. But this is only one of many possible ways to use watermarking in digital media distribution.

Another interesting field of study in watermarking is value adding services, which enhance the value of the multimedia content by offering extra services and information. Therefore, the embedded information is beneficial to the user and intentional attacks are not expected. Although such intentional attacks may not

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 82 – 95, 2006.

appear, some attacks must be considered. For example in print-and-scan process, digital-to-analog-to-digital conversion, and some geometrical distortions are bound to happen. Additionally, a common way to store and handle images is in compressed JPEG format.

So far, few methods have been proposed to overcome problems caused by geometrical distortions and print-and-scan process in image watermarking. Lin and Chang [1], for example, propose a model for the print-scan process by considering pixel value and geometric distortions separately. They propose a watermarking method based on log polar map of discrete Fourier transform (DFT) magnitudes (i.e., the Fourier-Mellin transform). Fourier-Mellin transform is also used by O'Ruanaidh and Pun in [2], because it is a domain that is invariant to rotation, scaling and translation.

Another approach is to use salient image feature points for determining geometrical distortions, as in [3] by Bas *et al*. First, the feature points of the image are extracted and the Delaunay tessellation is performed on this set of points. The watermark is embedded using a classical additive scheme inside each triangle of the tessellation. The detection is done using correlation properties on the different triangles. The main advantage of this method is that the orientation of the watermark is carried by the content of the image itself. Unfortunately, as feature points move or even disappear when geometrical attacks are applied to the watermarked image, also the tessellation may vary. Therefore it is difficult to extract the feature points identical to those of an original image.

The third proposed approach uses a special template watermark to detect transformations undergone by a watermarked image. Pereira and Pun [4] proposed to embed a template consisting of a random arrangement of peaks in the Fourier domain. These points can then be used to estimate the geometric transformations that the image has gone through. Template embedding algorithms are fairly robust and simple to implement unlike algorithms based on feature points or invariant domains. Pereira and Pun used the magnitude of the Fourier transform domain, which has a few drawbacks: every embedded point affects the whole image in spatial domain and therefore the amount of peaks and the strength at which they are embedded is restricted by the visibility constraint of the watermark. In addition, the template in the magnitudes of the Fourier transform domain cannot be used to detect translation.

Kutter found out in [5] that a separate template signal is not always necessary and used watermark itself as a calibration signal. Consequently, a watermark is embedded several times at different, horizontally and vertically shifted locations into the blue image component and has a form of a weighted spread spectrum signal. The watermark can then be recovered after geometric distortions by applying auto-correlation. Deguillaume *et al.* [6] took this idea a bit further and embedded a periodical structure with many repetitions in order to get a high number of peaks. They reasoned that a larger number of peaks survive compared to [5], even under severe signal fading. After finding the peaks by employing auto-correlation function, they used the fact that random points are less likely to present significant alignments than the correct ones. Then, with the Hough or the Radon transform, a robust estimator of the correct underlying grid can be found from the periods of the aligned points. This estimator can also be used for calculating the affine transform parameters.

In this paper we propose a multilevel watermarking method, in which we adopt the utilization of a grid structure [6] in order to invert the geometrical attacks. We propose methods to enhance the accuracy of the estimation, specifically considering estimation after a print-scan attack. For embedding the actual multibit message we consider wavelet domain additive spread spectrum techniques with a blind correlation based receiver. As pointed out in [7], the wavelet domain has several advantages when considering watermarking. In our approach, approximation coefficients are utilized to attain robustness against print-scan and compression attacks. Generally the vulnerability of the wavelet domain methods is due to the fact that even a small geometrical distortion changes the energy distribution so that robust extraction is not possible. We show with experiments that consequently to an accurate affine transform parameter estimation, watermarking methods in wavelet domain can be utilized. The robustness and the imperceptuality of the grid and the message are enhanced utilizing HVS modeling and calculating the full band JND profile and an energy weighted JND accordingly. We propose a method to efficiently utilize JND when two different domains are used for embedding and multilevel techniques are utilized.

In section 2, we describe the procedure for estimating the affine transform parameters; the embedding of a robust and invisible grid, the estimation of rotation, scale and translation. In section 3, the procedure for embedding the multibit message and the message extraction process is explained. Section 4 contains the experiments on image quality, the robustness of message extraction after print-scan and JPEG compression attacks.

2 Estimation of Affine Transform

For estimation of rotation and scaling the ACF (autocorrelation function) of a periodic watermark, proposed by Deguillaume *et al*. [6], is adopted. A visually adapted periodic watermark is embedded in spatial domain, the Wiener estimate of the watermark is calculated and autocorrelation function peaks are utilized for determining rotation angle and scaling ratio. A highly robust grid is embedded utilizing JND analysis which is based on the method proposed in [8], where a perceptual model is used to calculate JND values for each pixel in an image.

The extraction of the autocorrelation peaks is enhanced utilizing Sobel filtering for autocorrelation function scaled to the range of [0 1]. The problem of missing and false peaks addressed by Deguillaume *et al*. [6] is considered and methods to overcome the problem are proposed. The validity of the proposed methods is shown in the experimental section

Translation of the image is determined making use of the underlying grid structure. The method is flexible in a sense that the marginal over which search is done can be determined. This way the computational complexity compared to the full search methods is much lower.

2.1 Embedding of the Periodic Watermark

A periodic watermark satisfying the equations 1 and 2 is generated from a pseudorandom sequence of values {-1,1}.

$$W(x+q_0N_0,y)=W(x,y);\quad q_0,N_0>1 \tag{1}$$

$$W(x,y+q_1N_1)=W(x,y);\quad q_1,N_1>1 \tag{2}$$

The watermark is embedded into the host image in spatial domain, utilizing equation

$$Y^*(x,y)=X(x,y)+\lambda_1\cdot JND_{fb}\cdot W(x,y), \tag{3}$$

where Y^* is the watermarked image, X is the luminance component of the original image, W is the periodic watermark and x and y describe the pixel position. JND_{fb} is the scaling factor attained from the JND profile, as explained in the next chapters, and λ_1 is an additional scaling factor. In our experiments the fundamental period was set to a fixed value of 16 and N_0=N_1, without a loss of generalization.

Several different models of the Human Visual System have been proposed for embedding a perceptually transparent and optimally robust watermark. Based on our earlier experiments and good results in [9], we base our watermark adaptation to the model proposed by Chou and Li [8]. The method is based on calculating a JND threshold for each pixel based on two properties of human eye, one of being the average background luminance behind the pixel and the other the spatial uniformity of the background luminance. The calculated full band JND profile is exploited for calculating a threshold for allowable distortion for every pixel value in an image. The overall calculation is accomplished using equations 1-5 in [8]. Fixed parameters are set to values proposed by the authors.

2.2 Estimation of Rotation, Scale and Translation from Printed and Scanned Image

During printing and scanning process, the watermarked image undergoes several attacks, including distortion of pixel values caused by luminance, contrast, gamma correction and chrominance variations, and blurring of adjacent pixels [1]. The most severe attacks, as far as watermarking is concerned, are caused by affine transforms as illustrated in figure 1. The image area to be scanned is defined by the user, meaning that a significant portion of background, additionally to the actual image, is usually also cropped during scanning process. Additionally, the image can be arbitrarily rotated and scaled.

The proposed algorithm includes the following steps for estimating the affine transform parameters. Considering a watermarked image $Y^{**}(x,y)$, where** denotes that multilevel watermarking has been applied, a windowing operation defined by equation 5 is utilized. We define windowing function

$$\xi(x,y)=\begin{cases}1, & N_3<x<N_4, M_3<y<M_4\\ 0, & otherwise\end{cases}, \tag{4}$$

in which the parameters N_3,N_4,M_3 and M_4 define the image area to be analyzed. The experiments supported the results of Alvarez-Rodríguez and Pérez-González [11] of the accuracy of the angle, and consequently the areas must be defined to be large enough. Utilizing equation 5, where $\times$ denotes the windowing operation the image, the image area to be analysed is defined as follows.

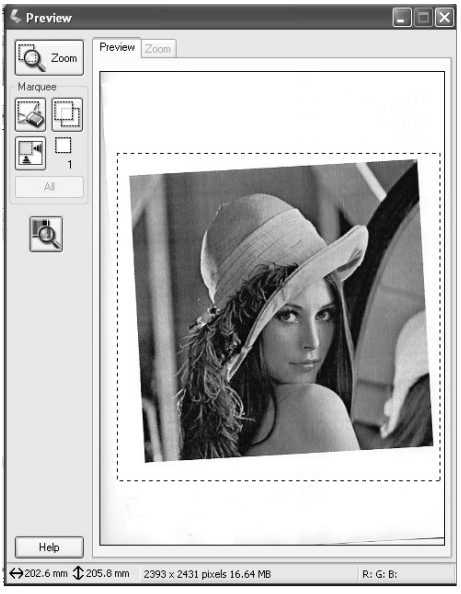


Fig. 1. The user interface of the EPSON perfection, 4180 photo scanner

$$\boldsymbol{Y}^{**(N_3,N_4,M_3,M_4)}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{\xi}(\boldsymbol{x},\boldsymbol{y}) \times \boldsymbol{Y}^{**}(\boldsymbol{x},\boldsymbol{y}) \tag{5}$$

The image area is filtered in order to get the Wiener estimate $\tilde{\boldsymbol{W}}(\boldsymbol{x},\boldsymbol{y})$ of the periodic grid structure

$$\tilde{\boldsymbol{W}}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{Y}^{**(N_3,N_4,M_3,M_4)}(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{h}(\boldsymbol{k}) * \boldsymbol{Y}^{**(N_3,N_4,M_3,M_4)}(\boldsymbol{x},\boldsymbol{y}), \tag{6}$$

where *h(k)* represents the adaptive Wiener filtering. Autocorrelation function $\boldsymbol{R}_{\tilde{\boldsymbol{W}},\tilde{\boldsymbol{W}}}(\boldsymbol{u},\boldsymbol{v})$ is utilized in order to reveal the periodicity in the extracted watermark estimate

$$\boldsymbol{R}_{\tilde{\boldsymbol{W}},\tilde{\boldsymbol{W}}}(\boldsymbol{u},\boldsymbol{v}) = \sum_{\boldsymbol{x}=N_3}^{N_4} \sum_{\boldsymbol{y}=M_3}^{M_4} \tilde{\boldsymbol{W}}(\boldsymbol{x},\boldsymbol{y}) \tilde{\boldsymbol{W}}(\boldsymbol{x}+\boldsymbol{u},\boldsymbol{y}+\boldsymbol{v}). \tag{7}$$

The autocorrelation is scaled to the range of [0,1]

$$\boldsymbol{R}_{\tilde{\boldsymbol{W}},\tilde{\boldsymbol{W}}}{}^{*}(\boldsymbol{u},\boldsymbol{v}) = \left| \boldsymbol{R}_{\tilde{\boldsymbol{W}},\tilde{\boldsymbol{W}}}(\boldsymbol{u},\boldsymbol{v}) \right| / \max(\boldsymbol{R}_{\tilde{\boldsymbol{W}},\tilde{\boldsymbol{W}}}(\boldsymbol{u},\boldsymbol{v})). \tag{8}$$

Sobel filtering is utilized to enhance the detection of the peaks generated by the periodicity. The horizontal edges are emphasized using the smoothing effect by approximating a vertical gradient. Filtering operation (equation 9) proved to be an effective and fast method to improve the distinguishing of peaks. In

$$R_{\tilde{W},\tilde{W}}^{**}(u,v) = h_s(k) * R_{\tilde{W},\tilde{W}}^{*}(u,v), \tag{9}$$

$h_s(k)$ represents the Sobel filtering kernel. A binary grid image $g(u,v)$ is generated

$$g(u,v) = \begin{cases} 1, & when \quad R_{\tilde{W},\tilde{W}}^{**}(u,v) \geq \gamma \\ 0, & when \quad R_{\tilde{W},\tilde{W}}^{**}(u,v) < \gamma \end{cases}. \tag{10}$$

As suggested by Kutter [5], iterative approaches can be utilized for determining the threshold. Although we in the experimental section utilize varying threshold settings, the proposed approach is not directly dependent on the threshold. As a consequence of Cauchy-Schwarz inequality, the autocorrelation function reaches its peak at the origin, additionally the periodicity of autocorrelation is the same for every pairs of N_3,N_4 and M_3,M_4. Utilizing these properties the final grid can be constructed as follows

$$g^{*}(u,v) = \sum g^{(N_3,N_4,M_3,M_4)}(u,v), \tag{11}$$

where summation is origin centric. The proposed method is effective on patching up the missing peaks. An example of extracted $g^*(u,v)$ is illustrated in figure 2. Usually the size of a scanned image is very large and estimation, specially the calculus of autocorrelation, is time consuming. Resizing to a smaller size reduces significantly the computational complexity. In the example the scanned image has been resized to 25 % of the original image size, therefore the computational savings are significant and the experiments showed that the accuracy of the estimation of affine parameters does not suffer. Additionally N_4-N_3>M_4-M_3 the grid structure is thus biased towards horizontal direction, decreasing the probability of confusing the main axes to the diagonals.

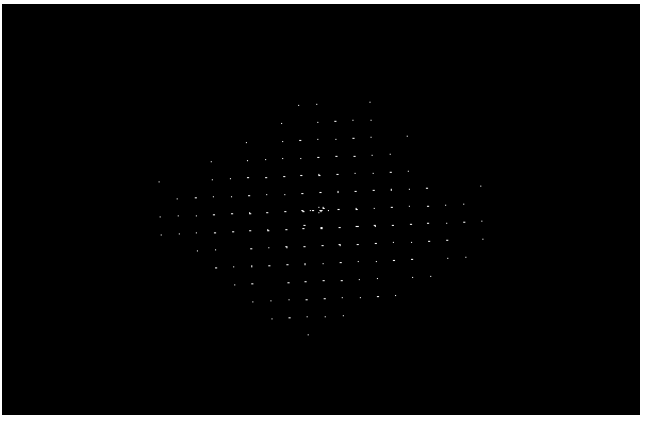


Fig. 2. The extracted $g^*(u,v)$ from Lena image (512x512), printed with HP Color LaserJet 4650, scanned with EPSON perfection, 4180 photo scanner with resolution 300 dpi

The Hough transform is utilized in order to calculate the discrete points of the grid structure, as suggested in [6]. The rotation angle is determined by examining the line segments (figure 3), determined from the peaks in the Hough transform matrix.

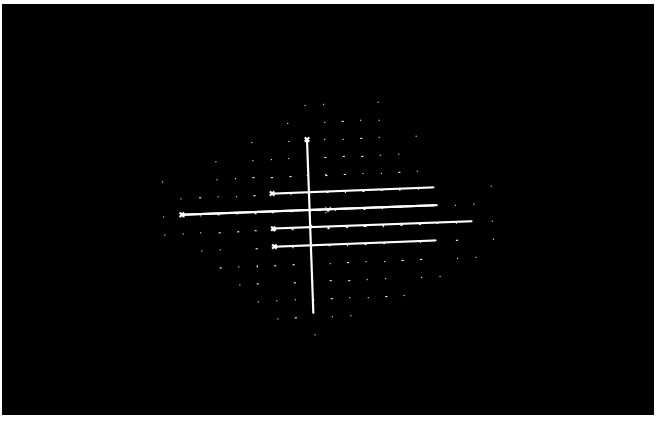


Fig. 3. The detected line segments

The following steps are utilized for the estimation of the precise angle. This is based on the fact that most of the noisy points are situated along the line going through the origin and in the angle of rotation.

1. Find the lines in the grid.
2. Sort the lines according to the length of the line segment
3. Find the most dominant direction of lines.
4. Select the one not going through the origin
5. Determine angle θ utilizing endpoints of the selected line.

The scaling ratio is determined from a grid structure rotated according to the determined angle θ. Rotation of the form

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{12}$$

results to $g^*(u',v')$. The effect of false peaks is reduced utilizing

$$g^{**}(u',v'), \quad \begin{cases} 1 \le v' < \max(v')/2 - k, \textit{ for horizontal scale} \\ 1 \le u' < \max(u')/2 - k, \textit{ for vertical scale} \end{cases}, \tag{13}$$

where k is an experimental constant.

The grid $g^{**}(u',v')$ is projected against horizontal and vertical axis

$$p(u') = \sum_{v'} g^{**}(u',v') \quad \textit{and} \quad p(v') = \sum_{u'} g^{**}(u',v') \tag{14}$$

As seen from figures 4 and 5, the effect of false peaks diminishes. Generally when γ is decreased this effect is strengthened and can be much more severe than in figure 4, where the false peaks appear in the center region.

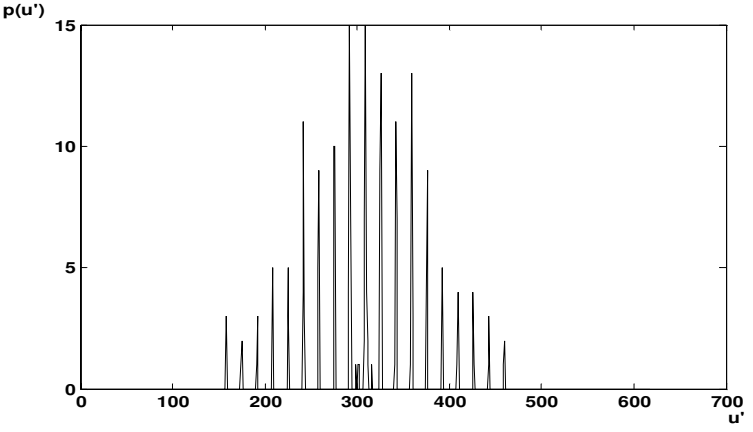


Fig. 4. p(u') calculated from g**(u',v')

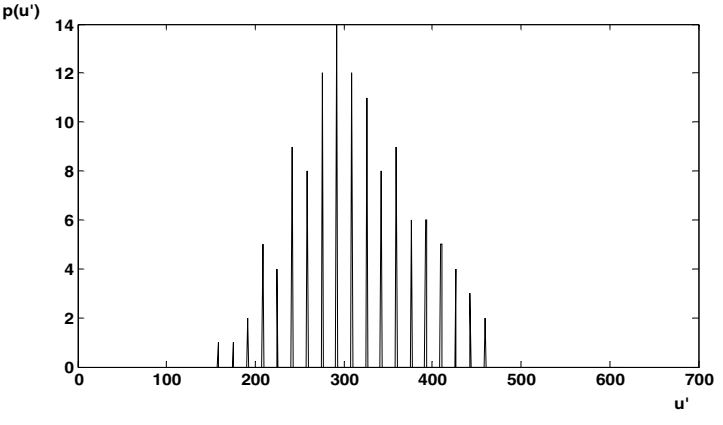


Fig. 5. p(u') calculated from g**(u',v') as proposed

The determination of the scaling ratio is realized by calculating the Euclidean distances between the clusters in *p(u')* and *p(v')* and applying voting strategy.

According to the estimated parameters, the angle of rotation and the horizontal and vertical scaling ratio, an inversion of rotation and scaling is done. An example of the result of the inversion process is illustrated in figure 6. An inversion commonly involves interpolation, which can be considered as another attack, we therefore run a few tests utilizing different methods. The experiments showed that the translation search method, explained in the next paragraph, performs best with bilinear methods, blockiness in the edges of the nearest neighbour methods being the main cause of the difference in performance.

Fig. 6. The image after inversion according to the estimated rotation and scale parameters

As suggested in [5], resilience to translation can be achieved utilizing an embedded reference watermark and performing a full correlation search. In our proposed method the embedded grid structure is utilized as such a reference watermark. As a consequence of the periodicity conditions of the grid defined by equations 1 and 2, four periodic pseudorandom vectors can be defined

$$\begin{aligned} &v_1\left(x+q_0N_0\right)=W\left(x,1\right)\\ &v_2\left(x+q_0N_0\right)=W\left(x,N_0-1\right)\\ &v_3\left(y+q_1N_1\right)=W\left(1,y\right)\\ &v_4\left(y+q_1N_1\right)=W\left(N_0-1,y\right);\quad N_0,N_1>1 \end{aligned} \quad . \tag{15}$$

In order to define the edges of the watermarked area of the printed and scanned image, mean removed cross-correlation $c_{vy}^{**(k,l)}(m)$ is calculated

$$c_{vy}^{**(k,l)}(m)=\begin{cases}\sum_{n=0}^{N-|m|-1}\left(v(n+m)-\frac{1}{N}\sum_{i=0}^{N-1}v_i^{k,l}\right)\left(y_n^{**(k,l)}-\frac{1}{N}\sum_{i=0}^{N-1}y_i^{**(k,l)}\right) & m\geq 0\\ c_{vy}^{*}{}^{**(k,l)}(-m) & \end{cases} , \tag{16}$$

where $k=1....\beta_1$ and $l=1....\beta_2$. Edges are evaluated by determining a significant drop in $\max(c_{vy^{(k,l)}})$ over all values of k and l, starting from β_1 and β_2. The method is repeated for every edge of the image, utilizing the vectors in equation 15 and image rotated by angles (0°,90°,180°,270°). The method allows the flexible setting up of the parameters β_1 and β_2, which consequently has a direct effect on the complexity of translation search. In figure 7 is illustrated the watermarked image area extracted from the printed and scanned image utilizing the proposed method.

Fig. 7. The image after translation search

3 Embedding and Extraction of the Multibit Message

As reported in [7], wavelet domain techniques have been considered by several authors for embedding watermarks and for data hiding, and consequently several methods have been proposed. Advantages of wavelet domain are due to the space-frequency localization, multiresolution representation, HVS modeling, linear complexity and adaptivity to the application case, as stated in [7]. However when considering blind message recovery or watermark detection from images that have undergone some form of affine transform, rotation, scale or translation, wavelet domain methods usually fail. Even a very small geometrical distortion [10] can distort the message. Based on our experiments, even a rotation of 0.3 degrees, with no other attacks, will cause significant BER and consequently impede utilization of additive spread spectrum techniques, the basic implementation of which otherwise would be very straightforward. Additionally, when utilizing separate synchronization watermark, probably in different domain than the actual message bits, problems in defining a method how to efficiently use available JND exist. Here, a two step approach is proposed and the validity of the method evaluated.

As a consequence of accurate determination of the affine parameters, we considered the applicability of additive spread spectrum technique in wavelet domain. The message bits are embedded to the image, where grid information has first been embedded, in wavelet domain. 1-level decomposition of the image into sub bands using Haar-wavelets is utilized and the message bits are embedded into the approximation coefficients of wavelet transform utilizing

$$\begin{cases} Y^{**}_{l,f}(n) = Y^{*}_{l,f}(n) + \lambda_2 \cdot \beta \cdot m(k) & , message\ bit = 1 \\ Y^{**}_{l,f}(n) = Y^{*}_{l,f}(n) - \lambda_2 \cdot \beta \cdot m(k) & , message\ bit = 0 \end{cases}, \tag{17}$$

where $\boldsymbol{Y}_{l,f}^{**}(\boldsymbol{n})$ is the watermarked sub band in the l[th] resolution level at f[th] frequency orientation and $\boldsymbol{Y}_{l,f}^{*}(\boldsymbol{n})$ is the corresponding sub band of *Y**. The chip rate for spreading is controlled by the length of the m-sequence, *m(k)*, and the addition or subtraction of the m-sequence is weighted using estimated JND profile and a scaling factor β. λ_2 is an additional scaling factor. Finally, the watermarked image is inverse transformed. The scaling factor β is derived from equations 1-5 in [8]. Decomposition of the JND profile is utilized, as proposed in [8], to derive JND energy weight for the corresponding low resolution sub band. The estimation of the JND profile is calculated for the low resolution band for the image where grid information has been embedded. The validity of the multilevel approach is verified, utilizing peak-signal-to-perceptible-noise ratio (PSPNR) [8], defined as

$$\boldsymbol{PSPNR} = 20 \cdot \log_{10} \frac{255}{\sqrt{\boldsymbol{E}\left\{\left[\left|\boldsymbol{X}(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{Y}^{**}(\boldsymbol{x},\boldsymbol{y})\right| - \boldsymbol{JND}_{\boldsymbol{fb}}(\boldsymbol{x},\boldsymbol{y})\right]^2 \cdot \boldsymbol{\delta}(\boldsymbol{x},\boldsymbol{y})\right\}}}, \tag{18}$$

where

$$\boldsymbol{\delta}(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} 1, & \boldsymbol{if}\ \left|\boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}) - \hat{\boldsymbol{p}}(\boldsymbol{x},\boldsymbol{y})\right| > \boldsymbol{JND}_{\boldsymbol{fb}}^{(1)}(\boldsymbol{x},\boldsymbol{y}) \\ 0, & \boldsymbol{if}\ \left|\boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}) - \hat{\boldsymbol{p}}(\boldsymbol{x},\boldsymbol{y})\right| \le \boldsymbol{JND}_{\boldsymbol{fb}}^{(1)}(\boldsymbol{x},\boldsymbol{y}) \end{cases} \tag{19}$$

and *Y**(x,y)* denotes the reconstructed pixel at (x,y) and $\boldsymbol{JND}_{\boldsymbol{fb}}^{(1)}(\boldsymbol{x},\boldsymbol{y})$ is the original JND profile. The PSPNR gives an estimation of how well the final multiple watermarked image keeps to the original JND. In figure 8, is depicted a multiple watermarked Lena image.

Fig. 8. The original Lena image on the left, the watermarked (α=1.0, λ_1=0.6, β=0.5,λ_2=1.0) Lena image on the right

Message extraction is based on a thresholded correlation receiver. The mean removed cross-correlation between the watermarked image signal and the equalized m-sequence is calculated. The detection scheme is blind and therefore original image is not needed for the message extraction. If we define *y**(n)* as the watermarked vector and *m(n)* as equalized m-sequence samples, the raw cross-correlation values $c_{my}(m)$ are defined as

$$c_{my}(m) = \begin{cases} \sum_{n=0}^{N-|m|-1} \left(m(n) - \frac{1}{N} \sum_{i=0}^{N-1} m_i \right) \left(y_{n+m}^{**} - \frac{1}{N} \sum_{i=0}^{N-1} y_i^{**} \right), m \geq 0 \\ c_{ym}^{*}(-m), m < 0 \end{cases} \quad (20)$$

The output vector has the format of

$$c(m) = c_{my}(m - N), m = 1, 2, ..., 2N - 1. \quad (21)$$

Equalization filter is utilized to improve performance. The equalization filter suppresses the low-frequency components with high energy and emphasizes the high-frequency part of the spectrum in order to obtain a more flat, noise-like spectrum. The filter is a fixed one-dimensional high pass filter d(n) = [-1 2 -1]. Threshold decision block brings the decision regarding the value of the watermark bit. If the cross-correlation value is above the predefined threshold value, then the decision is made for the existence of message bit value 1, otherwise zero is detected. Furthermore, Hamming coding (7,4) is utilized in order to increase reliability of message extraction.

4 Experiments

In the experiments we utilized HP Color LaserJet 4650 PCL 6 printer and EPSON perfection, 4180 photo scanner. An errorcoded message of 112 bits was embedded to the images and reliability of the extraction measured. The effect of different parameter settings, affecting imperceptuality and robustness was experimented. The rotation angle was varied (0-15 degrees) and scanning area was changed. Additionally the effect of JPEG compression on robustness was experimented. Prior to the

Table 1. Robustness of message extraction and perceptual quality

Parameter settings	PSNR	PSPNR	Scaled 25% success ratio	average BER when not a success
α=1.0, λ_1=1.0, β=0.5,λ_2=1.0, γ=0.17, CR=0	34.2	46.7	67%	1.25%
α=1.0, λ_1=0.8, β=0.5,λ_2=1.0, γ=0.13, CR=0	35.9	50.1	75%	3.5%
α=1.0, λ_1=0.7, β=0.5,λ_2=1.0, γ=0.14, CR=0	36.9	51.6	100%	0%
α=1.0, λ_1=0.6, β=0.5,λ_2=1.0, γ=0.15, CR=0	37.9	54.1	83%	1.79%
α=1.0, λ_1=1.0, β=0.5,λ_2=1.0,γ=0.17, CR=80	34.2	46.7	100%	0%
α=1.0, λ_1=0.8, β=0.5,λ_2=1.0,γ=0.13, CR=80	35.9	50.1	83%	17,9%

estimation of the affine transform parameters, the scanned image area was rescaled to 25%. By doing this the computational complexity is reduced significantly. In table 1 is reported the results with different settings, the success ratio indicating the ratio of extraction attempts when the BER=0%. The average BER when extraction was not successful is a function of synchronization error (angle, scale, translation), errors due to the inversion process and parameter settings.

We also experimented more closely the effectiveness of the approach presented with equation 11. The images were watermarked with same parameters, α=1.0, λ_1=1.0, β=0.5, λ_2=1.0. The images were then JPEG compressed with compression ratio of 50 and printed and scanned for message extraction. During extraction the threshold was set to γ=0.09. The results are shown in table 2 and clearly show that the proposed method is effective.

Table 2. The efficiency of utilizing overlapping areas in the analysis of rotation angle and scale

	Scaled 25% success ratio	average BER when not a success
four nonoverlapping areas utilized	0%	7.7%
five overlapping areas utilized	83%	1.78%

Overall, the results showed that with no other attacks than printing-scanning process the quality of the watermarked image is good, even when measured with PSNR. High PSPNR values indicate that two level approach is performing well in respect to the original JND. The results also indicate that when JPEG compression is applied, a stronger grid structure should be embedded. However our tests on utilizing overlapping image areas for analyzing grid structure are still preliminary and we expect improvements on the results in the future. Additionally, as already pointed out, the threshold settings can be made adaptive so we will study more closely the relation of the length of the detected line segments on the accuracy of the estimation of rotation angle and scale. The success ratio of message extraction is at least moderate and more efficient error coding methods would improve results. In addition, the fairly low average BER (<3.5%), when message extraction is not successful, indicates that in most of those cases synchronization is not lost. Therefore more tests will be performed in order to distinguish and analyze the possible source of errors.

5 Conclusions

In this paper we presented a method to embed a multibit message into image utilizing multilevel watermarking principles. We exploited the previous research done on geometrical distortions and watermarking and presented methods how to efficiently embed a periodic watermark and how the estimation of the affine transformation parameters can be done accurately. As the interest in wavelet domain methods is high, specifically considering the problems caused by compression, a wavelet domain method for embedding the actual multibit message was considered. Thus far, we only

considered additive spread spectrum techniques and we realize that the capacity of such techniques is quite low, we therefore will continue on experimenting other kinds of techniques. The results showed that message extraction with a fairly high success ratio is possible even when the watermarked images are compressed before printing and scanning procedure with JPEG compression ratio 50.

References

1. Ching-Yung Lin and Shih-Fu Chang: Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process. Intl. Symp. on Multimedia Information Processing (ISMIP 99), Taipei, Taiwan (1999)
2. O'Ruanaidh, J. and Pun, T.: Rotation, Scale and Translation Invariant Spread Spectrum Digital Image Watermarking. Signal Processing, Vol. 66, No. 3. (1998) 303-317
3. Bas P., Chassery J-M., Macq B.: Geometrically Invariant Watermarking Using Feature Points. IEEE Transactions on Image Processing, Vol. 11. (2002) 1014-1028
4. Pereira, S., Pun, T.: Fast robust template matching for affine resistant image watermarking. International Workshop on Information Hiding. ser. Lecture Notes in Computer Science, Vol. LNCS 1768. Berlin, Germany: Springer-Verlag (1999) 200–210
5. Kutter, M: Watermarking resistant to translation, rotation and scaling. proc. SPIE, Multimedia Systems and Applications, Vol. 3528. Boston, MA (1998) 423-421
6. Deguillaume, F., Voloshynovskiy, S., Pun, T.: Method for the Estimation and Recovering from General Affine Transforms. proc. SPIE, Electronic Imaging 2002, Security and Watermarking of Multimedia Contents IV, Vol. 4675. (2002) 313-322.
7. Meerwald, P., Uhl, A.: A survey of wavelet-domain watermarking algorithms.. Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III, P. W. Wong and E. J. Delp, eds., 4314, SPIE, (San Jose, CA, USA). (2001)
8. Chou, D-H., Li, Y-C.: A Perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile. IEEE transactions on circuits ans systems for video technology, vol. 5, no. 6.,pp.467-476, Dec. 1995
9. Keskinarkaus, A., Cvejic, N., Niskanen, A., Seppänen, T., Sauvola, J. : Improvements on watermarking images with m-sequences. Proc. International Workshop on Digital Watermarking, Seoul, Korea, 333 - 344.2002.
10. Gui Xie, Hong Shen: Rotation-Invariant Log-Polar transform and its Application in Watermarking. WSEAS Transactions on Information Science and Applications, vol.1,no. 5, November 2004.
11. Alvarez-Rodríguez, M, and Pérez-González, F,. Analysis of pilot-based synchronization algorithms for watermarking of still images. Signal Processing: Image Communication, vol. 17, no. 8, pp.:661-633, September 2002.

A New Multi-set Modulation Technique for Increasing Hiding Capacity of Binary Watermark for Print and Scan Processes

C. Culnane, H. Treharne, and A.T.S. Ho

Department of Computing, School of Electronic and Physical Sciences,
University of Surrey, Guildford, Surrey, GU2 7XH
csm1cc@surrey.ac.uk

Abstract. In this paper we propose a multi-set modulation technique to increase the hiding capacity within a binary document image. As part of this technique we propose an Automatic Threshold Calculation and Threshold Buffering, Shifted Space Distribution and Letter Space Compensation technique. The Automatic Threshold Calculation is used to distinguish word spaces from letter spaces. The Threshold Buffering is used to reduce the chance of misinterpretation of spaces during the detection phase, following printing and scanning. The Shifted Space Distribution and Letter Space Compensation techniques robustly embed a watermark into the binary document image. The Automatic Threshold Calculation has been shown to be successful in identifying word spaces for different types of fonts and font sizes. The combination of the Shifted Space Distribution, Letter Space Compensation and Threshold Buffering techniques have been shown to create a watermark that is robust to printing and scanning.

1 Introduction

Several approaches have been proposed which examine the digital watermarking of binary documents. Low and Maxemchuk proposed a method of line and word shifting in [1]. Low et al. [2] compared two methods of line and word shifting for binary text documents whilst Wu et al. [3] examined the full range of binary documents. Ho et al. proposed a method of pixel flipping in [4]. Koch and Zhao [5] proposed a method of embedding data into a binary image based on the percentage of pixels that were white in a block of an image. However, the watermark was not robust to print and scan operations. The potential for digital watermarks is well documented [6], but text documents, more so than other documents, are often printed and transferred from the digital realm to the analog, by way of a printer. The analog copy can then be transferred back into the digital realm with the use of a scanner. There is a need for a watermark that is robust to these print and scan operations. A print and scan resilient system is proposed in [7], but this was for images and not binary document images. One of the notable methods for binary document images was proposed by Zou

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 96–110, 2006.

& Shi [8] in which they identified a way of embedding bits in the lines of text using inter-word space modulation. The fundamental concept of their work is to divide the spaces, between words in a line, into two sets. The total space in these sets is adjusted to create a detectable difference. The difference between the two sets indicates whether a '0' or a '1' is embedded. In [8] they state that their approach is robust to printing the watermarked document, photocopying it ten times and then scanning it. Their approach is limited to providing 1 bit of embedding space per line of text.

Creating a watermarking scheme robust to printing and scanning presents a number of different challenges. Traditionally, in binary digital watermarking, a watermark is generated by flipping pixels to change the image. This is not a suitable method to use in a print and scan system. When a document is printed and then scanned the pixel locations change, so recovering such a watermark is extremely difficult. Furthermore, during the print and scan process noise and distortion is introduced, and the robustness of the watermark must be demonstrated in the presence of these difficulties.

In this paper we present an approach which improves upon the method presented by Zou and Shi [8]. Our contribution is to increase the capacity of the system by introducing a Multi-set Modulated Word Space technique whereby the capacity is not fixed at 1 bit per line but is dynamically calculated based on the content of that line. In order to maintain a good level of robustness a method of Automatic Threshold Calculation is proposed. This is complemented by a Threshold Buffering technique. In order to embed data, taking into account its perceptibility, we propose a Shifted Space Distribution method for distributing space between sets, whilst a Letter Space Compensation technique is used to ensure the line stays the same overall length. The benefit of the proposed approach is to gain a greater capacity whilst still maintaining robustness to print and scan. The greater capacity of the watermark could potentially be used for authenticating binary images and documents.

2 Multi-set Modulated Word Space

In Section 3 and 4 we introduce our Multi-set Modulated Word space technique. Figure 1 provides an overview of our watermark embedding and detection processes based on this technique. Boxes that are common in both processes use the same algorithms. We have reused the horizontal and vertical profile technique from [8] but we have expanded on the data embedding technique and modified the method of set division to allow multiple pairs of sets in one line. Our new Automatic Threshold Calculation and Threshold Buffering techniques makes the embedding and detection processes more adaptive. We have introduced our own geometric distortion correction technique in place of the one in [8] because it is covered by a patent.

Before we introduce the algorithms used in our approach we clarify some terminology related to spaces in a text document. Word spaces can be defined as the white space between adjacent words. Letter spaces can be defined as

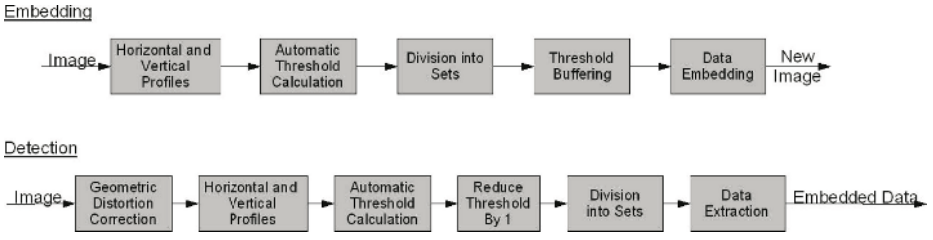


Fig. 1. Flow Diagram of Watermarking Processes

the spaces between adjacent letters within a word. The basic concept takes the principle idea in [8] and expands it further, by having multiple pairs of sets in one line. The line is divided according to how many word spaces should be in each set, so the capacity of a line depends on its content. Word and letters spaces are distinguished with the use of an Automatic Threshold Calculation. Automatic Threshold Calculation is adequate for basic embedding and detection. However, to handle the distortions caused by Print and Scan operations a buffer is created around the *threshold*. This is to avoid misinterpretation of letter and word spaces during detection.

3 Embedding Process

3.1 Horizontal and Vertical Profiles

The first part of our embedding process uses the standard approach of splitting a document into lines of text, and those lines into letters using horizontal and vertical profiles, as seen in [8] and earlier described in [1]. They are graphs of the pixels present in the horizontal or vertical plane. The profiles are calculated by counting the number of black pixels present in the relevant plane.

Figure 2 shows a horizontal profile, which is used to distinguish the separate lines of text within the document. Zero values in the profile distinguish the individual lines. The position and size of lines can be found by analysing the zero values and the groups of sequential positive values.

For each line found using the horizontal profile, a vertical profile is calculated, as can be seen in Figure 3. The same process of counting black pixels is used. This time the groups of black pixels represent letters or words, whilst the zero values are the spaces.

3.2 Automatic Threshold Calculation

After profiling a document and all the spaces have been found a *threshold* is calculated. Our Automatic Threshold Calculation algorithm aims to correctly distinguish between word and letter spaces. Initial attempts to use a static *threshold* were not successful. A static *threshold* makes a watermark more perceptible. This is due to letter spaces being incorrectly classified as word spaces. If this happens,

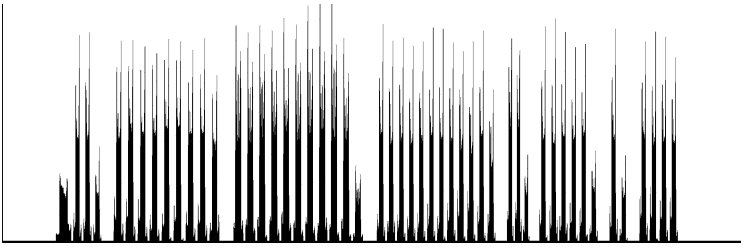


Fig. 2. Horizontal Profile

Fig. 3. Vertical Profile

the embedding process could increase the space between letters in a word. Too much space between letters could be noticeable to the human eye. It is therefore important to correctly classify the spaces found in a line.

Figure 4 shows a graph of the comparison of average word space width for the Times New Roman and Verdana fonts. It is clear that there is a significant difference between the average word space width for the two fonts at the same point size. The difference is also not static and the variation increases the larger the font size. A similar effect was seen when comparing other fonts. This clearly shows the requirement for some form of Automatic Threshold Calculation which can vary according to the content of a particular line.

The Automatic Threshold Calculation is based on the standard deviation of the spaces that are present in a line. Initially, the algorithm used a combination of the mean and the standard deviation. However, the mean was sensitive to the changes that could occur during the print and scan operation and this causes the *threshold* to vary and thus the composition of the sets to change. Hence, it could lead to errors in the calculation of the total space in a set and thus introduce detection errors. The standard deviation is, however, less sensitive to the changes and is therefore more stable.

3.3 Division into Sets

The *threshold*, described in Section 3.2, determines what will be considered a word space and what will be a letter space. Figure 5 shows an extract from a line, approximately a third of the line is shown. The word spaces have been identified as WS_1 through to WS_6. Table 1 shows the pixels width of each of the annotated spaces. The significance of the shading is explained below. A_1 is a letter space that will be of interest after the embedding.

Word spaces can be divided into multiple pairs of sets. A default value of three spaces in each set was chosen. This was chosen because typically lines contain

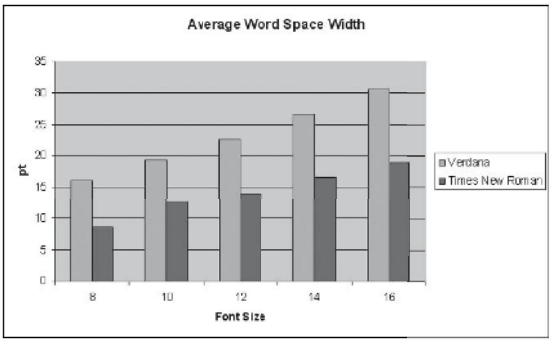


Fig. 4. Average Font Size

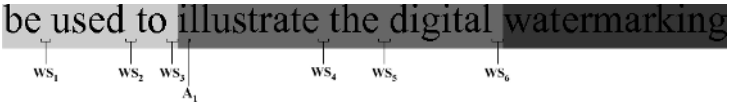


Fig. 5. Set Division

in the region of twelve spaces. As a result, a value of four did not increase the capacity enough, whilst a value of two did not provide the required level of robustness.

Only pairs of sets are created, with any spare word spaces being assigned to a spare group, situated on the right hand side of the line. This is signified by the darkest shading in Figure 5. Each pair of sets provides one bit of possible data embedding. But since we have multiple sets in each line we can embed more than 1 bit per line. Figure 5 shows the last two sets for a line, the complete line contained four sets. Let ϕ refer to a set whilst the subscript indicates the set number.

$$\phi_3 = WS_1 + WS_2 + WS_3 = 41$$
$$\phi_4 = WS_4 + WS_5 + WS_6 = 42$$

Table 1. Table of Space Widths

	Width (px)
WS_1	13
WS_2	13
WS_3	15
WS_4	13
WS_5	15
WS_6	14
A_1	3

Where A_1 is a letter space that is part of ϕ_4. It is not included in the calculation of total set space, but is associated with ϕ_4.

3.4 Threshold Buffering

Initial tests have shown that the minor changes occurring between letter and word spaces during printing and scanning results in spaces being interpreted differently during embedding and detection. To avoid this, and to increase the robustness, a system of creating a buffer around the *threshold* was added.

The goal of this algorithm is to create word spaces that are greater than or equal to the sum of *threshold* and *thresholdBuffer*, and to ensure that letter spaces are strictly less than or equal to the *threshold* minus the *thresholdBuffer*. By expanding the word spaces and contracting the letter spaces the distinction between letter and word spaces becomes clearer. For each set we determine all the word spaces that need expanding. We also determine all the letter spaces that require contracting. Any spare space obtained by contracting the letter spaces is distributed evenly to the word spaces requiring expansion. At this point if all word spaces are greater than or equal to the sum identified above an appropriate buffering has been achieved. (Note that any excess spare space is distributed evenly amongst all word spaces in a set.) If more space is needed to expand the word spaces, further reductions are made to the letter spaces. (Note that no letter spaces are reduced below 1px.) Only letter spaces from within a set are used to create the buffer. Letter spaces from other sets and the spare set are not used. This is to reduce the inter-set dependency and because there is no guarantee that a spare set will exist. In situations where there is not enough spare letter space in a set, the maximum buffer possible, given the available spare space, is created.

3.5 Data Embedding

The process of data embedding is based on the concepts presented in [8]. Data is embedded on a line by line basis within the document by making a detectable difference in the total size of the two sets. The detectable difference that should be present between two sets is referred to as the embedding strength. This is achieved by making changes to the spaces of the individual sets and then we achieve the desired change in total set size, thus creating a detectable difference. Using Figure 5 as an example, recall that $\phi_3 = 41$ and $\phi_4 = 42$. The parameters for the line in Figure 5 are as follows:

- *threshold* = 10
- *thresholdBuffer* = 3
- Embedding Strength (ϵ) = 6

To embed a '1' ϕ_3 must be at least ϵ greater than ϕ_4, conversely to embed a '0', ϕ_4 must be at least ϵ greater than ϕ_3. Clearly this is not the case in the example above and so ϕ_3 and ϕ_4 must be adjusted accordingly as follows:

$$\text{embed '1'} : \phi'_3 - \phi'_4 = \epsilon$$

$$\text{embed '0'} : \phi'_3 - \phi'_4 = -\epsilon$$

where ϕ'_3 and ϕ'_4 indicates the adjusted sets, which are the original sets augmented with half of the embedding strength, assuming that ϕ_3 and ϕ_4 are equal.

It is often the case that the two sets are not equal, as in Figure 5, and so some extra work is needed before we can augment the sets. Since ϕ_4 is one pixel bigger than ϕ_3 this difference must first be eradicated. The difference between them is divided by two and one set is made bigger whilst the other is made smaller by this amount. In this case the difference is only one pixel, which cannot be evenly split between the two. When this occurs the set to be enlarged is assigned the extra pixel (ϕ_3 in this case). It is important that the total length of the line is maintained, in order to reduce perceptibility. As a result we have:

$$\phi'_3 = \phi_3 + 1 + 3$$

$$\phi'_4 = \phi_4 - 1 - 3$$

Shifted Space Distribution. In the above we identified the required changes to the spaces in a set. The Shifted Space Distribution technique makes the necessary changes but is careful to observe the threshold buffering and ensure the minimum perceptibility of the watermark. The process of increasing a space width is easy, since there is no critical limit on increasing a space's width. However, the process of reducing a space width must be carefully controlled. Word spaces must never be reduced below the sum of the *threshold* and *thresholdBuffer*, in this case 13. This is to ensure spaces do not change from word spaces to letter spaces, which could cause errors in the detection of the watermark. This limit on the level of reduction potentially causes a problem of not being able to satisfy the required changes. For example by reducing word spaces WS_5 and WS_6 in ϕ_4 to 13 pixels each we achieve an overall reduction of three pixels for this set. However, a reduction of four pixels is required in order to make the required change.

Letter Space Compensation. Having made the maximum reduction to the word spaces in a set, and if we have not established the required changes the Letter Space Compensation technique identifies letter spaces from that set which can be reduced. This allows us to use the amount of reduction from the letter spaces to increase the other set. In our example this means identifying two pixels in ϕ_4 (A_1 from Figure 6) to be transferred to ϕ_3. You cannot simply add pixels to ϕ_3 and leave ϕ_4 as it is because we must maintain the length of the line. The rule of not reducing letter spaces below 1 pixel is used, as in Threshold Buffering, so that the threshold calculation is not affected during detection. The Letter Space Compensation technique ensures that the embedding strength is maintained across a pair of sets.

Figure 6 shows the annotated image with the data embedded. Table 2 shows the new space widths and the original widths. Note that the word spaces WS_4, WS_5 and WS_6 have not been reduced below 13. Also note that the letter space, A_1 has been reduced from three to one to allow the transfer of the remainder to the operation to increase the size of ϕ_3. Figure 6 provides the following set sizes:

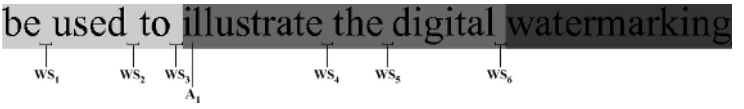


Fig. 6. Extract of line with embedded data

Table 2. Table of New Space Widths Compared with Original

	Width (px)	Original Width (px)
WS_1	15	13
WS_2	15	13
WS_3	16	15
WS_4	13	13
WS_5	13	15
WS_6	13	14
A_1	1	3

$$\phi_3 = 15 + 15 + 16 = 46$$

$$\phi_4 = 13 + 13 + 13 = 39$$

$$\text{where} \quad \epsilon = 46 - 39 = 7$$

The embedding strength (ϵ) is finally calculated to be seven, one more than required. This is due to the rounding operation when splitting a one pixel difference between the two sets.

4 Detection

The main concerns of the detection process, illustrated in Figure 1, are not Threshold Buffering or Shifted Space Distribution but detecting the watermark accurately with increased capacity and dealing with geometric distortion caused by printing and scanning. Therefore, before we create horizontal and vertical profiles, as in Section 3.1, we must deal with geometric distortion of the watermarked image.

4.1 Geometric Distortion

We used a method for correcting geometric distortion based on the use of the horizontal profile and the calculation of the total amount of white space. It is assumed that there are only two points where the total amount of white space is at a maximum:

1. when the document is straight (0°)
2. when the document is rotated (180°)

Assuming that the document will be distorted by less than 1°, in either direction, a horizontal profile can be generated for each rotated value: -1°, 0°, and +1°.

From this horizontal profile the total amount of white space is calculated by summing the parts of the profile that are at zero. Finding the maximum total white space value of the profiles determines the amount of rotation required to correct the distortion. The range can be widened if the image appears to be suffering from a greater degree of rotation. Obviously, rotations performed on high resolution images are computationally intensive. For example, it was found that an A4 page scanned at 600x600 pixels requires one gigabyte of memory to be rotated. However, the principle of the process holds true for smaller images. In our experiments, images are scaled to one half of their original size before they are rotated and profiled. Once the required amount of rotation is found the original is rotated by that amount.

During the initial experimentation described in Section 5.2, it was discovered that the amount of rotation an image suffered during the Print and Scan process could be as little as 0.25°. This small rotation had a detrimental effect on lines with a small font size. As a result a smaller rotation increment was needed, that gradually increased the precision of the rotation. For example, if the maximum amount of white space was found at 1°, the values from 0.0° to 2.0° would be tested at increments of 0.1°. If the maximum amount of white space was found at 0.2°, the values between 0.10° and 0.30° would be tested at increments of 0.01°.

The process of rotating the document to correct rotation can cause a degradation in the quality of the lettering. We noticed that noise, in the form of white pixels, is added to some letters. To counter this a Gaussian blur is applied to the image after the rotation has been completed. This is to mitigate against the chance of letters being split apart due to the distortion correction. If this occurs the horizontal and vertical profiles may incorrectly divide the document.

4.2 Differences and Similarities with Embedding Process

Once the *threshold* is calculated using the Automatic Threshold Calculation algorithm the *threshold* is reduced by 1. The *threshold* is reduced by one during the detection process, to allow for changes in spaces due to noise and distortion. This is possible due to the Threshold Buffering conducted during the embedding process. The value of one was chosen because it allows small distortions in word spaces with the minimum risk of misinterpreting letter spaces. The Threshold Buffering cannot be assumed to have created a buffer of three on every occasion, therefore a reduction of two or three would be more likely to cause misinterpretations. Other than this reduction the same process of dividing the sets is undertaken and creating multiple sets is the same as in Section 3.3.

4.3 Data Extraction

The final part of the detection process is data extraction which is achieved by analysing the total word space differences between each set in each pair of sets

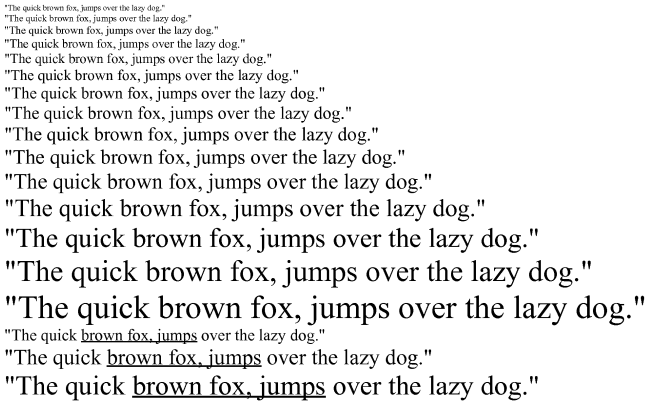


Fig. 7. Extract from a Sample Test Document for Times New Roman

in each line of a document. A '1' bit is present if set A is bigger than set B whilst a '0' bit is present if set B is bigger than set A.

5 Experimentation

This section evaluates the effectiveness of the Automatic Threshold Calculation, Threshold Buffering and Shifted Space Distribution techniques.

5.1 Automatic Threshold Calculation

The Automatic Threshold Calculation was tested using a sample document containing various font types, sizes and styles. This sample document was created in the following fonts: Arial, Arial Narrow, Comic Sans, Courier New, MS Sans Serif, Script, Tahoma, Times New Roman, and Verdana.

Figure 7 shows an extract from a sample test document. One such document was created for all of the above fonts. Each line in the document represents a particular test, with each test being conducted on multiple font sizes. The following tests are contained within the document: font size, partial underlining, partial bold styling, two different fonts in a line, no spaces in a line, two different font sizes in a line and a line of single letter words. The partial underlining test aimed to test the effect of having part of the line underlined. It should be noted that the Automatic Threshold Calculation did not divide the underlined section, it viewed it as a single long word. Tests on no spaces and single letter words in a line were chosen as extreme tests and were not expected to be successful.

Each document was processed using the Automatic Threshold Calculation and the number of word spaces detected was recorded. The documents were also manually analysed to record a benchmark of the correct number of word spaces that should be detected. From these we could analyse the number of errors. A negative comparison indicates that too few word spaces were detected, whilst

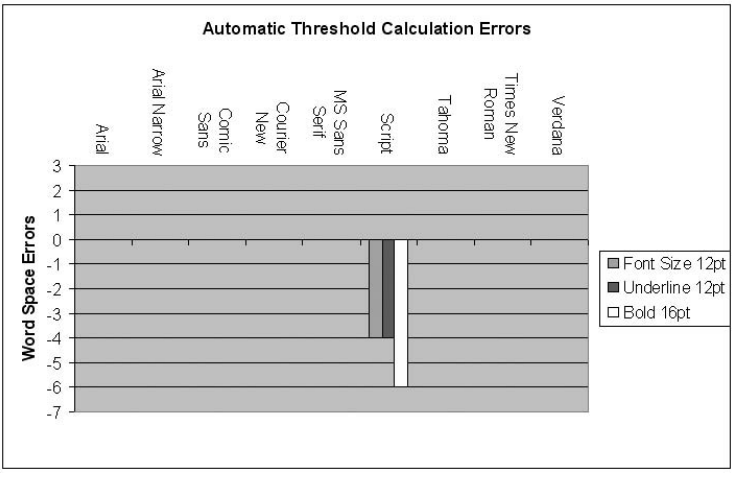


Fig. 8. Threshold Errors for Font Size, Underline, and Bold Test

a positive value indicates too many word spaces were detected. The positive and negative errors are reflected on the y axis as illustrated in Figures 8 and 9, respectively.

Figure 8 shows an extract for the results for font size, underlining and bold styling. We focus on three font types because they reflected the overall results for all font sizing, underlining and bold styling. All the fonts, except Script, handled the styling and sizes of fonts correctly. The Script font is a handwriting style font that joins some letters together. As a result, there are fewer letter spaces. This makes it difficult to distinguish between letter and word spaces. The algorithm is designed so that it is better to detect fewer word spaces at the cost of capacity than to incorrectly detect letter spaces as word spaces at the cost of robustness.

Overall the Automatic Threshold Calculation dealt very well with the challenges of font size and different fonts. We have attempted some extreme tests which stress the algorithm and identify ways in which the algorithm can be improved under these conditions. In most cases it also successfully handled multiple font types in a line and partial styling. Arial and Times New Roman both had single errors, but the rest, other than Script, were successful in handling the multiple font types and partial styling.

Figure 9 shows an extract of the results in which a line is made up of two fonts, two font sizes, no spaces and single letter words. Again these particular results were chosen because they were representative of their class. Most of the standard document fonts, with two different fonts in a line, were successfully handled, with the exception of Courier New. The Script font again had problems in all areas.

It is interesting to note that, all the fonts, except Arial Narrow and Times New Roman had errors with the no spaces test. In practice the algorithm uses the rule that if the standard deviation of all the spaces is below 1.5 then the

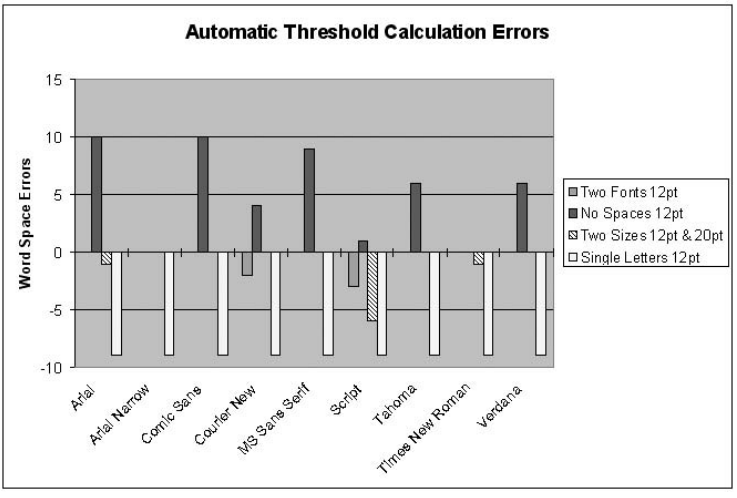


Fig. 9. Threshold Errors for Complex Tests

threshold is set to be the maximum space width, giving rise to no word spaces being found. In practice it is unlikely that such a sentence would appear without a single space.

All the fonts failed the single letter test. The regularity of the spaces means that the standard deviation is always low, resulting in the use of the above rule and thus the inability to recognise word spaces. This results in an undue reduction in capacity. It is not a major concern that the Automatic Threshold Calculation failed this test, since it is highly unlikely that a line would contain just single letters separated by spaces.

5.2 Watermarking

Test Setup. The test document was an A4 page of text. A copy was created for each of the fonts tested in Section 5.1, all having a font size of 12pt. Each was saved as a PDF file and converted to a PNG file at 600dpi. The watermarked documents were printed on a HP PSC 2110 in Normal mode. The printed documents were scanned on the same HP PSC 2110 at various resolutions in Black & White mode with automatic straighten and colour adjustment switched off.

Experimentation. Table 3 contains the watermarks and their respective ID's. Each watermark was embedded in a copy of each document in each font, giving

Table 3. Watermarks and ID's

Watermark ID	Watermark
A	WATERMARKED!
B	THE EXAMPLE!
C	ZZZZZZZZZZZZ

Table 4. Print & Scan Results

			Bit Error Rate								
			150dpi			300dpi			600dpi		
Font	**Capacity (bits)**	**Lines**	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
Arial	88	50	9	13	19	5	5	4	1	0	1
Arial Narrow	99	48	12	14	57	2	0	2	2	3	1
Comic Sans	69	41	27	19	28	1	21	1	0	0	0
Courier New	52	50	2	3	4	2	2	0	2	0	2
MS Sans Serif	90	51	16	10	16	1	3	5	0	1	0
Script	72	46	29	39	31	38	35	35	30	35	35
Tahoma	82	47	6	6	6	0	3	3	2	1	1
Times New Roman	90	49	45	10	28	1	13	34	0	3	1
Verdana	68	47	22	20	22	0	0	7	0	0	0

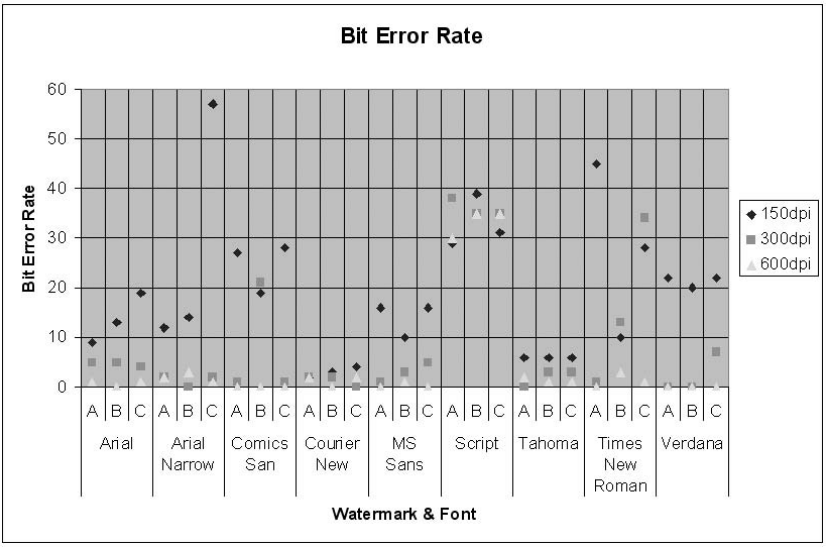


Fig. 10. Bit Error Rates

a total of 27 documents. The data was embedded as a binary string, using 8-bits per character. Where a document did not have the capacity to fit the entire watermark, the maximum to the nearest letter was embedded.

Table 4 shows the bit error rate (BER) results from the documents being scanned at 150, 300, and 600 dpi. These results are illustrated in Figure 10. The BER was calculated by comparing the binary string detected with the one embedded. There were occurrences of fewer sets being detected during the detection phase from the embedding phase. In these situations, if it was clear where the set had been lost, the detected string was padded with the inverse bit of the original. This would result in a lost set being counted as a bit error, without having a significant impact on all the remaining bits in the string, that would otherwise have been shifted one place to the left.

The results show that it is possible to achieve a zero bit error result. They also demonstrate that in most cases 150dpi is not a high enough resolution by which to scan the document. The best results were obtained at 600dpi. The Script font did not perform well, although this was to be expected since it performed badly in the Automatic Threshold Calculation tests. The Verdana font performed well with five out of the six test above 150dpi resulting in no bit errors. Courier New provided the most consistent results, and interestingly coped well at 150dpi. The Tahoma font also performed better than most at 150dpi and consistently over the other resolutions.

Table 4 also shows the number of lines of text in each document. This demonstrates the increase in capacity, for example Verdana and Comic Sans provided 1.4 and 1.7 bits of capacity per line, respectively. This is in contrast to just 1 bit per line in [8].

Overall the results demonstrate that the principle of the system works. Further experiments are given in [9]. With some further improvements it may be possible achieve even more zero bit errors.

6 Conclusion

The proposed method has been shown to provide a greater capacity whilst still being robust to print and scan. The Automatic Threshold Calculation has shown to be useful in handling multiple fonts and different font sizes. Both the Tahoma and Comic Sans fonts correctly classified spaces with zero errors, except in the extreme tests. The Shifted Space Distribution has produced results which appear imperceptible to the human eye. The Letter Space Compensation technique has improved the robustness of the watermark. Without this technique we would not have been able to maintain the embedding strength. At 600dpi both the Comic Sans and Verdana font were able to detect the watermark with a zero BER.

The capacity in [8] is restricted to embedding one bit per line. In our approach it is dependent on the content of each line and can vary from one line to the next. For example, a line in a smaller sized font will have more word spaces, and thus a higher capacity. The capacity can be varied by adjusting how many word spaces should be present in each set. The fewer the number of word spaces the greater the capacity, but the watermark is less robust. A direct comparison with [8] was not possible due to the lack of algorithmic detail.

7 Future Work

Our immediate future plans involve improving the current algorithm and dealing with noise which can result from scanning documents. We will also need to consider the issues related to achieving secrecy.

Multi-set Modulated Word Space. There are a number of possible improvements to the method for embedding data and the use of the space. The Threshold Buffering technique requires further work to identify a bound below

the *threshold* value in the extreme cases identified in Section 5.1. Using the space in the spare group may also provide a way of increasing the robustness of the watermark and eliminating the problems of losing sets between embedding and detection.

Noise Removal. The current noise removal procedure is done manually. The method described by Zou and Shi in [8] removed isolated black pixels. This was not implemented because the noise we saw was greater than single black pixels. Our initial experiments confirm that it may be possible to remove noise from around the outside of the text using horizontal and vertical profiling. Further research is needed to remove noise from between words or lines.

Acknowledgements. We would like to thank the reviewers for their helpful comments.

References

1. S. H. Low, N. F. Maxemchuk, and A. P. Lapone. Document identification for copyright protection using centroid detection. *IEEE Transactions on Communication*, 46(3):372–383, 1998.
2. S. H. Low and N. F. Maxemchuk. Performance comparison of two text marking methods. *IEEE Journal on Special Areas in Communications*, 16(4):561–572, 1998.
3. M. Wu, E. Tang, and B. Liu. Data hiding in digital binary images. In *International Conference on Multimedia and Expositions*, volume 1, pages 393–396, Jul 31 - Aug 2 2000.
4. A.T.S. Ho, N. B. Puhan, A. Makur, P. Marziliano, and Y. L. Guan. Imperceptible data embedding in sharply-contrasted binary images. In *ICARCV*, volume 2, pages 958 – 963, Dec 2004.
5. J. Zhao and E. Koch. Embedding robust labels into images for copyright protection. In *International Congress on Intellectual Property Rights for Specialised Information, Knowledge and New Technologies*, Vienna, Austria, 21–25 1995.
6. M. L. Miller I. J. Cox and J. A. Bloom. *Digital watermarking : principles and practice*. Morgan Kaufmann, 2001.
7. A.T.S. Ho and F. Shu. A print-and-scan resilient digital watermark for card authentication. In *ICICS-PCM*, volume 2, pages 1149 – 1152, Singapore, Dec 2003.
8. D. Zou and Y. Q. Shi. Formatted text document data hiding robust to printing, copying and scanning. In *IEEE International Symposium on Circuits and Systems (ISCAS05)*, Kobe, Japan, May 2005.
9. C. Culnane. Digital watermarking of binary text documents, robust to print and scan. Master's thesis, University of Surrey, 2006.

A Novel Multibit Watermarking Scheme Combining Spread Spectrum and Quantization

Xinshan Zhu[1], Zhi Tang[1], and Liesen Yang[1,2]

[1] Institute of Computer Science & Technology of Peking University, Beijing 100871, China
{zhuxinshan, tangzhi, yangliesen}@icst.pku.edu.cn
[2] National Key Laboratory of Text Processing Technology, Peking University, Beijing 100871, China

Abstract. This paper presents a new multibit watermarking method. The method uses multiple orthonormalized watermark patterns of the same size as the host signal. In particular, the elements of each watermark pattern follow independent normal distribution. Each bit of the transmitted message is hidden using the dither quantizers to modify the projection of the host signal onto its corresponding watermark pattern. As a result, every hidden bit is spread over all elements of the host data and the extracting procedure is blind. Meanwhile, we consider how to choose a suitable quantization step size under the given distortion constraint. It is also proved mathematically that the upper bound of the bit error probability of our method is equal to one of the spread-transform dithered modulation (STDM) under the same situations. Experimental results show our scheme performs better than STDM in terms of bit error probability.

1 Introduction

Digital watermarking is now one of the active research topics in the multimedia area. The goal is to conceal auxiliary information within a host digital signal. This hidden information should be detectable even if the watermarked signal is distorted (to some extent).

Over the last decade, a variety of watermarking algorithms have been proposed in the literature [1]. One class of the proposed methods are based on spread spectrum (SS) modulation technique [2], which embeds information by linearly combining the host signal with a small pseudo-noise signal modulated by the the embedded signal. SS manifests satisfied robustness to interfering noise and lossy compression. However, SS doesn't possess the host interference cancellation and blind detection properties [3]. Presently, the host interference cancellation methods, such as quantization index modulation (QIM) [4], scalar Costa scheme (SCS) [5], have received considerable attention. In QIM and SCS, the information is embedded using dither quantization. They are both developed based on the strong information-theoretic analysis, so are able to embed a multibit watermark message. Spread-transform dither modulation (STDM) [6] is a special case of

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 111–122, 2006.

QIM, which effectively combines quantization-based schemes with the diversity afforded by spread-spectrum methods. However, only one spread vector is used in STDM and every bit of the hidden message is spread over one block of the host signal. More recently, some methods are proposed to improve the robustness of QIM against amplitude scalings in [7,8].

This paper presents a novel multibit watermarking scheme, which modifies the host signal using quantization-based embedder along multiple orthonormalized directions. The remainder of this paper is structured as follows: Section 2 describes our method involving the embedder and detector in details. Section 3 discusses how to choose a suitable quantization step size for a given acceptable distortion constraint, and two specific strategies are presented. The bit error probability of detector is analyzed mathematically in Section 4 and a serial of tests are done to evaluate our method in Section 5. Finally, Section 6 concludes the paper.

2 The Proposed Method

The study is based on a usual watermarking model with side information shown in Fig. 1. The hidden watermark message m and the host signal $\boldsymbol{s} \in \mathbb{R}^N$ are input into the watermark embedder, which outputs an appreciate watermark signal $\boldsymbol{w} \in \mathbb{R}^N$. The host signal could be a vector of pixel values, Discrete Cosine Transform (DCT) coefficients or any other transform domain coefficients from a host content. Then $\boldsymbol{w}$ is added to it to produce the watermarked signal $\boldsymbol{x} \in \mathbb{R}^N$, that is

$$\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{w}. \tag{1}$$

Next, the watermarked signal $\boldsymbol{x}$ might undergo a number of distortions that are modelled as an unknown noise source, $\boldsymbol{v}$. Finally, the watermark detector receives a distorted, watermarked signal, $\boldsymbol{y}$, i.e.,

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{v}, \tag{2}$$

and decodes a message $\widehat{m}$. Both, watermark embedding and detecting, depend on the cryptographic key K based on security consideration. In what follows, attention is restricted to the design of embedder and detector.

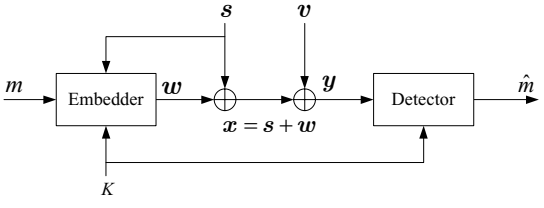


Fig. 1. The general watermarking model with side information

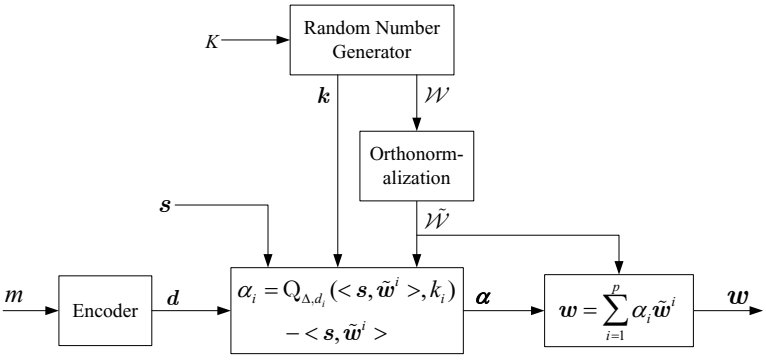


Fig. 2. The construction of embedder in our method

2.1 Embedder

The construction of embedder in our method is shown in Fig. 2, and we explain it in details as follows. Assume that $\boldsymbol{b}$ is the binary representation of m. $\boldsymbol{b}$ is encoded into a sequence of watermark letters $\boldsymbol{d}$ of length p with $d_i \in \mathcal{D}$, where $\mathcal{D}$ can be binary $\mathcal{D} \in \{0, 1\}$ or multilevel $\mathcal{D} \in \{1, 2, \ldots, D\}$ with $D = |\mathcal{D}|$. Generally speaking, we have $p < N$.

A set of p random watermark patterns $\mathcal{W} = \{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^p\}$ are generated by a random number generator (RNG) initialized with the key K. Each pattern corresponds to a watermark letter of the sequence $\boldsymbol{d}$. In particular, the watermark pattern is the same size as the host signal $\boldsymbol{s}$ and each element of it follows the independent standard normal distribution $N(0, 1)$. The watermark patterns are then orthonormalized with the Gram-Schmidt algorithm [9] to obtain $\widetilde{\mathcal{W}} = \{\widetilde{\boldsymbol{w}}^1, \widetilde{\boldsymbol{w}}^2, \ldots, \widetilde{\boldsymbol{w}}^p\}$ with

$$< \widetilde{\boldsymbol{w}}^i, \widetilde{\boldsymbol{w}}^j >= \delta(i - j), \quad \forall i, j, 1 \leq i \leq p, 1 \leq j \leq p, \tag{3}$$

where $< \cdot, \cdot >$ denotes the inner product operator and the Dirac delta function, $\delta(\cdot)$, is defined as

$$\delta(u) = \begin{cases} 1, & u = 0, \\ 0, & u \neq 0. \end{cases}$$

Without loss of generality, we assume that we want to hide only one binary digit of information, i.e., $\mathcal{D} \in \{0, 1\}$. Each watermark letter is embedded into the host signal $\boldsymbol{s}$ through modifying the projection of $\boldsymbol{s}$ onto its corresponding orthonormalized watermark pattern. For this purpose, two dithered quantizers [4,6] with step size Δ are constructed as

$$Q_{\Delta,d}(u, k) = round(\frac{u - (k + d/D)\Delta}{\Delta})\Delta + (k + \frac{d}{D})\Delta, \quad d \in \{0, 1\}, \tag{4}$$

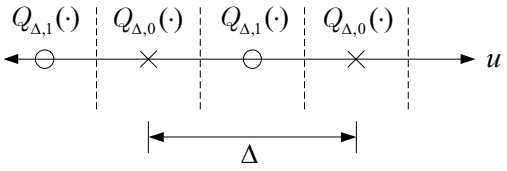


Fig. 3. Centroids and decision regions. The centroids for the quantizer $Q_{\Delta,0}$ and $Q_{\Delta,1}$ are respectively marked with $\times$'s and $\circ$'s.

where the variable k with range $[-0.5, 0.5)$ is key-dependent so as to introduce an additional degree of uncertainty, and function $round(\cdot)$ denotes rounding value to the nearest integer. Fig. 3. illustrates these two quantizers with decoder decision regions. The projection $< \boldsymbol{s}, \widetilde{\boldsymbol{w}}^i >$ is quantized using the chosen quantizer according to the ith watermark letter d_i. As a result, we obtain a quantization error sequence $\boldsymbol{\alpha}$ with

$$\alpha_i = Q_{\Delta,d_i}(< \boldsymbol{s}, \widetilde{\boldsymbol{w}}^i >, k_i) - < \boldsymbol{s}, \widetilde{\boldsymbol{w}}^i >, \tag{5}$$

where the sequence $\boldsymbol{k}$ is generated from the watermark key K, whose elements follow independent uniform distribution between $[-0.5, 0.5)$. Thus, the watermarked signal $\boldsymbol{x}$ is obtained as

$$\boldsymbol{x} = \boldsymbol{s} + \sum_{i=1}^{p} \alpha_i \widetilde{\boldsymbol{w}}^i. \tag{6}$$

2.2 Detector

The construction of detector in our method is illustrated in Fig. 4. First, the set of orthonormalized watermark patterns $\widetilde{\mathcal{W}} = \{\widetilde{\boldsymbol{w}}^1, \widetilde{\boldsymbol{w}}^2, \ldots, \widetilde{\boldsymbol{w}}^p\}$ are generated from the watermark key K as does embedder. The hidden watermark letter sequence is determined according to the rule

$$\widehat{d}_i = arg \min_{d \in \mathcal{D}} | < \boldsymbol{y}, \widetilde{\boldsymbol{w}}^i > - Q_{\Delta,d}(< \boldsymbol{y}, \widetilde{\boldsymbol{w}}^i >, k_i)|. \tag{7}$$

At last, the extracted sequence $\widehat{\boldsymbol{d}}$ is decoded into the message $\widehat{m}$.

The proposed method explores the idea of SS watermarking [2,10], but modifies the form of watermark detector (comparing detection value with a predetermined threshold to decide if the watermark exists or not in SS watermarking). And the use of quantization-based watermark embedder makes it possess the host interference cancellation and blind detection properties as QIM. Additionally, our method is also different from the traditional STDM [6], a special case of QIM. The latter one uses only one spreading vector, however, multiple spreading vectors (watermark patterns) are applied in our method and every hidden bit are spread over all elements of the host data.

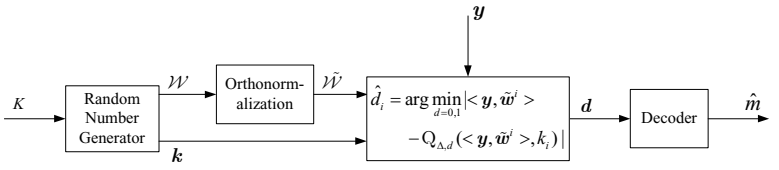


Fig. 4. The construction of detector in our method

3 The Choice of Quantization Step Size

In this section, we will analyze how to choose a suitable quantization step size Δ for a given acceptable distortion constraint. As the previous literatures, the document-to-watermark ratio (DWR) is introduced to measure quality loss of the host signal due to watermark embedding, which is defined as

$$DWR = \frac{\|\boldsymbol{s}\|^2}{\|\boldsymbol{w}\|^2}, \tag{8}$$

where $\|\cdot\|$ stands for Euclidean (i.e., ℓ_2) norm.

Under some given DWR, the step size Δ is chosen so that

$$\|\boldsymbol{w}\|^2 \leq \frac{\|\boldsymbol{s}\|^2}{DWR} \tag{9}$$

From Equation (6), we can write

$$\boldsymbol{w} = \sum_{i=1}^{p} \alpha_i \widetilde{\boldsymbol{w}}^i \tag{10}$$

Thus, according to Equation (3), we obtain

$$\|\boldsymbol{w}\|^2 = \sum_{i=1}^{p} \alpha_i^2 \tag{11}$$

Taking into consideration the fact that $|\alpha_i| \leq \frac{\Delta}{2} (1 \leq i \leq p)$ (see Equation (4) and (5)), a conservative choice strategy of Δ is expressed by

$$\sum_{i=1}^{p} (\frac{\Delta}{2})^2 \leq \frac{\|\boldsymbol{s}\|^2}{DWR}. \tag{12}$$

Inequality (12) can be reduced to

$$\Delta \leq \frac{1}{r} \|\boldsymbol{s}\|, \tag{13}$$

where the ratio r is defined as $r = \sqrt{p \cdot DWR}/2$. Obviously, the distortion constraint (9) must be met under the situation in (13). Note that Inequality (13) gives a very simple choice strategy of Δ.

Further, it is shown [11] that the quantization error α_i is uniformly distributed over the interval $[-\Delta/2, \Delta/2]$. Thereby, it is immediate to write the embedding distortion in (11) as

$$E[\|\boldsymbol{w}\|^2] = \sum_{i=1}^{p} E[\alpha_i^2] = \frac{p\Delta^2}{12}, \tag{14}$$

where $E[\cdot]$ denotes the expectation operator. When substituting $\|\boldsymbol{w}\|^2$ by its expectation $E[\|\boldsymbol{w}\|^2]$, Inequality (9) can be simplified as

$$\Delta \leq \frac{3}{r}\|\boldsymbol{s}\|. \tag{15}$$

Inequality (15) gives another strategy to choose step size Δ in a statistical sense.

4 Analysis of Bit Error Probability

From the detector decision regions illustrated in Fig. 3, it is guaranteed to not make an error when extracting each watermark letter as long as

$$|<\boldsymbol{v}, \widetilde{\boldsymbol{w}}>| < \frac{\Delta}{4}. \tag{16}$$

where $\widetilde{\boldsymbol{w}} \in \widetilde{\mathcal{W}}$. However, it is also possible that no error is made while the constraint (16) isn't met. Therefore, we have

$$P_e \leq \overline{P} = 1 - P(|<\boldsymbol{v}, \widetilde{\boldsymbol{w}}>| < \frac{\Delta}{4}) = P(|<\boldsymbol{v}, \widetilde{\boldsymbol{w}}>| \geq \frac{\Delta}{4}). \tag{17}$$

where P_e denotes the bit error probability and $\overline{P}$ is an upper bound of it. For simplicity, we just analyze how to calculate $\overline{P}$.

Assume that $\boldsymbol{v}$ is independent of $\widetilde{\boldsymbol{w}}$. The assumption is reasonable, because the generation of $\widetilde{\mathcal{W}}$ depends on the cryptographic key K and the attacker had no access to it. The distribution on $<\boldsymbol{v}, \widetilde{\boldsymbol{w}}>$ may be computed by first writing it as $\sum_{i=1}^{N} v_i \widetilde{w}_i$, where v_i is a constant. Since $\widetilde{\boldsymbol{w}}$ is derived through orthonormalizing the watermark patterns $\mathcal{W}$, each element of it, $\widetilde{w}_i$, still follows independent normal distribution with zero mean and variance $\sigma_{\widetilde{w}}^2$. Considering that $\sum_{i=1}^{N} \widetilde{w}_i^2 = 1$, it is easy to obtain $\sigma_{\widetilde{w}}^2 = 1/N$. Using the well-known formula for the distribution of a linear combination of variables that are independent and normally distributed, $<\boldsymbol{v}, \widetilde{\boldsymbol{w}}>$ will be distributed according to

$$N(0, \sigma_{\widetilde{w}}^2 \sum_{i=1}^{N} v_i^2) = N(0, \frac{\|\boldsymbol{v}\|^2}{N}). \tag{18}$$

Thus, $\overline{P}$ in (17) can be expressed as

$$\overline{P} = erfc(\frac{\sqrt{N}\Delta}{4\sqrt{2}\|\boldsymbol{v}\|}). \tag{19}$$

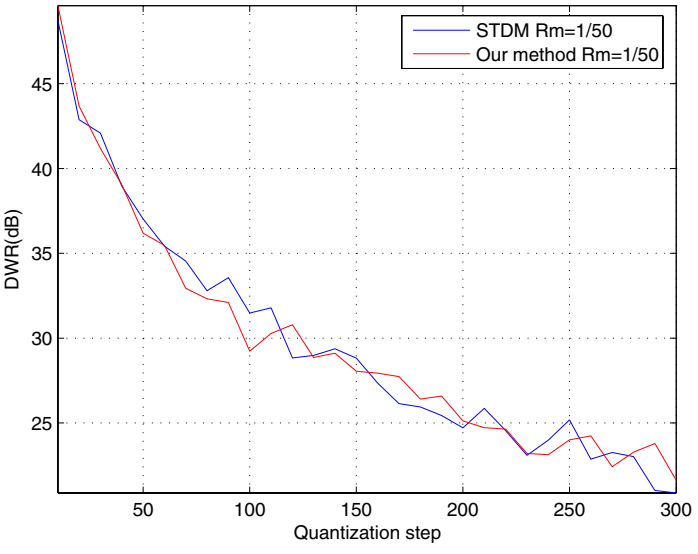


Fig. 5. The embedding distortion measured by DWR as a function of quantization step

where $erfc(\cdot)$ denotes the complementary error function defined as

$$erfc(u) = \frac{2}{\sqrt{\pi}} \int_u^{\infty} e^{-t^2} dt$$

Further, Equation (19) is transformed into Equation (20) as

$$\overline{P} = erfc(\frac{1}{\sqrt{2}} \sqrt{\frac{N\Delta^2 WNR}{16\|\boldsymbol{w}\|^2}}). \tag{20}$$

where WNR denotes the Watermark-to-Noise Ratio, which is defined as $WNR = \|\boldsymbol{w}\|^2/\|\boldsymbol{v}\|^2$. Substituting $\|\boldsymbol{w}\|^2$ by its expectation $E[\|\boldsymbol{w}\|^2]$ in (14), Equation (20) can be written as

$$\overline{P} \simeq erfc(\frac{1}{\sqrt{2}} \sqrt{\frac{3WNR}{4R_m}}). \tag{21}$$

where the rate R_m, is defined as $R_m = p/N$. Equation (21) demonstrates that the upper bound of the bit error probability of our method is equal to one of STDM [3] for the same R_m and WNR.

5 Experimental Results

In this section, some significant experimental results will be presented to illustrate the performance of our method. A total of 10^6 host signals of size 1200 are exploited in all tests, whose elements are independently drawn from a uniform

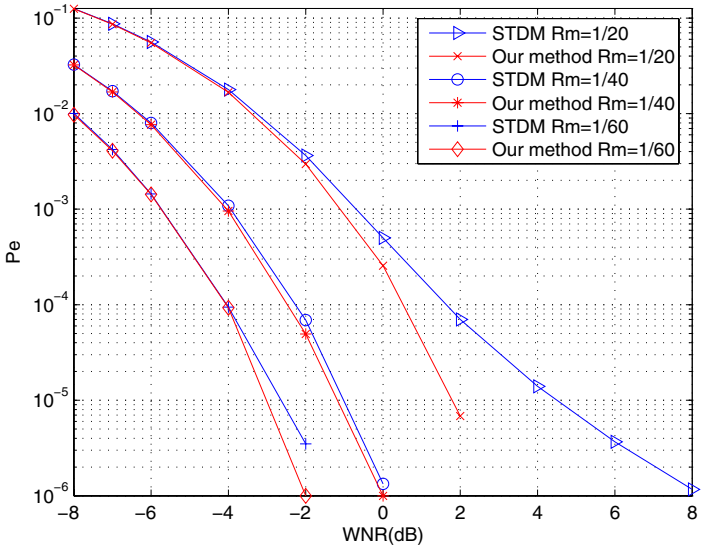


Fig. 6. The bit error probability versus WNR for different values of R_m: Gaussian noise, $\Delta = 200$

distribution over $[0, 255]$ and are rounded to the nearest integers. The uncoded binary STDM [6] are also tested in the host data set for comparison. The implementation of STDM refers to the code available at [12]. Here, we are only interested in the relative performance of different schemes, so it is sufficient to consider the uncoded case. The STDM is adopted, because it represents a popular types of watermarking methods, and is similar to our method in watermark embedding and detecting.

First, watermarking embedding is carried out with a set of quantization steps from 10 to 300 and the resulting quality loss of the host signals is evaluated. The DWR in decibels (dB), i.e., $DWR(dB) = 10\log_{10}(DWR)$, is used to measure the quality of the watermarked signals. As shown in Fig. 5, the quality of the watermarked signals decreases for increasing value of quantization step. Principally, the embedding distortion resulted from the proposed method is almost equal to that caused by STDM for the same quantization step. However, if the chosen quantization step exceeds 120, our method can induce relatively less distortion than STDM.

Then, the bit error probability is tested for different embedding rate R_m. The quantization step, Δ is set to 200. According to Equation (11) and the definition of DWR, we can derive that the DWR increases as the value of R_m decreases for the fixing quantization step size. Gaussian noise is used as the channel distortion in the test, which is measured by the WNR in decibels (dB) defined as $WNR = 10\log_{10}(WNR)$. Fig. 6 is a plot of the bit error probability versus the WNR for values of R_m ranging from 1/60 to 1/20. On each curve

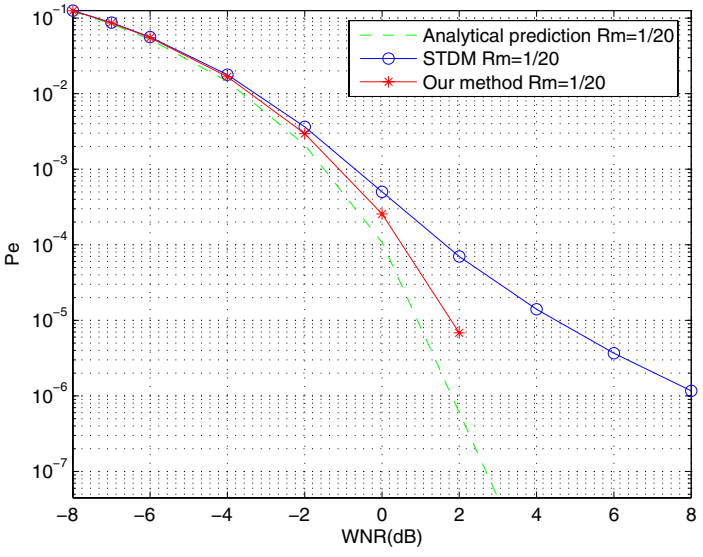
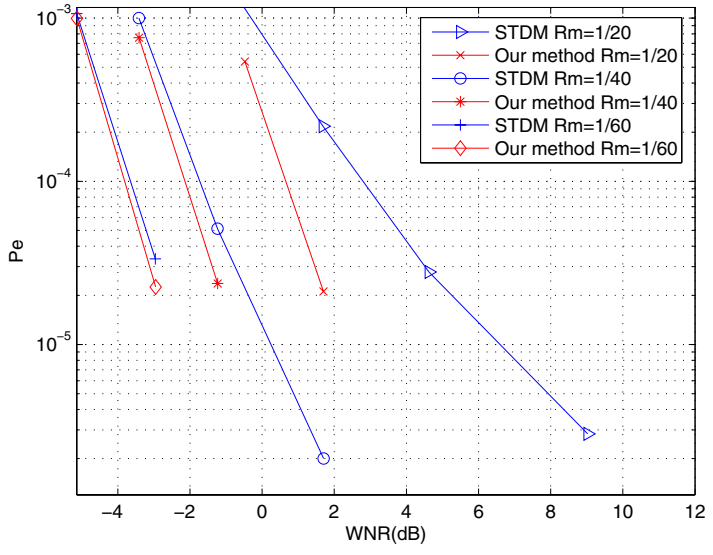


Fig. 7. The measured and predicted values of the bit error probability versus WNR: Gaussian noise, $R_m = 1/20$, $\Delta = 200$

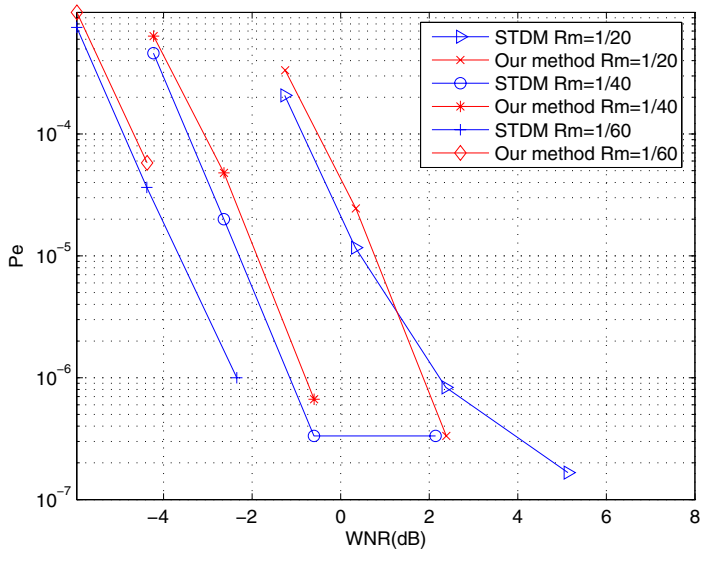
of Fig. 6 ten test points should be marked, however, some points with higher values of WNR are not marked due to their corresponding bit error probability being zero. It can be observed that the bit error probability decreases for these two methods for decreasing value of R_m. Comparing with STDM, our method achieves a lower bit error probability for the same R_m and WNR. Moreover, the advantage becomes more significant as WNR increases, in particular, the bit error probability of zero is obtained more quickly by our method than by STDM.

Further, we calculated the bit error probabilities using (21) for different WNRs and $R_m = 1/20$, which are depicted in Fig. 7 together with the measured results. We found that the measure results from our method is closer to the analytical ones than STDM, although (21) was derived earlier from STDM. Especially when $WNR \geq 4dB$, the bit error probability of our method decreases to $P_e = 0$, but one of STDM decreases smoothly as WNR increases so that the gap to the analytical one becomes larger and larger. In addition, the measured bit error probabilities are larger than the predicted ones, which is not consistent with our analysis in Section 4. That might be caused by the effects of rounding and clipping.

It is well known that the conventional dither modulation based schemes are largely vulnerable to amplitude scalings. Thus, it is necessary to evaluate the performance of our method with this kind of attack. Other test situations are same as the above one including of the choice of Δ and R_m. The amplitude scaling attacks are performed with the gain factor, β from 0.91 to 1.11. Fig. 8(a) depicts the probability of bit error in the case $\beta < 1$, and Fig. 8(b) in the case $\beta > 1$. In Fig. 8(a), five test points should be marked on each curve, and six



(a)



(b)

Fig. 8. The bit error probability versus WNR for different values of R_m: amplitude scaling, $\Delta = 200$. a) $\beta < 1$. (b) $\beta > 1$.

test ones on each curve of Fig. 8(b). However, some points with higher values of WNR are not marked due to their corresponding bit error probability being zero. It can be seen that amplitude scaling attack affects the performance of our

method and STDM much more than Gaussian noise attack. It is interesting to note that our method performs better than STDM in the case $\beta < 1$, but does worse while $\beta > 1$ and WNR is lower. In the second case, as WNR increases, the bit error probability of our method falls off sharply and reaches the value of zero earlier than that of STDM.

6 Conclusion

A new novel multibit watermarking scheme has been proposed in this paper. The approach combines SS and quantization. Unlike the traditional STDM, multiple orthonormalized watermark patterns serve as the spreading vectors, and the size of each pattern is same as the host signal. Each bit of the transmitted message is embedded using two dither quantizers to modify the projection of the host signal onto its corresponding watermark pattern, so that every hidden bit is able to be spread over the whole host signal. In particular, the normally distributed random number sequences are adopted as watermark patterns, which is convenient to analyze the performance of watermark detector. Meanwhile, we present two strategies to adaptively choose a suitable quantization step under the given distortion constraint. Moreover, the upper bound of the bit error probability of detector is derived, which is equal to one of STDM. Experimental results demonstrate that the proposed method can achieve a lower probability of bit error than STDM with respect to Gaussian noise and amplitude scaling attacks.

Note that the paper only developed a rudimentary implementation of the proposed method. Future work should focus on the theoretical analysis of the performance involving transparency, capacity and robustness, etc. as well as the practical implementation with multimedia signals.

Acknowledgments. The authors would like to thank the anonymous reviewers for their detailed comments that improved both the editorial and technical quality of this paper substantially.

References

1. P. Moulin, R. Koetter: Data-hiding codes. Proceedings IEEE. **Vol. 93, no. 12** (Dec. 2005) 2083–2127
2. I. J. Cox and J. Kilian, F. T. Leighton, T. Shamoon: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing. **Vol. 6, no. 12** (Dec. 1997) 1673–1687
3. B. Chen, G. W. Wornell: Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory. **Vol. 47, no. 4** (May 2001) 1423–1443
4. Brian Chen, Gregory W. Wornell: Quantization index modulation methods for digital watermarking and information embedding of multimedia. Journal of VLSI Signal Processing. **Vol. 27** (2001) 7–33

5. J. J. Eggers, R. Bauml, R. Tzschoppe, B. Girod: Scalar costa scheme for information embedding. IEEE Transactions on Signal Processing. **Vol. 51, no. 4** (April 2003) 1003–1019
6. B. Chen, G. W. Wornell: Achievable performance of digital watermarking systems. IEEE Int. Conf. on Multimedia Computing and Systems. **Vol. 1** (1999) 13–18
7. Fabricio Ourique, Vinicius Licks, Ramiro Jordan Fernando, Perez-Gonzalez: Angle qim: a novel watermark embedding scheme robust against amplitude scaling distortions. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. **Vol. 2** (March 2005) 797–800
8. Fernando Perez-Gonzalez, Carlos Mosquera, Mauro Barni, Andrea Abrardo: Rational dither modulation: A high-rate data-hiding method invariant to gain attacks. IEEE Transactions on Signal Processing. **Vol. 53, no. 10** (October 2005) 3960–3975
9. G. Strang: Linear algebra and its applicarions. Harcourt Brace Jovanivich College Publishers. (1998)
10. Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom: Digital watermarking. San Francisco: Academic Press. (2002)
11. N. S. Jayant, P. Noll: Digital coding of waveforms: principles and applications to speech and video. New Jersey: Prentice-Hall, Englewood Cliffs. (1984)
12. Peter Meewald: C source code. http://www.ganesh.org/˜ pmeerw/c_source/.

Wavelet Analysis Based Blind Watermarking for 3-D Surface Meshes*

Min-Su Kim[1,2], Jae-Won Cho[1,2], Rémy Prost[1], and Ho-Youl Jung[2,**]

[1] CREATIS, INSA-Lyon, CNRS UMR 5515, INSERM U630, 69100, Villeurbanne, France
{kim, cho, prost}@creatis.insa-lyon.fr
[2] MSP Lab., Yeungnam Univ., 214-1 Dae-dong, Gyeongsan-si, 712-749, Gyeongsangbuk-do, Korea
Tel.: +82538103545, Fax: +82538104742
hoyoul@yu.ac.kr

Abstract. As most previous *wavelet analysis* based *3-D mesh watermarking* methods embed the watermark information into wavelet coefficients arranged in a certain order, they have not been used as *blind schemes* since the connectivity information must be exactly known in the watermark extraction process. In this paper, we propose a blind watermarking method based on wavelet analysis for 3-D mesh model. Two new techniques are introduced. One is to exploit the *statistical features* of *scale coefficients* on an approximation (low resolution) level for watermark embedding. Another is to extract the hidden watermark, not from the same resolution level as used in embedding process, but directly from the spatial domain. As the proposed watermark detection does not require the wavelet analysis, any pre-processing such as registration and re-sampling, is not needed. These techniques allow to detect the watermark without referring to the original meshes. In addition, the proposed are applicable directly to *irregular meshes* by using irregular wavelet analysis. Through simulations, we prove that our method is fairly robust against various attacks including topological ones.

Keywords: Watermarking, blind detection, wavelet transform, scaling coefficients, topological attacks.

1 Introduction

With the remarkable growth of the network technology such as WWW (World Wide Web), digital media enables us to copy, modify, store, and distribute digital data without effort. As a result, it has become a new issue to research schemes for copyright protection. Traditional data protection techniques such as encryption are not adequate for copyright enforcement, because the protection cannot

* This work was supported by the Ministry of Information & Communications, Korea, under the Information Technology Research Center (ITRC) Program (204-B-000-215).

** Corresponding author.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 123–137, 2006.

be ensured after the data is decrypted. Watermarking provides a mechanism for copyright protection by embedding information, called a watermark, into host data [1]. Note that so-called fragile or semi-fragile watermarking techniques have also been widely used for content authentication and tamper proofing [2]. Here, we address only watermarking technique for copyright protection, namely robust watermarking. Recently, with the interest and requirement of 3-D models such as VRML (Virtual Reality Modeling Language), CAD (Computer Aided Design), polygonal mesh models and medical objects, several watermarking techniques for 3-D mesh models have been developed. 3-D polygonal mesh models have serious difficulties for watermark embedding. While image data is represented by brightness (or amplitudes of RGB components in the case of color images) of pixels sampled over a regular grid in two dimension, 3-D polygonal models have no unique representation, i.e., no implicit order and connectivity of vertices [3]. This creates synchronization problem during the watermark extraction, which makes it difficult to develop robust watermarking techniques. For this reason, most techniques developed for other types of multimedia such as audio, image and video stream are not effective for 3-D meshes. Furthermore, a variety of complex geometrical and topological operations could disturb the watermark extraction for assertion of ownership [3].

The watermarking system can be classified into informed detection and blind detection according to detection procedure. Although it has been known that blind schemes are less robust than informed ones, they are more useful for various applications where a host signal is not available in the watermark detection procedure[1]. For examples, owner identification and copy control systems cannot refer to original data [1,4,5]. Furthermore, the use of informed watermarking can cause to confuse the proof of ownership if an illegal user asserts that he is the copyright holder with a corrupt watermarked data as his original [6].

Most multiresolution based watermarking approaches have been developed to achieve both robustness against various attacks and invisibility of hidden watermark [7,8,9,10,11]. Kanai *et al.* [9] proposed a watermarking algorithm based on wavelet transform. Similar approaches, using Burt-Adelson style pyramid and mesh spectral analysis were also published in [8] and [12], respectively. The multiresolution techniques could achieve a high transparency of watermark, but have not been used as an blind scheme since the connectivity information of vertices must be exactly known for multi-resolution analysis in the watermark extraction process. In particular, the connectivity information is fatally destroyed in the case of topological attacks (also called synchronization attacks) such as simplification and subdivision. Recently, there have been some trials that apply the spectral analysis based techniques directly to point-sampled geometry that is independent of vertex connectivity information [13,14]. However, they are not blind schemes. Obviously, it is required to develop multiresolution based blind watermarking methods for 3D meshes.

In this paper, we propose wavelet analysis based blind watermarking methods which are fairly robust against various attacks including topological ones. In sharp contrast with most conventional wavelet analysis based methods [9,10,11]

where the watermark is embedded into wavelet coefficients, the proposed methods embed the watermark into scale coefficients (e.g. approximated meshes). The methods use statistical features of scale coefficients which are less sensitive to topological attacks. The distribution of scale coefficients is modified by two approaches that were applied directly to vertex norms in the spatial domain in our previous works [15]. One is to shift the mean value of the distribution according to the watermark bit and the other is to change its variance. We use histogram mapping functions for the modification. To provide the blind detection against topological attacks, the watermark is extracted directly from vector norms in the spatial domain. Then, the proposed methods do not require the original model in the process of watermark detection.

The rest of this paper is organized as follows. In Section 2, we introduce some related works including wavelet analysis for 3D surface meshes and conventional wavelet analysis based watermarking methods. The proposed watermark embedding and extraction methods are presented in Section 3. Section 4 shows the simulation results of the proposed against various attacks. Finally, we draw a conclusion.

2 Related Works

2.1 Wavelet Analysis

Wavelet analysis, one of the most useful multiresolution representation techniques, have been used in a broad range of applications, including image compression, physical simulation, hierarchical optimization, and numerical analysis. The basic idea behind wavelet analysis is to decompose a complicated function into a simpler coarse-resolution part, together with a collection of perturbations called wavelet coefficients. Lounsbery [16] extended wavelet analysis to be applied to 3D surface meshes. In this subsection, we describe briefly the wavelet analysis and some matters to be investigated for blind watermarking.

Wavelet analysis scheme simplifies the original meshes by reversing an subdivision scheme. The simplification is repeated until the resulting mesh cannot be simplified anymore. For meshes homeomorphic to a sphere, the simplest mesh is a tetrahedron. We obtain a hierarchy of meshes from the simplest one M^0, called base mesh, to the original mesh M^J. Following [11], the wavelet decomposition can be applied to the geometry of the different meshes which are linked by the following matrix relations:

$$C^{j-1} = A^j C^j \tag{1}$$

$$D^{j-1} = B^j C^j \tag{2}$$

$$C^j = P^j C^{j-1} + Q^j D^{j-1} \tag{3}$$

where C^j is the $v^j \times 3$ matrix representing the coordinates of the scale coefficients of coarse meshes (also called approximated meshes) at the resolution level

j, v^j is the number of vertices for each mesh M^j. D^{j-1} is the $(v^j - v^{j-1}) \times 3$ matrix of the wavelet coefficients at level j. A^j and B^j are the analysis filters, P^j and Q^j are the synthesis filters. Note that the coefficients of the matrix C^J denotes the real spatial coordinates of the model.

In Lounsbery [16], the mesh hierarchy by forward operation can be considered as successive quadrisections of the base mesh (M^0) faces followed by deformation of edges midpoint to fit the surface to be approximated. Conversely, 4:1 face coarsening is the inverse operation of quadrisection. In this scheme the wavelets functions are hat functions associated with odd vertices of the mesh at resolution j and linearly vanishing on the opposite edges. The scaling functions are also hat functions but with a twice wider support and associated with even vertices. For more details of Lounsbery's scheme, see [16].

Recently, the wavelet analysis has been extended to irregular meshes, in which vertices can have any degree [17]. This approach allows to directly process the output meshes of iso-surface which are extracted by usual algorithm such as the well-known Marching Cubes[18]. Due to the irregular connectivity, surface analysis by wavelets cannot be implemented without the knowledge of face fusions which is needed for mesh coarsening. Coarsening is the inverse problem of successive subdivision of the base mesh. In sharp contrast with Lounsbery's regular 4:1 face merge, irregular approach merges the faces into 4:1, 3:1, 2:1, or keep the original face at the current level. In addition, an edge in a face can be switched to improve the merging efficiency. For more details of irregular wavelet analysis, see [17]. Note that our proposed methods will use the irregular wavelet analysis so as to hide the watermark information directly into irregular meshes.

Unlike discrete wavelet transform (DWT) of 2D images, the wavelet coefficients of 3D meshes depend on the seed-triangle group merged during connectivity graph simplification. This means that the original meshes can be perfectly reconstructed by inverse wavelet analysis in terms of geometry (coordinates of vertex) information, but cannot be synthesized exactly with respect to the topology (index of vertex) information. This is mainly caused by the fact that the index of vertices is newly assigned in the process of the inverse wavelet analysis [17]. It might be necessary to reconstruct the topology for watermarking application [11].

2.2 Conventional Wavelet Analysis Based Watermarking Methods

Kanai *et al.* [9], first introduced wavelet analysis based watermarking method for 3D meshes. They employed Lounsbery's scheme [16] for wavelet analysis. They embedded the watermark into the wavelet coefficients, by exploiting the fact that the wavelet transform domain based methods have higher transparency than spatial domain based ones as commonly known in 2D image watermarking applications. The watermark is embedded at various resolution levels to achieve high capacity of watermark. They insisted that error-free detection is possible for the similarity transform, as the wavelet coefficient vector norm is invariant to rotation and translation. However, they employed an informed watermark detection technique which needs the original model. This is mainly caused by that the connectivity information is needed for watermark detection. In addition, the

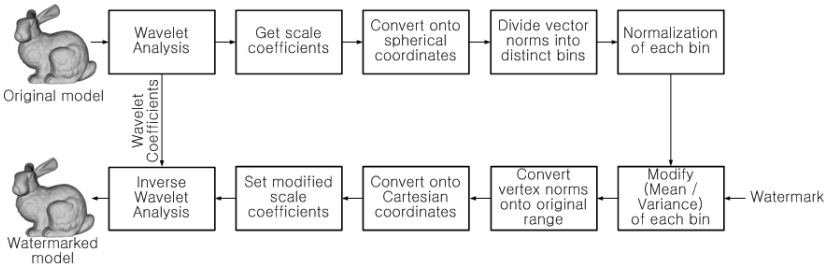


Fig. 1. Watermark embedding

method is limited to apply only to regular meshes that have 4:1 subdivision connectivity. Although they insisted that it can be applied to irregular meshes by using remeshing technique, the remeshed model has different topology from that of the original.

There have been some attempts to extend the wavelet analysis based watermarking to be blind one. Uccheddu *et al.* [10] modified the vector norm of wavelet coefficient according to the angle and direction of the wavelet coefficients vector. They used principal component analysis (PCA) scheme so that the method can extract the hidden watermark without referring to original meshes. However, this method is still limited on regular meshes as in [9]. The method is robust against various against geometrical attacks. However, as they mentioned in [10], it might be vulnerable to topological attacks that cause a synchronization problem. A proper synchronization algorithm should be required in order to ensure robustness against such topology attacks. Another similar approach has been developed in our previous works [11] by introducing a vertex-face re-ordering algorithm as a pre-processing for wavelet analysis. As the re-ordering algorithm is designed to have the same connectivity information in both watermark embedding and detection procedures, blind watermark detection is possible in the case of vertex random re-ordering attack as well as various geometrical attacks such as additive noise, smoothing, similarity/affine transform. This method is not limited to regular meshes. It can be applied to irregular meshes by using irregular wavelet analysis [17]. However, this method make it difficult to extract the watermark without referring to original for various topology attacks such as simplification and subdivision. As results, there has been no wavelet analysis based blind watermarking method that is robust against topological attacks as well as geometrical attacks, despite of its efficiency in terms of the transparency and the capacity of watermark.

3 Proposed Watermarking Methods

In general, it is very important to determine a watermark carrier, also called primitive, that can effectively preserve watermark from attacks. For example, if the wavelet coefficients or scale coefficients arranged in a certain order are

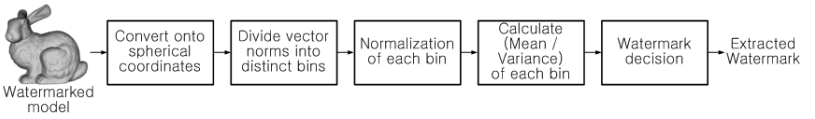


Fig. 2. Watermark extraction

used as the watermark carrier, the connectivity information cannot be retrieved exactly without referring to the original model after topological attacks. This is caused by the fact that 3D polygonal meshes do not have implicit order and connectivity of vertices. For the same reason, pre-processing such as registration and re-sampling is required as in [7,8], or the robustness against topological attacks cannot be guaranteed [9,10,11].

In this paper, we propose a statistical approach. For the sake of achieving the robustness against various topological attacks, we introduce two techniques that have not been employed in the previous wavelet analysis based methods. One is to use statistical features of scale coefficients on approximated meshes as a watermark carrier. The statistical features might be less sensitive than the scale (or wavelet) coefficients themselves in a certain order. The proposed methods modify the distribution of scale coefficients to embed the watermark. As the number of scale coefficients decreases after wavelet analysis, our methods are more applicable to relatively large models. Another technique is to extract the hidden watermark directly from vertex norms in spatial domain. We do not use wavelet analysis in the watermark detection. Then, it is not necessary to use any pre-processing such as registration, re-sampling and re-ordering for wavelet analysis in the process of watermark detection. Furthermore, the technique can reduce drastically the computational complexity and processing time. While this technique can slightly decrease the robustness against geometrical attacks, it allows our methods to be robust against topological attacks. The distribution of scale coefficients is modified by two methods. Method I is to shift the mean value of the distribution according to the watermark bit and Method II is to change its variance. We use histogram mapping functions for the modification. Fig. 1 and Fig. 2 show the proposed watermark embedding and extraction procedures.

To embed the watermark, we first perform forward wavelet analysis with the original meshes. Then, we obtain a set of the scale coefficient vector C^j at approximation (resolution) level j, from Eq. (1) as follows:

$$C^j = [\ c_0,\ c_1, \dots\ ,\ c_{I_j-2},\ c_{I_j-1}\ ]^t \tag{4}$$

where $c_i = (x_i\ y_i\ z_i)^t$, and I_j is the number of scale coefficients. The resolution level j can be determined by considering the capacity and the invisibility of the watermark embedding. Each scale coefficient in Cartesian coordinates, c_i, is converted to spherical coordinates $(\rho_i\ \theta_i\ \phi_i)$, where ρ_i denotes vertex norm of c_i. The proposed method uses only scale coefficient vector norm, ρ_i, for watermarking and keeps the other two components, θ_i and ϕ_i, intact. Then, the probability distribution of ρ_i is divided into N distinct bins with equal range,

according to their magnitude. The maximum ρ_{max} and the minimum ρ_{min} are calculated prior to build N bins. Each bin is used independently to hide one bit of watermark. The robustness of the watermark can be increased by using less number of bins. Vertex norms belonging to the n-th bin are mapped into the normalized range. The m-th normalized vertex norm in n-th bin is denoted as $\rho_{n,m}$. We change the mean (or variance) of each bin via transforming $\rho_{n,m}$ by histogram mapping function. The normalization and distribution modification steps are different in Method I and Method II, respectively. More details about the two steps are described in following subsections. All transformed vertex norms in each bin are mapped onto the original range. Then, all the bins are combined and converted to Cartesian coordinates. Finally, inverse wavelet analysis is performed to get the watermarked meshes.

Watermark extraction procedures begin with spherical coordinates conversion of vertex on the watermarked meshes. Note that the vertex norms are taken from the spatial domain without wavelet analysis. Similar to the watermark embedding process, vertex norms are classified into N bins and mapped onto the normalized range. The mean (or variance) of vertex norms are calculated in each bin and compared to the corresponding threshold. The watermark detection process does not require the original meshes.

3.1 Method I

This method embeds watermarks information by shifting the mean value of each bin according to an assigned watermark bit. In this subsection, we describe mainly the normalization and the mean modification steps in the watermark embedding procedures and watermark decision step in the extraction procedures.

Watermark Embedding. In the normalization step, vertex norms in each bin, $\rho_{n,m}$, are all mapped into the range of $[0, 1]$. Now, we assume that each bin has a uniform distribution over the interval $[0, 1]$, and then the mean $\mu_n = \frac{1}{2}$. The mean is used as a threshold value when shifting the mean of each bin to a certain level.

In the mean modification step, the mean of n-th bin, μ_n, is shifted by a factor $+\alpha$ (or $-\alpha$) to embed n-th watermark bit $\omega_n = +1$ (or $\omega_n = -1$), as follows:

$$\mu'_n = \begin{cases} \frac{1}{2} + \alpha \text{ if } \omega_n = +1 \\ \frac{1}{2} - \alpha \text{ if } \omega_n = -1 \end{cases} \tag{5}$$

where α $(0 < \alpha < \frac{1}{2})$ is the strength factor that can control the robustness and the transparency of watermark. To shift the mean to the desired level, the vertex norms, $\rho_{n,m}$, are transformed iteratively by a histogram mapping function as follows:

$$\rho'_{n,m} = (\rho_{n,m})^{k_n} \text{ for } 0 < k_n < \infty \text{ and } k_n \in \mathbb{R} \tag{6}$$

If the parameter k_n is selected in $]1, \infty[$, $\rho_{n,m}$ are transformed into $\rho'_{n,m}$ with relatively small value. Moreover, $\rho'_{n,m}$ becomes smaller as increasing k_n.

It means a reduction of the mean. On the other hand, the mean value increases for decreasing k_n on the range $]0,1[$. Note that the parameter k_n is determined according to the distribution of the each bin.

Watermark Extraction. The extraction procedure is quite simple. The vertex norms taken from the watermarked mesh model without wavelet analysis are classified into N bins and mapped onto the normalized range of $[0,1]$. The mean of each bin, μ_n'', is calculated and compared to the threshold, $\frac{1}{2}$. The watermark hidden in the n-th bin, ω_n'', is extracted by means of

$$\omega_n'' = \begin{cases} +1, \text{ if } \mu_n'' > \frac{1}{2} \\ -1, \text{ if } \mu_n'' < \frac{1}{2} \end{cases} \tag{7}$$

3.2 Method II

This method is to change the variance value of each bin according to an assigned watermark bit. Both the watermark embedding and extraction of the method are quite similar to Method I. We also describe the normalization and the variance modification steps in the watermark embedding procedures and watermark decision step in extraction procedures. For simplicity, we use the same notation as in Method I.

Watermark Embedding. In the normalization step, vertex norms in each bin, $\rho_{n,m}$, are all mapped into the range of $[-1,1]$. Assume that each bin has a uniform distribution over the interval $[-1,1]$, and then the variance $\sigma_n^2 = \frac{1}{3}$. The variance is used as a threshold value when modifying the variance of each bin to a certain level.

In the variance modification step, the variance of n-th bin, σ_n^2, is modified by a factor $+\alpha$ (or $-\alpha$) to embed n-th watermark bit $\omega_n = +1$ (or $\omega_n = -1$), as follows:

$$\sigma_n^{2'} = \begin{cases} \frac{1}{3} + \alpha \text{ if } \omega_n = +1 \\ \frac{1}{3} - \alpha \text{ if } \omega_n = -1 \end{cases} \tag{8}$$

where α $(0 < \alpha < \frac{1}{3})$ is the strength factor. To modify the variance to the desired level, the vertex norms, $\rho_{n,m}$, are transformed iteratively by a histogram mapping function as given by,

$$\rho'_{n,m} = sign(\rho_{n,m}) \left|\rho_{n,m}\right|^{k_n} \text{ for } 0 < k_n < \infty \text{ and } k_n \in \mathbb{R} \tag{9}$$

where $sign(x)$ denotes the sign of x. If the parameter k_n is selected in $]1,\infty[$, $\rho_{n,m}$ is transformed into $\rho'_{n,m}$ with relatively small absolute value while maintaining its sign. Moreover, the absolute value of transformed variable becomes smaller as increasing k_n. It means a reduction of the variance. On the other hand, the variance increases for decreasing k_n on the range $]0,1[$.

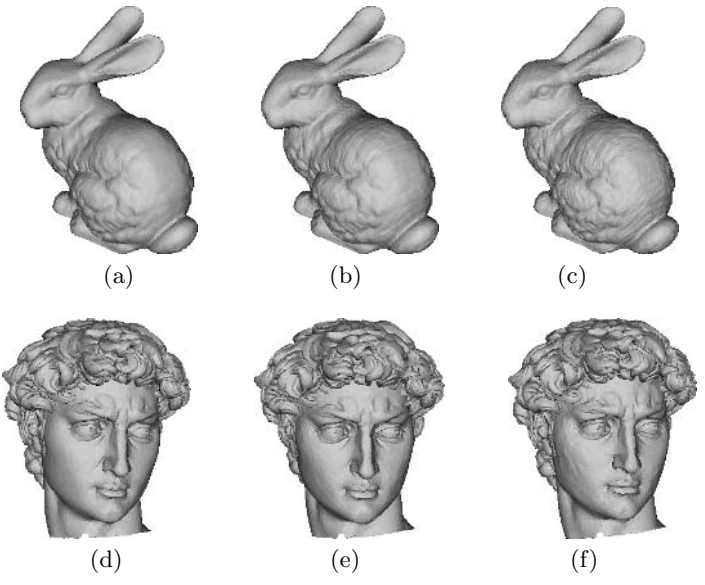


Fig. 3. Test models and watermarked models, (a) Stanford bunny, the original, (b) $\alpha = 0.07$ by Method I (c) $\alpha = 0.14$ by Method II, and (d) davidhead, the original, (e) $\alpha = 0.1$ by Method I, and (f) $\alpha = 0.2$ by Method II

Watermark Extraction. Watermark extraction process for this method is also quite simple. The variance of each bin, $\sigma_n^{2''}$, is calculated and compared with the threshold value, $\frac{1}{3}$. The watermark hidden in the n-th bin, ω_n'', is extracted by means of

$$\omega_n'' = \begin{cases} +1, \text{ if } \sigma_n^{2''} > \frac{1}{3} \\ -1, \text{ if } \sigma_n^{2''} < \frac{1}{3} \end{cases} \tag{10}$$

4 Experimental Results

In this section, we show the experimental results of our proposal. We use Stanford bunny (34,834 vertices, 69,451 faces), davidhead (24,085 vertices, 47,753 faces) as shown in Fig. 3. We embedded 64 bits of watermark. The quality of the geometry of the model was measured by Metro [19], which compute the forward and backward RMS (Root Mean Square) errors. We used the maximum RMS ($MRMS$) between the two RMS values, denoted as $E(V, V')$. Here, V and V' indicate the original and watermarked meshes, respectively.

To evaluate the transparency of our proposed methods, we embedded the watermark into different resolution levels while keeping similar $E(V, V')$. The same 64 bits of watermark information are embedded into the $J-2$-th (C^{J-2}), $J-1$-th (C^{J-1}) and J-th (C^J) levels, respectively. The watermarked bunny models are

shown in Fig. 4. Even though all meshes have similar $E(V, V') \approx 0.70 \times 10^{-4}$, embedding at lower resolution level (here, C^{J-2} and C^{J-1}) is visually better than higher resolution level (C^J).

The robustness is evaluated by correlation coefficient, ($Corr$), between the designed and extracted watermark.

$$Corr = \frac{\sum_{n=0}^{N-1} (\omega_n'' - \varpi'')(\omega_n - \varpi)}{\sqrt{\sum_{n=0}^{N-1} (\omega_n'' - \varpi'')^2 \times \sum_{n=0}^{N-1} (\omega_n - \varpi)^2}} \tag{11}$$

where ϖ indicates the average of the watermark and $Corr$ exists in the range of $[-1, 1]$.

To evaluate the robustness of the proposed methods against various attacks, the watermark information was embedded into the scale coefficients at the $J-1$-th resolution level, as example. The strength factor α was determined experimentally so that both methods have similar quality for each model in terms of $MRMS$. The watermarked meshes are given in Fig. 3. Table 1 shows the performances in terms of $MRMS$ and $Corr$ in the case of no attack. Here, the strength factors of each watermark are also listed. The watermark was also extracted from the scale coefficients at the same resolution level as used in watermark embedding, and its correlation coefficients, $Corr^{J-1}$, are listed. As a reference, the

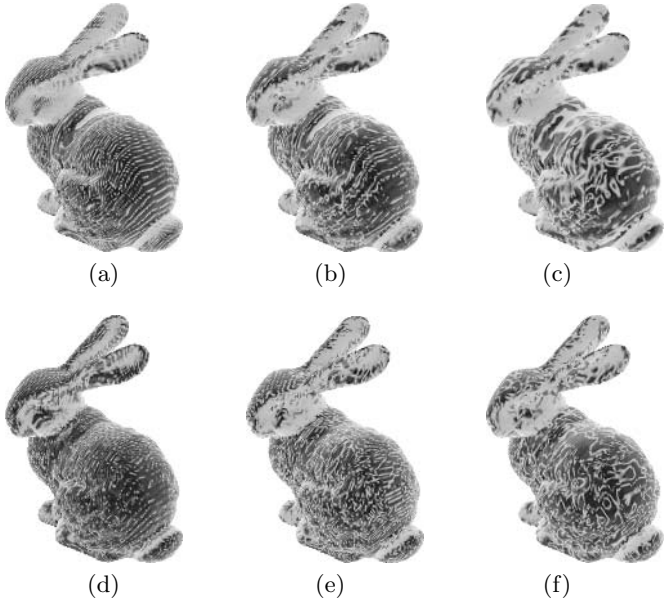


Fig. 4. Watermark transparency when 64 watermark bits are embedded in Stanford bunny at different resolution levels, (a) C^J, (b) C^{J-1}, (c) C^{J-2} by Method I, and (d) C^J, (e) C^{J-1}, (f) C^{J-2} by Method II. Dark region indicates the distorted one caused by watermark embedding.

Table 1. Evaluation of watermarked meshes when no attack

Method	Model	α	$E(V, V')$	The proposed		Cho *et al.*[15]
				$Corr^{J-1}$	$Corr$	$Corr$
Method I	bunny	0.070	0.69×10^{-4}	1.00	0.88	1.00
	davidhead	0.100	137.09×10^{-4}	1.00	1.00	1.00
Method II	bunny	0.140	0.70×10^{-4}	1.00	0.94	1.00
	davidhead	0.200	136.81×10^{-4}	1.00	1.00	1.00

Table 2. Evaluation of robustness against additive binary noise attacks

Method	Model	Error rate	$E(V, V')$	The proposed		Cho *et al.*[15]
				$Corr^{J-1}$	$Corr$	$Corr$
Method I	bunny	0.1%	0.97×10^{-4}	0.94	0.75	1.00
		0.3%	2.07×10^{-4}	0.53	0.53	0.53
		0.5%	3.24×10^{-4}	0.09	0.00	0.28
	davidhead	0.1%	164.85×10^{-4}	1.00	0.84	0.97
		0.3%	308.76×10^{-4}	0.38	0.25	0.38
		0.5%	472.93×10^{-4}	0.06	0.09	0.09
Method II	bunny	0.1%	0.95×10^{-4}	1.00	0.91	1.00
		0.3%	2.05×10^{-4}	-0.03	0.13	-0.09
		0.5%	3.24×10^{-4}	0.00	0.06	0.03
	davidhead	0.1%	165.69×10^{-4}	1.00	0.97	1.00
		0.3%	309.83×10^{-4}	-0.13	0.06	-0.16
		0.5%	470.70×10^{-4}	0.06	0.16	0.09

simulation results with the spatial domain based statistical methods [15] are given in the table. As expected, wavelet analysis based methods with detecting the watermark at $J-1$-th level have similar watermark detection rate to the spatial domain based methods, but our proposed methods with detecting at J-th level shows slightly lower detection rate. This is mainly caused by the fact that wavelet coefficients at $J-1$-th level act like noise when the hidden watermark is extracted in the spatial domain (C^J).

The watermarked models listed in Table 1 are used to evaluate the robustness against several attacks. For the noise attacks, binary random noise was added to each vertex of the watermarked model with three different error rates: 0.1%, 0.3%, 0.5%. Here, the error rate represents the amplitude of noise as a fraction of the maximum vertex norm of the object. The robustness against the noise attacks is shown in Table 2. Fairly good robustness can be expected for the error rate less than 0.1% both for the proposed.

Table 3 shows the performance of the smoothing attacks [20]. Three different pairs of iteration and relaxation were applied. The robustness depends on the smoothness of the original meshes. A model that has a lot of bumpy area such as davidhead is relatively sensitive to smoothing attacks than bunny model.

Table 3. Evaluation of robustness against laplacian smoothing attacks

Method	Model	# of iteration, relaxation	$E(V, V')$	The proposed		Cho *et al.*[15]
				$Corr^{J-1}$	$Corr$	$Corr$
Method I	bunny	(10,0.03)	0.73×10^{-4}	1.00	0.85	1.00
		(30,0.03)	0.99×10^{-4}	0.88	0.75	0.87
		(50,0.03)	1.33×10^{-4}	0.76	0.70	0.84
	davidhead	(10,0.03)	179.03×10^{-4}	0.80	0.66	0.78
		(30,0.03)	352.98×10^{-4}	0.28	0.28	0.41
		(50,0.03)	515.13×10^{-4}	0.13	0.19	0.19
Method II	bunny	(10,0.03)	0.68×10^{-4}	1.00	0.91	1.00
		(30,0.03)	0.92×10^{-4}	0.94	0.91	0.91
		(50,0.03)	1.27×10^{-4}	0.88	0.75	0.06
	davidhead	(10,0.03)	178.52×10^{-4}	1.00	1.00	1.00
		(30,0.03)	354.79×10^{-4}	0.35	0.09	0.22
		(50,0.03)	518.41×10^{-4}	0.16	0.09	0.13

Table 4. Evaluation of robustness against simplification attacks

Method	Model	Reduction ratio	$E(V, V')$	The proposed		Cho *et al.*[15]
				$Corr^{J-1}$	$Corr$	$Corr$
Method I	bunny	30%	0.75×10^{-4}	-	0.72	1.00
		50%	0.80×10^{-4}	-	0.69	0.88
		70%	0.92×10^{-4}	-	0.48	0.58
		90%	3.44×10^{-4}	-	0.06	0.42
	davidhead	30%	352.37×10^{-4}	-	0.91	0.85
		50%	422.06×10^{-4}	-	0.78	0.81
		70%	581.23×10^{-4}	-	0.85	0.85
		90%	1015.49×10^{-4}	-	0.16	0.35
Method II	bunny	30%	0.73×10^{-4}	-	0.78	1.00
		50%	0.77×10^{-4}	-	0.81	0.97
		70%	0.90×10^{-4}	-	0.81	0.94
		90%	3.12×10^{-4}	-	0.76	0.88
	davidhead	30%	509.21×10^{-4}	-	0.91	1.00
		50%	573.84×10^{-4}	-	0.91	0.94
		70%	695.56×10^{-4}	-	0.91	0.94
		90%	1419.40×10^{-4}	-	0.34	0.28

Simplification attacks were carried out by using quadric error metrics [21]. The robustness is shown in Table 4, where the percentage represents the number of removed vertices as a fraction of total number of vertices. It is very difficult to retrieve the hidden watermark from $J-1$-th resolution level in the case of simplification attacks that destroy fatally the connectivity information. This table shows that the proposed methods have good robustness comparable to the spatial domain based approaches. Subdivision attacks were also carried out. Each triangle was uniformly divided into four cells. The performance is listed in

Table 5. Evaluation of robustness against 1-to-4 subdivision attacks

Method	Model	$E(V, V')$	The proposed		Cho *et al.*[15]
			$Corr^{J-1}$	$Corr$	$Corr$
Method I	bunny	0.69×10^{-4}	-	0.76	1.00
	davidhead	136.29×10^{-4}	-	0.48	0.94
Method II	bunny	0.69×10^{-4}	-	0.88	1.00
	davidhead	135.04×10^{-4}	-	0.94	0.97

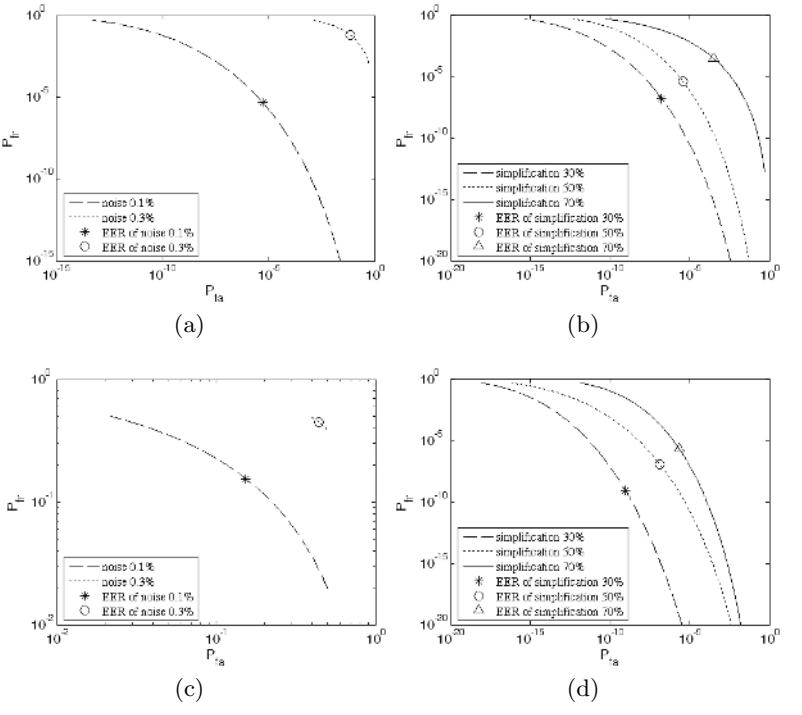


Fig. 5. ROC curves of bunny model (a) watermarked by Method I and attacked by adding noise, (a) watermarked by Method I and attacked by simplification, (c) watermarked by Method II and attacked by adding noise, (d) watermarked by Method II and attacked by simplification

Table 5. The results show that the proposed methods are fairly robust to such attacks.

The propose methods were analyzed by *ROC* (Receiver Operating Characteristic) curve that represents the relation between probability of false rejections P_{fr} and probability of false alarms P_{fa} varying the decision threshold for declaring the watermark present. The probability density functions for P_{fr} and P_{fa} were measured experimentally with 100 correct and 100 wrong keys, and

approximated to Gaussian distribution. In these simulations, we used the same watermarked model of bunny as used in Table 1. Fig. 5 shows the *ROC* curves when additive noise and simplification attacks are respectively applied into the watermarked bunny. *EER* (Equal Error Ratio) is also indicated in this figure. Both the methods have fairly secure for the simplification attacks, but Method II shows relatively lower performance than Method I in the case of noise attacks.

5 Conclusion

In this paper, we propose blind watermarking methods based on wavelet analysis for 3-D mesh model. To achieve the watermark transparency and the robustness against various topological attacks, watermark information is embedded into scale coefficients at lower resolution level by modifying their distribution and extracted, not from the same resolution level as used in embedding process, but directly from the spatial domain. As the watermark detection process does not require wavelet analysis or any pre-processing, it is quite simple. Through simulations, we proved that the proposed has fairly good performance in terms of the watermark transparency and robustness against various attacks. Even though the proposed methods are not highly robust, our attempts demonstrate a possible, blind watermarking based on wavelet analysis for 3-D mesh model.

References

1. Cox, I., Miller, M.L., Bloom, J.A.: Digital watermarking. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2002)
2. Cayre, F., Macq, B.: Data hiding on 3-d triangle meshes. IEEE Trans. Signal Processing **51** (2003) 939–949
3. Zhi-qiang, Y., Ip, H.H.S., Kwok, L.F.: Robust watermarking of 3d polygonal models based on vertice scrambling. In: Computer Graphics International, IEEE Computer Society (2003) 254–257
4. Kejariwal, A.: Watermarking. IEEE Potentials **Oct./Nov.** (2003) 37–40
5. Wong, P.H.W., Au, O.C., Yeung, Y.M.: Novel blind multiple watermarking technique for images. IEEE Trans. Circuits Syst. Video Techn. **13** (2003) 813–830
6. Craver, S., Memon, N.D., Yeo, B.L., Yeung, M.M.: Can invisible watermarks resolve rightful ownerships? In: Storage and Retrieval for Image and Video Databases (SPIE). (1997) 310–321
7. Praun, E., Hoppe, H., Finkelstein, A.: Robust mesh watermarking. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. (1999) 49–56
8. Yin, K., Pan, Z., Shi, J., Zhang, D.: Robust mesh watermarking based on multiresolution processing. Computers and Graphics **25** (2001) 409–420
9. Kanai, S., Date, D., Kishinami, T.: Digital watermarking for 3d polygon using multiresolution wavelet decomposition. In: Proc. Sixth IFIP WG 5.2 GEO-6, Tokyo, Japan (1998) 296–307
10. Uccheddu, F., Corsini, M., Barni, M.: Wavelet-based blind watermarking of 3d models. In: Proceedings of the 2004 multimedia and security workshop on Multimedia and security, ACM Press (2004) 143–154

11. Kim, M.S., Valette, S., Jung, H.Y., Prost, R.: Watermarking of 3d irregular meshes based on wavelet multiresolution analysis. Lecture Notes in Computer Science **3710** (2005) 313–324
12. Ohbuchi, R., Takahashi, S., Miyazawa, T., Mukaiyama, A.: Watermarking 3d polygonal meshes in the mesh spectral domain. In: GRIN'01: No description on Graphics interface 2001, Toronto, Ont., Canada, Canadian Information Processing Society (2001) 9–17
13. Cotting, D., Weyrich, T., Pauly, M., Gross, M.: Robust watermarking of point-sampled geometry. In: Proceedings of International Conference on Shape Modeling and Applications 2004 (SMI'04). (2004) 233–242
14. Ohbuchi, R., Mukaiyama, A., Takahashi, S.: Watermarking a 3d shape defined as a point set. In: Proceedings of 2004 International Conference on Cyberworlds. (2004) 392–399
15. Cho, J.W., Prost, R., Jung, H.Y.: An oblivious watermarking for 3-d polygonal meshes using distribution of vertex norms. IEEE Trans. Signal Processing (to be appeared, final manuscript is available at `http://yu.ac.kr/ hoyoul/IEEE_sp_final.pdf`)
16. Lounsbery, M.: Multiresolution Analysis for Surfaces of Arbitrary Topological Type. PhD thesis, Dept. of Computer Science and Engineering, U. of Washington (1994)
17. Valette, S., Prost, R.: Multiresolution analysis of irregular surface meshes. IEEE Trans. Visual. Comput. Graphics **10** (2004) 113–122
18. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, ACM Press (1987) 163–169
19. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: Measuring error on simplified surfaces. Computer Graphics Forum **17** (1998) 167–174
20. Field, D.: Laplacian smoothing and delaunay triangulation. Communication and Applied Numerical Methods **4** (1988) 709–712
21. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: SIGGRAPH '97, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (1997) 209–216

Watermarking for 3D Keyframe Animation Based on Geometry and Interpolator

Suk-Hwan Lee[1], Ki-Ryong Kwon[2,⋆], and Dong Kyue Kim[3]

[1] Dept. of Information Security, Tongmyong University,
skylee@tu.ac.kr
[2] Div. of Electronic, Computer&Telecommunication Eng., Pukyong National Univ.
krkwon@pknu.ac.kr
[3] Dept. of Electronics and Computer Eng., Hanyang University
dqkim@hanyang.ac.kr

Abstract. This paper presents a novel watermarking scheme for 3D keyframe animation based on geometric structure and interpolator. The geometric watermarking embeds the watermark into the distribution of vertex coordinates in each of initial mesh models. The interpolator watermarking embeds the same watermark into key values in position interpolator. Experimental results show that the proposed scheme has the robustness against various geometric attacks and timeline attacks as well as the invisibility.

Keywords: 3D Keyframe Animation, Watermarking, Geometry, PositionInterpolator.

1 Introduction

Recent development in computer graphics technology enabled real-time 3D reality modelling. 3D computer animation has become an important application area of 3D reality modelling. 3D computer movies and games have rapidly become major 3D computer animation applications, and are fast growing business applications in 3D contents industry. As the 3D character animation market grows, copy right protection to protect ani-mated movies and games against illegal copy and reproduction has been increasingly important issue in this emerging market. Researchers have investigated various watermarking/fingerprinting schemes for copyright protection and illegal copy tracing for various media contents such as digital audio, still image, and video.[1],[2] Developing effective and efficient watermarking schemes for 3D graphics modelling has become an important research focus.[3]−[6] However very little research work has been conducted to protect 3D computer animation contents by watermarking.

An animation in 3D graphics is interpreted as a simulation of movement created by displaying a series of moving objects including mesh or texture in 3D space. Keyframe animation that applies the above interpretation is widely used in 3D graphics animation. Key-frame animation registers the animated key values of important frames selected and generates the rest frames by interpolating

⋆ Corresponding Author.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 138–149, 2006.

the registered key values. An animation can be defined informally as an interpolation process in a virtual environment. VRML[7] and MPEG-4 AFX[8] control animation by using interpolator nodes that are designed for linear keyframe animation.

This paper presents a novel watermarking scheme for keyframe animation. The proposed scheme randomly selects embedding meshes, which are transform nodes in the entire hierarchical structure of an animation. The watermark is then embedded into vertex coordinates of the selected transform nodes. Experimental results show that the proposed scheme is robust to geometrical attacks and timeline attacks that can be performed by existing 3D graphics editing tools.

In section 2, keyframe animation is briefly described. Section 3 presents the proposed scheme for watermarking keyframe animation. Experimental results are discussed in section 4. Section 5 concludes the paper.

2 Keyframe Animation

In this section, keyframe animation is explained based on VRML[7] technology. There are six major types of interpolator nodes used in VRML: ColorInterpolator, CoordinateInter-polator, NormalInterpolator, OrientationInterpolator, PositionInterpolator, and Sca-larInterpolator. PositionInterpolator and OrientationInterpolator together can be used to create a simple keyframe animation. A PositionInterpolator linearly interpolates a set of key values each of which represents an arbitrary position of an object. A key value (denoted as KeyValue hereafter) of the PositionInterpolator is defined as a field (or vector) that consists of 3 floating point numbers in Cartesian coordinates (x, y, z). Note that the KeyValue field of an interpolator must encapsulate key values for the required interpolation process. An OrientationInterpolator interpolates among a set of orientation values in the keyValue field. In other words, an OrientationInterpolator interpolates linearly the arc lengths along the shortest path that is computed on the unit sphere between two orientations. A KeyValue for the OrientationInterpolator consists of 4 floating point numbers (x, y, z, θ), where the symbol θ denotes the angle for rotating a position of an object to the direction of $\mathbf{x}, \mathbf{y}, \mathbf{z}$ axes. Figure 1 shows a code segment of a VRML animation file that consists of transform nodes of each mesh model called model (local) coordinate system. Transform matrix of translation, scale, rotation in a transform node of each mesh model with hierarchical structure are also shown in the Figure 1. These matrices represent the coordinate system for modeling each mesh model. Each field in a transform node is performed to the order of scale, rotation, and translation. From the Figure 1, we can identify targets for robust embedding of watermarks as follows:

- Transform matrix of translation, scale, rotation in a transform node
- Keys or KeyValues in an interpolator node: These values represent the moving information of an object and are important to be considered in 3D animation watermarking unlike 3D (still) graphics watermarking.

– The geometric properties such as vertex coordinate and connectivity (CoordIndex node) in IndexedFaceSet node of each initial mesh model: 3D animation watermarking can use these properties in the hierarchical structure.

Our proposed watermarking scheme first determines the embedding targets as keyValues in interpolator nodes and vertex coordinates in each mesh model with hierarchical structure. The watermarks are then embedded into vertex coordinates and keyValues of PositionInterpolator in the selected transform nodes. The proposed scheme is described in detail in the next section.

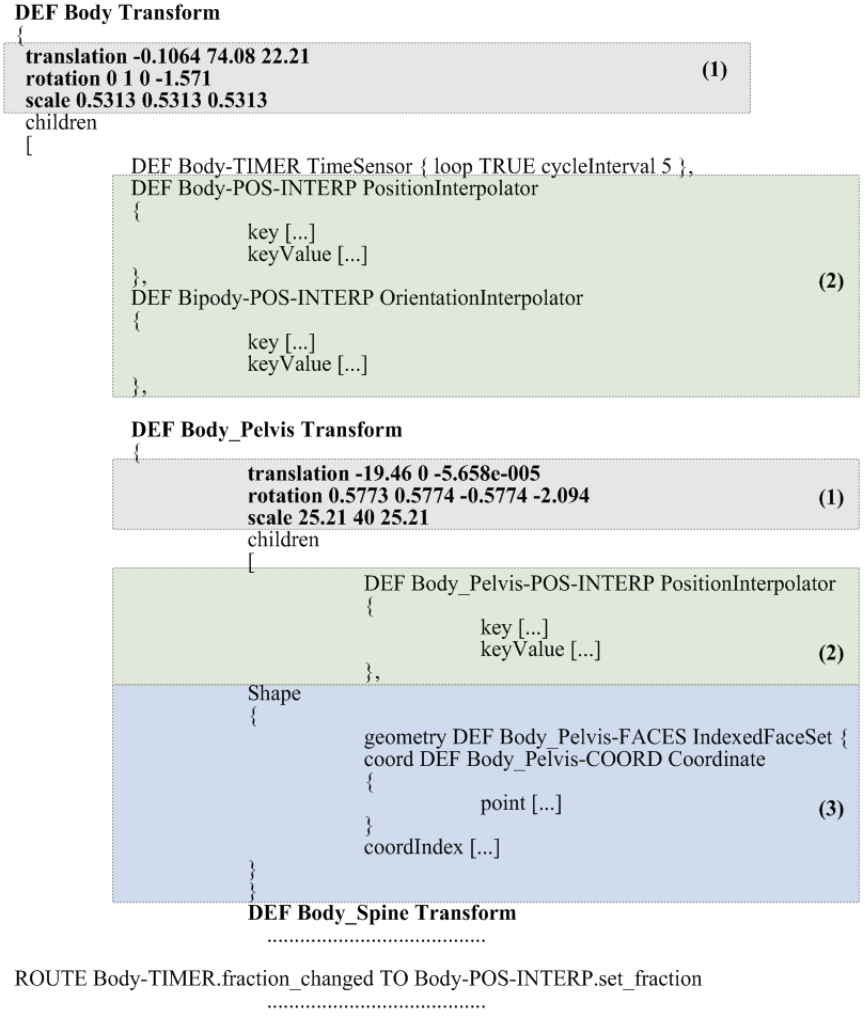


Fig. 1. VRML code structure for animation

3 Proposed Watermarking Scheme

The block diagram of the proposed 3D animation watermarking scheme is shown in the Figure 2. Meshes in the hierarchical structure are called as transform nodes in this paper.

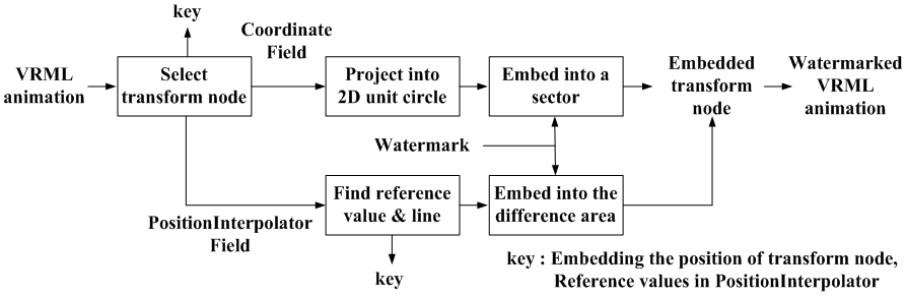


Fig. 2. The proposed scheme for 3D animation watermarking

3.1 Geometric Watermarking

All unit vectors $\hat{\mathbf{v}}_{i\in[0,N_{TR}]}$ of vertices $\mathbf{v}_{i\in[0,N_{TR}]}$ in a selected transform node TR_i are projected into 2D coordinate system (X_{local}, Y_{local}) within the unit circle. The unit circle is divided equally into n sectors each of which is embedded N bits of watermark. More precisely, a binary bit of watermark is embedded into a sector with a center point $\mathbf{c}_{i\in[1,n]}$ of vectors that are projected into a sector. The center point is characterized by the Equation (1), where N_k is the number of vectors $\hat{v}_{xj}X_{local} + \hat{v}_{yj}Y_{local}$ that are projected into sector k. Figure 3 shows embedding process of geometric watermarking. Figure 3-(a) depicts projection into unit circle of 2D local coordinate system and Figure 3-(b) shows embedding a watermark bit into a sector of unit circle.

$$\mathbf{c}_{k\in[1,n]} = \frac{1}{N_k}\sum_{j=1}^{N_k}(\hat{v}_{xj}X_{local} + \hat{v}_{yj}Y_{local}) = c_{kx}X_{local} + c_{ky}Y_{local} \tag{1}$$

For robust watermarking, the target points $\{\mathbf{o}_{w=0}, \mathbf{o}_{w=1}\}$ selected are the mid-points in two halved areas of a sector. Thus, the target points of k th sector $\{\mathbf{o}_{w=0}, \mathbf{o}_{w=1}\}$ are

$$\mathbf{o}_{w=0} = \frac{1}{\sqrt{2}}cos\frac{4k-3}{2n}\pi X_{local} + \frac{1}{\sqrt{2}}sin\frac{4k-3}{2n}\pi Y_{local} = o_{x0}X_{local} + o_{y0}Y_{local} \tag{2}$$

$$\mathbf{o}_{w=1} = \frac{1}{\sqrt{2}}cos\frac{4k-1}{2n}\pi X_{local} + \frac{1}{\sqrt{2}}sin\frac{4k-1}{2n}\pi Y_{local} = o_{x1}X_{local} + o_{y1}Y_{local}. \tag{3}$$

A center point $\mathbf{c}_{i\in[1,n]}$ is moved toward the target point $\mathbf{o}_{w=1}$ of the right halved area of the sector if a watermark bit w is 1, or is moved toward the target point $\mathbf{o}_{w=0}$ of left side one if a watermark bit w is 0, as shown in Figure 3-(b). To

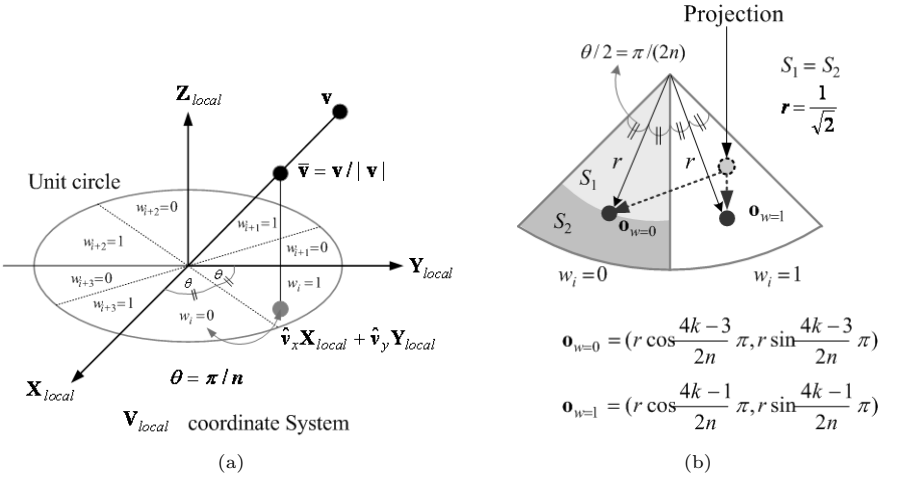


Fig. 3. Embedding geometric watermarking into a transform node (a) Projection and (b) Emebdding

move the center point toward a target point according to the watermark bit, all projected vertices $\hat{\mathbf{v}}_{j\in[0,N_{k_j}]} = \hat{v}_{xj}X_{local} + \hat{v}_{yj}Y_{local}$ in a sector are changed to preserve invisibility by δ.

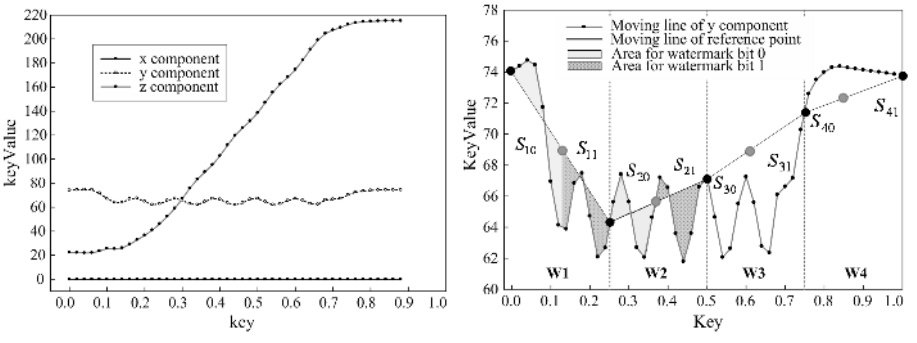
$$\hat{v}'_{xj} = \hat{v}_{xj} + \delta_{xj}, \hat{v}'_{yj} = \hat{v}_{yj} + \delta_{yj} \tag{4}$$

δ_{xj} is Δx_j if $\alpha(o_{kx} - c_{kx}) > \Delta x_j$ and $-\Delta x_j$ if $\alpha(o_{kx} - c_{kx}) < -\Delta x_j$. Otherwise, $\delta_{xj} = \alpha(o_{kx} - c_{kx})$. δ_{yj} is Δy_j if $\alpha(o_{ky} - c_{ky}) > \Delta y_j$ and $-\Delta y_j$ if $\alpha(o_{yx} - c_{yx}) < -\Delta y_j$. Otherwise, $\delta_{yj} = \alpha(o_{ky} - c_{ky})$. $\Delta x_j, \Delta y_j$ are the moving limit ranges. These values are calculated by $\Delta x_j = min|\hat{v}_{xj} - \hat{v}_{val(x)}|$ and $\Delta y_j = min|\hat{v}_{yj} - \hat{v}_{val(y)}|$ respectively, where $\hat{v}_{val(x)}, \hat{v}_{val(y)}$ are X_{local}, Y_{local} coordinate values of valence vertices that are connected to $\hat{\mathbf{v}}_j$.The value of n is 4 in order to quadrisect a unit circle. Value 0.5 is chosen for the moving factor α to realize robust and invisible watermarking.

3.2 Interpolator Watermarking

PositionInterpolator consists of the 3D coordinate and keyValues. KeyValues change over key times that represent the 3D motion position path of an object. The watermark is embedded into these two components of the selected transform node. A transform node in the hierarchical structure is first selected randomly and then the watermark is embedded into the components of the selected node with velocity by using area difference. The watermark is embedded into the components with the variance of velocity, $\sigma = \sqrt{\sum_{i=0}^{N_{key}}(v_i - \bar{v})^2/N_{key}} > 0.5$, where $\bar{v}$ is the mean velocity and N_{key} is the number of keys. To embed n bits of watermark into each component, the key time is divided into n equal

parts with $n+1$ reference points (i.e. $r_{i\in[0,n]}, r_0 = key[0], \text{and} r_n = key[n]$). Each divided part $W_{i\in[1,n]}$ is represented by $(key[r_{i-1}], key[r_i])_{i\in[1,n]}$. From now on, the keyValue is denoted by KV for simplicity. If $KV[r_i]$ of the reference point $r_{i\in[0,n]}$ is not given, $KV[r_i]$ will be generated by interpolating the neighborhood KVs. $KV[r_i]$ must then be stored to extract the watermark. Figure 4 (b) shows 4 watermark bits embedded into 4 divided parts with 5 reference points $r_{i\in[0,4]}$ by using the area difference. For embedding one bit w_i into a



(a) (b)

Fig. 4. Embedding watermark by using the area difference (a) PositionInterpolator in Bip transform node of Wailer animation in 3D-MAX: The number of keys is 45. (b) Watermark embedding in the keyValues of each component in PositionInterpolator by using area difference.

part $W_i = (key[r_{i-1}], key[r_i])$, the area difference S_i between the reference line through $(key[r_{i-1}], KV[r_{i-1}])$,$(key[r_i], KV[r_i])$ and the moving line of the original keyValues $KV[j], r_{j-1} < j < r_j$ is calculated. The reference line is obtained by the Equation (5).

$$y = \frac{KV[r_i] - KV[r_{i-1}]}{key[r_i] - key[r_{i-1}]}x + \frac{key[r_i]KV[r_{i-1}] - key[r_{i-1}]KV[r_i]}{key[r_i] - key[r_{i-1}]} \tag{5}$$

where x, y are variables representing key and KV. The area difference S_i is further divided into two areas S_{i0} and S_{i1}, each of which represents the area difference within $\{key[r_{i-1}], (key[r_i]+key[r_{i-1}])/2\}$ and $\{(key[r_i]+key[r_{i-1}])/2, key[r_i]\}$ respectively. Let us assume key times of $key[j]$ be $(r_{i-1} < j < (r_i + r_{i-1})/2, j \in [1, N_{i0}])$ in $\{key[r_{i-1}], (key[r_i] + key[r_{i-1}])/2\}$ and $((r_i + r_{i-1})/2 < j < r_{i-1}, j \in [N_{i0}+1, N_{i1}-N_{i0}]$ in $\{(key[r_i] + key[r_{i-1}])/2, key[r_i]\}$. The area difference $S_{i0(ori1)}$ is

$$S_{i0(ori1)} = S_{triangle,first} + S_{triangle,last} + \sum_i S_{trapesoid} + \sum_i S_{twist_trapesoid} \tag{6}$$

If w_i is 0, S_{i0} is made larger than S_{i1} by increasing velocity of key times in S_{i0}

$$\frac{KV'_{i0}[j] - KV_{i0}[j-1]}{key_{i0}[j] - key_{i0}[j-1]} = \alpha \times \frac{KV_{i0}[j] - KV_{i0}[j-1]}{key_{i0}[j] - key_{i0}[j-1]}, j \in [1, N_{i0}] \tag{7}$$

and decreasing velocity of key times in S_{i1},

$$\frac{KV'_{i1}[j] - KV_{i1}[j-1]}{key_{i0}[j] - key_{i1}[j-1]} = \frac{1}{\alpha} \times \frac{KV_{i1}[j] - KV_{i1}[j-1]}{key_{i1}[j] - key_{i1}[j-1]}, j \in [N_{i0}+1, N_{i1}-N_{i0}] \cdot \tag{8}$$

Thus,

$$KV'_{i0} = \alpha \times KV[j] + (1-\alpha) \times KV[j-1], j \in [1, N_{i0}] \tag{9}$$

$$KV'_{i1} = \frac{1}{\alpha} \times KV[j] + (1-\frac{1}{\alpha}) \times KV[j-1], j \in [N_{i0}+1, N_{i1}-N_{i0}] \tag{10}$$

On the contrary, S_{i1} is made larger than S_{i0} if w_i is 1.

3.3 Watermark Extracting

n bits out of total m bits of watermark are embedded into vertex coordinates and KVs in PositionInterpolator of a transform node. The index of the embedded transform node and KVs of reference key points in PositionInterpolator are used for extracting the embedded watermark. The process of watermark extracting is similar to the embedding process. Projecting vertex coordinates in the embedded transform node into 2D unit circle is performed first. The center point $\hat{\mathbf{c}}_{k \in [1,n]} = \hat{c}_{kx} X_{local} + \hat{c}_{ky} Y_{local}$ of each sector in a circle is then calculated. A bit w_k watermark can be extracted by using the angle $\theta_k = tan^{-1}(\hat{c}_{ky}/\hat{c}_{kx})$,$(2(k-1)\pi/n \leq \theta_k \leq 2k\pi/n)$ of center point $\hat{\mathbf{c}}_{k \in [1,n]}$ as follows:

$$w'_k = 0, \quad if \ \frac{2(k-1)\pi}{n} < \theta_k \leq \frac{(2k-1)\pi}{n}$$

$$w'_k = 1, \quad else \ \frac{(2k-1)\pi}{n} < \theta_k \leq \frac{2k\pi}{n}$$

Before extracting the watermark in PositionInterpolator, the lines of reference values $KV[r_i]_{i \in [0,n]}$ are compared with those of reference values $KV'[r_i]_{i \in [0,n]}$ in attacked animation. If these lines are identical and overlapped each other, the watermark can be extracted without the rescaling process. If not, in case of key time scaling or cropping, the watermark will be extracted after performing the rescaling process that changes the reference points $r'_{i \in [0,n]}$ so that these lines of reference values are identical. A watermark bit w_k can be extracted by comparing with the difference area of each part: $w'_k = 0$ if $S_{k0} > S_{k1}$ and $w'_k = 1$ if $S_{k0} < S_{k1}$.

4 Experimental Results

To evaluate the performance of the proposed scheme, we conducted experiments with VRML animation data of Wailer that are provided in 3DS-MAX sample animation. Wailer has 76 transform nodes and 100 frames. Each transform node has the different number of keys [0 1]. After randomly selecting 25 transform nodes, the watermark with 100bit length is embedded into coordIndex nodes and PositionInterpolator nodes of these transform nodes. Each of these selected transform nodes has 4bits of watermark in both coordIndex node and PositionInterpolator node. The robustness against various 3D animation attacks and the invisibility of the watermark are evaluated by the experiments. We use simple Signal to Noisy

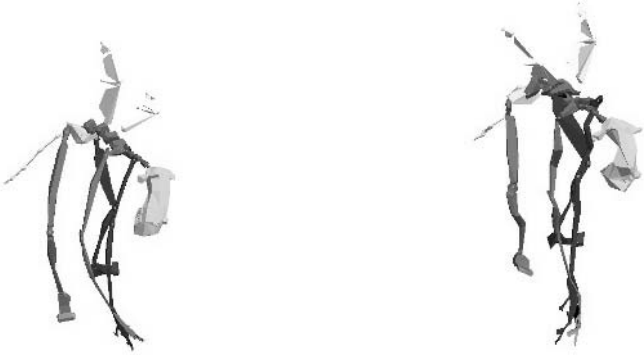
(a) (b)

Fig. 5. Wailer animation frame (a) Original 1st. frame and (b) Watermarked 1st. frame

ration (SNR) of vertex coordinates and keyValues for the invisibility evaluation. The SNR for our experiments is defined as $SNR = 10log_{10}\frac{var(||a-\bar{a}||}{var(||a-a'||}$ where a is the coordinate of a vertex or keyValue in a key time of original animation, $\bar{a}$ is the mean value of a, and a' is that of watermarked animation. $var(a)$ is the variance of a. The average SNR of the watermarked transform nodes is 38.8 dB at vertex coordinate and 39.1 dB at PositionInterpolator. The average SNR calculated for all transform nodes is about 39.5 dB at vertex coordinate and 42 dB at PositionInterpolator. Figure 5 shows the first frame of the original Wailer and the watermarked Wailer. As we can see from this Figure the watermark is invisible.

Currently available editing tools for 3D animation can be used to perform various mali-cious attacks to the animation: geometric attacks, timeline attacks, and format conversion. The geometric attacks change the coordinates and topology of the vertices in IndexFaceSet node. Well known geometric attacks are bend, taper, noise, patch deform, mesh smooth, HSDS modifier, polygon cutting/divider/extrude, vertex deletion, rotation, scaling, translation, and so on. Timeline attacks can alter the timeline scaling and the number of key times.

In this experiment, we evaluated the robustness of our proposed scheme against the geometric attacks and timeline attacks using 3DS-MAX tool. If the watermarked animations were attacked by geometric attacks, the watermark that embedded into PositionInterpolator can be extracted without bit error. On the other hand, if the moving position of the water-marked animations were changed by timeline attacks, the watermark can be extracted without bit error in CoordIndex. The experimental results of robustness against geometrical attacks and timeline attacks are summarized and shown in Table 1. Each parameter in table 1 represents the strength of an attack.



(a) (b)

Fig. 6. Two Geometric Attacks to Watermarked Wailer (a) Bend and (b) Noise

All transform nodes of watermarked animation were edited by the following geometric attacks. Bend has three parameters (angle, direction, bend axis) that bend an object angle and direction on the basis of a bend axis. Bit Error Ratio (BER) of watermark is 0.07, that is extracted in CoordIndex nodes of animation that are bended to (90, 22, z) as shown in Figure 6-(a). Taper has four parameters: amount, curve, primary, and effect. Taper makes an object become concave or convex shape on the basis of taper axis. An amount value and a curve value represent a taper level and the curve shape of tapered object. The primary is the center axis of taper and the effect is the axis direction of curve. BER of the extracted watermark is 0.06 in CoordIndex of animation that tapered to (1.2, 0.5, z, xy) in all transform nodes. Noise has the following parameters: seed, scale, roughness, iterations, and strength of x, y, z. An object can be made to be uneven shape irregularly by the strength and the seed. The scale is total size of noise and the iteration is the number of fractal effects. BER of the extracted watermark is 0.06 in CoordIndex of animation noised to (29, 200, 1, 6, 2, 2, 2) as shown in Figure 6-(b). Mesh smooth has two parameters: iteration, smoothness. These parameters can make the shape of an object smooth by dividing polygon surface. The iteration is the subdivision number of polygon. BER of the

watermark is 0.05 in CoordIndex node of animation that is subdivided to (1, 1.0) in all transform nodes. About 40%-50% of total numbers of polygons or vertices are cut. BERs of the watermark in these attacked animations are about 0.18-0.25.

Both key and keyValue of an interpolator can be used for timeline attacks. Rescaling time controls time length with frame count. If animation time is reduced by half of time length, the number of key is reduced by half and key is reordered in [0 1], as shown in Figure 7-(a). Down-scaled transform node in timeline may not have some reference keys for watermark extracting in PositionInterpolator. In such a case, the watermark has to be extracted after performing the rescaling process with another reference keys. BER of the watermark in animation with half-scaled timeline is 0.10 since the proposed scheme embeds the permuted watermark bit into x,y,z coordinates of transform node. PositionInterpolator can be double-scaled timeline. This interpolator has half as many key as original interpolator. Since all keys are still alive, the watermark can be extracted without bit error after re-scaling process. In key addition/deletion experiment, 20keys in interpolators of all transform nodes were added randomly into key positions or deleted randomly. BER of the watermark in key addition/deletion is about 0.03 since the area difference may be different because of the changed moving line. In motion change experiment, the motion of animation is changed on the base of arbitrary PositionInterpolator $\mathbf{M}'_{TR}$ that was generated randomly with 35 keys as shown in Figure 7-(b). Thus, $\mathbf{V}'_{world} = \mathbf{V}_{local} \times \hat{\mathbf{M}}'_{world}$ and $\hat{\mathbf{M}}'_{world} = \mathbf{M}_{TR} \times \mathbf{M}'_{TR}$. When original PositionInterpolator $\mathbf{M}_{TR}$ is unknown, the watermark has to be extracted from the changed PositionInterpolator $\hat{\mathbf{M}}_{TR}$. In this case, BER of the watermark is about 0.30 that the watermark can still alive about 70%.

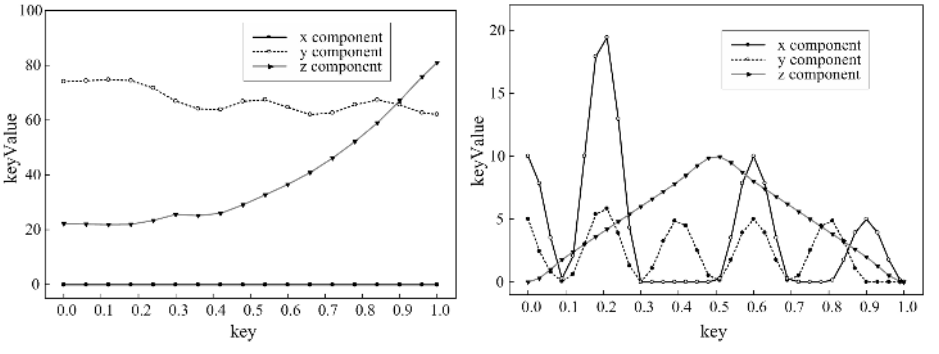


Fig. 7. PositionInterpolator in Bip transform node of (a) 50 frames and (b) PositionInterpolator for motion change

Table 1. The experimental results for robustness of the proposed scheme against various attacks

Attack		Parameter	BER	
			CoordIndex	Position-Interpolator
Geometrical attack	Bend	(90, 22, z)	0.07	-
	Taper	(1.2,0.5,z,xy)	0.06	-
	Noise	(29,200,1,6,2,2,2)	0.05	-
	Mesh Smooth	(1,1.0)	0.05	-
	Polygon Cutting	40%	0.21	-
	Polygon Extrude	50%	0.18	-
	Vertex deletion	50%	0.25	-
Time line attack	Down-scaling time	50 frame	-	0.10
	Up-scaling time	200 frame	-	-
	Key addition	20 keys	-	0.03
	Key deletion	20 keys	-	0.03
	Motion change		-	0.30

5 Conclusions

This paper presents a novel watermarking scheme for 3D keyframe animation based on geometric watermarking and interpolator watermarking. The geometric watermark-ing embeds the watermark into the distribution of vertex coordinates in each of initial mesh models. The interpolator watermarking embeds the same watermark in the geo-metric watermarking into key values in position interpolator. Experimental results showed that the proposed scheme has the robustness against various geometric attacks and timeline attacks, as well as the invisibility.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-042-D00225).

References

1. J. Cox, J. Kilian, T. Leighton, T. Shamoon: Secure spread spectrum watermarking for multimedia. IEEE Trans. on Image Processing, vol. 6. no. 12 (1997) 1673-1687
2. W. Zhu, Z. Xiong, Y.-Q. Zhang: Multiresolution watermarking for image and video. IEEE Trans. on Circuits and Systems for Video Technology, vol. 9, no. 4 (1999) 545-550

3. R. Ohbuchi, H. Masuda, M. Aono: Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modification. IEEE JSAC, vol. 16, no. 4 (1998) 551-560
4. O. Benedens: Geometry-Based Watermarking of 3D Models. IEEE CG&A, (1999) 46-55
5. S. Lee, T. Kim, B. Kim, S. Kwon, K. Kwon, K. Lee: 3D Polygonal Meshes Watermarking Using Normal Vector Distributions. in the Proceedings of IEEE International Conference on Multimedia and Expo, vol. III, no. 12 (2003) 105-108
6. K. Kwon, S. Kwon, S. Lee, T. Kim, K. Lee: Watermarking for 3D Polygonal Meshes Using Normal Vector Distributions of Each Patch. in the Proceedings of IEEE International Conference on Image Processing, (2003)
7. ISO/IEC 14772-1: VRML: The virtual reality modeling language
8. ISO/IEC14496-16: Information Technology - Coding of Audio-Visual Objects - Part 16: Animation Framework Extension (AFX)
9. E.S. Jang, James D.K.Kim, S.Y. Jung, M.-J. Han, S.O. Woo, S.-J. Lee: Interpolator Data Compression for MPEG-4 Animation. IEEE Trans. On Circuits and Systems for Video Technology, vol. 14, no. 7 (2004) 989-1008

A Robust Video Watermarking Scheme Via Temporal Segmentation and Middle Frequency Component Adaptive Modification

Liesen Yang[1,2] and Zongming Guo[1,2]

[1] Institute of Computer Science and Technology
100871 Peking University Beijing China
{yangliesen, guozongming}@icst.pku.edu.cn
[2] National Key Laboratory of Text Processing Technology
100871 Peking University Beijing China

Abstract. A robust video watermarking scheme via temporal segmentation and middle frequency component adaptive modification is presented. First, the video clip is sliced into a shot sequence by temporal video segmentation. Then, several consecutive shots are selected from the shot sequence to makeup a video segment with proper duration. The watermark is embedded into the video segment by equally dividing it into many units, and modifying the middle frequency component of frames in each unit. In addition, the scheme adjusts the modification strength of every coefficient adaptively according to the change between consecutive frames and Watson's DCT based visual model. Experimental results show that the scheme is resilient to many kinds of temporal desynchronization, spatial desynchronization and photometric distortion at the same time.

1 Introduction

Video watermarking is an important technique for digital video copyright protection. One of the most technical challenges for video watermarking is how to survive normal signal processing which happened in various applications. This is the problem of robustness of watermarking. The normal signal processing techniques related to video will lead to three categories of distortion: temporal desynchronization (such as frame dropping, frame inserting, frame rate converting, etc.), spatial desynchronization (such as rotating, scaling, translating, aspect ratio, clipping, shearing, bending, and perspective transforming etc.) and photometric distortion (such as lossy compressing, adding noise, amplitude changing, linear filtering, etc.) [1].

Basic ideas and methods for improving the robustness of watermarking include spread spectrum watermarking, redundant embedding, embedding in perceptually significant coefficients, embedding in coefficients of known robustness, etc. The methods aimed to resist spatial and temporal desynchronization include exhaustive search, autocorrelation, synchronization template, invariant watermarks, etc. [1]. Based on those ideas and methods, some new methods resilient

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 150–161, 2006.

to one or more of the three categories of distortions including temporal desynchronization, spatial desynchronization and photometric distortion have been proposed.

Watermarking along time axis is regarded as a perfect method to resist spatial desynchronization (i.e. geometric distortions). Haitsma et al. proposed a watermarking method based on modifying the mean luminance of every frame along time axis, which is robust against various geometric attacks, but cannot survive even simple temporal desynchronization [2]. Sang et al. proposed a watermarking method based on modifying the variance of every frame of the sequence. It is robust to geometric attacks, but is weak against temporal desynchronization [3].

Niu et al. proposed a method to embed watermark via modifying three points in every frame along time axis. It can resist many kinds of geometrical distortions, lossy compression, and frame dropping in certain rate [4]. Chen et al. proposed a method to embed watermark by modification of middle frequency component of every frame. It is robust against spatial desynchronization, and can resist temporal desynchronization to some extent [5]. Tsang et al. proposed an "add and subtract" watermarking scheme which utilize the characteristic of temporal redundancy of video sequence to improve the quality and robustness of the watermarked sequence [6].

Watermarking along time axis is also a good idea to resist temporal desynchronization. Lin et al. recovered temporal synchronization by introducing redundancy in the structure of the embedded watermark [7]. Sun et al. recovered the temporal synchronization based on profile statistics matching [8].

No frame dependent watermarking scheme above can resist frame conversion without changing duration. The watermarking scheme based on scene or shot can resist more distortions. Jung et al. proposed a scene based watermarking scheme in [9], which is robust against various kinds of temporal desynchronization and simple geometric transformation. However, the video quality degrades visibly due to the logarithmic mapping of frames. Sun et al. proposed an ICA and temporal segmentation based watermarking scheme [10], which is robust against many kinds of temporal desynchronization and common photometric distortion such as lossy compression. However, the ability resilient to spatial desynchronization depends on the image watermarking methods used in this scheme. The watermarking schemes based on 3D DFT, 3D DCT or 3D DWT can resist some kinds of distortions, but the computational complexity is very high [11,12,13].

Harmanci et al. proposed a scheme, which is robust against geometrical distortion to some extent [14], and can resist temporal desynchronization such as frame insertion by interpolating, scene editing, cutting, and swapping. However, it needs side information for watermark detection.

A robust video watermarking scheme is presented in this paper. It can resist spatial desynchronization by embedding watermark along time axis via modifying middle frequency components of every frame, and it can recover temporal synchronization by redundant embedding, temporal video segmentation and exhaustive searching among the shot boundaries within a short range. In addition, it can resist photometric distortion by adjusting the modifying strength

adaptively according to the change between consecutive frames and Watson's DCT based visual model.

The rest of this paper is organized as follows. In section 2, we give an overview of the proposed scheme. In section 3, we describe something about temporal segmentation. The watermarking algorithm for a video segment with proper duration is described is section 4. In section 5, we describe the watermarking algorithm for a video program. Experimental results are demonstrated in section 6 to verify the performance of the proposed scheme. Finally, conclusion is given in section 7.

2 Overview

We will integrate three kinds of techniques to resist the three categories of distortions respectively. In the rest of the paper, we refer DCT to the 8×8 block DCT of luminance of a frame.

2.1 Watermarking Along Time Axis to Resist Spatial Desynchronization

Let M be the total number of 8×8 blocks in one frame, and $B(p,m)$ be the $m\text{-}th$ block of the $p\text{-}th$ frame, $m = 0, \ldots, M-1$. Let $DCT(p,m,i)$ be the $i\text{-}th$ coefficient in the DCT of $B(p,m)$ in $zigzag$ order, $i = S, \ldots, T$, $0 < S < T < 63$. Let E_m be the mean value of the middle-frequency energy of DCT of all blocks in one frame, and let $E_m(p)$ be the E_m of the $p\text{-}th$ frame.

$$E_m(p) = \frac{1}{M} \sum_{m=0}^{M-1} \sum_{i=S}^{T} |DCT(p,m,i)| \tag{1}$$

Due to consecutive frames in one shot are close to each other, the $E_m(p)$ is close to $E_m(p-1)$ and $E_m(p+1)$. Let $\delta(p)$ be the prediction error of $E_m(p)$ by $E_m(p-1)$ and $E_m(p+1)$

$$\delta(p) = E_m(p) - \frac{E_m(p-1) + E_m(p+1)}{2} \tag{2}$$

Then $\delta(p)$ is near to zero. On the other hand, $E_m(p)$ can be significantly changed with precondition of fidelity preserving by modifying $DCT(p,m,i)$ in certain way. Then the absolute value of $\delta(p)$ will be significantly changed and be far from zero, one watermark bit can be embedded into the three frames. The watermark bit can be extracted according to the sign of $\delta(p)$.

Furthermore, the sign of $\delta(p)$ is nearly unchanged under sustainable geometric distortion, so the watermark can survive spatial desynchronization.

2.2 Resist Temporal Desynchronization Via Temporal Segmentation and Exhaustive Search

As stated in section 1, none of the frame dependent watermarking scheme can resist frame conversion without changing duration. So does the method described

in subsection 2.1. In order to resist temporal desynchronization such as frame rate conversion without changing duration, we can substitute three consecutive video units for the three frames used to embed one watermark bit in subsection 2.1. The three consecutive video units corresponding to the bit embedded will survive frame rate conversion without changing duration, and so the watermark bit embedded can be extracted from those video units.

However, how can we determine the start frame and the end frame of every unit used for watermarking?

The shot boundary is an important feature for video, and it can survive normal processing, so the shot boundary can be used to recover the synchronization of every unit used for watermarking. If the length of a watermark is 64, and the duration of one unit is 0.2 second, then $64 \times 3 = 192$ units with duration of $192 \times 0.2 = 38.4$ seconds, will be needed to embed a watermark. However, the duration of a video shot is usually less than 30 seconds, which is too short to embed a watermark. We can select several consecutive shots to makeup a longer subsequence to embed a watermark. Of course, exhaustive searching among the shot boundaries in a short range is needed to recover the actual interval for watermark embedding, and the search space is very small.

Let the watermark after encoding, spreading spectrum and modulating be a pseudo-random sequence $w(n)$ of length N where $w(n) \in \{1, -1\}$, $n = 0 \sim N-1$, then for a video segment with proper duration, we can equally divide it into $3N$ units along time axis. The numbers accorded to each unit are set in consecutive order, starting with number 0.

Let $g(k)$ be the mean value of the E_m of frames in the *k-th* unit.

$$g(k) = \frac{1}{Q_k} \sum_p E_m(p) \tag{3}$$

where Q_k is the total number of frames in the *k-th* unit. Due to consecutive frames in one shot are close to each other, the $g(3n+1)$ is close to $g(3n)$ and $g(3n+2)$. Let $\varepsilon(n)$ be the prediction error of $g(3n+1)$ by $g(3n)$ and $g(3n+2)$.

$$\varepsilon(n) = g(3n+1) - \frac{g(3n) + g(3n+2)}{2} \qquad n = 0 \sim N-1 \tag{4}$$

Then $\varepsilon(n)$ is near to zero. The watermark bit $w(n)$ can be embedded into the $(3n)$-*th*, $(3n+1)$-*th*, $(3n+2)$-*th* three units by modifying the E_m of frame in each unit respectively to shift $\varepsilon(n)$ far away from zero and $\varepsilon(n) < 0$ if $w(n) = -1$, $\varepsilon(n) > 0$ if $w(n) = 1$.

2.3 Adjust Modifying Strength Adaptively to Resist Photometric Distortion

Due to *persistence of vision*, the resolving power of human eyes degrades quickly when consecutive frames change acutely. Therefore, frames that change acutely can tolerate more modification with precondition of keeping the modification imperceptible. In addition, according to the DCT frequency sensitivity table

of Watson's DCT based visual model [1], the sensitivity of each coefficient is different with each other. We can adjust the modifying strength of each DCT coefficient adaptively.

3 Temporal Video Segmentation

Temporal video segmentation,which is also referred as shot boundary detection, is to slice the video sequence into many shots. There are many methods to detect shot boundary such as methods based on comparison of pair of pixels, template comparison, histogram comparison, etc. [15]. The DC-image based method proposed in [16] by Ye et al. is the easiest method in compressed domain to implement and the best method for cut detection [15]. In addition, the data size of DC-image is far less than that of a video frame, and the DC-image is insensitive to noise, and easy to get from the DC terms of DCT transformation. Therefore, we choose this method to detect shot boundary in the proposed watermarking scheme.

4 Watermarking for a Video Segment with Proper Duration

As described in subsection 2.2, the watermark is supposed to be a pseudo-random sequence $w(n)$ of length N, where $w(n) \in \{-1, 1\}$, $n = 0 \sim N-1$. For a video segment with proper duration, we can equally divide it into $3N$ units along time axis. The numbers accorded to each unit is set in consecutive order, starting with number 0.

4.1 Watermark Embedding

If $w(n) = -1$, we can decrease E_m of every frame in the $(3n+1)$-*th* unit and increase E_m of every frame in the $(3n)$-*th* and $(3n+2)$-*th* units to embed $w(n)$ into these three units in terms of Equation (2). If $w(n) = 1$, we can do oppositely. Two steps are needed to embed a watermark bit.

Step 1. Modify the middle frequency coefficients of frames in the $(3n+1)$-*th* unit.

Suppose frame p belongs to the $(3n+1)$-*th* unit. First, some of the middle frequency coefficients will be selected for modifying in the light of the watermarking key. Let $DCT(p,m,i)'$ be the new value of $DCT(p,m,i)$ after modifying.

$$DCT(p,m,i)' = DCT(p,m,i) + \Delta(p,m,i) \tag{5}$$

where $\Delta(p,m,i)$ is the modification computed by Equation (6).

$$\Delta(p,m,i) = \begin{cases} 0 & \text{if } |DCT(p,m,i)| <= T_A \\ \lambda(i)w(n) & \text{if } T_A < DCT(p,m,i) < T(p,m) \\ -\lambda(i)w(n) & \text{if } -T(p,m) < DCT(p,m,i) < -T_A \\ \frac{DCT(p,m,i)}{T(p,m)}\lambda(i)w(n) & \text{if } T(p,m) \le |DCT(p,m,i)| \end{cases} \tag{6}$$

where $\lambda(i)$ is the *i-th* sensitivity coefficient of Watson's DCT frequency sensitivity table in *zigzag* order, T_A is a constant, $T(p,m)$ is a linear combination of two constants T_m and T_M, such that $T_A \leq T_m \leq T(p,m) \leq T_M$. $T(p,m)$ is computed as follows

$$T(p,m) = \alpha(p,m)T_m + (1-\alpha(p,m))T_M \tag{7}$$

where $\alpha(p,m)$ is the linear combination coefficient,and it is determined by three variables: the change between frames p and $p-1$, the change between block $B(p,m)$ and $B(p-1,m)$, and the middle frequency energy of block $B(p,m)$.

In order to keep the watermark imperceptible, we clamp $\Delta(p,m,i)$ by the interval $[-C(p,i), C(p,i)]$, where $C(p,i)$ is an experimental function.

$$\begin{aligned} C(p,i) &= \lambda(i) \cdot max\Big(min(6, \frac{D(p)}{3}), \phi(i)\Big) \\ \phi(i) &= \begin{cases} \lambda(i) & \text{if} \quad i \leq 5 \\ max(\lambda(i), 2) & \text{if} \quad i > 5 \end{cases} \end{aligned} \tag{8}$$

where $D(p)$ is the change between frames p and $p-1$, and can be determined by the following equation.

$$\begin{aligned} D(p) &= \frac{1}{M} \sum_{d(p,x,y) \geq 5} d(p,x,y) \\ d(p,x,y) &= |f(p,x,y) - f(p-1,x,y)| \end{aligned} \tag{9}$$

where $f(p,x,y)$ is the intensity value of the pixel at the coordinates (x,y) in the DC-image of the luminance of frame p.

Step 2. Modify the middle frequency coefficients of frames in the $(3n)$-*th* and $(3n+2)$-*th* units .

We can modify the middle frequency coefficients of frames in the two units according to method described in *step 1.* with substituting $-w(n)$ for $w(n)$ in Equation (6).

Using the two steps above, we can embed every watermark bit.

4.2 Watermark Detection

First, we extract the feature vector $\mathbf{V} = (v(0), v(1), \ldots, v(n), \ldots, v(N-1))$ according to Equation (10).

$$v(n) = \begin{cases} 1 & \text{if} \quad \varepsilon(n) > \varepsilon_0 \\ -1 & \text{if} \quad \varepsilon(n) < -\varepsilon_0 \\ 0 & \text{if} \quad -\varepsilon_0 \leq \varepsilon(n) \leq \varepsilon_0 \end{cases} \tag{10}$$

where $n = 0 \sim N-1$, ε_0 is a small positive constant, and $\varepsilon(n)$ is computed according to Equation (4).

Then, we compute the normalized correlation of feature vector $\mathbf{V}$ and watermark vector $\mathbf{W}_r = (w(0), w(1), \ldots, w(n), \ldots, w(N-1))$.

$$\tau = \frac{\mathbf{V} \cdot \mathbf{W}_r}{|\mathbf{V}|\,|\mathbf{W}_r|} \tag{11}$$

If $\tau \geq \tau_{nc}$, where τ_{nc} is the threshold for watermark detection, then a watermark is detected.

5 Watermarking Scheme for a Video Program

For a given video program, temporal segmentation is needed to slice the video program into a shot sequence before watermark embedding or detection. We can select several consecutive shots from the shot sequence to makeup a video segment with proper duration, so the method described in section 4 can be used to embed or detect watermark. Suppose the minimum duration of the video segment into which a watermark can embed is L_{min} seconds.

5.1 Watermark Embedding

We pop shot from the head of the shot sequence and append it to the end of the video segment for watermarking embedding until the duration of the video segment is not less than L_{min}, then a watermark can be embedded into it using the method described in subsection 4.1. After that delete all shots in the video segment, and embed watermark using the method described above until the shot sequence is empty (Fig.1).

5.2 Watermark Detection

After watermarking, a video program may suffer from some kinds of temporal desynchronization such as randomly frame inserting, frame dropping and frame rate conversion without changing frame. All those distortions may change the duration of the video sequence. The duration of a video may change from L_{min} to a value within interval $[D_{min}, D_{max}]$, where $0 < D_{min} < L_{min} < D_{max}$. D_{min} and D_{max}can be determined according to the requirement of applications.

When watermark detection process starts, we append the shot popped from the head of the shot sequence to the end of the video segment used for watermarking detection continuously until the duration of the video segment is not less than D_{min}. Then we check if the video segment is watermarked using the method described in subsection 4.2. If a watermark is detected, then output the results and delete all the shots in the video segment. After that check if the rest shot sequence is watermarked until it is empty. If no watermark is detected, then append the shot popped from the head of the shot sequence to the end of the video segment, until a watermark is detected or the duration of the video segment is larger than D_{max}. If the duration of the video segment is larger than

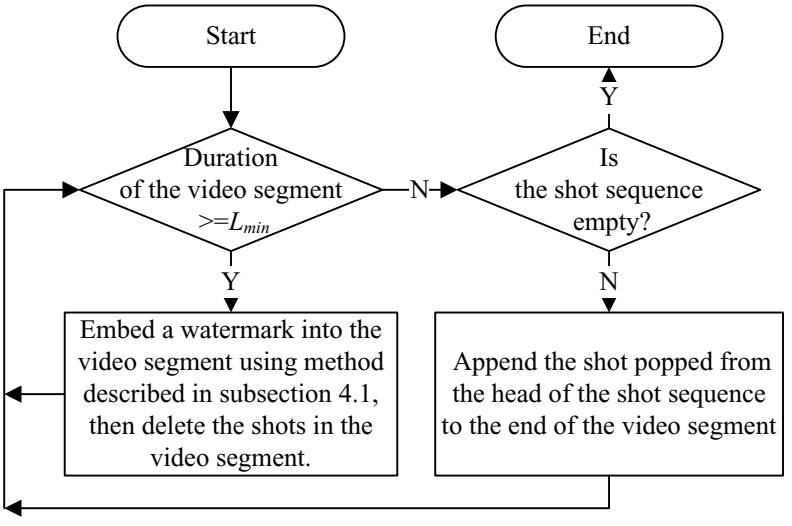


Fig. 1. Watermark embedding method for a video program

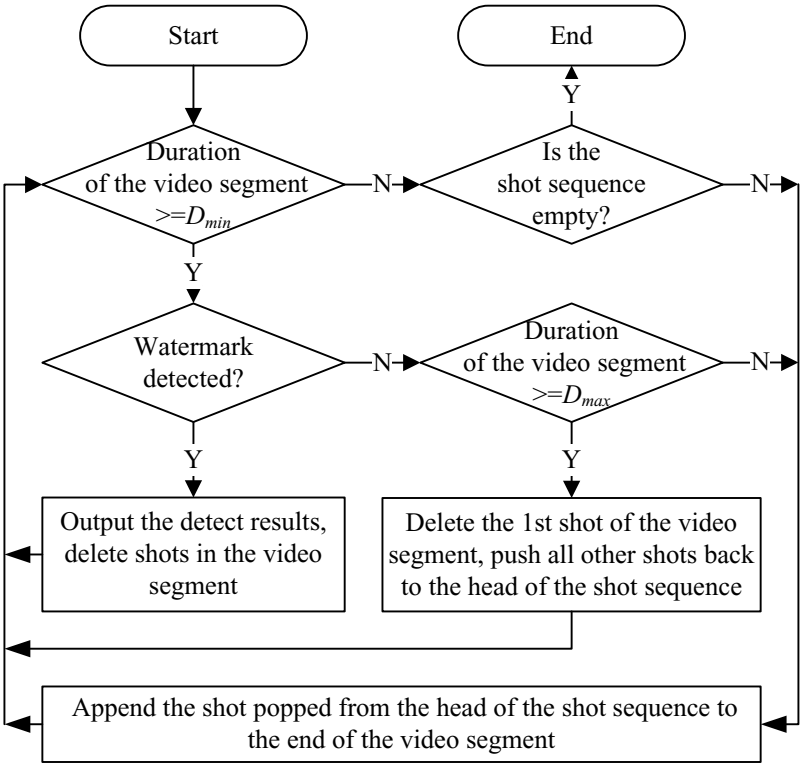


Fig. 2. Watermarking detection method for a video program

D_{max} but no watermark is detected, then delete the first shot of the video segment, and push all other shots back to the head of the shot sequence. Then we check if the rest shot sequence is watermarked using the method described above until it is empty (Fig.2).

6 Experimental Results

Three kinds of video programs and 33 kinds of distortions are selected to test the performance of the scheme presented in this paper. Clip VS_A is selected from TV program "News Collection" produced by CCTV. Clip VS_B is selected from 2D cartoon film "The Prince of Egypt" produced by DreamWorks. Clip VS_C is selected from the part of Normandy landing battle in film "Saving Private Ryan" produced by DreamWorks and Paramount Pictures, in which frames change acutely. The duration of each clip is 8 minutes or so, the frame rate of each clip is 25 frames per second (fps).

In our experiments, the threshold for watermark detection $\tau_{nc} = 0.6$, the length of watermark $N = 64$, the minimum duration of the video segment into which a watermark can embed $L_{min} = 0.2 \times 3 \times N = 38.4s$, $D_{min} = 0.7 \times L_{min}$, $D_{max} = 1.8 \times L_{min}$.

All the 33 kinds of distortions to be tested include several kinds of temporal desynchronization, spatial desynchronization, photometric distortion and their

Table 1. Denotations

Symbol	Stands for
TD	Temporal Desynchronization
SD	Spatial Desynchronization
PD	Photometric Distortion
T1	Frame inserting randomly up to 20%
T2	Frame dropping randomly up to 20%
T3	Frame rate converting to 30fps without changing duration
T4	Frame rate converting to 15fps without changing duration
T5	Frame rate converting to 20fps without changing frame
T6	Frame rate converting to 30fps without changing frame
S1	Scaling(XY:1/2)
S2	Scaling(XY:2)
S3	Scaling(X:1/2,Y:5/4)
S4	Rotating 5 degrees
S5	Rotating 45 degrees
S6	Perspective transformation
S7	Clipping to a center quarter (lost 75% of the watermarked image of a frame), then scaling to the original size
P1	Adding noise (FFDShow Noise: Luminance noise + Flickering + Shaking + Vertical line + Dust + Scratch)
P2	Amplitude changing (16∼128 to 64∼240)
P3	Linear filtering (Gaussian Filtering, radius = 5)
P4	Lossy compressing(VCD Quality: MPEG-1, 1.15Mbps)

Table 2. Watermarks detected under distortions

Clip		VS_A	VS_B	VS_C
Watermarks Embedded		9	10	11
TD	T1	9	10	11
	T2	9	10	11
	T3	9	10	11
	T4	9	10	8
	T5	9	10	11
	T6	9	10	11
SD	S1	9	10	11
	S2	9	10	11
	S3	9	10	11
	S4	9	10	11
	S5	9	10	11
	S6	9	10	11
	S7	9	10	8
PD	P1	9	10	11
	P2	9	10	11
	P3	9	10	11
	P4	9	10	11
TD + SD	T5+S3	9	10	11
	T3+S5	9	10	8
	T1+S6	9	10	8
	T2+S4	9	10	8
TD + PD	T4+P1	9	8	8
	T2+P2	9	9	8
	T6+P3	9	10	11
	T1+P4	9	10	9
SD + PD	S3+P1	9	3	10
	S5+P2	9	9	5
	S6+P3	9	8	7
	S4+P4	9	6	7
TD + SD + PD	T5+S3+P1	9	3	8
	T2+S5+P3	9	6	4
	T4+S6+P2	9	8	5
	T1+S4+P4	9	3	7

combined distortions. The denotations are shown in table 1. The experimental results are shown in table 2 and 3.

From table 2 we can see that the detection rate of all the three clips is nearly 100% when only one kind of distortion (T1∼T6, S1∼S7, P1∼P4) occurs. It shows that the proposed scheme is robust against temporal desynchronization, spatial desynchronization and photometric distortion.

Furthermore,we can see that when some kinds of combined distortions happen, the detection rate is high. It shows that the proposed scheme is robust against combined distortions of the three categories of distortions.

Table 3. Video quality after watermarking

Clip	VS_A	VS_B	VS_C
Average PSNR (dB)	52.05	54.32	52.18
Average MSE	0.45	0.32	0.57
Average RMSE	0.65	0.53	0.69

For clip VS_A , the detecting rate is 100% under all the 33 test terms. For clip VS_B and VS_C, the detecting rate is 100% under 23 and 17 test terms respectively. This shows that the proposed scheme is suitable for many kinds of video clips.

From table 3 we can see, the average PSNRs of the three watermarked clips are larger than 52dB, which means that the proposed scheme can preserve video quality very well.

Compared with the existing methods, it can resist more kinds of temporal desynchronization than those frame dependent methods proposed in [2,3,4,5], and can resist more kinds of spatial desynchronization than the methods proposed in [7,8] and "scene" or "shots" based method proposed in [9,10]. It can resist more kinds of distortions than those 3D transformation based methods [11,12,13] with lower computational complexity. In addition, it can detect watermark blindly, which is contrast with the method proposed in [14].

On the other hand, the embedding rate of the proposed scheme is lower than the frame dependent or shot based methods.

From all test results above we can see, the proposed scheme can resist many kinds of temporal desynchronization, spatial desynchronization, photometric distortion and combined distortions.

7 Conclusion

A robust video watermarking scheme based on temporal segmentation and middle frequency component adaptive modification is proposed in this paper. The scheme can resist spatial desynchronization by embedding watermark along time axis via modifying middle frequency components of every frame, and it can recover temporal synchronization by redundant embedding, temporal video segmentation and exhaustive search among the shot boundaries in a short range. In addition, the proposed scheme can resist photometric distortion by adjusting the modifying strength adaptively according to the change between consecutive frames and Watson's DCT based visual model. Experimental results show that the scheme is robust against many kinds of temporal desynchronization, spatial desynchronization and photometric distortion and some of their combined distortions. Compared with the existing methods, the proposed scheme can resist more kinds of distortions. On the other hand, the embedding rate of the proposed scheme is lower than some of the existing methods. In our experiment, the detection rate is not very high for video clips VS_B and VS_C under some combined distortions. Research on improving the detection rate under those combined distortions for video program like VS_B and VS_C becomes future tasks.

References

1. Ingemar J. Cox, Matthew L. Miller and Jeffrey Bloom. "Digital watermarking". *San Francisco, Calif: Morgan Kaufmann*, 2002.
2. Jaap Haitsma, Ton Kalker. "A watermarking scheme for digital cinema". In: *Proceedings of the IEEE International Conference on Image Processing*. 2001, 487–489.
3. Sang Maodong, Yang Wenxue, Zhao Yao. "Time-axis based video watermarking resisting to geometrical attacks". In: *Proceedings of the IEEE International Conference on Signal Processing*, Vol. 3. 2004, 2350–2353.
4. X.M. Niu, M.Schmucker, C.Busch, S.H. Sun. "A video watermarking against geometrical distortions". *Chinese Journal of Electronics*. 2003, 12(4):548–552.
5. Chen Zhenyong, Deng Junhui, Tang Long, Tang Zesheng. "A novel video watermarking resistant to spatio-temporal desynchronization". In: *Proceedings of the 12th National Conference on Image and Graphics (NCIG'2005), Beijing, China, Tsinghua University Press*. 2005, 161–166(in Chinese)
6. K. F. Tsang , Oscar C. L. Au. "Robust and high-quality video watermarking with the use of temporal redundancy". In: *Proceedings of SPIE Vol. 4314, Security and Watermarking of Multimedia Contents III*. 2001, 55-63
7. Eugene T. Lin, Edward J. Delp. "Temporal synchronization in video watermarking". In:*Proceeding of the SPIE Vol. 4675, Security and Watermarking of Multimedia Contents IV*. 2002, 478–490.
8. Shih-Wei Sun, Pao-Chi Chang. "Video watermarking synchronization based on profile statistics". *IEEE Aerospace and Electronic Systems Magazine*. 2004, 19(5): 21–25
9. Hans Jung, Y.Y. Lee, Suk Lee. "RST-resilient video watermarking using scene-based feature extraction". *EURASIP Journal on Applied Signal Processing*. 2004, 14:2113–2131.
10. Jiande Sun, Ju Liu, Huibo Hu, Tao Luo. "An ICA and temporal segmentation composite video watermarking scheme". In: *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*. 2004, 286-289.
11. Young-Yoon Lee, Han-Seung Jung, Sang-Uk Lee. "3D DFT-based video watermarking using perceptual models". In: *Proceedings of the 46th IEEE International Midwest Symposium on Circuits and Systems*, Vol. 3. 2003, 1579-1582.
12. Jie Yang, Junping Hu, Ping Liu. "Video watermarking by 3D DCT". In: *Proceedings of the IEEE International Conference on Signal Processing*, Vol. 1. 2004, 861–864.
13. Campisi, P. Neri. "A video watermarking in the 3D-DWT domain using perceptual masking". In: *Proceedings of the IEEE International Conference on Image Processing*, Vol. 1. 2005, 997–1000.
14. Oztan Harmanci, Mehmet Kucukgoz, M. Kivanc Mihcak. "Temporal synchronization of watermarked video using image hashing". In: *Proceedings of the SPIE Vol.5681, Security, Steganography, and Watermarking of Multimedia Contents VII*. 2005, 370–380.
15. Irena Koprinska, Sergio Carrato. "Temporal video segmentation: a survey", *Signal Processing: Image Communication*. 200l, 16(5):477–500
16. Boon-Lock Ye, Bede Liu. "Rapid scene analysis on compressed video", *IEEE Transactions on Circuits & Systems for Video Technology*. 1995, 5(6):533–544.

Capacity Enhancement of Compressed Domain Watermarking Channel Using Duo-binary Coding

Ivan Damnjanovic and Ebroul Izquierdo

Queen Mary, University of London, Department of Electronic Engineering,
Mile End Road, London, E1 4NS, United Kingdom
{ivan.damnjanovic, ebroul.izquierdo}@elec.qmul.ac.uk

Abstract. One of the main goals of watermarking is to optimize capacity while preserving high video fidelity. This paper describes a watermarking scheme based on the spread spectrum paradigm with capacity enhancement using a state-of-the-art error correction technique – duo-binary turbo coding. A new watermark composition with novel bit-wise watermark bits interleaving scheme and bit-rate control on the macro-block level is proposed. In previous works, the perceptual watermark adjustment was mainly based on Watson Just Noticeable Difference (JND) model. A new JND estimation model based on block classification is presented. In addition, experimental results on perceptibility and robustness to transcoding are reported.

Keywords: Digital watermarking, MPEG2, turbo coding, JND model, block classification.

1 Introduction

The huge expansion of digital production and processing technology and the Internet, have made possible to distribute and share unlimited quantities of digital material by anyone, anytime and anywhere. Digital watermarking arose as a possible solution to not only inherent copyright issues, but also a range of other interesting applications such as authentication, broadcast monitoring and data embedding.

Looking at real-time applications, several techniques have been reported in the literature aiming at watermarking in the compressed domain. Many of these are based on embedding a watermark into a video sequence using the spread spectrum paradigm. Hartung and Girod proposed to extend their technique for spread spectrum watermarking in uncompressed domain to compressed domain [1]. The watermark, consisted of a sequence of bits {-1,-1}, is spread by a large chip-rate factor, modulated by pseudo-random sequence and then organized into frames, which are the same size as video frames. To embed watermark to a DCT block of the video frame, an 8x8 block from the same relative position in the watermark frame is DCT transformed and added to the block. If watermarking of an AC coefficient yields a Huffman code (n_1) that is longer then the original unwatermarked coefficient (n_0), watermarking is discarded for that coefficient in order to preserve the bit-rate of the sequence.

This technique had the major impact on the digital video watermarking research. Many authors were influenced by the work of Hartung and Girod: [2], [3], [4], [5].

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 162 – 176, 2006.

Chung et al. in [2] proposed to use direct sequence spread spectrum method during MPEG-2 compression process. The authors introduced a new model for perceptual adjustment based on block classification in DCT domain according to its edginess and texture energy. Simitopoulos et al. proposed improved perceptual adjustment model and to add watermark bits to quantized DCT coefficients [3]. This watermarking scheme was reported to be able to withstand attacks such as transcoding, and filtering, and even geometric attacks, if methods for reversing such attacks are incorporated. Ambroze et al. [4] also based their model on the Hartung-Girod technique and examined the capacity improvement provided by forward error control (FEC) coding and perceptual adjustment based on Watson JND model. They found that using multiple parallel concatenated convolutional codes (3PCCCs) typically gives 0.5 Kbps payload under a combined compression and geometric attack. Pranata et al. in [5] proposed bit-rate control scheme that evaluates the combined bit lengths of a set of watermarked VLC codewords, and successively replaces watermarked coefficients that introduced the largest increase of the set length with the corresponding original coefficients until a target bit length is achieved.

This paper is divided into five sections. Section 2 gives brief description of implemented scheme with special attention to a new block-wise random watermark bits interleaving and a novel method for bit-rate preservation called bit-rate control on the macro-block level. Detailed description of a new perceptual adjustment model is given in section 3. Section 4 presents in-depth analysis of error correction coding (ECC) applicability to the given watermarking model. Achievable capacity rates using ECC for a given signal-to-noise ratio in the channel are first theoretically analysed, protection of the watermarking channel with a state-of-the-art duo-binary turbo coder is proposed and the robustness of the technique to transcoding is evaluated. Section 5 provides the conclusion of the paper.

2 MPEG2 Watermarking in DCT Domain

The targeted application is data embedding and indexing in professional environments where intentional attacks are not expected, so the watermark needs to be robust against typical video editing processes, such as transcoding, logo insertion, cross-fade etc. Hence, focus was given to requirements related to high imperceptibility and the trade off between imperceptibility and watermark capacity. A minimum duration of the watermarking video segment from which it will be possible to extract the watermark is often limited by a time window of 5 seconds [6]. For the MPEG2 standard in PAL sequences it can be seen as 8 I frames. In this way only 8 frames needs to be processed at the watermark decoder. Due to temporal compression, the embedding space in inter-frames is considerably low, so this can be seen as reasonable trade-off between the payload and the computational cost.

The principle of the watermarking scheme used in this work, is given in Figure 1. The scheme is based on the popular spread-spectrum paradigm with novel bit-wise watermark bits interleaving scheme. Each of the n watermark message bits is repeated 64 times to form 8x8 block. These blocks are then randomly spread through 8 watermarking frames and then modulated by pseudo sequence [7]. In that way, every watermark bit has almost the same Signal-to-Noise ratio and detection

probability, since the bits are evenly spread through textured, edge and plain areas. In addition, with the interleaved bit spreading, distribution of detection values can be approximated with normal distribution giving a much easier way for theoretical analysis of the watermarking system performance. Before embedding the watermark to DCT coefficient, its amplitude is adjusted using the information from corresponding DCT block in the original sequence.

The MPEG-2 video bit-stream is divided in the packets. Every packet has size of the packet written in the packet header. Concerning the bit-rate, we can change coefficients as much as we like as long as the size of the packet is not altered. Hence, DCT coefficients that have Huffman code n_1 bigger then original n_0 can be written in the new bit-stream if the amount in which they increase bit-rate is the same as that the amount of other coefficients decrease bit-rate. We are proposing bit-rate control on the macro-block level. If the watermarked macro-block size is bigger then the size of unwatermarked macro-block, watermarked AC coefficients with largest VLC difference is swapped with the original ones till the macro-block size is smaller or equal to the original one. Further dissemination and comparison with Hartung and Girod approach can be found in [7].

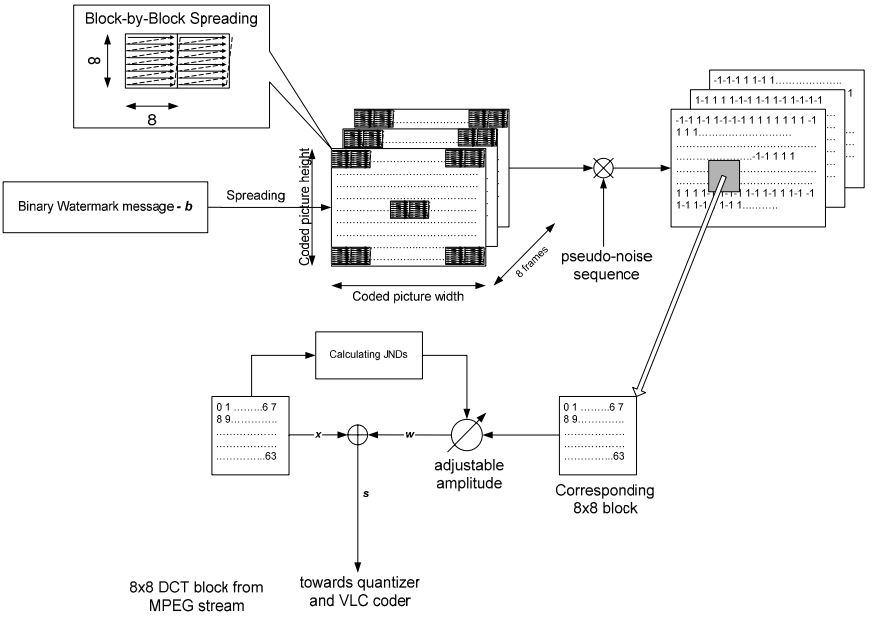


Fig. 1. Watermarking embedding scheme

3 Perceptual Adaptation in DCT Domain

One of the main watermarking requirements in a professional environment is high imperceptibility of the watermark and induced distortions. The JND model used in

this work tends to exploit three basic types of phenomena: non-uniform frequency response of human eye (contrast sensitivity - t_{CSF}), sensitivity to the different brightness levels (luminance masking - t_l) and sensitivity to one frequency component in the presence of another (contrast or texture masking - t_c):

$$t_{JND}(n_1,n_2,i,j)=t_{CSF}(i,j)\times t_l(n_1,n_2)\times t_c(n_1,n_2,i,j) \tag{1}$$

where the indices n_1 and n_2 show the position of the 8x8 DCT block in the image or the video frame, while i and j represent position of the coefficient in the DCT-block.

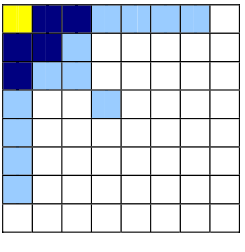
The visibility threshold $t_{CSF,}$ as a function of spatial frequency response in specific viewing conditions, is usually derived by the model presented in [8]. These thresholds for pre-determined viewing conditions can be also given in 8x8 contrast sensitivity table as the one used in our experiments and reproduced from [9].

The human visual system's sensitivity to variations in luminance is dependent on the local mean luminance. Zhang et al. in [10] argued that Watson model over-simplifies the viewing conditions for practical images. They stated that gamma-correction of the display tube and ambient illumination falling on the display partially compensate effect of Weber-Fechner's law and as a result give higher visibility thresholds in either very dark or very bright regions, which Watson model fails to acknowledge. Hence, they approximate luminance thresholds with two functions, for low region ($L\leqslant 128$) and for high region of luminance ($L>128$):

$$t_l(n_1,n_2)=\begin{cases} k_1\left(1-\dfrac{C(n_1,n_2,0,0)}{128}\right)^{\lambda_1}+1 & \text{if} \quad C(n_1,n_2,0,0)\leqslant 128 \\ k_2\left(\dfrac{C(n_1,n_2,0,0)}{128}-1\right)^{\lambda_2}+1 & \text{otherwise} \end{cases} \tag{2}$$

where k_1=2, k_2=0.8, λ_1=3, λ_2=2.

To evaluate the effect of contrast masking more accurately, it is essential to classify DCT blocks according to their energy. It is well known fact that noise is less visible in the regions where texture energy is high and it is easy to spot in smooth areas. On the other hand, HVS is sensitive to the noise near a luminance edge in an image, since the edge structure is simpler then texture one and a human observer have better idea about edge look. Hence, block classification will lead to better adaptation of the watermark to different part of the image.



	DC
	LR (low frequency)
	ER (edge)
	HR (high frequency)

Fig. 2. DCT block classification

First, an 8x8 block is divided into four areas shown in Figure 2 and the absolute sums of the DCT coefficients in the areas are denoted with DC – mean block luminance, LR – low frequency region, ER – edge region and HR for high-frequency region.

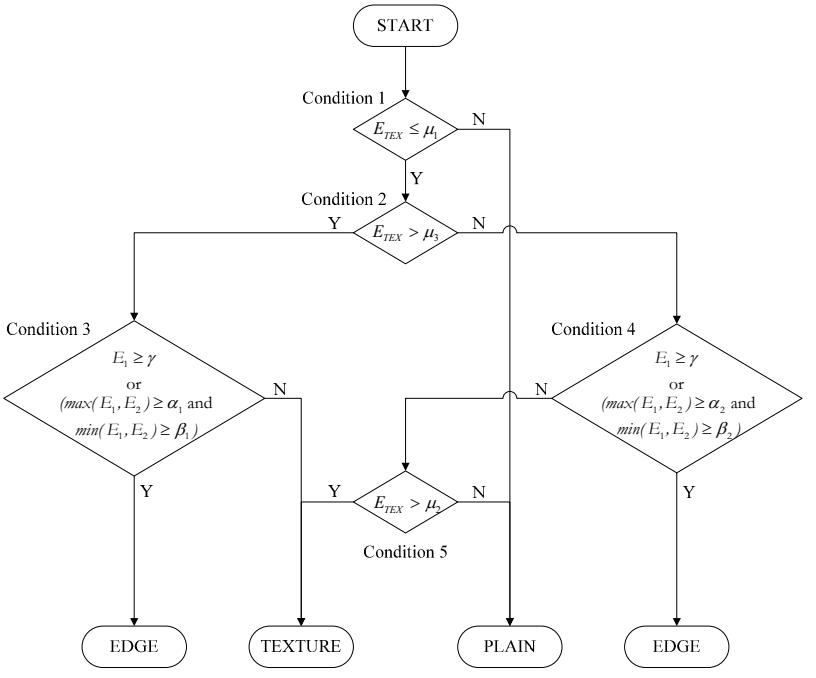


Fig. 3. Block classification algorithm

The texture energy of the DCT block can be approximated by:

$$E_{TEX} = E + H \tag{3}$$

where E and H represent the sums of the absolute values of DCT coefficients in ER and HR regions respectively. Since information about edges is reflected in LR and ER portions and texture is mainly reflected in HR portion, it was determined that high magnitudes of the following two ratios indicate the presence of an edge:

$$E_1 = \frac{\overline{L} + \overline{E}}{\overline{H}} \tag{4}$$

$$E_2 = \frac{\overline{L}}{\overline{E}} \tag{5}$$

where $\overline{L}$, $\overline{E}$ and $\overline{H}$ denote mean energies in low-frequency, edge and high-frequency blocks respectively. Energies E_{TEX}, E_1 and E_2 blocks are used in block classification algorithm given in Figure 3 [10] to derive block class, where α_1=0.7, β_1=0.5, α_2=7, β_2=5, γ =16, μ_1=125, μ_2=290, μ_3=900.

According to the block class and its texture energy, the inter-band elevation factor is derived using the following formula:

$$\xi(n_1,n_2) = \begin{cases} 1+1.25\cdot\dfrac{E_{TEX}(n_1,n_2)-\mu_2}{2\mu_3-\mu_2} & \text{for TEXTURE block} \\ 1 & otherwise \end{cases} \tag{6}$$

To consider the intra-band masking effect Watson's contrast masking model was used:

$$t_C(n_1,n_2,i,j) = \xi(n_1,n_2)\cdot max\left\{1,\left(\frac{|C(n_1,n_2,i,j)|}{t_{CSF}(n_1,n_2,i,j)\cdot t_l(n_1,n_2,i,j)}\right)^{w(i,j)}\right\} \tag{7}$$

In our experiments, we first attempted to find a suitable the value for w. In the Watson model this parameter was fixed to 0.7, which in specific set-up conditions overestimate the JND especially around edges (Figure 4). Zhang proposed a much lower value of w=0.36. On the other hand, Zhang model tends to underestimate JND in EDGE blocks and gives the low watermark power in edgy sequences. In addition, we observed that in the presence of noise around edges, Zhang method tends to augment it. Thus, noise becomes more visible and annoying. As a consequence of this observations, we propose the above described method with w=0.36.

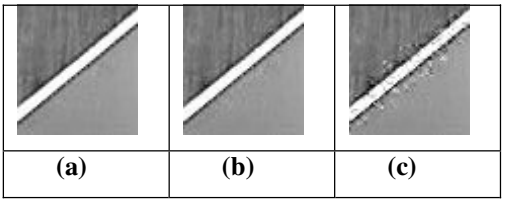


Fig. 4. Edge detail from Table Tennis: **(a)** original and watermarked with Watson model **(b)** w=0.36 and **(c)** w=0.7

The presented technique were evaluated using four typical MPEG2 sequences (Flower Garden, Mobile and Calendar, Suzy and BBC3) and results were compared with Watson [11] and Zhang [10] JND models. All sequences were 375 frames long, PAL (704x576, 25 fps), with GoP IBBP structure, size 12 and bit-rate 6 Mbps. These sequences are used since they have good proportion of plain (Suzy), edge (BBC3) and texture details (Flower Garden, Mobil and Calendar).

The result of the PSNR test is given in Table 1. A watermark is embedded in each of the five sequences and they are compared with the originals frame by frame. The table shows minimal PSNR values, maximal PSNRs and average PSNR for a whole

sequence. The most interesting is the minimal value, which presents the most degraded I frame in a sequence. From the given results, it is possible to see that in most degraded frames a difference in PSNR between the proposed method and the other two is never bigger then 1.5 dB. The minimal PSNR value of 35.82 dB confirms that high fidelity is preserved and that watermarked frames are indistinguishable from originals.

It is worth noticing that sequences that have higher percentage of texture blocks (Flower Garden – 17.98% of texture blocks, Mobile and Calendar – 17.01%) are more degraded then the one consisted mainly of plain and edge blocks (Suzy – 0.05% of texture blocks and BBC3 – 1.76 %). This is consistent with the findings on sensitivity of Human Visual System that the noise is less visible in highly textured areas.

Table 1. Peak Signal to Noise Ratio comparison of three methods

		Perceptual adjustment method		
	PSNR	Zhang	Watson	Proposed
Flower Garden	Min	37.26	37.01	36.54
	Avg	39.78	39.36	38.88
	Max	47.65	46.26	45.82
Mobile and Calendar	Min	36.31	37.06	35.82
	Avg	40.86	41.79	40.05
	Max	49.55	50.17	47.64
Suzy	Min	44.46	44.85	44.15
	Avg	48.8	48.99	48.31
	Max	56.9	56.99	55.61
BBC3	Min	40.39	39.61	38.98
	Avg	47.84	46.26	45.53
	Max	55.53	53.73	53.01

The results of perceptual evaluation showed that proposed method is comparable to other two methods and has high imperceptibility. However, the main advantage of the proposed combined method can be seen through a watermark to host ratio (*WHR*) that is watermark signal-to-noise ratio in an embedding window:

$$WHR = N_{nz} \cdot \frac{\mu_\alpha^2}{\overline{x^2}} \tag{8}$$

Table 2 shows embedding statistics when using three methods. There are three 8-frame embedding windows (EW) in any of tested sequences. For every embedding window, mean AC coefficients power ($\overline{x^2}$), number of non-zero AC coefficients (N_{nz}), mean embedding amplitudes (μ_α) and watermark to host ratios (WHR*)* are given. The WHR values for three methods show that the proposed method outperforms the other two methods by a large margin.

As expected, the lowest mean embedding amplitudes are observed in "Suzy" sequence, which mainly consists of plain blocks (93.95%). However, watermark to noise ratios in this sequence are quite high due to the low mean power of DCT coefficients and relatively high number of non-zero DCT coefficients. The most demanding sequence, as mentioned before, is the "BBC3" sequence. Consisting mainly of low-frequency transitions from black to white and vice-versa, this sequence contains relatively small number of DCT coefficients with high values describing strong edges with high luminescence changes.

Table 2. Mean amplitudes and signal-to-noise ratio for different embedding windows

				Zhang		Watson		Proposed	
	EW	$\overline{x^2}$	N_{nz}	μ_α	*WHR*	μ_α	*WHR*	μ_α	*WHR*
F	1	2700.77	1117868	4.39	7976.86	4.9	9937.911	5.23	11321.56
	2	2873.88	1040038	4.33	6785.10	4.87	8582.98	5.23	9898.83
	3	3170.28	835122	3.87	3945.24	4.39	5076.69	4.7	5818.99
M	1	3339.83	846145	2.85	2057.83	2.24	1271.20	3.56	3210.85
	2	3269.65	864405	3.02	798.40	2.47	1612.90	3.75	3717.73
	3	3055.7	930484	3.54	1077.95	2.99	2722.32	4.34	5735.58
S	1	347.05	980522	2.15	13059.97	2.07	12106.15	2.23	14049.96
	2	348.76	925484	1.96	10194.23	1.91	9680.74	2.03	10935.39
	3	309.4	995247	2.03	13255.70	1.96	12357.27	2.1	14185.65
B	1	4919.6	609132	2.31	660.70	3.02	1129.26	3.72	1713.43
	2	3603.53	648995	2.78	1391.88	3.01	1631.72	3.83	2641.86
	3	4939.23	613720	2.3	657.30	3.17	1248.61	3.81	1803.68

For the further dissemination of the three amplitude adjustment methods, we will focus on the first embedding window of the "BBC3" sequence, which has lowest WHR values. To estimate the maximal payload of the watermarking message, we define signal-to-noise ratio per watermarking bit:

$$\frac{S}{N}[\text{dB}] = 10 \cdot log_{10} \frac{WHR}{n} \tag{9}$$

where n is the number of embedded bits.

Maximal capacity for given AWGN channel is achievable with Gaussian signaling which will maximize mutual information between input and output signal. However, in the given watermarking system we are restricted to BPSK (Binary Phase Shift Keying) signaling, since watermarking bits are taking only two values either +1 or -1. In that way, since the input signal is not ideal for the given AWGN channel, mutual information between the input and the output signal will not be maximal leading to a lower maximum achievable payload:

$$C_{BPSK} = 1 - \frac{1}{\sqrt{2\pi}} \int_{-5}^{+20} e^{-\frac{1}{2}(y-\sqrt{\frac{S}{N}})^2} \cdot log_2(1 + e^{-2y\sqrt{\frac{S}{N}}})dy \tag{10}$$

Equation 10 is solvable numerically and represents the actual capacity boundary of our watermarking system. It was shown [12] that with BPSK signaling it is possible to achieve bit error rate of 10^{-5} with an SNR of 9.6dB. Using this value for the SNR and the WHR values for the first segment BBC sequence from table 2, we can calculate using equation 9 that in order to achieve bit error rate as low as 10^{-5}, we can embed maximum 72 bits using the Zhang method, 123 bit using Watson method or 188 bits using the proposed combined method for the watermark amplitude adjustment.

4 Capacity Enhancement Using Error Correction Coding

The latest generation of watermarking techniques models the process as communication through a noisy channel. The channel noise is originated by two different sources. The video itself does not carry any useful information regarding the watermark message and from a watermarking point of view can be considered as noise. Following the approach presented in the section 2 that uses block-wise watermark bits interleaving, we can approximate this noise with Gaussian distribution. The other is noise originated by attacks and it is as well usually modelled as Gaussian white noise in the evaluations of watermarking systems.

In 1993, C. Berrou, A. Glavieux, and P. Thitimajshima made a major breakthrough in channel coding theory with their pioneering work introducing Turbo coders, which enable near Shannon limit capacity [13]. This technique is widely used in communication via low SNR channels, such as mobile communication, deep space communication and more recently in watermarking. The watermarking channel, as stated previously, has a small signal-to-noise ratio and a potentially large bit error rate due to the noise introduced by the host signal and attacks. In such an environment, it is essential to protect the watermark message by introducing redundant bits, which will be used for error correction. Before turbo codes were introduced, there was the wide spread conviction that the Shannon capacity limit is achievable only if near infinite complexity is introduced in the decoder [12]. It was argued that prohibitively large codes are required to approach this limit. Indeed, Shannon et al. [14] derived the lower bound on probability of the codeword error P_B, known as sphere packing bound:

$$P_B > 2^{N(E_{sp}(R)+o(N)}; \quad E_{sp}(R) = \max_{q} \max_{\rho>1}(E_0(q,\rho) - \rho R) \tag{11}$$

where $E_0(q,\rho)$ is Gallager exponent that depends on the symbol probability distribution q and the optimization parameter ρ and R is code rate. It is worth of noticing that there is exponential dependence of the lower bound on code length N. Schlegel and Perez in [15] plotted this bound for rate 1/2 and BPSK signalling, together with different turbo codes (Figure 5).

It is possible to see that turbo codes with code length of 10^{+4}-10^{+5} give near optimal performance and longer codewords can introduce only small gains. For the code lengths smaller then 10^{+4}, sub-optimality in the performance of an error-controlling scheme is inevitable. However, per bit signal-to-noise ratio is inversely proportional to the number of embedded bits, which is depicted in Figure 5 with green dash-dot line for the “BBC3” sequence and the proposed perceptual model. It is clear that in

the given set-up only sub-optimal gains can be achieved with error correction coding. To embed more bits and to achieve better performance, we need either to embed with stronger amplitudes, which will introduce perceptual degradations, or to embed the bits in the longer video segment, which is opposite to the minimum watermarking segment requirement. Still, even with sub-optimal performance due to short code length, we were able to almost double the number of embedded bits. Using duo-binary turbo codes presented in this section, we embedded and decoded 352 bits without errors.

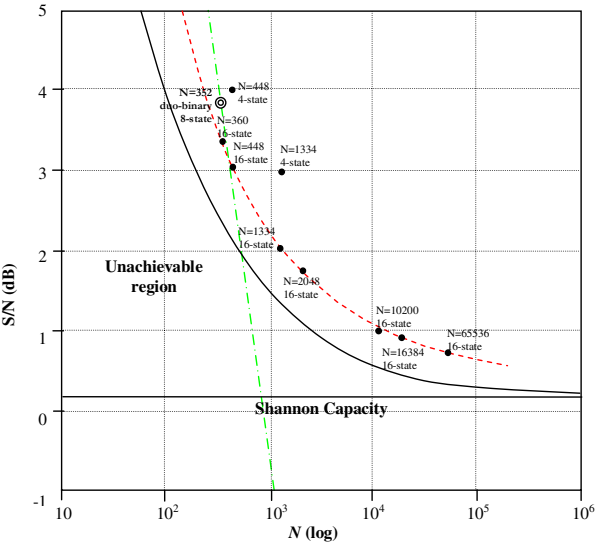


Fig. 5. The sphere-packing bound and performance of different turbo codes depending on the code length N

The classical turbo coder is a parallel concatenation of two binary rate 1/2 Recursive Systematic Convolutional (RSC) encoders that are separated by an interleaver. The overall TC rate is 1/3 without puncturing. To reduce the rate and number of bits that needs to be embedded in the sequence, puncturing mechanism needs to be used. However, puncturing unavoidably leads to sub-optimal performance of a turbo code. More recently, Berrou et al. [16] argued that non-binary turbo codes based on RSCs with $m \geq 2$ input bits outperforms classical binary turbo coders. Duo-binary turbo codes consist of two binary RSC encoders of rate 2/3 and an interleaver of length k. Each binary RSC encoder encodes pair of data bits and produces one redundancy bit, so desired rate 1/2 is the natural rate of the double binary TC, so no puncturing is needed yielding better protection.

We considered the 8-state duo-binary TC with RSCs that have generator polynomial $\mathbf{G} = \{g_1, g_2\} = \{15,13\}$ as has been adopted by the ETSI (European Telecommunications Standards Institute) standards for Digital Video Broadcasting with Return Channel via Satellite (DVB-RCS) [17] and Digital Video Broadcasting

with Return Channel via Terrestrial (DVB-RCT) [18] as shown in the Figure 6. The tail-biting [19] technique, also called Circular Recursive Systematic Convolutional (CRSC) [20], is used to convert the convolutional code to block code that allows any state of the encoder as the initial state. This technique encodes input bit sequence twice, first time the RSC initial state is all-zero state and final state is used to calculate the initial state for the second encoding, which will also be final state after second encoding. Therefore, the technique assures that initial and final state will be the same, so there is no need to tail bits to derive the encoders to the all-zero state.

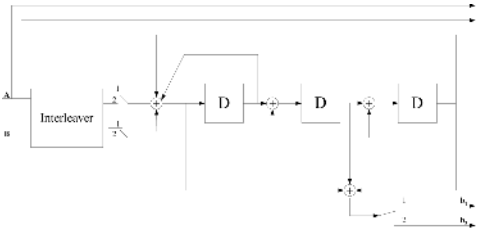


Fig. 6. Duo-Binary Turbo Encoder

At low error rates or high signal-to-noise ratio, the performance of the classical turbo coder fluctuates due to the "error floor". The higher minimum distance can reduce the error floor effect at low error rates. Duo-binary turbo coders normally have better performance than classical turbo coders due to larger minimum distance. The minimum distance of turbo codes depends on the interleaver. The interleaver design is a critical issue and the performance of the turbo code depends on how well the information bits are scattered by the interleaver to encode the information by second binary RSC encoder. To get better performance for the duo-binary code for watermarking channel, the particular block length is selected that behave better in the low error rates. This can be accomplished by using All-zero iterative method [21] to check the performance of the duo-binary turbo code.

The turbo-decoder is composed of two Maximum A Posteriori (MAP) [16] decoders, one for each stream produced by the singular RSC block as shown in Figure 7. The first MAP decoder receive the two distorted systematic bits ($\mathbf{A^r_k}$, $\mathbf{B^r_k}$) after channel along with the parity $\mathbf{b^r_{k1}}$ for first binary RSC encoder and produce the *extrinsic information* $\mathbf{E_{12}}$ that is interleaved and feed to the second MAP decoder as the *a priori* information. The second MAP decoder produces the *extrinsic information* $\mathbf{E_{21}}$ based on interleaved distorted systematic bits ($\mathbf{A^r_k}$, $\mathbf{B^r_k}$), distorted parity by second binary RSC encoder $\mathbf{b^r_{k2}}$ and *a priori* information from first MAP decoder. Then $\mathbf{E_{21}}$ is used as the *a priori* information of the first MAP decoder. After a certain number of iterations, usually 3 to 10, the *a posteriori probability* (APP) is taken, deinterleaved and performed hard decision to get transmitted information. Number of iterations in the experiment was derived by comparing decoded message with original one and maximal number of iterations needed was 10. In the real world implementation, this would be done with additional CRC (Cyclic Redundancy Check) bits, which would be used for stopping of iterative process.

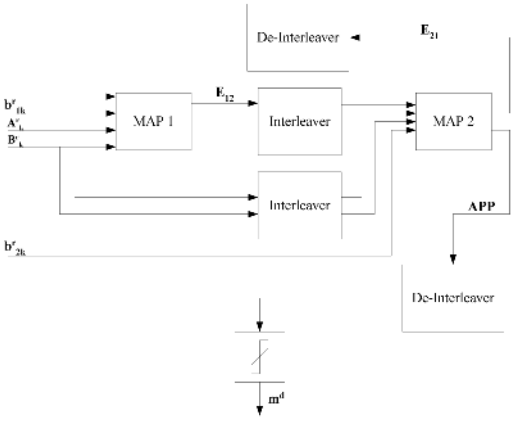


Fig. 7. Iterative Turbo Decoding based on MAP algorithm for duo-binary TC

To evaluate the performance of the watermarking system protected by turbo coding technique described above, we embedded 10^{+5} bits with turbo coding protection and compared results with embedding unprotected messages. Uncoded and turbo coded (both classical parallel concatenated and duo-binary) messages of different sizes (64-640) were spread through 8 I frames (5 seconds of video sequence) and embedded in sequences.

Table 3. Peak Signal to Noise Ratio comparison – watermarked vs. watermarked and transcoded to 2Mbps

	PSNR	Wat	Wat+T2Mbps
BBC3	Min	38.98	28.05
	Avg	45.53	32.38
	Max	53.01	42.68

We also tested the robustness on transcoding to 2 Mbps. For the transcoding test, ffmpeg software [22] was employed. Since ffmpeg coder mainly compresses a sequence by changing quantization steps, it introduces severe degradations to the sequence. In Table 3, PSNR levels after watermarking and transcoding to 2 Mbps for the "BBC3" sequence are compared with PSNR levels after watermarking of the "BBC3" sequence.

Bit Error Rate (BER) results with and without turbo coding are given in Figure 8. During experiments without attacks, errors were not observed when communicating 192-bit messages without turbo coding, while with classical turbo coder with puncturing [7] payload is increased to 256 bits per 8 I frames that is 256 bits in 5 seconds of video. If a watermarked sequence is transcoded to 2 Mbps, a 96-bit watermarked message embedded without protection can persist. In the case of protection with a classical turbo code with UMTS interleaver, given the number of

communicated bits (10^{+5}) and without errors observed using 128-bit messages, it can be said with 99% confidence that BER will be as low as $4.61*10^{-5}$ [1]. This small gain is rather disappointing and can hardly justify additional computational costs. Due to small watermarking messages and puncturing mechanism, protection is suboptimal and increase in payload is small.

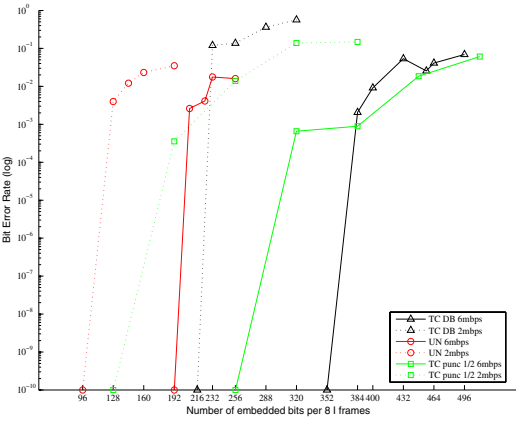


Fig. 8. Bit Error Rates for duo-binary protection (TC DB), classical turbo coder with puncturing (TC punct ½) and unprotected (Uncoded) watermark message: without attack – 6Mbps and with transcoding attack - 2Mbps

However, when duo-binary turbo codes are used, the iterative nature of turbo coding shows more than a double gain in the embedded bits for uncoded watermarking messages at 6Mbps and after transcoding at 2 Mbps. A 352-bit watermark message is separated into two 176-bit sequences that are encoded with duo-binary turbo coder of rate 1/2 and after watermarking channel and turbo decoding, there is no error found. However, in order to resist transcoding watermark message needs to be at most 216 bits long. Again, given the experiment set-up, we can be 99% confident that bit error rate will be lower then $4.61*10^{-5}$.

5 Conclusions

The watermarking technique based on a spread spectrum paradigm is presented. The watermark is spread by a large chip factor, modulated by a pseudo sequence and then added to the DCT coefficients of an MPEG2 sequence. Detection probability was increased by a new block-wise random watermark bits interleaving. In addition, since a transmission channel has its own particular capacity, the bit-rate of a video stream

[1] If errors have not been observed in N experiments and if desired confidence level is C%, it can be shown that [23]:

$$BER < -\frac{ln(1-C)}{N}$$

needs to be chosen to comply with the capacity of the channel. Therefore, watermarking of a compressed video bit-stream must not increase its bit-rate. A novel technique for bit-rate control on the macro-block level increased the number of watermarked coefficients in comparison with existing schemes.

The proposed perceptual adjustment method takes advantages of information about local characteristics of the picture, information that can be easily extracted from DCT coefficients. Two state-of-the-art methods for Just Noticeable Difference (JND) estimation were considered and a new model capitalizing on the good characteristics of two models was proposed. The results of experiments with the new model showed PSNR levels comparable to the previous model and at the same time significant increase in the payload.

To boost the capacity of our technique we introduced state-of-the-art error correction coding technique – duo-binary turbo coding. It was shown that duo-binary turbo codes can effectively increase the watermark payload. Duo-binary codes perform better then classical turbo coders in protection of watermarking channel, since they have natural rate of 1/2 and no puncturing is needed. Beyond that, they are computationally less expensive, show better convergence for iterative decoding and have a large minimum distance.

Acknowledgments

Work partially supported by European Community under the Information Society Technologies (IST) programme of the 6th FP for RTD - project EASAIER contract IST-033902. The author is solely responsible for the content of this paper. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

References

[1] Hartung F., Girod B.: Watermarking of uncompressed and compressed video. Signal Processing, vol. 66, no. 3 (1998) 283-302

[2] Chung T., Hong M., Oh Y., Shin D., Park S.: Digital watermarking for copyright protection of mpeg2 compressed video. IEEE Trans. on Consumer Electronics, vol. 44, iss. 3, (1998) 895-901

[3] Simitopoulos D., Tsaftaris S., Boulgouris N. V., Brissouli A., Strintzis M. G.: FastWatermarking of MPEG-1/2 Streams Using Compressed-Domain Perceptual Embedding and a Generalized Correlator Detector. EURASIP Journal on Applied Signal Processing, vol.8 (2004) 1088-1106

[4] Ambroze A., Wade G., Serdean C., Tomlinson M., Stander J., Borda M.: Turbo Code Protection of Video Watermark Channel. IEE Proc. Vis. Image Signal Processing, vol.148, No.1 (2001) 54-58

[5] Pranata S., Wahadaniah V., Guan Y. L., Chua H. C.: Improved Bit Rate Control for Real-Time MPEG Watermarking. EURASIP Journal on Applied Signal Processing, vol.14 (2004) 2132–2141

[6] Cheveau L.: Choosing A Watermarking System for Digital Television – The Technology and The Compromises. *IBC2002* www.broadcastpapers.com/asset/IBCEBUWatermarking03.htm
[7] Damnjanovic I., Izquierdo E.: Turbo Coding Protection of Compressed Domain Watermarking Channel. Proc. of IEEE International Conference on Computer as a Tool, Belgrade, Serbia and Montenegro (2005)
[8] Ahumada A.J., Peterson H.A., Watson A. B.: An Improved Detection Model for DCT Coefficients Quantization. Proc. of SPIE, Human Vision, Visual and Digital Display IV, vol. 1913-14 (1993) 191-201
[9] Cox I., Miller M., Bloom J.: Digital Watermarking. Morgan Kaufmann Publisher (2001) 1-55860-714-5
[10] Zhang X.H., Lin W.S., Xue P.: Improved Estimation for Just-noticeable Visual Distortions. Signal Processing, vol. 85, no. 4 (2005) 795-808
[11] Watson A.B.: DCT quantization matrices visually optimized for individual images. Proc. of SPIE, Human Vision, Visual and Digital Display IV, vol. 1913-14 (1993) 202-216
[12] Schlegel C. B., Perez L.C.: Trellis and Turbo Coding. IEEE Press (2004) 0-471-22755-2
[13] Berrou C., Glavieux A., Thitimajshima P.: Near Shannon limit error- correcting coding and decoding: Turbo Codes. Proc. Int. Conf. Comm. (1993) 1064-1070
[14] Shannon C. E., Gallager R. G., Berlekamp E. R.: Lower bounds to error probabilities for coding on discrete memoryless channels. Information and Control, vol.10 (1967) part I – no. 1 65-103 and part II – no.5 522-552
[15] Schlegel C.B., Perez L.C.: On Error Bounds and Turbo-Codes. IEEE Communications Letters, vol. 3, no. 7 (1999) 205-207
[16] Berrou C., Jézéquel M., Douillard C., Kerouédan S.: The advantages of non-binary turbo codes. Proc. Information Theory Workshop, Cairns, Australia (2001) 61-63
[17] Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems. ETSI EN 301 790, V1.4.1 (2005) 23-26
[18] Digital Video Broadcasting (DVB); Interaction channel for Digital Terrestrial Television (RCT) incorporating Multiple Access OFDM. ETSI EN 301 958, V1.1.1 (2002) 28-30
[19] Bettstetter C.: Turbo decoding with tail-biting trellises. Diplomarbeit, Technischen Universitat Munchen (1998)
[20] Berrou C., Douillard C., Jézéquel M.: Multiple parallel concatenation of circular recursive systematic convolutional (CRSC) codes. Annals of Telecommunications, vol. 54, no. 3–4 (1999) 166 – 172
[21] Garello R., Vila A.: The all-zero iterative decoding algorithm for turbo code minimum distance computation. Proceedings of IEEE Int. Conf. Commun. (ICC'04), Paris, France, (2004) 361–364
[22] FFMPEG Multimedia Systems – version: ffmpeg-0.4.9-pre1 http://ffmpeg.sourceforge.net/index.php
[23] HFTA-05.0: Statistical Confidence Levels for Estimating BER Probability. Application Note 1095 (2000)

Detection of Image Splicing Based on Hilbert-Huang Transform and Moments of Characteristic Functions with Wavelet Decomposition

Dongdong Fu, Yun Q. Shi, and Wei Su

Dept. of Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, New Jersey, USA
{df7, shi}@njit.edu

Abstract. Image splicing is a commonly used technique in image tampering. This paper presents a novel approach to passive detection of image splicing. In the proposed scheme, the image splicing detection problem is tackled as a two-class classification problem under the pattern recognition framework. Considering the high non-linearity and non-stationarity nature of image splicing operation, a recently developed Hilbert-Huang transform (HHT) is utilized to generate features for classification. Furthermore, a well established statistical natural image model based on moments of characteristic functions with wavelet decomposition is employed to distinguish the spliced images from the authentic images. We use support vector machine (SVM) as the classifier. The initial experimental results demonstrate that the proposed scheme outperforms the prior arts.

Keywords: image splicing, Hilbert-Huang transform (HHT), characteristic functions, support vector machine (SVM).

1 Introduction

In recent years, the advent and popularity of digital camera have made the digital image prevailing in our daily life. With widely available modern image processing tools, however, digital image can be easily modified without leaving any visual clue. Therefore, there is an increasingly urgent concern on the authenticity of digital image, especially for forensic purpose.

Image tampering refers to the malicious manipulation of images to mislead the observers. Even though the digital image tampering is quite a simple thing with modern and readily available photo-editing software, the detection of digital image tampering is actually a tough mission. There are generally two categories of digital image tampering detection algorithms, i.e., active detection algorithm and passive/blind detection algorithm. Active detection approach generally embeds digital watermark or digital signature at the source side while verify it at the receiver side. Although active method can provide relatively reliable detection accuracy, its requirement of embedding information at source side is hard to meet in some practical

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 177 – 187, 2006.

applications. In contrast to active approach, the passive/blind detection scheme does not need any prior information to be embedded beforehand. Since most of today's digital images do not have digital watermarking/signature embedded, more and more attentions are attracted in the passive detection approach.

There are various types of operations may be involved in image tampering, such as splicing, re-sampling, rescaling, and recompression etc. "Divide and conquer" strategy is often used in the image tampering detection. In [1], Popescu proposed several methods for the detection of different forms of tampered images separately.

Image splicing is one of fundamental techniques in image tampering, which joints together visual objects from different images or different regions in a same image by cut-and-paste operation only [3, 4, 5]. In [2], the speech signal splicing is considered by Farid as a highly non-linear process and thus higher order spectral analysis, specifically bicoherence, is introduced to deal with this problem. Ng et al. [3, 4] extended the idea to tackle the problem of image splicing. They also established a well designed image splicing evaluation database [5] and generously made it publicly available. However, the detection accuracy of their approach is not satisfactory.

In this paper, we focus on the passive detection of image splicing. We present a novel blind image tampering detection scheme based on the Hilbert-Huang Transform (HHT) and a statistical natural image model. The experimental results demonstrate that the proposed scheme outperforms the prior arts when applied on the publicly available evaluation database [5].

The rest of this paper is organized as follows. The analysis and procedure of feature extraction are described in Section 2. The experimental results are given in Section 3 and Section 4 draws the conclusions.

2 Feature Extraction

The image splicing problem is tackled as a two-class classification problem under the pattern recognition framework in the proposed scheme. That is, a given image is classified as authentic or spliced depending on the classification of representative features we extracted from the image. Hence, the extraction of appropriate features is the most important part in our scheme.

2.1 Features Based on Hilbert-Huang Transform

Since each part of the spliced image is actually a part of an authentic image, it is very hard to model the image splicing process. Furthermore the image itself is non-stationary, which makes the problem even harder. In [2], the speech signal splicing is considered as a highly non-linear process and thus higher order spectral analysis, specifically bicoherence, is introduced to deal with this problem. Ng et al. extended this scheme into image splicing detection [3, 4]. But their method did not provide satisfactory detection accuracy (only **72%** as reported in [3, 4]). Therefore, more efficient tool is needed in the detection of image splicing. In this paper, we also consider the image splicing operation as a highly non-linear process but in a different way. Our proposed scheme utilizes the newly developed Hilbert-Huang Transform (HHT) [6] to generate more effective features for classification.

2.1.1 Hilbert-Huang Transform

The HHT technology is a highly efficient tool capable of analyzing time-varying processes [6]. It has attracted increasingly interests in a wide variety of signal processing fields. It is especially suitable for analyzing nonlinear and non-stationary signals, which is exactly the case in the image splicing detection application.

The HHT typically consists of two parts. The signal is firstly decomposed into intrinsic mode functions (IMFs) by the empirical mode decomposition (EMD). Hilbert spectral analysis (HAS) is then applied to extract the characteristics of the IMFs. Instead of the fixed basis functions as used in FFT and DCT, the HHT adaptively derives its basis functions from the signal itself. Therefore, HHT demonstrates excellent performance in analyzing nonlinear and non-stationary signals.

An intrinsic mode function (IMF) is a function that satisfies the following two conditions [6]:

(1) In the whole data set, the number of extrema and the number of zero crossings must either equal or differ at most by one.

(2) At any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

The IMFs of a signal can be obtained by sifting algorithm [6].

After the decomposition step, the data are reduced to several IMF components. Hilbert transform is then performed on each IMF component. For a given arbitrary signal *x(t)*, the Hilbert transform can be expressed as follows:

$$y(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{x(t')}{t - t'} dt' \tag{1}$$

where *P* indicates the Cauchy principal value. With this definition, *x(t)* and *y(t)* can be used to define an analytic signal *z(t)*:

$$z(t) = x(t) + iy(t) = a(t)e^{i\theta(t)} \tag{2}$$

where, $a(t) = [x^2(t) + y^2(t)]^{1/2}$ and $\theta(t) = \arctan(\frac{y(t)}{x(t)})$ are the amplitude and phase of this analytical signal, respectively. Furthermore, the instantaneous frequency is defined as:

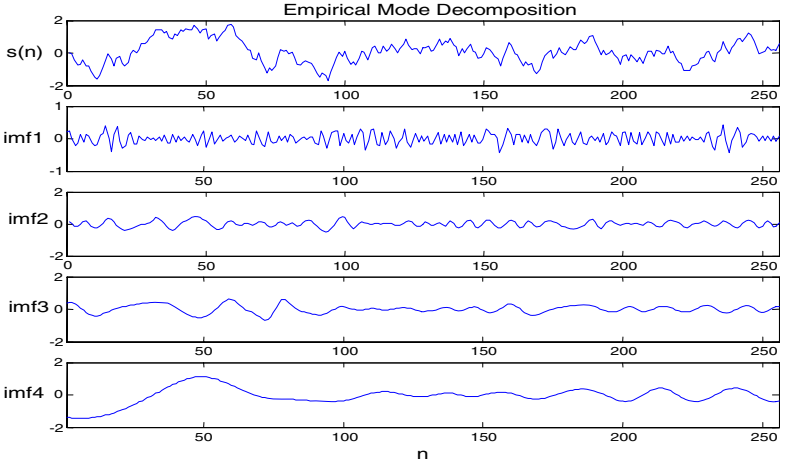
$$\omega = \frac{d\theta(t)}{dt} \tag{3}$$

The Hilbert transform enables us to represent the amplitude and the instantaneous frequency as functions of time. This frequency-time distribution of the amplitude is designated as the Hilbert amplitude spectrum. The frequency in the Hilbert spectrum indicates that an oscillation with such a frequency exists.

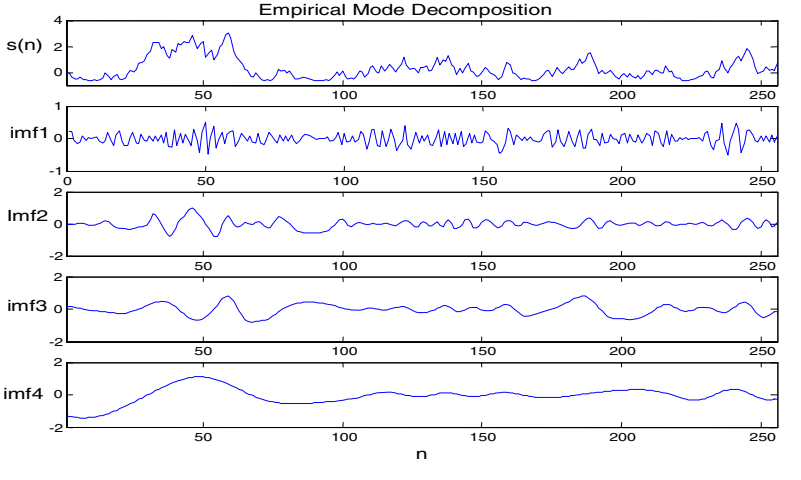
Similar to [2], we also model the splicing operation as a non-linear process. To illustrate the effectiveness of the HHT to detect the non-linearity, Fig. 1 shows the EMD of random signals with different amount of global nonlinearities. For a given input signal *v(n)*, the global non-linearity is with following form:

$$s(n) = \alpha \cdot v^2(n) + v(n) \tag{4}$$

where parameter α controls the strength of the non-linearity. Since image signals are usually modeled as Markov process, we generate a first order Markov random sequence $v(n)$ with 256 samples. Different amounts of non-linearity are then applied to this signal. Only the four highest frequency IMFs for each signal are displayed in Fig. 1. It is obvious that there are increasing activities in these highest few IMFs as the non-linearity increases, i.e., with increasing α.

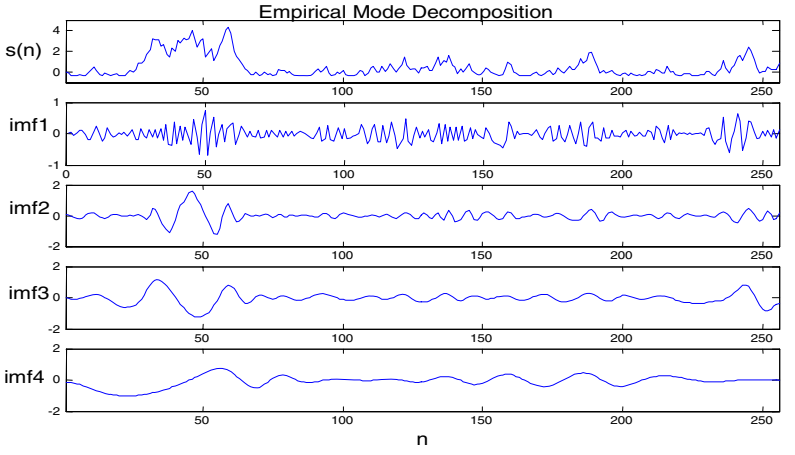


(a)

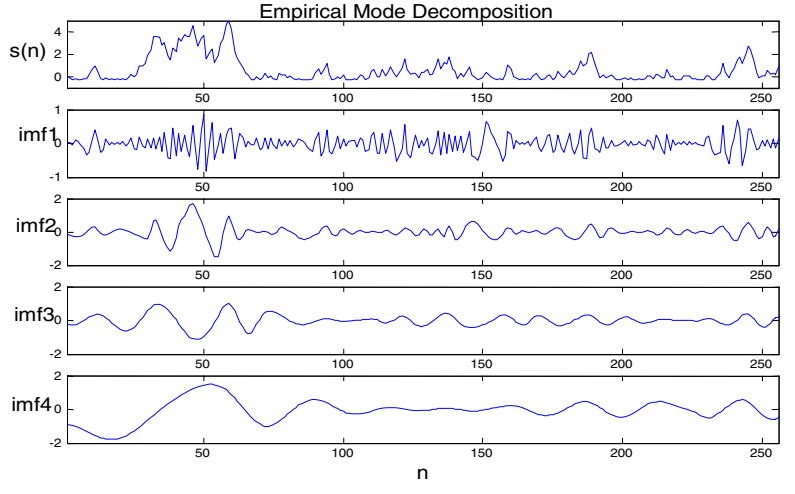


(b)

Fig. 1. EMD for detecting global non-linearities. Signal $s(n) = \alpha \cdot v^2(n) + v(n)$, $v(n)$ is a first order Markov random sequence with 256 samples, (a) α =0; (b) α =0.4; (c) α =0.8; (d) α =1.0. (The vertical axis stands for the amplitude of signal and the horizontal axis for time.)



(c)



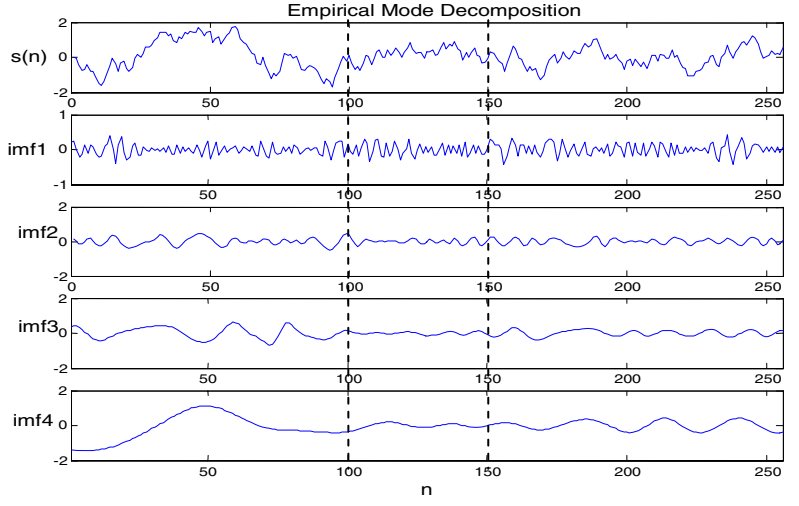
(d)

Fig. 1. (*continued*)

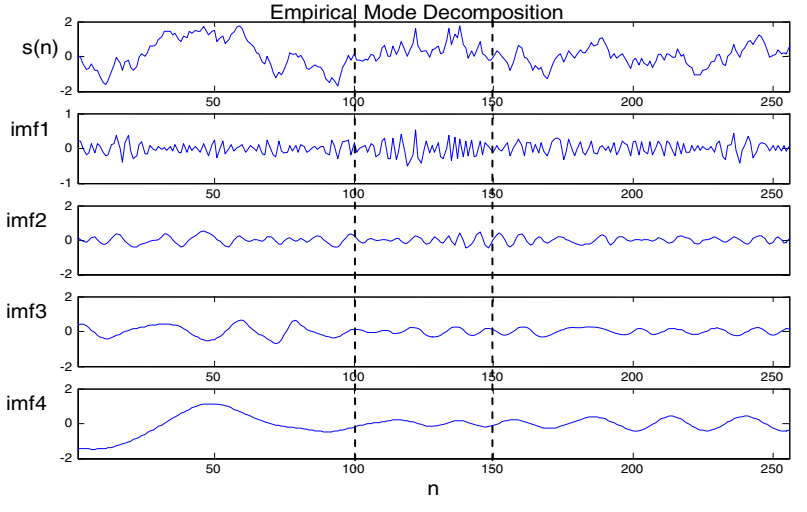
Fig. 2 shows the EMD of signal with local non-linearity, i.e., only part (between the two dash lines in Fig. 2) of the original signal *v(n)* goes though the non-linear process defined in Eq. (4). As can be seen in Fig. 2, the local non-linearity has also introduced increasing activities in the non-linear area (between the two dash lines) for the highest a few IMFs.

The Matlab source code of EMD used in the experiments is obtained from [7].

These two experiments have demonstrated that the EMD can capture both the global and local non-linearity in its highest a few IMFs. Therefore, it is reasonable to expect the HHT can be effective in the detection of image splicing.



(a)



(b)

Fig. 2. EMD for detecting local non-linearities. Signal $s(n) = \begin{cases} \alpha \bullet v^2(n) + v(n) & 100 \le n \le 150 \\ v(n) & otherwise \end{cases}$ $v(n)$ is a first order Markov random sequence with 256 samples, (a)α =0; (b)α = 1.0. (The vertical axis stands for the amplitude of signal and the horizontal axis for time)

2.1.2 HHT Feature Generation

As describe above, the image splicing process can be generally considered as a nonlinear process. Since the HHT is an excellent tool to capture the characteristics of

the nonlinear and no-stationary process. It is reasonable for us to utilize the HHT to generate the features for classification.

The detail procedure for extracting HHT features is illustrated in Fig. 3. 1-D EMD is used in this paper.

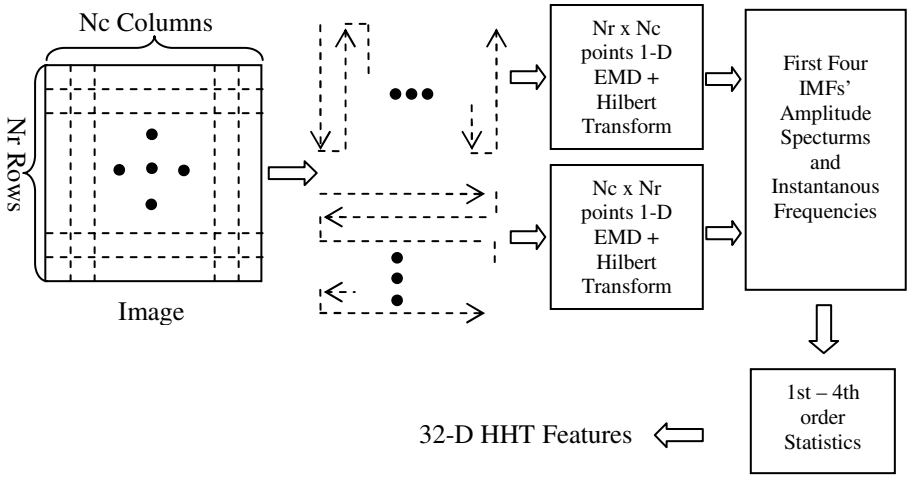


Fig. 3. The HHT feature extraction procedure

As shown in Fig. 3, a given 2-D image is firstly expanded into 1-D row signal and column signal in *snake* scanning order (i.e. "row-by-row and end-to-head" and "column-by-column and end-to-head", respectively). 1-D HHT is then performed on the 1-D row and column signal, respectively. After that, the Hilbert spectrums for the row and column signal are obtained. It has been shown in the literature [8] that, after EMD, the nonlinear distortion of a signal is mainly distributed into the IMFs of higher frequency components and affect a little to those IMFs of lower frequency. Our experiments illustrated in Fig. 1 and Fig. 2 have also confirmed this point. In our scheme, hence, only the amplitudes and instantaneous frequencies of the four highest frequency IMFs' Hilbert spectrums are used for the feature construction. Since we extract the 1^{st} to 4^{th} order statistics (mean, variance, skewness and kurtosis) for each IMF's amplitude spectrum and instantaneous frequency, there are totally 32-D features generated. We denote these features as HHT features in this paper.

2.2 Features Based on Moments of Characteristic Functions Using Wavelet Decomposition

In addition to the features extracted by using HHT, we also propose a natural image model to capture the differences between authentic image and spliced image which is introduced by splicing operation.

In [9], Farid and Lyu presented a statistical model for natural images consisting of higher-order wavelet statistics. These statistics are based on decomposition of images

with separable quadrature mirror filters. They used the higher-order statistics of both the wavelet coefficients and the linear prediction error of the wavelet coefficients in the high-frequency subbands as features for distinguishing various types of image tampering operations in digital forensics. They have used this natural image model for classification of natural images versus stego image, natural images versus photorealistic computer graphics and natural images versus print-and-scanned images. Even though their model performs reasonably well for differentiating between natural images and the above mentioned un-natural images, it does not work well for blind detection of image splicing as shown in [3]. Therefore, more effective natural image model is called for.

In this paper, we propose a statistical natural image model for detecting image splicing, which is based on moments of characteristics function using wavelet decomposition. In our previous work [10], we have shown that the proposed model performs better than that of [9] in distinguishing natural images from stego images. In this model, the statistical moments of characteristic functions of a prediction-error image, test image, and all of their wavelet subbands are selected as features for classification. This model captures certain statistical characteristics that are inherent to natural images. In this paper, we apply this model to detect image splicing and the experimental results have shown reasonably good performance. We will describe the detail of this statistical natural image model as follows.

The histogram of an image is basically the probability mass function (pmf) of the image (only differing by a scalar). Multiplying components of the pmf by a correspondingly shifted unit impulse results in a probability density function (pdf). In the context of discrete Fourier transform (DFT), if unit impulses is ignored, implying that pmf and pdf are exchangeable. Thus, the pdf can be thought as a normalized version of a histogram. According to [11, pp. 145-148], one interpretation of characteristic function (CF) is that the CF is simply the Fourier transform of the pdf (with a reversal in the sign of the exponent).

Due to the decorrelation capability of a discrete wavelet transform (DWT), coefficients of different subbands at the same level are less correlated to one another. Therefore, the features generated from different wavelet subbands at the same level are less correlated to one another. This property is desirable for image modeling.

We propose to use statistical moments of the CFs of both a test image and its wavelet subbands as features for natural image modeling, which are defined as follows:

$$M_n = \sum_{j=1}^{(N/2)} f_j^n \left| H\left(f_j\right) \right| \Big/ \sum_{j=1}^{(N/2)} \left| H\left(f_j\right) \right| \tag{5}$$

where $H(f_i)$ is the CF component at frequency f_i, N is the total number of points in the horizontal axis of the histogram. The zero frequency component of the CF, i.e., $H(f_0)$, has been excluded in the calculation of moments because it represents only the summation of all components in the discrete histogram.

Since the neighboring pixels in natural image have strong correlations between one another, pixel grayscale values in the original cover image can be predicted by using its neighboring pixels' grayscale values, and obtain a prediction-error image by

subtracting the predicted image from a test image. Here, such a prediction-error image removes various content information of the image itself and enhance the information introduced by image tampering operations, thus making the model more efficient. The prediction algorithm is expressed as follows [12]:

$$\hat{x} = \begin{cases} \max(a,b) & c \le \min(a,b) \\ \min(a,b) & c \ge \max(a,b) \\ a+b-c & otherwise \end{cases} \tag{6}$$

where *a, b, c* is a context of a pixel *x* under consideration, and $\hat{x}$ is the prediction value of *x*. The location of *a, b, c* can be shown in Fig. 4.

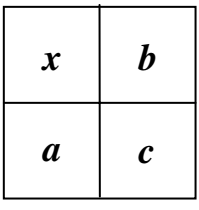


Fig. 4. Prediction context

In our experimental work, a test image will be decomposed using a three-level Haar transform. For each level, there are four subbands, resulting in 12 subbands in total. If the original image is considered as level-0 LL subband, we have a total of 13 subbands. For each subband, the first three moments of characteristic functions are derived according to Equation (5), resulting in a set of 39 features. Similarly, for the prediction error image, another set of 39 features can be generated. Thus, a 78-D feature vector is produced for the test image. Our extensive experimental study has shown that using more than three-level wavelet decomposition and including more than the first three order moments do not further improve the performance, while leading to higher computational complexity. Hence the 78-D feature vectors are used in our proposed natural image model. We denote these features as model features in this paper.

3 Experimental Results

Once the proposed features have been generated, we need to evaluate the effectiveness of these features. In our experiments, we adopt support vector machine (SVM) as the classifier. The second order polynomial kernel is used in the experiments. The Matlab SVM code is obtained from LIBSVM [13].

Columbia image splicing detection evaluation dataset [5] is the only publicly available image splicing database so far. To fairly evaluate the performance of the

Table 1. Performances of proposed detection scheme (TN: True negative; TP: True positive; Accuracy = (TN+TP)/2)

	TN	TP	Accuracy
HHT Features (32-D)	67.78%	79.31%	73.55%
Model Features (78-D)	76.49%	73.91%	75.23%
Model + HHT (110-D)	80.25%	80.03%	80.15%

proposed image splicing detection scheme, we work on this database in our experiments. There are totally 933 authentic images and 912 spliced images in this dataset. The details of the images are described in [5].

In the classification process, we randomly selected 5/6 of total 933 authentic images and 5/6 of total 912 spliced images for training and the remaining 1/6 of the authentic images and spliced images for testing the trained classifier.

The experimental results are shown in Table 1. The experimental results reported here are the averages of the 20 times of tests.

In Table 1, we illustrate the classification performances by using HHT features and model features only in the first two rows, respectively. As shown in the last row of Table 1, combining HHT features and model based features together, our proposed scheme can achieve a promising detection accuracy of **80.15%**, which is **8%** higher than that of the prior arts (only **72%** as reported in [3, 4]) on the same evaluation dataset.

4 Conclusions

In this paper, we have proposed a novel scheme for passive detection of image splicing. The contributions of this paper are as follows:

1. A newly developed non-linear and non-stationary analysis tool, the Hilbert-Huang Transform (HHT), has been proposed to extract features for image splicing. To the best of our knowledge, this is the first time that the HHT has been applied to the digital image splicing detection.

2. A statistical natural image model has been proposed to distinguish natural images from spliced images, which is based on moments of characteristics functions using wavelet decomposition.

The initial experimental results have demonstrated that the proposed scheme outperforms the prior arts by a large margin when applied on the same publicly available image splicing evaluation dataset. More experimental works are needed in future.

Acknowledgement

We sincerely thank Mr. Shunquan Tan for bringing the HHT (Hilbert-Huang transform) technology into our group discussion.

References

1. Popescu, A.C.: Statistical tools for digital image forensics. Ph.D. Dissertation, Department of Computer Science, Dartmouth College (2005)
2. Farid, H.: Detection digital forgeries using bispectral analysis. Technical Report, AIM-1657, MIT AI Memo (1999)
3. Ng, T.-T., Chang, S.-F.: Blind detection of photomontage using higher order statistics. ADVENT Technical Report #201-2004-1, Columbia University, June (2004)
4. Ng, T.-T., Chang, S.-F., Sun, Q.: Blind detection of photomontage using higher order statistics. IEEE International Symposium on Circuits and Systems (ISCAS), Vancouver, Canada, May (2004)
5. Columbia DVMM Research Lab (2004): Columbia Image Splicing Detection Evaluation Dataset,http://www.ee.columbia.edu/dvmm/researchProjects/AuthenticationWatermarking/BlindImageVideoforensic/
6. Huang, N.E., Shen, Z., and Long, S.R.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London, vol. A, no. 454. (1998) 903-995
7. http://perso.ens-lyon.fr/patrick.flandrin/emd.html
8. Yang, Z., Qi, D., and Yang, L.: Signal period analysis based on Hilbert-Huang transform and its application to texture analysis. International Conference of Image and Graphic, Hong Kong (2004)
9. Farid, H., and Lyu, S.: Higher-order wavelet statistics and their application to digital forensics. IEEE Workshop on Statistical Analysis in Computer Vision, Madison, Wisconsin (2003)
10. Shi, Y.Q., Xuan, G., Zou, D., Gao, J., Yang, C., Zhang, Z., Chai, P., Chen, W., Chen, C.: Steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network. International Conference on Multimedia and Expo, Amsterdam, Netherlands (2005)
11. Leon-Garcia, A.: Probability and Random Processes for Electrical Engineering, 2nd Edition. Reading, MA: Addison-Wesley Publishing Company (1994)
12. Weinberger, M., Seroussi, G. and Sapiro, G.: LOCOI: A low complexity context-based lossless image compression algorithm. Proceeding of IEEE Data Compression Conference (1996) 140-149
13. Chang, C.C. and Lin, C.J.: LIBSVM: a library for support vector machines. (2001) (http://www.csie.ntu.edu.tw/~cjlin/libsvm)

Intellectual Property Rights Management Using Combination Encryption in MPEG-4*

Goo-Rak Kwon, Kwan-Hee Lee, Sang-Jae Nam, and Sung-Jea Ko

School of Electrical Engineering, Korea University
5-1 Anam-Dong, Sungbuk-Ku, Seoul 136-701, Korea
Tel.: +82-2-3290-3228
grkwon@dali.korea.ac.kr

Abstract. In this paper, we propose a new encryption method for intellectual property rights management of MPEG-4 coder. DCT coefficients and motion vectors (MVs) are used for the combination encryption of the MPEG-4 video. Experimental results indicate that the proposed joint and partial encryption technique provides levels of security and achieves a simple and coding-efficient architecture with no adverse impact on error resilience.

1 Introduction

With the advance of multimedia technology, multimedia sharing among multiple devices has become the main issue. This allows users to expect the peer-to-peer distribution of unprotected and protected contents over public network. This occurred recently with the multimedia revolution in digital audio and video (A/V) contents. Many A/V processing software including DVD players, CD rippers, MP3 encoders, and A/V players have been posted for free on the Web allowing users to build their own A/V record collections from their own CD and DVD. Inevitably, this situation has caused an incredible piracy activity and Web sites have begun to provide copyrighted A/V data for free. In order to protect the contents from illegal attacks, digital right management (DRM) is required.

The DRM system generally provides two essential functions: management of digital rights by identifying, describing, and setting the rules of the content usage, and digital management of right by securing the contents and enforcing usage rules. Various encryption techniques for digital right management (DRM) have been proposed [1,2,3,4,5,6,7,8,9,10,11,12,13]. These techniques are classified into two approaches: scrambling and watermarking. As scrambling is generally based on old and proven cryptographic tools, it efficiently ensures confidentiality, authenticity, and integrity of messages when they are transmitted over an open network. However, it does not protect against unauthorized copying after the message has been successfully transmitted and descrambled [1]. This kind of protection can be handled by watermarking [2,3,4,5], which is a more recent topic that has attracted a large amount of research and is perceived as a

* This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul Industry-Academy-Research Cooperation Project.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 188–196, 2006.

complementary aid in encryption. A digital watermark is a piece of information inserted and hidden in the media content [6,7,8,9,10,11,12,13]. This information is imperceptible to a human observer but can be easily detected by a computer. Moreover, the main advantage of this technique is to provide the nonseparability of the hidden information and the content. A watermarking system consists of an embedding algorithm and a detecting function. The embedding algorithm inserts a message into a media and the detecting function is then used to verify the authenticity of the media by detecting the message. The most important properties of a watermarking scheme include robustness, fidelity, tamper resistance, and data payload [3].

In this paper, we propose a new solution for DRM called joint and partial encryption. DCT coefficients and motion vectors (MVs) in MPEG-4 coder are used for the encryption of the video. The encryption process of the proposed method is very simple with no adverse impact on error resiliency and coding efficiency. And it also provides levels of security with authorized information and allows more flexible selective encryption.

The rest of the paper is organized as follows. In Section 2, the proposed encryption techniques are presented. Experimental results are shown in Section 3 and Section 4 gives some concluding remarks.

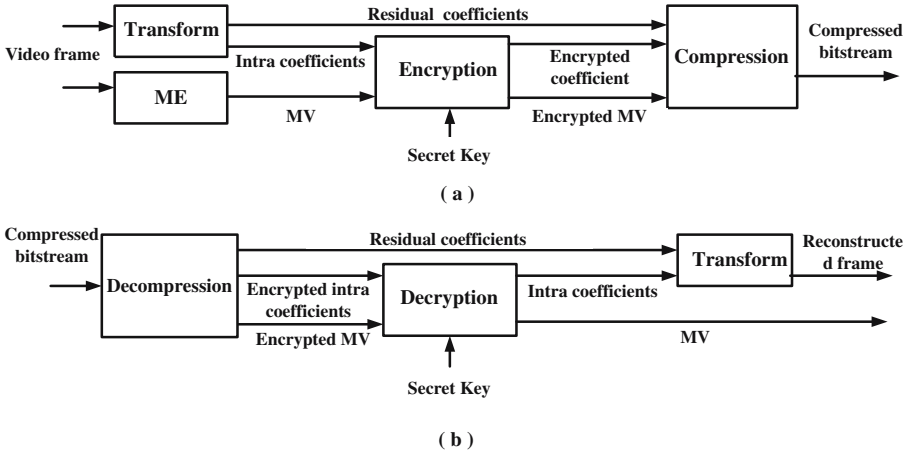


Fig. 1. The encryption /decryption processing in the proposed DRM system. (a) Encryption. (b) Decryption.

2 Proposed Encryption Technique

Before explaining the proposed encryption technique in detail, we first introduce the concept of the proposed joint and partial encryption. The joint encryption method can provide levels of security for contents encryption. The security level can be determined by how many independent encryption methods are combined.

The partial encryption method obtains the higher security by simply encrypting only significant parts of the compressed data. Next, we provide a detailed explanation on the proposed encryption technique.

In order to protect the contents against eavesdropping and moreover against illegal mass distribution after descrambling, we propose an A/V watermarking-scrambling method based on our previous research results [13]. Fig 1 shows the block diagram of the proposed encryption/decryption algorithms for the video data. In the first step, DCT coefficients of macroblocks (MBs) in an intraframe and MVs of an interframe are encrypted. We employ two types of encryption methods. The first is an encryption of the DCT coefficients of MBs in an intraframe which were extracted from the MPEG-4 bitstream. The second is an encryption of MVs of an interframe. The encryption/ decryption algorithms are very simple to implement. In addition, encryption is performed prior to compression process consisting of quantization, huffman, run-length, entropy coding, and so on. At the decoder, the compressed video bitstream is first decompressed by entropy decoding and dequantization. The authorized device has Security Key to decrypt the decompressed coefficients (MVs) prior to inverse transformation (motion compensation).

2.1 Proposed Segment-Based DCT Coefficient (S-DCT) Encryption for Intraframes

To encrypt intraframes in MPEG-4, the proposed S-DCTC encryption method is applied to each MB in intraframes. The conventional approaches for encryption change the sign bit of each coefficient in the 8×8 DCT based frames [4] . This sign encryption efficiently provides an impact on distortion. However, in this method, high computational complexity is required to encrypt all coefficients for each MB. And it is easy to crack the encryption since the encryption scheme is very simple. To solve these problems, we propose the S-DCTC encryption method that scrambles the sign bits of DC and AC coefficients. In the proposed method, in order to reduce computational complexity, we first group several MBs into a segment. Instead of encrypting all coefficients, we encrypt the biggest DC coefficient among DC coefficients of MBs in the segment and the biggest AC coefficient of each MB. Changing sign bits of DC coefficients or AC coefficients heavily depends on the index of MBs in Table 1. As a result, Fig. 2 shows the effectiveness on efficient distortion for the sign-encryption of each DC or AC coefficients in intraframes. Each

Table 1. Intra / Inter encryption more set based on Public Key

Key	Intra frame	Inter frame
0	$DC' = -DC$	$MV' = MV + \Delta\alpha$
1	$AC' = -AC$	$MV' = -MV$
...	...	...
i	DCT sign encryption	MV sign/phase angle encryption

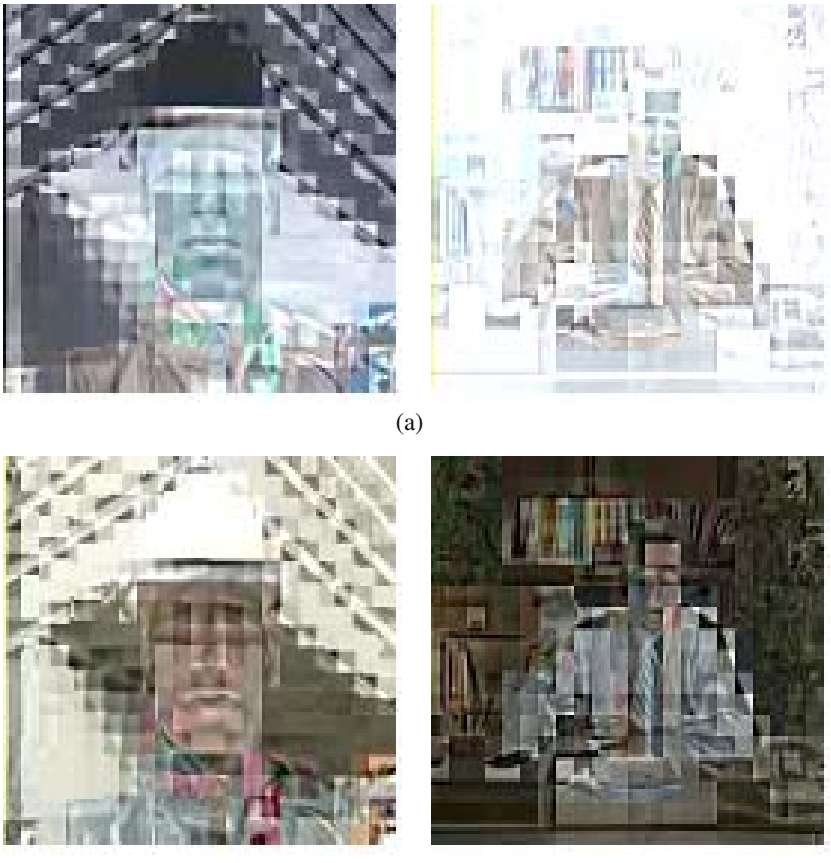


Fig. 2. Picture of encrypted DCT coefficient. (a) DC sign encryption. (b) AC sign encryption.

encryption mode is defined as shown in Table 1, where the index k is the number of MBs in the segment. This encryption mode set in Table I is encrypted with Device Key at Contents Provider and transmitted to User. Note that the encrypted mode set is considered as Secret Key in the proposed DRM system. Therefore, a user who does not have the same key for decryption can not access the original contents without visual quality degradation.

2.2 Proposed MV Encryption for Interframes

In MPEG-4 video, DCT coefficients of interframes such as P or B-frames represent residual errors remained by motion compensation. In this case, DCT coefficients have typically small values. Thus, it is not good enough to apply the S-DCTC encryption method for interframes. That is, for unauthorized users, we

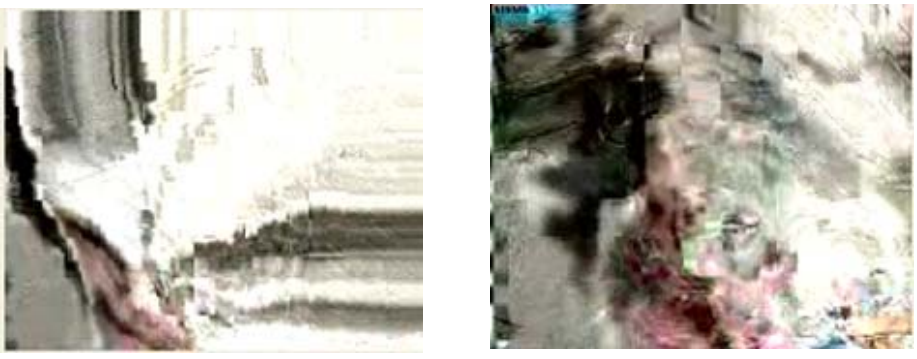
Fig. 3. Picture of encrypted MV. (a) MV phase angle encryption. (b) MV sign encryption.

can not guarantee visual quality degradation caused by video encryption [11,12]. In order to solve this problem, we propose two video encryption methods scrambling MVs in interframes of MPEG-4 video: phase angle and sign scrambling. The phase angle, θ, is calculated as

$$\theta[i] = \arctan\left[\frac{MV_v[i]}{MV_h[i]}\right] (0 \leq i \leq MB), \tag{1}$$

where MV_v and MV_h, respectively, are the vertical and horizontal components of an MV. θ can be changed by adding weighting factors to each component. If the vertical and horizontal components of an MV are changed, the phase angle of an MV is modified. Modified MV, MV' , is given by

$$\begin{aligned} MV'_v &= MV_v + \Delta v, \\ MV'_h &= MV_h + \Delta h. \end{aligned} \tag{2}$$



(a) (b)

Fig. 4. Encryption image of proposed method. (a) Original image. (b) Proposed encrypted image.

The MV sign encryption scheme is very simple to implement. It is only flipping the direction of an MV with Secret Key. The direction of an MV is changed as follows:

$$\begin{aligned} MV'_v &= -MV_v, \\ MV'_h &= -MV_h. \end{aligned} \tag{3}$$

Fig. 3 shows the result of MV encryption. MV phase encryption methods are applied in Fig. 3(a). Fig. 3(b) implements MV sign encryption methods. In other case, Fig. 5 shows that differential encrypting methods. DC sign and MV phase angle encryption methods are applied in Fig. 5(a). Fig. 5(b) shows the result of AC sign and MV sign encryption methods. It provides a very good compromise between compression ratio and coding efficiency. It only increases a little bit overhead while encrypting MVs.

3 Experimental Result

Through simulation, we have compared the performance of the proposed method with the existing methods. And experimental testing is performed on the "FOREMAN" and "STEFAN" sequences with CIF format. Fig. 4 shows the displays of the encrypted video on unauthorized decoder and authorized decoder. The encryption is applied to MPEG-4 Simple Profile @ Level 3 (SP@L3).

Table 2 shows the performance comparison between compression time ratio and compression bit overhead including the process of compression and encryption with the original image. This encryption has a negligible impact on the bitrate and very low computational cost. After encryption, the bit overhead of the encryption does not exceed above 3% in comparison with the bitrate at the encoding without encryption. The amount of computation in the encryption process is also little and the compression time ratio is very short. As a result, Table 3 summarizes the performance of the combinations with the proposed encryption

Table 2. Comparisons of different encryption method for the 300 frames of "FOREMAN, STEFAN"

Encryption method	FOREMAN (CIF)		STEFAN (CIF)	
	Compression time ratio	Bit overhead (%)	Compression time ratio	Bit overhead (%)
No encryption	1	1	1	1
DC sign	1	1	1	1
AC sign	1	1	1	1
MV sign	1	1	1	1
MV phase angle	1.2	1.4	1.5	2.5
DC sign	1.2	1.4	1.5	2.5
+ MV phase angle	1.2	1.4	1.5	2.5
AC sign + MV sign	1	1	1	1
Proposed method (DC+AC+MV sign and MV phase angle)	1.2	1.4	1.5	2.5

Table 3. Comparison of differnet scrambling techniques for the FOREMAN sequence, all at the same PSNR

Algorithm	Bit rate (kbit/s)	Bit Overhead	Complexity	Security
No scrambling	47.23	0	N/A	No
Sign+Slice+MV(sign) [5]	53.38	23.6	20(%), +slice memory	high
Coefficient shuffling within block [4]	73.21	55	very simple	low
Proposed A/V Encryption S-DCTC + MV	52.68	11.2	+slice memory (shuffling table)	high+

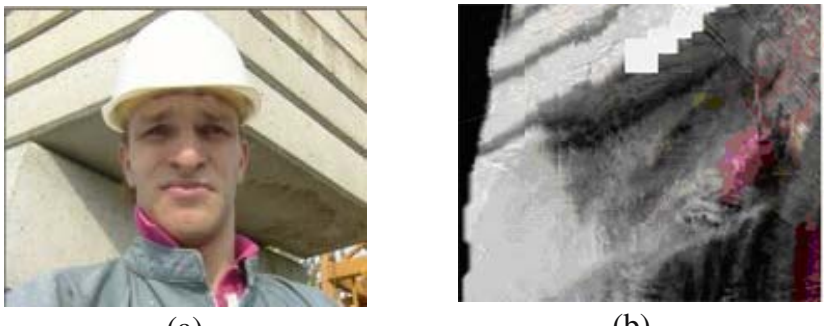


Fig. 5. Proposed hybrid encryption. (a) DC sign + MV phase angle encryption. (b) AC sign + MV sign encryption.

method. In [5], several scrambling algorithms are mixed. Therefore, security level is high but scrambling overhead increases the bit rate by about 23.6%. The implementation of [4] is very simple but overhead increment is about 55%. Note that in Table VI the proposed method outperforms conventional methods [4,5] in terms of the scrambling overhead, complexity, the security. Hence, the proposed method achieves low bit overhead and complexity and also provides levels of security with the combination of encryption methods.

4 Conclusions

We have presented the new solution for DRM called joint and partial encryption. DCT coefficients and MVs in MPEG-4 coder were used for the encryption of the video. The scrambling technique provides that the users who have Secret Key can only access the scrambled MPEG-4 contents and moreover protect illegal copies of descrambled contents by using the robust watermark. In addition, we present the minimal cost encryption scheme for securing the copyrighted MPEG-4 A/V data using the DES encryption technique. And the proposed scrambling and watermarking techniques achieve a very good compromise between several desirable properties such as speed, security, and file size. Therefore, it is very suitable for network real-time A/V applications.

Acknowledgments

This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul Industry-Academy-Research Cooperation Project.

References

1. Schneier, B.: Applied cryptography. Wiley and Sons. (1996)
2. Piva, A., Bartolini, F. and Barni, M.: Managing copyright in open networks. IEEE Internet Computing. **6** (2002) 18–26
3. Petitcolas, F. A. P., Anderson, R. J. and Kuhn, M. G.: Information hiding: a survey. Proceedings of the IEEE. **87** (1999) 1062–1078
4. Tang, L.: Methods for encrypting and decrypting MPEG video data efficiently. in Proc. ACM Multimedia 96. (1996) 219–230
5. Zeng, W. and Lei. S.: Efficient frequency domain selective scrambling of digital video. IEEE Trans. on Multimedia. **5** (2003) 118–129
6. Cox, I. J., Miller, M. L. and Bloom, J. A.: Watermarking applications and their properties. Proc. IEEE International Conference on Information Technology. Coding and Computing. (2000) 6–10
7. Khan, J. Y., and Das, P.: MPEG4 Video over Packet Switched Connection of the WCDMA Air Interface. Proc. of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications. **5** (2002) 2189–2193
8. Gall, D. L.: MPEG: a video compression standard for multimedia applications. Communications of the ACM. **34** (1991) 46–58

9. Spanos, G. and Maples, T.: Performance Study of a Selective Encryption Scheme for the Security of Networked, Real-time Video. Proceedings of 4th International Conference on Computer Communications and Networks. (1995)
10. Agi, I. and Gong, L.: An Empirical Study of Mpeg Video Transmissions. In Proceedings of the Internet Society Symposium on Network and Distributed System Security. (1996) 137–144
11. Qiao, L. and Nahrstedt, K.: A New Algorithm for MPEG Video Encryption. Proceedings of The First International Conference on Imaging Science, Systems, and Technology (CISST97). (1997) 21–29
12. Qiao, L. and Nahrstedt, K.: Comparison of MPEG encryption algorithms. Computers and Graphics. **22** (1998) 437–448
13. Kwon, G. R., Lee, T. Y., Kim, K. H., Jin, J. D., and Ko, S. J.: Multimedia digital right management using selective scrambling for mobile handset. LNAI. **3802** (2005) 1098–1103
14. EPIC(Electronic Privacy Information Center) Digital Right Management and Privacy Page. http://www.epic.org/privacy/drm/

Data Hiding in Film Grain

Dekun Zou, Jun Tian, Jeffrey Bloom, and Jiefu Zhai

Thomson Corporate Research
2 Independence Way, Princeton, NJ 08540, USA
{dekun.zou, jun.tian, jeffrey.bloom, jiefu.zhai1}@thomson.net

Abstract. This paper presents a data hiding technique based on a new compression enhancement called Film Grain Technology. Film grain is a mid-frequency noise-like pattern naturally appearing in imagery captured on film. The Film Grain Technology is a method for modeling and removing the film grain, thus enhancing the compression efficiency, and then using the model parameters to create synthetic film grain at the decoder. We propose slight modifications to the decoder that enable the synthetic film grain to represent metadata available at the decoder. We examine a number of implementation approaches and report results of fidelity and robustness experiments.

Keywords: digital watermarking, data hiding, video watermarking, film grain, film grain technology.

1 Introduction

When a flat field is captured by an optical camera and transferred to film and that film is subsequently developed and printed, the result is not a flat field. The resulting field has a "grain" texture known as *film grain*. The grain pattern in a single frame is well modeled by band-limited Gaussian noise whose variance and pass-band are determined by the kind of film stock used, the development and printing processes applied and the brightness of the underlying imagery. Film grain is well known to *Directors of Photography* (DP) on a motion picture project. The DP selects the film stock and specifies the processing to obtain a desired film grain effect.

The inspiration behind this paper is that the film grain pattern in a motion picture has a striking similarity to the white noise reference patterns typically used in digital watermarking. We investigate some possible techniques for replacing the natural film grain inherent in a motion picture with a synthetic film grain designed to encode some metadata about the content.

While film grain enhances the *feel* of a motion picture, it also makes compression of that content more difficult as the pattern is noise-like and independent from that of the adjacent frames. The two obvious options are to spend more bits to maintain the film grain or sacrifice the film grain allowing it to be lost during quantization. However, recent work has provided a third alternative [2]. Special filters have been developed to estimate and model the film grain inherent in each frame. Once an estimate of the film grain has been obtained, it is removed leaving a version of the content that does not

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 197 – 211, 2006.

contain any film grain. This version of the content is easier to compress. The model parameters are then sent along with the compressed content to the decoder. At the decoder, a film grain synthesizer uses the model parameters to synthesize a film grain that has similar properties as the original, removed grain. The synthesized grain is then added back to the reconstructed imagery. This technique, known as Film Grain Technology (FGT), has been incorporated in a number of international standards and provides a platform for implementing a data hiding scheme.

We begin this paper with an introduction to Film Grain Technology and then provide specific implementation details for a modification that can use the film grain to represent metadata. Finally, we present some experimental results and discuss next steps.

2 Film Grain Technology

Motion pictures are formed by exposure and development of photographic emulsion. They typically contain some kind of noise, which is called *film grain.* Film grain originates in the physical process of exposure and development of photographic film and exhibits quasi-random characteristics. Film grain forms in different intensities, sizes, and colors depending on the film stock, the developing and printing processes and the brightness of the underlying imagery. Human eyes can not distinguish a single grain; instead groups of these random patterns can be identified more easily. It is clearly noticeable at high resolution and therefore has become a distinguished trait that should be preserved during compression. Figure 1 gives examples of various film grain patterns that differ in intensity and grain size.

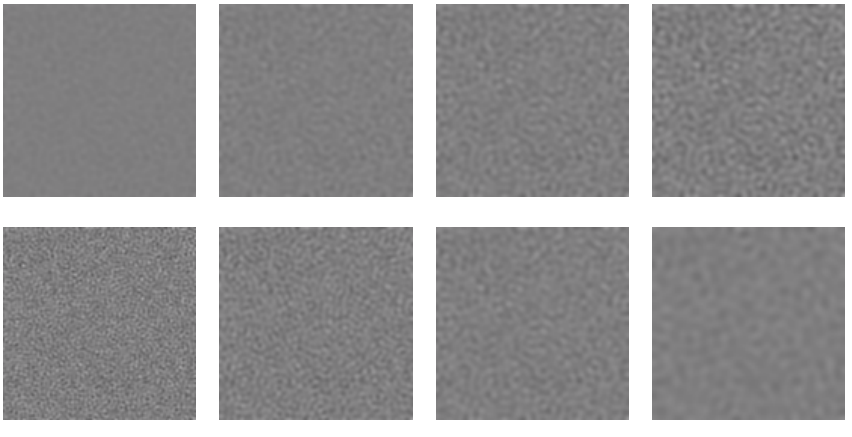


Fig. 1. Illustration of various film grain patterns

As we have mentioned, film grain is generally regarded as quasi-random pattern and contains very high entropy, which makes it very hard to compress. There are a few mathematical models which could be used to generate artificial film grain. In recent work, Gomila [2] presents Film Grain Technology as a new tool that allows

encoding of film grain in motion pictures by means of a parameterized model which is transmitted as supplemental information. To support FGT, the MPEG-4 AVC standard has defined a film grain characteristics Supplemental Enhancement Information (SEI) message in the Fidelity Range Extension (FREXt) Amendment [3][4]. The FGT has also been adopted in HD-DVD as a mandatory part of the format specifications [5].

The basic framework of FGT is illustrated in Figure 2. At the server side, the input video is sent to film grain modeler and film grain remover[1] simultaneously. The result of the film grain removal process is then compressed and the results of film grain modeler are encapsulated in SEI messages, which can be sent independently or embedded into compressed bit-stream for transmission. At the receiver side, if the SMPTE Specifications [4] are enforced, both the SEI messages and the output of the video decoder are sent to a *film grain simulator*, where bit-accurate film grain simulation is done. Bit accurate here has the same meaning as in video decoding specifications such as H.264/AVC: regardless of the means by which film grain simulation is implemented, the result of generated film grain should be exactly the same as the one generated by a reference implementation.

A method for simulating film grain is preformed by generating random noise pattern and passing that pattern through a separable, 2D band-pass filter. Thus, the characteristics of a pattern are specified by the Gaussian variance (controlling the intensity of the grain) and the four cutoff frequencies of the band-pass filter which characterize the size of grain.

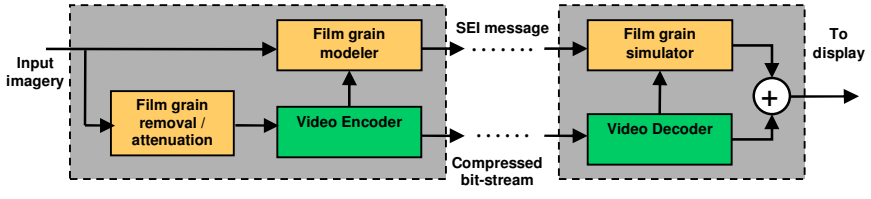


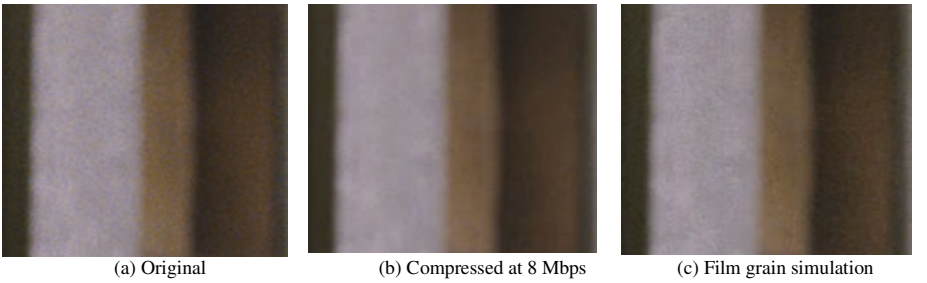
Fig. 2. Framework of Film Grain Technology (FGT)

If the SMPTE Specifications are enforced, in the context of the H.264/AVC, the synthesized film grain is generated and added on an 8×8 block basis. Thus, this frequency filtering model is performed block by block. Conceptually the generation of film grain is done in the following way: for each 8×8 block an 8×8 Gaussian random field is generated. The variance of the Gaussian random variable is controlled by the film grain intensity parameter, which characterizes the strength of the grain. The size of the grain is controlled by a frequency filtering using the four cutoff frequencies: lower and upper horizontal cutoff and lower and upper vertical cutoff. In practice, this filtering is often done in the DCT domain and the two band-pass filters are actually low-pass filters. Using low-pass instead of band-pass is a simplification in order to reduce complexity. Such simplification supports most of the applications

[1] Film grain removal is actually an optional function as we expect the compression to essentially remove any film grain.

without adversely affecting the visual quality. The intensity of the film grain pattern is actually dependent on the brightness of the underlying image data. The pattern intensity typically peaks at mid-intensity and decays in dark and bright regions. In very dark or very light regions, the pattern intensity can often be zero indicating that film grain is not visible in such extremes. One artifact of block-based film grain generation is that the full frame pattern can have blocking artifacts. To reduce these artifacts, a simple deblocking scheme is applied to the generated film grain. In the FGT specification for HD-DVD [4], the above steps are simplified by employing lookup table for low complexity implementation.

Fig. 3. Test frame: 704x480 crop from a frame of the StEM clip



(a) Original (b) Compressed at 8 Mbps (c) Film grain simulation

Fig. 4. Film grain simulation results on a close-up of one small part of the tested frame of Figure3

Figure 3 shows a 704×480 crop from a frame of the DCI-ASC mini-movie known as StEM (the original frame from which this is cropped is 1920×1080). In order to illustrate the film grain, we have further cropped a small region from the upper left and shown scaled up versions in Figure 4. Figure 4(a) shows the portion of the original picture. Figure 4(b) shows this same region after H.264 compression at 8 Mbps. As can be seen, the film grain pattern has largely been removed by the compression. The initial film grain has been modeled and simulated. Figure 4(c) shows the result after the film grain simulation is added to the image of Figure 4(b). Comparison of the three images in Figure 4 illustrates that the FGT approach can restore some of the film grain warmth to the compressed imagery.

3 FGT-Based Watermarking Techniques

A simple approach to using film grain for data hiding is shown in Figure 5. The film grain in the input content is modeled and removed. The model parameters are used to generate one or many synthetic film grain patterns and the watermark payload is used to select or modulate these patterns before they are added back to the "cleaned" version of the content.

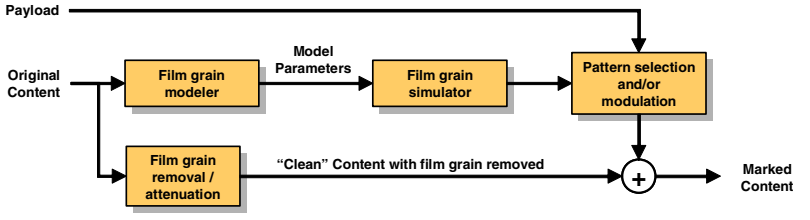


Fig. 5. Simplified approach to film grain data hiding

This basic idea must now be placed in the context of the FGT framework that was illustrated in Figure 2. The main difference is that the "clean" version of the content is subject to a compression/decompression cycle before the synthetic film grain is introduced. In addition, the generation of the synthetic film grain, and thus the incorporation of a watermark payload, takes place at the client, not at the server. This is shown in Figure 6 below.

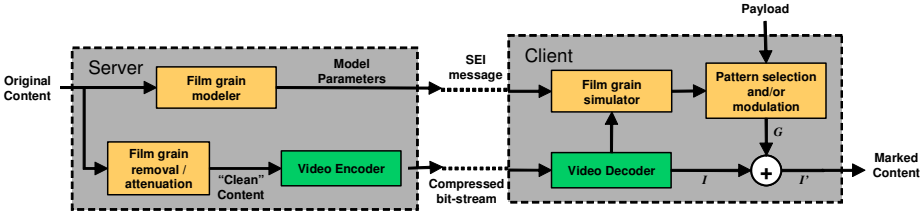


Fig. 6. Data hiding in the context of the FGT framework

Introduction of the watermark payload at the client has a number of implications on the applications for which this technology would be appropriate. First, there is a security implication as the client necessarily has access to an important part of the watermark embedder. Second, this approach now lends itself to applications in which the data to be embedded is available only at the client. For example, the client may introduce its own ID into the content. The client may introduce a time stamp indicating the playout time. The client may be redistributing the uncompressed content to a number of other processes and thus use the payload to identify each of those processes.

The basic idea is to make the film grain pattern payload dependent. Several approaches are proposed. For this discussion, let I denotes a decompressed picture before film grain addition and let G denote a film grain pattern, which has the same

dimensions as the original image. Finally, let I' denotes the image with film grain added.

As discussed in Chapter 4 of Cox, Miller, Bloom [1], there are a number of ways to select or modulate patterns in order to represent messages. The direct message coding method designates a different pattern for each message. If the message, in this case, is the ID of the client, this can be accomplished by allowing each client to seed the film grain pattern generator with a different initial seed. This approach is examined further in Section 0. It is limited, however, to applications with few messages.

For more flexibility in coding arbitrary payloads (up to a maximum length), an alphabet of symbols or film grain patterns is defined and multiplexing is used to combine the symbols. For illustration purposes, Section 0 presents a time division multiplexing approach where one symbol is embedded into each frame of the image sequence.

Both approaches rely on the fact that the visual properties of the synthesized film grain pattern are based on the model parameters and do not depend on the specific key used to generate that pattern. The use of different seeds will result in different film grain patterns all with the same visual properties. Thus, data is embedded by selection of one or another visually equivalent pattern.

3.1 Direct Coding Using Unique Client Seed

In FGT, the film grain patterns are generated block by block. For each block, a seed is used to generate the pattern. For one block, the seed is determined by the seed used in the previous block. A function is used to transform one seed into the next. The details can be found in the FGT specifications for various standards (e.g., [4]). The seeds are reinitialized after each SEI message is encountered. Therefore, once the initial seed is set, the whole pattern will be determined. Different clients can be assigned different initial seeds. The embedded film grain pattern will then be different for each user. The embedding procedure is illustrated in Figure 7.

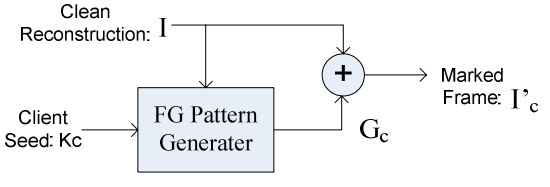


Fig. 7. Direct coding FG embedding

For each frame, the detection process begins with a registration. For the current implementation we register with the original content, but other techniques can be used (see Section 8.3 of [1]). The *original* picture is then subtracted from the suspect picture. Here, the image used as the *original* is actually the decompressed clean image; Image *I* from Figure 7. The remaining difference is an estimate of the embedded film grain pattern. Since the film grain pattern generation process is deterministic given a particular initial seed and the clean content, and assuming that the initial seeds of each of the clients is known to the detector, a library of reference

patterns can be generated at the detector. The reference pattern with the highest correlation to the extracted film grain estimate is the pattern most likely to have been embedded. This, in turn, identifies the most likely client. Figure 8 shows the detection procedure.

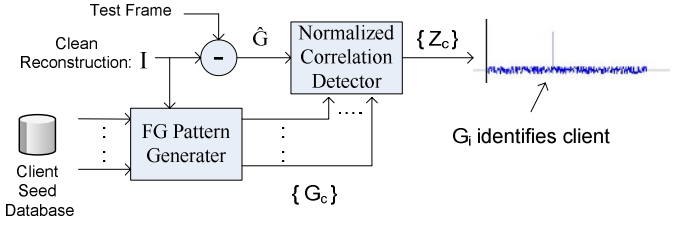


Fig. 8. Direct decoding of FG patterns

3.2 Space / Time Division Multiplexing

Since the synthetic film grain patterns are generated on a block by block basis, there is the opportunity to embed a different symbol into each block. Using a symbol alphabet of size 2, this technique would embed one coded bit in each 8×8 block of image data corresponding to 5280 coded bits in each 704×480 frame or 32400 coded bits in a 1920×1080 frame. The data rate can be increased by a factor of n by defining a set of 2^n different film grain patterns for each block.

It is possible that the intensity of the film grain pattern in some blocks will be zero. This will be the case for very bright and very dark regions. Any information scheduled to be stored in these blocks would be lost. There are many ways to address this including reliance on error correction coding or protocols for skipping blocks that cannot reliable hold information. Alternatively, an entire frame or even groups of frames can be used to embed each symbol. Using groups of frames would provide some robustness to dropping of frames as can occur in advanced coding schemes. In the experiments of Section 0, we embed one symbol in each frame.

In the remainder of this section, we present two different methods for embedding one bit per frame. Both of these methods are constructed around the specific details of FGT. Specifically, it has been mentioned that the synthetic film grain patterns are generated on a block by block basis and that the patterns are dependent on both the local intensity of the imagery and the cutoff frequencies specified in the model. In practice, these patterns are precalculated and stored in a database. For each combination of cutoff frequencies, the database contains an 8x8 array of 8x8 patterns. Allowing for a 4-pixel overlap in the horizontal direction, this yields 120 different patterns. Also available are the inverses of these patterns. The client uses a deterministic procedure to select one of the 240 available patterns for the block and then modulates that according to the local image intensity.

The method that we present partitions the set of 240 patterns into two sets. One set represents a '0' value bit and the other set represents a '1' value bit. Two variations are presented. These two differ primarily in the assumption we make about information shared between the embedder and detector.

3.2.1 Shared Seed

The first variation presented assumes that the pattern selection function uses a seed value that is known by the detector. This would be the case when all clients use the same seed. The set of 120 8×8 film grain patterns are collectively referred to as the '1' set and the set of 120 inverse patterns are the '0' set. The seed value for each block is used to select one pattern from the '1' set. This continues for each block in the frame generating a full frame film grain image. If the bit for the current frame is also a '1', then the generated pattern is used. If the bit is a '0', then the inverse pattern is used.

Since the detector has the same seed, it can also create the '1' pattern which is used as a reference pattern for correlation analysis. If the correlation is positive, then we conclude that a '1' bit was embedded. If the correlation is negative, then we conclude that a '0' bit was embedded.

For a given block, the embedding is depicted in Figure 9 and can be approximated as in Equations (1), where G_k is the k^{th} 8×8 pattern in the '1' set. This is only an approximation because a deblocking filter is subsequently applied across the blocks.

$$I'_k = I_k + \alpha G_k \text{ where } \alpha = \begin{cases} -1 & b = 0 \\ +1 & b = 1 \end{cases} \tag{1}$$

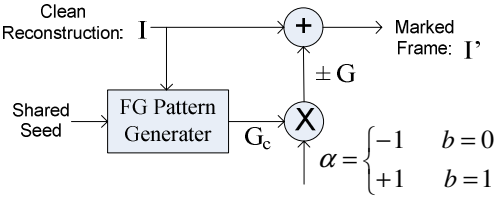


Fig. 9. Multiplex FG embedding with a shared seed

The detection procedure is much simple than previous methods. Since the pattern we have added or subtracted is already known, the sign of the correlation between this pattern and the difference picture will determine the bit embedded in this block. Figure 10 illustrates the detection process. Each frame of the test sequence is registered and compared to the corresponding frame of sequence *I*. The difference is an estimate of the added film grain pattern and is correlated with the reference pattern for that frame. A positive correlation suggests a '1' bit and a negative correlation suggests a '0' bit.

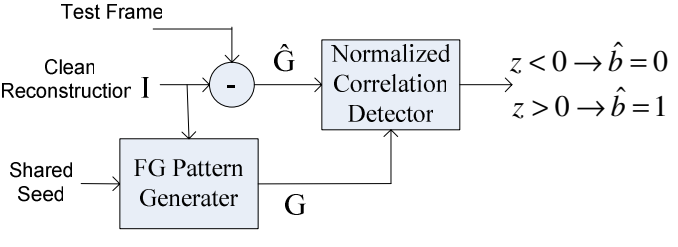


Fig. 10. Multiplex FG detection with a shared seed

3.2.2 Detection with Unknown Seed

There may be applications in which the detector cannot reproduce the pattern selection process used during embedding. This would be the case when each client uses a differ-rent seed for the selection process. To address this problem we partition the set of 240 8×8 patterns differently (recall there are 120 patterns in the database plus the inverse of each.

Before introducing the partition strategy, let us first assume that the set has been appropriately partitioned into two sets of 120 patterns each, one set labeled the '0' set and one labeled the '1' set. To embed a '0', a film grain pattern from the '0' set is selected for each block of the frame. To embed a '1', a pattern from the '1' set is selected for each block of the frame. Each block has a different pattern, but all the patterns chosen for a particular frame come from the same set. In order to recover the bit without being able to reconstruct the selection process, the detector will correlate the film grain estimate from each block with each of the 240 reference patterns in the database[2]. The pattern that yields the highest correlation is the pattern most likely to have been embedded. This process is repeated for each block and all of the results are combined (e.g., by voting) to obtain the recovered bit value.

If we were designing the database of patterns for this purpose, we would want each of the patterns in the '0' set to be orthogonal to all of the patterns in the '1' set and each of the patterns in the '1' set to be orthogonal to all of the patterns in the '0' set. This would minimize the likelihood that a block in which a '1' had been embedded ends up having a high correlation with a '0' pattern and vice versa. However, we do not have the luxury of designing the database. It has already been designed and is written in the FGT specification. The next best thing is to partition the set of 240 patterns so as to minimize the maximum correlation across the two sets. Two patterns with high cross correlation should be placed in the same set.

For block k, the embedding can be approximated by Equation (2) where G_k^0 denotes a randomly selected pattern from the '0' set and G_k^1 denotes a pattern randomly selected from the '1' set. Again, this is an approximation because a deblocking filter is subsequently applied across blocks. Figure 11 illustrates the embedding process.

$$I'_k = \begin{cases} I_k + G_k^{\,0} & b = 0 \\ I_k + G_k^{\,1} & b = 1 \end{cases} \tag{2}$$

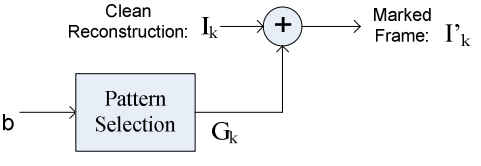


Fig. 11. Multiplex FG embedding with a unique seed

The detection procedure is illustrated in Figure 12. The original movie is used for both geometric/temporal registration and subtraction. Each extracted difference image

[2] In fact, only 120 correlations are necessary since the correlation with one pattern will have the same magnitude and opposite sign as the correlation with the inverse of the pattern.

is divided into blocks and correlation detection will determine which pattern was most likely embedded.

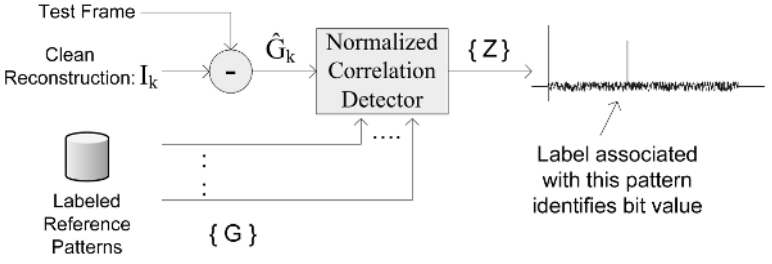


Fig. 12. Multiplex FG watermark detection without knowledge of embedding seed

4 Properties

The approach presented here results in watermarked imagery that has the same fidelity as FGT itself. The FGT film grain estimation, modeling, and synthesis process results in imagery with convincing film grain after decompression. Each frame, and even each block within each frame, contains a different, signal independent film grain pattern. The data hiding approach simply takes control over the synthesis process, forcing the output to specific states specified by the data payload rather than allowing it to be completely random. In Section 0 we describe the experimental method used to confirm the fidelity. Note that the noise-like watermark pattern is not designed to be imperceptible. It is designed to be indistinguishable from the synthetic film grain pattern normally introduced by FGT during decompression.

The detection presented here is an informed detection. A version of the original content, with film grain already removed, is used for registration. In addition, it is used to remove as much of the host image as possible leaving an estimate of the added synthetic film grain for detection. This detection process is described more fully in Section 0. Note that generation of the reference patterns requires knowledge of the film grain model parameters obtained from the original content as well as intensity information from the original content and, potentially a key unique to the particular title. Therefore, even with an alternative registration approach, this method does not lend itself to blind detection.

Film grain watermarking has proven to be robust to a number of different processes. In Section 0, we present data for embedding efficiency as well as robustness data for distortions due to compression and noise removal that both tend to remove film grain.

Since the reference patterns are published in the FGT specification, it will be difficult to create a secure application using the multiplexing methods of Section 0. However, the direct coding method has more potential for security. In general, we recommend use of the methods presented here for applications that do not require security against unauthorized removal.

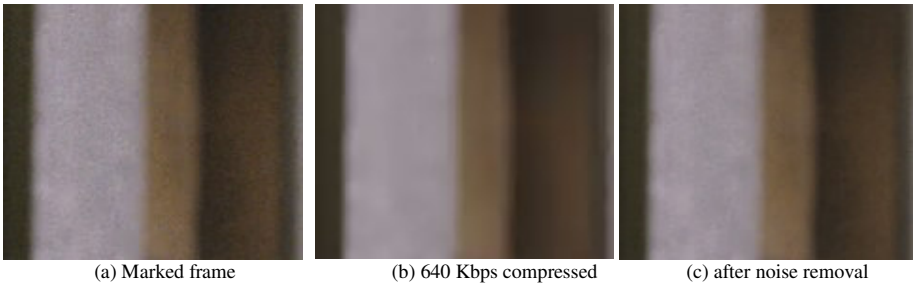
5 Experiments

The shared seed temporal multiplex approach of Section 0 was implemented because it is the closest to the existing FGT standard requiring the fewest changes to the existing film grain generators. Parameters are estimated and the resulting imagery is examined by viewing experts. If any block on any frame has an undesirable synthetic film grain appearance, the parameters can be adjusted. The result of this tedious process is a predefined set of patterns to be used for each frame. Every client process will produce the exact same, "bit accurate", reconstruction as that approved by the viewing experts. The premise used in these experiments is that use of the inverse pattern will be equally acceptable. Informal fidelity assessments by our local film grain experts support this premise.

Both fidelity and robustness of proposed FGT based video watermark methods were tested using 1437 frames from the DCI-ASC Mini-movie, StEM. The resolution is 704×480.

5.1 Fidelity

For fidelity assessment we compare a sequence with standard FGT film grain (not representing any payload data) with a watermarked sequence in which the film grain does represent payload data. Clearly, this experiment need not be done but only to verify that there are no unanticipated consequences. Viewers included 2 film grain experts and 4 other viewers experienced in viewing compression artifacts, but not necessarily film grain artifacts. Side-by-side, the differences in film grain could not be perceived and none of the testers could identify which was the watermarked sequence and which was the sequence with standard film grain. For illustration purpose, consider the image in Figure 4(a) showing an enlargement of a part of the original image with its natural film grain. The same region from a watermarked version of the frame is shown in Figure 13(a). Compare this with the picture in Figure 4(c), where a standard film grain pattern is added.



(a) Marked frame (b) 640 Kbps compressed (c) after noise removal

Fig. 13. Frames after watermark embedding

5.2 Embedding Efficiency

In order to assess the efficiency of the embedder, the embedder output is fed directly into the detection process without any additional distortions. The payload for this

experiment was a random bitstream with an equal number of 1's and 0's. The detection measure used is normalized correlation which has a range of -1 to +1. The reference pattern used is a full frame constructed from the preselected reference blocks corresponding to a 1 bit. A histogram of the resulting correlation values for 1437 frames is shown in Figure 14.

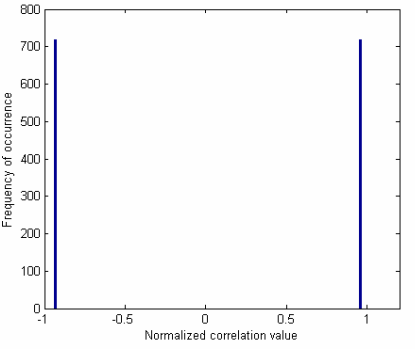


Fig. 14. Watermark efficiency using a reference pattern constructed from the preselected reference blocks corresponding to a 1 bit

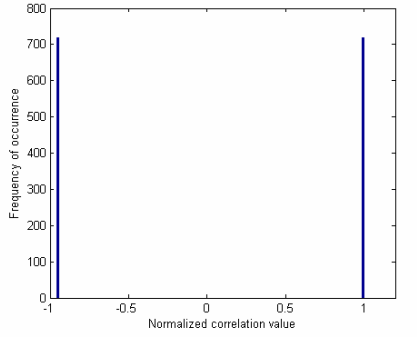


Fig. 15. Watermark efficiency using a deblocked positive reference pattern for detection

Since an equal number of 1s and 0s are embedded, two sharp peaks can be observed in the histogram. The magnitudes of the correlation values are slightly smaller than 1. This is unlikely due to clipping as very dark and very light regions do not get film grain added. We suspect that the reason is that the reference pattern we used is the film grain pattern before deblocking. Therefore, it is slightly different from the pattern actually added. Note that the deblocking filter applied to the inverse pattern does not result in the inverse of the result of the deblocking filter applied to the positive pattern. Therefore, we would need to generate two reference patterns to correctly address this problem. To test this hypothesis, the deblocking filter is applied to the positive reference pattern and the correlations are again performed. The results of this test are shown in Figure 15. As expected, the magnitudes for the 1's case have moved up to +1. Use of the deblocked inverse reference pattern will have the same effect on the 0's case.

5.3 False Positive Probability

In this presentation, we do not provide a formal false positive analysis. We do, however provide a preliminary look at the detection values obtained when the content has not been watermarked. The test material is the original movie clip with its natural film grain. This is provided as input to the detector. Again the number of frames in this test is 1437. The distribution of correlation values is shown in Figure 16. In this small test, no frame yielded a detection value with magnitude greater than 0.01. If these values are typical for various reference patterns and content, we can expect to be able to set a correlation threshold that safely distinguishes between marked, undistorted content (Figure 14) and unmarked content (Figure 16). To confirm this, a larger false positive analysis must be performed.

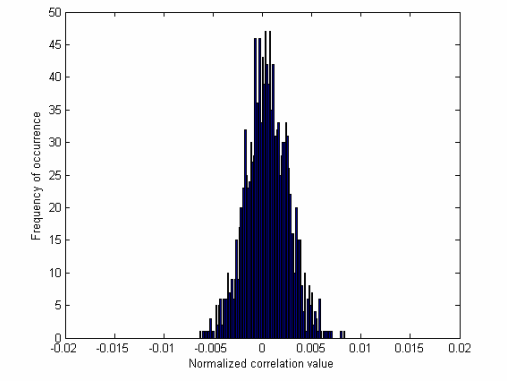
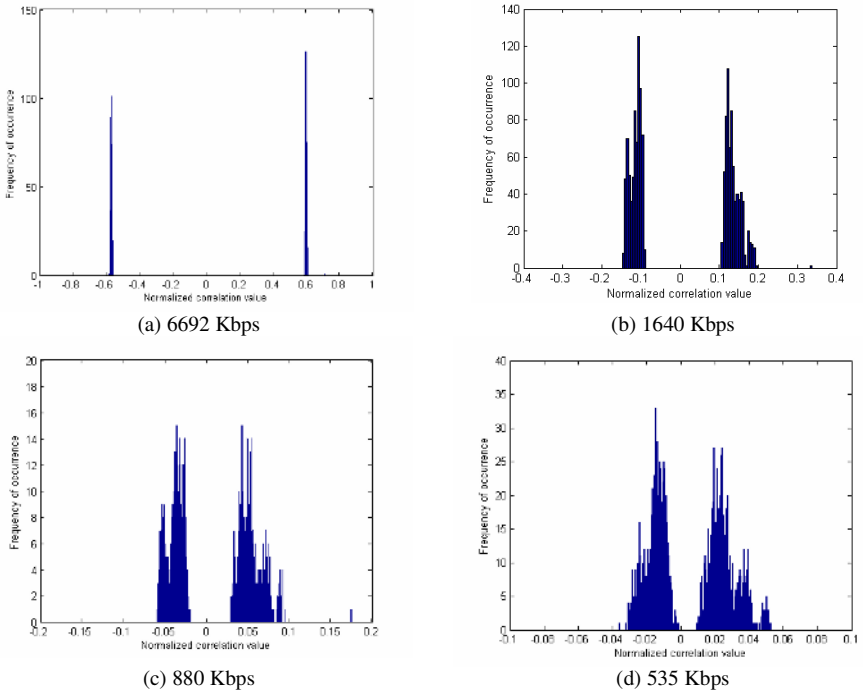


Fig. 16. Correlation distribution with unmarked content. Note the x-axis scale. No frame yielded a detection value with magnitude greater than 0.01.

5.4 Robustness

In this presentation, we examine the robustness of the watermark to lossy compression and Gaussian noise removal. An H.264 encoder/decoder was used for lossy compression. Various compression bitrates were tested from a maximum of



(a) 6692 Kbps (b) 1640 Kbps

(c) 880 Kbps (d) 535 Kbps

Fig. 17. Distribution of detection values after compression with various bit rates

11 Mbps to a minimum of 470 Kbps. Examples of the visual impact of these distortions are shown in Figure 13(b) for compression and Figure 13(c) for noise removal. Perceptually, the film grain has been greatly reduced.

The distribution of detection values for some of the compression cases are shown in Figure 17. These figures show that the magnitudes of the detection values decrease with increased compression. Note that the x-axes are different for the different figures. The average detection magnitudes in these tests are summarized in Table 1. At 535 Kbps, the correlation values begin to overlap the levels seen for unmarked content. This suggests that we may not be able to distinguish, with certainty, between an unmarked sequence and a sequence that has been marked and significantly compressed. At least we may not be able to make this distinction based solely on the detection values. Advanced message coding and the use of error correcting codes can improve the certainty.

However, despite these low detection values, it is important to note that all of the correlations have the correct sign and thus each of the 1437 bits was correctly recovered in each of the robustness experiments performed.

The Gaussian noise removal did remove the visual noise, but enough of the film grain pattern remains for solid detection. The distribution of detection values for the 1437 frames is shown in Fig. 18. Again, all bits were correctly recovered.

Table 1. The mean magnitude of the detections value decreases along with compression bitrate

Compression Bit Rate (Kbps)	Average Detection Value
11741	0.7493
6692	0.5841
1640	0.1256
880	0.0458
535	0.0207
477	0.0167

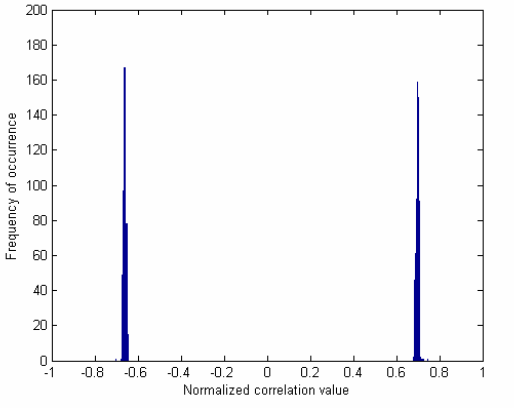


Fig. 18. Distribution of detection values after Gaussian noise removal

6 Conclusion and Next Steps

In this paper, we have presented a data hiding technique custom designed for applications using Film Grain Technology. Although FGT may not be widely known to the watermarking community, it has been incorporated in a number of important international compression standards and will become a widely distributed technology. This work extends the utility of FGT for data hiding applications.

The fidelity of the approach is established by the inability of expert viewers to correctly classify watermarked sequences and standard film grain sequences. This fidelity analysis was performed on a small sample of imagery and must be expanded.

The work presented here took a preliminary look at the false positive characteristics. This was helpful in providing a context for assessing the robustness results. However, a more thorough false positive analysis is required. Such an analysis will examine much more content of differing types and a number of different reference patterns. The resulting histograms can be modeled and the models can be used to establish the relationship between the threshold and the probability of a false positive. In addition to such an empirical study, an analytical study that characterizes the distribution of detection values is desired.

In the robustness experiments described, all of the bits were correctly recovered despite low detection values. Additional robustness experiments are needed to identify the limits at which the bit error rate becomes non-zero and examine the interaction of those errors with an appropriate error correcting code. In addition, we need to assess the robustness to other types of distortions.

References

[1] I. Cox, M. Miller, and J. Bloom: Digital Watermarking: Principles & Practice, San Mateo, CA: Morgan Kaufman, 2001.

[2] C. Gomila: SEI message for film grain encoding. Contribution JVT-H022 to 8th JVT Meeting, Geneva, Switzerland, 23-27 May, 2003.

[3] ITU-T Recommendation H.264 | ISO/IEC 14496-10 International Standard with Amendment 1.

[4] SMPTE RDD 5-2006: Film Grain Technology - Specifications for H.264 | MPEG-4 AVC Bitstreams.

[5] HD DVD-Video Specifications for High Density Read-Only Disc, Version 1.0, August, 2005.

Joint Screening Halftoning and Visual Cryptography for Image Protection*

Chao-Yung Hsu[1,2], Chun-Shien Lu[2,**], and Soo-Chang Pei[1]

[1]Graduate Institute of Communication Eng., National Taiwan University, Taipei, Taiwan, ROC
[2]Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC
lcs@iis.sinica.edu.tw

Abstract. Since digital right management of digital media data has received considerable attention recently, protection of halftone image documents becomes another important topic. Image-based visual cryptography is found to provide an alternative for applications of copyright protection by overlapping more than one secret embedded image to show the hidden information. In this paper, we propose a novel screening halftoning-based visual cryptography method for halftone image protection. Compared with the existing methods, the major contributions of our method contain (i) improved quality of the halftone images and extracted secrets; (ii) unlimited database size of protected halftone images; (iii) more than two halftone images can be overlapped to show the hidden secret; (iv) only one conjugate screen pair in our method is able to achieve the maximum clarity of extracted secrets in random screening. Experimental results and comparisons with a state of the art method demonstrate the effectiveness of our method.

1 Introduction

With the advent of media data digitization and popularization of bi-level devices such as printers, scanners, and fax machines in our daily lives, digital halftoning has been an indispensable technology. Halftoning [8,9,12] refers to the physical process of converting a continuous tone image to a special image format, halftone image, which is composed of white and black dots, as shown in Fig. 1. We can find that the halftone image approximately keeps the visual characteristic of the continuous tone image.

Since digital right management (DRM) of digital media data has received considerable attention recently, protection of halftone image documents becomes another important topic. Two popular embedding-based copyright protection schemes for halftoning images are invisible watermarking and watermarking-based visual cryptography. Invisible watermarking refers to the insertion of invisible watermarks into the multimedia data for copyright protection [2]. The

* This research was supported by the National Science Council NDAP-R & DTD-Digital Archives System Related Technology Research & Development Project: NSC 95-2422-H-001-008 and 94-2422-H-003-007.

** Contact author.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 212–225, 2006.

characteristic of this method is that the copyright of *halftone* images can be verified by the watermark, which is extracted from a *scanned* suspect image. Unfortunately, the geometric distortions, which are induced during the scanning process, have not been efficiently dealt with the currently known halftone image watermarking approaches [4,5]. On the other hand, halftone images can be more efficiently protected by exploiting the unique characteristic of visual cryptography by way of double-side printing, for example. In view of this, this paper will focus on digital halftone image protection by means of watermarking-based visual cryptography.

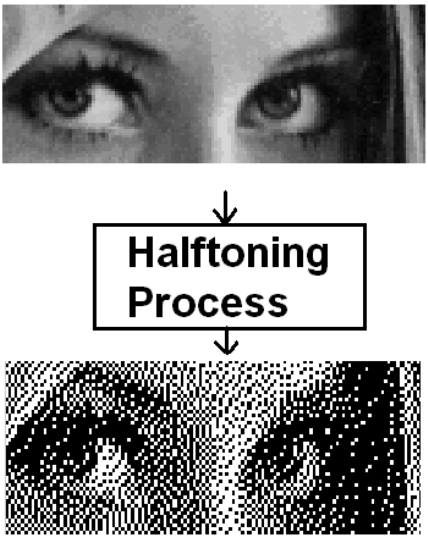


Fig. 1. Digital halftoning process

Visual cryptography [10] is a technique of hiding information into cover data and extracting the hidden information by overlapping more than one stego data. This technique provides an alternative to protect copyrights of halftone images without incurring the process of print-and-scan and avoiding the induced distortions. In the literature, few methods were proposed to combine halftoning and visual cryptography for the purpose of image protection. Fu and Au [3] proposed a method, called DHSED, to hide binary patterns into two images, which are generated by different error diffusion techniques. Specifically, one is generated by regular error diffusion and the other is generated by stochastic error diffusion. If these two images are overlapped, then the hidden visual pattern appears. In [6], they further proposed to use self-conjugate error diffusion for data hiding so that better image quality and more visible visual patterns can be satisfied.

However, the aforementioned methods still cannot be used for copyright protection because each image is required to be processed twice in different

ways. It is foreseeable that the size of the image database is doubled accordingly.

In order to keep the size of an image database unchanged, Pei and Guo [11] proposed a noise-balanced error diffusion technique such that the two to be overlapped halftone images can be generated from different gray-tone images and the extracted information still exhibits acceptable visual quality. The weakness is that each stego image must be restricted to be superimposed with a key image so as to successfully reveal the hidden information. This restriction will pose the problems of insecurity and inflexibility.

In order that the hidden information can be extracted by overlapping *any* two images, Knox [7] proposed a novel halftone image watermarking scheme based on stochastic screen patterns. In this method, a stochastic screen block is first selected and one or more than one stochastic screen blocks that are related to the first one are derived. Then, the first halftone image is generated from the screen image that is yielded from the first screen block and the second halftone image is generated from the screen image that is formed by randomly combining the other screen blocks with the watermark signal. In this study, the watermark signal is composed of two components: the dark component and the bright component, as shown in Fig.2. However, our studies find that Knox's method still exhibits two major disadvantages: worse halftoning quality and limited database size of protected images.

Fig. 2. A watermark signal is composed of the bright and dark components

In this work, we investigate a joint screening halftoning and visual cryptography scheme for image copyright protection. The major differences distinguishing the state of the art technology presented by Knox [7] from ours include: (1) halftone images with better quality can be obtained by two conjugate basic screens; (2) the size of a halftone image database is not limited.

2 Background

Both the technologies of screening halftoning and visual cryptography are briefly described to complete this paper.

2.1 Screening Halftoning

In screening halftoning, a so-called threshold matrix or screen block, as shown in Fig. 3, is needed to perform continue tone-to-halftone transformation. In fact, the output pixel block is independently determined by comparing the corresponding input pixel block with a screen block. Let T be a two-dimensional screen block and let I be an input image. In the implementation, both the elements of T and I are normalized to fall within the interval [0 1] during the halftoning process. Specifically, the screening halftoning process is performed as follows to obtain the halftone image H, whose pixel value is defined as

$$H(x,y) = \begin{cases} 1, & \text{if } I(x,y) \geq T(x,y); \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In Eq. (1), $H(x,y) = 0/1$ denotes that the halftone pixel is black/white.

6	11	7	10
15	1	16	4
8	9	5	12
13	3	14	2

Fig. 3. An example of a 4×4 screen block

2.2 Visual Cryptography

The basic concept of visual cryptography [10] states that a secret message is divided into s partitions, $Share_1$, $Share_2$, $\cdots$, and $Share_s$, which are viewed as

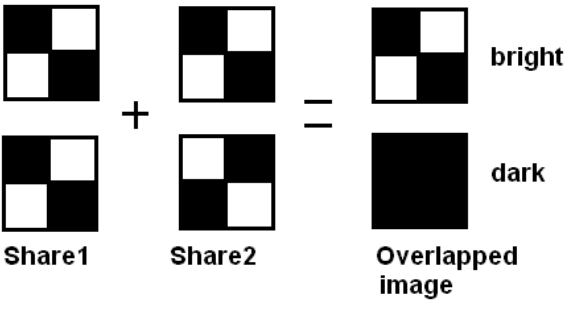


Fig. 4. An example of 2×2 image visual cryptography. In the overlapped image, bright area is labeled with level 0.5 while dark area is labeled with level 0.

random noise images. When any k ($k \leq s$) partitions are overlapped together, the secret will appear on the overlapped image. Here, a secret contains two levels of illumination (see Fig. 2): bright areas are labeled with level 0.5 and level 0 is used to represent dark areas. Fig. 4 illustrates two simple examples of secret sharing from two shares of size 2×2. The first example indicates that if two shares are the same, then bright illumination will be shown, while the second one shows that if two shares are different, then dark illumination will be shown. A practical example of visual cryptography is shown in Fig. 5.

3 Our Method

In [7], Knox proposed to generate halftone images by means of combining one or more stochastic (random) screens. However, we find that combination of random screens results in poor quality of halftone images because random screening cannot disperse block and white dots uniformly, as shown in Fig. 6. We can observe that the halftone image generated by random screening is worse than classic screening in visual quality. On the other hand, if the size of an image database is large, then more random screens are required in order to make sure that any two images are generated from different screen combinations. As a result, how to generate sufficient number of screens is problematic for a large image database in [7].

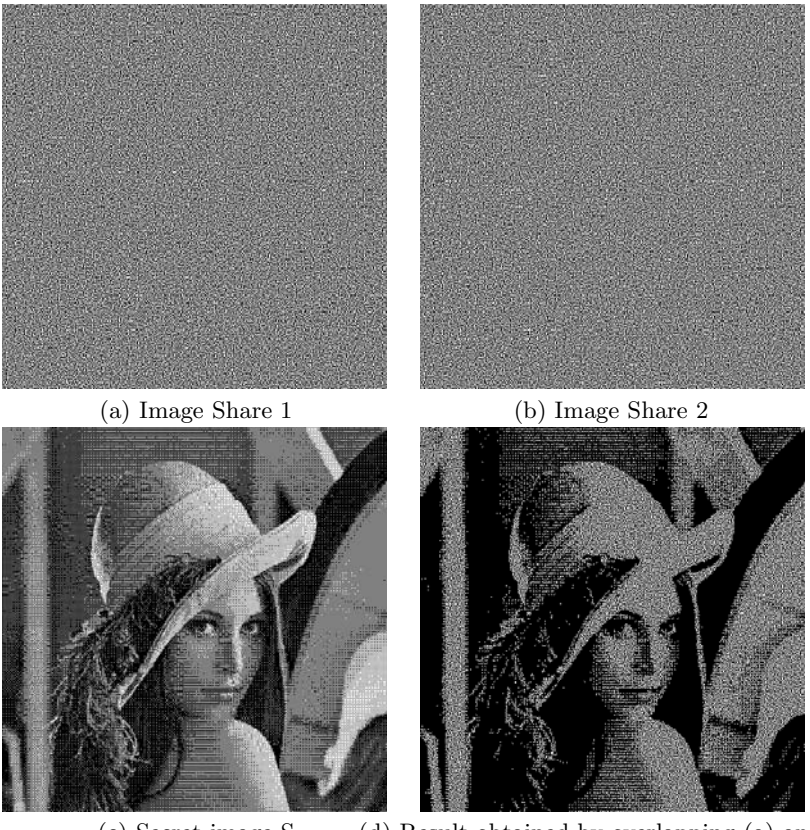
In order to deal with these problems, we propose a new method, which relies only on a pair of conjugate screens. Our scheme is composed of three parts: generation of basic screen blocks, generation of screen block group, and generation of screen images. We further employ "average dark degree" to analyze the quality of the extracted secrets.

The block diagram of our method is shown in Fig. 7 for clarification.

3.1 Basic Screen Pair Generation

To generate a pair of basic screens, we select arbitrarily from a pool of screen blocks the first screen block of size $m \times m$, denoted as S_1. Usually, the initial screen is designed with a property that larger threshold values intersect with smaller ones for consideration of good halftone image quality. This interleaving structure is helpful to generate uniformly distributed white and black pixels in a halftone image such that various gray levels can be represented to show good quality of resultant halftone images.

Fig. 3 shows an example of a selected screen block of size 4×4. We can derive the second screen S_2, which is conjugate to S_1, by arbitrarily exchanging the positions of the first $\frac{m \times m}{2}$ larger threshold values (e.g., indicated with 9,10,...,16) with the remaining ones (e.g., indicated with 1,2,...,8). This pair of screen blocks is crisscross in that the process of generating conjugate screen blocks does not incur noises that are sensitive to human eyes. An example of a screen block conjugate to Fig. 3 is shown in Fig. 8. In addition, we can generate more screen blocks by randomly combining the conjugate pair of screen blocks, as discussed in next section.



(a) Image Share 1 (b) Image Share 2

(c) Secret image S (d) Result obtained by overlapping (a) and (b)

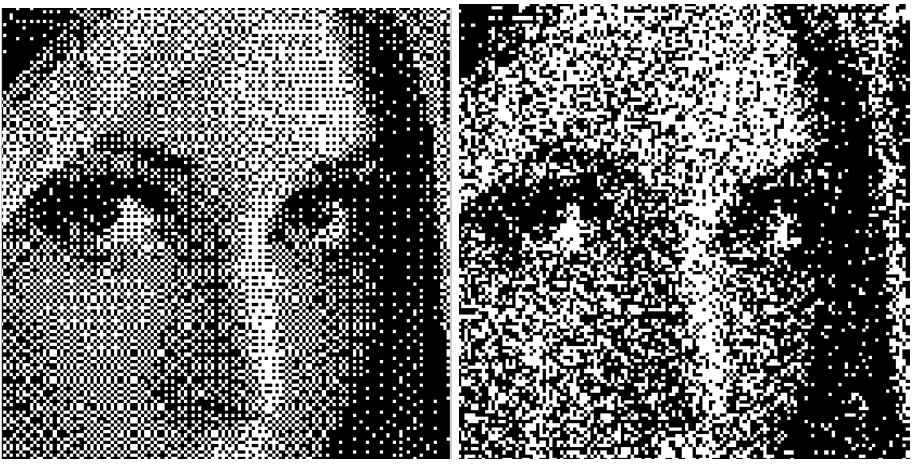
Fig. 5. An illustration of image visual cryptography

3.2 Group of Screen Blocks

If a new pair of screen blocks are generated from a pair of basic conjugate screen blocks (e.g., Fig. 3 and Fig. 8), they will possess conjugate halftone structure, too. By exploiting this property, we can randomly combine two basic screen blocks to generate extended screen blocks of size $2m \times 2m$, as shown in Fig. 9. The extended screens will be used to screen a cover image to finish secret embedding. In fact, the group of eight extended screen blocks shown in Fig. 9 will be used to generate screen images. In the group, each extended screen block (e.g., (a)) has a corresponding conjugate partner ((b)) and six half-conjugate partners ((c)∼(h)). Of course, the size of the extended screen group should be properly determined.

3.3 Secret-Dependent Screen Image Generation

Given a group of extended screen blocks and a secret message (or watermark signal) that is to be embedded into a cover image, a secret-dependent screen



(a) Result of classical screening (b) Result of stochastic screening

Fig. 6. Visual quality comparison between classical and stochastic screening results

image having the size the same with the cover halftone image will be generated. This procedure contains three steps. First, the embedded signal is divided into several message blocks, each of which has the same size with the extended screens. Second, a screen block is randomly selected and is fixedly used for message blocks belonging to bright component, as shown in Fig. 2. Third, if the message block belongs to dark areas, then an extended screen is randomly selected from the screen group. After performing the above procedure, the selected extended screens constitute a secret-dependent screen image, which can be used to generate stego halftone images via, for example, Eq. (1) for visual cryptography.

3.4 Quality Metric of an Extracted Secret

When any two halftone images (shares) are overlapped, it is important to measure whether the extracted hidden message is visually acceptable. As shown in Fig. 2, we are interested in the average dark degree of the extracted messages.

Let I_d and I_b, respectively, denote the average illumination in the dark and bright areas of an overlapped image. Let B_d denote the average dark degree of an extracted secret in an overlapped image. B_d is defined as the number of black pixels over the number of total pixels in the dark area of an overlapped image. Let B_1 and B_2, which are independent uniform random variables within the interval $[0, 1]$, denote the average dark degree in the dark area of the first and second share images, respectively. We will now analyze and compare the achievable average dark degree between random screening [7] and our screening halftoning-based method.

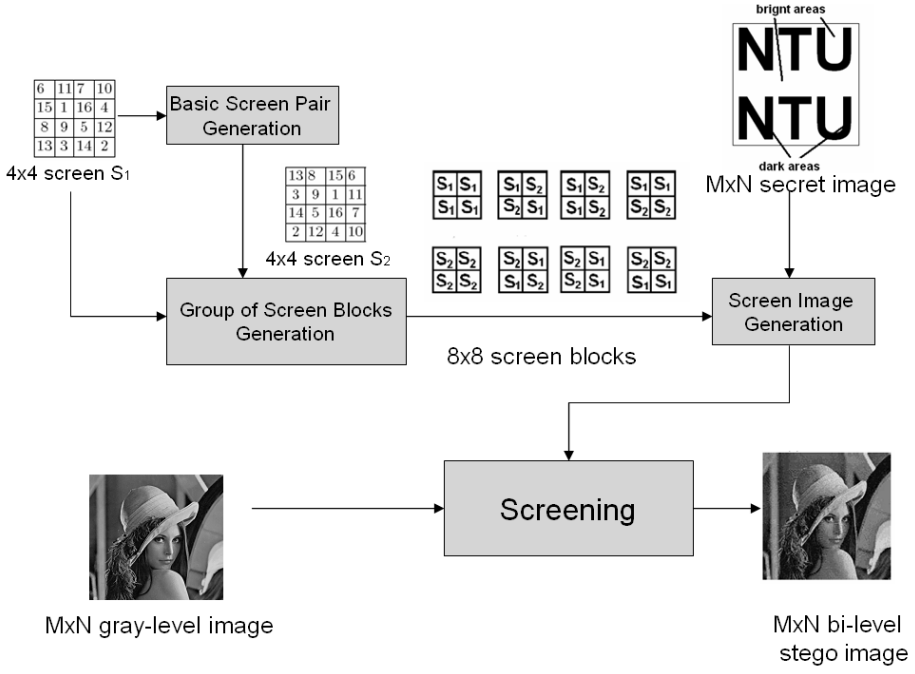


Fig. 7. Block diagram of our method

13	8	15	6
3	9	1	11
14	5	16	7
2	12	4	10

Fig. 8. Screen block conjugate to the one in Fig. 3

In the random screening process, by considering n random screens the average dark degree can be derived as:

$$B_d = \frac{n-1}{n}(1-(1-B_1)\cdot(1-B_2)) + \frac{1}{n}\max(B_1, B_2). \tag{2}$$

In Eq. (2), $(1-(1-B_1)\cdot(1-B_2))$ denotes the average dark degree when B_1 and B_2 are, respectively, generated from different screens. According to random screening, this probability is $\frac{n-1}{n}$. In addition, $max(B_1, B_2)$ represents the maximum average dark degree when B_1 and B_2 are both generated from the same screen. The probability for this situation in random screening is $\frac{1}{n}$. When n approaches infinity, the maximum B_d can be derived as:

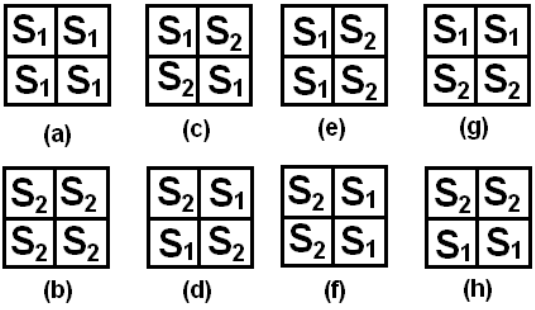


Fig. 9. A group of eight extended screens. In this group, each extended screen block has a corresponding conjugate partner ((b)) and six half-conjugate partners ((c) (h)).

$$B_d = \lim_{n\to\infty} \frac{n-1}{n}(1-(1-B_1)\cdot(1-B_2)) + \frac{1}{n}\max(B_1, B_2) = 1-(1-B_1)\cdot(1-B_2). \tag{3}$$

Moreover, the mean value of the maximum dark degree, $E(B_d)$, in random screening can be derived as:

$$\begin{aligned} E(B_d) &= E(1-(1-B_1)\cdot(1-B_2)) \\ &= \int_{b_1}\int_{b_2}(1-(1-b_1)(1-b_2))f(b_1,b_2)db_2db_1 \\ &= 1-\int_{b_1}\int_{b_2}db_2db_1 = 0.75, \end{aligned} \tag{4}$$

where $f(b_1, b_2)$ denotes the joint probability density function of B_1 and B_2, which is equal to 1 because B_1 and B_2 are independent uniform random varibles. Thus, we know that the upper bound of $E(B_d)$ in random screening is 0.75.

In our screening halftoning-based method, since each extended screen block in a group of eight extended screen blocks (Fig. 9) has one corresponding conjugate partner and six half-conjugate partners, the average dark degree of an extracted secret in an overlapped image is calculated as:

$$\begin{aligned} B_d &= \frac{1}{8}min\left(B_1+B_2,1\right) \\ &\quad +\frac{6}{8}\left(\frac{1}{2}min\left(B_1+B_2,1\right)+\frac{1}{2}max\left(B_1,B_2\right)\right) \\ &\quad +\frac{1}{8}max\left(B_1,B_2\right) \\ &= \frac{1}{2}min\left(B_1+B_2,1\right)+\frac{1}{2}max\left(B_1,B_2\right), \end{aligned} \tag{5}$$

where $min\left(B_1+B_2,1\right)$ is the minimum average dark degree when B_1 and B_2 are generated from a conjugate screen pair (which occurs with probability $\frac{1}{8}$),

$\frac{1}{2}min\,(B_1 + B_2, 1) + \frac{1}{2}max\,(B_1, B_2)$ represents the average dark degree when B_1 and B_2 are generated from a half-conjugate screen pair (which occurs with probability $\frac{6}{8}$), and $max\,(B_1, B_2)$ is the maximum average dark degree when B_1 and B_2 are generated from the same screen (which occurs with probability $\frac{1}{8}$). The mean of average dark degree achieved by means of our method can be derived as:

$$\begin{aligned} E(B_d) &= E(\frac{1}{2}\min(B_1 + B_2, 1) + \frac{1}{2}(B_1, B_2)) \\ &= \frac{1}{2}(\int_0^1 \int_0^{1-b_1} (b_1 + b_2) db_2 db_1 + \frac{1}{2} + \int_0^1 \int_0^{b_1} b_1 db_2 db_1 + \int_0^1 \int_0^{b_2} b_2 db_1 db_2) \\ &= 0.75. \end{aligned} \tag{6}$$

In this paper, we only use two basic screen blocks ($n = 2$ in our method) to generate a group of extended screen blocks to satisfy visually acceptable halftone images. As a result, we know that the mean value of B_d in our method achieves the upper bound (corresponding to $n \to \infty$) in random screening.

Since the quality of stego halftone image obtained using our method is better than that obtained using random screening, we will show later in the experimental results that our extracted secrets on the overlapped image is more clear than Knox's.

4 Experimental Results

In this section, we will demonstrate the performance of the proposed joint screening halftoning and visual cryptography scheme for image copyright protection. Our experiment was conducted using ten common 10 images of size 512×512, as shown in Fig. 10. The embedded secret (Fig. 2) is an image with size the same as the cover image. In order for performance evaluation, the average dark degree, denoted as B_d, in the dark area of an overlapping image is employed. In addition, the halftone PSNR (HTPSNR) between the cover (I) and stego (I^e) halftone images defined as

$$HTPSNR(I, I^e) = PSNR(HVS(I), HVS(I^e)), \tag{7}$$

is used for objective quality evaluation, where $HVS()$ denotes a contrast sensitivity function of the human visual system [1].

Since we have investigated to find only Knox's method [7] among the existing approaches can achieve the fact that the hidden information can be extracted by overlapping *any* two images. As a result, Knox's method is regarded as state of the art technology in this respect and is selected for the purpose of performance comparison.

Experimental results regarding halftone PSNR and average dark degree are summarized in Table 1. The results obtained from stochastic screening [7] with 2, 8, and 64 stochastic screens, respectively, were also used for comparisons.

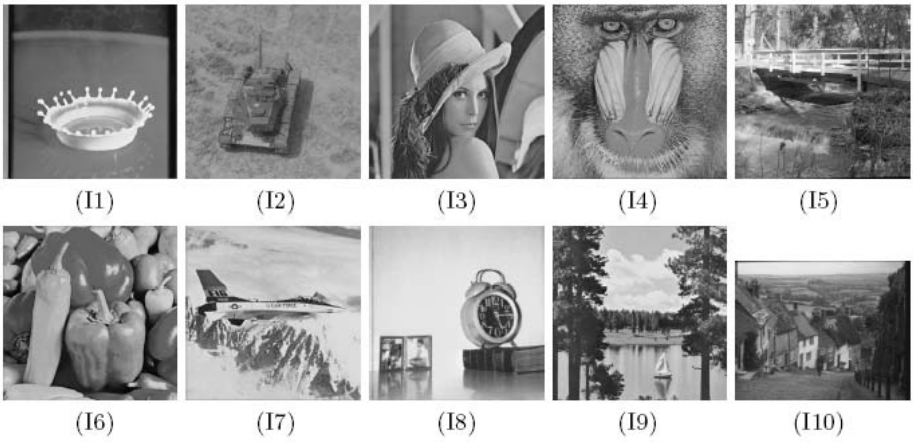
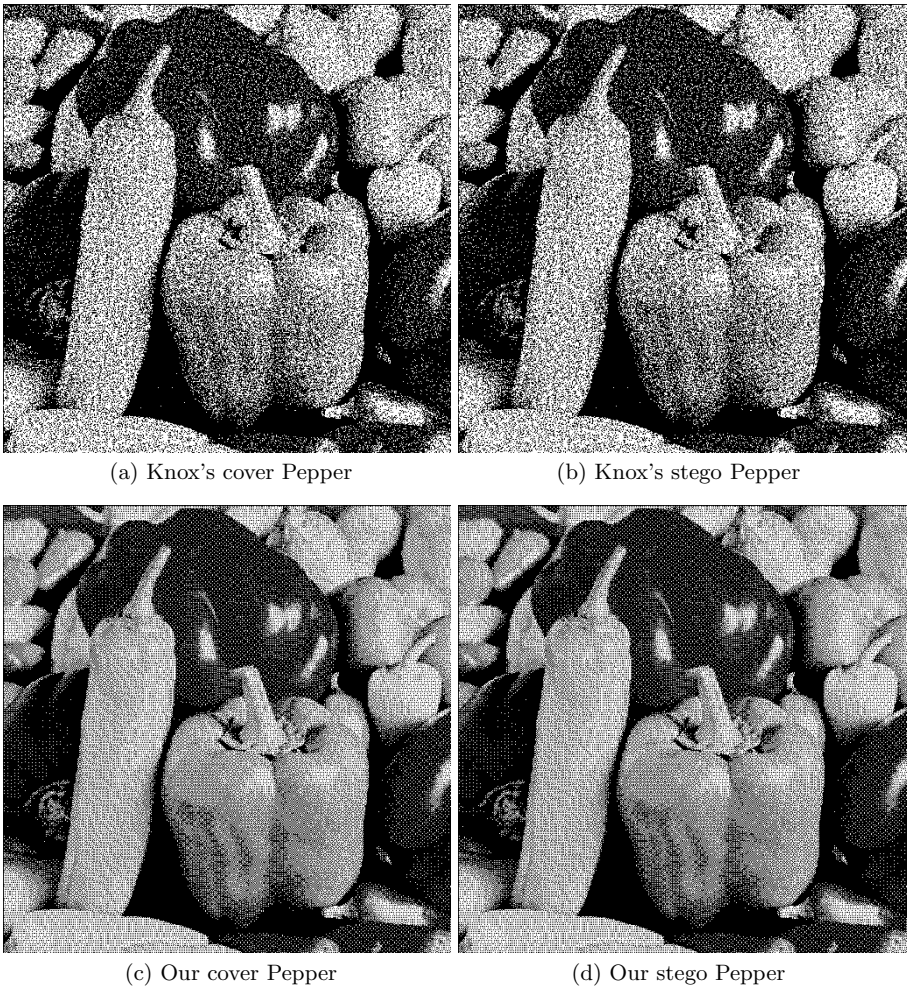


Fig. 10. Cover images

Table 1. Comparison of halftone PSNR (HTPSNR) and average dark degree between Knox's method [7] and our method

	Proposed Method		Stochastic Screen n=2		Stochastic Screen n=8		Stochastic Screen n=64	
	HTPSNR	B_d	HTPSNR	B_d	HTPSNR	B_d	HTPSNR	B_d
I1	29.91	0.85	26.78	0.82	24.48	0.85	24.27	0.86
I2	29.77	0.77	24.95	0.71	22.38	0.76	22.02	0.77
I3	29.15	0.81	25.46	0.77	23.39	0.82	23.03	0.83
I4	28.21	0.76	23.81	0.71	22.51	0.76	22.50	0.77
I5	29.01	0.76	24.96	0.71	23.05	0.76	22.96	0.77
I6	30.01	0.80	24.75	0.76	23.41	0.80	23.37	0.81
I7	29.27	0.68	23.56	0.65	23.12	0.69	23.29	0.70
I8	30.10	0.67	25.95	0.63	24.07	0.67	24.01	0.68
I9	29.63	0.78	25.30	0.74	23.84	0.78	23.41	0.79
I10	28.85	0.80	25.35	0.76	23.03	0.80	22.68	0.81

According to Table. 1, it is observed that our method achieves higher quality of stego halftone images under the constraint that the average dark degrees of extracted messages between Knox's method and ours are approximately the same. An illustration of quality comparison between the cover and stego halftone images, respectively, obtained using Knox's method and our method is shown in Fig. 11 for visual inspection. We can observe that both Knox's cover and stego images appear to be rather noisy, while the visual quality of our cover and stego images looks naturally and smoothly In addition, no perceptual differences can be perceived by comparing the cover and stego halftone images.

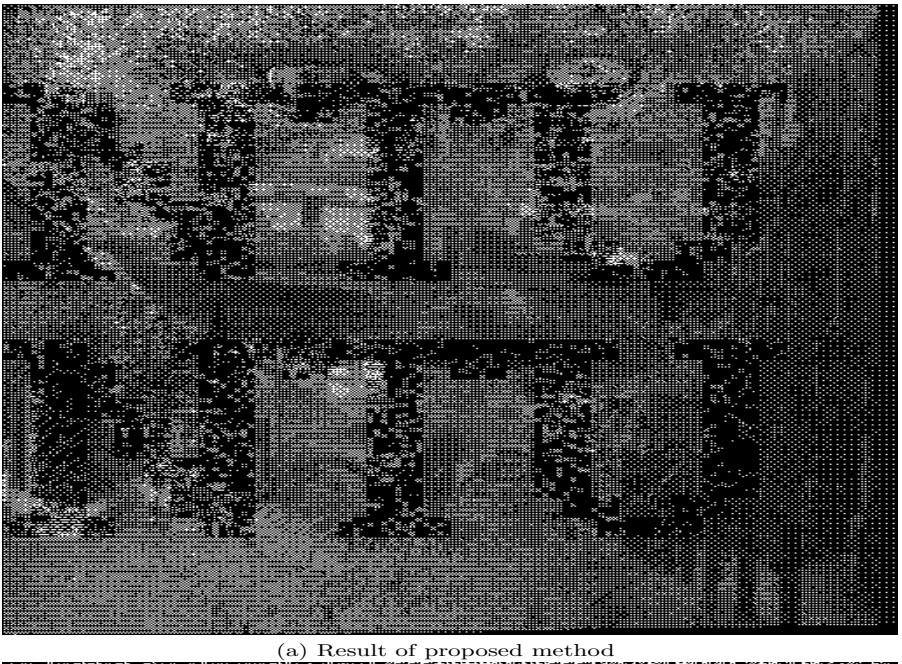


(a) Knox's cover Pepper (b) Knox's stego Pepper

(c) Our cover Pepper (d) Our stego Pepper

Fig. 11. Perceptual quality comparison between the cover and stego images obtained using Knox's method [7] and our method

In Fig. 12, we further show the extracted secret message by overlapping more than two embedded images for visual inspection subjectively. It can be observed that although both the dark degree obtained by our method and Knox's method is almost the same, our extracted secret image appears to be more clearer because knox's overlapped halftone image is rather noisy. In addition, if only two ($n = 2$) screens is used in Knox's method, then their extracted secrets become visually unclear.



(a) Result of proposed method

(b) Result of stochastic screening

Fig. 12. Comparison of secret extraction between our screening halftoning-based method and Knox's method [7]

5 Conclusion

In this paper, we study screening halftoning-based visual cryptography for image copyright protection. Our contributions contain (i) better quality of halftone images and the revealed secrets; (ii) unlimited size of image database; (iii) more than two halftone images can be overlapped to show the hidden secret; (iv) only one conjugate screen pair in our method is able to achieve the upper bound of average dark degree in random screening. The currently known methods have not achieved the above characteristics, simultaneously.

Future work will extend the current work for secret communication by studying the tradeoff between the resolution and quality of the embedded secrets.

References

1. P. J. Barten, "Physical model for the contrast sensitivity of the human eye," *in Proc. IS&T/SPIE Int. Symp. on Electronic Imaging Science and Technology*, Vol. 1666, San Jose, CA, Feb. 9-14, pp. 57-74, 1992.
2. I. J. Cox, M.L. Miller, and J.A. Bloom, Digital Watermarking, Morgan Kaufmann, 2002.
3. M. S. Fu and O. C. Au, "Hiding data in halftone image using modified data hiding error diffusion," *Proc. SPIE Conf. Visual Communication and Image Processing*, Vol. 4067, pp. 1671-1680, 2000.
4. M. S. Fu, and O.C. Au, "Data hiding in halftone images by stochastic error diffusion," *Proc. ICASSP* , Vol. 3, pp. 1965-1968, 2001.
5. M. S. Fu, and O.C. Au, "Data hiding watermarking for halftone images," *IEEE Trans. on Image Processing*, Vol. 11, pp. 477-484, 2002.
6. M.S. Fu, O.C. Au, "A novel self-conjugate halftone image watermarking technique," *Proc. of IEEE Int. Symposium on Circuits and Systems*, Vol. 3, pp. 790-793, 2003.
7. K. T. Knox, "Digital watermarking using stochastic screen patterns," U.S. Patent 5 734 752, September 1996.
8. D. E. Knuth, Digital halftones by dot diffusion, *ACM Trans. On Graphics*, Vol. 6, No. 4, pp. 245-273, 1987.
9. D. L. Lau, and G. R. Arce, Modern digital halftoning, Marcel Dekker, 2001.
10. M. Noar and A. Shamir, "Visual Cryptography," *Advances in Cryptography Eurocrypt'94*, Lecture Notes in Computer Science, Springer-Verlag, Vol. 950, pp. 1-12, 1995.
11. S. C. Pei, and J. M. Guo, "Hybrid pixel-based data hiding and block-based watermarking for error-diffused halftone images," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 13, pp. 867-884, 2003.
12. P. W. Wong, and N. D. Memon, "Image processing for halftones," *IEEE Signal Processing Magazine*, Vol. 20, pp. 59-70, 2003.

Robust Audio Watermarking Based on Low-Order Zernike Moments

Shijun Xiang[1,2], Jiwu Huang[1,2],
Rui Yang[1,2], Chuntao Wang[1,2], and Hongmei Liu[1,2]

[1] School of Information Science and Technology,
Sun Yat-sen University, Guangzhou 510275, China
[2] Guangdong Key Laboratory of Information Security Technology,
Guangzhou 510275, China
isshjw@mail.sysu.edu.cn

Abstract. Extensive testing shows that the audio Zernike moments in lower orders are very robust to common signal processing operations, such as MP3 compression, low-pass filtering, etc. Based on the observations, in this paper, a robust watermark scheme is proposed by embedding the bits into the low-order moments. By analyzing and deducting the linear relationship between the audio amplitude and moments, watermarking the low-order moments is achieved in time domain by scaling sample values directly. Thus, the degradation in audio reconstruction from a limited number of watermarked Zernike moments is avoided. Experimental works show that the proposed algorithm achieves strong robustness performance due to the superiorities of exploited low-order moments.

1 Introduction

Due to RST (Rotation, Scale, Translation) invariance of image Zernike moments [1], Zernike transform is widely applied in some image processing fields, such as image recognition [2], robust image watermarking [3][4][5][6], and image authentication [7]. In [2], the authors introduced the RST invariance of image Zernike moments for image recognition, and pointed out the low order moments represent image shape while the higher order ones fill the high frequency details. In [8], the authors analyzed the reconstruction power of image Zernike moments and the reasons of the reconstruction degradation by using a limited number of Zernike moments. In the applications using Zernike moments, how to regenerate the signal from the moments is an important issue. In [3], the watermarked image was degraded in the reconstruction procedure due to the high-frequency details in higher order moments is lost. By converting the watermark, a vector composed by some selected moments, into the spatial domain signal in [4], or by remaining and adding higher order information before the watermark is embedded in [5], the watermarked image avoided the degradation caused by the reconstruction. The above methods share a idea that the image Zernike moments are robust to geometric attacks.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 226–240, 2006.

Naturally, we are motivated in a way that the application of Zernike moments on audio watermarking is beneficial? Digital audio, one-dimensional (1-D) discrete signal, may be mapped into two-dimensional (2-D) form for performing Zernike transform. In this way, it is possible to discover the characteristics of Zernike moments on audio signal processing. According to the best of our knowledge, there has no any relative report on Zernike transform in audio applications. Possibly, it is due to the following some reasons: 1) Audio is 1-D signal; 2) Audio Zernike transform may introduce a series of unknown problems, such as synchronization and reconstruction; 3) Compared with image Zernike moments, the characteristics of Zernike moments on audio are sealed yet, which are required to be opened.

In this paper, audio Zernike transform is performed by mapping audio into 2-D form. Furthermore, the features of audio moments are investigated by using extensive experimental works. It is noted that, 1) the reconstruction degradation from moments is unavoidable and the quality of regenerated audio is unacceptable; 2) the low-order moments capture the basic shape of audio signal and represent its low-frequency components. As a result, the low-order moments are very robust to common signal processing operations, such as MP3 compression and low-pass filtering, etc. By using the advantages of the low-order moments, a robust multi-bit audio watermarking algorithm is proposed. In order to avoid the degradation in the reconstruction procedure, we analyze and deduct the linear relationship between the audio amplitude and its moments in proposed strategy. Watermarking the audio Zernike moments in lower orders is achieved by scaling the sample values, and thus the degradation in the reconstruction procedure is avoided. The watermarked audio is imperceptible. Simulation results show that the proposed algorithm is very robust to common signal processing operations and attacks in Stirmark Benchmark for Audio.

In the next section, we introduce the theory of Zernike transform. We then investigate the characteristics of audio Zernike moments via extensive testing. This is followed by a description of a general framework for our proposed watermark embedding and detecting strategy. We then analyze the watermark performance and test the watermark robustness on some common signal processing and most attacks in Stirmark Benchmark for Audio. Finally, we draw the conclusions.

2 Zernike Moments

In this section, we describe Zernike moments and their properties. Some of the materials in the following are based on [1][2][4]. Zernike introduced a set of complex polynomials which form a complete orthogonal set over the interior of the unit circle, i.e., $x^2 + y^2 = 1$. Let the set of these polynomials be denoted by $V_{nm}(x, y)$. The form of these polynomials is:

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \cdot exp(jm\theta) \tag{1}$$

where n is positive integer or zero, m an integers subject to constraints $(n-|m|)$ is nonnegative and even, ρ is the length of vector from origin to (x, y) pixel, θ is the angle between vector ρ and x axis in counterclockwise. $R_{nm}(\rho)$ is radial polynomial, defined as:

$$R_{nm}(\rho) = \sum_{s}^{n-\frac{|m|}{2}} (-1)^s \cdot \frac{(n-s)!}{s! \cdot (\frac{n+|m|}{2}+s)! \cdot (\frac{n-|m|}{2}+s)!} \cdot \rho^{n-2s} \tag{2}$$

Note that $R_{n,m}(\rho) = R_{n,-m}(\rho)$. So $V_{n,-m}(\rho,\theta) = V^*_{n,m}(\rho,\theta)$. These polynomials are orthogonal and satisfy $\iint\limits_{x^2+y^2\le 1} V^*_{n,m}(x,y) \cdot V_{p,q}(x,y)\,\mathrm{d}x\mathrm{d}y = \frac{\pi}{n+1}\delta_{np}\delta_{mq}$, with $\delta_{np} = \begin{cases} 1 & n = p \\ 0 & n \neq p \end{cases}$. Zernike moments are the projection of the function onto these orthogonal basis functions. The Zernike moment of order n with repetition m for a continuous 2-D function $f(x, y)$ that vanishes outside the unit circle is

$$A_{nm} = \frac{n+1}{\pi} \iint\limits_{x^2+y^2\le 1} f(x,y) \cdot V^*_{n,m}(x,y)\,\mathrm{d}x\mathrm{d}y \tag{3}$$

For a 2-D digital signal, like digital image, the integrals are replaced by summations to

$$A_{nm} = \frac{n+1}{\pi} \sum_{n=0}^{+\infty} \sum_{m} f(x,y) \cdot V^*_{n,m}(x,y) \tag{4}$$

Suppose that one knows all moments A_{nm} of $f(x, y)$ up to given order N_{max}. It is desired to reconstruct a discrete function, $\hat{f}(x, y)$ by the following formula

$$\hat{f}(x,y) = \sum_{n=0}^{N_{max}} \sum_{m} A_{nm} \cdot V_{nm}(\rho,\theta) \tag{5}$$

Theoretically, as N_{max} increasing, $\hat{f}(x, y)$ goes to $f(x, y)$.

3 Audio Zernike Moments

In this section, audio Zernike transform is achieved by mapping 1-D audio signal into 2-D form. Then, the characteristics of audio Zernike moments are investigated based on extensive testing. It is found that the low-order moments are robust to common audio signal processing. And, the reconstruction degradation from moments is unavoidable and distorted severely.

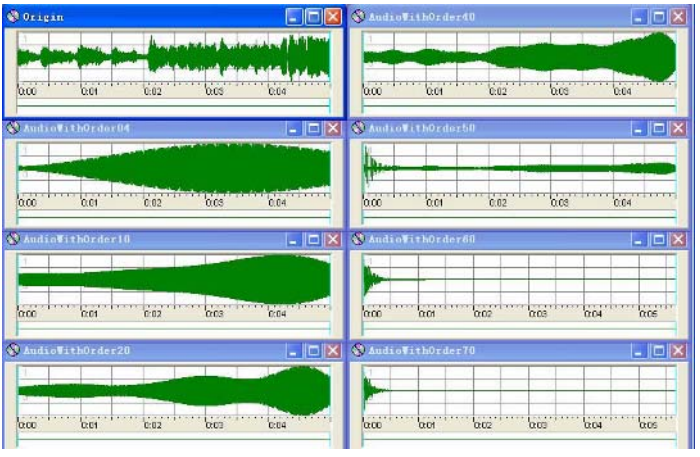


Fig. 1. The original audio and the reconstructed audio under the different order N_{max}

3.1 Mapping

A 1-D digital audio signal, may be mapped into a 2-D form by using the following projection:

$$\begin{cases} L = R \times R + M \\ f(x, y) = g(x \cdot R + y) \end{cases} \tag{6}$$

where $f(x, y)$ is corresponding audio version after projection, L is the length of audio, M is the rest of audio samples, and R is the width or height in $f(x, y)$, the value of which is as large as possible under the constraint of Equation (6).

3.2 Reconstruction Degradation

After mapping, Zernike decomposition and reconstruction procedures on audio signal are performed, referred to Equation (4) and (5). We choose a clip from our test data set, flute music denoted as *flute.wav* (16-bit signed mono audio file sampled at 44.1 kHz with the length of 5s), for testing. The number of the given max order N_{max} is assigned to 4, 10, 20, 30, 40, 45, 50, 60 and 70, respectively. The waveforms of original one and the reconstructed audios are aligned in Fig.1. As to other kinds of audio, such as pop music, piano music and speech, etc., the simulation results are similar.

In Fig.1, *Origin.wav* is the original audio while *AudioWithOrder**.wav denote the reconstructed ones, in which N_{max} is assigned as $'*'$. It is noted that the low-order moments captured the basic shape of audio signal while the higher order ones fill the high frequency details. This observation is similar to that in images [2]. In detail, when N_{max} is less than 50, the bigger N_{max} is used, the closer to the original audio the reconstructed audio is. When N_{max} is greater than 50, the

reconstructed audio is distorted seriously. The degradation caused in reconstruction procedure is due to that when N_{max} is lower the high frequency information is discarded, while N_{max} is higher the cumulative computation error occurs in the reconstruction [8]. Referred to Fig.1, it is evident that the reconstruction degradation from limited moments is unavoidable.

3.3 Selection of Robust Features

In order to apply Zernike moments in audio watermarking, it is necessary to investigate the robustness performance of audio Zernike moments to common signal processing manipulations, such as MP3 compression, low-pass filtering, etc. In the following experimental works, we design the following mathematical expression to compute the modification of moments before and after audio processed,

$$E_{bn} = \sum_{m} |A_{nm}|, \quad E_{an} = \sum_{m} |A'_{nm}| \tag{7}$$

where A'_{nm} is the corresponding version of A_{nm} after undergoing some signal processing operations. E_{bn} and E_{an} denote the total amplitude of all moments

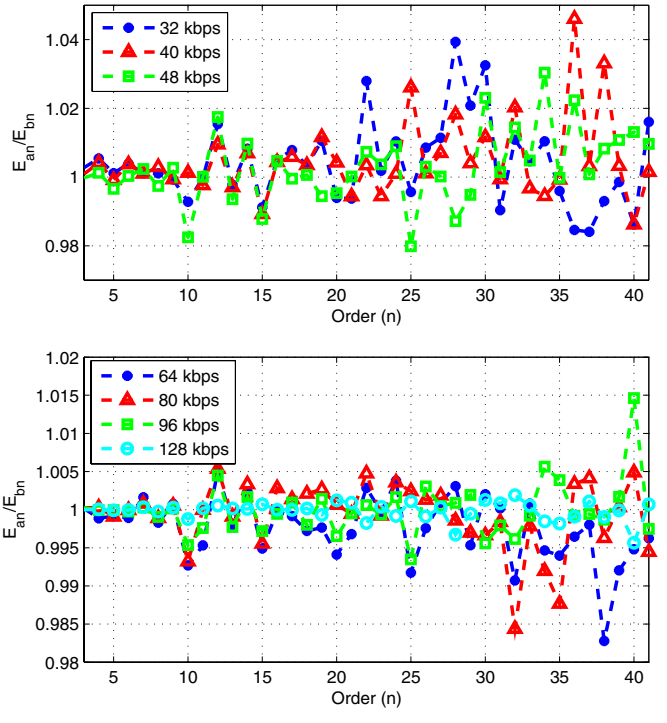


Fig. 2. The effects of MP3 compression with the bit rates of 32, 40, 48, 64, 80, 96 and 128 kbps

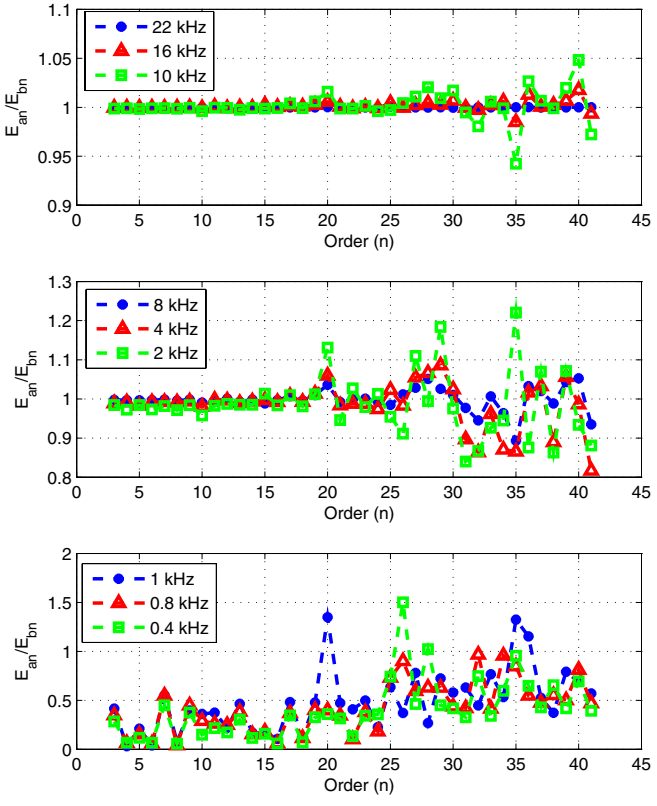


Fig. 3. The effects of low-pass filtering with the cut-off frequency of 0.4, 0.8, 1, 2, 4, 8, 10, 16 and 22 kHz

with the given order n before and after processed, respectively. And, $0 \leq n \leq N_{max}$.

We select *flute.wav* as the example clip to test the effect of MP3 compression, and low-pass filtering. Fig.2 and Fig.3 have the same scaling in both horizontal (given order n) and vertical (E_{an}/E_{bn}) axis. As to other kinds of audio, such as pop music, piano music and speech, etc., the simulation results are similar.

In above experiments, MP3 compression and low-pass filtering operations are performed by using the software CoolEditPro v2.1. Based on the extensive testing with different audio signals, we have the following observations:

i. Zernike transform of 1-D signal may be achieved by mapping the signal into 2-D form. It is noted that the low-order moments capture the basic shape of the signal but the reconstruction degradation from Zernike moments is large and unavoidable, referred to Fig.1.

ii. Based on the extensive experiments, it is also found that the low-order moments are robust to common signal processing operations. The moments under order 10 is very robust to MP3 compression even with the lowest bit rate of 32 kbps, referred to Fig.2. The moments under order 20 is robust to low-pass filtering up to with cut-off frequency of 2 kHz, referred to Fig.3.

As a conclusion, if we embed the watermark into those moments under order 10 and try to avoid the degradation in reconstruction procedure, it is expected that the watermark will be very robust to these common signal processing manipulations and some hostile attacks.

4 Proposed Watermark Algorithm

In this section, a robust audio watermark algorithm is presented. The watermark bits are embedded into the low-order moments to achieve good robustness. We deduct the linear relation between audio amplitude and its moments. In the proposed watermark scheme, by applying the linear relation we watermark the low-order moments by scaling audio amplitude in time domain directly, and thus the generated watermarked audio avoids the reconstruction distortion. To resist amplitude scaling attack, the use of three successive segments as a group is designed to embed one bit of information by modifying the low-order moments in each three segments.

4.1 Watermark Embedding

Embedding Scheme: The basic idea of the embedding algorithm is to split the original audio to many segments, three segments as a group. Mapping the segments into 2-D form and performing Zernike transform. Then embed one bit of watermark into the low-order Zernike moments. According to the difference of the moments before and after watermarking, a corresponding scaling factor is computed. Finally, the watermark audio is generated by scaling original one. The embedding model is shown in Fig.4.

In the algorithm, the adaptive embedding strategy is introduced to control the embedding strength, achieving the value as large as possible under the imperceptivity constraint. The detail is described as below. Suppose that SNR_1 is the SNR of the watermarked audio versus the original one, SNR_0 is a predefined value. If $SNR_1 < SNR_0$, the embedding strength factor d will be automatically modified until $SNR_1 \geq SNR_0$. The watermarked audio becomes more similar to original one as d decreasing.

It is noted that utilizing the relationships among different audio sample sections to embed data is proposed in [9]. This strategy is one type of modified patchwork scheme [10]. However, what proposed in this paper is different from [9]. Instead of in the time domain, we embed watermark signal in the low-order Zernike moments in order to achieve better robustness performance.

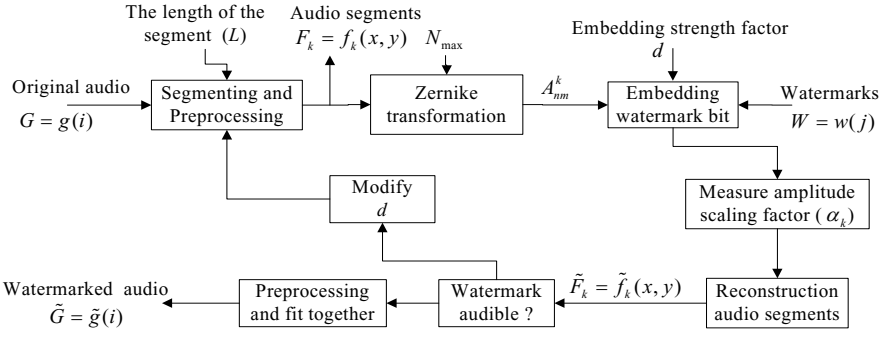


Fig. 4. The watermark embedding scheme

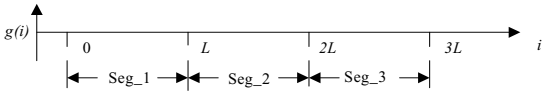


Fig. 5. Three consecutive sample segments

Embedding Strategy: The original audio, $g(x)$, is split into proper segments. Suppose each segment includes L samples, as shown in Fig.5. Generally, L is designed according to the embedding capacity and SNR of the watermarked audio. After mapped into 2-D form, $f_k(x,y)$, by using Equation (6), Zernike transform is performed on each segment with a given order n. The total modulus of the moments in the k^{th} segment is denoted by E_k, as shown in Equation (8). n is suggested lower than 10 to achieve good robustness.

$$E_k = \sum_m |A_{nm}|, \quad n < N_{max} \tag{8}$$

Denote the total modulus of the n order moments in the *three* consecutive segments as E_{k-1}, E_k and E_{k+1}, respectively. Their relations may be obtained from the following Equation,

$$\begin{cases} A = E_{max} - E_{med} \\ B = E_{med} - E_{min} \end{cases} \tag{9}$$

where A and B stand for the differences, respectively. And, $E_{max} = maximum(E_{k-1}, E_k, E_{k+1})$, $E_{med} = meddium(E_{k-1}, E_k, E_{k+1})$ and $E_{min} = minimum(E_{k-1}, E_k, E_{k+1})$. So we exploit Equation (10) to embed one watermark bit $w(i)$,

$$\begin{cases} A - B \geq S & \text{if } w(i) = 1 \\ B - A \geq S & \text{if } w(i) = 0 \end{cases} \tag{10}$$

where $S = d \cdot (E_{k-1} + E_k + E_{k+1})$ is the embedding strength.

Assumed that after embedding one watermark bit, E_{k-1}, E_k and E_{k+1} go to $\tilde{E}_{k-1}$, $\tilde{E}_k$ and $\tilde{E}_{k+1}$, respectively. It is equivalent to E_{k-1}, E_k and E_{k+1} by the corresponding factor α_{k-1}, α_k and α_{k+1}, which may be computed by the following expressions,

$$\alpha_{k-1} = \frac{\tilde{E}_{k-1}}{E_{k-1}}, \; \alpha_k = \frac{\tilde{E}_k}{E_k} \text{ and } \alpha_{k+1} = \frac{\tilde{E}_{k+1}}{E_{k+1}} \tag{11}$$

According to Equation (8), E_k is linear to A_{nm}^k. It means that the corresponding moments $\tilde{A}_{nm}^k$ after watermarking may be expressed as

$$\tilde{A}_{nm}^k = A_{nm}^k \cdot \alpha_k \tag{12}$$

According to the analysis in Section 3.2, the serious reconstruction degradation will be caused if the watermarked signal is regenerated from the modified moments $\tilde{A}_{nm}^k$. Thus, It is required to introduce a new strategy to reconstruct the watermarked audio.

4.2 The Reconstruction Strategy

Now, we focus on aiming at resolving the reconstruction degradation. Consider amplitude linear scaling of the signal $f(x, y)$ through a factor α. Assumed that the scaled signal and moments are denoted by $\tilde{f}(x, y)$ and $\tilde{A}_{nm}$, respectively. We have the following expression,

$$\begin{aligned} \tilde{A}_{nm} &= \frac{n+1}{\pi} \sum_{n=0}^{+\infty} \sum_{m} \tilde{f}(x, y) \cdot V_{nm}^*(x, y) \\ &= \frac{n+1}{\pi} \sum_{n=0}^{+\infty} \sum_{m} \alpha \cdot f(x, y) \cdot V_{nm}^*(x, y) \\ &= \alpha \cdot A_{nm} \end{aligned} \tag{13}$$

From Equation (13), it is noted that the relation between audio sample values and the moments is mathematically linear. The linear relation has been verified by extensive testing. This conclusion is very useful. It means that the modification of Zernike moments may be mapped as the operation of scaling audio amplitude. Using this conclusion, we introduce the following strategy to generate the watermarked audio by scaling the sample values in each segment, referred to Equation (14).

$$\tilde{f}_k(x, y) = \alpha_k \cdot f_k(x, y) \tag{14}$$

where α_k is the amplitude scaling factor of the k^{th} audio segment, computed by using Equation (11), $f_k(x, y)$ and $\tilde{f}_k(x, y)$ denote the k^{th} segment of the original 2-D signal and the watermarked 2-D signal, respectively.

The process is repeated to embed the watermark bits. Finally, by using Equation (6) we obtain the reconstructed watermarked audio, $\tilde{g}(x)$.

How to reduce the reconstruction degradation from Zernike moments is an important issue in watermark applications [4]. In the proposed strategy, by using the linear relation between the audio and its moments, the degradation can be avoided. The watermark is designed to embed into the moments under 10 orders to achieve strong robustness. Additionally, since the watermarked audio is reconstructed by scaling sample values, the computation cost is low.

4.3 Watermark Extraction

In the detection, the watermarked audio, which has undergone some signal processing operations, for example, MP3 compression, low-pass, is performed Zernike transform as in the watermark embedding. Considering the synchronization attacks such as cropping, data structure of hidden bits, as shown in Fig.6, is designed. The synchronization codes [9][11] are introduced to locate the embedding region of watermark bits. According to the requirement, we may embed one or many synchronization codes.

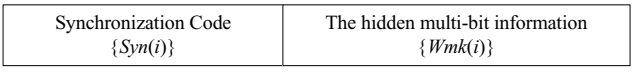


Fig. 6. Data structure of hidden bit stream [11]

As in Equation (9), we compute $\hat{E}_{k-1}$, $\hat{E}_k$ and $\hat{E}_{k+1}$, which are ordered to obtain $\hat{E}_{min}$, $\hat{E}_{med}$ and $\hat{E}_{max}$. Similar to Equation (10), we have

$$\begin{cases} \hat{A} = \hat{E}_{max} - \hat{E}_{med} \\ \hat{B} = \hat{E}_{med} - \hat{E}_{min} \end{cases} \tag{15}$$

Comparing $\hat{A}$ and $\hat{B}$, we get the hidden bit by using the following rule,

$$\hat{w}(i) = \begin{cases} 1 & \text{if } \hat{A} - \hat{B} \geq 0 \\ 0 & \text{if } \hat{A} - \hat{B} < 0 \end{cases} \tag{16}$$

The process is repeated until all hidden bits are extracted. The parameters, the length of segment L, the given order n and the synchronization sequence $Syn(i)$, are beforehand known, so the detection process is blind.

5 Performance Analysis

In this section, we evaluate the performance of the proposed algorithm in terms of data embedding capacity, resisting amplitude modification attack, and error probability of synchronization codes and watermarks.

The embedding capacity, denoted by C, refers to the number of bits that are embedded into the audio signal within a unit of time. Suppose that the sampling rate of audio is R (Hz). In our algorithm, $C = R/(3 \cdot L)$ (bps).

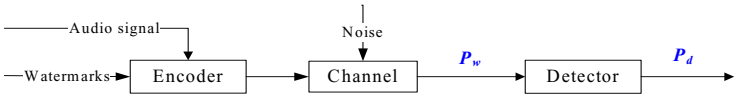


Fig. 7. The watermark bit error probability in the channel and detector

In this paper, we embed one watermark bit into the relative relation in each three audio segments. Our goal is to resist amplitude scaling. Referred to Equation (8) and (13), whether amplitude scaling attack occurs or not, the magnitude relation between A and B keeps unchanged. It means that the algorithm is immune to such attack.

5.1 Error Analysis on Synchronization Code Searching

There are two types of errors in searching synchronization codes, false positive error and false negative error. A false positive error occurs when a synchronization code is supposed to be detected in the location where no synchronization code is embedded, while a false negative error occurs when an existing synchronization code is missed. Once a false positive error occurs, the bits after the locations of the false synchronization code will be regarded as the watermark bits. When a false negative error takes place, some watermark bits will be lost. The false positive error probability of the synchronization code P_1 can be calculated as follows,

$$P_1 = \frac{1}{2^{N_1}} \cdot \sum_{k=0}^{T} C_{N_1}^k \tag{17}$$

where N_1 is the length of a synchronization code, and T is the threshold used to judge the existence of synchronization code.

Generally, we use the following formulation to evaluate the false negative error probability P_2 of the synchronization code according to the bit error probability, denoted as P_d, in the detector.

$$P_2 = \sum_{k=T+1}^{N_1} C_{N_1}^k \cdot (P_d)^k \cdot (1 - P_d)^{(N_1 - k)} \tag{18}$$

5.2 Error Analysis on Watermark Extraction

It is noted that the introduction of synchronization codes in the algorithm may make the difference between the bit error probability of the watermark in the detector P_d and in the channel P_w, illustrated in Fig.7.

Suppose that x, the number of synchronization codes, are embedded and the number of the false positive synchronization codes and false negative synchronization codes detected is y and z, respectively. So the error probability P_w may

be expressed as follows. The false positive error probability P_1 can be expressed as $y/(x+y-z)$ here.

$$P_w = \frac{(x-z)\cdot N_2 \cdot P_{sw} + y \cdot N_2 \cdot P_{aw}}{(x+y-z)\cdot N_2} = (1-P_1)\cdot P_{sw} + P_1 \cdot P_{aw} \tag{19}$$

where, N_2 is the length of the watermark bits, which follow a corresponding synchronization code, P_{sw} and P_{aw} denote the error probability of the watermarks in case of false negative and positive synchronization code occurring. From the view of point in probability theory, the value of P_{sw} and P_{aw} is approximately P_d and 50%. Accordingly, we have the following formulation.

$$P_w = (1-P_1)\cdot P_{sw} + P_1 \cdot P_{aw} \approx (1-P_1)\cdot P_d + P_1 \cdot 50\% \tag{20}$$

From Equation (20), it is noted that the bit error probability of the watermark in the channel is different from that in the detector after introducing synchronization code, and the difference mainly relies on the number of the false positive synchronization code. The occurring of the false negative synchronization code will lead to the loss of some hidden information bits, the effect of which on the error probability of the watermark may be ignored. When the value of y go to ZERO, P_1 goes to ZERO, thus P_w goes to P_d.

6 Experimental Results

The proposed algorithm is applied to a set of audio signals including pop, light, rock, piano, drum and electronic organ. $N_{max} = 10$ and the moments in order 8 is watermarked to achieve good robustness. The length of segments L = 225 is mapped into 15×15 2-D form. A clip (20s, mono, 16 bits/sample, 44.1 kHz and WAVE format) from the light music titled *'Danube'* is the example audio watermarked with 13 repeated 100 bits of binary sequence composed of a 31-bit synchronization code and the 69-bit watermark, with the embedding factor $d = 0.25$. The SNR is 25.63 dB beyond the 20 dB requested by the IFPI, with the ODG (Objective Difference Grade) of -3.60 implemented by EAQUAL 0.1.3 alpha [12][13][14] considered HAS (Human Auditory System). The subjective listening test shows the watermarked audio is perceptibly very similar to original one. It is an evidence that the proposed watermark strategy has removed the reconstruction degradation caused by limited order moments. It is owed to that the watermarked signal is regenerated by amplitude scaling operation in time domain.

We test the robustness of the proposed algorithm with BER (Bit Error Rate). The audio editing and attacking tools adopted in our experiments are CoolEditPro v2.1, GoldWave v4.25 and Stirmark Benchmark for Audio v0.2 [15][16]. The test results under common audio signal processing, cropping, and attacks in Stirmark for Audio are listed in Tables 1-3.

From Table 1 we can see that our algorithm is robust enough to some common audio signal processing manipulations, such as, MP3 compression of 32 kbps,

Table 1. Robustness Performance to Common Attacks

Attack Type	**BER(%)**	**Attack Type**	**BER(%)**
Requantization 16 → 32 → 16(bit)	0	Resample 44.1 → 16 → 44.1(kHz)	0
MP3 (32 kbps)	1.15	MP3 (40∼128 kbps)	0
Low pass (9 kHz)	0	Low pass (8 kHz)	3.61
Low pass (6 kHz)	7.46	Low pass (4 kHz)	8.46
Low pass (3 kHz)	9.31	Volume (50∼150%)	0

Table 2. Robustness Performance to cropping attacks

Attack Type	**BER(%)**	**Attack Type**	**BER(%)**
Cropping (1s)	0	Cropping (2s)	0
Cropping (3s)	0	Cropping (4s)	0
Cropping (5s)	0	Cropping (6s)	0

Table 3. Robustness performance to the attacks in StirMark Benchmark for Audio v0.2

Attack Type	**BER(%)**	**Attack Type**	**BER(%)**
AddBrumm_100	0	AddNoise_500	0
AddBrumm_1100	6.23	AddNoise_700	1.23
AddBrumm_2100	18.15	AddNoise_900	2.22
Compressor	0	ExtraStereo_30	0
Amplify	0	ExtraStereo_50	0
Exchange	0	ExtraStereo_70	0
ZeroCross	5	Normalize	0
Stat1	0	Stat2	0
Nothing	0	Smooth	0
Original	0	Smooth2	0
Invert	0	RC_LowPass	0
ZeroLength	0	Lsbzero	0
AddSinus	15.07	ZeroCross	0
AddDynNoise	0	ZeroRemove	0
FFT_Invert	0	FFT_RealReverse	0
Echo	6.84	FlippSample	5.61
FFT_HLPass	6.69	RC_HighPass	6.23
CutSample	Failed	CopySample	Failed
FFT_Stat1	Failed	AddFFTNoise	Failed
FFT_Test	Failed	VoiceRemove	Failed

low pass of 3 kHz, etc. It is owed to that the watermark bits are embedded into Zernike moments in lower orders which have been verified robust to common signal processing.

Table 2 shows the strong robustness to cropping with the threshold $T = 3$, referred to Equation (17). In our experiments, by randomly cropping one portion of the audio even with the length of 6s, it is noted that a portion of watermark, 4

frame in 13, is lost, but the remanent watermark, 9 frame, is still extracted at a low bit error rate. The reason is that the displacement of sample positions in the embedding and extracting is tracked by resynchronization via synchronization codes.

Stirmark Benchmark for Audio is a common robustness evaluation tool for audio watermarking techniques. All listed operations are performed by using default parameters implemented in the system. From Table 3, it is found that the watermark shows stronger resistance to those common attacks. In the cases of failure ($'Failed'$ means the BER is over 20%), the audio quality is distorted largely.

7 Conclusions

In this paper, we propose a multi-bit audio watermarking method based on the robustness of the low-order Zernike moments.

Via extensive experiments, we show the advantages of the proposed features, the merits of the low-order moments. The moments in lower orders are very robust to common signal processing, such as MP3 compression. Accordingly, by applying the investigated feature combined with synchronization match technique, a robust audio watermarking scheme is designed. By using the linear relation between the audio and its moments, the low-order moments are watermarked by scaling audio sample values directly. As a result, the generated watermarked audio has avoided the reconstruction distortion, and the watermark is imperceptible. Finally, the performance of the proposed algorithm is investigated.

The extensive experimental works have shown that the proposed watermark strategy has strong robustness to common signal processing and most attacks in Stirmark Benchmark for Audio. The watermark also achieves good robustness against cropping.

The DA/AD conversion (a common signal processing operation) [17] and the TSM (Time-scale Modification) attacks [18] are still challenging issues in audio watermarking community. One consideration of the further work is to improve the robustness of the watermark to the two attacks according to their distortion models [17][18]. Additionally, more detail evaluation based on the actual benchmark [19] will be reported in future researches.

Acknowledgments

Authors appreciate the support by NSFC (60325208,90604008), 973 Program (2006CB303104), NSF of Guangdong (04205407). We also thank the anonymous reviewers for their constructive suggestions.

References

1. Cho-Huak Teh and Roland T. Chin: On Image Analysis by the Methods of Moments. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.10 (1988) 496-513

2. Khotanzad and Y. H. Hong: Invariant Image Recognition by Zernike Moments. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.12, (1990) 489-497
3. M. Farzam and S. Shahram Shirani: A Robust Multimedia Watermarking Technique Using Zernike Transform. Proc. of IEEE International Workshop Multimedia Signal Processing, (2001) 529-534
4. H. S. Kim and H. K. Lee: Invariant Image Watermark Using Zernike Moments. IEEE Transaction on Circuits and Systems for Video Technology, Vol.13, No.8, (2003) 766-775
5. Y. Q. Xin, Simon Liao and Miroslaw Pawlak: A Multibit Geometrically Robust Image Watermark Based on Zernike Moments. Proc. of the 17th International Conference on Pattern Recognition, (2004) 861-864
6. J. Chen, H. X. Yao, W. G. and S. H. Liu: A Robust Watermarking Method Based on Wavelet and Zernike Transform. Proc. of the 2004 International Symposium on Circuits and Systems, Vol.2 (2004) 23-26
7. H. M. Liu, J. L Lin and J. W. Huang: Image Authentication Using Content Based Watermark. Proc. of the 2004 International Symposium on Circuits and Systems, (2005) 4014-4017
8. S. X. Liao and M. Pawlak: On the Accuracy of Zernike Moments for Image Analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.20 (1998) 1358C1364
9. W. N. Lie and L.C. Chang: Robust and High-Quality Time-Domain Audio Watermarking Subject to Psychoacoustic Masking. Proc. of IEEE International Symposium on Circuits and Systems, Vol.2 (2002) 45-48
10. I. K. Yeo and H. J. Kim. Modified patchwork algorithm: A Novel Audio Watermarking Scheme. IEEE Transaction on Speech and Audio Processing, Vol.11 (2003) 381-386
11. S. Q Wu, J. W. Huang, D. R. Huang and Y. Q. Shi: Efficiently Self-Synchronized Audio Watermarking for Assured Audio Data Transmission. IEEE Transactions on Broadcasting, Vol.51 (2005) 69-76
12. http://www.mp3-tech.org/programmer/sources/eaqual.tgz
13. International Telecommunication Union: Method for Objective Measurements of Perceived Audio Quality (PEAQ). ITU-R BS 1387, (1998)
14. M. Arnold: Subjective and Objective Quality Evaluation of Watermarked Audio Tracks. Web Delivering of Music, (2002) 161-167
15. M. Steinebach, F.A.P. Petitcolas, F. Raynal, J. Dittmann, C. Fontaine, S. Seibel, N. Fates and L.C. Ferri: StirMark benchmark: audio watermarking attacks. Proc. of International Conference on Information Technology: Coding and Computing, (2001) 49-54
16. http://www.petitcolas.net/fabien/watermarking/stirmark/
17. S. J. Xiang and J. W. Huang: Analysis of D/A and A/D Conversions in Quantization-Based Audio Watermarking. International Journal of Network Security, Vol. 3 (2006) 230-238
18. S. J. Xiang, J. W. Huang and R. Yang: Time-scale Invariant Audio Watermarking Based on the Statistical Features in Time Domain. Proc. of the 8^{th} Information Hiding Workshop, (2006)
19. A. Lang, J. Dittmann: Profiles for Evaluation - the Usage of Audio WET. Proc. of SPIE Symposium on Electronic Imaging, Vol. 6072, 60721J, (2006)

Analysis of Optimal Search Interval for Estimation of Modified Quantization Step Size in Quantization-Based Audio Watermark Detection

Siho Kim[1] and Keunsung Bae[2]

[1] Telecommunication Network Business, Samsung Electronics Co., Ltd.
Gumi-City, Gyeong-Buk 730-350, Korea
[2] School of Electronic and Electrical Engineering, Kyungpook National University
Daegu 702-701, Korea
{si5, ksbae}@mir.knu.ac.kr

Abstract. The quantization-based watermarking schemes such as QIM or SCS are known to be very vulnerable to the amplitude modification attack. The amplitude modification attack results in the change of quantization step size so the estimation of a modified quantization step size is required before watermark detection. In this paper, we analyze the quantization error function of the audio signal having any shape of probability density function, and analytically determine the search interval that minimizes the quantization error considering both detection performance and computational complexity. It is shown that the appropriate search interval can be determined from the frame-based mean and variance of the input signal without regard to its shape of probability density function. Experimental results for real audio data verify that the derived search interval provides the accurate estimation of the modified quantization step size under amplitude modification attack.

Keywords: Audio watermarking, Amplitude scaling attack, Quantization-based, Quantization step size.

1 Introduction

Over the last few years, considerable audio watermarking algorithms have been proposed such as SS (spread spectrum) coding [1], phase coding [2], echo hiding [2-5], and so on. Recently, informed watermarking scheme [6-8] based on the Costa's dirty paper coding [9] is rapidly growing to replace the SS-based techniques. These kinds of watermarking do not need original host signals for watermark detection and host signals do not affect the performance of watermark detection, while the blind SS and echo coding suffer significantly from the host signal interference. Using the Costa's result, Chen et al. [6] proposed QIM (Quantization Index Modulation), and Eggers et al. [7] proposed SCS (Scalar Costa Scheme). However these schemes are known to be very vulnerable to amplitude modification attack. If the quantization step sizes used in embedding and extracting are different due to amplitude scaling, the detection performance can be degraded seriously. Thus it is required to estimate the

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 241 – 252, 2006.

modified quantization step size before extracting watermark information. To solve this problem, Eggers et al. [10] used a pilot-based estimation scheme, which embeds a pilot sequence via secure SCS watermarking and estimates the possible amplitude modification using securely embedded pilot sequence before watermark detection. This method requires a large number of samples for a pilot signal and the embedding space for watermark message is encroached due to a pilot signal. Lee et al. [11] proposed a preprocessed decoding scheme for the estimation of a scale factor using the EM (Expectation Maximization) algorithm, which does not encroach on the watermarking capacity since it uses the received signal itself for the estimation of the scale factor. However, EM algorithm needs a large number of samples for accurate estimation of a scale factor and then it can cause the impractical complexity. Kim et al. [12] proposed a robust algorithm to estimate the modified quantization step size, which searches the quantization step size to minimize the quantization error of the received audio signal on the analysis frame basis. It does not need a pilot signal and just use the received signal itself. However, it is important to determine the appropriate search interval that satisfies both detection performance and computational complexity.

In this paper, we analyze the quantization error function for the audio signal having any shape of probability density function using the Gaussian mixture model, and derive the equation to determine an appropriate search interval analytically. Especially we show that the search interval can be determined from the frame-based mean and variance of the input signal without regard to its shape of probability density function. The derived formula for optimal search interval is validated through the experiments with real audio data under amplitude modification and AWGN attacks with restricted power.

This paper is organized as follows. In section 2, we briefly review the blind watermarking schemes with scalar quantizers based on the Costa's dirty paper coding. In section 3, the optimal search interval for efficient estimation of a modified quantization step size is derived and explained in detail. In section 4, experimental results are shown with our discussions, and finally we make a conclusion in section 5.

2 Quantization-Based Watermarking Scheme

In quantization-based watermarking such as QIM or SCS, the binary watermark message $d \in \{0,1\}$ is embedded into a host signal using a dithered scalar quantizer, $Q_{\Delta,d}$, which is defined by equations (1) and (2).

$$Q_{\Delta,d}(x) \equiv Q_{\Delta}\left(x + \frac{\Delta}{2} \cdot d\right) - \frac{\Delta}{2} \cdot d \tag{1}$$

$$Q_{\Delta}(x) \equiv \left\lfloor \frac{x}{\Delta} + 0.5 \right\rfloor \cdot \Delta \tag{2}$$

where $Q_\Delta(\cdot)$ is a uniform scalar quantizater and Δ is a quantization step size. Then the watermarked signal, s, is obtained from the host signal, x, as follows.

$$s = x + \alpha \cdot [Q_{\Delta_e,d}(x) - x] \tag{3}$$

where α and Δ_e are embedding parameters. For a given watermark power or embedding distortion σ_w^2, these parameters are related by

$$\alpha = \sqrt{\frac{12 \cdot \sigma_w^2}{\Delta_e^2}} \tag{4}$$

As shown in equation (4), selecting an optimal α is equivalent to finding the optimal quantization step size Δ_e. However, it is hard to find analytically the optimal value of α for the structured codebook. In SCS [7], the optimal α and Δ_e are selected based on numerical optimization and the resulting optimal values are approximated by

$$\Delta_{e,opt} = \sqrt{12(\sigma_w^2 + 2.71\sigma_v^2)} \tag{5}$$

where σ_v^2 denotes the power of additive white Gaussian noise (AWGN) and σ_w^2 satisfies the condition of $(\sigma_w^2 + \sigma_v^2) << \sigma_x^2$. The QIM corresponds to a special case of the Costa's transmission scheme, where α =1 regardless of the noise variance σ_v^2.

If the amplitude modification attack that scales the watermarked signal by a scaling factor g, and AWGN attack are conducted, then the received signal r is written by equation (6).

$$r = g \cdot (s + v) \tag{6}$$

As a result, the quantization step size in the received signal becomes $\Delta_d = g \cdot \Delta_e$. Hence it is necessary to estimate the scale factor g before decoding process. If we use Δ_e instead of the changed step size Δ_d in decoding process, the watermark detection performance may be degraded seriously. Using the Δ_d, the estimated watermark signal, $\hat{d}$, is obtained by comparing the quantization error with $\Delta_d / 4$ as follows.

$$e = r - Q_{\Delta_d,0}(r) \tag{7}$$

$$\hat{d} = \begin{cases} 0, & |e| \le \dfrac{\Delta_d}{4} \\ 1, & |e| > \dfrac{\Delta_d}{4} \end{cases} \tag{8}$$

3 Determining the Search Interval for Estimation of Modified Quantization Step Size

3.1 Estimation Scheme of Modified Quantization Step Size [12]

The received watermarked signal that is attacked by amplitude scaling and AWGN can be rewritten by equation (9).

$$\begin{aligned} r &= g \cdot (s + v) \\ &= g \cdot x + g \cdot \alpha[Q_{\Delta_e,d}(x) - x]) + g \cdot v \\ &= g \cdot Q_{\Delta e,d}(x) + g(1-\alpha)(x - Q_{\Delta e,d}(x)) + g \cdot v \\ &= g \cdot Q_{\Delta e,d}(x) + w' + v' \end{aligned} \tag{9}$$

where w' and v' are defined as $g(1-\alpha)(x - Q_{\Delta e,d}(x))$ and $g \cdot v$, respectively. In general, the mean value of w' is zero because the mean of an audio signal can be assumed zero, but v' may be not. So we remove the mean of the received watermarked signal before processing. To find the modified quantization step size $g \cdot \Delta_e$, the quantization error function was defined in [12] as follows.

$$QE(\Delta) = E\left[\left(r - Q_\Delta(r)\right)^2\right] \tag{10}$$

The quantization error function QE has minimum value when the quantization step size, Δ, of a uniform scalar quantizer, $Q_\Delta(r)$, is equal to $g \cdot \Delta_e / 2$. The estimated quantization step size, $\hat{\Delta}_d$, is then obtained by equation (11).

$$\hat{\Delta}_d = 2 \cdot \arg\min_{\Delta}\left(QE(\Delta)\right) \tag{11}$$

By normalizing the QE function with $\Delta^2/12$, equations (10) and (11) can be rewritten as (12) and (13).

$$QE_N(\Delta) = 1 - \frac{12}{\Delta^2} \cdot QE(\Delta) \tag{12}$$

$$\hat{\Delta}_d = 2 \cdot \arg\max_{\Delta}\left(QE_N(\Delta)\right) \tag{13}$$

To estimate the quantization step size, we should find the point that QE_N has a maximum value in a given searching range, which is properly selected in the neighborhood of the quantization step size.

3.2 Analysis of QE Curve

To find the optimal search interval satisfying both detection performance and computational complexity at the same time, first we should investigate the

characteristic of the QE function. The QE function represents the quantization error caused by quantizing the received audio signal, which was quantized with Δ_q, with a quantization step size Δ. For simplicity, we consider the QIM method with $\alpha = 1$. We define the host signal quantized with Δ_q as s_{Δ_q}, and quantization error caused by quantizing s_{Δ_q} with Δ as $e_\Delta(k)$. As shown in figure 1, $e_\Delta(k)$ increases in proportion to k, that is to say, $e_\Delta(k) = k(\Delta - \Delta_q)$. However if the value of $e_\Delta(k)$ exceeds $\Delta/2$, it becomes $-\Delta/2$ because it is quantized with next codeword. Finally we can obtain $e_\Delta(k)$ like figure 1(b). Therefore the quantization error, $e_\Delta(k)$, can be represented as follows.

$$e_\Delta(k) = \mathrm{mod}\left((\Delta - \Delta_q)\cdot k + \frac{\Delta}{2}, \Delta \right) - \frac{\Delta}{2}, \quad k = 0, \pm 1, \pm 2, \ldots \tag{14}$$

where mod denotes a modulo operation. If the slope, i.e., difference between Δ_q and Δ, is large, the power of $e_\Delta(k)$, QE, follows the average noise power of quantization error, $\Delta^2/12$. But in a small slope, it shows different curve so we now analyze it.

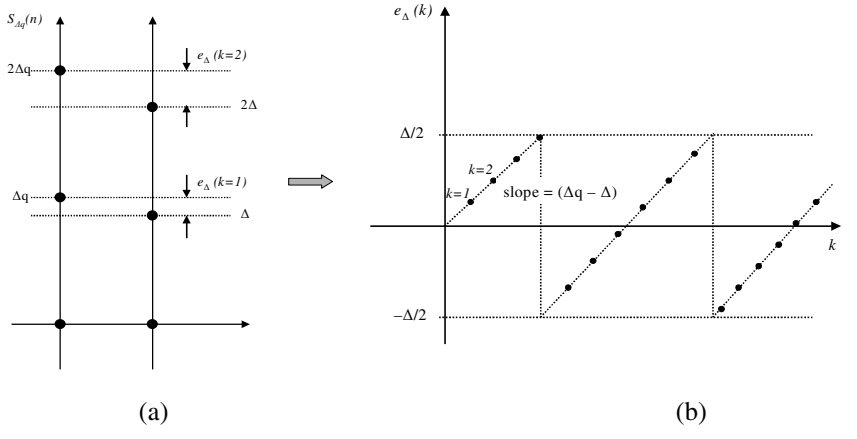


Fig. 1. Quantization error caused by quantizing s_{Δ_q} with Δ

If we denote the probability density function of a host signal as $f(x)$, then QE function can be rewritten by equation (15) as shown in [12].

$$QE(\Delta) = \sum_{k=-\infty}^{\infty} \left[\Delta_q \cdot f(k\cdot\Delta_q)\cdot e_\Delta(k)^2 \right] \tag{15}$$

From equation (15), we can suppose that the shape of QE curve have relevance to the probability density function of a host signal. In general, any probability density function can be presented by Gaussian mixture model as shown in equation (16) if the number of mixture, *L*, is infinite.

$$f(x) = \sum_{i=1}^{L} \eta_i \cdot \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \tag{16}$$

where σ_i and μ_i is the standard deviation and mean of the i^{th} mixture component. And η_i is the existence probability of i^{th} mixture. When the quantization step size Δ is in the neighborhood of Δ_q, the quantization error, $e_\Delta(k)$, can be represented as a linear line with slope δ if we assume that there are no samples over maximum value, *K*, as shown figure 2. Then, $e_\Delta(k)$ is expressed by equation (17).

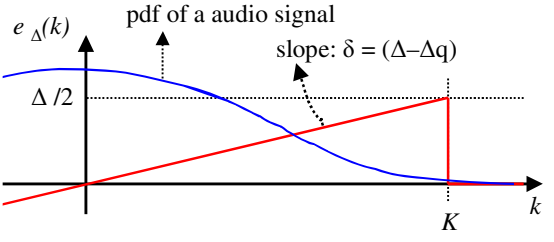


Fig. 2. Approximation of error function

$$e_\Delta(k) = \delta \cdot k \,\big|_{-K \le k \le K} \tag{17}$$

From equation (16) and (17), the QE function is rewritten by equation (18).

$$\begin{aligned} QE(\Delta) &= \int \sum_{i=1}^{L} \eta_i \frac{\Delta_q}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(\Delta_q k-\mu_i)^2}{2\sigma_i^2}} \cdot (\delta \cdot k)^2 dk \\ &= \sum_{i=1}^{L} \eta_i \cdot \int_{K_i^{\min}}^{K_i^{\max}} \frac{\Delta_q}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(\Delta_q k-\mu_i)^2}{2\sigma_i^2}} \cdot (\delta \cdot k)^2 dk \end{aligned} \tag{18}$$

In equation (18), the maximum value, $K_i^{\max}$, and minimum value, $K_i^{\min}$, of i^{th} mixture is given as shown in equation (19).

$$K_i^{\max} = \frac{(\mu_i + m \cdot \sigma_i)}{\Delta_q}, \; K_i^{\min} = \frac{(\mu_i - m \cdot \sigma_i)}{\Delta_q} \tag{19}$$

If we define $K_i = \max(|K_i^{\max}|, |K_i^{\min}|)$, the effective range of δ at i^{th} mixture is given like equation (20).

$$|\delta \cdot K_i| \leq \frac{\Delta}{2} \; or \; -\frac{\Delta}{2K_i} \leq \delta \leq \frac{\Delta}{2K_i} \tag{20}$$

By expanding (18), we can derive QE like equation (21).

$$QE(\Delta) = \sum_{i=1}^{L} \eta_i \cdot \frac{(\Delta - \Delta_q)^2}{\Delta_q^2} \cdot \left(\sigma_i^2 \cdot \left(-\sqrt{\frac{2}{\pi}} \cdot m \cdot e^{-\frac{m^2}{2}} + erf\left(\frac{m}{\sqrt{2}}\right) \right) + \mu_i^2 \cdot erf\left(\frac{m}{\sqrt{2}}\right) \right) \tag{21}$$

If m is larger than 3, $\left(-\sqrt{\frac{2}{\pi}} \cdot m \cdot e^{-\frac{m^2}{2}} + erf\left(\frac{m}{\sqrt{2}}\right) \right) \cong 1$ and $erf\left(\frac{m}{\sqrt{2}}\right) \cong 1$ can be assumed. Then QE is approximated as follows.

$$\begin{aligned} QE(\Delta) &\cong \sum_{i=1}^{L} \eta_i \cdot \frac{(\Delta - \Delta_q)^2}{\Delta_q^2} \cdot (\sigma_i^2 + \mu_i^2) \Big|_{m>3} \\ &\cong \frac{(\Delta - \Delta_q)^2}{\Delta_q^2} \sum_{i=1}^{L} \eta_i \cdot (\sigma_i^2 + \mu_i^2) \end{aligned} \tag{22}$$

In equation (22), $\sigma_i^2 + \mu_i^2$ means the power of i^{th} mixture, then $\sum_{i=1}^{L} \eta_l \cdot (\sigma_i^2 + \mu_i^2)$ is the total power of a host signal. Finally, QE can be expressed by following formula,

$$QE(\Delta) \cong \frac{(\Delta - \Delta_q)^2}{\Delta_q^2} (\sigma_s^2 + \mu_s^2) \tag{23}$$

where σ_s and μ_s are the standard deviation and mean of a host signal, respectively. In general, μ_s is zero in audio signal. From equation (23), we can verify that the optimal search interval is not related to the shape of probability density function of host signal.

3.3 Determining the Search Interval

Using equation (23), we should find the optimal search interval that can detect the value that exceeds some threshold of the peak of QE_N, i.e., $\beta \cdot QE_N$. In other words, we can obtain δ satisfying equation (24).

$$QE_N(\Delta_q + \delta) = \beta \cdot QE_N(\Delta_q) \; |_{0<\beta<1} \tag{24}$$

And then the search interval, $w(\Delta)$, which guarantees the detection of the modified quantization step size, is calculated by

$$w(\Delta_q) = 2 \cdot |\delta| \tag{25}$$

By substituting (23) into (24) using equation (12) and solving it, we can obtain the following equation,

$$\delta \cong \Delta_q^2 \cdot \sqrt{\frac{1-\beta}{12 \cdot (\sigma_s^2 + \mu_s^2)}} \tag{26}$$

If we calculate the effective range of β using equation (20) and (26), equation (27) is obtained.

$$\beta_{\min} \le \beta < 1 \left| \beta_{\min} = \max\left(1 - \frac{3 \cdot (\sigma_i^2 + \mu_i^2)}{(|\mu_i| + m \cdot \sigma_i)^2} \right), \; i = 1, \ldots, L \right. \tag{27}$$

The δ in equation (26) is valid only when β is in the range of equation (27). However, to find $\beta_{\min}$, we should calculate all means and standard deviations of Gaussian mixture. Instead of that, we will calculate the theoretical maximum value of $\beta_{\min}$. In equation (27), $\beta_{\min}$ is rewritten as follows.

$$\begin{aligned} \beta_{\min} &= \max\left(1 - \frac{3 \cdot (\sigma_i^2 + \mu_i^2)}{(|\mu_i| + m \cdot \sigma_i)^2} \right) \\ &= \max\left(1 - \frac{3 \cdot (1 + n_i^{\,2})}{(n_i + m)^2} \right) \Bigg| \mu_i = n \cdot \sigma_i \end{aligned} \tag{28}$$

Let us define $\beta_{\min}(i) \equiv 1 - \frac{3 \cdot (1 + n_i^{\,2})}{(n_i + m)^2}$. To find the maximum value of $\beta_{\min}(i)$, we should solve following equation.

$$\frac{\partial \beta_{\min}(i)}{\partial n_i} = 0 \tag{29}$$

By solving equation (29), we can obtain $n = \frac{1}{m}$ and the maximum value of $\beta_{\min}(i)$ is obtained as given in equation (30).

$$\beta_{\min}(i) \le 1 - \frac{3}{1 + m^2} \Bigg| n_i = 1/m \tag{30}$$

As a result, the effective range of β is given as follows.

$$\beta_{\min} \le \beta < 1 \left| \beta_{\min} = 1 - \frac{3}{1 + m^2} \right. \tag{31}$$

4 Experimental Result

In order to validate the derived search interval and evaluate its detection performance, we applied it to an audio watermarking system. The audio watermarking system used in the experiments is illustrated in figure 3, which simply embeds the binary watermark message into an audio signal sample by sample in time domain. The audio signal is normalized to the predefined power σ_N^2 before watermark embedding and extracting. In the extractor, Δ estimation block consists of two components, which are coarse searching and fine searching. In coarse searching process, it searches the coarse position of peak of QE using the search interval obtained from equation (26). Then, in fine searching process, the neighborhood of the coarse peak position is searched by dense interval. It provides the detection of an exact quantization step size with low computational load.

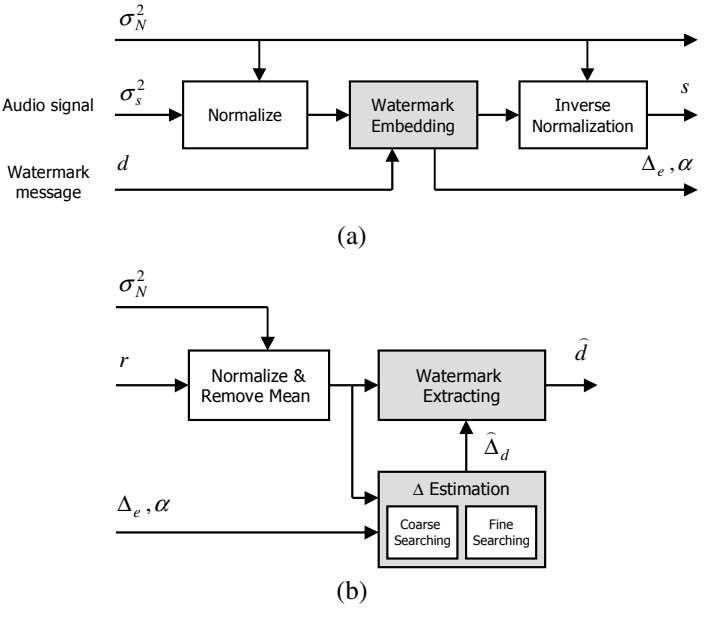


Fig. 3. Audio watermarking system used in experiments (a) Embedder (b) Extractor

First, we compared the measured peak width of the QE curve for a real audio signal and the estimated one from equation (26). In the experiment, we set the parameter β to 0.85. The quantization step size of received audio signal, Δ_q, is 50 and the frame length is 1024. Figure 4 shows the result for the 80 frames of an audio signal and we can see that two values are almost same. In other words, δ in equation (26) provides the exact peak width of QE curve. Figure 5 shows the trace of the estimated $\hat{\Delta}_d$ on a frame basis. In the figure, D1 and D2 denote σ_w^2 and ($\sigma_w^2 + \sigma_v^2$),

respectively. When the value of D2/D1 is 1.1 or 1.3, the estimated value of $\hat{\Delta}_d$ is almost exact, while it shows some inaccurate results in case of D2/D1= 1.5.

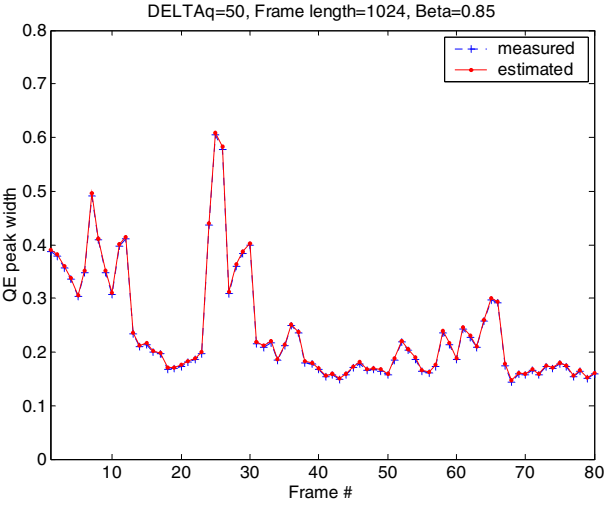


Fig. 4. Comparison of the peak width of QE curve

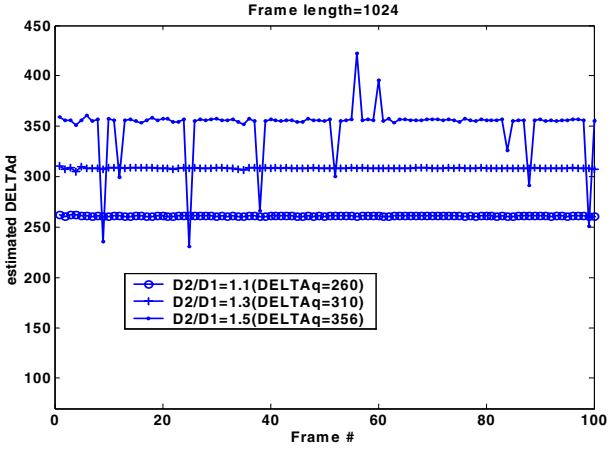


Fig. 5. Trace of the estimated $\hat{\Delta}_d$ on a frame basis

We evaluated the detection error rate (BER: Bit Error Rate) in the presence of the AWGN. Figure 6 shows the results and we can see that BER increases proportionally to AWGN. The figure shows the result for three different methods for extracting watermark information: 'Exact' is with the exact quantization step size $g \cdot \Delta_e$ that we

modified for amplitude modification attacks, 'Not compensated' with the quantization step size Δ_e used for embedding, and 'Compensated' with the estimated quantization step size $\hat{\Delta}_d$ using the proposed algorithm. As expected, 'Not compensated' shows the poorest detection performance even though AWGN is weak. But the proposed method provides almost the same detection performance with that of an exact modified quantization step size under limited noise power although BER increases as D2/D1 increases. However, it is shown that we can improve the BER by increasing the frame length N as shown figure 6(b).

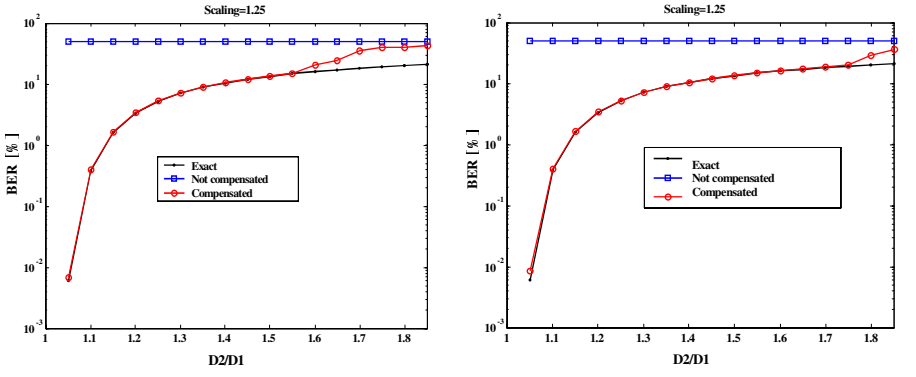


Fig. 6. BER as D2/D1 (a) N=1024 (b) N=8192

5 Conclusion

In this paper, we analyze the QE curve for the audio signal with any shape of probability density function using Gaussian mixture model and analytically derived the search interval considering both detection performance and computational complexity. We showed that the optimal, in some sense, search interval could be determined from the frame-based mean and variance of the input signal without regard to its shape of probability density function. The experimental results with real audio data validated the derived formula for an optimal search interval, and the estimated modified quantization step size using the search interval provided a good performance under amplitude modification and AWGN attacks with restricted power.

References

1. L. Boney, A. Tewfik and K. Hamdy, "Digital watermarks for audio signals," IEEE Int. Conference on Multimedia Computing and Systems (1996) 473-480
2. W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for data hiding," IBM Systems Journal, Vol.35, Nos 3&4 (1996)
3. D. Gruhl, Anthony Lu, "Echo hiding," in Proc. Information Hiding Workshop, Cambridge University, U.K. (1996) 295-315

4. Hyen-O Oh, Dae-Hee Youn, Jin-Woo Hong, Jong-Won Seok, “Imperceptible echo for robust audio watermarking,” AES 113th Convention, Los Angeles, CA, USA, Oct. (2002)
5. Siho Kim, Hongseok Kwon, Keunsung Bae, “Modification of polar echo kernel for performance improvement of audio watermarking,” International Workshop on Digital Watermarking 2003, Seoul, (2003) 477-487
6. B. Chen and G. W. Wornell, “Quantization index modulation: A class of provably good methods for digital watermarking and information embedding,” IEEE Transaction on Information Theory, Vol.47, No.4, (2001)
7. J. J. Eggers, J. K. Su, and B. Girod, “A blind watermarking scheme based on structured codebooks,” in Secure Image and Image Authentication, Proc. IEE Colloquium, London, UK, (2000) 4/1-4/6
8. P Moulin, M. K. Mihcak, and G. I. Lin. “An information-theoretic model for image watermarking and data hiding,” IEEE Int. Conference on Image Proc., Vancouver, B. C., September (2000)
9. M. H. M. Costa, “Writing on dirty paper,” IEEE Transactions on Information Theory, Vol.29, No.7, (1983) 439-441
10. J. J. Eggers, R. Bauml, and B. Girod, “Estimation of amplitude modifications before SCS watermark detection,” SPIE: Multimedia Systems and Applications IV, Vol.4675, San Jose, USA, (2002) 387-398
11. Kiryung Lee, Dongsik Kim, Taejeong Kim, and Kyungae Moon, “EM estimation of scale factor for quantization-based audio watermarking,” International Workshop on Digital Watermarking 2003, Seoul, (2003) 335-346
12. Siho Kim, and Keunsung Bae, “Robust estimation of amplitude modifications for scalar costa scheme based audio watermarking,” International Workshop on Digital Watermarking 2004, Seoul, (2004) 121-135

Universal JPEG Steganalysis in the Compressed Frequency Domain

Johann Barbier[1,2], Éric Filiol[1], and Kichenakoumar Mayoura[1]

[1] École Supérieure et d'Application des Transmissions,
Laboratoire de Virologie et Cryptologie,
BP 18, 35998 Rennes Cedex, France
[2] Centre d'Électronique de l'ARmement, Département de Cryptologie,
La Roche Marguerite, BP 57419,
35174 Bruz Cedex, France
johann.barbier@dga.defense.gouv.fr

Abstract. We present in this paper a new approach for universal JPEG steganalysis and propose studying statistics of the compressed DCT coefficients. This approach is motivated by the *Avalanche Criterion* of the JPEG lossless compression step. This criterion makes possible the design of detectors whose detection rates are independent of the payload. We design an universal steganalytic scheme using blocks of the JPEG file binary output stream. We compute higher order statistics over their Hamming weights and combined them with a Kullbak-Leibler distance between the probability density function of these weights and a benchmark one. We evaluate the universality of our detector through its capacity to efficiently detect the use of a new algorithm not used during the training step. To that goal, we examinate training sets produced by Outguess, F5 and JPhide-and-Seek. The experimental results we obtained show that our scheme is able to detect the use of new algorithms with high detection rate ($\approx 90\%$) even with very low embedding rates ($< 10^{-5}$).

Keywords: universal steganalysis, JPEG, Kullbak-Leibler distance, Fisher discriminant.

Introduction

Steganography is the science of *covered writing*. Its purpose is to hide information in a cover medium so that it is "hard" for everyone to detect the existence of the embedded information. On the opposite side, steganalytic schemes tend to detect hidden information in a mass of cover media. Let Alice and Bob communicate using a steganographic algorithm $\mathcal{A}$, for instance, to hide the world their love affair, and Eve, the paparazi, who will earn lot of money if she can prove Alice and Bob are lovers. In a classical model and according to the Kerchoff's principles, Eve knows all the steganographic techniques Alice and Bob are likely to use. So, she can design dedicated methods to detect the use of $\mathcal{A}$ specifically; this is

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 253–267, 2006.

called *specific steganalysis*. In a harder model of attack, we make the hypothesis that Alice and Bob keep their steganographic algorithm secret and Eve does not know the specifications of $\mathcal{A}$. Her goal is now to build a detector, which does not depend on $\mathcal{A}$ and which distinguishes cover and stego media in order to prove that Alice and Bob indeed share secrets through steganography; this is called *universal steganalysis*.

Specific and universal steganalysis do not achieve the same goal; the specific steganalysis answers the question: "*Is the medium was embedded with the algorithm $\mathcal{A}$?*" and the universal steganalysis answers the question : "*Is the medium a stego medium ?*". Even if the universal steganalysis is more general and so less efficient than the specific one for detecting the use of a given steganographic algorithm, there are two main interests for using it. First, universal steganalysis schemes are independent of the steganographic algorithms; stego media embedded with an *unknown* algorithm may also be detected by such schemes. Secondly, it is the only possible way to detect the use of steganographic algorithm for which no specific steganalysis is known. So, the central property universal steganalysis schemes should verify is the following one: given a set of known steganographic algorithms for the training step, we are able to detect the use a new steganographic algorithm which is not in the previous set. If it is not the case, the considered scheme is vulgarly dependent on a steganographic algorithm and is a specific steganalysis scheme. In the remaining of this paper this main property will be called *universality property*. We also propose to extend the definition of universal steganalysis to a stronger concept of *unconditional steganalysis*. A steganalytic scheme would be said *unconditional* if and only if, given a set of known steganographic algorithms, the scheme is able to detect the use of *any* new steganographic algorithm which is not in the previous set. The universal steganalysis will be studied through the scope of the universality property and the efficiency of the proposed scheme will be measured by the detection rates when detecting algorithms which are not in the training set.

In this paper, we take place in Eve's shoes, and our goal is to detect the existence of embedded message into JPEG images. The training set of our universal steganalytic scheme is composed of images embbeded with the well known steganographic algorithms, Outguess [1], F5 [2] and JPHide [3] but it can also be designed with another algorithms in the same way. In JPEG steganalysis, people traditionally try to find detectable properties directly studying statistics of the DCT coefficients or of the decompressed images. By contrast, we propose to examine Huffman compressed data, which are DCT coefficients compressed first by RLE and then by Huffman compression algorithms. We point out new statistical features to detect hidden information in JPEG images. We examine different cases of training sets and evaluate the universality property of our scheme.

In the first section, we quickly present the JPEG standard and DCT-based steganography. We also present a new approach for JPEG steganalysis based on statistics in the compressed frequency domain. In the second section, we recall state of the art JPEG steganalysis techniques, and put our approach back in its place. Then, we present our statistical features and evaluate the efficiency

of the scheme using Outguess, F5, and JPHide algorithms. In section 3, we explain the design of our Fisher classifier and detail the experimental framework and the results we obtained. Finally, we conclude in the last section and give some discussions.

1 JPEG Steganography

1.1 The JPEG Format

The Joint Photographic Expert Group (JPEG) was created in 1986. This Group worked on digital compression and coding of continuous-tone still images. These studies have led to the CCITT[1] recommendation T.81 and the ISO[2] Standard 10918-1.

The JPEG format defines four types of compression modes which are sequential, progressive, hierarchical and lossless. In our case, the progressive mode is used.

DCT[3]-Based Coding. The figure 1 explains the main procedures for all encoding processes based on the DCT. In order to simplify, the diagram operates on a single-component image.

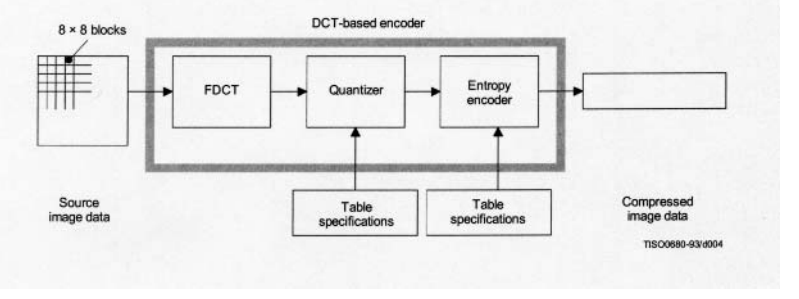


Fig. 1. DCT-based encoder simplified diagram

Main Characteristics of Coding Processes. A digital image can be represented by pixels. The three color coefficients (Red, Green, Blue or RGB) for each pixel are transformed into a new coding scheme: one luminance coefficient (Y) and two chrominance coefficients (U and V or also called Cb and Cr).

After the conversion from RGB to YCbCr, the values, are gouped in 8×8 pixels blocks, and transformed by a forward DCT. Most of the frequency coefficients obtained are very low and we can remove a lot of them and still reconstruct the

[1] International Telegraph and Telephone Consultative Committee.
[2] International Standard Organisation.
[3] Discrete Cosine Transform.

original values. The low frequencies are conserved while the high frequencies are removed.

After the DCT transformation on each block, the DCT coefficients are quantized. This step called quantization is the main lossy process. The coefficients are divided with fixed values coming from a specified table and then rounded. Most of the quantized DCT coefficients are equal to zero.

The "zig-zag" order consists to order the coefficients in each 8×8 block (most of them are equal to zero).

After the "zig-zag" sequence, the last steps are lossless compression. First a simple RLE[4] is used to compress the high frequency coefficients. Then a Huffman coding procedure is applied. Finally, the output is the JPEG raw binary data.

1.2 Embedding Information in the DCT Coefficients

The JPEG compression process can be divided into two main parts: the first one computes quantized DCT coefficients from a bitmap image $\mathcal{B}$ and some parameters $\mathcal{P}_1$; it will be noted $\mathcal{C}_l$.

$$\mathcal{C}_l : (\mathcal{B}, \mathcal{P}_1) \longrightarrow (DCT_i), \text{ where } DCT_i \in \mathbb{Z}.$$

$\mathcal{C}_l$ is a lossy compression, that means $\mathcal{C}_l$ is not a bijective mapping. So, if we apply $\mathcal{D}_l$, the decompression algorithm associated to $\mathcal{C}_l$ we don't retrieve $\mathcal{B}$.

$$\mathcal{D}_l : ((DCT_i), \mathcal{P}_1) \longrightarrow \mathcal{B}^{'} \text{ with } \mathcal{B}^{'} \neq \mathcal{B}.$$

The second one computes a string of binary compressed data from quantized DCT coefficients and some parameters $\mathcal{P}_2$; it will be noted $\mathcal{C}_u$.

$$\mathcal{C}_u : ((DCT_i), \mathcal{P}_2) \longrightarrow (b_j) \text{ where } b_i \in \mathbb{F}_2.$$

$\mathcal{C}_u$ is an unlossy compression, that implies it is a bijective mapping.

Since $\mathcal{C}_l$ is not a bijective mapping, one cannot naturally hide information during the first step, otherwise some of the embedded information will not be retrieved. Information can only be hidden during the second step. This step, as we saw previously, is divided into zig-zag re-ordering, RLE and Huffmann compressions. So, the only practical way to embed any information is in DCT coefficients, after RLE or Huffmann compressions. To minimize the distortions of the original image, DCT are the most adapted.

The main problem, when embedding information in DCT coefficients, is to preserve the statistics of the cover medium. State of the art steganographic systems take care of keeping DCT statistics unchanged, histogram for example, but even if DCT statistics are preserved, many steganalysis schemes [4,5,6,7,8] are based on deviations of some decompressed cover image statistics. It seems that both cannot be preserved at the same time.

[4] Run Length Encoding.

2 Detecting JPEG Stego Images

2.1 JPEG Steganalysis Methods

Different approaches have been used to detect stego images. The first one consists in studying directly DCT coefficients like J. Fridich [9,10] who looked at first order statistics and at the discontinuity of DCT coefficients at the borders of blocks for detecting the use of F5 and Outguess. She also pointed out some other features for the frequency domain [11,12] for JPEG syteganalysis.

The second approach is dedicated to the spatial domain. H. Farid and S. Lyu obtained classifier with a high detection rate by combining Support Vector Machines (SVM) with higher order statistics [5,6] or with wavelet transform statistics [7,8] of decompressed JPEG image. J. J. Harmsen et al. [13] proposed to use a Fisher discriminant instead of a SVM and I. Avicib et al. [4] introduced metrics based on images quality.

Previous methods have even been used together [14] to increase the accuracy of detectors. Among these techniques we can distinguish two categories of steganalysis: *specific steganalysis* and *universal steganalysis*.

Specific Steganalysis. Specific steganalysis is dedicated to only a given embedding algorithm. It may be very accurate for detecting images embedded with the given steganographic algorithm but it fails to detect those embedded with another algorithm. Techniques developped in [9,10,11,13] are specific.

Universal Steganalysis. Universal steganalysis enables to detect stego images whatever the steganographic system be used. Because it can detect a larger class of stego images, it is generally less accurate for one given steganographic algorithm. Methods presented in [4,5,12,14,6,7,8] are universal.

In this paper, we will study an universal method adapted for the compressed frequency domain. But, unlike classical universal steganalysis schemes, the main point we want to show is the ability of our universal steganalyzer to detect the use of a steganographic algorithm not used during the training step.

2.2 A New Point of View

We have to keep in mind three important intuitive assertions:

- embedding information in DCT_i, will change $\mathcal{D}_l((DCT_i), \mathcal{P}_1)$ but also $\mathcal{C}_u((DCT_i), \mathcal{P}_2)$.
- one cannot preserve at the same time the statistics of DCT_i, those of $\mathcal{D}_l((DCT_i), \mathcal{P}_1)$ and $\mathcal{C}_u((DCT_i), \mathcal{P}_2)$.
- hiding information tends to introduce a variation of entropy.

Most of steganalytic techniques consist in observing some statistical deviations directly on DCT coefficients or in $\mathcal{D}_l((DCT_i), \mathcal{P}_1)$. We propose here to explore statistics in $\mathcal{C}_u((DCT_i), \mathcal{P}_2)$.

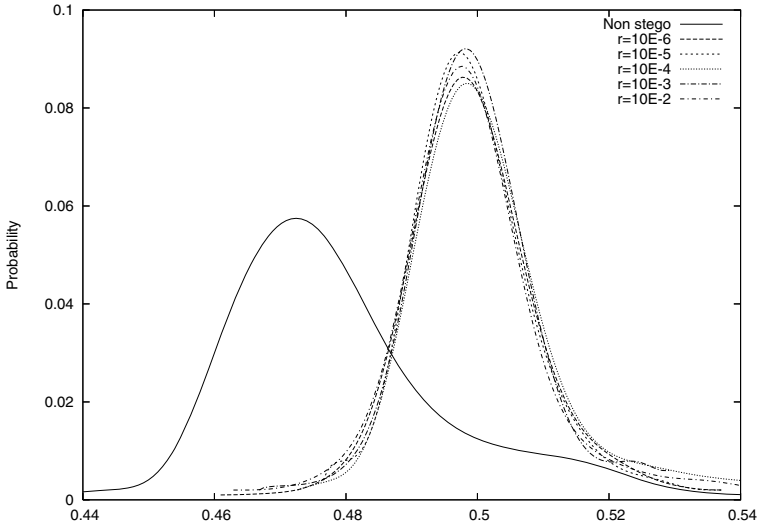


Fig. 2. Density probability functions of P for JPHide stego and non stego images

Let I a given JPEG image to analyse and (b_j)[5] the output of $\mathcal{C}_u$. We noticed a variation of the entropy of the output stream when the image has been embedded with a steganographic scheme. The binary entropy $H(I)$ is given by

$$H(I) = -P(I)\log P(I) - (1 - P(I))\log(1 - P(I)), \tag{1}$$

where $P(I)$ is the probability that b_j is equal to 0. Observing a deviation of the binary entropy is equivalent to observe a deviation of P. For non stego images, P follows a Gamma probability density function, whereas the probability density function is different for stego images. More surprisingly, P follows a normal $\mathcal{N}(0.5, \sigma)$ probability function and so, whatever the embedding rate, r, is, as shown in the figure 2. This difference of probability laws for stego and non stego images is explained by the avalanche criterion [15] of the RLE and Huffman compression step. As shown in figure 3, when only few bits of the DCT coefficients LSB are flipped, after RLE and Huffman compression almost half the bits are flipped. So, when embedding few bytes, $P(I)$ becomes closer to 0.5. These phenomena is amplified since the avalanche criterion is close to 0.5 when only few bytes of DCT coefficients are changed and since steganography systems embed additional DCT coefficients to keep first order statistics unchanged. This criterion makes possible the existence of steganalyzers which the detection rates are quasi-independent of the payload.

Moreover, we noticed a variation of higher order statistics of $\mathcal{C}_u((DCT_i), \mathcal{P}_2)$ when a message is embedded, despite $\mathcal{C}_u$ is a bijective mapping and the stegano-

[5] (b_j) is only composed of the RLE and Huffman compressed DCT coefficients and does not include the JPEG file header.

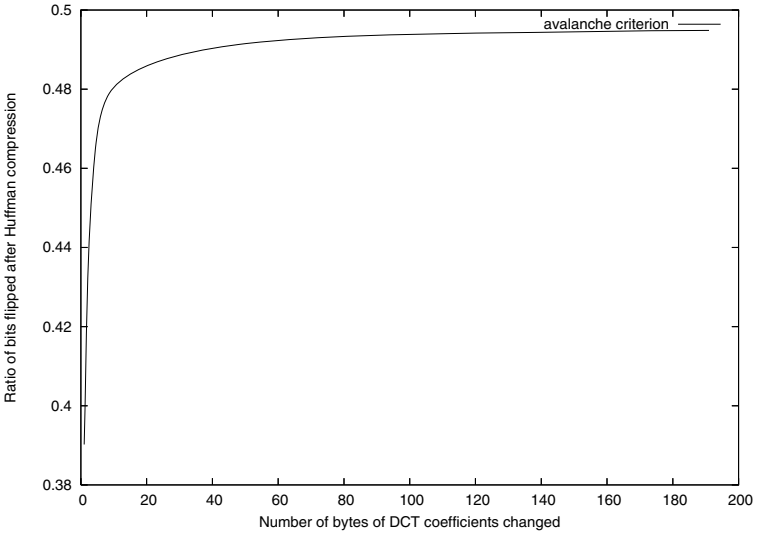


Fig. 3. Avalanche criterion of RLE+Huffman compression function

graphic algorithm $\mathcal{A}$ tends to preserve the statistics of DCT coefficients. We have designed a steganographic distinguisher based on this deviations.

2.3 Universal Steganalysis Scheme

As previously, let I a given JPEG image to analyse, (b_j) the output of $\mathcal{C}_u$ and $P(I)$ the probability that b_j is equal to 0. To obtain a set of statistics on (b_j), we divide the stream (b_j) into blocks B_i of size s bytes, so that

$$B_i = b_{i\times 8s+1} \dots b_{(i+1)\times 8s} \in \mathbb{F}_2^s. \tag{2}$$

We estimate the Hamming weights, $w(B_i) = \sum_{j=1}^{s} b_{i\times 8s+j}$, for the stream blocks.

Let $X \in \Omega = [0, 8s]$ be the random variable which values are the $w(B_i)$. We compute the probability density function followed by X and its moments of order i, $\mathcal{M}_i(I)$. As illustrated in figure 5, X does not follow the same probability density function whether I is a stego image or not. So, we experimentaly compute the average probability function $p(x)$ followed by X when I is a cover media (figure 6) and introduce the Kullbak-Leibler distance to measure the dissimilary between the observed probability density function $\hat{p}(x)$ and $p(x)$. This distance $\mathcal{D}(\hat{p}, p)$ is defined by

$$\mathcal{D}(\hat{p}, p) = \sum_{x\in\Omega} \hat{p}(x).\log\frac{\hat{p}(x)}{p(x)}. \tag{3}$$

As the Kullbak-Leibler distance is not symmetric, we define $\mathcal{D}_1(I) = \mathcal{D}(\hat{p}, p)$ and $\mathcal{D}_2(I) = \mathcal{D}(p, \hat{p})$.

Fig. 4. image_12547.jpg

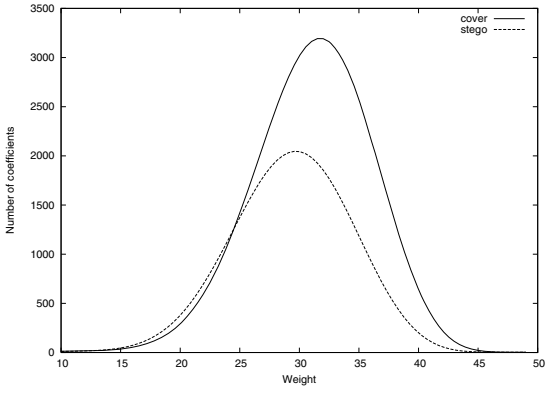


Fig. 5. Deviation of weight for image_12547.jpg, using Outguess and $s = 8$

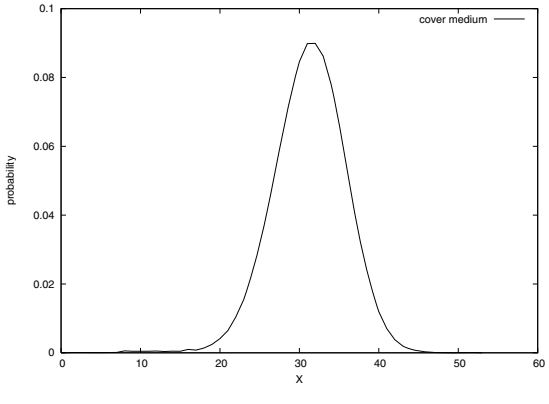


Fig. 6. Average probability density function for cover media for $s = 8$

Now, we will map I to the statistical vector $\mathcal{V}(I)$ of $k+3$ coordinates defined by

$$I \longrightarrow \mathcal{V}(I) = (\mathcal{M}_0(I), \ldots, \mathcal{M}_k(I), P(I), \mathcal{D}_1(I), \mathcal{D}_2(I)), \tag{4}$$

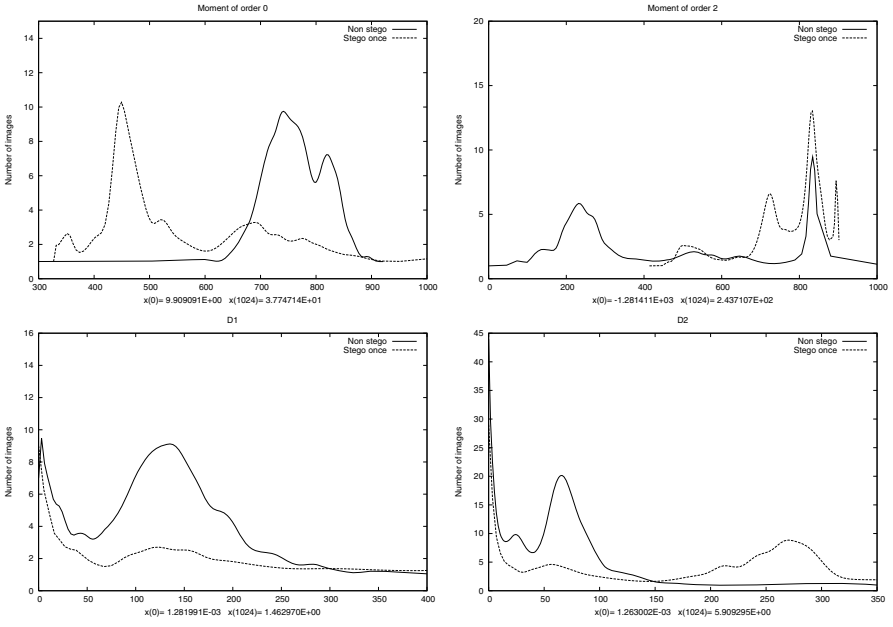


Fig. 7. Histograms of $\mathcal{M}_0$, $\mathcal{M}_2$, $\mathcal{D}_1$, $\mathcal{D}_2$ for Outguess and $s = 8$

and design an universal steganalysis scheme which the parameters are (s, k). Each component of the statistical vector does not follow the same probability density function whether the image I is a stego one or not, as illustrated in figure 7 [6]

We have pointed out some statistics in the compressed frequency domain which are liable to deviation when information is embedded with a steganographic algorithm such as Outguess, F5 or JPHide. These statistical features are independent of a specific algorithm, but to prove our scheme is an efficient universal steganalysis scheme we still have to evaluate its universality property, as discussed in the introduction.

3 Experimental Results

3.1 Classifier Design

We need a set, $\mathcal{C}$ of cover media and a set, $\mathcal{S}$ of stego images. For convenience, these samples have the same cardinality n, but the following method can be easily adapted with learning sets of different cardinalities.

First, for each set, we compute $\mathcal{V}_c = \{\mathcal{V}(I)|I \in \mathcal{C}\}$ as defined in (4), and $\mathcal{V}_s = \{\mathcal{V}(I)|I \in \mathcal{S}\}$ which are subsets of $\mathbb{R}^{k+3}$. We denote g_c, respectively g_s, the barycenter of $\mathcal{V}_c$, respectively $\mathcal{V}_s$, and g the barycenter of g_c, g_s. Then, we take g as the origin of the system of coordinates and compute the covariance matrices,

[6] The x axis have been linearly transformed. The statistical value i is mapped to $x(i)$.

V_c and V_s. Finally, we compute the intraclass and interclass variance matrices W and B defined under our hypothesis by

$$B = \frac{1}{2}(g_c - g_s)(g_c - g_s)^{'}, \tag{5}$$

$$W = \frac{1}{2}\left(V_c + V_s\right). \tag{6}$$

The variance matrix, V is given by $V = B + W$.

The Fisher discrimination analysis [16] consists in finding a projection axis which discriminates the best $\mathcal{V}_c$ and $\mathcal{V}_s$ and thus $\mathcal{C}$ and $\mathcal{S}$. This axis, (g_c, g_s), is defined by the vector

$$u = W^{-1}(g_c - g_s), \tag{7}$$

where $M = W^{-1}$ can be considered as a metric. Actually, a new image I, representated by the point p will be said to be in $\mathcal{C}$, if $d^2(p, g_c) > d^2(p, g_s)$, where d is a distance based on the metric M. According to the Mahalanobis-Fisher rule, we decide that I is in $\mathcal{C}$ if and only if

$$p.u = pM(g_c - g_s) > T, \tag{8}$$

where T is the detection threshold. Another metric can also be considered, setting $M = V^{-1}$.

3.2 Learning Step

For each training of our classifiers, we used between 3,000 and 4,000 randomly choosen images from a database of about 100,000 JPEG images downloaded from the web, notably *https://www.worldprints.com* in 2000. The database is composed of grayscale and color images of different sizes. We disposed of a set $\mathcal{A} = \{Outguess, F5, JPHide\}$ of three known algorithms, for which known specific attacks exist. Our goal was to produce a subset $A^{'} \subset \mathcal{A}$ for training our classifier so as to, at least the use of one algorithm in $\bar{A}'$ can be efficiently detected. We chose to configure our scheme with $k = 5$, which is a good trade off between reasonable computing time and a good detection accuracy. We tested different values for s: 8, 16, 32 and 64. To show the effectiveness of our approach with very low embedding rates, we mixed stego images with an embedding rate from 10^{-6} to 10^{-2}. We tried all the subsets but $\mathcal{A}$ and the empty set. For illustration, we give in table 1 the best experimental parameters obtained for a subset $A_1 = \{F5, JPHide\}$ of two algorithms and $A_2 = \{Outguess\}$, a subset of one algorithm. The training set for A_1 was composed of 2,000 cover media and 2,000 stego images embedded respectively by F5 and JPHide with the embedding rates 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3} and 10^{-2}. The training set for A_2, figure 8, was composed of 1,500 cover media and 1,500 stego images embedded by Outguess with the same embedding rates. For each training set, we determined the discriminant factor u for the metrics W^{-1} and V^{-1} as defined in section 3.1. We also chose the value for s which gave us the best detection rate for the training set, in order to preserve the detection of cover media during the wild detection step.

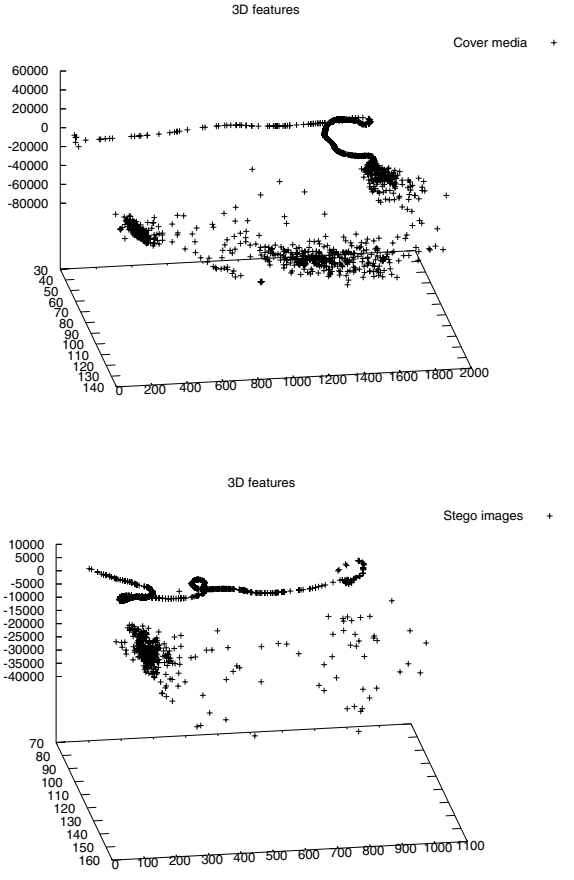


Fig. 8. Statistical vectors for A_2 projected onto $\mathbb{R}^3$. On the top $\mathcal{V}_c$, on the bottom $\mathcal{V}_s$.

Table 1. Optimal parameters for A_1 and A_2

	A_1	A_2
Metric	W^{-1}	W^{-1}
threshold	-0.2576	+5.2196
s	16	32
u	$\begin{pmatrix} +8.246617E-02 \\ -2.786796E-02 \\ -7.513114E-04 \\ +4.230118E-05 \\ +7.628112E-07 \\ -1.217546E+00 \\ +6.209087E-02 \end{pmatrix}$	$\begin{pmatrix} -6.075087E-02 \\ -3.839914E-04 \\ +4.421784E-04 \\ +2.525400E-06 \\ -5.790950E-08 \\ -3.763350E+00 \\ -4.005215E+00 \end{pmatrix}$

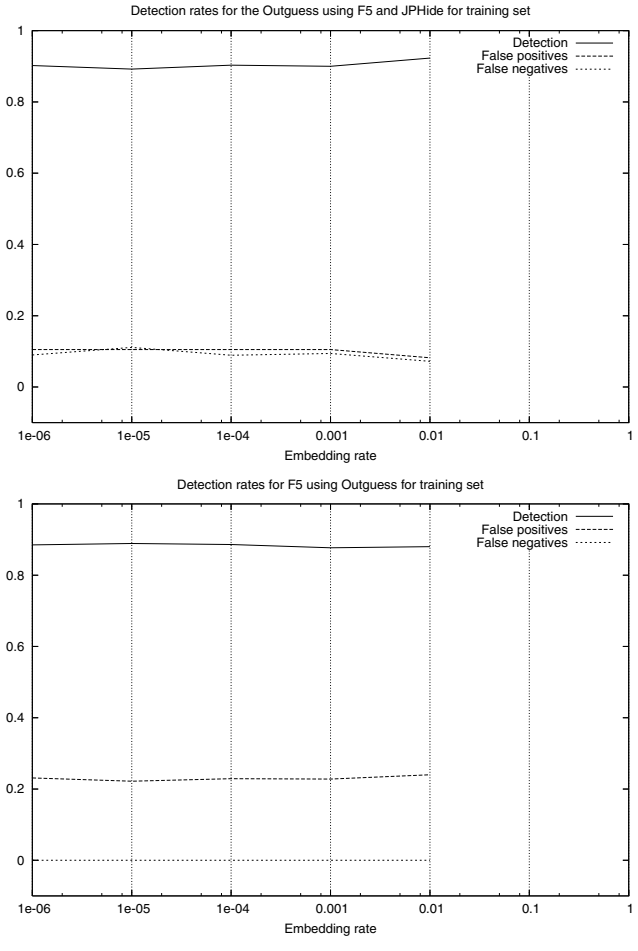


Fig. 9. Detection curves for D_1 detecting Outguess and D_2 detecting F5

3.3 Wild Detection Step

To show the efficiency of our scheme, we randomly generated challenge sets composed of 1,000 cover media and 1,000 stego images embedded with an embedding rate from 10^{-6} to 10^{-2}. After having trained two distinguishers D_1 and D_2, as explained in section 3.2, with respectively A_1 and A_2, we made them detect the use of new algorithms, Outguess for D_1 and $F5$ for D_2. The efficiency of D_1 and D_2 is presented in the figure 9. Two main conclusions can be drawn when observing these results. Firstly, D_1 and D_2 are able to detect efficiently the use of an algorithm which has not been used in the training set, that proves the universality property of our scheme. Nevertheless, this only proves that our detectors may detect images embeded with an unknown steganographic algorithm but with no confidence it will work with all unknown steganographic algorithm. Finally, the

detection rate appear to be constant and independent of the embedding rate, according to the hypothesis we made in section 2.2.

More precisely, we observed what follows.

- D_1 detects the use of Outguess with detection rate 90,4%, positive error rate 10% and negative error rate 9,12%.
- D_2 detect the use of F5 with detection rate 88,7%, positive error rate 22,5% and negative error rate less than 0,1%.

Obviously, these results depend on the distribution of cover media and stego images, but they give us a lower bound of the dectection rate. For both, the worste cases are obtained with sets only composed of cover media. So, for D_1, the detection rate is higher than 90% and for D_2 higher than $77,5\%$, whatever the distribution of cover media and stego images is.

4 Conclusion

We have proposed a new approach for universal JPEG steganalysis which is based on statistics of the compressed frequency domain and benefits from the statistical deviation of the entropy of the binary output stream. The avalanche criterion of the JPEG lossless compression step makes this deviation quasi-independent of the embedding rate and so, makes possible the design of steganographic detectors which the efficiencies do not depend on the payload. We design such a steganalyzer with very high and constant detection rates, as illustrated in section 3.3. The experimental results show that our steganalysis scheme is able to efficiently detect the use of new algorithms which are not used in the training step, even if the embedding rate is very low ($\approx 10^{-6}$).

Since universal detectors are less accurate but more general than specific ones, they are more oriented to detect the use of unknown steganographic systems or those for which no specific attack is known. Universal steganalysis provides another kind of detection services and should be run in parallel with specific detectors, for instance in a operational steganalytic system.

In future researches, we will try to improve the efficiency of our scheme using Support Vector Machines instead of Fisher discriminant. We hope to benefit from the non linearity of certain kernels and so increase our detection rates. This new universal scheme pointes out new statistics features we will combine to improve our specific steganalytic techniques. We are also working on combining detectors running in different domains.

Acknowledgments

We want to warmly thank Emmanuel Mayer for many useful advices on implementation and Franck Landelle for his accurate knowledge of statistics. Many discussions with them have greatfully improved the quality of this work. We also thank Emmanuel Bresson and Didier Alquié for their comments on the paper.

References

1. Provos, N.: Defending against statistical steganalysis. In: 10th USENIX Security Symposium. (2001)
2. Westfeld, A.: F5-a steganographic algorithm. [21] 289–302
3. Latham, A.: Steganography: JPHIDE AND JPSEEK (1999) http://linux01.gwdg.de/∼alatham/stego.html.
4. Avicibaş, I., Memon, N., Sankur, B.: Steganalysis based on image quality metrics. In: Proc. SPIE, Security and Watermarking of Multimedia Contents III. Volume 4314. (2001) 523–531
5. Farid, H.: Detecting hidden messages using higher-order statistical models. In: ICIP (2). (2002) 905–908
6. Lyu, S., Farid, H.: Detecting hidden messages using higher-order statistics and support vector machines. [20] 340–354
7. Lyu, S., Farid, H.: Steganalysis using color wavelet statistics and one-class support vector machines. In: Proc. SPIE, Security and Watermarking of Multimedia Contents VI. (2004)
8. Lyu, S., Farid, H.: Steganalysis using higher-order image statistics. IEEE Transactions on Information Forensics and Security (1) (2006)
9. Fridrich, J.J., Goljan, M., Hogea, D.: Steganalysis of jpeg images: Breaking the f5 algorithm. [20] 310–323
10. Fridrich, J., Goljan, M., Hogea, D.: New methodology for breaking steganographic techniques for jpegs. In: Proc. SPIE, Security and Watermarking of Multimedia Contents V. (2003) 143–155
11. Fridrich, J.J.: Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes. [22] 67–81
12. Fridrich, J., Pevny, T.: Multiclass blind steganalysis for jpeg images. In: Proc. SPIE, Security and Watermarking of Multimedia Contents VIII. (2006)
13. Harmsen, J.J., Pearlman, W.A.: Kernel fisher discriminant for steganalysis of jpeg hiding methods. In: ACM Multimedia and Security. (2005)
14. Lin, G.S., Yeh, C.H., Kuo, C.C.J.: Data hiding domain classification for blind image steganalysis. In: ICME. (2004) 907–910
15. Feistel, H.: Cryptography and computer privacy. Scientific American **228**(5) (1973) 15–23
16. Saporta, G.: Probabilité, analyse des données et statistiques (in french). Technip (1990)
17. Brown, C.W., Shepherd, B.J.: Graphics File Formats, reference and guide. Manning (1995)
18. Kalker, T., Cox, I.J., Ro, Y.M., eds.: Digital Watermarking, Second International Workshop, IWDW 2003, Seoul, Korea, October 20-22, 2003, Revised Papers. In Kalker, T., Cox, I.J., Ro, Y.M., eds.: IWDW. Volume 2939 of Lecture Notes in Computer Science., Springer (2004)
19. Chandramouli, R., Kharrazi, M., Memon, N.D.: Image steganography and steganalysis: Concepts and practice. [18] 35–49
20. Petitcolas, F.A.P., ed.: Information Hiding, 5th International Workshop, IH 2002, Noordwijkerhout, The Netherlands, October 7-9, 2002, Revised Papers. In Petitcolas, F.A.P., ed.: Information Hiding. Volume 2578 of Lecture Notes in Computer Science., Springer (2003)
21. Moskowitz, I.S., ed.: Information Hiding, 4th International Workshop, IHW 2001, Pittsburgh, PA, USA, April 25-27, 2001, Proceedings. In Moskowitz, I.S., ed.: Information Hiding. Volume 2137 of Lecture Notes in Computer Science., Springer (2001)

22. Fridrich, J.J., ed.: Information Hiding, 6th International Workshop, IH 2004, Toronto, Canada, May 23-25, 2004, Revised Selected Papers. In Fridrich, J.J., ed.: Information Hiding. Volume 3200 of Lecture Notes in Computer Science., Springer (2004)
23. Simmons, G.J.: The prisoners' problem and the subliminal channel. In: CRYPTO. (1983) 51–67
24. Wallace, G.K.: The jpeg still picture compression standard. Commun. ACM **34**(4) (1991) 30–44

Attack on JPEG2000 Steganography Using LRCA

Hwajong Oh, Kwangsoo Lee, and Sangjin Lee

Center for Information Security Technologies (CIST),
Korea University, Seoul, Korea
{ghost, kslee}@cist.korea.ac.kr, sangjin@korea.ac.kr

Abstract. In this paper, using a new steganalytic method, namely the left-and-right cube analysis (LRCA), we explore the steganography exploiting the JPEG2000 image as the carrier medium, where the steganography works by substituting message bits for the least significant bits (LSBs) of the quantized wavelet coefficients. For the achievement of the LRCA implementation on JPEG2000 images, we establish an appropriate sampling rule to extract the sample vectors from the image data, *i.e.,* the quantized wavelet coefficients, in the wavelet domain. In experiments, we used the LSB steganography and the bit-plane complexity segmentation (BPCS) steganography in order to generate the stego images. In results, the proposed method works well for both steganographic methods, and outperforms some other previous steganalytic methods.

1 Introduction

Recently, with the rapid development of information technology and the increase in internet usage, the digital media such as text, image, audio, and video are being widely used. Following the trend, steganography exploiting the digital media is fast becoming a controversial security issue. Steganography is something that allows one to insert secret messages inside the digital media, with its focus on not letting anyone other than the sender and the recipient to recognize its existence. Because the steganography can be abused in crimes, the way to counteract the effects of this technology, steganalysis, is new area of study in the upcoming 'Ubiquitous Age'. Steganalysis is a science to detect digital media with secret messages concealed in them.

The JPEG 2000 standard [7] is a new still image compression standard developed by the JPEG group. It is expected to replace the JPEG standard, which is now the most widely used image compression standard, in almost every sector - the internet, digital photos, mobiles, medical images and so on. There are not many steganographic methods proposed for the JPEG2000 image. A typical example would be the so-called BPCS steganography suggested by H. Noda et al. in [5]. The method basically takes a form of the well-known LSB steganography, that is in a way of substituting message bits for the least significant bits (LSBs) of the media samples, and specially uses namely the bit-plane complexity

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 268–281, 2006.

segmentation (BPCS) method to choose the media samples, *i.e.,* the quantized wavelet coefficients, for carrying message bits.

Recently, K. Lee et al. [6] proposed a novel steganalysis methodology, namely the left-and-right cube analysis (LRCA), which targets the LSB steganography. The LRCA uses the high-dimensional features of samples in digital media, and it is an epoch-making method in the effectiveness and accuracy. In the literature, regardless of what type was used for digital media, the LRCA algorithm was described in a general form on a given set of sample vectors as an input. This enables us to apply it to various kinds of digital media whenever we find an appropriate sampling rule to extract the sample vectors in accordance with the media type.

In this paper, using the LRCA algorithm, we explore the LSB steganography exploiting JPEG2000 images as carrier media. In order to achieve the LRCA implementation on JPEG2000 images, we establish an appropriate sampling rule to extract the sample vectors from the image data in the wavelet domain. For experiments, we used the LSB steganography and the BPCS steganography in order to generate the stego images. In results, the proposed method works well for both steganographic methods, and additionally, comparative results show that the proposed method outperforms some previous methods [2,3,8] that are applicable to the JPEG2000 images and use the first-order statistics of the quantized wavelet coefficients.

The order of the paper is as follows. In the second section of the paper, Preliminary, we will briefly explain the JPEG2000 image compression format and explain, equally briefly, the JPEG2000 LSB substitution steganography and the JPEG2000 BPCS steganography. Also we will be explaining the existent detection methods, the chi-square attack and Xiaoyi's method. In the third section we will explain the LRCA algorithm followed by the fourth section explanation on how to apply the LRCA on the JPEG2000 format. In the fifth section we will display the experiment results and conclude with the sixth section.

2 Preliminaries

2.1 The JPEG 2000 Standard

The JPEG 2000 standard is a wavelet-based image compression standard. It offers compression performance superior to the current standards (*e.g.,* JPEG) both at high bit rates and at low bit rates. It also provides many useful features like multiple resolution representation, progressive transmission by the pixel and resolution accuracy, lossless and lossy compression, region of interest (ROI) coding, etc., (refer to the literature [7] for detail). The JPEG 2000 encoding process is depicted in Fig. 1 (a). At the preprocessing step, the source image is decomposed into components and the image components are (optionally) divided into rectangular tiles. All samples in the image are dc level shifted and the color transform takes place. The forward inter-component transform step, which performs the core processing of the compression process, is done by the two-dimensional discrete wavelet transform (DWT). The DWT can be implemented by means of the

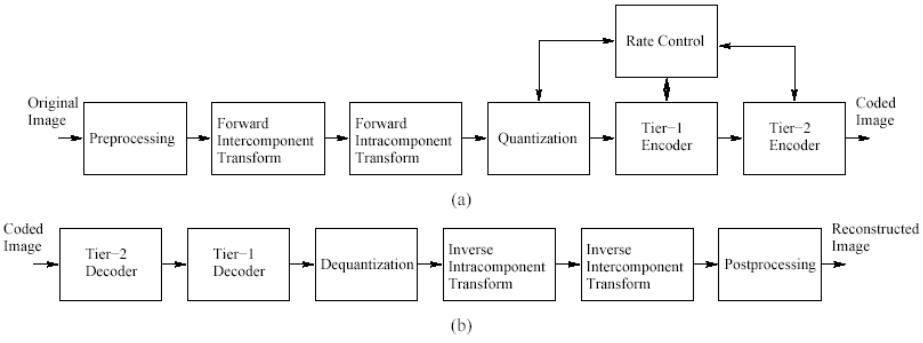


Fig. 1. Block diagram of the JPEG 2000 (a) encoder and (b) decoder

Daubechies 9/7 filter and Le Gall 5/3 filter which correspond to the irreversible transform and the reversible transform respectively. After transformation, all the DWT coefficients are quantized by the quantization step-sizes, where one quantization step-size is allowed per subband and for each subband the quantization step size is decided by the dynamic range of the subband. For reversible (lossless) compression, the quantization step-size needs to be one. In the following steps, the subbands of quantized coefficients are gathered into code-blocks, and the bit planes in a code block are entropy coded by means of an arithmetic coding, *i.e.,* the MQ coding supported by the standard. The bit-stream organizer makes the final bit stream according to the calculated distortion rate using a bit-rate control method. The JPEG 2000 decoding process is the reverse procedure of the encoding process and it is depicted in Fig. 1 (b).

2.2 LSB Steganography for JPEG2000

The LSB steganography works by substituting message bits for the least significant bits (LSBs) of samples in digital media. It is not only simple and fast, but also able to adopt various types of digital media as carrier media. For the security concern, however, it should be careful to choose samples for carrying message bits. Basically, the message-carrying samples should be randomly selected, and besides, the characteristics of the media type should be considered in the selection. In case of JPEG2000 images, we should consider the subband characteristics. Fig. 2, for example, displays the process of two-level wavelet decomposition for the well-known 'Barbara' image and the typical subband structure of R-level wavelet decomposition. The LL_0 band, which is shown at the top-left position of the figure, contains global and very important information of the image content. So, the change in wavelet coefficient values of the LL_0 band makes a very reactive distortion when reconstructed, and this happens even for a small change. Thus, the LL_0 band should be excluded from the embedding process of steganography. On the other hand, other subbands contain detailed information of the image content, about the variation in horizontal, vertical, and

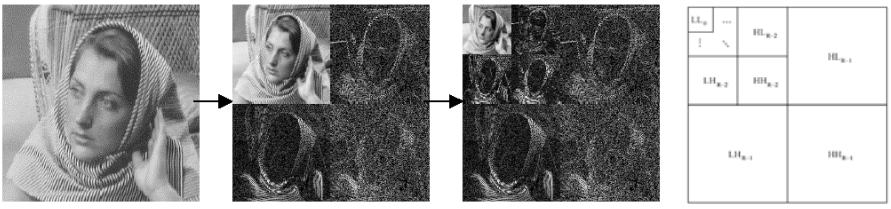


Fig. 2. An example of the two-level wavelet decomposition and its typical structure

diagonal directions according to their levels; the higher the level is, the more detailed the information is. One can see that the HH_{R-1} band, which is shown at bottom-right position of the figure, have many coefficient values placed on zero. So it should be also excluded from the embedding process because the use of the HH_{R-1} band could be easily detected by simply counting the coefficients of the zero value.

2.3 The BPCS Steganography for JPEG2000

The bit-plane complexity segmentation (BPCS) steganography was proposed by E. Kawaguchi and R. O. Eason [1]. It was designed to embed a big amount of secret messages in an uncompressed image on the spatial domain by utilizing the human visual system (HVS). Their core idea is to substitute message bits for the visually complex blocks in the decomposed bit-planes of an image. The embedding process of the BPCS steganography can be summarized as follows: The source image is decomposed into bit-planes and the bit-planes are divided into small rectangular blocks. For each binary block, the block complexity is measured by counting the number of changes in binary values of two samples which are horizontally or vertically adjacent. And then the binary blocks can be classified into "informative" or "noise-like" blocks by means of a threshold α. The message bit stream is divided into blocks of same size, and for these message blocks, the block complexities are measured in the same way. The message blocks showing lower complexities than the threshold α are conjugated by the exclusive-or (XOR) operation with a checkerboard pattern, and make a conjugation map to be required for the decoding process of the hidden message. Finally, the conjugation map is inserted usually in the first noise-like block of the carrier image and the preprocessed message blocks are substituted for the noise-like blocks; the image blocks used for the embedding process are selected sequentially or randomly. H. Noda et al. applied this method to the JPEG2000 format [5]. Their method embeds a secret message in the bit-planes of the quantized wavelet coefficients and the embedding process is similar to that of the original BPCS steganography. In order to assess the detection reliability of the proposed method, although all the bit-planes can be used in the BPCS steganography, we will only use the LSB plane of the subbands excluding the LL_0 band and the LL_{R-1} band to generate the stego images in our experiments.

3 Principle of Left-and-Right Cube Analysis

In this section, we give a brief review of the left-and-right cube analysis (LRCA) [6].

3.1 The Sample Vector Space

The digital media can be represented by the succession of samples $s_1, s_2, \cdots, s_N$ whose values s_i are integer values. A *sample vector* means a n-tuple $(s_{i_1}, s_{i_2}, \cdots, s_{i_n})$, where $1 \le i_j \le N$. The sample vectors are used as basic units for the LRCA. Let S be the set of sample vectors drawn from a digital media. (We will come back to the issue of how to draw the sample vectors from the JPEG2000 image in next section.) A sample vector match a point of the n-dimensional discrete vector space Z^n, and the set S matches a subset of Z^n. A function $f_S : Z^n \to \{0, 1\}$, called the *state function with* S, is defined by

$$f_S(p) = \begin{cases} 1 & \text{if } p \in S \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{1}$$

for every $p \in Z^n$. A point $p \in Z^n$ is said to be filled with S if $f_S(p) = 1$, and the point p is said to be empty with S otherwise.

Let $\mathcal{F}$ be the collection of subsets of Z^n. For a set of sample vectors S, a measure $\gamma_S : \mathcal{F} \to [0, \infty]$, called the *the complexity measure with* S, is defined by

$$\gamma_S(A) = \sum_{s \in A} f_S(s) \ . \tag{2}$$

for every $A \in \mathcal{F}$. A set $A \in \mathcal{F}$ is said to be m-complex with S if $\gamma_S(A) = m$. From the definition, it follows that $0 \le \gamma_S(A) \le |A|$ for every $A \in \mathcal{F}$.

For a positive integer δ, the *δ-cube* is defined by the set of the form,

$$Q(a; \delta) = \{(\sigma_1, \cdots, \sigma_n) \in Z^n \colon \ \sigma_i = \alpha_i \ \text{ or } \ \sigma_i = \alpha_i + \delta, \ 1 \le i \le n\} \ . \tag{3}$$

Here, $a = (\alpha_1, \cdots, \alpha_n) \in Z^n$. Given a set of sample vectors S, the δ-cubes can be classified by their complexities with S. Since every δ-cube has 2^n points, it can show 2^{2^n} patterns and have 2^n-complexity to the maximum (see Fig. 3). Let Ω_δ be the collection of δ-cubes in Z^n, and let $\Omega_\delta(m)$ be the sub-collection of Ω_δ that consists of m-complex δ-cubes with S. Then, the collection Ω_δ is partitioned into the sub-collections $\Omega_\delta(m)$:

$$\Omega_\delta = \Omega_\delta(0) \cup \Omega_\delta(1) \cup \cdots \cup \Omega_\delta(2^n) \ . \tag{4}$$

3.2 LR Cube Model

From now on we will assume that the δ value is fixed to a positive *odd* integer. Let P be the set of the points a of Z^n whose coordinates α_i are *even* integers. For a point $a \in P$, the set

$$Q_L(a; \delta) = \{s \in Z^n : \ \sigma_i = \alpha_i \ \text{ or } \ \sigma_i = \alpha_i - \delta, \ 1 \le i \le n\} \tag{5}$$

is called the *left δ-cube with corner at* a, and the set

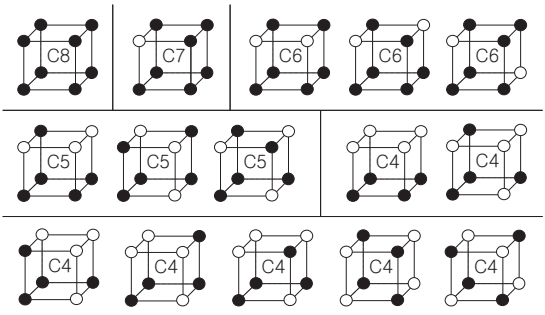


Fig. 3. Pattern inventory of δ-cubes in Z^3 [6]

$$Q_R(a;\delta) = \{s \in Z^n : \ \sigma_i = \alpha_i \ \text{ or } \ \sigma_i = \alpha_i + \delta, \ 1 \le i \le n\} \tag{6}$$

is called the *right δ-cube with corner at a* (see Fig. 4).

It is clear that every point of Z^n is contained in exactly one of the left δ-cubes. So the left δ-cubes are pairwise disjoint and cover the lattice space Z^n. The same is true for right δ-cubes:

$$Z^n = \dot{\bigcup}_{a\in P} Q_L(a;\delta) \text{ and } Z^n = \dot{\bigcup}_{a\in P} Q_R(a;\delta) \ . \tag{7}$$

Let $\mathcal{L}_\delta$ be the collection of left δ-cubes with corners at P, and let $\mathcal{R}_\delta$ be the collection of right δ-cubes with corners at P:

$$\mathcal{L}_\delta = \{Q_L(a;\delta) : a \in P\} \text{ and } \mathcal{R}_\delta = \{Q_R(a;\delta) : a \in P\} \ . \tag{8}$$

For a set of sample vectors S, let $\mathcal{L}_\delta(m)$ be the sub-collection of $\mathcal{L}_\delta$ that consists of the left m-complex δ-cubes with S, and let $\mathcal{R}_\delta(m)$ be the sub-collection of $\mathcal{R}_\delta$ that consists of the right m-complex δ-cubes with S. Then $\mathcal{L}_\delta$ and $\mathcal{R}_\delta$ are partitioned as follows:

$$\mathcal{L}_\delta = \mathcal{L}_\delta(0) \cup \cdots \cup \mathcal{L}_\delta(2^n) \text{ and } \mathcal{R}_\delta = \mathcal{R}_\delta(0) \cup \cdots \cup \mathcal{R}_\delta(2^n) \ . \tag{9}$$

The LR cube analysis assumes that the following statements are hold:

LR cube assumption: Given a cover-signal I, there exist a set of sample vectors S drawn from I and a positive odd integer δ such that

$$E\Big[\Big|\mathcal{L}_\delta[m]\Big|\Big] = E\Big[\Big|\mathcal{R}_\delta[m]\Big|\Big] \ , \tag{10}$$

for all $m = 1, \cdots, 2^n$. Here $E(\cdot)$ represents the expected value.

The LR cube assumption implies that a cover-signal I includes a set of sample sequence S with which the left and right δ-cubes show a similar complex levels for some positive odd δ.

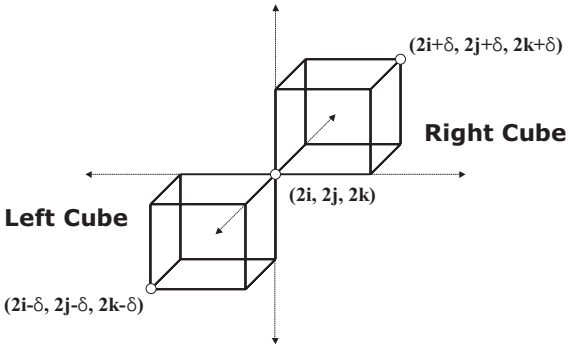


Fig. 4. The 3-dimensional left and right δ-cube [6]

3.3 Statistical Analysis

The LRCA is to measure the difference between the two distributions of the left δ cubes and the right δ cubes. We assume that a set of sample vectors S is given and the δ value is set by a positive odd integer. One can induce the state function $f_S(\cdot)$ with S by Eqn. (1), and accumulate the collections $\mathcal{L}_\delta - \mathcal{L}_\delta(0)$ and $\mathcal{R}_\delta - \mathcal{R}_\delta(0)$ of left and right δ-cubes with corners at P that are not 0-complex. Then one can partition the collections $\mathcal{L}_\delta - \mathcal{L}_\delta(0)$ and $\mathcal{R}_\delta - \mathcal{R}_\delta(0)$ into $L_\delta(m)$ and $R_\delta(m)$ with the complexities $m = 1, 2, \cdots, 2^n$, respectively, and can obtain the two distributions $|\mathcal{L}_\delta(m)|$ and $|\mathcal{R}_\delta(m)|$ for $m = 1, 2, \cdots, 2^n$.

They use a χ^2-test to determine whether the set of sample vectors S shows the distortion of the two distributions of the left and right δ-cubes. The expected distribution $E^*_\delta(m)$ for the χ^2-test can be computed from the two distribution. We assume that the two distributions are similar for a cover signal. As the results, we can take the arithmetic mean,

$$E^*_\delta(m) = \frac{|\mathcal{L}_\delta(m)| + |\mathcal{R}_\delta(m)|}{2} \ , \tag{11}$$

to determine the expected distribution. The expected distribution is compared with the observed distribution

$$E_\delta(m) = |\mathcal{L}_\delta(m)| \ . \tag{12}$$

The χ^2 value for the difference between the two distributions is given as

$$\chi^2 = \sum_{m=1}^{\nu+1} \frac{(E_\delta(m) - E^*_\delta(m))^2}{E^*_\delta(m)} \ , \tag{13}$$

where ν is the degrees of freedom, that is, the maximal complexity of the left and right δ-cubes minus one. The p-value p is then given by the cumulative distribution function

$$p = \int_0^{\chi^2} \frac{t^{(\nu-2)/2} e^{-t/2}}{2^{\nu/2}\Gamma(\nu/2)} dt \ , \tag{14}$$

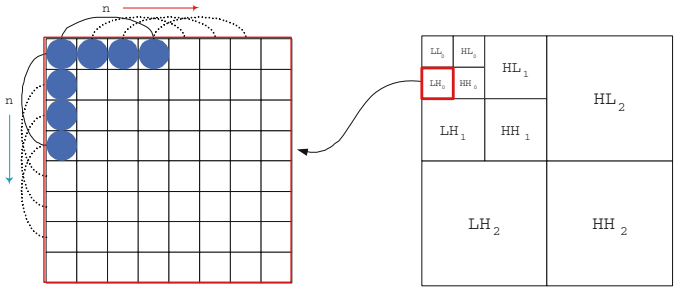


Fig. 5. LRCA Sampling method description for the JPEG2000 format

In the application of LRCA to detection of hidden messages, the p-value will be used as discrimination between cover-signals and stego-signals: for a given decision threshold T_p, if $p > T_p$, then the signal I will be regarded as a stego-signal.

4 Applying the LRCA to JPEG2000 Format

In this section we explain the sampling method to apply the LRCA algorithm to the JPEG2000 format. For the ease of description, we only consider the the gray-scale JPEG2000 image. This can be surely expanded to be used in the multi-channel case.

In the JPEG2000, when the original image is two-dimensional wavelet transformed, in each sub-band not only has the coefficients which not only contains the spatial information, but the frequency substance. In the LL_0 band, the coefficients which are the smaller version of the original image is saved, and in other sub-bands not only contain a high frequency substance but the coefficient values that contain the spatial information. That is why the wavelet sub-band is considered a separate image which contains the original spatial information. Working from here, we applied the similar sampling method used in LR cube analysis to the JPEG2000 format [6]. to apply a new method and use the grey image. This method uses diverse information about adjacent pixels like the image spatial correlation. It uses n-dimensional vector whose elements are successional adjacent n wavelet coefficients at horizontal and vertical direction as a sample vector, like Fig. 5.

In a formalized description, let $g(i, j)$ denote an $N \times M$ sub-band, where $i = 0, 1, \cdots, N-1, j = 0, 1, \cdots, M-1$. Given a dimension n of sample vectors, let $s_v(i, j)$ denote the vertical sample vector at the coefficient position (i, j), where the components $\sigma_k = g(i + k - 1, j)$ for $k = 1, 2, \cdots, n$. Then, the set of vertical sample vectors S_v is the set of the form,

$$S_v = \{s_v(i, j) : 0 \leq i \leq N - n \;\; and \;\; 0 \leq j \leq M - 1\} \tag{15}$$

In the similar way, let $s_h(i, j)$ denote the horizontal sample vector at the coefficient position (i, j), where the components $\sigma_k = g(i, j+k-1)$ for $k = 1, 2, \cdots, n$.

Then the set of horizontal sample vectors S_h is the set of the form,

$$S_h = \{s_h(i,j) : 0 \le i \le N-1 \ \ and \ \ 0 \le j \le M-n\} \tag{16}$$

Their union set $S = S_v \cup S_h$ is the set of n-dimensional sample vectors extracted form sub-band $g(\cdot,\cdot)$. and In the similar way, we use the whole sample vectors extracted from the whole sub-band embedded the secret message as the set of n-dimensional sample vectors for LR cube analysis.

5 Experimental Result

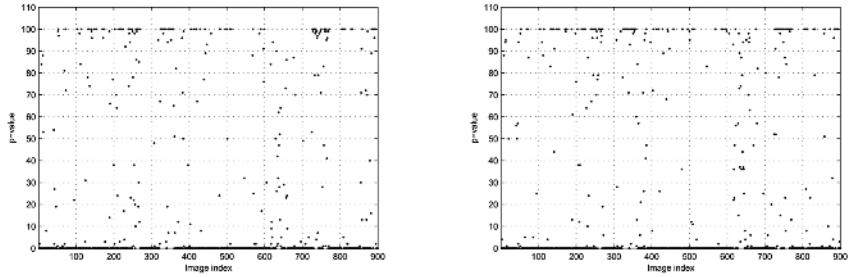
For the experiment we used JASPER ver 1.700, the official software of the JPEG2000 committee, as the JPEG2000 encoder. As an steganographic embedding tool, we implemented the JPEG2000 LSB substitution steganographic embedding tool (J2STEG.exe) and the JPEG2000 BPCS steganographic embedding tool (J2BPCS) using JASPER code. Similarly, as a steganalysis tool, the JASPER code was remodeled to implement the chi-square attack, the method by X. Yu et al., and the J2LRCA tool(our method). As an experiment image, we used 963 jpg images of the size 512×512 in the CBIR image database in Washington university. Firstly, we down-scaled the 963 jpg images into grey scale, and with the JASPER encoder, compressed these and made jp2 images, and then with the J2STEG and the J2BPCS we embedded random message with diverse embedding rates to make a stego-image. While embedding there were no embedding in the LL_0 band and the top frequency bands. In the BPCS steganography the block sizes were fit at 4×4, and was used only in the LSB bit-plane. Table 1 shows total cover-images and stego-images made for the experiment.

Table 1. Cover and stego images for experiments

Image	Embedding capacity(bpp)/(byte)		Threshold	Volume
Cover	·	·	·	963
LSB Stego	0.10	818	·	963
	0.20	1637		963
	0.30	2455		963
BPCS Stego	0.15	1228	15	963
	0.25	2046	14	963
	0.40	3274	13	963

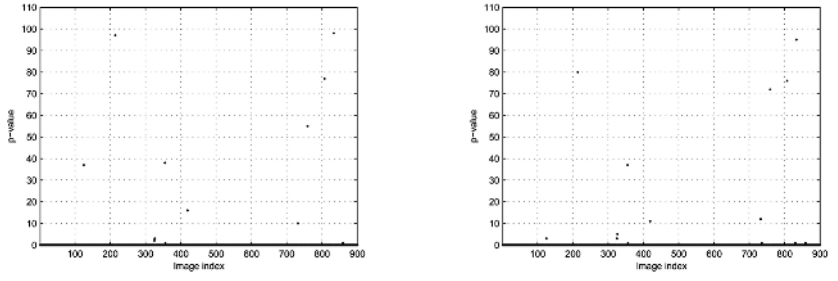
5.1 The Result of the Chi-Square Attack

Fig. 7 is the result of the original chi-square attack. Fig. 7 (a) is the result for 963 stego-images with 30% secret messages inserted using JPEG2000 LSB steganography, and Fig. 7 (b) is the result for 963 stego-images with 40% secret message



(a) LSB 30% embedded (b) BPCS 40% embedded

Fig. 6. The result of the chi-square attack



(a) LSB 30% embedded (b) BPCS 40% embedded

Fig. 7. The result of the chi-square attack

inserted. We can observe that the outputted p-value is very erratic, ranging from 0 to 100, not being detected properly. In both cases the secret messages were inserted randomly so that the original chi-square attack could not detect the stego image. As the result of the experiment we could discovered that with the original chi-square attack method, only the stego-images with a secret message embedding rate of over 80% could be properly detected. Also, the extended chi-square attack result showed an irregular p-value according to the area it was sampled. The PoVs value of the observed wavelet coefficients have quite similar frequencies, so despite the fact that it was an cover-image, the result of the extended chi-square attack showed a high p-value, which was a false positive detection. The result of the experiment showed that the extended chi-square attack outputted a high p-value for most images and could not detect the stego-image.

5.2 The Result of the X. Yu et al.'s Method

Fig.7 is the result of the X. Yu et al.'s method. Fig.7 (a) is the result for 963 stego-images with 30% secret messages embedded using JPEG2000 LSB steganography, and Fig.7 (b) is the result for 963 stego-images with 40% secret message embedded. Most were not detected, with the p-value being 0. As a

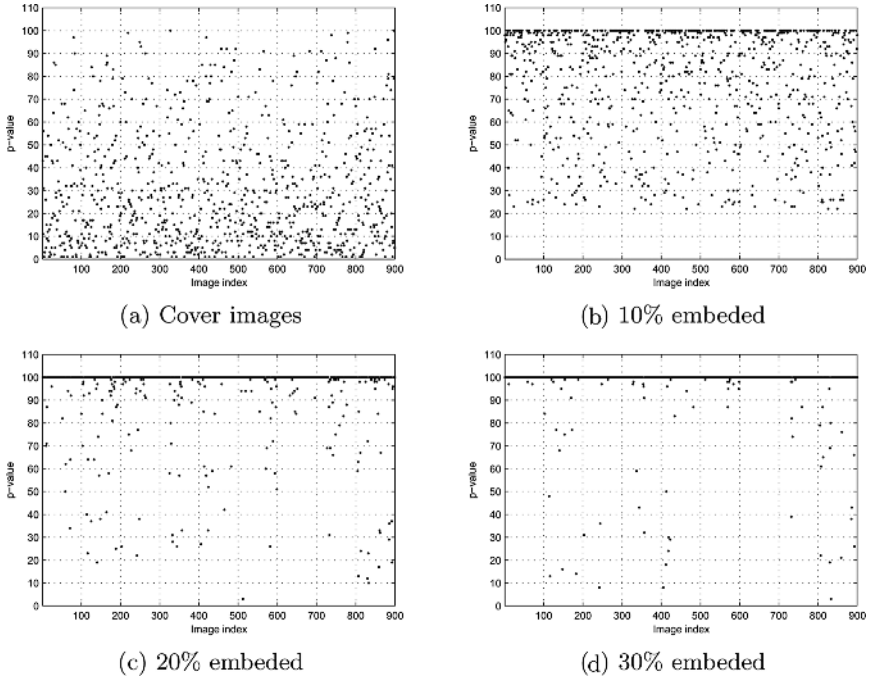
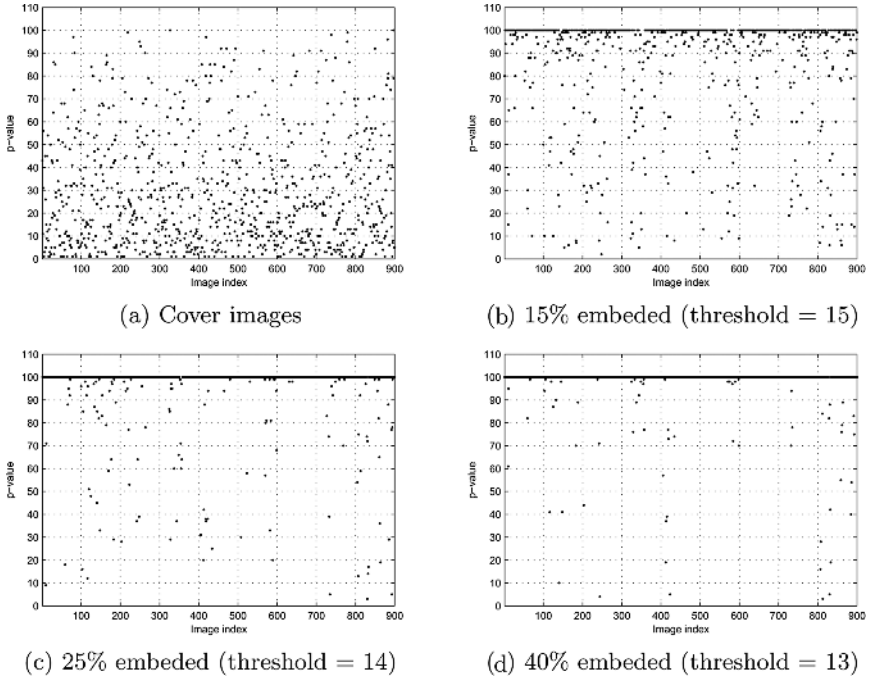


Fig. 8. LRCA test result for JPEG2000 LSB stego images images

result of the experiment it was clear that even though a large number of secret messages were embedded, the X. Yu et al.'s method could not faithfully detect the stego-image. This is because the author of this method assumed that, if a secret message is embedded, the symmetry with 0 as its axis, the distinct feature of wavelet coefficients histogram, would disappear, but it didn't. When a secret messages is embedded using the LSB substitution type steganography, only the frequency of the adjacent coefficients change, without any changes in the symmetry.

5.3 The Result of the LRCA

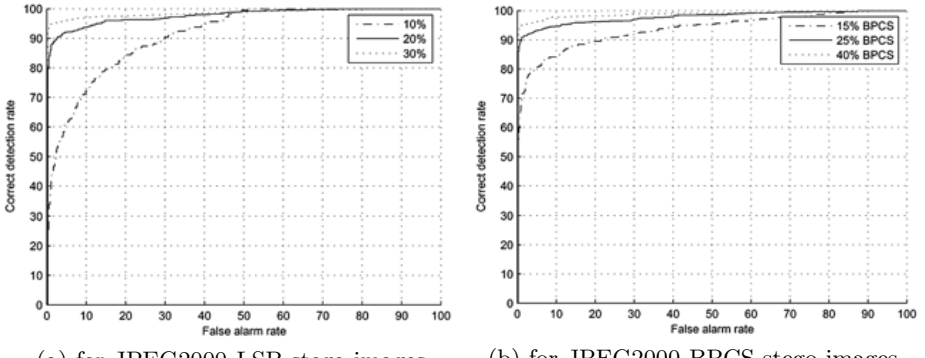
Fig. 8 is a experimental result of the LR cube analysis for stego-images made by the JPEG2000 LSB substitution steganography. We used a 5-dimensional sampling and a δ value 5 as the parameters of LR cube analysis. From many diverse experiments with different dimension and δ values, the above parameters showed the best results. The higher the dimensions for the sample vector, the higher the complexity of LR cube, and it displayed well the special feature of the cover image and stego-image, with 5-dimensions as its maximum. But dimensions more than 7 had too much information scattered, and those special features of the cover-image and the stego-image were gone. As a result of the experiment, it



(a) Cover images (b) 15% embeded (threshold = 15)

(c) 25% embeded (threshold = 14) (d) 40% embeded (threshold = 13)

Fig. 9. LRCA test result for JPEG2000 BPCS stego images



(a) for JPEG2000 LSB stego images (b) for JPEG2000 BPCS stego images

Fig. 10. ROC curve for LRCA test result

reacted to a stego-image with the secret messages embedding rate of lower than 10%, and in an stego-image with more than 30% most outputted a p-value of near a 100. Fig. 10 (a) is the graph of the ROC curve, which is drawn from the result above. As you can see in the graph, stego-images of 20%, 30% showed a highly successful detection rate even in a very low false positive detection rate. Also, it showed a relatively high detection rate in stego-images of 10%. Fig. 9 shows

a LRCA experiment result with the stego-images made by JPEG2000 BPCS steganography. So, as the experiment results for the JPEG2000 LSB substitution steganography, we can see that it detects well. Fig. 10 (b) is a graph of the ROC curve of this result. Even stego-images with the secret messages embedding rate of 15% were detected fairly well.

6 Conclusion

In this paper we applied the LRCA algorithm to the JPEG2000 format, the new image compression standard. For this we suggested sampling method that would best suit the JPEG2000 format. This sampling method uses the correlation between adjacent wavelet coefficients. We experimented on the JPEG2000 LSB steganography, the JPEG2000 BPCS steganography, but these cannot be detected by the existent steganalytic methods, the chi-square attack and X. Yu et al.'s method. In an experiment of LR cube analysis, despite the low 10% embedding rate of secret messages, it showed a remarkable detection ability. In conclusion, the LSB substitution steganography is dangerous in the wavelet domain, and the BPCS steganography, a sort of the LSB substitution steganography, is not safe either.

Acknowledgements

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

References

1. E. Kawaguchi and R. O. Eason, "Principle and Appications of BPCS Steganography," *Proceedings of SPIE*, vol. 3528, 1998.
2. A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," *Information Hiding: 3rd International Workshop,* A. Pfitzmann (Ed.): IH'99, LNCS 1768, pp. 61–76, Springer-Verlag Berlin Heidelberg 2000.
3. N. Provos and P. Honeyman, "Detecting Steganographic Content on the Internet," *CITI Technical Report 01–11,* 2001.
4. A. Westfeld, "Detecting Low Embedding Rates," *Information Hiding: 5th International Workshop,* F.A.P. Peticolas (Ed.): IH2002, LNCS 2578, pp. 324–339, 2003, Springer-Verlag Berlin Heidelberg 2003.
5. H. Noda, J. Spaulding, M. N. Shirazi, and E. Kawaguchi, "Application of Bit-Plane Decomposition Steganography to JPEG2000 Encoded Images," *IEEE Signal Processing Letters,* vol. 9, 2002.
6. K. Lee, C. Jung, S. Lee, and J. Lim, "New Steganalysis Methodology: LR Cube Analysis for the Detection of LSB Steganography," *Information Hiding: 7th International Workshop,* Mauro Barni et al. (Eds.): IH 2005, LNCS 3727, pp. 312–326, 2005, Springer-Verlag Berlin Heidelberg 2005.

7. "The JPEG-2000 Still Image Compression Standard," Michael Adams, ISO/IEC JTC 1/SC 29/WG 1N 2412, 2001.
8. X. Yu, T. Tan, and Y. Wang, "Reliable Detection of BPCS-Steganography in Natural Images," *Proceedings of Third International Conference on Image and Graphics,* ICIG'04, pp. 333–336. z
9. CBIR Image Database, University of Washington, `http://www.cs.washington.edu/research/imagedatabase/groundtruth` .

A Low-Cost Attack on Branch-Based Software Watermarking Schemes

Gaurav Gupta and Josef Pieprzyk

Centre for Advanced Computing - Algorithms and Cryptography,
Department of Computing, Division of Information and Communication Sciences,
Macquarie University, Sydney, NSW - 2109,
Australia
{ggupta, josef}@ics.mq.edu.au
http://www.comp.mq.edu.au/∼ggupta

Abstract. In 2005, Ginger Myles and Hongxia Jin proposed a software watermarking scheme based on converting *jump instructions* or *unconditional branch statements* (UBSs) by calls to a *fingerprint branch function* (FBF) that computes the correct target address of the UBS as a function of the generated fingerprint and integrity check. If the program is tampered with, the fingerprint and integrity checks change and the target address will not be computed correctly. In this paper, we present an attack based on tracking stack pointer modifications to break the scheme and provide implementation details. The key element of the attack is to remove the fingerprint and integrity check generating code from the program after disassociating the target address from the fingerprint and integrity value. Using the debugging tools that give vast control to the attacker to track stack pointer operations, we perform both subtractive and watermark replacement attacks. The major steps in the attack are automated resulting in a fast and low-cost attack.

Keywords: software, watermark, unconditional branch, breakpoint.

1 Introduction

In recent years, watermarking and fingerprinting have gathered significant attention due to the growing concerns over digital piracy and forgery of multimedia documents including software. *Fingerprinting* and *Software Authentication* are two major security aspects that have emerged. While the former is related to preventing illegal distribution, the latter tries to ensure that the software has not been tampered with. Numerous models have been proposed with these objectives, embedding watermarks, fingerprints, and integrity checks in the source codes and/or executable codes. Software watermarking schemes can be classified as follows:

– *Graph-based/branch-based software watermarking*: The software is treated as a graph G_s with sequential blocks of code as nodes and transfer instructions such as function calls and branch statements as edges connecting the nodes.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 282–293, 2006.

The watermark is a separate code and realized as a graph G_w. The two graphs G_s and G_w are connected by inserting additional edges (implemented as branch statements). The resulting watermarked graph is $G_{s'} = G_s + G_w$ and source code s' is decoded from $G_{s'}$.

Venkatesan et al. [14] proposed the first graph-based software watermarking scheme. The central idea is to convert the software and the watermark code into digraphs and add new edges between the two graphs implemented by adding function calls between the software and watermark code. This scheme lacks error-correcting capabilities and is susceptible to re-ordering of instructions and addition of new function calls. Another problem in the scheme is that the random walk mentioned in the paper (refers to the next node to be added in the watermarked software graph being selected randomly from the software graph and the watermark graph) is not actually *random*. The node visited next is based on the number of remaining nodes belonging to software graph N_s and the number of remaining nodes belonging to watermark graph N_w. The next node is chosen from the watermark nodes with a probability of $\frac{N_w}{N_w+N_s}$ and from the software nodes with a probability of $\frac{N_s}{N_w+N_s}$. In a typical scenario, $N_s \gg N_w$, hence the watermark is skewed towards the tail of the watermarked program. This information is useful for probabilistic attacks. Alternatively, a pseudo-random permutation of the nodes to be visited can be generated. For further literature in graph-based software watermarking, the reader is referred to [1,2,3,4,13]. None of these schemes are completely secure against instruction and block re-ordering attacks.

- *Register-based software watermarking*: Registers used to store variables are changed depending on the watermark bit to be embedded by replacing higher level language code with an inline assembly code. The attacker intends to re-allocate variables in registers if the watermark has to be removed. Though, no such attack has yet been proposed.

  Register-based software watermarking based on the QP algorithm (named after authors Qu and Potkonjak) [10,11] is presented in [7]. It modifies registers used to store variables depending on which variables are required at the same time. The scheme is susceptible to register re-allocation attacks. A secondary watermark destroys the old watermark and inserting bogus methods renders the original watermark useless by changing the interference graph.
- *Thread-based software watermarking*: Nagra et al. [9] propose encoding the watermark in the sequence of the threads that are executed. For example, there are 3 threads; T_1, T_2, T_3, $T_1 \rightarrow T_2 \rightarrow T_3$ encodes watermark $(000)_2$ and $T_1 \rightarrow T_3 \rightarrow T_2$ encodes watermark $(001)_2$ and so on. However, without any additional error-control mechanism, changing threads that execute piece of a code would destroy the watermark. Again, there has been no scheme claiming to break the watermark using suggested approach.
- *Obfuscation-based software watermarking*: This class of watermarking is applicable to object-oriented softwares. Class C with functions $\{f_1, f_2, \ldots, f_n\}$ is partitioned into k subclasses $\{C_1, C_2, \ldots, C_k\}$ and the watermark is

encoded in the allocation of the functionalities Examples of such proposed schemes are [5,6,12].

This paper is organized as follows. Section 2 addresses related work in branch-based software watermarking and Section 3 describes watermarking scheme of Myles and Jin that we propose to attack. This is followed by a description of our attack in Section 4. Section 5 provides implementation details and results. We conclude our paper with a note on future enhancements in Section 6.

2 Related Work

There have been several research projects dealing with *branch-based* software watermarking. These schemes exploit possibilities to modify the program's execution path by altering branch statements. In this section, we discuss two papers closely related to our attack. The first by Collberg et al. [1] that introduces *Branch Functions*. The second paper is by Myles and Jin [8] and describes the watermarking scheme that we attack in this paper.

Collberg et al. introduce the notion of *Branch Function* [1]. *Jump instructions* or *unconditional branch statements* (UBSs) are replaced by calls to the *branch function* (for the sake of consistency, by *branch*, we mean an unconditional branch statement from now on) and modifies its own return address in order to return the control to the target of the branch statement. Figure 1 illustrates this process. If the program contains a jump instruction from l_{begin} to l_{end}, several intermediate *pit stops* are inserted so that the control-flow graph becomes $l_{begin} \rightarrow a_1 \rightarrow a_2 \rightarrow \ldots \rightarrow l_{end}$ such that l_{begin} has a jump instruction to a_1 which has a jump instruction to a_2 and so on. The pit stops are inserted using the rule:

$address(a_i) < address(a_{i+1})$, if watermark bit $w_i = 1$
$address(a_i) > address(a_{i+1})$, if watermark bit $w_i = 0$

Finally all the jump instructions are replaced by call to the branch function that determines the correct target address based on the calling address and returns the control to it.

Obvious attacks on such a scheme are adding an additional pit stop to the chain $l_{begin} \rightarrow a_1 \rightarrow a_2 \rightarrow \ldots \rightarrow l_{end}$ such that it becomes $l_{begin} \rightarrow a_1 \rightarrow a_{extra} \rightarrow a_2 \rightarrow \ldots \rightarrow l_{end}$ or deleting an existing pit stop such that it becomes $l_{begin} \rightarrow a_2 \rightarrow \ldots \rightarrow l_{end}$. The goal is to disturb the chain (thereby modify the watermark) yet keep the origin and target the same (hence keeping the execution path intact). Making similar changes, inserting secondary watermark is a trivial.

Myles and Jin propose an alternative fingerprinting model in [8]. The underlying concept remains the same, that is, a *branch function* transferring control to the target of the UBS, but in this case, the branch function contains the fingerprint-generating code, hence the name *Fingerprint Branch Function* (FBF). FBF also computes an integrity check on the source code to ensure that

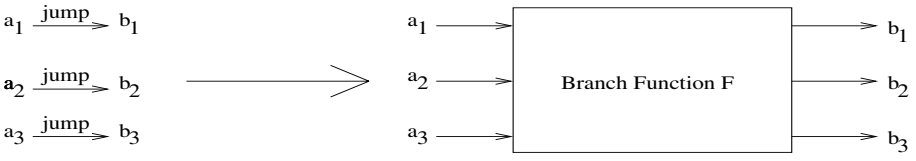


Fig. 1. Insertion of branch function F that changes the return address according to the calling address and transfers control to target of the UBS. The *jump instructions* are now replaced by calls to F.

it is not modified. In the following section, we discuss this scheme in detail and analyze its flaws and weaknesses.

3 Discussion on Watermarking Scheme

The watermarking scheme is applied to software containing *branch statements*. These statements are then replaced by calls to FBF which returns control to the target address. The target address is generated from a recursive process of deriving new keys from old keys and checking the program for integrity. Additionally, an integrity check branch function (ICBF) is inserted in the program that verifies the integrity of FBF. If the user manipulates the program, the keys derived and integrity check value would change and hence the target address will change. The modified target address can be valid (belonging to code section of the program) which will result in incorrect execution of the program. Alternatively the target address can be invalid (lying outside the code section) resulting in runtime error. We will now discuss the two algorithms in the scheme, *"embed"* that inserts the watermark in the software and *"recognize"* that extracts the watermark from the watermarked software. They are defined as:

1. embed$(P, AM, key_{AM}, key_{FM}) \rightarrow P', FM$
2. recognize$(P', key_{AM}, key_{FM}) \rightarrow AM, FM$

where

- P is the original software,
- AM is the authorship mark,
- key_{AM} is the secret input sequence to generate a trace of the program used to embed the watermark - the same for all copies of watermarked software,
- key_{FM} is an initial secret key for deriving further keys - different for each copy of the watermarked program,
- FM is the fingerprint mark
- P' is the watermarked software

3.1 Watermark Embedding

The steps involved in the ***embed*** algorithm are:

1. Let α be the set of all functions in P. Run the program with a secret input sequence key_{AM}.

2. Obtain set F of functions that lie in the execution path when the program is run with input sequence key_{AM}, let $\beta = \alpha - F$.
3. The number of UBSs in functions that belong to F is n and the number of UBSs in functions from β is m.
4. Insert the two integer arrays; T of size n and R of size m in the data section of the program.

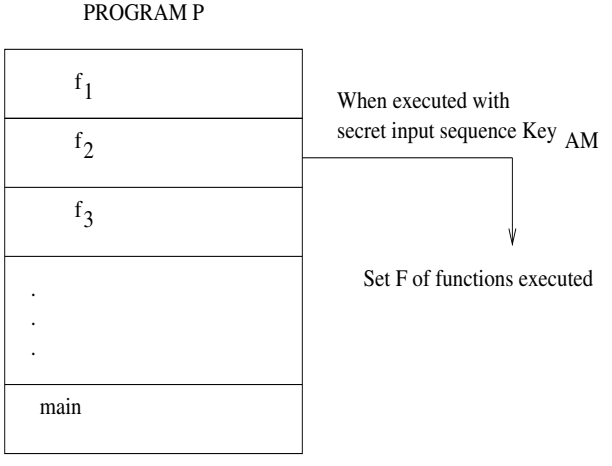


Fig. 2. The set of functions F is executed when the program P is run with the secret input key_{AM}

5. Compute displacement d_i between source s_i and target t_i of UBSs in functions that belong to F, so for instructions of the form s_i : $jmp\ t_i$, the displacement $d_i = t_i - s_i$
6. In the program P, insert FBF ξ that performs the following tasks:
   (a) Initializes $k_0 = key_{FM}$.
   (b) For $1 \leq i \leq n$,
      i. Computes integrity check value v_i.
      ii. Computes key k_i from (k_{i-1}, v_i, AM) by applying a one-way hash function SHA_1.

$$k_i = SHA_1[(k_{i-1} \bigoplus AM) \| v_i] \tag{1}$$

   (c) Stores d_i at $h(k_i)^{th}$ location in array T ($T[h(k_i)] = d_i$) where h is a hash function, $h : \{k_1, k_2, \ldots, k_n\} \rightarrow \{1, 2, \ldots, m\} (n \leq m)$.

7. Compute displacements e_i between source s_i and target t_i of UBSs in functions that belong to β.
8. Insert ICBF ϕ in the program that:
   (a) Computes integrity check value v_i. This value confirms the integrity of code section of the program containing ξ.

(b) Stores displacement e_i in array R at index computed as a one-way hash function of v_i ($R[h(u_i)] = e_i$). The hash function h is the same that was used in Step 6.(c).

9. Replace all UBSs in F by calls to ξ and UBSs in β by calls to ϕ.

The fingerprint is generated as the embedding process executes. The final fingerprint is combination of all derived keys - $FM = k_1 \| k_2 \| \dots k_n$. Users u_i, u_j have distinct initializing keys key_{FM_i}, key_{FM_j}, hence final fingerprints FM_i, FM_j are different.

3.2 Watermark Recognition

The *recognize* algorithm is run with the inputs P', key_{AM}, key_{FM} and outputs the authorship mark AM and fingerprint mark FM. When the program is run with the secret input key_{AM}, the function set F is executed which generates the fingerprint mark $FM = k_1 \| k_2 \| \dots k_n$ by initializing $k_0 = key_{FM}$ and deriving successive keys using Equation (1). The authorship mark AM can be extracted by isolating the one-way hash function $k_i = SHA_1[(k_{i-1} \bigoplus AM) \| v_i]$.

4 Proposed Attack

Objective of the attacker is to convert the fingerprinted program P' to the original program P. Since the displacements in T are permuted, determining the correct target address of UBSs is computationally infeasible. Even if the size of T is small, the program can have error-guards that intentionally corrupt the program after a specific number of run-time errors, making hit-and-trial attack impossible. The function ϕ checks the integrity of ξ, adding to the security of the scheme and thereby making the attack more difficult.

In ξ, the integrity check is done and a key is generated. The key is then mapped to the index in the displacement array where the correct displacement is stored. Security of the scheme depends on the correct execution path being a function of keys and integrity checks. If the key generated or the integrity value is incorrect, the displacement is wrong, and therefore the execution path is wrong. We concentrate our attack on this dependence. As soon as we can *disassociate* the correct execution path from the keys and integrity check, the code generating keys and integrity check can be deleted. The authors of [8] claim that the attacker needs to analyze the data section of the program to notice any changes and read the displacement array. This claim is fallacious as an attacker can track register values, including the stack pointer (SP) at:

1. Entry point of ξ: SP = sp_{i_1}
2. Exit/ Return instruction of ξ: SP = sp_{i_2}

The difference $sp_{i_2} - sp_{i_1}$ gives the displacement value d_i. Identification of the instructions participating in fingerprint generation is also achievable. According to [8], *"In the second phase of the algorithm, the branches in each function f*

that belongs to F are replaced by calls to the FBF". We can create a mapping of functions being called by other functions and thereby create sets of functions which all point to one particular function. ξ can be identified by the stack-pointer modifying statements and the set F can be identified as the set of functions calling ξ. Therefore, key_{AM} is no longer required to identify the set of functions participating in watermarking. Within the set F, each instruction calling ξ and having memory address sp_1 can now be replaced by an unconditional branch to the instruction at sp_2. This can be achieved using inline assembly programming. For example, in C++, a user can make use of *_asm* blocks. As a result, the displacement and hence the correct target address is no longer a function of the key and integrity check.

An example of such a block modifying the stack pointer is given below,

```
_asm {
1:      pop ECX;
2:      add ECX,dis;
3:      push ECX;
}
```

In the above code, statement 1 extracts the current value of Stack Pointer into register ECX. Statement 2 adds the intended displacement *dis* to the popped value and statement 3 pushes back the modified value onto the Stack. The Stack Pointer now contains a modified return address. If *dis* is positive, the new address a_t is greater than the original return address a_r ($a_t > a_r$) and the control is transferred "forward". If it is negative ($a_t < a_r$), control is transferred "backward". Observe that ϕ calls can similarly be replaced by the original UBSs.

After changing calls to ξ and ϕ by UBSs, the two functions (ξ, ϕ) can be deleted. When the *recognize* algorithm is run with input key_{AM}, key_{FM}, the inputs are unused dead variables, the algorithm doesn't output the fingerprint mark FM and the recognition algorithm fails. The resulting software is equivalent to an un-watermarked software.

Summarizing our described process, the steps performed by the attacker are:

1. *Identify* ξ: This task is accomplished by locating stack-pointer modifying statements. For example, in C/C++, searching for *_asm* blocks. If a program contains multiple *_asm* blocks, the ones with modification operation on ESP (Stack Pointer) requires to be targeted.
2. *Identify* F: After identifying ξ, the fact that only the functions that belong to F call ξ can be utilized to identify F.
3. *Displacement computation*: Stack pointer values are recorded at the entry and exit points of ξ (sp_{i_1} and sp_{i_2} respectively) and displacement d_i is equal to $sp_{i_2} - sp_{i_1}$. Target instructions are determined from calling instruction and displacement. In our implementation, we use breakpoints to track the register values.

4. *Replacement of ξ calls to UBSs*: If the purpose of the attack is to remove the watermark, the function calls to ξ are replaced by UBSs to obtain the original watermarked code.
5. *Creating a modified watermarked program*: The attacker can embed his/her own authorship mark AM' after removing the original authorship mark AM. For a successful attack, (AM', FM') should be recognized on running *recognize* algorithm with parameters P', key_{FM}, key_{AM} where $FM' \neq FM$.
   (a) For all f that belong to F, compute the displacement between the calling address and the target address and store in an array along with the calling address.
   (b) Replace the UBSs by call to a new Fingerprint Branch Function, $\widetilde{\xi}$.
   (c) $\widetilde{\xi}$ **need not** compute integrity check but simple calculates a new key based on the old key and attacker's authorship mark AM'.

$$k_i = SHA1[k_{i-1} \bigoplus AM']. \tag{2}$$

   Comparing (1) and (2), $k'_i \neq k_i$, $1 \leq i \leq n$.
   (d) Map the keys to correct displacement using hash,

$$h : \{k'_1, k'_2, \ldots, k'_n\} \rightarrow \{1, 2, \ldots, m\} (n \leq m)$$

$$T[h(k'_i)] = d_i$$

The key sequence FM' generated is different from the original key sequence FM as the individual keys are different. More formally,

$$k'_1 \neq k_1, k'_2 \neq k_2, \ldots, k'_n \neq k_n$$

$$\Rightarrow \{k'_1, k'_2, \ldots, k'_n\} \neq \{k_1, k_2, \ldots, k_n\}$$

$$\Rightarrow \{k'_1, k'_2, \ldots, k'_n\} \neq FM$$

$$\Rightarrow FM' \neq FM$$

The recognition algorithm now outputs FM', AM' when executed with the inputs P', key_{AM}, key_{FM}.

In terms of efficiency, the overall complexity of attack depends on complexities of steps 3 and 4 as others are one-off steps. Steps 3 and 4 have linear complexity and hence the attack has $O(n)$ complexity. Steps 1 and 2 are automated and no human inspection is required to identify ξ and F.

5 Implementation Details and Results

We have implemented the watermarking scheme in Visual C++ and carried out the attack using the same. The features useful in doing so are the debug lookup windows - disassembly and register. The stack pointer value can then

be tracked by using breakpoints under debugging mode and there is minimal manual intervention or inspection required. The following is disassembled code of the watermarked program used to compute displacement values.

Function f_i that belongs to F calling FBF ξ in statement 94:

```
0041198C  rep stos dword ptr es:[edi]
0041198E  mov         eax,dword ptr [a]
00411991  cmp eax,dword ptr [b]
00411994  jle         greater+2Bh (41199Bh)
00411996  call        fingerprint (411271h)
0041199B  push        offset string " is greater \n" (4177A8h)
004119A0  mov         esi,esp
004119A2  mov         eax,dword ptr [b]
004119A5  push        eax
004119A6  mov         ecx,dword ptr [__imp_std::cout (41A350h)]
004119AC  call        dword ptr
          [__imp_std::basic_ostream<char,
          std::char_traits<char>>::operator<< (41A354h)]
004119B2  cmp         esi,esp
004119B4 call @ILT+425(__RTC_CheckEsp) (4111AEh)
004119B9  push eax
004119BA  call std::operator<<<std::char_traits<char> > (411168h)
004119BF  add esp,8
004119C2  jmp         l1+27h (4119EBh)
004119C4  push        offset string " is greater \n" (4177A8h)
004119C9  mov         esi,esp
004119CB  mov         eax,dword ptr [a]
-------------------------------------------------------------------

Fingerprint branch function code modifying return address:

00414AF2  mov         eax,ebp
00414AF4  add         eax,4
00414AF7  mov         ebx,esp
00414AF9  mov         esp,eax
00414AFB pop ecx
00414AFC  sub eax,eax
00414AFE  add         eax,0Ah
00414B01  add ecx,dword ptr [dis (419334h)]
00414B07  push ecx
00414B08  mov esp,ebx
-------------------------------------------------------------------
```

Register values are tracked while the program is executed and the following results are obtained:
Statement 00414AF2: EIP stores calling address, EIP=00411996.

Statement 00414AFB: Return address, stored in the stack pointer, is popped into ECX, ECX = 0041199B.
Statement 00414B01: ECX adds displacement value to calling address, ECX = 004119C4.
Statement 00414B07: ECX value is pushed onto stack pointer. *fingerprint();* returns control to this address.

In a nutshell, instruction 94 calls *fingerprint();* which returns control to instruction 98 (the target of the original UBS) based on the value of *dis* looked up from array T. The attacker can thus compute the difference between ECX value at statement 80 (ECX_{80}) and ECX value at statement 83 (ECX_{83}) to find the value of displacement, then replace *fingerprint();* call at statement 94 by UBS transferring control to $\Psi(\Phi(94) + ECX_{83} - ECX_{80})$ (where $\Phi(x)$ denotes address of instruction x and $\Psi(y)$ represents instruction at address y).

6 Conclusion and Future Work

In this paper, we present a successful low-cost attack on the branch-based watermarking scheme proposed in [8]. The cost of the attack is low in terms of hardware resources required since the only resources required are a functional computer with sufficient memory, storage and speed. The attack is efficient as manual inspection is required only during the step in which displacement values are noted from the disassembly register window. Even this is a debugger-specific constraint and in theory, it can be automated, however, we are unaware of an existing debugger that can perform this task. We provided an implementation of our scheme and some practical examples. The work lays a strong foundation for attacking similar software watermarking models [1,2,3,4,13] that depend on branching and inserting bogus functions in the program in order to embed a watermark. This paper also shows that tracking registers and branches is a trivial task using debugging tools and hence opens up a very interesting question of how can the watermarking schemes survive attacks with such advanced capabilities? Our future work is concerned with the following:

– We have shown that the attack is correct in theory and implemented a semi-automated version of the attack. We will work towards enhancing the implementation such that a fingerprinted program written in any language can be attacked. Practically, this is feasible since the attack operates on the disassembled code which, irrespective of the programming language in which it is written, is similar. However, the challenge would be to make the register tracking process compiler-independent. We also intend to design attacks for other branch-based watermarking schemes. Since the central security guard in such schemes is the dependency of the target address on integrity check and watermark values, they can be attacked in a manner similar to our attack.

– Modifying scheme proposed in [8] so that the attack described in this paper is rendered ineffective by creating more complex dependency of inherent functionality of the program on the keys generated so that the attacker cannot remove fingerprint code without affecting the correct execution of the program. This can be done by introducing parameters other than displacement to bind the program's execution to the keys generated. As a simple example, a program may modify itself choosing from a set of modifications based on the key generated.

Acknowledgements

We would like to extend our appreciation for the contributions made by Saurabh Singh towards this research. The second author is supported by Australian Research Council grants DP0345366 and DF0451484.

References

1. Christian Collberg, Edward Carter, Saumya Debray, Andrew Huntwork, Cullen Linn, and Mike Stepp. Dynamic path-based software watermarking. In *Proceedings of Conference on Programming Language Design and Implementation*, volume 39, pages 107–118, June 2004.
2. Christian Collberg, Andrew Huntwork, Edward Carter, and Gregg Townsend. Graph theoretic software watermarks: Implementation, analysis, and attacks. In *Proceedings of 6th Information Hiding Workshop, LNCS*, volume 3200, pages 192–207, 2004.
3. Christian Collberg, Stephen Kobourov, Edward Carter, and Clark Thomborson. Error-correcting graphs for software watermarking. In *Proceedings of 29th Workshop on Graph Theoretic Concepts in Computer Science*, pages 156–167, 2003.
4. Christian Collberg and Clark Thomborson. Software watermarking: Models and dynamic embeddings. In *Proceedings of Principles of Programming Languages 1999, POPL'99*, pages 311–324, 1999.
5. Christian S. Collberg and Clark Thomborson. Watermarking, tamper-proofing, and obfuscation - tools for software protection. In *IEEE Transactions on Software Engineering*, volume 28, pages 735–746, August 2002.
6. Kazuhide Fukushima and Kouichi Sakurai. A software fingerprinting scheme for java using classfiles obfuscation. In *Proceedings of Information Security Applications, LNCS*, volume 2908, pages 303–316, 2004.
7. Ginger Myles and Christian Collberg. Software watermarking through register allocation: Implementation, analysis, and attacks. In *Proceedings of International Conference on Information Security and Cryptology, LNCS*, volume 2971, pages 274–293, 2003.
8. Ginger Myles and Hongxia Jin. Self-validating branch-based software watermarking. In *Proceedings of 7th Information Hiding Workshop, LNCS*, volume 3727, pages 342–356, 2005.
9. Jasvir Nagra and Clark Thomborson. Threading software watermarks. In *Proceedings of 6th Information Hiding Workshop, LNCS*, volume 3200, pages 208–223, 2004.

10. Gang Qu and Miodrag Potkonjak. Analysis of watermarking techniques for graph coloring problem. In *Proceedings of International Conference on Computer Aided Design*, pages 190–193, 1998.
11. Gang Qu and Miodrag Potkonjak. Hiding signatures in graph coloring solutions. In *Proceedings of 3rd Information Hiding Workshop, LNCS*, volume 1768, pages 348–367, 1999.
12. Mikhail Sosonkin, Gleb Naumovich, and Nasir Memon. Obfuscation of design intent in object-oriented applications. In *Proceedings of 3rd ACM workshop on Digital Rights Management*, pages 142–153, 2003.
13. Clark Thomborson, Jasvir Nagra, Ram Somaraju, and Charles He. Tamper-proofing software watermarks. In *Proceedings of Australasian Information Security Workshop*, volume 32, pages 27–36, 2004.
14. Ramarathnam Venkatesan, Vijay Vazirani, and Saurabh Sinha. A graph theoretic approach to software watermarking. In *Proceedings of 4th Information Hiding Workshop, LNCS*, volume 2137, pages 157–168, 2001.

Geometric Invariant Domain for Image Watermarking

Chaw-Seng Woo[1], Jiang Du[1], and Binh Pham[2]

[1] Information Security Institute (ISI), Faculty of Information Technology, Queensland University of Technology GPO Box 2434, Brisbane, QLD4001, Australia
cs.woo@student.qut.edu.au, j.du@isrc.qut.edu.au
[2] Faculty of Information Technology, Queensland University of Technology GPO Box 2434, Brisbane, QLD4001, Australia
b.pham@qut.edu.au

Abstract. To enable copyright protection and authentication, robust digital watermark can be embedded into multimedia contents imperceptibly. However, geometric distortions pose a significant threat to robust image watermarking because it can desynchronize the watermark information while preserving the visual quality. To overcome this, we developed an invariant domain with three transforms; Fast Fourier Transform (FFT), Log-Polar Mapping (LPM), and Dual Tree-Complex Wavelet Transform (DT-CWT). Shift invariance is obtained using FFT. Rotation and scaling invariance are achieved by taking the DT-CWT of a LPM output. Unlike most invariant schemes, our method eliminates explicit re-synchronization. The method resists geometric distortions at both global and local scales. It is also robust against JPEG compression and common image processing. In addition, it exploits perceptual masking property of the DT-CWT subbands, and its watermark detection step does not require the cover image. Experiment on a large set of natural images shows the robustness of the new scheme.

1 Introduction

To enable copyright protection and authentication, digital watermark can be embedded into multimedia contents imperceptibly. The robust watermark must stays intact with the content under various distortions to serve this purpose. However, there are geometrical manipulations, compression techniques, and common image processing operations that can defeat many watermarking schemes. Specifically, basic geometrical attacks such as rotation, scaling, and translation (RST) pose significant threats to image watermarking due to its ease of implementation and de-synchronization effects. Compared to image re-synchronization techniques [1, 2], transform invariant approaches [3] has several advantages. Firstly, the latter is independent of image contents and its features. This is advantageous especially for images without distinctive features. Secondly, the latter generally has lower interpolation errors due to the absence of re-synchronization steps. Combination of transforms used in invariant domains may introduce much more errors. Finally, blind watermark detection can be implemented easily for better practicality.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 294 – 307, 2006.

Invariant domain methods rely on the invariant properties of a transformed domain to resist distortions. Ruanaidh [3] developed a framework for RST invariant domain using the Fourier-Mellin (FM) transform. However, the watermark detection described was a non-blind method. Later, another invariant domain method was developed in [4] that works on one-dimensional (1-D) signals with a small search space. However, it was not designed to resist cropping attack. Based on the FM framework, phase information was used to construct an invariant domain [5, 6]. Later, it was improved by using LPM and phase-only filtering [7]. However, the method still requires a resynchronization step. Following that, Radon and Fourier transforms was experimented in [8] but it requires exhaustive search to resist scaling attack. In another attempt [9-11], the first FFT step in the FM framework was replaced with a robust centroid but the centroid itself could be the target of attack.

In summary, many attempts to create a RST invariant domain had been reported throughout the years. However, many methods are not truly invariant because they still need re-synchronization in a small search space. We developed an invariant domain that omits the need of re-synchronization, enables blind watermark detection, and exploits perceptual masking property of Dual Tree-Complex Wavelet Transform (DT-CWT) subbands. In addition, our work is also motivated by the lack of literature in robust watermarking that incorporates shift invariant wavelets. There are not many wavelets that have shift invariant property.

We justify the adoption of the DT-CWT in Section 2 and explain the watermarking scheme in Section 3. After that, Section 4 gives experimental results that support its robustness. Several factors worth consideration are discussed in Section 5 before we conclude the work in Section 6.

2 Background

An invariant domain for image watermarking was designed by taking advantage of DT-CWT properties. Many geometrical attacks can be modeled as combinations of basic operations. Focusing on this, the invariant domain must resist three basic geometrical operations, i.e. rotation, scaling, and translation (RST). Shift invariance of the FFT, rotation linearity and scaling linearity of the LPM, and approximate shift invariance of the DT-CWT are employed to produce an invariant domain. The construction of this invariant domain is detailed in Section 3. One major advantage of the scheme is the implicit invariant property of the domain, thus eliminating the need for re-synchronization. In addition, perceptual masking property of the DT-CWT subbands is used in watermark embedding.

2.1 The Dual Tree-Complex Wavelet Transform

The conventional discrete wavelet transform (DWT) decompose a signal into low and high frequencies using a binary tree structure. DT-CWT consists of two trees, each of it has linear phase filters that give the real and imaginary coefficients in its forward transform. Odd-length filters in one tree are paired with even-length filters in another tree. The final outputs are averaged to give approximate shift invariance. In its inverse transform, biorthogonal filters are applied in each tree separately. The filters used in

the forward transform and inverse transform are almost orthogonal. We can only achieve approximate shift invariance with DT-CWT because filters with compact support will not have zero gain in its stop bands in real life. This is also due to the little differences between the frequency responses of odd-length and even-length filters [12, 13].

2.2 Advantages of the DT-CWT

Compared to the other FM-based watermarking methods, our method does not require any re-synchronization. The proposed method also enables blind watermark detection through dynamic thresholding. In addition, perceptual masking can be implemented easily by using the DT-CWT subbands during watermark embedding.

Wavelet-based watermarking methods enjoy multi-scale analysis and spatial information which are not available in Fourier transform-based methods. Nevertheless, wavelet-based methods lacked shift invariance until recent years. Kingsbury and Selesnick designed wavelet transforms with such property [13]. One of the best transforms is the DT-CWT. The application of DT-CWT in robust watermarking was reported by several researchers recently [14, 15]. However, they all require re-synchronization to combat geometric distortions.

We use shift invariance of DT-CWT to achieve RST invariance without re-synchronization. This aims at exploiting the advantages of DT-CWT compared to FFT and DWT. Two major shortcomings of FFT are the lack of multi-resolution sampling and perceptual masking property. Therefore, multi-resolution analysis and Human Visual System (HVS) modeling are not implicitly present in FFT methods [16]. Adversely, wavelet-based methods can be implemented with HVS masking easily. This is possible because wavelet-based methods encode spatial and frequency information in its transform domain, and they are superior compared to Discrete Cosine Transform (DCT) and DFT approaches which only store frequency information. DT-CWT was designed to provide approximate shift invariant property [17]. The DT-CWT provides two properties that are absent in DWT, i.e. approximate shift invariance and directional selectivity. Besides that, the perfect reconstruction property eliminates block artifacts in the reconstructed stego image. Although undecimated wavelet transform can offer shift invariance, it requires a huge amount of computation and high redundancy. Compared with the steerable pyramids method, DT-CWT offers well-balanced properties of shift invariance, directional selectivity, and redundancy [12]. Moreover, DT-CWT had been reported to offer better fidelity and higher robustness compared to DWT [18]. Table 1 summarizes the comparisons.

Table 1. Summary of DT-CWT, DWT, and FFT properties

Property	DT-CWT	DWT	FFT
Shift invariance	Yes	No	Yes
Perfect reconstruction	Yes	Yes	Yes
Perceptual masking	Yes	Yes	No
Multi-resolution sampling	Yes	Yes	No

3 Watermarking Method

We investigated a RST invariant domain watermarking scheme as shown in Figure 1 to exploit the advantages of DT-CWT. It uses the properties of FFT, LPM, and DT-CWT to achieve RST invariance.

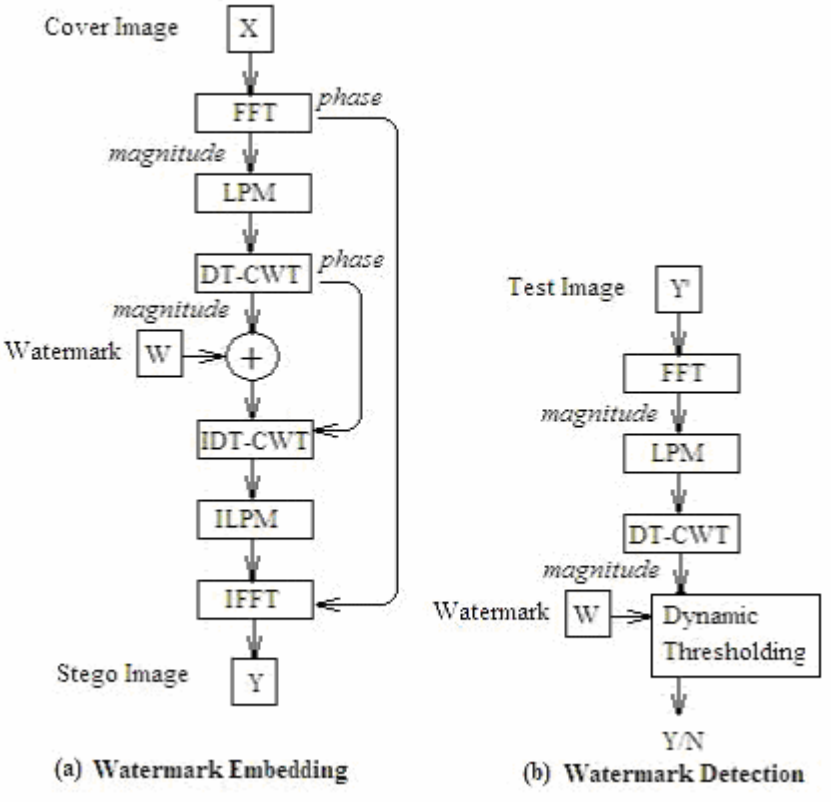


Fig. 1. Invariant Domain Watermarking Scheme

Given an image $I(x,y)$, and its FFT as $I(u,v)$, we can write the LPM of FFT as $I(\rho,\theta)$ where $u = e^{\rho}\cos\theta$ and $v = e^{\rho}\sin\theta$. Let α, β ,and (x_0,y_0) be the parameters of rotation, scaling, and translation respectively. Then, image rotation in $I(x,y)$ corresponds to cyclical shift of α along the angle θ axis in LPM of the magnitude component of FFT. Also, image scaling in $I(x,y)$ corresponds to translational shift of $\ln\beta$ along the log-radius ρ axis in LPM of the magnitude component of FFT. Finally, translation in $I(x,y)$ does not change the coefficients in LPM of the magnitude component of FFT because the FFT's magnitude component is shift invariant. Therefore, RST operations in $I(x,y)$ are transformed into linear shift in the FFT-LPM output. By sending this output to a DT-CWT, we obtain RST invariance with the magnitude component of the final output due to its shift invariant property. The consistent response of DT-CWT to

linear shift allows us to use a correlator detector with dynamic thresholding in blind watermark detection.

3.1 Watermark Embedding

To embed a watermark pattern **W** into an image **X**, we perform a series of forward transformation as shown in Figure 1(a). The cover image **X** is sent through FFT, LPM, and finally DT-CWT to produce invariant domain coefficients. Then, the watermark pattern **W** is embedded using an additive embedding technique with a global weight factor **f** and a simple HVS masking as follows.

$$I'^{\theta}_{0}(i,j) = I^{\theta}_{0}(i,j) \times \left[1 + \mathbf{f} \times w^{\theta}(i,j)\right] \tag{1}$$

where $I'^{\theta}_{0}(i,j)$ is the watermarked subband coefficients with the subbands $\theta \in \{0,1,2,3,4,5\}$, $I^{\theta}_{0}(i,j)$ is the DT-CWT subband coefficients transformed from the cover image **X**, the embedding weight factor $\mathbf{f} \in \{\mathbb{R}^{+}\}$, and $w^{\theta}(i,j)$ is the watermark pattern **W** arranged in the subbands θ dimension.

The watermark pattern $\mathbf{W} \in \{-n, +n\}$, $n \in \{\mathbb{R}^{+}\}$ is a pseudo random pattern to mimic random noise. It has zero mean in order to minimize the changes made to the cover image. Note that the magnitude components of FFT and DT-CWT are sent to its subsequent steps because they have the invariant properties. Its corresponding phase information is used in the inverse transformation steps to construct the stego image **Y**. In addition, the backward transformation steps must take the reverse order because they are not commutative with the forward transformation steps, i.e. inverse DT-CWT is carried out first, followed by inverse LPM, and finally inverse FFT.

We expect the transformed domain to be RST invariant. However, interpolation error introduced by the transform may affect its performance. This is particularly true for LPM where complete and unique mapping is difficult. As a result, robustness to rotation and scaling attacks may be degraded.

3.2 Watermark Detection

To determine whether the watermark **W** exist in a given test image **Y'** which could possibly be attacked, a series of forward transformation depicted in Figure 1(b) is performed. It consists of a FFT, followed by a LPM, and finally a DT-CWT. The magnitude components of FFT and DT-CWT steps are sent to its subsequent steps because it has shift invariant property. Then, all of the 6 subband coefficients in the invariant domain are used in a dynamic thresholding computation. These steps are similar to the embedding process because we need to transform the test image into the same domain for RST invariance.

Blind watermark detection is enabled through a cross correlation computation based on the Neyman-Pearson criterion [19, 20]. We adapted the computation of the correlation value ρ and its threshold value T_ρ to cater for 6 subbands in the invariant domain with false detection probability $P_f \leq 10^{-8}$. If the calculated value ρ is greater than its corresponding threshold T_ρ, then the watermark **W** is detected, otherwise **W** is absent. The commonly chosen false detection probability P_f range from 10^{-6} to 10^{-12}

[21], and we take an intermediate value of 10^{-8}. Besides the RST invariance provided by the combination of three transforms mentioned above, the correlation detector with thresholding can discard changes resulting from amplitude scaling [4].

Instead of working on 3 subbands of DWT as reported in [19], we adapted the dynamic thresholding method to cater for 6 subbands of DT-CWT. As a result, we change the computation of ρ and T_ρ as follows. The correlation between the invariant domain coefficients and the watermark pattern **W** is given by Eq.(2).

$$\rho = \frac{1}{6MN} \sum_{\theta=0}^{5} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I'^{\theta}_{0}(i,j) w^{\theta}(i,j) \tag{2}$$

where $I'^{\theta}_{0}(i,j)$ is the DT-CWT subband coefficients for the test image **Y'**, $w^{\theta}(i,j)$ is the watermark pattern **W** arranged in the subbands θ with $\theta \in \{0,1,2,3,4,5\}$, and $2M \times 2N$ is the size of the test image **Y'**.

The computation of the threshold value T_ρ is also adapted to DT-CWT subbands [19]. The probability of missing the watermark at a false detection rate is minimized using the following cases:

Case A: the image has no watermark.
Case B: the image is watermarked with **W'**, **W'** ≠ **W**.
Case C: the image is watermarked with **W**.

The watermark embedded $w^{\theta}(i,j)$ is binary valued independent random variables with zero mean. Using the Central Limit Theorem (CLT), we assume $I'^{\theta}_{0}(i,j)$ to be independent variable with Gaussian distribution, and the random variable ρ also has Gaussian distribution. Then, the false detection probability P_f = Prob($\rho > T_\rho$ | Case A OR Case B) is estimated using the variance of ρ for Case A

$$\sigma^2_{\rho A} = \frac{\sigma^2_w}{(6MN)^2} \sum_{\theta=0}^{5} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} E\left[\left(I^{\theta}_{0}(i,j)\right)^2\right] \tag{3}$$

and Case B

$$\sigma^2_{\rho B} = \frac{\sigma^2_w}{(6MN)^2} \sum_{\theta=0}^{5} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} E\left[\left(I^{\theta}_{0}(i,j)\right)^2\right] + \mathbf{f}^2 \sigma^2_w E\left[\left(I^{\theta}_{0}(i,j)\right)^2\right]. \tag{4}$$

Given a higher variance, Case B has higher probability. Therefore,

$$P_f \le \frac{1}{2} \operatorname{erfc}\left(\frac{T_\rho}{\sqrt{2\sigma^2_{\rho B}}}\right). \tag{5}$$

From mass experimental results for $P_f \le 10^{-8}$[19], the threshold value is computed by

$$T_\rho = 3.97\sqrt{2\sigma^2_{\rho B}}\,. \tag{6}$$

To estimate the variance for Case B, the mean square value of watermarked coefficients is

$$E\left[I'^{\theta}_{0}(i,j)^2\right] = E\left[I^{\theta}_{0}(i,j)^2\right] + \mathbf{f}^2\, E\left[w^{\theta}_{0}(i,j)^2\right] + 2\mathbf{f}\, E\left[I^{\theta}_{0}(i,j)\;\; w^{\theta}_{0}(i,j)\;\right] \tag{7}$$

With $\sigma_w^2 = \sigma_w^4 = 1$, $E\left[I^{\theta}_{0}(i,j)\right] = E\left[w^{\theta}_{0}(i,j)\right] = 0$, and that $w^{\theta}_{0}(i,i)$ is independent of $I^{\theta}_{0}(i,j)$, we obtain

$$\sigma^2_{\rho B} = \frac{1}{(6MN)^2} \sum_{\theta=0}^{5} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} E\left[\left(I'^{\theta}_{0}(i,j)\right)^2\right] . \tag{8}$$

An unbiased estimate is

$$\sigma^2_{\rho B} \approx \frac{1}{(6MN)^2} \sum_{\theta=0}^{5} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(I'^{\theta}_{0}(i,j)\right)^2 \tag{9}$$

where $I'^{\theta}_{0}(i,j)$ is the DT-CWT subband coefficients for test image **Y'** with the subbands $\theta \in \{0,1,2,3,4,5\}$, and $2M \times 2N$ is the size of the test image **Y'**.

4 Analysis of Experiment Results

To evaluate the robustness of the implemented watermarking scheme, we performed a set of attacks on the stego images using *StirMark 3.1* [22, 23] and carried out the watermark detection steps. Most of the complex image manipulations such as projection can be modeled as local RST operations. Some of the basic attacks are listed in Table 2. All attacks were performed using *StirMark* except the RST-JPEG combined attack marked with asterisk (*) which was implemented in *Matlab*. Details of each attack are provided for completeness.

The images are selected to represent various characteristics, and are illustrated in Figure 2. They are all gray scale images with standard dimension 256×256 pixels. The images are identified by name: *Lena, Baboon, Cameraman, Pepper*, and *Fishing boat*. *Lena* has a mixture of characteristics (i.e. smooth background, while the hat has complex textures and big curves); *Baboon* represents images with large areas of complex texture (i.e. the fur) and homogeneous areas (i.e. the face); *Cameraman* is chosen for its flat regions (i.e. the sky) and high contrast regions (i.e. the shirt and its background); *Pepper* provides luminosity changes (i.e. light reflection surfaces); *Fishing boat* contains smooth parts (i.e. the clouds) as well as other features.

The experiments begun with watermark embedding, followed by attacks mentioned in Table 2, and finally watermark detection. A watermark of 128×128 pseudo-random binary values was generated and embedded into all of the DT-CWT subbands in the invariant domain. The embedding weight factor **f** was computed using the average values of all subband coefficients.

Fig. 2. Test images (Upper left: *Lena*, Upper middle: *Baboon*, Upper right: *Cameraman*, Lower left: *Pepper*, Lower middle: *Fishing boat*)

Table 2. Robustness attacks using *StirMark*

Attack	Description
Rotation with Cropping	Rotation angle from –2° to 90° with cropping
Scaling	Scaling factor from 0.5 to 2.0
Translation	Circular shift 50% of image size
JPEG compression	Quality factor of 10% to 90%
Random bending	Random bending attack
Row and column removal	Remove 1 to 17 rows and columns.
Median filtering	Kernel size from 2×2 to 4×4
Cropping	Crop off 1% to 75% of image size
Gaussian filtering	Kernel size 3×3
Linear transform	General linear transformation
Aspect ratio change	Change aspect ratio in x and y directions
Rotation with cropping and scaling	Rotation angle from –2° to 90° with scaling
Sharpening	Kernel size 3×3
Shearing	Shear in x and y directions
Combination of RST and JPEG compression*	Circular shift 10 columns to the right, scale down to 220×280 pixels, rotate at 15° anti-clockwise, and JPEG compress with quality factor 50%

Table 3 lists the average results of robustness tests. The column of "5 images" contains average score of all the test images mentioned above whereas the column of "3 images" contains average score of *Baboon*, *Lena*, and *Fishing boat* images. The

scores are normalized to the range from 0 to 1. A score of 1.000 means the watermark was detected in all images for all levels of attacks in that category. Adversely, a score of 0.000 indicates no watermark was detected in all cases. The results in "3 images" column is used to compare the performance of our method with Kim's method [11], which was reported to outperform several state of the art watermarking methods. Under rotation with cropping attack, our method scored 0.875 whereas Kim's method has 0.95. Our method suffers much information lost under cropping. This can be improved by watermarking small blocks of the image instead of the whole image. For scaling attack, our method achieved 0.722 compared to Kim's 0.87. Although our scores were slightly lower than Kim's scores in these categories, the situation was reversed in the other two attacks, i.e. our scores for random bending attack (RBA) and JPEG compression are 1.000. Kim's scores for RBA and JPEG attacks were 0.95 and 0.90 respectively. These could be due to the threshold value chosen in the experiments. We fixed the false positive probability to be less than 10^{-8} for all types of attacks whereas Kim's mass tests yielded best result of 7.8×10^{-2}. Therefore, we could safely conclude that our results are better.

Table 3. Average results for robustness tests

Attack	Average results	
	5 images	3 images
Rotation with cropping	0.875	0.875
Scaling	0.700	0.722
Translation	1.000	1.000
JPEG compression	1.000	1.000
Random bending	1.000	1.000
Row and column removal	1.000	1.000
Median filtering	1.000	1.000
Cropping	0.667	0.667
Gaussian filtering	1.000	1.000
Linear transform	1.000	1.000
Aspect ratio change	1.000	1.000
Rotation with cropping and scaling	1.000	1.000
Sharpening	1.000	1.000
Shearing	1.000	1.000
Combination of RST with JPEG compression*	1.000	1.000

4.1 Rotation

Rotation operation is transformed into linear shift operation in LPM, and later transformed into the invariant domain with DT-CWT operation. Therefore, our method can resist most of the rotation levels. The lowest score appeared at 45° rotation because it has the least correlation value and biggest cropped area compared to other levels of

rotation. However, in another set of experiments implemented using *Matlab*, watermark was detected in all degrees of rotation. This can be explained by the difference in the cropping methods between *StirMark* and *Matlab*. The former causes too much information lost compared to the latter. Figure 3 illustrates a comparison between them for a 45° rotation with cropping.

4.2 Scaling

Scaling in the spatial domain is transformed into linear shift in the LPM output. Then, it is further transformed into invariant coefficients in the DT-CWT output. As a result, the implemented method can resist scaling attack. It is generally agreed that scaling smaller than half of the original image size would ruin the commercial value of its output. This is the same for scaling larger than twice the original size. The high level of information lost at scaling factor 0.5 caused the method to fail for all test images. This is further deteriorated by the LPM. However, the method performed well for other scaling factors tested.

Fig. 3. Comparison of rotation with cropping under different implementations. Left: *Lena* rotated using *StirMark*; Right: *Lena* rotated using *Matlab*.

4.3 Translation

The magnitude component of FFT is invariant to translation attacks. Consequently, the watermark was detected in all images under circular shift operation. The dynamically computed ρ value stays above the threshold T_ρ for all the tests.

4.4 JPEG Compression

The watermark was detected in all levels of compression quality with JPEG algorithm. Due to the chosen false detection probability of 10^{-8}, the dynamically computed ρ value always stays above the threshold T_ρ for all compression quality factors.

4.5 Random Bending Attack

Local geometrical distortions were carried out on the stego image using the random bending attack (RBA). Although the stego image and its corresponding distorted images appear similar to human eyes, the distorted regions in the attacked images desynchronized pixel locations. This type of attack presents a tough problem to many

robust watermarking schemes because the assumption that distortion is uniform globally does not hold. However, the watermark was detected in all cases using our watermarking scheme. The local shifting in spatial domain caused by RBA has little effect on the magnitude component of FFT and DT-CWT coefficients due to its shift invariance.

4.6 Common Image Processing Operations

Many common image processing operations can be done easily with off-the-shelf software packages, yet these operations can cause de-synchronization in the watermark information and affect watermark detection. Such operations include cropping, median filtering, Gaussian filtering, linear transform, aspect ratio change, image sharpening, and shearing. The test results show that cropping has the most severe effect on watermark detection, especially when 75% of the stego image is cropped off. The remaining stego image area can only provide limited information for watermark detection. On the other hand, the implemented scheme attained 100% successful detection under many other common attacks. The robustness of wavelet-based methods against these attacks is provided by the multiscale and spatial information encoded. Using DT-CWT with a correlation-based watermark detection, the scheme can resist modifications caused by these attacks.

4.7 Combined RST and JPEG Compression

The combination of RST operations followed by JPEG compression mentioned in Table 2 cannot defeat the watermarking scheme. The attacked stego image of *Lena* is illustrated in Figure 4. In addition, the watermark was also detected in many other combinations with varying levels of distortions. This shows that the scheme is extremely robust to geometrical attacks.

Fig. 4. *Lena* attacked with a combination of RST operation and JPEG compression

4.8 Mass Test

Using 500 test images obtained at the website http://www.cs.utah.edu/~sbasu/cbir.html, we embedded watermarks into it and perform watermark detection. The images consist of human body, natural scenes, buildings, transports, animals, and

plants. The results provided very high reliability with watermarks detected in all images except 4 of them.

5 Discussions

We investigated a RST invariant domain watermarking scheme that explored the many advantages offered by DT-CWT. However, some limitations arise from the adoption of the RST invariant framework.

5.1 Advantages

The major contribution of this work lies in the application of DT-CWT properties in developing a RST invariant domain. Our method does not require any re-synchronization. It also enables perceptual masking and blind watermark detection. Although the magnitude component of DT-CWT is only approximately shift invariant, the experimental results proved the high level of robustness. The attacks tested include the basic RST operations, JPEG compression, some common image processing operations, and local geometrical distortion. Images with various characteristics were experimented in the mass test.

The multi-resolution samples of DT-CWT provided fine-tuning capabilities. For instance, watermarking in the high level subbands of the transform increases the robustness at the cost of stego image fidelity. Since the LPM and ILPM operations introduce much interpolation errors, the scheme requires a trade-off between robustness and fidelity. Therefore, we chose to embed the watermark in the lowest level subbands.

To improve stego image fidelity, perceptual masking was applied during watermarking embedding by adjusting the embedding weight according to local coefficient values. A simple approach of such masking was carried out using the DT-CWT subbands.

The perfect reconstruction feature of DT-CWT can compensate the visual quality degradation caused by LPM. Despite the interpolation errors introduced in LPM and ILPM, the *Lena* stego image achieved good visual quality of 38dB in terms of peak-signal-to-noise ratio (PSNR).

5.2 Limitations

There are a few limitations attached to the scheme. One important issue to be resolved is to overcome interpolation errors caused by LPM and improve stego image fidelity. LPM and its inverse operation cause image quality degradation. This is due to interpolations involved in the transform. Ruanaidh [3] minimize such impact with a one-way transform for embedding and another way for detection. Lin [4] avoided this by eliminating *strong invariant* requirement and simplified the data complexity into 1-D stream. They mentioned that invertibility offered by *strong invariant* is not essential because they can substitute it with a watermark extraction function which gives approximately similar results. However, we prefer *strong invariant* for its invertibility and to achieve robustness. Therefore, we suggest the use of a large mapping space with redundancies to overcome under-sampling in LPM. This will improve the visual

quality of the stego image. We found that over-sampling an image with LPM into a space 5 times the original size could give nearly perfect inversion when ILPM is applied. Nevertheless, this will increase the computation cycles.

The adoption of RST invariant framework in the watermarking scheme inherently required large amount of computation. This is caused by 2-D FFT and LPM operations. In addition, DT-CWT also involves a certain amount of computation. Despite all of these, the embedding and detection process performed within acceptable time frame on an average desktop computer. Using a *Pentium III* 800MHz machine with 256 MB of memory, one loop of embedding and detection process does not exceed 2 minutes.

6 Conclusions

Robust watermarks can be applied in DRM scenarios to protect the media contents. However, geometric distortions pose a significant challenge to robust watermarking. Invariant domain that resists RST attacks is a promising approach to overcome such problem. We developed a RST invariant domain taking advantages of DT-CWT properties and HVS masking. The magnitude component of DT-CWT is shift invariant. In addition, DT-CWT offers high robustness with multi-resolution sampling, perceptual masking, and perfect reconstruction. Our method does not require any re-synchronization, enables blind watermark detection and implicit perceptual masking. The invariant domain not only resisted basic RST attacks, it also survives JPEG compression, common image processing operations, and local geometrical distortion. The watermark scheme is also robust under extreme distortions created with combination of these attacks.

References

1. Bas, P., J.-M. Chassery, and B. Macq, *Geometrically invariant watermarking using feature points.* Image Processing, IEEE Transactions on, 2002. **11**(9): p. 1014-1028.
2. Pawlak, M. and Y. Xin. *Robust image watermarking: an invariant domain approach*. in *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*. 2002.pp.885-888 vol.2
3. Ruanaidh, J.J.K.O. and T. Pun, *Rotation, scale and translation invariant spread spectrum digital image watermarking.* Signal Processing, 1998. **66**(3): p. 303-317.
4. Lin, C.-Y., et al., *Rotation, scale, and translation resilient watermarking for images.* Image Processing, IEEE Transactions on, 2001. **10**(5): p. 767-782.
5. Zheng, D. and J. Zhao. *RST invariant digital image watermarking based on resynchronization*. in *Electrical and Computer Engineering, 2004. Canadian Conference on*. 2004.pp.1281-1284 Vol.3
6. Zheng, D., Y. Liu, and J. Zhao. *RST invariant digital image watermarking based on a new phase-only filtering method*. in *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*. 2004.pp.25-28 vol.1
7. Liu, Y. and J. Zhao. *A new filtering method for RST invariant image watermarking*. in *Haptic, Audio and Visual Environments and Their Applications, 2003. HAVE 2003. Proceedings. The 2nd IEEE Internatioal Workshop on*. 2003.pp.101-106

8. Xuan, J., H. Zhang, and L. Wang. *Rotation, scaling and translation invariant image watermarking based on Radon transform*. in *Visual Communications and Image Processing 2005*. 2005: SPIE--The International Society for Optical Engineering.pp.1499-1505
9. Kim, B.-S., et al., *Robust digital image watermarking method against geometrical attacks.* Real-Time Imaging, 2003. **9**(2): p. 139-149.
10. Kim, B.-S., J.-G. Choi, and K.-H. Park, *RST-Resistant Image Watermarking Using Invariant Centroid and Reordered Fourier-Mellin Transform*. 2939 ed. Lecture Notes in Computer Science, ed. T. Kalker, I.J. Cox, and Y.M. Ro. Vol. 2939/2004. 2004. 370-381.
11. Kim, H.S. and H.-K. Lee, *Invariant image watermark using Zernike moments.* Circuits and Systems for Video Technology, IEEE Transactions on, 2003. **13**(8): p. 766-775.
12. Kingsbury, N.G., *Complex wavelets for shift invariant analysis and filtering of signals.* Journal of Applied and Computational Harmonic Analysis, 2001. **10**(3): p. 234-253.
13. Selesnick, I.W., R.G. Baraniuk, and N.C. Kingsbury, *The dual-tree complex wavelet transform.* Signal Processing Magazine, IEEE, 2005. **22**(6): p. 123-151.
14. Loo, P. and N.G. Kingsbury. *Motion estimation based registration of geometrically distorted images for watermark recovery*. in *SPIE Conference on Security and Watermarking of Multimedia Contents III*. 2001. San Diego, USA.: SPIE.pp.606-617
15. Day, M.-L., S.-Y. Lee, and I.-C. Jou, *Watermark Re-synchronization Using Sinusoidal Signals in DT-CWT Domain*. 3333 ed. Lecture Notes in Computer Science, ed. K. Aizawa, Y. Nakamura, and S.i. Satoh. Vol. 3333/2004. 2004: Springer-Verlag GmbH. 394-401.
16. Serdean, C., et al. *Protecting Intellectual Rights: Digital Watermark in the Wavelet Domain*. in *IEEE Int. Workshop on Trends and Recent Achievements in IT*. 2002. Cluj-Napoca, Romania
17. Kingsbury, N. *Shift invariant properties of the dual-tree complex wavelet transform*. in *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*. 1999. Phoenix, Arizona, USA.pp.1221-1224
18. Terzija, N. and W. Geisselhardt. *Digital image watermarking using complex wavelet transform*. in *Proceedings of the 2004 workshop on Multimedia and security, International Multimedia Conference*. 2004. Magdeburg, Germany: ACM Press.pp.193-198
19. Barni, M., F. Bartolini, and A. Piva, *Improved wavelet-based watermarking through pixel-wise masking.* Image Processing, IEEE Transactions on, 2001. **10**(5): p. 783-791.
20. Woo, C.-S., J. Du, and B. Pham. *Performance Factors Analysis of a Wavelet-based Watermarking Method*. in The Third Australasian Information Security Workshop (AISW 2005). 2005. Newcastle, NSW, Australia: Australian Computer Society.pp.89-97
21. Cox, I., M.L. Miller, and J.A. Bloom, *Digital watermarking*. 2001, San Francisco, CA, USA.: Morgan Kaufmann Publishers Inc. 539.
22. Petitcolas, F.A.P., *Watermarking schemes evaluation.* Signal Processing Magazine, IEEE, 2000. **17**(5): p. 58-64.
23. Petitcolas, F.A.P., R.J. Anderson, and M.G. Kuhn. *Attacks on Copyright Marking Systems*. in *Information Hiding: Second International Workshop, IH'98*. 1998. Portland, Oregon, USA.: Springer Berlin / Heidelberg.pp.218-239

Desynchronization in Compression Process for Collusion Resilient Video Fingerprint

Zhongxuan Liu, Shiguo Lian, Ronggang Wang, and Zhen Ren

France Telecom R & D Beijing, 2 Science Institute South Rd, Beijing, 100080, China
zhongxuan.liu@orange-ft.com

Abstract. Till now, few desynchronization methods for video fingerprint have been presented, which are implemented in raw data. In this paper, a compression compliant video desynchronization method for collusion resilient fingerprint is proposed. The technique can simultaneously apply random space/time desynchronization and compression to videos. In our experiments, with little visual degradation, the joint process costs no more time and bandwidth than those of MPEG2 encoding/decoding. By this method, the video quality degrades dramatically for colluded copies. Besides evaluating the method by compression quality, compression time, and visual quality, we also discussed the system security. Two attacks are considered for the security evaluation: re-synchronization attack (including Most Similar Frame Collusion and Random Similar Frame Replacement) and re-desynchronization attack. Schemes for robustness to these attacks are also shown. Two theorems are presented to point out the security limit of single time desynchronization and single space desynchronization and the influence of related parameters to security.

1 Introduction

The fast development of digital techniques not only improves the convenience of processing and transmitting digital media data, but also increases the menace of illegal redistribution of multimedia objects. Embedding imperceptible information into media is a promising solution which can indicate the ownership or the user of the media. Fingerprint is just the technique to indicate user of the media by embedding the user's information imperceptibly. Then the copy received by every user will be visually the same while in fact different copy. When the media is illegally redistributed, the information in the media is used to identify the illegal users.

The most serious menace to fingerprint is the collusion attack. This attack combines several copies of a media and derives a new copy hard to identify the attackers. In [1], the security of fingerprint for linear collusion (such as averaging and cut-and-paste collusion attacks) and nonlinear collusion (such as minimum/maximum/median/minmax/modified negative and randomized negative collusion attacks) are analyzed. In [2], scalable video collusion attack is considered. In [3], a new attack to coded fingerprint called LCCA (linear combined collusion attack) is proposed. [4] described a more general signal processing

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 308–322, 2006.

attack in which the colluders employ multiple-input single-output linear shift invariant (MISO-LSI) filtering plus additive Gaussian noise. Combined collusion with frame-dropping attack is considered in [5].

There're two classes of methods to cope with above collusion attacks: one is only devising different fingerprints for different users without changing the carriers (image, audio and video for fingerprint embedding), and the other called desynchronized fingerprint (DF) [6] [7] [8] is to change the carriers for each user and also embed different fingerprints for different users. The main difficulty of former scheme is the hardness of making the fingerprints robust to collusion. The orthogonal Gaussian fingerprint has the shortcoming of detection difficulty for threshold decision under collusion and high time cost when user number enlarges [1]. The BIBD based ACC (anti-collusion code) fingerprint is not robust to LCCA [3]. Although the methods in [9] and [10] are demonstrated to be more robust to collusion than BIBD based ACC fingerprint and spectrum based ECC (error correcting code) fingerprint respectively, they can not solve the problem of collusion essentially.

By desynchronizing the carrier, the later scheme (DF) degrades the quality of the colluded copy seriously. Then the difficult problem of devising and embedding collusion robust fingerprint for identifying colluders is avoided. Mao proposed the DF for raw video [7]. The space desynchronization is implemented by 2-D global affine warping and local bending. The time desynchronization is processed by forming frames utilizing motion vectors estimated by optical flow algorithm. There're two problems for Mao's method: Firstly, the computing and transmission efficiency (for saving bandwidth, videos should be compressed before transmission) of the method are very low. Mao's method is very time costing because motion estimation by optical flow is needed to be done for every pair of neighbor frames. The bandwidth consumption of the compressed form of Mao's method's output will be large because space desynchronization will increase bandwidth consumption (see Section 2.2). Secondly, the security of desynchronization has not been carefully analyzed for attacks when desynchronization algorithm has been known to the attackers.

Our paper will solve above two problems. On one hand, a method called desynchronized fingerprint in compression process (DFCP) is proposed to avoid the heavy time cost of motion estimation for desynchronization. In fact, the time cost of joint process of desynchronization and compression in our scheme is near to the only compressing by MPEG2. The output bandwidth consumption is also near to the MPEG2 stream. The only price is the little visual degradation compared with no desynchronization. The elementary idea of our method is to utilize the estimated motion information of compression process for interpolating desynchronized frames. On the other hand, security of the proposed scheme is also analyzed for attacks when attacker knows the copies have been desynchronized. Two classes of this kind of attacks are proposed and analyzed: re-synchronization attacks (Most Similar Frame Collusion and Similar Frame Replacement) and re-desynchronization attacks. Two theorems are presented to show the security limit of both space and time desynchronization individually.

The proposed scheme of combining desynchronization and MPEG2 compression into a single process is shown in Section 2. We analyzed the compression performance, computing efficiency, visual quality and security of our algorithm in Section 3 followed by the conclusion and future work in Section 4.

2 Time and Space Desynchronization During Compression

2.1 Time Desynchronization

Time Sampling Generation. In Mao's paper [7], constrained random temporal re-sampling is used to form time sampling. The random sequence in Mao's scheme is not appropriate for video time sampling during compressing process for two aspects: firstly, the random numbers used in Mao's method can be any precision such as 5.2437, while motion vector only have limited precision (such as 1/4 pixel in H.264 and 1/2 pixel in MPEG-2), so interpolation for such frame will be difficult; secondly, I frames are not appropriate for frame interpolation because of two reasons: on one hand, I frames only have intra-frame prediction, then the motion vector used for forming it will be useless for encoding; on the other hand, the low quality of I frame will influence the compression efficiency and visual quality of P, B frames in the same GOP (group of pictures). Considering the above two aspects, we do the following three steps as shown in Figure 1 to form the random sequence appropriate for our system:

1. Forming 'random sequence' : a random increasing sequence is formed. For not influencing visual quality and audio-video synchronization apparently, maximum two numbers are permitted between two neighbor integers and at least one number among three neighbor integers;
2. Forming 'rounded sequence': the elements e of 'random sequence' is rounded by a certain steps s ($s = 1/2$ in Fig. 1): $e' = [e/s] \times s$ ($[x]$ is the maximum number no more than x). For example, 1.4 is rounded to be 1.5 and 2.8 is to be 3;
3. Appointing 'frame type': I, P, B structure is given to the sequence of step 1 correspondingly. For example frame 1.5 in 'rounded sequence' is appointed to be I frame;
4. Forming 'result sequence': the 'rounded sequence' corresponding to I frames are regulated to their most neighbor integer elements. Then the new sequence is used for time sampling. For example, frame 1.5 in 'Rounded sequence' is changed to be frame 1 because it has been appointed to be I frame.

Time Desynchronization. Different from Mao's method, we implement time desynchronization by skipping and interpolating frames (see Figure 1). For forming the result sequence in Figure 1 $(1, 3, 3.5, 4, 4.5, 5, 7)$: the frames just like $2, 6$ frames are skipped and the fractional frames like $(3.5, 4.5)$ are interpolated.

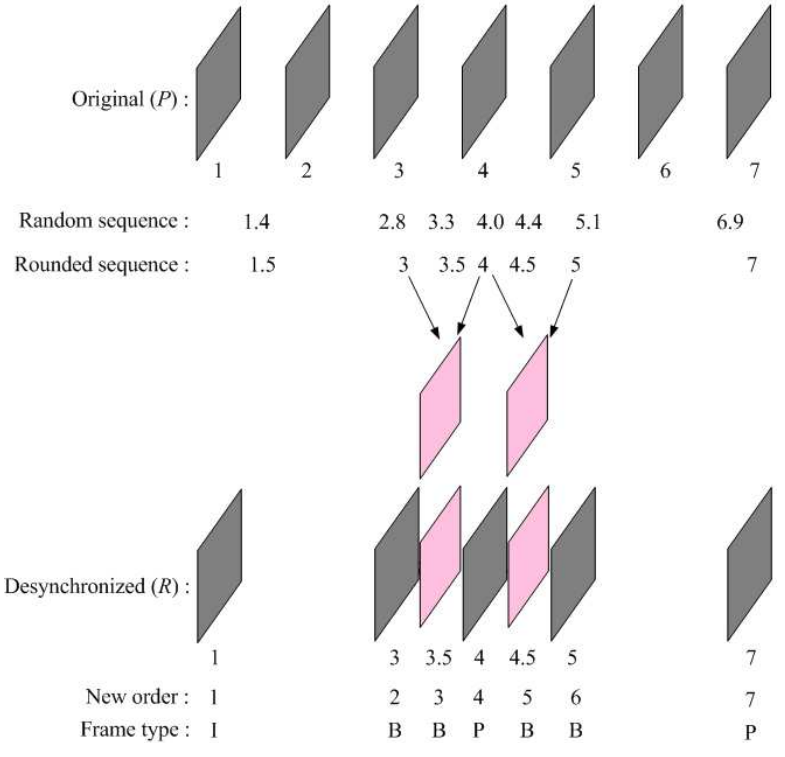


Fig. 1. Time desynchronization

2.2 Space Desynchronization

In Mao's paper [7], space desynchronization is implemented by two methods: RST (rotation, scaling and translation) and random bending. Theoretically, the space desynchronization in [7] also can be used in desynchronization during compression process. But considering compression process, RST and local bending are not appropriate: the rotation, scaling and local bending can not only increase computing cost (space desynchronization itself and searching time for motion estimation), but also greatly degrade the compression efficiency (increasing residue energy of motion compensated frames). Then we only utilize translation of images. Firstly, quality of colluded video also degrades drastically (Section 3.3); secondly, the security of the system will be enough which is explained in Section 3.4.

In our method, horizontal and vertical translation parameter pair is (T_x, T_y) (T_x, T_y are elements generated from uniform random distribution $[-7, 7]$ and randomly set every S_T frames, here $S_T = 10$). The parameters of rest frames are linearly interpolated according to their respective nearest two randomly set parameters. Because of the translation, some boundary parts of the translated frames should be filled. For saving time cost, we only set pixel values to be

their nearest outwards pixels. The result of space desynchronization is shown in Figure 2. The three images (from left to right) are the desynchronized copies with $(T_x = -2, T_y = 4)$, $(T_x = 6, T_y = 2)$ and their colluded result respectively.

Fig. 2. Illustration for the space desynchronization: left: copy with $T_x = -2, T_y = 4$; middle: copy with $T_x = 6, T_y = 2$; right: colluded result of the two copies

2.3 Desynchronized Fingerprint in Compression Process (DFCP)

In [7] the space and time desynchronization are implemented by interpolating frames utilizing optical flow motion estimation and space desynchronization compensation of instant times. Because of the huge data quantity of videos, compression should be processed before transmission. The most time consuming part of computing in compression is motion estimation (for example, motion estimation costs 81.78% of time for H.264 encoding [11]). Then for practical desynchronized fingerprint system using Mao's method, the motion should be estimated twice. For saving time cost, we use a method just estimating motion vector once for both space/time desynchronization and compression as shown in the following content.

The integral system is illustrated in Fig. 3. For convenience, we only give the example of the scheme for MPEG2 which is still the most widely used (discussion for other standards is in Section 4). As an example, the video sequence in Figure 1 is considered (for the first four encoded frames): The frames of original video are marked as $P_1, P_2, ..., P_7$. For convenience, the frames of the result sequence are marked as $R_1, R_3, R_{3.5}, ..., R_7$. Five steps are for the system (Fig. 3) in following:

1. A key is used to generate space desynchronization pattern sequence ((T_x, T_y) in Section 2.2) and time desynchronization pattern sequence $(1, 3, 3.5, ..., 7)$;
2. Coding pictures are selected according to time desynchronization pattern: P_2 and P_6 are skipped and others are left;
3. The left pictures are space desynchronized according to their corresponding space desynchronization patterns;
4. Video is compressed with fractional frames interpolated;
5. Fingerprint is embedded into the sequence before (or after) run length and entropy coding.

For explaining the time desynchronization and compression step (step 4), sequence in Figure 1 is processed as follows:

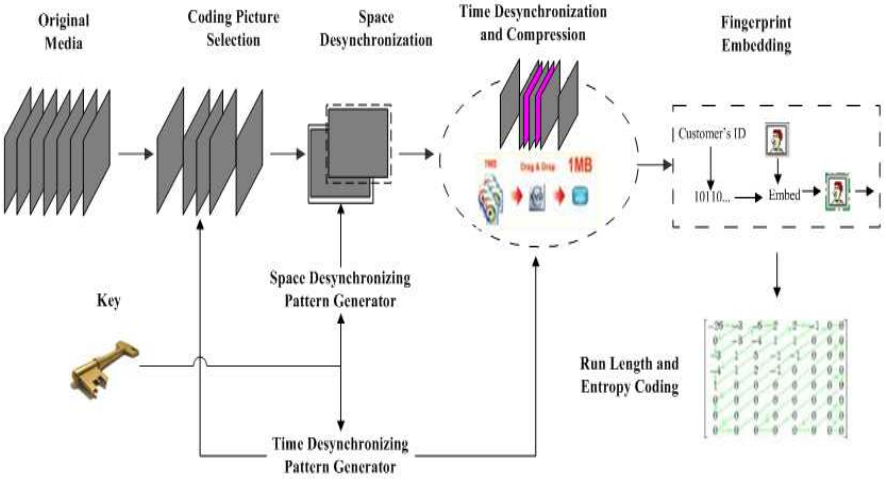


Fig. 3. System structure of DFCP

1. The first step is to form R_1. For this, frame P_1 is space desynchronized to be R_1 and intra-coded;
2. The second step is to form R_4. This is computed by firstly space desynchronizing P_4 to form $\bar{P}_4$ and then compressing the frame by motion compensation using frame R_1 and $\bar{P}_4$;
3. The third step, the same process is also for forming R_3 except using both frame R_1 and R_4 for compensation;
4. For the fourth step, the $R_{3.5}$ is computed by interpolating frame algorithm utilizing R_3 and R_4. After that, $R_{3.5}$ is encoded using compensation of R_1 and R_4. Here, the space desynchronization of $R_{3.5}$ depends on that of R_3 and R_4. In our method, motion compensation interpolation is used [12] for the frame interpolation:

$$f(x,n) = \frac{f(x+v/2, n-1) + f(x-v/2, n+1)}{2}.$$

where $f(x,n)$ is the value at position of x and frame of n, v denotes motion vecotr at the position x in frame of $n+1$.

3 Performance of the DFCP

3.1 Compression Performance

One of the most attractive qualities of our method is that this desynchronization algorithm can improve the compression performance (see Table 1). 'FileOri' is the original compressed file size, 'FileDes' is the desynchronized and compressed file size, and 'Ratio' is the size decreasing ratio. '$_q$' is the QCIF image with size 176×144. '$_c$' is the CIF image with size 352×288. In our experiments, one

third of the sequence is lengthened by interpolating half fractional frames, and two third of the sequence is shortened by skipping one frame every neighbor two frames. For each sequence, 100 frames are used. Here, the computer is of 1.7GHz CPU/512M RAM. Because our experiments are mainly to validate the transmission and time (Section 3.2) efficiency of joint process of desynchronization and compression, influence of watermarking embedding is not considered in the experiments. From Table 1, all of the videos have smaller file sizes when processed by DFCP compared with only MPEG2 compressed streams.

The reason that DFCP can improve compression performance is: in DFCP, desynchronization is implemented by skipping and inserting the same number of frames. The diminished rate of skipping one frame is larger than the increased rate of inserting one frame because inserted frames are formed using neighbor frames then the compensated residue energy is very low.

Table 1. Data quantity comparison of MPEG2 encoding without and with space/time desynchronization

	FileOri (k)	FileDes (k)	Ratio (%)
$foreman_q$	2032	1852	-8.86
$paris_q$	1898	1584	-8.27
bus_q	2035	1933	-5.01
$football_q$	1977	1809	-8.50
$foreman_c$	2035	1863	-8.45
$paris_c$	2035	1979	-2.75
bus_c	2035	1974	-3.00
$football_c$	2035	1747	-14.15

3.2 Computing Efficiency

The time cost of DFCP compared with MPEG2 for encoding and decoding is in Table 2. 'EOri' and 'DOri' are the time cost of MPEG2 encoding and decoding time respectively. 'EDes' and 'DDes' are the time cost of joint encoding and desynchronization process and decoding for the desynchronized and compressed stream process respectively. 'R1' and 'R2' are the time cost increasing for 'EDes' and 'DDes' compared with 'EOri' and 'DOri' respectively. For most sequences, the encoding and decoding time for our algorithm is less than that of MPEG2 respectively. For fast moving videos such as "football", the time cost saving is more apparent. This can be explained as follows: firstly, space desynchronization costs little time (see Table 3); secondly, motion estimation costs most of the time, and inserting and skipping frames diminish half of the motion estimation requirements for related frames. In Table 3, 'EDes' is the time cost of the joint process of desynchronization and compression. 'ESpaceDes' is the time cost of space desynchronization in 'EDes'. 'Ratio' is the ratio of 'ESpaceDes' to 'EDes'.

Table 2. Time consumption comparison of MPEG2 encoding-decoding without and with space/time desynchronization ((s) means second)

	EOri (s)	EDes (s)	R1 (%)	DOri(s)	DDes (s)	R2 (%)
$foreman_q$	7.20	7.37	2.4	2.58	2.52	-2.3
$paris_q$	6.26	6.40	2.2	2.51	2.45	-2.4
bus_q	8.59	8.60	0.1	2.48	2.40	-3.2
$footall_q$	10.62	10.11	-4.7	2.45	2.37	-3.3
$foreman_c$	28.64	27.84	-2.8	7.04	6.97	-1.0
$paris_c$	21.25	21.76	-2.4	6.80	6.98	2.7
bus_c	32.83	32.61	-0.7	6.87	6.85	-0.3
$football_c$	40.03	38.55	-0.4	6.91	6.85	-0.9

Table 3. Time consumption comparison of without and with space desynchronization (both with time desynchronization)

	EDes (s)	ESpaceDes (s)	Ratio (%)
$foreman_q$	7.15	0.17	2.38
$paris_q$	6.18	0.12	1.96
bus_q	8.45	0.19	2.25
$footall_q$	9.98	0.11	1.10
$foreman_c$	27.78	0.59	2.12
$paris_c$	21.21	0.55	2.59
bus_c	32.62	0.56	1.72
$footall_c$	37.68	0.54	1.43

3.3 Visual Quality

Although desynchronization decreases the bandwidth of compressed video, little video quality degradation can be perceived. This depends on four aspects:

1. Only a small portion of total frames are skipped and inserted. For example, a film of an hour and 30 frames/sec has 10800 frames. For such a film, if a new copy is made by only skipping 10 frames randomly during the first half of film, which are 0.01% total frames, and inserting the same number of frames before end. Another copy is the original film. Then these two copies will generate a visually very bad copy by average collusion because the second half of the colluded film will be like the lower right image in Fig. 4;
2. Another degradation of DFCP is caused by frame interpolation, for example, by frame $R_{3.5}$. But this degradation is very small considering two aspects: a) If the motion between the two neighbor frames R_3, R_4 is not obvious, $R_{3.5}$ can be made by R_3 or R_4; b) Otherwise, the quality of the interpolated frame $R_{3.5}$ can be fine because there have been quite a few effective methods proposed for this question [12];

3. The third degradation is caused by space desynchronization. Parameters of translation scope, S_T are used as in Section 2.2 for maintaining visual quality after space desynchronization;
4. Because collusion-resistent code is avoided, the embedded bits are much less than non-desynchronized fingerprint methods. For example, in Boneh and Shaw method [13], a code of length $O(\log^4 N \log^2(1/\epsilon))$ is used for catching up to $\log N$ users out of a total of N users with error probability $\epsilon < 1/N$. While in our method, a code of length $\lceil \log_2 N \rceil$ is enough. This means the less degradation caussed by watermark.

The visual effect of our method is shown in Figure 4. The upper left image is one of the original frames. The upper right and lower left images are frames from two copies with different desynchronization patterns. The lower right one is the colluded frame of the upper right and lower left images. From the figure, DFCP will influence visual quality little if fitful parameters are used. The degradation of the colluded copy ensure the security of our method to collusion attack.

Fig. 4. Visual quality comparison. Upper left: one of the original frames. Upper right and lower left images: frames from two copies with different desynchronization patterns. Lower right: the colluded frame of the upper right and lower left images.

3.4 Security Analysis

Different desynchronization for each copy can solve the collusion problem by degrading the quality of colluded medias [6]. The brute-force space of time and space desynchronization is very large. For a video of 30 fps with length of one hour (total frames: 108000), the parameter space of time desynchronization is

larger than: $4^{\frac{108000}{4}} = 2^{64000}$ (because one frame should be extracted for every two frames, and time samples of former half videos are free to select). The parameter space of space desynchronization is no less than $(15 \times 15)^{108000/10} \approx 2^{86339}$ (for configuration in Section 2.2). Then total parameter space is larger than 2^{150339}. So directly reversing the desynchronized copy by estimating key is not practical. In the following, two more effective attacks for DFCP are analyzed:

Attacks by Re-synchronization. Instead of estimating parameters for generating desynchronization forms, the colluders may form a new copy by re-synchronizing the copies attending collusion referencing one of the copies and combining these synchronized copies. The following are two improved re-synchronization collusion schemes (see Fig. 5):

Assume $(P_1^n, P_2^n, ..., P_M^n)$, $n = 1, ..., N$ are the N video copies and $K_i^{n,j} = \text{argmin}_j \text{dist}(P_i^1, P_j^n), D_i^n = \min_j \text{dist}(P_i^1, P_j^n), (i = 1, ..., M; n = 2, ..., N; j \in [1, M])$ where $\text{dist}(\cdot, \cdot)$ is the mean square distance

Definition 1. *Most Similar Frame Collusion (MSFC):*

$$\bar{P}_i^{MSFC} = F(P_i^1, P_{K_i^{2,j}}^2, ..., P_{K_i^{N,j}}^N)(i = 1, ..., M);$$

Definition 2. *Random Similar Frame Replacement (RSFR):*

$$\bar{P}_i^{RSFR} = P_{K_i^{k,j}}^k$$

where $i \in G_k, (k = 1, ..., K; G_1 \cup G_2 \cup ... \cup G_K = \{1, ..., M\})$.

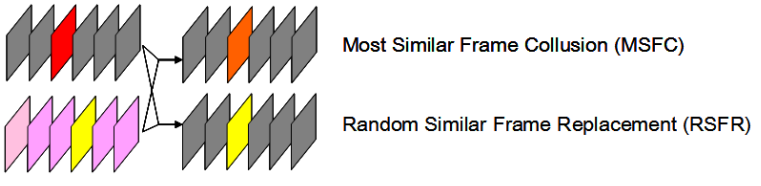


Fig. 5. Illustration for MSFC and RSFR attacks. The yellow frame is the most similar frame of pink video to the red frame.

MSFC is to utilize the most similar frames in other copies to frames in a certain copy to form colluded copy (see Fig. 5). When only completely similar frames (not considering the embedded watermark) are used for collusion, the MSFC is named as Similar Frame Collusion (SFC) attack. Function of $(F(\cdot, \cdot, ..., \cdot))$ in MSFC can be any non-desynchronized collusion for fingerprints such as averaging collusion, cut-and-past collusion, nonlinear collusion [1], LCCA [3] et al. The technique against MSFC lies on method of improving difficulty of finding appropriate 'most similar frame' (MSF) and ensuring still extracting fingerprint after attack. Both using time and space desynchronization is good solving to implement this goal:

The searching space for finding MSF is too large considering both the desynchronization. For example, for the translation parameters in Section 2.2, 10 seconds video for a one hour video is searched and N copies attend the collusion, MSFC needs $15 \times 15 \times 30 \times 10 \times 10800 \times (N-1) \approx (N-1)2^{30}$ frame comparisons for forming the $N-1$ resynchronized copies and colludes them with the reference copy.

RSFR is an attack exchanging the corresponding frames (similar sampling time) of multiple copies (see Fig. 5). Considering the human's cute perception to sharp space desynchronization, RSFR will diminish the visual quality of motion pictures if different space desynchronization are used for different copies. An improved attack is selecting most appropriate frame in multiple copies. The counteraction to RSFR is to independently insert fingerprint watermark to individual frames.

From above discussion, the importance of simultaneously using space and time desynchronization is shown. In the following we will analyze the security limits of only space and only time desynchronization under SFC attack by two theorems. The first theorem shows that with only space desynchronization, the system is very probably to be attacked by SFC:

Theorem 1 (Collusion Probability for Space-only desynchronization under SFC attack). *Assume* $(P_1^n, P_2^n, ..., P_M^n)$, $n = 1, ..., N$ *are the* N *video copies and* P_m^n, $n = 1, ..., N; m \in [1, M]$ *are the frames which are with different space desynchronizing forms. Assume that* K *probable modes of space desynchronization (with equal probability) exist for each frame. The probability of collusion for each frame is no less than:*

$$1 - \frac{(K-1)(K-2)\cdots(K-N+1)}{K^{N-1}}.$$

Here collusion means that there exist at least one frame from other copies can be found which is the same to this frame if not considering the embedded watermark.

Proof. It is because probability that no frames pair are the same is: $\frac{K}{K} \cdot \frac{K-1}{K} \cdot ... \frac{K-N+1}{K}$. □

According to the theorem: firstly, when $K < N$ the probability of collusion is 1; secondly, the probability increases when K increases. For example, when $N = 100, K = 1000$ the probability is still 0.9940. Practically, the collusion probability is larger than this: on one hand space desynchronization is not totally random because continuity is needed for neighbor frame space desynchronization. On the other hand space desynchronization parameter space for a certain frame is limited (for example, the parameter space for a frame in [7] is $8 \times 8 \times 4 = 256$).

The second theorem indicates the average colluder number for time-only desynchronization under SFC attack.

Theorem 2 (Average Colluder number for Time-only desynchronization under SFC attack). *Assume the video sent to every user is composed*

of M frames, the video is desynchronized by inserting M_2 fractional frames and skipping M_2 frames, $M_1 = M - M_2$, total probable fractional frames are M' , c colluders attend the collusion, then the average colluder number of the video frames (the number of frames which are similar to this frame (itself included)) is:

$$CN = \frac{M_1}{M} \cdot CN_1 + \frac{M_2}{M} \cdot CN_2$$

where $CN_1 = \sum_{l=1}^{c} l \cdot \frac{M_2^{c-l} \cdot M_1^{l-1}}{M^{c-1}} \cdot C_{c-1}^{l-1}$, $CN_2 = \sum_{l=1}^{c} l \cdot \frac{(M'-M_2)^{c-l} \cdot M_2^{l-1}}{(M')^{c-1}} \cdot C_{c-1}^{l-1}$.

Proof. This theorem is direct when the following is considered:

Firstly, for an integer frame in copy one, the probability that the frame is also in copy two is $\frac{C_{M-1}^{M_1-1}}{C_M^{M_1}} = \frac{\frac{(M-1)!}{(M_1-1)!(M-M_1)!}}{\frac{M!}{M_1!(M-M_1)!}} = \frac{M_1}{M}$ and the frame is not in copy two is $1 - \frac{M_1}{M} = \frac{M_2}{M}$. For fractional frame, these two values are respectively: $\frac{M_2}{M'}$ and $\frac{M'-M_2}{M'}$.

Secondly, the probable colluder number for each frame is between 1 and N. Then average expected colluder number for each frame is the sum of the probability weighted colluder number from 1 to N.

Thirdly, there're M_1 integer frames and M_2 fractional frames in a certain copy. Then the average expected colluder number of a sequence is the number weighted sum of the expected colluder number of integer and fractional frames. □

Let $a = \frac{M_2}{M}$, $b = \frac{M_2}{M'}$, then CN is decided by a, b, c. Fig. 6 summarizes relation between CN and c when $a = 0.4, b = 0.1, 0.2, 0.4, 0.8$ and $a = 0.1, 0.2, 0.4, 0.8, b = 0.4$. From the figure, when a, b fixed, CN linearly increases proportional to c; when a is fixed and b increases (as in the left image), CN increases; when b is fixed and a increases (as in the right image), CN decreases.

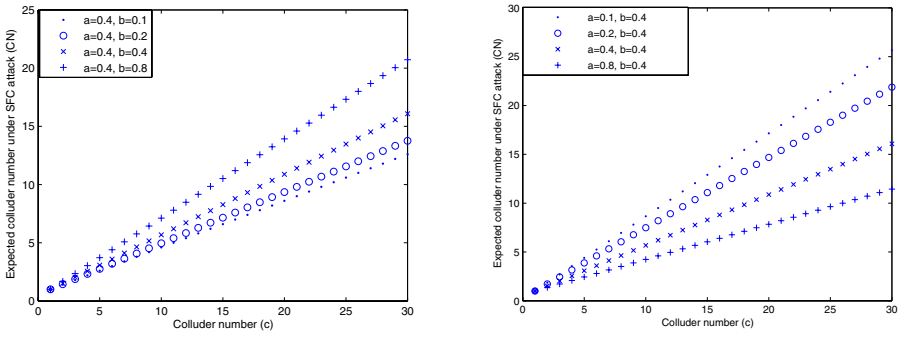


Fig. 6. Expected colluder number for SFC attack when only time desynchronization is used. Left: fixing a and changing b; Right: fixing b and changing a. Horizontal axis: colluders number; Vertical axis: Expected colluder number under SFC attack.

Attacks by Re-desynchronization (RDA). Desynchronization is a very interesting topic for watermarking. On one hand, it is one of the most annoying attack for watermarking [16] and until now there's no very effective method for it; on the other hand, desynchronization seems to be the most hopeful solving for fingerprint against collusion attack. Then a direct attack for desynchronized fingerprint is to re-desynchronize the desynchronized fingerprinted copy (RDA) (see Figure 7). In the following, we'll analyze the security of desynchronized fingerprint (by both oblivious and non-oblivious watermarking [6]) for RDA.

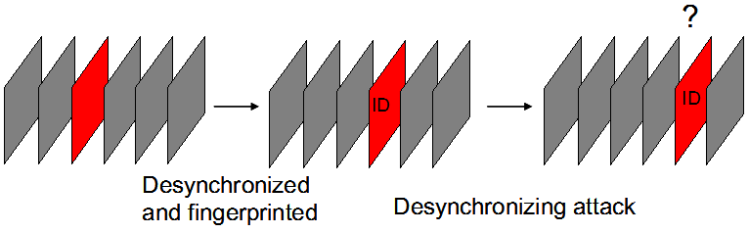


Fig. 7. Illustration for the re-desynchronization attack

For extraction convenience, the oblivious watermarking fingerprint [6] is obviously better than non-oblivious one because the extraction will not need the reference media. In fact, non-oblivious method will cause two questions: transmitting original media or desynchronizing parameters will cause security problem; transmitting additional information will cost more bandwidth. But considering the RDA, the first scheme is not so good. The RDA will make the embedded fingerprint watermark lose synchronization. A solving is to embed desynchronization robust watermarking such as those based on geometric transformation inversion, immune embedding space, synchronization-insensitive watermarking, synchronization marks, synchronization on content et al [16] [17].

For the non-oblivious watermarking fingerprint [6], if the derived copy is attacked by RDA, the media is firstly re-synchronized to the original media and then fingerprint is extracted. Assume the original copy is C_{ori}. If the derived copy C_f embedded by desynchronized fingerprint is attacked by RDA, the resultant C_{RDA} is registered to C_f [16] [18] for extracting fingerprint.

Principles for Ensuring Security. According to the above discussion, there are some security principles for desynchronized fingerprint:

1. Both space and time desynchronization are applied to the video copies (for robustness to MSFC attack);
2. Different keys are applied to neighbor frames (for robustness to SFC attack);
3. The keys of video should be robust to frame skipping and insertion [14] (for robustness to RDA and RSFR attacks);
4. The watermark embedded in the frames should be robust to geometrical distortion (for robustness to RDA attack).

4 Conclusion and Future Work

In this paper, an effective scheme is proposed for desynchronizing copies and embedding fingerprint during compression. Firstly, this scheme will maintain sufficient visual quality; secondly, this scheme does not diminish the computing efficiency; thirdly, the used bandwidth of the compressed desynchronized copies will be slightly less than original compressed copies. Experiments show the effectiveness of our method. Additionally, a concrete analysis to the security of desynchronized fingerprint is given, especially for the proposed attacks: re-synchronization attacks and re-desynchronization ones. The elementary principles for secure desynchronized fingerprint embedding are shown. Two theorems are given to indicate the security limit of only space or time desynchronization individually.

Compared with the scheme of devising fingerprint code for collusion robustness, research on desynchronization for collusion robust fingerprint has just begun. There're many aspects waiting for research: 1) in our scheme, fingerprint is embedded during (after desynchronization) or after compression. Because most watermark algorithms in compression domain or during compression are not very robust to re-compression, finding watermark technique robust enough is the next step research; 2) better method of boundary processing for space desynchronization is to be considered; 3) theoretical analysis to desynchronization, for example, by the information theory, is needed; 4) in MSFC attack, image registration has not been considered which will also be researched in the future.

References

1. Wu, M., Trappe, W., Wang, Z.J., and Liu, K.J.R.: Collusion-resistant fingerprinting for multimedia. IEEE Signal Processing Magazine **21(2)** (2004) 15-27.
2. Zhao, H.V., and Liu, K.J.R.: Resistance analysis of scalable video fingerprinting systems under fair collusion attacks. IEEE ICIP 2005, **III.** 85-88, Genova, Italy.
3. Wu, Y.D.: Linear Combination Collusion Attack and its Application on an Anti-Collusion Fingerprinting. IEEE ICASSP 2005, **II.** 13-16, Philadelphia, USA.
4. Su, J.K., Eggers, J.J., and Girod, B.: Capacity of digital watermarks subjected to an optimal collusion attacks. In European Signal Processing Conference (EUSIPCO 2000), 2000, Tampere, Finland.
5. Baaziz, N., and Sami, Y.: Attacks on Collusion-Secure Fingerprinting for Multicast Video Protocols. In International Conference of Distributed Frameworks for Multimedia Applications, DFMA 2005, 210-216, Besançon, France.
6. Celik, M.U., Sharma, G., and Tekalp, A.M.: Collusion-Resilient Fingerprinting by Random Pre-Warping. IEEE Signal Processing Letters **11(10)** (2004) 831-835.
7. Mao, Y.N., and Mihcak, K.: Collusion-Resistant Intentional De-Synchronization for Digital Video Fingerprinting. IEEE ICIP 2005, **I.** 237-240, Genova, Italy.
8. Yu, E., and Craver, S.: Fingerprinting with Wow. SPIE 18th Annual Symposium of Electronic Imaging, 15-19 January 2006, San Jose, California, USA.
9. Li, Z., and Trappe, W.: Collusion-resistant fingerprints from WBE sequence sets. IEEE International Conference on Communications 2005, **II.** 1336-1340, Seoul, Korea.

10. Swaminathan, A., He, S., and Wu, M.: Exploring QIM based Anti-Collusion Fingerprinting for Multimedia. SPIE Conference on Security, Watermarking and Stegonography, 15-19 January 2006, San Jose, California, USA.
11. Zhang, J.N., He, Y.W., Yang, S.Q., and Zhong, Y.Z.: Performance and complexity joint optimization for H.264 video coding. IEEE ISCAS 2003, **II.** 888-891, Bangkok, Thailand.
12. Zhang, J.N., Sun, L.F., Yang, S.Q., and Zhong, Y.Z.: Position Prediction Motion-Compensated Interpolation for Frame Rate Up-Conversion Using Temporal Modeling. IEEE ICIP 2005, **I.** 53-56, Genova, Italy.
13. Boneh, D., and Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Trans. Information Theory **44 (5)** (1998) 1897-1905.
14. Lin, E.T., and Delp, E.J.: Temporal Synchronization in Video Watermarking. IEEE Trans. Signal Processing **52 (10)** (2004) 3007-3022.
15. Cayre, F., Fontaine, C., and Furon, T.: Watermarking Attack: Security of WSS Techniques. IWDW 2004, 171-183, Seoul, Korea.
16. Licks, V., and Jordan, R.: Geometric Attacks on Image Watermarking Systems. IEEE Multimedia **12 (3)** (2005) 68-78.
17. Delannay, D.: Digital Watermarking Algorithms Robust against Loss of Synchronization. Ph.D. thesis, April 2004.
18. Barni, M.: Coping with Random Bending Attack by Means of Exhaustive Search Detection. In First Wavila Challenge, WACHA 2005, Barcelona, Spain, June 2005.

Lossless Data Hiding Using Histogram Shifting Method Based on Integer Wavelets

Guorong Xuan[1], Qiuming Yao[1], Chengyun Yang[1], Jianjiong Gao[1], Peiqi Chai[1], Yun Q. Shi[2], and Zhicheng Ni[2]

[1] Dept. of Computer Science, Tongji University, Shanghai, P.R. China
grxuan@public1.sta.net.cn
[2] Dept. of Electrical & Computer Engineering, New Jersey Institute of Technology
Newark, New Jersey, USA
shi@njit.edu

Abstract. This paper[1] proposes a histogram shifting method for image lossless data hiding in integer wavelet transform domain. This algorithm hides data into wavelet coefficients of high frequency subbands. It shifts a part of the histogram of high frequency wavelet subbands and thus embeds data by using the created histogram zero-point. This shifting process may be sequentially carried out if necessary. Histogram modification technique is applied to prevent overflow and underflow. The performance of this proposed technique in terms of the data embedding payload versus the visual quality of marked images is compared with that of the existing lossless data hiding methods implemented in the spatial domain, integer cosine transform domain, and integer wavelet transform domain. The experimental results have demonstrated the superiority of the proposed method over the existing methods. That is, the proposed method has a larger embedding payload in the same visual quality (measured by PSNR (peak signal noise ratio)) or has a higher PSNR in the same payload.

Keywords: Histogram Shifting, Lossless Data Hiding, Integer Wavelets.

1 Introduction

This paper focuses on the image lossless data hiding, which requires not only correct retrieval of the hidden data but also inverting the marked image back to the original cover image without any distortion.

Recently, Ni et al. [1,2] proposed an image lossless data hiding algorithm using pairs of zero-points and peak-points, in which the part of an image histogram is shifted to embed data. Independently, Leest et al. [3] proposed a similar method. However, both of these two methods are implemented in the spatial domain. It is

[1] This research is supported partly by National Natural Science Foundation of China (NSFC) on the project "The Research of Theory and Key Technology of Lossless Data Hiding (90304017)".

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 323 – 332, 2006.

well-known that the histogram distribution varies dramatically from image to image. Consequently, it is hard for these two methods to achieve high data embedding payload (often referred to as capacity as well) with a reasonably high visual quality (often measured by PSNR (peak signal to noise ratio)). Since the wavelet coefficients of high frequency subbands have Laplacian-like distribution, meaning that there is a high peak in the histogram around zero and small magnitudes on both sides, we propose to apply the histogram shifting technique in the wavelet domain. Because of the losslessness requirement, we chose to work in the integer wavelet transform domain.

During the shifting of histograms of high-frequency integer wavelet subbands, the overflow (e.g., the pixel grayscale value exceeding 255 for an 8-bit image) and/or underflow (e.g., the pixel grayscale value below 0 for an 8-bit image) may take place, thus violating the losslessness requirement. In order to overcome overflow and/or underflow, the histogram modification technique is adopted, which have been used in our previous works on image lossless data hiding using integer wavelet transform [4, 5, 6, 7, 8].

Experimental works have been conducted to compare the performance of this proposed new technique with that of the existing techniques [1, 2, 7, 9, 10], showing the superiority of the proposed technique.

The rest of the paper is organized as follows. The integer wavelet transform and histogram modification are introduced in Section 2. In Sections 3, the algorithm of wavelet histogram shifting is presented. Some experimental results are reported in Section 4. Conclusions are drawn in Sections 5.

2 Integer Wavelet Transform and Histogram Modification

2.1 Integer Wavelet Transform

Since it is required to reconstruct the original image with no distortion, we use the integer lifting scheme wavelet transform in this framework. Specifically, we adopt the CDF(2,2) and similar series used in JPEG2000 standard [11]. Table 1 below lists the forward and inverse transform of CDF(2,2) integer wavelet transform.

Table 1. CDF(2,2) integer wavelet transform

Forward transform	
Splitting:	$s_i \leftarrow x_{2i}$; $d_i \leftarrow x_{2i+1}$
Dual lifting:	$d_i \leftarrow d_i - \{(s_i + s_{i+1})/2\}$
Primary lifting:	$s_i \leftarrow s_i + \{(d_{i-1} + d_i)/4\}$
Inverse transform	
Inverse primal lifting:	$s_i \leftarrow s_i - \{(d_{i-1} + d_i)/4\}$
Inverse dual lifting:	$d_i \leftarrow d_i + \{(s_i + s_{i+1})/2\}$
Merging:	$x_{2i} \leftarrow s_i$; $x_{2i+1} \leftarrow d_i$

After integer wavelet transform, it has four sub-bands. We will embed the information into three high frequency subbands.

2.2 Histogram Modification

For a given image, after data embedding in some IWT coefficients, it is possible to cause *overflow/underflow*, which means that after inverse wavelet transform the grayscale values of some pixels in the marked image may exceed the upper bound (255 for an eight-bit grayscale image) and/or the lower bound (0 for an eight-bit grayscale image). In order to prevent the overflow/underflow, we adopt histogram modification, which narrows the histogram from both sides as shown in Figure 1. Please refer to [8] for the detailed algorithm. The bookkeeping information will be embedded into the cover media together with the information data.

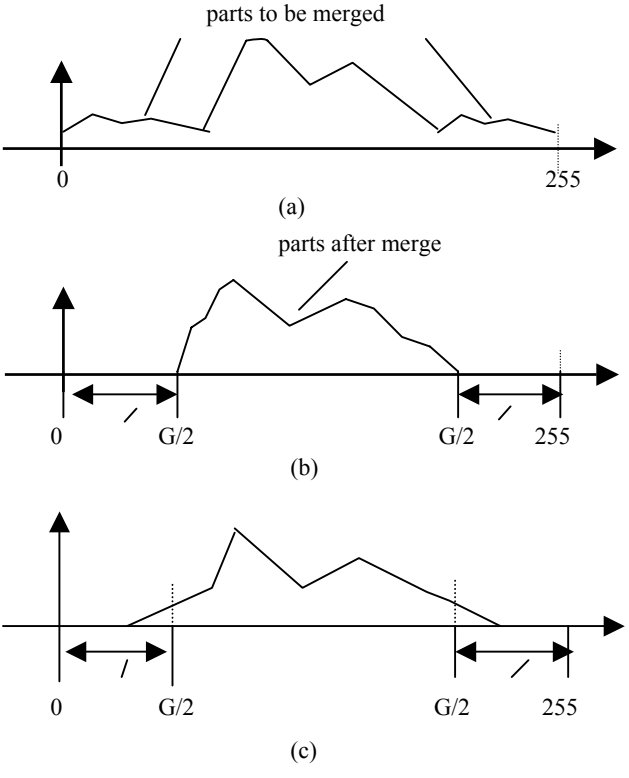


Fig. 1. Grayscale histogram modification: (a) original histogram; (b) modified histogram; (c) histogram after data embedding

3 Lossless Data Hiding Based on Integer Wavelet Histogram Shifting

3.1 Introduction to Wavelet Histogram Shifting

After integer wavelet transform, the histograms of high frequency subbands, referred to as wavelet histogram in the rest of this paper, are calculated. There the horizontal

axis represents the wavelet coefficients' value and the vertical axis the occurrence numbers of the corresponding wavelet coefficients. As mentioned, Ni et al. [1, 2] proposed the histogram shifting method in the spatial domain, while independently Leest et al. [3] proposed the histogram gap function method in the spatial domain.

In the following discussion, we consider a simple example shown in Figure 2 to demonstrate the principle of data embedding using histogram shifting. There, Figure 2 (a) is the original histogram of an integer wavelet high-frequency subband. In Figure 2 (b), a zero-point (no any coefficients in this subband assume this specific value: Z). That is, we shift the part of histogram with values larger than Z towards the right-hand side by one unit. It means the original Z+1 value now becomes Z+2, and the original Z+2 becomes Z+3 and so on. Another part of the histogram with the value less than and equal to Z remains unchanged.

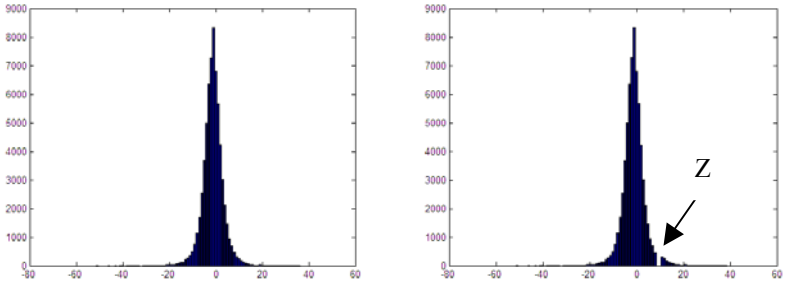


Fig. 2. An example showing how a zero point is generated: (a) original histogram (b) histogram after a zero point is created

In data embedding, we scan all of the IWT coefficients in the high-frequency subband. Once an IWT coefficient of value "Z" is encountered, if the to-be-embedded bit is "1", this coefficient's value will be added by 1, i.e., becoming "Z+1". If the to-be-embedded bit is "0", the coefficient's value remains to be "Z". The data extraction is actually the reverse process of data embedding. When an IWT coefficient of value "Z+1" is met, bit "1" is extracted and the coefficient's value reduces to "Z". When the coefficient of value "Z" is met, bit "0" is extracted. After all data have been extracted, the part of the histogram equal to or larger than "Z+2" needs to be shift towards the left-hand side by one unit. Clearly, the histogram shifting can also be carried out towards the left-hand side. Obviously, the payload is the occurrence number of coefficients having value "Z" in the histogram. Note that the sequence in which the wavelet coefficients are encountered in data embedding can be controlled by using a key in order to make hidden data secure. If the number of to-be-embedded bits is large, it usually needs multiple zeros and the corresponding shifting to accommodate the large payload.

The process of data embedding and data extraction illustrated above is summarized below. That is, we first shift the histogram shown in Figure 2 starting from value "Z+1" towards the right-hand one-by-one, leaving the value "Z+1" empty, i.e., creating a zero-point at "Z+1" in the histogram. Then according to the to-be-

embedded bit sequence, we either keep those coefficients having a value "Z" unchanged (if embedding a bit "0") or we change the coefficient from value "Z" to value "Z+1" (if embedding a bit "1"). During the data retrieval, we extract a bit "0" from those coefficients having value "Z". We extract a bit "1" from those coefficients having value "Z+1". Furthermore, we reduce the value of the coefficients from "Z+1" back to "Z". After all the hidden bits have been extracted out, we need to shift the part of the histogram larger than "Z+1" towards the left-hand side by one unit.

Since the histogram of IWT high frequency subbands obeys Laplacian-like distribution, the algorithm can embed data in both sides of the histogram alternatively until all the to-be-embedded bits are embedded. The proposed data embedding and data extraction algorithms are presented below in detail.

3.2 Data Embedding Algorithm

Assume there are M bits which are supposed to be embedded into a high frequency subband of IWT. We embed the data in the following way, as shown in Figure 3.

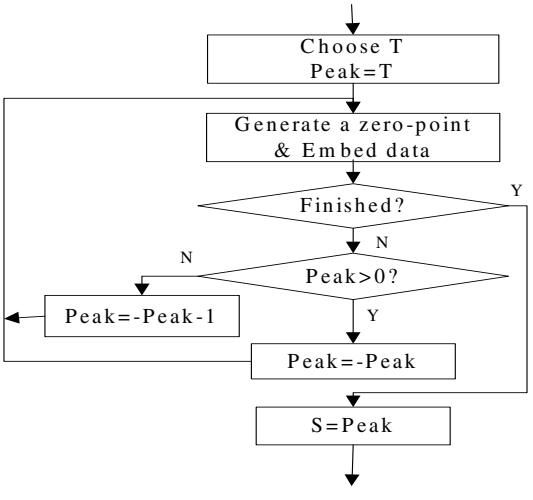


Fig. 3. Data embedding flowchart

(1) Set a threshold T>0, to let the number of the high frequency wavelet coefficients in [-T,T] is greater than M. And set the *Peak*=T.

(2) In the wavelet histogram, move the histogram (the value is greater than *Peak*) to the right-hand side by one unit to leave a zero-point at the value *Peak*+1. Then embed data in this point.

(3) If there are to-be-embedded data remaining, let *Peak* = (-*Peak)*, and move the histogram (less than *Peak*) to the left-hand side by 1 unit to leave a zero-point at the value (-*Peak*-1). And embed data in this point.

(4) If all the data are embedded, then stop here and record the *Peak* value as stop peak value, S. Otherwise, *Peak* =(*-Peak-1)*, go back to (2) to continue to embed the remaining to-be-embedded data.

3.3 Data Extraction Algorithm

Data extraction is the reverse process of data embedding. Assume the stop peak value is S, the threshold is T. Figure 4 is the data extraction diagram.

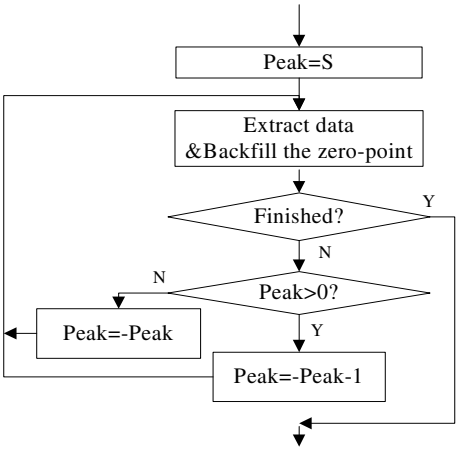


Fig. 4. Data extracting flowchart

(1) Set *Peak* = S.

(2) Decode with the stopping value *Peak*. (In what follows, assume Peak>0. If Peak <0, the decoding process runs correspondingly as shown in Fig. 4.) Extract all the data until *Peak*+1 becomes a zero-point. Move all the histogram (greater than *Peak*+1) to the left-hand by one unit to cover the zero-point.

(3) If the extracted data is less than M, set *Peak*= -*Peak*-1. Continue to extract data until it becomes a zero-point in the position (*Peak*-1). Then move histogram (less than *Peak*-1) to the right-hand side by one unit to cover the zero-point.

(4) If all the hidden bits have been extracted, stop. Otherwise, set *Peak*= -*Peak*, go back to (2) to continue to extract the data.

3.4 Threshold Selection and Wavelet Coefficient Modification Analysis

The payload with in the Peak equals to the number of IWT coefficients assuming the Peak value. Once threshold T is set, according to the algorithm, we will embed the data in the range of [-T, T]. Hence, the total payload is the total number of coefficients assuming the values in the range of [-T, T]. For instance, if T=5, then the possible zero-points will be 6, -6, 5, -5, 4, -4, … S, where S is the stop value. From threshold T and stop value S, we can calculate the total number of zero-points: $2*(T-|S|+1)-u(S)$, where u(.) is the unit step function. If there are to-be-embedded data left

when S =0, it shows that the range [-T, T] is not wide enough. We should increase the threshold T. If the payload is enough, different T will lead to different stop value S and PSNR of the marked image versus the original cover image. Table 2 shows that if we embed 0.02 bpp (bits per pixel) data into the Lena image, it achieves the highest PSNR when T =3. Hence we should choose the T which has the highest PSNR.

Table 2. Threshold T on stop value S and PSNR

Threshold T	Stop value S	PSNR
T=0	0	55.80
T=1	1	56.32
T=2	2	56.73
T=3	3	56.90
T=4	4	56.75
T=5	5	56.33

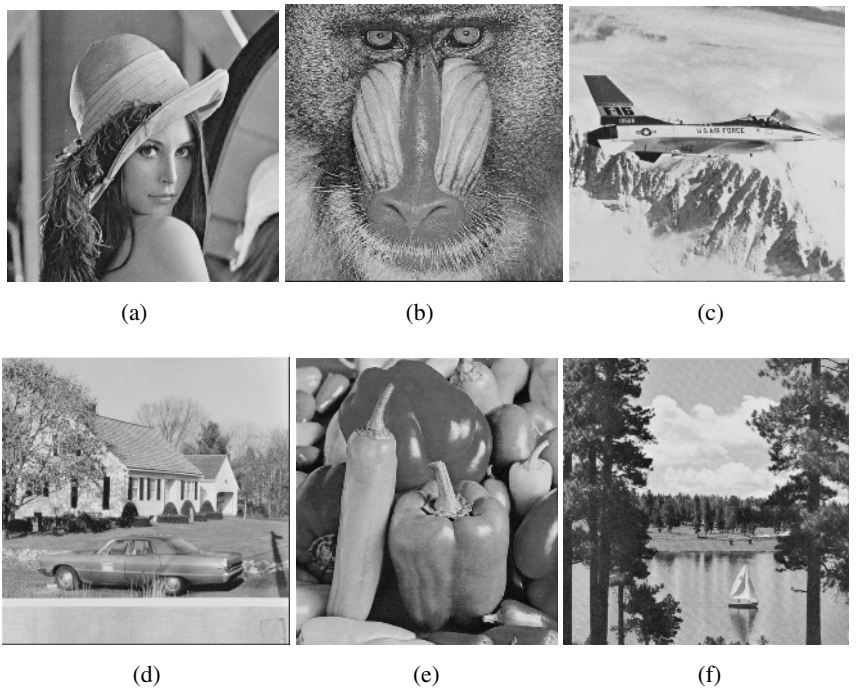
The modification of wavelet coefficients (without loss of generality, assume S>0 and W>0) can be classified into three categories. When wavelet coefficients W<S, the coefficients are intact. When $S \le W \le T$, the modification of wavelet coefficients is W-S (when embed 0) or W-S+1 (when embed 1). When coefficients W>T, the modification of W is T-S+1. From the above, we conclude that if the payload is small (S is close to T), majority coefficients are in category 1. Hence PSNR is high. If the payload is too large and leads S close to 0, the advantage of this proposed method is not obvious.

4 Some Experimental Results

Experiments on six frequently used images, i.e., Lena, Baboon, Airplane, House, Peppers, Sailboat are reported here. From Figure 5, we can find that the visual quality of the marked images is still acceptable when 131k bits are embedded into these grayscale images of 512x512x8, i.e., the embedding payload is 0.5 bpp.

Table 3 and 4 show the PSNR for different payload in the images Lena and Baboon. It shows that the increase of threshold T does not always lead to the increase of payload. When payload is smaller, we can choose larger threshold T. Hence fewer coefficients are changed during the data embedding and the resultant PSNR is higher. When payload is larger, the total number of needed zero points and threshold T also need to be larger.

Figure 6 depicts the performance comparison between our method and several other most advanced lossless data hiding methods, including Ni et al.'s method [1,2], Tian's Difference Expansion [9], Companding on integer DCT [10], and Threshold Embedding [7]. Note that the performance in terms of PSNR versus data embedding payload by threshold embedding [7] is superior to that achieved by the methods reported in [4, 5, 6]. Therefore, this indicates that our proposed method reported in this paper has the best performance in terms of PSNR versus payload, compared with these prior arts.



(a) (b) (c)

(d) (e) (f)

Fig. 5. PSNR of marked images with a payload of 0.5bpp: (a) Lena: 41.07 dB, (b) Baboon: 31.18 dB, (c) Airplane: 42.71 dB, (d) House: 40.90 dB, (e) Peppers: 39.71 dB, (f) Sailboat: 36.91 dB

Table 3. Experimental results on Lena image

Payload (bpp)	PSNR (dB)	Threshold T	Stop values S
0.1	50.71	4	3
0.15	49.14	2	1
0.2	47.77	1	0
0.3	45.08	2	-1
0.4	42.85	3	-1
0.5	41.07	4	0
0.6	39.38	5	0

Table 4. Experimental results on Baboon image

Payload (bpp)	PSNR (dB)	Threshold T	Stop value S
0.1	45.31	3	-2
0.15	42.19	4	-2
0.2	40.16	3	0
0.3	39.61	3	0
0.4	33.80	8	0
0.5	31.18	13	0

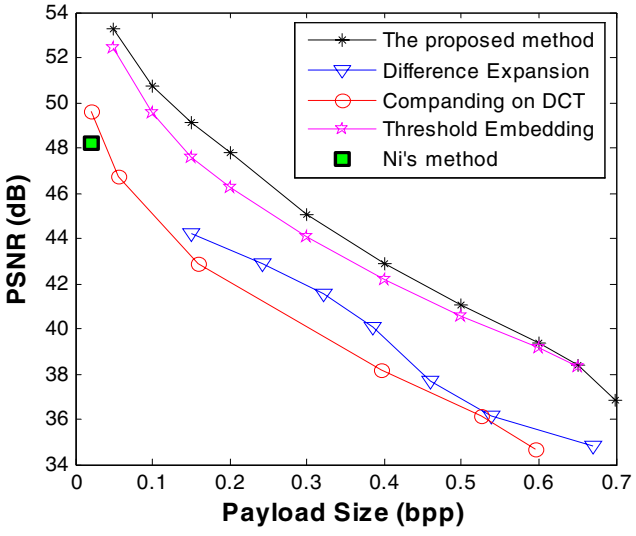


Fig. 6. Performance comparison of histogram shifting method v.s. several other most advanced lossless data hiding methods on Lena image

Table 5 and 6 are the detailed comparison results with Ni et al.'s method. Table 5 shows that our method has higher PSNR while the payload is same. Table 6 shows that the payload of our method is about four times that in Ni et al.'s method at the same PSNR. Hence, our method has better performance than Ni et al.'s method.

Table 5. PSNR for same payload

Images	Payload	PSNR (dB)	
		ours	Ni et al.'s
Lena	5,460	56.90	48.2
Airplane	16,171	53.60	48.3
Baboon	5,421	51.80	48.2
Peppers	5,449	55.29	48.2
Sailboat	7,301	52.79	48.2
House	14,310	53.74	48.3

Table 6. Payload for same PSNR

Images	PSNR	Payload (dB)	
		ours	Ni et al.'s
Lena	48.2	47,186	5,460
Airplane	48.3	65,536	16,171
Baboon	48.2	17,040	5,421
Peppers	48.2	39,321	5,449
Sailboat	48.2	31,457	7,301
House	48.3	57,664	14,310

5 Summary

This paper proposed a novel lossless data hiding method based on the histogram shifting, integer wavelet transform and histogram modification. The experimental results and theoretical analysis show that the proposed method has better performance than the similar methods in the spatial domain, integer DCT domain and integer wavelet domain. The proposed method has larger payload at the same PSNR. Especially, the proposed method has very high PSNR while the payload is small.

References

1. Z. Ni, Y. Q. Shi, N. Ansari and W. Su: Reversible Data Hiding. IEEE International Symposium on Circuits and Systems (ISCAS03), May 2003, Bangkok, Thailand.
2. Z. Ni, Y. Q. Shi, N. Ansari and W. Su: Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, pp. 354-362, March 2006.
3. A. Leest, M. Veen, and F. Bruekers: Reversible Image Watermarking. IEEE Proceedings of ICIP'03, vol.2, pp.731-734, September 2003.
4. G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni and W. Su: Distortionless Data Hiding Based on Integer Wavelet Transform. IEE Electronics Letters, vol. 38, no. 25, pp. 1646-1648, December 2002
5. G. Xuan, Y. Q. Shi, Z. Ni: Lossless Data Hiding Using Integer Wavelet Transform and Spread Spectrum. IEEE International Workshop on Multimedia Signal Processing (MMSP04), Siena, Italy, September 2004.
6. G. Xuan, Y. Q. Shi, Z. Ni: Reversible Data Hiding Using Integer Wavelet Transform and Companding Technique. Proceedings of International Workshop on Digital Watermarking (IWDW04), Korea, October 2004
7. G. Xuan, Y. Q. Shi, C. Yang, Y. Zheng, D. Zou, P. Chai: Lossless data hiding using integer wavelet transform and threshold embedding technique. IEEE International Conference on Multimedia and Expo (ICME05), Amsterdam, Netherlands, July, 2005.
8. G. Xuan, C. Yang, Y. Q. Shi and Z. Ni: High Capacity Lossless Data Hiding Algorithms. IEEE International Symposium on Circuits and Systems (ISCAS04), Vancouver, Canada, May 2004.
9. J. Tian: Reversible Data Embedding Using a Difference Expansion. IEEE Transactions on Circuits and Systems for Video Technology, Aug. 2003, 890-896.
10. B. Yang, M. Schmucker, W. Funk, C. Busch, S. Sun: Integer DCT-based Reversible Watermarking for Images Using Companding Technique. Proceedings of SPIE, Security and watermarking of Multimedia Content, Electronic Imaging, San Jose (USA), 2004
11. M. Rabbani and R. Joshi: An Overview of the JPEG2000 Still Image Compression Standard, Signal Processing: Image Communication 17 (2002) 3–48.

Analysis and Comparison of Typical Reversible Watermarking Methods

Yongjian Hu[1,2], Byeungwoo Jeon[1], Zhiquan Lin[2], and Hui Yang[2]

[1] School of Information and Communication Engineering
Sungkyunkwan University, Suwon Gyeonggi-do 440-746, Korea
bjeon@yurim.skku.ac.kr
[2] College of Automation Science and Engineering
South China University of Technology, Guangzhou 510641, PRC
eeyjhu@scut.edu.cn

Abstract. In sensitive imagery, such as deep space exploration, military reconnaissance and medical diagnosis, traditional watermarks can be hardly found useful. The main reason is that the users are too worried about the loss of original information after the image being embedded with other data. Although early watermarking methods only distort the host signal imperceptibly, there is still some host information that may be permanently (irreversibly) lost. To avoid this disadvantage, some researchers (e.g. [1]-[3]) proposed the concept of reversible (lossless) watermark. Recently, more and more reversible watermarking methods have been proposed. However, the influence of [1]-[3] is obvious. In this paper, we focus on analyzing and comparing these three reversible watermarking methods. Our investigation covers several aspects including data hiding capacity, image quality, capacity resilience and control, computational complexity, security, and blind data extraction. Such analysis and comparison provide indispensable information for the design of new reversible watermarking techniques.

Keywords: reversible watermarking, lossless watermark, data hiding, data compression.

1 Introduction

Traditional watermarking methods accomplish watermark insertion into the host signal by sacrificing imperceptible host information. For example, one can replace the LSB (least significant bit) plane of the host image with a fragile watermark [4]. But such a strategy is not suitable for some new application scenarios like remote sensing, military imagery and medical diagnosis, where authentication or data integrity verification is only allowed under the condition that no host information is permanently distorted.

As a solution to this problem, some researchers proposed the concept of reversible (lossless) watermark (e.g. [1]-[3]). However, as pointed out in [1], invertible authentication is not possible if we insist that all images, including "random" images, be authenticable. A real random image can not accommo-

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 333–347, 2006.

date an extra authentication message without permanent damage to some of the image bits. But the visual redundancy of natural images makes it possible to embed a weak imperceptible signal in the image making it capable of authenticating itself without accessing the original or other auxiliary data derived from the original.

One of early works was done by [5], where Honsinger *et al.* proposed to embed watermark bits using addition modulo 256. This hiding scheme works well as long as the pixel value is not close to upper or lower bound. However, if there are many pixels close to upper or lower bound, flipped pixels would cause annoying salt and pepper noise in watermarked images.

The real interest on reversible watermarking was raised by the publication of several papers, in particular, [1]-[3], where potential applications of reversible watermarks are clearly exhibited; moreover, not only principles but also embedding schemes are described. In their well-known paper [1], Fridrich *et al.* proposed three reversible data embedding methods including robust watermarking, fragile watermarking, and compressed domain watermarking. Later they [2] proposed another method called the RS method, where they are able to embed message bits by manipulating the LSB (least significant bit) plane, even in images where there is not any structure in their LSB plane. Such a scheme is significant to retain high image quality. It motivates many other later works (e.g. [6] and [10]). Unlike Fridrich's spatial domain methods, Tian explored a new way to perform reversible data hiding. He [3] proposed a transform domain method, which can achieve large data hiding capacity by means of modifying and replacing the integer wavelet transform coefficients.

The influence of [1]-[3] can be easily found in literature. For example, Celik *et al.* [6] enhanced Fridrich's LSB modification scheme and proposed a generalized-LSB data hiding algorithm. It allows two or more LSBs overwritten to suit a requirement of bit rate per sample. Their latest work [7] is also based on [2] and [6]. In [8], Alattar extended Tian's algorithm in [3] and proposed to use difference expansion of vectors instead of pixel pairs. In [9], Tian refined his previous work in [3]. Instead of changing the bits after the MSB (most significant bit) bit in the binary representation of a difference number, he changed the LSB bit. The well-defined difference sets help [9] achieve both good capacity and image quality. In [10], Kamstra *et al.* presented two techniques. The first one improved Fridrich's work in [2]. It used the information of neighboring pixels and a variant of Sweldens' lifting scheme to predict the LSB plane. The second one further developed the work in [9]. To lessen overhead cost, they sorted the pixel pairs before embedding, and then, from the capacity control point of view, they only choose necessary locations for embedding. These two schemes are beneficial to image quality while satisfying the requirement of the payload. Some other works like [11]-[15] are not direct extensions of [1]-[3], but they still benefited from the principles and schemes proposed by [1]-[3]. So far, there are grossly two types of techniques for reversible watermarking, one is based on compression (e.g. [1]-[3], [6]-[11]); the other is based on the modification of signal features (e.g. [14]). [1]-[3] are typical methods of the first type.

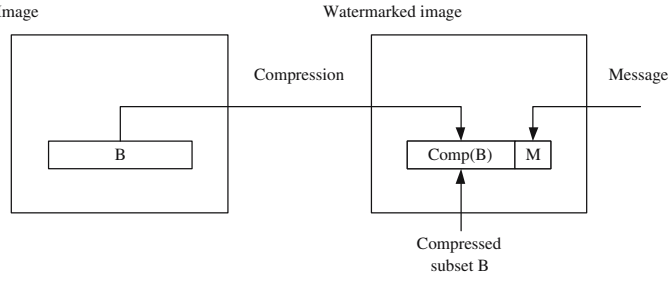


Fig. 1. Diagram for Fridrich's method using lossless bit-plane compression

In this paper, we analyze and compare the classic works in [1], [2] and [3]. Our investigation covers several aspects including data hiding capacity, image quality, capacity resilience and control, computational complexity, and security. The paper is organized as follows. Sections 2, 3 and 4 first review and then analyze [1], [2] and [3], respectively. Section 5 gives the comparison of the three methods. The conclusion is drawn in Section 6.

2 Analysis of Invertible Authentication Using Lossless Bit-Plane Compression

2.1 Overview

In [1], Fridrich *et al.* proposed three authentication methods: invertible authentication using robust watermarks, invertible authentication using lossless bit-plane compression, and authentication of JPEG files. Due to length limitation, we only address their second method. The main idea of using lossless bit-plane compression can be described in Fig.1 (refer to [16]). It includes three steps: (i) determine the key bit-plane B; (ii) concatenate the compressed B, the hash value and the padding random bits into a stream; and (iii) replace the key bit-plane with the combined bit stream.

2.2 Analysis

We first investigate its data hiding capacity, image quality and capacity resilience. The redundancy (spare space) obtained from compression on the key bit-plane is referred to as data hiding capacity. Actually, it is the maximum capability to accommodate the secret data. The payload is authentication message M of 128 bits, which may be the output of the hash function MD5. Ideally, M is embedded into the region B without resulting in any perceptual distortion. The bit stream to be embedded is better composed of only M and the compressed B. Unfortunately, Fridrich's algorithm takes bit-plane-wise watermark embedding. In other words, the whole key bit-plane will be changed for embedding, no matter whether the redundancy is equal to or greater than the payload M. It

Fig. 2. Watermarked images using bit-plane-based scheme

Table 1. Performance evaluation on Fridrich's bit-plane-compression-based algorithm. The level of bit-plane is from 0 (the LSB plane) to 7 (the MSB plane). N_{kb} represents the key bit-plane. R_{kb} represents the redundancy on the key bit-plane. R_{kbl} represents the redundancy on the bit-plane immediately below. "-" means that the compressed data size is larger than the original data size. The unit of R_{kb} and R_{kbl} is byte.

Image	N_{kb}	R_{kb}	R_{kbl}	$PSNR(dB)$
Lena	3	3085	-1095	33.08
Girl	3	2351	-1088	33.21
Man	3	1059	-1006	33.22
$F-16$	2	359	-1509	39.68
Baboon	4	311	-1394	27.06
Peppers	4	7868	-30	27.06
Sailboat	4	4014	-110	27.09

implies that the capacity is not assigned according to the size of the payload and some spare space is most probably not used. Fridrich *et al.* proposed to pad the surplus spare space with random bits. So, besides M and the compressed B, the final combined bit stream contains some padding bits because of the deficiency of the embedding scheme. Such a scheme apparently harm visual quality. Due to noise-like structure, the key bit-plane is often one level higher than the LSB plane. So the negative impact on image quality is obvious.

Another disadvantage of the algorithm is the abrupt change of redundancy on different bit-planes. For example, the redundancy on one bit-plane is far below a payload, but on its immediately higher bit-plane, it is probably several times larger than the required. This could cause great change in the performance of the algorithm with different payload sizes.

We test Fridrich's algorithm on a set of gray images of $512 \times 512 \times 8 bits$. The JBIG implementation is from ImageMagicK. The experimental results are shown in Table 1 and Fig. 2. From Table 1 we observe that, although the authentication message is 16 *bytes* (128 *bits*), the key bit-plane often has the hiding capacity (redundancy) tens, even hundreds of times of the payload. We also note that the redundancy difference between the key bit-plane and its lower neighbor is enormous. This problem can not be solved by simply choosing different JBIG coders because compression efficiency depends on both the coder and the input data. Table 1 confirms our analysis on image quality. It shows that the watermarked image quality is poor and often below 40 dB. Though the recovery process allows reconstruction of the original host signal with no distortion, it is still desirable to keep the embedding distortion to a minimum, so that applications that do not have access to the extraction and recovery process do not incur a heavy penalty in image quality [6].

Table 1 also shows that the performance of this algorithm varies greatly with different images. Usually, smooth images (e.g. F-16) choose a low level bit-plane as the key bit-plane, and thus, have relatively higher PSNR (peak signal to noise ratio), whereas, textured images (e.g. baboon) choose a high bit-plane as the key bit-plane and have lower PSNR.

Now we investigate watermark security and blind data extraction. Since the combined bit stream is encrypted before embedding, the watermark security is guaranteed by cryptography. Data extraction proceeds in an inverse order of embedding. After decrypting the key bit-plane, the algorithm relies on the 20-byte JBIG file header to get back the size of the compressed B. The following 128 bits are the authentication data. We can reconstruct the original key bit-plane through decompressing the compressed B. Apparently, blind data extraction benefits from the use of key bit-plane and JBIG coding. It does not need any other synchronization signal.

The main contribution of [1] is that it presents a framework of implementing compression-based reversible watermarking. It is worth mentioning that their third method about authentication of JPEG files, which we do not discuss in this paper, even explains how to adapt the capacity to the payload so as to facilitate the blind data extraction.

3 Analysis of the RS Lossless Data Embedding Method

3.1 Overview

In [2], Fridrich *et al.* proposed another distortion-free data embedding method called the RS method. Unlike their previous bit-plane-based method, this method depends on the property of each pixel group for data hiding. Before embedding, the image is divided into disjoint groups of n adjacent pixels. Then, two functions, the discrimination function f and the flipping function F, are respectively defined. Using f and F, all pixel groups in the image are classified into three groups: R (regular groups), S (singular groups) and U (unusable groups). By turning R groups into S groups or vice versa, binary watermark bits {0,1} are

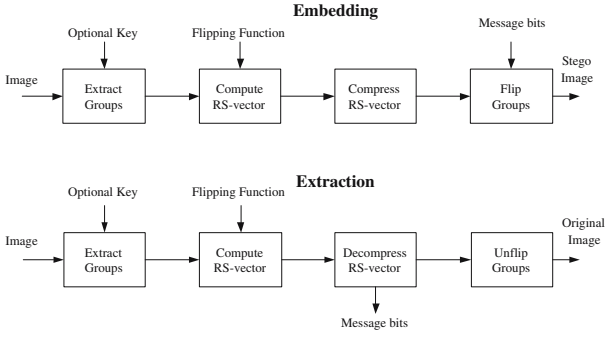


Fig. 3. Diagram for Fridrich's RS lossless data embedding method

embedded into the image. Embedding only takes place in R and S groups. U groups are not used in the process of watermarking. Fig. 3 shows the framework of this algorithm.

3.2 Analysis

Implicitly, the discrimination function f reflects the correlation between the pixels in a group G. If those pixels are strongly correlated, the sum of absolute difference between those pixel values is small. The stronger the correlation is, the smaller the sum. When all pixel values are identical, the sum reaches its minimum, i.e., $f(G) = 0$. On the other side, from communication point of view, embedding can be regarded as adding noise into transmission channel, so it may decrease the correlation between neighboring pixels in the image. Since embedding is implemented by performing F on G, it often increases the value of the discrimination function. So, in [2], the group with $f(F(G)) > f(G)$ is defined as R (regular) group.

The flipping function F is a two-cycle permutation operation on gray values. Use of F will not cause overflow or underflow of gray values. The embedding based on F is one of advantages of this method. However, the flipping amplitude A directly affects the image quality. Large A would severely degrade the image quality. But performing F on LSB plane, i.e. A=1, still achieves a reasonable capacity. It is the other advantages of this method.

Now we discuss its data hiding capacity. The RS method denotes "1" to R groups and "0" to S groups to perform binary insertion, so the capacity depends on both the number of R groups (N_R) and that of S groups (N_S), more exactly, the bias between N_R and N_S. The data hiding capacity Cap is calculated by $N_R + N_S - |C|$, where C is the compressed bit stream of the image status, i.e. the RS-vector, and $|\cdot|$ represents the length of a stream. According to Shannon's entropy coding theory, as the bias between N_R and N_S increases, $|C|$ becomes shorter due to more efficient compression. For a random image, N_R and N_S are very close, and the bias approaches 0, so there is hardly any redundancy. On the

Table 2. Performance evaluation on Fridrich's RS method. Negative symbol "-" means that it needs the spare space. The unit of N_{RS} and Cap is bit. $n = 4$.

		Lena	*Girl*	*Man*	$F-16$	*Baboon*	*Peppers*	*Sailboat*
A=1	N_{RS}	41902	42433	42310	41351	44599	44326	44589
	Cap	1726	1209	1510	4535	-241	414	173
	PSNR	53.05	53.02	53.02	53.16	-	52.83	52.80
A=2	N_{RS}	45953	46280	45743	46124	46184	47396	46912
	Cap	7313	5680	5591	12116	864	3428	2424
	PSNR	46.65	46.66	46.69	46.65	46.66	46.54	46.58
A=3	N_{RS}	48537	49359	47410	46722	47672	49476	48360
	Cap	13713	11159	9786	17506	2304	8028	4952
	PSNR	42.93	42.91	42.99	43.02	42.97	42.82	42.91
A=4	N_{RS}	49623	49742	48186	45840	48478	50959	49384
	Cap	18655	16934	13178	20144	3950	12415	7744
	PSNR	40.28	40.28	40.44	40.67	40.39	40.18	40.35
A=5	N_{RS}	49382	50468	48534	43874	49337	51321	49732
	Cap	21734	18708	16550	21554	5721	15705	10844
	PSNR	38.39	38.50	38.42	38.91	38.41	38.22	38.37
A=6	N_{RS}	48327	49126	47698	42260	49853	51261	49888
	Cap	24479	20070	17466	22604	7437	19717	13168
	PSNR	36.89	36.96	36.93	37.48	36.76	36.65	36.77
A=7	N_{RS}	47123	47676	47197	40261	50413	50904	49648
	Cap	26075	21620	20221	22733	8741	22896	14768
	PSNR	35.69	35.76	35.66	36.36	35.36	35.35	35.41

contrary, smooth images can provide larger capacity than textured images due to stronger correlation among pixel groups.

Besides the influence of image characteristics on capacity, the group size n also affects capacity. It seems that small n would bring about larger number of groups, and thus, larger capacity. However, as stated in [2], small groups only generate small bias between N_R and N_S. We suppose that too few pixels in a group can not reflect the characteristics of a region, and thus, prevent the use of the property of each group, which is the foundation of this algorithm. Fridrich *et al.* found that a moderate size of n (n=4) is a good choice. Since the algorithm flips all pixels in a group for the embedding of one message bit, the group size would also affects image quality. Large group size surely raises large artifacts.

We test the RS method on the same image set as in subsection 2.2. The experimental results are shown in Tables 2 and 3 and Fig. 4. The entropy coder we use to compress the RS-vector is the new version of CACM coding

in [17]. Tables 2 and 3 exhibit that, when the flipping amplitude A is not large (e.g., $A \leq 7$), A has stronger influence on Cap than on the sum of N_R and N_S, i.e., N_{RS}. In other words, A has more influence on the bias between N_R and N_S. Large A often leads to large capacity. It can be observed that the capacity increases rapidly from A=1 to 5 but slowly from A=6 to 7. The sum N_{RS} reaches the maximum around A=4 or A=5. When A=6 in Table 2 and A=4 in Table 3, although the capacity still increases, N_{RS} probably begins to decrease. When A passes some value, the capacity begins to decrease. Table 3 shows that the capacity in A=19 is less than that in A=7. Generally, the watermarked images become noisier as A increases.

Table 3. Performance evaluation on Fridrich's RS method. Negative symbol "-" means that it needs the spare space. The unit of N_{RS} and Cap is *bit*. $n = 2$.

		Lena	*Girl*	*Man*	$F-16$	*Baboon*	*Peppers*	*Sailboat*
A=1	N_{RS}	53782	52982	54806	49840	60826	57019	58369
	Cap	974	758	878	3256	-518	187	-103
	PSNR	55.02	55.07	54.94	55.36	-	54.75	-
A=2	N_{RS}	55399	54103	55966	53048	60968	57409	58767
	Cap	5343	4511	3926	9608	296	2793	1935
	PSNR	48.84	48.96	48.79	49.07	48.43	48.72	48.61
A=3	N_{RS}	56499	57077	56313	52969	61406	58681	59126
	Cap	10211	9117	7225	14441	1526	6697	3926
	PSNR	45.23	45.32	45.27	45.49	44.88	45.07	45.04
A=4	N_{RS}	56567	55855	56210	51316	61596	59268	59352
	Cap	14543	14327	9490	16788	2508	10764	6352
	PSNR	42.73	42.79	42.74	43.18	42.37	42.51	42.58
A=5	N_{RS}	55589	57208	55137	48526	61610	59575	59316
	Cap	17101	16256	12601	17854	4058	14047	9460
	PSNR	40.88	40.99	40.93	41.47	40.46	40.59	40.64
A=6	N_{RS}	53611	54135	53586	46209	60839	58828	58608
	Cap	19555	17567	13810	18825	5471	17476	11128
	PSNR	39.47	39.54	39.46	40.12	38.90	39.03	39.07
A=7	N_{RS}	51591	52216	52439	43545	60785	58157	58101
	Cap	20655	18648	15519	18913	6425	20917	12381
	PSNR	38.33	38.36	38.19	39.03	37.58	37.74	37.78
A=19	N_{RS}	30779	38274	36485	25096	53297	36060	45907
	Cap	18371	18122	18181	12968	17049	24388	24371
	PSNR	31.87	31.77	31.09	32.82	29.45	31.14	30.12

Fig. 4. Watermarked images using Fridrich's RS method. A=1 (left) and 7 (right). $n = 4$. Lena (upper row) and Girl (lower row).

The characteristics of the image apparently affect the performance of the method. Generally, textured images can provide much less spare space than other images. For example, *baboon* (when A=1) in Table 2 and *baboon* and *sailboat* (when A=1) in Table 3 can not provide any spare space for data hiding at all.

As for the influence of group size, it can be observed that the capacity under n=4 is larger than that under n=2 though N_{RS} in the former situation is less than that in the latter situation. On the other side, the image quality in the former situation is inferior to that in the latter situation.

Fig. 4 also shows that the group-based embedding scheme yields more obvious embedding errors on smooth images like *Girl* than on other images like *Lena*.

Tables 2 and 3 show that the RS method yields both good PSNR values and good capacity resilience in data hiding. When capacity increases, the PSNR values gradually decrease, that is, the image quality gradually degrades. But we have to point out that the image quality measurement like the PSNR is not very sensitive to the flipping operation in the RS method. Generally, the PSNR or MSE (mean-squared-error) can only reflect the average error in an image.

In the RS method, watermark security may be obtained by using common encryption algorithms to encrypt the compressed RS-vector C and the message

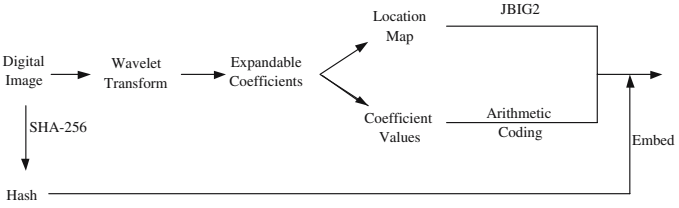


Fig. 5. Diagram for Tian's reversible watermarking algorithm

bits before embedding. As for capacity control, Fridrich *et al.* did not discuss this problem. However, one can borrow the idea of the third method in [1] to realize the economical way of using spare space. Generally, capacity control can be related to blind data extraction. Using only required spare space is not only good for image quality but also for the blind data extraction. Readers can refer to [1] for detailed information. The other way to implement blind data extraction is setting flag bits to record the beginnings and ends of both C and message bit-stream. Since the capacity is usually large, this strategy is possible.

The main contribution of [2] is that it presents a way to embed large amount of message bits while keeping low image distortion. It depends upon the change of the state of pixel groups for embedding. On the other hand, changing the state of a group can be easily realized through low bit-plane manipulation, even LSB plane manipulation.

4 Analysis of Tian's Reversible Watermarking Method

4.1 Overview

In [3], Tian proposed a reversible watermarking method based on modifying the difference between a pair of pixel values while keeping the average of them unchanged. The whole image is divided into disjoint pixel pairs in the row or column direction and perform the so-called integer wavelet transform as follows.

$$l_i = \lfloor \frac{x_{2i} + x_{2i+1}}{2} \rfloor \tag{1}$$

$$h_i = x_{2i} - x_{2i+1} \tag{2}$$

where l_i and h_i represent the average of and the difference between the two pixels in a pair, respectively. The method can be conceptually depicted in Fig. 5. From the frequency transformation point of view, l_i is the low frequency component of the host signal while h_i is the high frequency component. Tian's method only changes the high frequency component.

4.2 Analysis

Since neighboring pixels in natural images are strongly correlated, the author of [3] proposed to slightly modify the difference number (high frequency coefficient

Table 4. Performance evaluation on Tian's method. N_{ep} represents the number of expandable pixel pairs. N_{ecb} represents the number of changeable bits plus the number of expanded bits of expandable h. Cap represents the capacity for hiding data besides the compressed location map and the compressed changeable bits. bpp represents the bit rate per pixel. The unit of Cap is bit. $|h|_{min}$, $|h|_{max}$ and $|h|_{mean}$ represent the minimum, the maximum and the mean of the absolute of expandable h, respectively.

	$Lena$	$Girl$	Man	$F-16$	$Baboon$	$Peppers$	$Sailboat$
N_{ep}	130468	119359	129809	129688	127744	128993	129567
N_{ecb}	351320	337195	377062	320735	497820	378157	292616
Cap	130432	40520	65704	51536	83088	77776	83664
bpp	0.49	0.15	0.25	0.19	0.31	0.29	0.32
$\|h\|_{min}$	1	1	1	1	1	1	1
$\|h\|_{max}$	66	69	127	114	121	104	90
$\|h\|_{mean}$	5.66	4.90	6.92	4.99	13.69	6.03	8.71
PSNR	31.94	33.67	30.57	32.11	26.38	32.24	29.32

h) between a pair of pixels for embedding. However, such manipulation may cause overflow or underflow, and thus, causes irreversible distortion to the image. To overcome this problem, all pixel pairs are classified before embedding into two types: "expandable" and "non-expandable". Only expandable h has its binary representation inserted with 1 bit, that is, the length of $|h|$ is increased by 1.

Although Tian used expandable h to explain the principle of data embedding, his implementation of reversible watermarking is not confined on expandable h. In practice, to facilitate blind date extraction, Tian defined changeable h. The constraint on changeable h is much loose than that on expandable h. Before embedding, each h is judged by a rule (Definition 4.2 in [3]) to determine whether it is changeable. Embedding only takes place in all hs with changeable bits, no matter whether h is expandable or non-expandable. Like the compression-based method in [2], the combined bit stream consists of three components: the authentication message bits, the JBIG-compressed location map, and the arithmetically compressed changeable bit stream (refer to Fig. 5), where the second and third components are used for locating expandable h and recovering the original changeable bit stream, respectively. The embedding is achieved by replacing all changeable bits with the above combined bit stream.

Tian avoided expanding the pixel pair with large difference number h in that such a pixel pair could lead to overflow or underflow. In essence, non-expandable pixel pairs often correspond to edges or rapidly changing image areas, which only cover a small portion of a natural image, so the number of non-expandable pixel pairs is small. This indicates that Tian's algorithm has large capacity. On the other side, the modification of h value could degrade the image apparently. When $|h|$ is small, the error from expanding $|h|$ and replacing the changeable bits is small; but when $|h|$ is large, the impact is severe. According to the definition of expandable pixel pair, $|h|$ theoretically varies in the interval of [1,127]. Therefore,

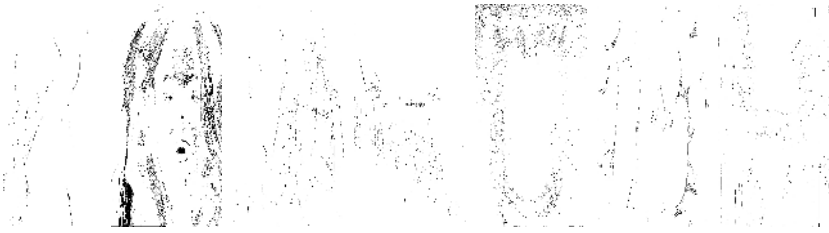


Fig. 6. From left to right, the location map of *Lean*, *Girl*, *Man*, $F-16$, *Baboon*, *Peppers* and *Sailboat* in Tian's algorithm. The binary image of location map is of 512×256. Brightness ("1") represents "expandable" pixel pairs, whereas, darkness ("0") represents "non-expandable" pixel pairs.

the embedding distortion is expected to be high. It is worth mentioning that Tian did not discard the pixel pairs with $h = 0$. Although one restriction in the definition of expandable h is $h \neq 0$ (Definition 4.1 in [3]), Tian modified h as $h = h + 1$ for all $h \geq 0$ before classifying h. After this preprocessing, there are only two types of pixel pairs in the image: $h \leq -1$ or $h \geq 1$. Tian let $h = h - 1$ for all $h \geq 0$ just before reconstructing the watermarked image. Actually, the pixel pairs with $h = 0$ correspond to flat regions where both pixel values in the pair are qual to each other.

We test Tian's algorithm on the same image set as in subsection 2.2. The integer wavelet transform is performed in the row direction. The experimental results are shown in Table 4 and Figs. 6 and 7. We still use the entropy coder in [17] and JBIG from ImageMagicK. Tian did not discuss the capacity control in that paper. Therefore, in order to test its performance under full payload, we pad the surplus spare space unused by the combined bit stream with random bits. Table 4 indicates that the (maximum) capacity is quite large, but the image quality is not satisfactory, and the PSNR values are around 30 dB.

The location maps in Fig. 6 verify our analysis about the characteristics of non-expandable h. Most non-expandable pairs appear in edges (see *Lena*) or rapidly changing areas (see *Baboon*). However, the location map of *Girl* surprises us. There are lots of non-expandable pairs in flat areas. We notice that one characteristic of those flat areas is high brightness. It means that the pixel values are close to 255. After examining the definition of expandable pixel pairs, we get the answer. In fact, the definition prevents extreme pixel pairs (e.g. x=253 and y=255) from being used.

Another finding is also involved in capacity. Table 4 shows that textured images (e.g. baboon) may have larger capacity than smooth images (e.g. $F-16$). This finding is different from those in Fridrich's methods. Generally, compression-based methods achieve higher capacity in smooth images than in textured images due to higher correlation among pixels in the former images. The Fridrich's methods verify this conclusion. Why is this conclusion not suitable for Tian's algorithm? Further study discovers that, although Tian's algorithm is based on compression, its capacity is also affected by the number of changeable bits. In smooth images, many pixel values are equal to each other. Tian used the

Fig. 7. Watermarked images using Tian's method

mentioned preprocessing scheme ($h = h + 1$) to make these pixel pairs expandable, but the number of changeable bit is 0. After $|h|$ is expanded by one bit, the number of changeable bit becomes 1. However, the whole length of the changeable bit stream is still shorter in smooth images than in textured images. For example, the number of changeable bits is 320735 (N_{ecb} in Table 4) in $F - 16$ but 497820 in *Baboon*. Experiments also shows that, if we do not use the pixel pairs with $h = 0$, the embedding bit rate will drop sharply, for example, from 0.49 *bpp* to 0.22 *bpp* in *Lena*. However, use of these pixel pairs also brings about a problem. It makes the embedding noise more visible in smooth image areas (see Fig. 7).

Image quality is affected by the change of the absolute of difference number h. There is direct link between the embedding noise and the mean of $|h|$. Table 4 exhibits that the larger the mean is, the lower the image quality.

The security of Tian's method is ensured by encryption algorithms. When decoding, Tian first retrieved all the changeable bits using Lemma 4.3 (refer to [3]). Then, from the decoding of the location map, the expandable hs are distinguished. Tian did not describe the exact process of blind data extraction. Intuitively, the three components of the embedded bit stream may be known via the flag bits set during embedding.

The main contribution of [3] is that it presents a transform-based reversible data hiding technique. Due to the use of transform domain image features, data hiding capacity is greatly increased.

5 Comparison of Three Reversible Watermarking Methods

Among the three algorithms, Tian's algorithm has the largest data hiding capacity; Fridrich's RS method produces the best quality of watermarked images; Fridrich's bit-plane-compression-based algorithm is inferior to the above two algorithms in most aspects.

In particular, we compare Tian's algorithm and Fridrich's RS method. When $n = 2$, a pixel group in Fridrich's RS method has the same size as a pixel pair in Tian's algorithm. To make a more fair comparison between their capacities, we also let the watermarked image quality of Fridrich's RS method be close to that of Tian's algorithm. The change in image quality can be achieved by increasing the flipping amplitude A. When A=19, their image quality is very near. Tables 3 and 4 show that the capacity of Tian's algorithm is far above that of Fridrich's RS method. For example, in image *Lena*, the embedding bit rate is 0.49 *bpp* for Tian's algorithm but 0.07 *bpp* for Fridrich's algorithm. Although the increase of A may increase the capacity, Table 3 demonstrates that it is impossible for Fridrich's algorithm to reach the capacity as large as that of Tian's algorithm. On the other side, Tian' method achieves large capacity at the cost of image quality. Table 4 shows that the interval between $|h|_{min}$ to $|h|_{max}$ of expandable $|h|$ is large.

Among the above three methods, computational complexity increases from Fridrich's bit-plane-compression-based algorithm to Fridrich's RS method and to Tian's algorithm.

6 Conclusions

Although all of the three methods are based on data compression, they show quite different ways to achieve reversible watermarking. The principles and data hiding schemes conveyed by these methods are very instructive. Their influence can be easily found in many later works. In fact, the extensions of the three methods by other researchers have yielded many good and more sophisticated algorithms.

So far, the efforts on analyzing and comparing typical reversible watermarking methods are rare in literature. Our work tries to benefit the design of future reversible watermarking algorithms by exhibiting the advantages and weaknesses of these classic ones. We focus on discussing data hiding capacity and image quality though we also address computational complexity, security and blind data extraction. For the sake of simplicity and clearness, we purposely neglect some security measures taken by the original algorithms while implementing the algorithms. It does not affect our conclusions.

Acknowledgement. This work was supported in part by the NRL program 2006-0397-000 of MOST, Seoul R&BD program (2005-0843-000) through SFCC, NSF of China 60572140, and NSF of Guangdong 04020004.

References

1. Fridrich, J., Goljan, M., and Du, R.: Invertible authentication. Proc. of SPIE 2001, Security and Watermarking of Multimedia Contents III,. Editor(s): Wong, P.W., Delp, E.J., Vol. 4314, pp. 197-208.
2. Fridrich, J., Goljan, M., and Du, R.: Distortion-free Data Embedding for Images. Proc. 4th Information Hiding Workshop, Pittsburgh, Pennsylvania, Apr. 25-27, 2001.

3. Tian, J.: Wavelet-based reversible watermarking for authentication. Proc. of SPIE 2002, Security and Watermarking of Multimedia Contents III. Editor(s): Wong, P.W., Delp, E.J., Vol. 4675, pp. 679-690.
4. Wong, P.W., and Memon, N.: Secret and public key image watermarking schemes for image authentication and ownership verification. IEEE Trans. on Image Processing. Vol. 10, No. 10, pp. 1593-1601, Oct. 2001.
5. Honsinger, C.W., Jones, P., Rabbani, M., and Stoffel, J. C.: Lossless recovery of an original image containing embedded data. U.S patent application, Docket No 77 102/E-D, 1999.
6. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless generalized-LSB data embedding. IEEE Trans. on Image Processing. Vol. 12, No. 2, pp. 157-160, Feb. 2005.
7. Celik, M.U., Sharma, G., Tekalp, A.M.: Lossless watermarking for image authentication: a new framework and an implementation. IEEE Trans. on Image Processing. Vol. 15, No. 4, pp. 1042-1049, Apr. 2006.
8. Alattar, A.M.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans. on Image Processing. Vol. 13, No. 8, pp. 1147-1156, Aug. 2004.
9. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. on Circuits and Systems for Video Technology. Vol. 13, No. 8, pp. 890-896, Aug. 2003.
10. Kamstra, L., Heijmans, H.J.A.M.: Reversible data embedding into images using wavelet techniques and sorting. IEEE Trans. on Image Processing. Vol. 14, No. 12, pp. 2082-2090, Dec. 2005.
11. van Leest, A.; van der Veen, M.; Bruekers, F.: Reversible image watermarking. Proc. IEEE Int. Conf. on Image Processing, vol. II, pp. 731-734. Barcelona, Spain, Sept. 2003.
12. Kalker, T., Willems, F.M.J.: Capacity bounds and constructions for reversible data-hiding. Proc. IEEE Int. Conf. on Digital Signal Processing, Vol. 1, pp. 71–76, Jul. 2002.
13. De Vleeschouwer, C., Delaigle, J.-F., Macq, B.: Circular interpretation of bijective transformations in lossless watermarking for media asset management. IEEE Trans. on Multimedia. Vol. 5, No. 1, pp. 97-105, Mar. 2003.
14. Ni, Z., Shi, Y.Q., Ansari, N., and Su, W.: Reversible data hiding. IEEE Trans. on Circuits and Systems for Video Technology. Vol. 16, No. 3, pp. 354-362, Mar. 2006.
15. Hu, Y.J., and Jeon, B.: Reversible visible watermarking and lossless recovery of original images. IEEE Trans. on Circuits and Systems for Video Technology. to appear.
16. Fridrich, J., Goljan, M., and Du, R.: Lossless Data Embedding - New Paradigm in Digital Watermarking. Special Issue on Emerging Applications of Multimedia Data Hiding. Vol. 2002, No.2, PDF Journal Editorial, pp. 185-196, 2002.
17. Moffat, A., Neal, R.M., and Witten, I.H.: Arithmetic coding revisited. ACM Trans. on Information Systems. Vol. 16, pp.56-294, Jul. 1998.

A Reversible Watermarking Based on Histogram Shifting

JinHa Hwang[1], JongWeon Kim[1], and JongUk Choi[1,2]

[1] Copyright Protection Research Institute, Sangmyung University,
7, Hongji-dong, Jongno-gu, Seoul, 110-743, Korea
{auking45, jwkim, juchoi}@smu.ac.kr
http://cpri.smu.ac.kr/
[2] MarkAny Inc., 10F, Ssanglim Bldg., 151-11, Ssanglim-dong,
Jung-gu, Seoul, 100-400, Korea
juchoi@markany.com
http://www.markany.com/

Abstract. In this paper, we propose a reversible watermarking algorithm where an original image can be recovered from watermarked image data. Most watermarking algorithms cause degradation of image quality in original digital content in the process of embedding watermark. In the proposed algorithm, the original image can be obtained when the degradation is removed from the watermarked image after extracting watermark information. In the proposed algorithm, we utilize a peak point of image histogram and the location map and modify pixel values slightly to embed data. Because the peak point of image histogram and location map are employed in this algorithm, there is no need of extra information transmitted to receiving side. Also, because a slight modification on pixel values is conducted, highly imperceptibly images can be achieved. As locations of watermark embedding are identified using location map, amount of watermark data can dramatically increases through recursive embedding. Experimental results show that it can embed 5K to 130K bits of additional data.

1 Introduction

Development of computer technology and widespread use of internet have driven this world into fast-changing digital place. With digitization of multimedia contents, everybody can access multimedia contents more easily than in analog age. Even if digitization of the multimedia contents provides more opportunities to media contents, it also provide easy access paths to copy and distribution of digital contents, because of characteristics of the digital data, represented by 0 and 1. As the copy and distribution of digital contents are widely conducted illegally in internet environment, the copyright holders began to pay attention to copyright protection technologies.

Of the technologies that can protect copyright of digital contents, watermarking technology has received keen interests from research communities. Watermarking technology hides copyright information into original contents to protect copyright, which eventually leads to modification of original contents. However, in order to

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 348 – 361, 2006.

achieve the goal of imperceptibility, the watermarking technology modifies original contents so that the modification is not perceptible to naked eyes using Human Visual System (HVS) modeling. As a result, the original image cannot be recovered when the image is watermarked. In the applications where a slight modification can lead to significant difference in final decision making process, such as medical images or military images systems, there has been a requirement of recovery to original contents. For the reason, there have been research efforts of reversible watermark that can recover to original images from watermarked images [2], [3].

The reversible watermark technologies that have been published can be categorized into embedding into spatial domain [1], [2], [3], and embedding in transformation domain [6], [8]. Of the embedding technology into spatial domain, the algorithm suggested by Z. Ni [1] embeds watermark using maximum point and minimum point of histogram, demonstrating excellent imperceptibility of more than 48dB. However, even if it can achieve high degree of data embedding by embedding watermark into maximum value points, the maximum point is changed after embedding watermark data. A problem of this approach is that maximum point and minimum point of histogram should be transmitted to the receiving side for data retrieval.

In this paper, an algorithm is proposed in which additional information is not required for recovery of original image, because we embed watermark data with location map which is composed of information of maximum point and minimum point of the histogram. In this algorithm, by embedding recursively watermark information using expansion of location map a reversible watermarking algorithm is suggested that can greatly enhance amount of embedded information.

In Section 2, we describe previous research efforts in reversible watermarking area for comparison, we explain suggested watermarking algorithm in section 3. In section 4, we analyze experimental results of the suggested algorithm and we describe summary of this research and future research direction in section 5.

2 Reversible Watermarking Algorithm

Reversible watermarking algorithms belongs to fragile watermarking area that has attracted researcher's attention in specific applications such as military data or medical imaging area. The algorithms embed fragile watermark into digital contents for the purposes of content authentication and checking integrity of watermarked contents, or others. The most significant advantage of the reversible watermarking algorithms is that it can recover original contents after removing degradation caused by information embedding after extracting watermark information. Recently many research reports have been published, which can be mainly categorized into three methods [9].

2.1 Reversible Watermarking Algorithms Based on Data Compression

Many reversible watermarking algorithms suggested so far belong to this category that embeds information by compressing spatial domain. A typical example of this technique can be found in research suggested by Fridrich [2], [3]. In the algorithm information can be effectively embedded when the image is partitioned into blocks. However, lossless compression technique should be employed to recover original image.

2.2 Reversible Watermarking Algorithms Based on Space Expansion

These algorithms generate a small value that contain characteristics of original image and then expand the value to embed into the expanded space. Very frequently watermark is embedded into LSB of the expanded space. In the algorithm suggested by Tian, characteristic values of original images are generated using difference values of images and average values and then watermark information are embedded into the space expanded by characteristic values using Integer Wavelet Transform [5]. This technique is very effective in low frequency images in which differences between pixels are small, because it utilizes difference values of images.

2.3 Reversible Watermarking Algorithms Based on Histograms

These algorithms generate information embedding space by modifying histogram values and then embed watermark data. The algorithm suggested by Ni chooses maximum value and minimum value of histogram and then modifies histogram values, showing excellent PSNR and highly increased amount of data embedding. However, a significant disadvantage of the algorithm is that it requires additional information in recovering original images [1]. Algorithm suggested by SK Lee embeds information into locations where the values of the difference images are –1 or +1 and partially solve the problem of Ni's algorithm [11]. However, because of modulo operations required to solve overflow phenomenon, it shows Salt & Pepper noisy problem [11].

Existing algorithms show differences and similarities between the algorithms and continue to improve their performance through complimentary approaches. In this research, a reversible watermarking algorithm is suggested based on histogram and it shows excellent performance measured by imperceptibility while it can allow much more information to be embedded into images.

3 A New Reversible Watermarking Algorithm

The algorithm suggested in this research embeds information by shifting pixels that are located between maximum point and minimum point of histogram. As the characteristics of images show that pixels located left point and right point of the maximum point take big values, and therefore those pixels are also selected for embedding payload. Then, two minimum points are identified so that one minimum point is located at left side of the maximum point while the other minimum point is located at the right side of the maximum point. The selection can drastically reduce degradation in image transformation. Figure 1 shows how maximum point, left minimum point, and right minimum point are selected in the histogram of Lena image (512×512×8). In selecting locations of payload, the maximum point is 155, while the left minimum point and right minimum point are 0 and 255, respectively.

3.1 Location Map

In the suggested algorithm, different from the algorithm suggested by Z. Ni [1], location map is generated to store location information of the pixels that have maximum

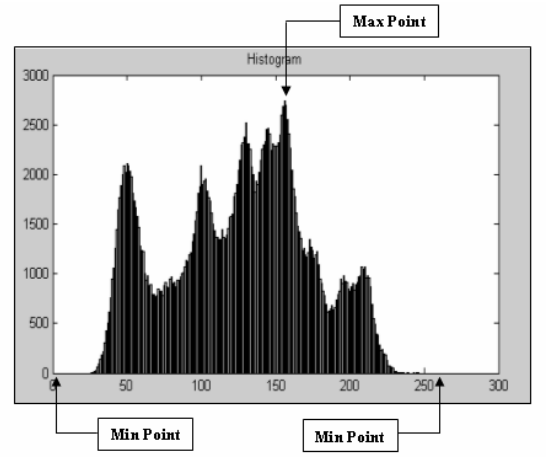


Fig. 1. Histogram of Lena image

point, left minimum point, and right minimum point so that no additional information is required in recovering original images. Figure 2 depicts structure of the location map.

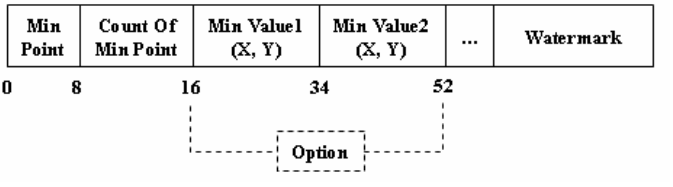


Fig. 2. Structure of Location Map

Location map consists of minimum point, number of pixels of minimum point when the value of the minimum point is not 0, coordinates of X and Y where the value of the minimum point is not 0. The data of minimum point is represented using 8 bits, as the histogram values range 0 to 255. The number of pixels having minimum point that are not 0 does not usually exceed 255 in the most images, and therefore is represented using 8 bits. Then, 9 bits (512×512) are assigned for the coordinate data of pixels that belong to minimum point of histogram, but do not have value of 0. This information is added to location map only when the value of minimum point is not 0. For example, if the number of pixel whose values are not 0 becomes 0, the information is not included in the location map, which reduces the amount of embedded information. In most cases of digital images, there exist more than two points where the value of minimum point is 0. Therefore in many cases, those information are not included in the location map. As the location map is added to the watermark data, the amount of embedded information can be decreased. However, it is not required for the additional information to be transmitted for recovery of original image.

3.2 Histogram Shifting

In this algorithm, histogram values are shifted in order to embed payload data which consists of locations map and randomly generated watermark information. Figure 3 depicts a mechanism in which histogram data is shifted in case of Lena image.

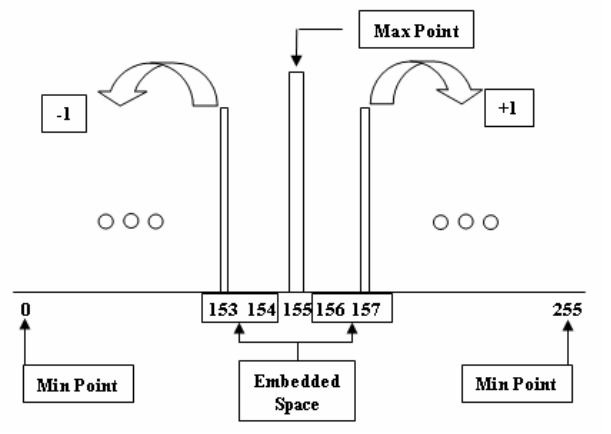


Fig. 3. Histogram Shifting Mechanism

Locations where payload data is embedded are left point and right point of the maximum point. In Figure 3, the maximum point exists at the location of 155, and therefore the locations of payload embedding are location of 154 and 156. To generate the embedding space, the pixels that are located in histogram between left minimum point and left side of the maximum point (pixel value of 154) are shifted one pixel left. To the contrary, the pixels that are located between right minimum point and right side of the maximum point(pixel value of 156) are shifted right one pixel. In this scheme, because we do not change the maximum point of histogram, the original image can be easily recovered if we know information where the minimum points are located. This process can be described in the following equation (1):

$$I'(i,j) = \begin{cases} I(i,j), & if\ I(i,j) = Max \\ I(i,j) - 1, & if\ \min(L) < I(i,j) < Max \\ I(i,j) + 1, & if\ Max < I(i,j) < \min(R) \end{cases} \quad (1)$$

In the equation, *I'(i,j)* is an image in which payload is embedded, while *I(i,j)* is an original image. *Max* is the maximum point of histogram, while *min(L)* and *min(R)* represent left and right minimum point, respectively.

3.3 Embedding Algorithm

Figure 4 depicts embedding process of the suggested algorithm. At first, maximum point of histogram of original image is identified and then pixels located at the right

and the left minimum point in the histogram are identified. Two pixels are selected, if possible, so that they are located in the left side and right side of the maximum point. Then, we generate a location map, consisting of minimum point, number of pixels of minimum point when the value of the minimum point is not 0, coordinates of X and Y where the value of the minimum point is not 0. In the next stage, the location map is combined with randomly generated watermark to generate payload that will be embedded into original image.

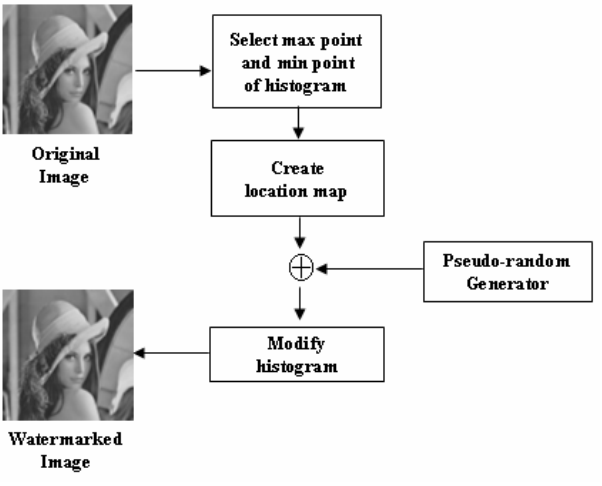


Fig. 4. Diagram of Watermark Embedding Process

$$P = L \bigcup W = p_1 p_2 \cdots p_j \tag{2}$$

The payload to be embedded into original image is $p_i \in \{0,1\},\ 1 \le i \le j$ where j is bits length of P, and L is Location Map. W represents watermark to be embedded.

Histogram values are shifted to make spaces for embedding payload data. We make spaces for embedding payload data by shifting just left point and right point of maximum point and then modify histogram values based on a bit stream of payload. Then we achieved the watermarked image. Condition of watermark embedding is as the followings:

$$I'(i,j) = \begin{cases} I(i,j)+1, if\ I(i,j) = Max-2\ and\ P=1 \\ I(i,j)-1, if\ I(i,j) = Max+2\ and\ P=1 \\ I(i,j), otherwise \end{cases} \tag{3}$$

In the above equation, P represents Payload, and Max represents maximum point in histogram. I represents original image while I' represents watermarked image.

3.4 Extraction and Recovery Algorithm

Figure 5 shows process of watermark extraction and recovery of original image.

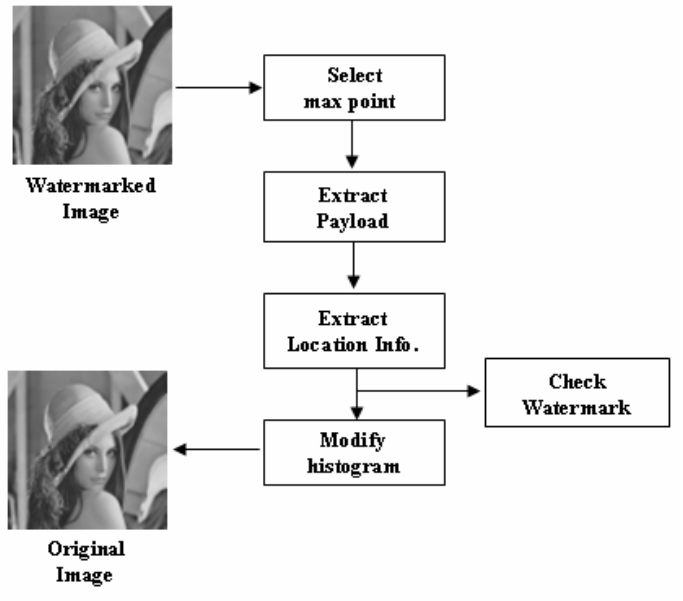


Fig. 5. Diagram of Watermark Extraction and Recovery Process

Watermark extraction process is initiated with identifying maximum point of histogram from watermarked image. Then, payload is generated from pixel values of right point and left point of maximum point. When the watermarked image is scanned, the extraction process extracts 0 to the pixels that have values 2 less than maximum point, or pixels that have values 2 greater than maximum point. Likewise, the extraction process extracts 1 to the pixels that have 1 less than maximum point, or pixels that have 1 greater than maximum point. This extraction process can be formulated in the following equations:

$$P = \begin{cases} 0, if\ I'(i,j) = Max - 2\ or\ I'(i,j) = Max + 2 \\ 1, if\ I'(i,j) = Max - 1\ or\ I'(i,j) = Max + 1 \end{cases} \tag{4}$$

When the payload is extracted, it should be disassembled into location map and watermark. With first 8bit information in the payload that is extracted from left hand of maximum point, coordinates of minimum point are identified. Then, with next 8bit information, it should be checked whether the values of minimum point are 0 or not. If the value of minimum point value is 0, then the next bit streams are real watermark data. Otherwise, if the value of minimum point are not 0, the next bit streams are coordinates of X and Y where the value of the minimum point is not 0. The next bit streams of end of the coordinates are real watermark data. Then the same process is

progressed in right side of maximum point. Finally, the watermarked image is authenticated by calculating the correlation of extracted watermark and random watermark generated using the private key.

Image recovery utilizes the location map information that is obtained from payload. As the locations of minimum points are identified from location map, pixel values can be calculated by assigning +1 or -1 to pixels that are located between minimum points and maximum points. The process is described in the following equation. In the equation Io represents recovered original image.

$$I_o(i,j) = \begin{cases} I'(i,j)+1, if\ \min(L) \le I'(i,j) < Max-1 \\ I'(i,j)-1, if\ Max+1 < I'(i,j) \le \min(R) \\ I'(i,j), otherwise \end{cases} \quad (5)$$

Based on the equation, the first image recovery is done. When the minimum points have value of zero, original image can be recovered completely. However, when the minimum points have values of non-zeros, the pixels of minimum points should be recovered using information additionally contained in location map. As the location data of pixels that belong to minimum points but have non-zero values is included in location map, it can be utilized in recovering minimum value pixels to original values and then recovering back to original image without any information loss.

3.5 Algorithm Extension

The reversible watermarking algorithm suggested in this research has made it possible to recover original image based on information of two points. In the same context, the payload information can be extended repeatedly to contain information of minimum points and maximum points in recovering original image. If the location map can be extended to carry more information, 10 times more information than basic reversible algorithm can be carried. The extended structure of location map is depicted in Figure 6.

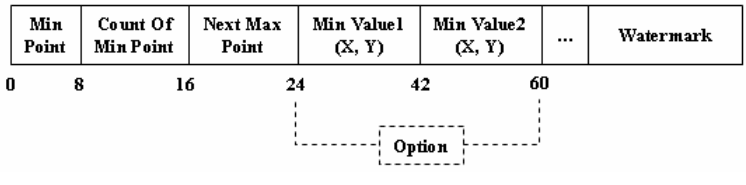


Fig. 6. Structure of Extended Location Map

In the extended location map, 8 bits of data is added to show that there are more than 2 points of information hiding. In the basic reversible watermark (BRW) algorithm, no additional information is needed, because the maximum point of the histogram does not change, and because the left side and right side of the maximum point are easily identified even after watermark embedding. However, in order to embed information into more than two points, the watermarking system needs information of the largest value point, in addition to the left side pixel and right side pixel

of maximum point. With the information of those points, information can be repeatedly added to payload data.

3.5.1 Embedding Algorithm

At first, the number of minimum value point should be identified to find locations of watermark embedding. The left side and right side minimum points located in neighboring area of the maximum point are identified. Then, next maximum point that does not belong to left side and right side point of the first maximum point should be found. The number of maximum points should be one less than the number of minimum points located at left side and right side of the maximum point. Figure 7 shows a case when six points are selected for watermark embedding.

As described in Figure 7, when second pair of maximum point and minimum point is selected between left minimum point and left side of the first maximum point, between right side of the maximum point and right minimum point, respectively, payload embedding begins from left of the maximum point. In Figure 7, at the left of the maximum point (point=156), there are three pairs: (0, 155), (1, 154), and (2, 150). At first, the innermost pair (2, 150) inside of the left of the maximum point (point=156) is selected for payload embedding. The left side point (155) and right side point (157) of the maximum points can be found without any additional information and therefore will be used for payload embedding in the final stage so that watermark can be extracted through a normal process. In order to embed payload into the innermost pair (2, 150) a location map is generated.

Location map generation for other pairs, except the innermost pair, follows the basic algorithm (BRW). However, in the extended algorithm information of the maximum point embedded in previous step should be added. In the example above, for the first pair of payload embedding, the maximum point will be zero, because the first pair is selected and no previous data exists. For the reason, the information of the maximum point becomes zero. However, when payload data is embed into the second pair (1, 154), the maximum point value will be 150. Likewise, by storing information of the maximum points sequentially, watermark can be repeatedly embedded, and then extracted through reversal process of recovery.

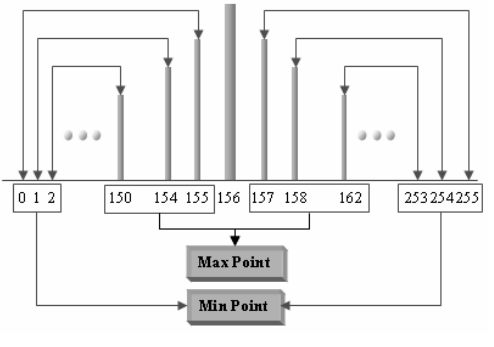


Fig. 7. Selection of 6 Points for Watermarking Embedding and Maximum point and Minimum Points

Figure 8 shows an example of location map generated by the extended algorithm.

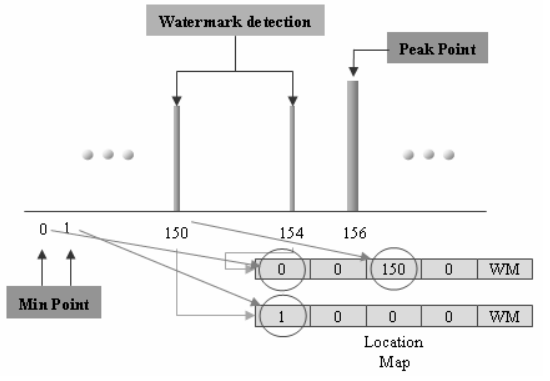


Fig. 8. Example of Location Map Generated from Extended Algorithm

3.5.2 Extracting and Recovery Algorithm

In the watermark extraction algorithm, the maximum point and minimum point in histogram should be identified. Payload extraction proceeds with confirmation whether the minimum point is located at left of the maximum point, or whether located at right of the maximum point. Using the location information of minimum point, the first payload can be extracted and then location map is obtained from the payload. The location map indicates where the first minimum point and the next maximum point are located. Based on the minimum point information, the first recovery can be done. When the image recover is done for the first time, then payload data is obtained again from the next maximum point information. From location map, this extraction and recovery processes are repeated until the next maximum value becomes zero, which indicates that the left recovery process is completed. Likewise, the extraction of location map and recovery of original image can be done through this process.

4 Experimental Results and Analysis

Images of 512X512 grayscale are used in experimenting invisibility and quantity of information hiding for the reversible watermarking algorithm suggested in this research. In Figure-9 and Figure-10, original Lena image and watermarked Lena images are shown for the case that two locations are selected for information hiding. As seen in the Figures, invisibility is so high that the difference between original images and watermarked image can not be discernable.

Table 1 and 2 summarizes the experimental results of the Ni's algorithm and our proposed algorithm respectively. Even though these tables show that two algorithms results in about the same PSNR values and capacity, Ni's algorithm has the

Fig. 9. Original Image

Fig. 10. Watermarked Image

significant disadvantage that it requires additional information in recovering original image, while in our proposed algorithm, no additional information requires to the receiving side for data retrieval because we use the location map in order to recover the original image. Table 1 shows that the average PSNR values of the watermarked image are above 47dB, the capacity ranges from 5k bits to 15k bits for 512×512×8 test grayscale images.

Table 1. Experimental Results of the proposed algorithm

Test Images (512×512×8)	PSNR(dB)	Capacity(bits)
Airplane	48.40	15,300
Baboon	44.53	5,328
Tiffany	44.47	8,578
Goldhill	48.22	4,929
House	48.37	12,225
Sailboat	48.25	7,051
Lena	48.22	5,336
Milk	48.33	11,691
Pepper	44.18	5,164

Table 2. Experimental Results of the Ni's algorithm

Test Images (512×512×8)	PSNR(dB)	Capacity(bits)
Airplane	48.3	16,171
Baboon	48.2	5,421
Tiffany	48.2	8,782
Bacteria	48.2	1,642
House	48.3	14,310
Sailboat	48.2	7,301
Lena	48.2	5,460
Blood	48.2	21,890
Pepper	48.2	5,449

Comparison between the existing reversible watermarking algorithm and the proposed algorithm in terms of PSNR and capacity is presented in Table 3.

Table 3. Comparison between two reversible watermarking algorithm and our proposed algorithm

Methods	PSNR(dB)	Capacity(bits)
Xuan's	24-36	15k-94k
Ni's	> 48	5k-60k
Ours	27-48	5k-130k

Table 4 shows the experimental result of the extended algorithm when we select 6 embedding points.

Table 4. Experimental result of the expanded algorithm (24 embedding points)

Test Images (512×512×8)	PSNR(dB)	Capacity(bits)
Airplane	39.23	41236
Baboon	35.22	15507
Tiffany	36.33	24727
Goldhill	38.89	14911
House	39.09	30309
Sailboat	39.94	20003
Lena	39.33	15323
Milk	39.07	32409
Pepper	37.43	13856

Figure 12 shows test results of PSNR and capacity of information hiding when the number of information embedding points increases through extended location map.

When the payload embedding points are extended to 24, PSNR decreases a little. However, the capacity of information hiding increases 10 times more than when 2 points are selected. Also, because pixels values are dispersed very evenly, much higher invisibility was gained even if PSNR is a little lower.

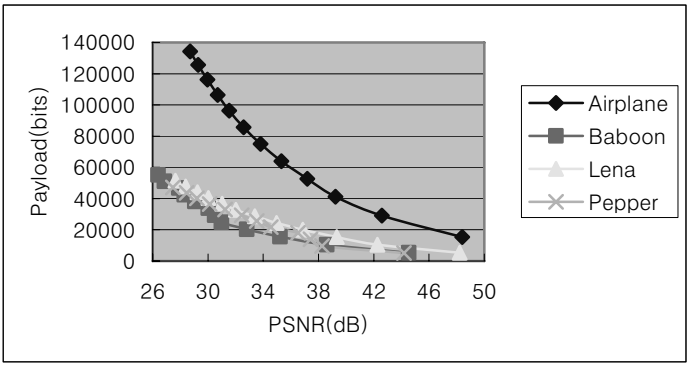


Fig. 11. Embedded payload size vs. PSNR in case of 24 embedding points

5 Conclusions

In this paper, a reversible watermark algorithm is suggested that embeds watermark by slightly modifying pixel values, based on information of maximum point and location map so that degradation of the image is not perceptible. Because of the locations map and utilization of maximum point, no additional information should be transmitted for recovery of original image. Also, because of the slight modification of the pixels values, much higher degree of invisibility can be obtained. In the experiments, the suggested algorithm shows the same level of performance, in terms of PSNR and capacity of information hiding, even with payload information. However, in the extended algorithm, by extending location map and increasing the number of watermark embedding locations, much more information can be embedded. As a result of the increased capacity of information hiding, 5k~130k bits of information was embedded in the experiment.

References

1. Z. Ni, Y. Q. Shi, N.Ansari, and S. Wei, "Reversible data Hiding," in ISCAS Proceedings of the 2003 International Symposium on Circuits and Systems, vol. 2, pp. II-912-II-915, Thailand, May 2003
2. J. Fridrich, J. Goljan, and R. Du, "Invertible authentication," in SPIE Proceedings of Security and Watermarking of Multimedia Content, pp. 197-208, San Jose, Jan 2002
3. J.Fridrich and M. Goljan, "Lossless data embedding for all image formats," in SPIE Proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, vol. 4675, pp. 572-583, San Jose, Jan 2002

4. J. Tian, "High capacity reversible data embedding and content authentication," in IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. III-517-20, Hong Kong, Apr 2003
5. J. Tian, "Reversible data embedding using a difference expansion," in IEEE Transactions on Circuits Systems and Video Technology, vol. 13, no. 8, pp. 890-896, Aug 2003
6. J. Tian, "Wavelet-based reversible watermarking for authentication," in Proceedings of SPIE Sec. and Watermarking of Multimedia Contents. IV, vol. 4675, Jan 2002.
7. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in Proc. Of the IEEE International Conference on Image Processing, vol. II, pp. 157-160, Sept 2002
8. G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, W. Su, "Distortionless Data Hiding Based on Integer Wavelet Transform," in IEE Electronics Letters, pp. 1646-1648, Dec 2002
9. J. Feng, I. Lin, C. Tsai, Y. Chu, "Reversible Watermarking: Current Status and Key Issues," in International Journal of Network Security, vol.2, no.3, pp. 161-171, May 2006
10. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Lossless generalized-lsb data embedding," in IEEE Transactions on Image Processing, vol. 14, no. 2, pp. 253-266, Feb 2005
11. Sang-Kwang Lee, Young-Ho Suh, Yo-Sung Ho, "Lossless Data Hiding Based on Histogram Modification of Difference Images," in PCM 2004, LNCS 3333, pp. 340-347, 2004

Towards Lower Bounds on Embedding Distortion in Information Hiding

Younhee Kim, Zoran Duric, and Dana Richards

George Mason University, Fairfax VA 22030, USA
{ykim9, zduric, richards}@cs.gmu.edu

Abstract. We propose two efficient information hiding algorithms in the least significant bits of JPEG coefficients of images. Our algorithms embed information by modifying JPEG coefficients in such a way that the introduced distortion is minimized. We derive the expected value of the additional error due to distortion as a function of the message length and the probability distribution of the JPEG quantization errors of cover images. We have implemented our methods in Java and performed the extensive experiments with them. The experiments have shown that our theoretical predictions agree closely with the actual introduced distortions.

1 Introduction

1.1 Motivation

Surveillance video data are often used as evidence of traffic accident or crime. The surveillance system widely uses closed-circuit television (CCTV), which is exemplified by the small camera at ATM machines or parking lots. The CCTV system records scenes in analog film. Due to the high cost of film maintenance, the security industry seeks a way to replace it with digital system and store-in-files instead of films. However, there is one issue in using the digitally stored video as evidence: authentication. Because of the ease of undetectable alteration, it is essential to ensure that the video has not been tampered with after it was archived. Federal Rules of Evidence (FRE) states that the original of a recording is required to prove the content of the recording (FRE 1002) [1].

There is a requirement for authentication of digital image as evidence. An authentication system should detect any tampering in a marked image. It may be desirable for some applications to provide an indication of how much alteration has occurred and where the alteration has occurred. Another requirement is that the message extracting process should not require the original image. There are two technical approaches which provide authentication of digital video data: cryptographic approach [1,2] and information hiding approach.

Cryptographic authentication, creates a digital digest of the original image and encrypts it with the signer's key, creating a digital signature. This digital signature [25] can be decrypted only by a key that is correspondent to the signer's key. The digital signature can prove data integrity: if the image is exactly the same as the original, the digest of the image will match the decrypted digest. In

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 362–376, 2006.

cryptographic authentication, the digital signature is attached with the original image or stored in a safe place for later use.

The information hiding approach inserts authentication data into the original image by modifying the image imperceptibly. Combining authentication data and image together is beneficial in many applications; however, the distortion caused by modifying the original image raises many concerns. The degradation in video quality is not noticeable by the human visual system; however, for example, it may affect image enhancement processing or a pattern-matching system in an attempt to recognize or identify a certain person appearing in the video. Such image processing algorithms require the highest possible image quality in order to work reliably. Law enforcement may ask questions such as "Is this image the same as what was originally captured?"

To overcome the concern of the information hiding approach to the authentication problem, we propose a embedding scheme to *minimize distortion due to embedding*. It maintains the high quality of image as well as combining the authentication data with to-be-authenticated data together, which makes the information hiding approach to authentication more attractive.

Distortion is a measure of the modification of the original data due to embedding information and it varies depending on the amount of information embedded in the image, which is called a *payload*. It is clear that a high payload increases the level of distortion. However, there has been very little work on how to optimally embed information in terms of the tradeoff between distortion and payload. We provide an analysis of distortion due to embedding with various payloads. This will allow users to achieve the maximum possible payloads with tolerable distortions of their data.

Most information hiding methods operate in two steps. First, a *cover object* is analyzed and the perceptually insignificant bits are identified. It is assumed that changing these bits will not make observable changes to the cover. Second, the identified bits are modified by the message bits to create a *stego object*. In this paper, cover object is an image in compressed JPEG [19] format. The perceptually insignificant bits correspond to a subset of LSBs of the JPEG coefficients. Although, the LSBs of JPEG coefficients are usually considered perceptually insignificant modifying some of these bits can produce detectable (but imperceptible) distortions of the original image. Our algorithms use parity codes and matrix-coding technique to minimize the distortion of the stego image relative to the cover image.

The paper is organized as follows. In Sec. 2 we briefly review the relevant prior work in the field. In Sec. 3 we provide technical background for our work including the basic facts about JPEG compression and the matrix coding. In Sec. 4 and 5 we describe our method and sketch the theoretical analysis of our method. In Sec. 6 we present some experimental results. Finally, in Sec. 7 we present the concluding remarks.

2 Related Work

With regard to the authentication that is based on information hiding, two problems are related: fragile watermarking and semi-fragile watermarking. In fragile

watermarking, the inserted watermark is lost or altered as soon as any modification occurs in the cover object. Watermark loss or alteration indicates that the cover object has been tampered with, while the recovery of the watermark within the data indicates data originality. In semi-fragile watermarking, the inserted watermark is designed to be destroyed by some manipulations but to survive innocuous manipulations, e.g., moderate image compression. Since we are interested in authenticating the original data we will only discuss fragile watermarking.

The early fragile technique for authentication involves inserting the mark in the least significant bits (LSBs) of the actual image pixels [7,8] and the added watermark is a pseudo-random sequence, which is not related to the content of the image. Wong [9] calculates a digest of the image using a hash function. The image ID, image size and user key are hashed and embedded by modifying the LSBs of pixels of the image. A hybrid method in color images was proposed by Yeung and Mintzer [10]; Fridrich and Goljan [11] proposed an improvement. Fragile watermarking systems in the transformation domain like JPEG have the advantage that the mark can be embedded in the compressed domain. Wu and Liu [29] describe a technique based on a modified JPEG encoder, which changes the quantized DCT coefficients before entropy coding. Kundur and Hatzinakos [21] and Xie and Arce [22] describe techniques based on the wavelet transform. Kundur modifies the Haar wavelet transformation coefficients while Xie modifies the SPIHT algorithm.

The goal of steganography is to insert a message into a carrier signal so that it cannot be detected by unintended recipients. Steganalysis attempts to discover hidden signals in suspected carriers or at the least detect which media contain hidden signals. Detailed survey of early algorithms and software for steganography and steganalysis can be found in [18,28]. An early quantitative technique for steganalysis was designed by Westfeld and Pfitzmann [26]. This research prompted interest in both improving statistical detection techniques [13,15] as well as building new steganographic methods that would be difficult to detect by statistical methods [27,24,16].

Various attempts have been made to make steganographic content difficult to detect, often by reducing the payload. Anderson and Petitcolas [3] suggested using the parity of a group of bits to encode a message bit; large groups of cover bits could be used to encode a single bit, the bits that need to be changed could be chosen in a way that would make detection hard. Westfeld [27] designed a steganographic algorithm $F5$ that uses matrix coding to minimize the modifications of the LSBs. Fridrich et al. [14,15] reported several techniques for detecting steganographic content in images. If a message is inserted into the LSBs of an image, some features of the image change in a manner that depends on the message size.

Sallee [24] developed a hiding method that preserves distributions of individual JPEG coefficients. Fridrich et al. [16] have proposed an information hiding method that guarantees low distortion rates of stego objects. The method makes use of the JPEG quantization errors by computing all rounding errors

of the JPEG coefficients. Note that for some coefficients the rounding error is $0.5 \pm \epsilon$. These coefficients can be rounded either down or up without a noticeable difference and they are considered changeable. Recently, Kim et al. [20] have described a parity-coding based hiding algorithm that minimizes distortion error by utilizing the rounding errors in JPEG quantization.

3 Technical Background

3.1 Information Hiding System

The goal of information hiding is to convey a message secretly and imperceptibly to people except a specific receiver. Generally, it modifies a cover object to embed message. We denote the cover object as a vector X and the message as M. M will be embedded in X by modifying X into $\hat{X}$, which is called a stego object.

$$X = (x_1, x_2, \ldots, x_l), \quad \hat{X} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_l), \quad M = (m_1, m_2, \ldots, m_k). \tag{1}$$

An information hiding algorithm has a pair of functions $\mathbf{f}$ and $\mathbf{g}$ such that

$$\hat{X} = \mathbf{f}(X, M), \quad M = \mathbf{g}(\hat{X}). \tag{2}$$

3.2 JPEG Image Format

We assume here that cover objects are image files in JPEG format, but our techniques are not limited to them. The JPEG image formatting removes some image details to obtain considerable saving of storage space without much loss of image quality. For the JPEG encoder, each channel is divided into 8×8 blocks and transformed using the two-dimensional discrete cosine transform (DCT). Let $f(i,j),\ i,j = 0,\ldots,N-1$ be an $N \times N$ image block in any of the channels and let $F(u,v),\ u,v = 0,\ldots,N-1$ be its DCT coefficient. See [17] for the mathematical specifics.

The coefficient $F(0,0)$ is the DC coefficient and all others are called AC coefficients. JPEG uses quantization and rounding formulas,

$$F'(u,v) = \frac{F(u,v)}{Q(u,v)}, \tag{3}$$

$$F''(u,v) = Round(F'(u,v)) \tag{4}$$

to obtain integer-valued coefficients $F''(u,v)$, where $Q(u,v)$ is a quantization table [17]. The process results in a quantization error:

$$\delta(u,v) = F''(u,v)Q(u,v) - F(u,v). \tag{5}$$

3.3 Minimizing Embedding Distortion

The cover object is obtained by a JPEG compression process and the JPEG coefficients and the corresponding rounding errors are known. Information hiding will add additional distortion beyond the quantization errors (see Eq. (5)).

Let X' and X'' be the vectors of DCT coefficients before and after the rounding, respectively (see Eq. (4)). The rounding error is given by $r_i = x''_i - x'_i$.

$$\begin{aligned} &X' = (x'_1, x'_2, \ldots, x'_l). \\ &X'' = Round(X') = (x''_1, x''_2, \ldots, x''_l). \\ &R = X'' - X' = (r_1, r_2, \ldots, r_l). \end{aligned}$$

Our analysis will assume that each element of R is an independent, identically distributed (i.i.d.) random variable and that its probability density $p(r), r \in [-0.5, 0.5]$ is known. A message $M = (m_1 \;\; m_2 \ldots m_k)$ is a binary sequence and each element, m_i, is a i.i.d. random variable. A message, M, is embedded into X, and the output of the embedding process is $\hat{X}$. In prior work, the cover object, X, was typically X'', but in this paper, X will be X'. Note that X' is only available during the JPEG encoding process.

We propose an embedding algorithm for minimizing distortion given rounding errors. We will show how bit-parity coding and matrix coding can be used to minimize the distortion.

4 Parity Coding

4.1 Embedding Algorithm

Our embedding algorithm makes use of given rounding errors. We seek a pair of functions $\mathbf{f}$ and $\mathbf{g}$ such that

$$\hat{X} = \mathbf{f}(X, R, M), \quad M = \mathbf{g}(\hat{X}) \tag{6}$$

and $\|\hat{X} - X\|_1$ is minimized. This approach assumes that encoding is done within JPEG, since R is known. Distortion is defined as:

$$D = |\hat{X} - X|. \tag{7}$$

Note that if $\hat{X} = X$ then $D = R$, i.e. if no information is embedded, the distortion is equivalent to the rounding error. Since embedding any message almost always requires changing bits, the best result that can be obtained is

$$\|D\|_1 \geq \|R\|_1.$$

We will show how bit-parity codes of length $n \leq l/k$ can be used to minimize the distortion $\|D\|_1$.

Embedding Algorithm. The embedding process divides X into blocks of length n. To embed a bit m_i the block $X^i = x_{n(i-1)+1}, \ldots, x_{ni}$ is considered. If the parity of the LSBs of X^i is equal to m_i, no change needs to be made to any x_j, so $\hat{X^i} = X^i$. On the other hand, if the parity of the LSBs of X^i is different from m_i, we need to select an $x_j \in X^i$ to replace it by either $\hat{x}_j = x''_j - 1$ or by $\hat{x}_j = x''_j + 1$; in the first case, the distortion will be $d_j = -1 + r_j$ and in the second case, it will be $d_j = 1 + r_j$. One exception applies to this embedding algorithm. When $x''_j = \pm 1$, we will make $\hat{x}_j = \pm 2$ to avoid creating additional zero-valued coefficients. Since we are interested in minimizing $|d_j|$, we should use $\hat{x}_j$ that minimizes it, that is

$$\hat{x}_j = \begin{cases} 2, & r_j \geq 0 \ \ \& \ \ x''_j = 1 \\ x''_j - 1, & r_j \geq 0 \ \ \& \ \ x''_j \neq 1 \\ -2, & r_j < 0 \ \ \& \ \ x''_j = -1 \\ x''_j + 1, & r_j < 0 \ \ \& \ \ x''_j \neq -1. \end{cases}$$

In terms of rounding error, r_j, the distortion is given by

$$d_j = \begin{cases} 1 + |r_j|, & x''_j r_j > 0 \ \ \& \ \ x''_j = \pm 1 \\ 1 - |r_j|, & \text{otherwise.} \end{cases}$$

Finally, the additional error due to distortion, ε_j is given by

$$\varepsilon_j = \begin{cases} 1, & x''_j r_j > 0 \ \ \& \ \ x''_j = \pm 1 \\ 1 - 2|r_j|, & \text{otherwise.} \end{cases} \tag{8}$$

A goal in information hiding is to design embedding functions such that $\|d\|_1$ is minimal. Since r_js are already given, minimizing $\|d\|_1$ is equivalent to minimizing $\boldsymbol{\Delta} = \sum_{j=1}^{l} \varepsilon_j$.

4.2 Embedding Distortion

Let us define X_p as a set of the coefficients such that their corresponding embedding error is $\varepsilon_j = 1 - 2|r_j|$ and X_q as a set of the coefficients such that $\varepsilon_j = 1$.

$$X_p = \{x_j |\ x''_j r_j \leq 0 \ \ \vee \ \ x''_j \neq \pm 1\}$$
$$X_q = \{x_j |\ x''_j r_j > 0 \ \ \wedge \ \ x''_j = \pm 1\}$$

Let $p = \frac{|X_p|}{|X_p| + |X_q|}$, i.e, the related proportion of all coefficients that belong to X_p and let $q = 1 - p$, i.e, the related proportion of X_q.

For a given block of coefficients, $X = \{x_1, \ \ \ldots, x_n\}$ of size n, there will be $0 \leq n_p \leq n$ coefficients from X_p and $n_q = n - n_p$ coefficients from X_q. For any n_p, the probability of the particular proportion of coefficients will be

$$P\{n_p = i\} = \binom{n}{i} p^i q^{n-i} \tag{9}$$

First, the distortion for those coefficients that belong to X_p is analyzed. We have assumed that r_js are i.i.d. random variables and that their probability density $f_r(x)$ is known. We can define this probability distribution for $\psi = |r|$ as

$$F_\psi(x) = \int_{-x}^{x} f_r(x)dx, \quad x \in [0, 0.5].$$

The probability distribution for $\nu = 1 - 2\psi$ is given by

$$F_\nu(x) = 1 - F_\psi(\frac{1-x}{2}), \quad x \in [0, 1]. \tag{10}$$

We are looking for a coefficient having the smallest embedding error within every block, X. If there are $n_p > 0$ coefficients from X_p in a given block, the algorithm will choose the coefficient corresponding to the minimal embedding error among the n_p coefficients. Since the embedding error for the coefficients from X_q is 1, which is always greater than the embedding errors of the coefficients from X_p, the remaining coefficients are not considered.

Given the probability distribution $F_\nu(x)$ for the embedding errors of the coefficients in X_p, the minimal additional error due to embedding is given by

$$\mu = \min_j \{\varepsilon_j\}, \;\; 1 \le j \le n.$$

The distribution of μ when $n_p = i$ is given by

$$F_\mu(x, i) = P_r\{\mu \le x | n_p = i\} = \begin{cases} U(x-1), & i = 0 \\ 1 - (1 - F_\nu(x))^i, & i \ge 1, \end{cases} \tag{11}$$

where

$$U(x-1) = 0, \;\; x < 1$$
$$U(x-1) = 1, \;\; x \ge 1$$

and $F_\nu(x)$ is given by Eq. (10).

After taking account of all possible combinations of the coefficients, the distribution of additional error will be given by

$$F_\mu(x) = \textstyle\sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i} F_\mu(x, i). \tag{12}$$

The expected value of the embedding error will then be given by

$$E[\mu] = \textstyle\sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i} E[\mu | n_p = i], \tag{13}$$

where

$$E[\mu | n_p = i] = \int_0^\infty x dF_\mu(x, i), \;\; i \ge 0$$

5 Modified Matrix Coding

5.1 Background

Matrix coding was proposed by Crandall [12] to improve embedding efficiency by decreasing the number of required bit changes. Westfeld [27] proposed *F5*, a steganographic algorithm which implemented the matrix coding. In F5, cover data is the set of LSBs of quantized DCT coefficients after rounding. The notation $(1, n, k)$, where $n = 2^k - 1$, denotes embedding k message bits into an n bit sized block by changing only one bit of it. The embedding process divides X into blocks of length n and message data M into blocks of length k. To embed the i^{th} message block, $\{m_{k(i-1)+1}, \ldots, m_{ki}\}$, a cover data block $\{x_{n(i-1)+1}, \ldots, x_{ni}\}$ is used. Let us denote M and X as the message block and the cover block. The advantage of matrix coding is that we change only one bit to embed several bits. A function b needs to be defined in matrix coding:

$$b(X) = \bigoplus_{j=1}^{n} (x_j) \cdot j. \tag{14}$$

To calculate α, the position of the bit that needs to be changed, we use

$$\alpha = M \oplus b(X). \tag{15}$$

If $\alpha \neq 0$, then bit α in the block of X should be flipped, 1 to 0 or 0 to 1. The modified block is then given by

$$\hat{X} = \begin{cases} X, & \text{if } \alpha = 0. \\ x_1, \ldots, \neg x_\alpha, \ldots, x_n & \text{if } \alpha \neq 0. \end{cases} \tag{16}$$

On the decoder's side, k message bits are obtained from an n bit sized cover data by computing the following:

$$M = b(\hat{X}). \tag{17}$$

We cannot tune *F5* [27] to minimize distortion since the bit flip is completely constrained. We propose to modify $F5$ to increase the number of possible bit-change choices in each block. We describe our approach for two bit-changes. We call our method Modified Matrix Encoding (MME) and denote MME3, MME4 when we extend it to 3 and 4 bit-changes respectively.

5.2 Embedding Algorithm

MME will find pairs of numbers (β, γ) such that $\beta \oplus \gamma = \alpha$. If we use the embedding technique described in Sec. 4.1 for each cover block, X of length n, we are given two vectors of coefficients $(x'_1, ..., x'_n), (x''_1, ..., x''_n)$, before and after rounding respectively. We know the rounding errors $(r_1, ..., r_n)$ and the message block M of length k. As in Sec. 2 $X = X'$. We compute α using (15) and the

pairs $(\beta_1, \gamma_1), ..., (\beta_h, \gamma_h)$ such that $\beta_i \oplus \gamma_i = \alpha$. Note that the number of pairs is $h = \frac{n-1}{2}$.

The embedding error using an unmodified $F5$ is given by $\varepsilon_0 = 1 - 2|r_\alpha|$, see (8). For each of the pairs (β_i, γ_i), the embedding error is given by one of four cases:

$$\varepsilon_i = \begin{cases} 2, & \text{if } x''_{\beta_i} r_{\beta_i} > 0 \ \& \ x''_{\gamma_i} r_{\gamma_i} > 0 \ \& \ x''_{\beta_i} = \pm 1 \ \& \ x''_{\gamma_i} \pm 1 \\ 2 - 2|r_{\gamma_i}|, & \text{if } x''_{\beta_i} r_{\beta_i} > 0 \ \ \& \ \ x''_{\beta_i} = \pm 1 \ \ \& \ \ x''_{\gamma_i} \neq \pm 1 \\ 2 - 2|r_{\beta_i}|, & \text{if } x''_{\gamma_i} r_{\gamma_i} > 0 \ \ \& \ \ x''_{\gamma_i} = \pm 1 \ \ \& \ \ x''_{\beta_i} \neq \pm 1 \\ 2 - 2(|r_{\beta_i}| + |r_{\gamma_i}|), & \text{otherwise.} \end{cases} \tag{18}$$

In order to decide how to create $\hat{X}$, we find

$$\mu = \min_j \{\varepsilon_j\}, \ \ 0 \leq j \leq \frac{n-1}{2}.$$

Given μ, we compute $\hat{X}$ by

$$\hat{X} = \begin{cases} X, & \text{if } \alpha = 0 \\ x_1, \ldots, \hat{x}_\alpha, \ldots, x_n, & \text{if } \mu = \varepsilon_0 \\ x_1, \ldots, \hat{x}_{\beta_i}, \ldots, \hat{x}_{\gamma_i}, \ldots, x_n, & \text{if } \mu = \varepsilon_i, \ i = 1, \ldots, \frac{n-1}{2}. \end{cases} \tag{19}$$

5.3 Embedding Distortion of MME

Let us denote X_p as a set of the coefficients such that their corresponding embedding errors will be $\varepsilon_0 = 1 - 2|r_j|$ when we change the coefficients. X_q is denoted as a set of the coefficients such that the embedding errors will be $\varepsilon = 1$ if we change the coefficients.

$$\begin{aligned} X_p &= \{x_j |\ x''_j r_j \leq 0 \ \ \vee \ \ x''_j \neq \pm 1\} \\ X_q &= \{x_j |\ x''_j r_j > 0 \ \ \wedge \ \ x''_j = \pm 1\} \end{aligned}$$

Let $p = \frac{|X_p|}{|X_p| + |X_q|}$, i.e, the related proportion of all coefficients that belong to X_p and let $q = 1 - p$, i.e, the related proportion of X_q. For a given block of coefficients, $X = x_1, \ \ \ldots, x_n$, of size n, the only cases we should care about are (a) $x_\alpha \in X_p \wedge (x_\beta \in X_p \wedge x_\gamma \in X_p)$ and (b) $x_\alpha \in X_q \wedge (x_\beta \in X_p \wedge x_\gamma \in X_p)$.

Let m be the number of pairs in which both x_β and x_γ are from X_p, $0 \leq m \leq h$. For any m, the probability of the particular proportion of coefficients will be

$$P\{m = i\} = \binom{h}{i} (p^2)^i (1 - p^2)^{h-i} \tag{20}$$

First, the distortion for coefficients that belong to X_p is analyzed. Again we assume that r_js are i.i.d. random variables and that their probability density $f_r(x)$ is known. Probability distribution for $y = |r|$ is given by

$$F_y(x) = \int_{-x}^{x} f_r(x) dx, \quad y \in [0, 0.5].$$

The probability density for $z = |r_1| + |r_2|$ is given by

$$f_z(x) = f_y(x) \bigotimes f_y(x), \quad z \in [0, 1],$$

where $\bigotimes$ stands for convolution. The probability distribution is given by

$$F_z(x) = \int_0^z f_z(x)dx, \quad z \in [0, 1].$$

The probability distribution for $\nu = 1 - 2y$ is given by

$$F_\nu(x) = 1 - F_y(\frac{1-x}{2}), \quad \nu \in [0, 1]. \tag{21}$$

The probability distribution for $\omega = 2 - 2z$ is given by

$$F_\omega(x) = 1 - F_z(2 - x), \quad \omega \in [0, 2]. \tag{22}$$

The embedding error due to change of $x_\alpha \in X_p$ will follow the distribution of $F_\nu(x)$ and changes of x_β and x_γ will follow the distribution of $F_\omega(x)$.

To estimate the probability distribution of the embedding distortion due to embedding for (t, n, k) matrix codes, we use the order statistics [23]. As the first approximation, we have only considered the case when all embedding errors are $\varepsilon_i \leq 1$. Now, we need to obtain distribution of the smallest error we should take for embedding with consideration of embedding errors greater than 1.

The distribution of μ when $n_p = i$ and $x_\alpha \in X_p$ is given by

$$F_\mu(x, i) = P_{i,X_p}\{\mu \leq x | n_p = i\} = \begin{cases} U(x-1), & i = 0 \\ 1 - (1 - F_\nu(x))(1 - F_\omega(x))^i, & i \geq 1, \end{cases} \tag{23}$$

where $U(x)$ was defined in Sec. 4.

The distribution of μ when $n_p = i$ and $x_\alpha \in X_q$ is given by

$$F_\mu(x, i) = P_{i,X_q}\{\mu \leq x | n_p = i\} = \begin{cases} U(x-1), & i = 0 \\ 1 - (1 - F_\omega(x))^i, & i \geq 1. \end{cases} \tag{24}$$

After taking account of all possible combinations of the coefficients, the distribution of additional error will be given by

$$F_\mu(x) = \textstyle\sum_{i=0}^{h} \binom{h}{i} p^i q^{h-i} (p\ F_\mu(x, i, X_p) + q\ F_\mu(x, i, X_q)). \tag{25}$$

The expected value of embedding distortion due to embedding, $E[\mu]$, is given by

$$E[\mu] = \sum_{i=0}^{h} \binom{h}{i} p^i q^{h-i} p E[\mu | n_p = i, X_p]\ qE[\mu | n_p = i, X_q],$$

where

$$E[\mu | n_p = i, X_p] = \textstyle\int_0^\infty x dF_\mu(x, i, X_p),$$
$$E[\mu | n_p = i, X_q] = \textstyle\int_0^\infty x dF_\mu(x, i, X_q).$$

Since changes occur in $\frac{n}{n+1}$ cases in any block, the expected embedding error per bit is given by

$$E[\|\epsilon\|_1] = E[\mu] \times \frac{n}{n+1}.$$

6 Experimental Results

We have implemented our algorithms in Java. In this section we demonstrate the operation of our methods on two test images. Figure 1 shows the test images, which are color JPEG images. Rounding error histograms are also shown in Fig. 1; we estimate the rounding-error distributions by normalizing the histograms.

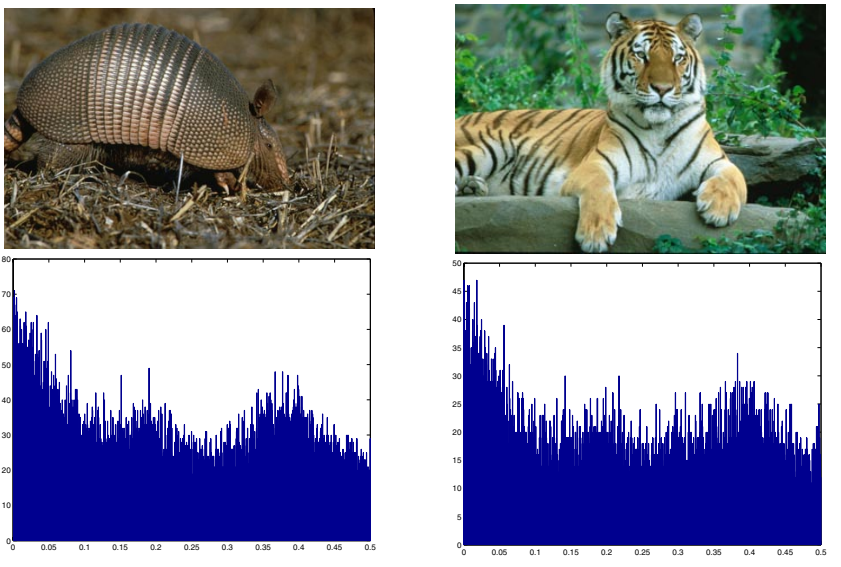


Fig. 1. Left column: armadillo image. Right column: tiger image. Top row: test images. Bottom row: rounding error histograms of the nonzero AC JPEG coefficients. The histogram is normalized to estimate a probability density of rounding errors.

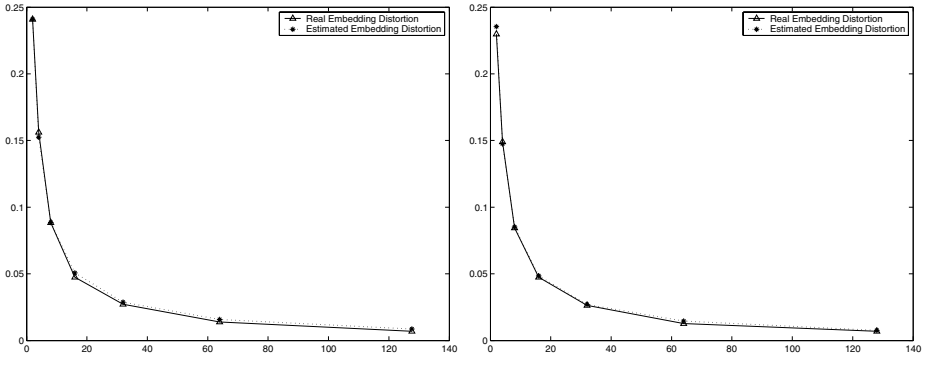


Fig. 2. Embedding error analysis of bit-parity coding in various block size n. Top: Theoretical embedding error and experimental embedding error for the armadillo image (left image in Fig. 1). Bottom: Comparison of the theoretical embedding error to the experimental embedding error for the tiger image (right image in Fig. 1).

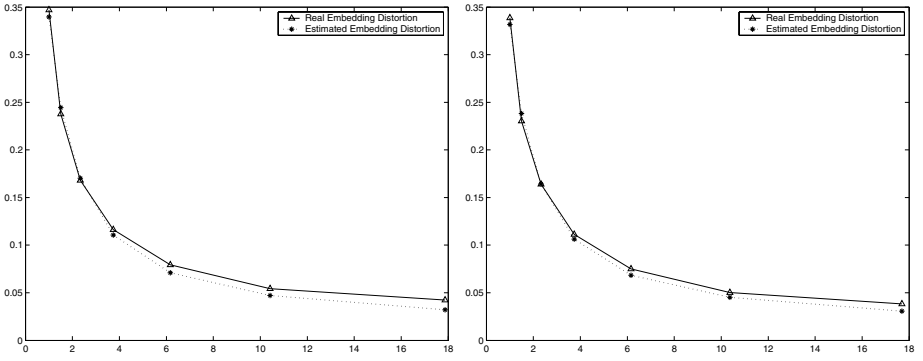


Fig. 3. Embedding error analysis of modified matrix encoding in embedding rates. X-axis is $\mu^{-1}1$. Top: result for for the armadillo image (left image in Fig. 1). Bottom: result for for the tiger image (right image in Fig. 1).

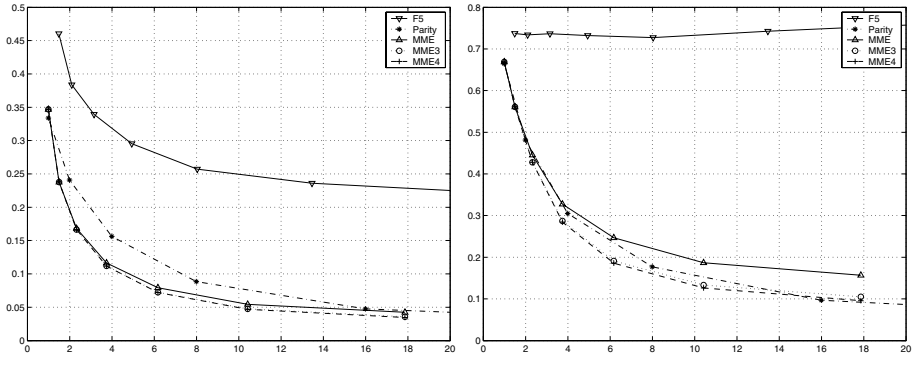


Fig. 4. Embedding error analysis for the armadillo image (left image in Fig. 1). Top row: Embedding distortion per embedding message bit with μ^{-1}. Bottom row: Embedding distortion per changing one bit with μ^{-1}.

The algorithm modifies a publicly available implementation of the JPEG image compression algorithm. After computing the DCT, all non-zero AC coefficients are marked for possible embedding and collected to form X'' with the corresponding rounding errors forming R. The implementation follows the algorithm described in Sec. 4 and 5. Our algorithm uses bit-parity and modified matrix encoding to choose the coefficients of which modifications introduce minimal embedding distortion.

All tests was accomplished with 7 different block-sizes for matrix coding $((t, 2^{k-1}, k),\ k = 1, \ldots, 7)$ and for bit-parity coding $(2^k,\ k = 1, \ldots, 7)$.

Figures 2 and 3 show the theoretical embedding error analysis for parity coding and MME. They plot the comparison of the predicted embedding error to the real experimental embedding error and show close agreement between the theoretical prediction and the actual embedding distortion.

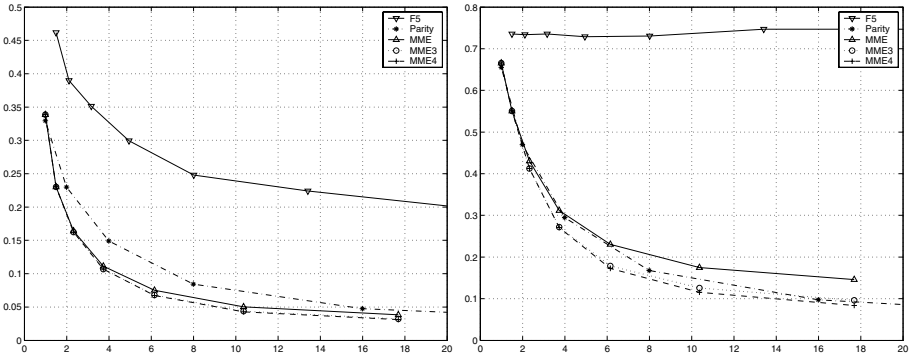


Fig. 5. Embedding error analysis for the tiger image (right image in Fig. 1). Top row: Embedding distortion per embedding message bit μ^{-1}. Bottom row: Embedding distortion per changing one bit μ^{-1}.

Figures 4 and 5 show the comparison of distortion in various embedding rates (μ^{-1}) using *F5*, *MME* and the extended versions of *MME*, MME3 and MME4, that modify up to 3 and 4 bits per block, respectively. (These algorithms are analyzed but not defined in this paper.) Note that the embedding errors caused by MME can be decreased by MME3 version noticeably, but benefit from MME4 is not much noticeable. The top graphs plot the distortions per embedding message bit in decreasing embedding rates, μ^{-1}. Note that the embedding rate is given by the block size divided by the number of message bits in the block. The bottom graphs plot the distortion per changed bit in decreasing embedding rates, μ^{-1}. The distortions due to our embedding algorithms are noticeably lower than one due to F5.

7 Conclusions

In this paper, we propose two efficient information hiding algorithms in the least significant bits of JPEG coefficients of images. Our algorithms embed information by modifying JPEG coefficients in such a way that the introduced distortion is minimized. We derive the expected value of the additional error due to distortion as a function of the message length and the probability distribution of the JPEG quantization errors of cover images. We have implemented our methods in Java and performed the extensive experiments with them. The experiments have shown that our theoretical predictions agree closely with the actual introduced distortions. Future work will include techniques for finding effective embedding algorithms using more sophisticated codes.

References

1. N. D. Beser, T. E. Duerr, and G. P. Staisiunas, "Authentication of digital video evidence," in *Proceedings of the SPIE International Conference on Applications of Digital Image Processing XXVI*, vol. 5203, November 2003, pp. 407–416.

2. A. Pramateftakis, T. Oelbaum, and K. Diepold, "Authentication of mpeg-4-based surveillance video," in *Proceedings of International Conference on Image Processing*, vol. 1, October 2004, pp. 33 – 37.
3. R. J.Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, May 1998.
4. P. Mouline and R. Koetter, "Data-hiding codes," *Proceedigns of the IEEE*, vol. 93, no. 12, pp. 2083–2126, December 2005.
5. F. Bartolini, A. Tefas, M. Barni, and I. Pitas, "Image authentication techniques for surveillance applications," *Proceedigns of the IEEE*, vol. 89, no. 10, pp. 1403–1418, December 2001.
6. E. T. Lin and E. J. Delp, "A review of fragile image watermarks," in *Proceedings of the Multimedia and Security Workshop Multimedia Contents*, October 1999, pp. 25–29.
7. T. A. Z. Van Schyndel, R G and C. F. Osborne, "A digital watermark," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, November 1994, pp. 86–90.
8. R. Wolfgang and E. Delp, "A watermark for digital images," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 3, September 1996, pp. 219–222.
9. P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for imageauthentication and ownership verification," *IEEE Transactions on Image Processing*, vol. 10, no. 10, 2001.
10. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, October 1997, pp. 680 – 683.
11. J. Fridrich and M. G. andArnold C.Baldoza, "New fragile authentication watermark for images," in *Proceedings of IEEE Internatinal Conference on Image Processing*, September 2000, pp. 446–449.
12. R. Crandall. "Some Notes on Steganography." Posted on Steganography Mailing List, 1998. http://os.inf.tu-dresden.de/ westfeld/crandall.pdf
13. J. Fridrich. "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes." *Proc. 6th Information Hiding Workshop*, Toronto, Canada, 2004.
14. J. Fridrich, M. Goljan, and R. Du. "Detecting LSB Steganography in color and gray-scale images", *IEEE Multimedia Magazine,* pp. 22–28, October 2001.
15. J. Fridrich, M. Goljan, and D. Hogea. "Steganalysis of JPEG images: Breaking the F5 algorithm", LNCS 2578, Springer-Verlag, Berlin Heidelberg, pp. 310–323, 2002.
16. J. Fridrich, M. Goljan, and D. Soukal. "Perturbed quantization steganography with wet paper codes." *Proc. ACM Multimedia Workshop*, Magdeburg, Germany, 2004.
17. R.C. Gonzales, R.E. Woods, "Digital Image Processing", Addison-Wesley, 2002
18. N. Johnson, Z. Duric, and S. Jajodia. *Information Hiding: Steganography and Watermarking — Attacks and Countermeasures.*, Kluwer Academic Publishers, Boston, 2000.
19. Joint Photographic Experts Group. http://www.jpeg.org/public/jpeghomepage.htm.
20. Y. Kim, Z. Duric, D.Richards. "Limited Distortion in LSB Steganography." *Proc. SPIE Electronic Image*, 2006
21. D. Kundur and D. Hatzinakos, "Towards a telltale watermarking technique for tamper-proofing," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, October 1998, pp. 409–413.

22. G. Liehua and X. Arce, "Joint wavelet compression and authentication watermarking," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, October 1998, pp. 427–431.
23. A. Paopoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Boston, MA, 1991.
24. P. Sallee. "Model-based steganography." *Proc. Information Hiding Workshop*, LNCS 2939, Springer-Verlag, Berlin, pp. 154–167, 2003.
25. B. Schneier, *Applied Cryptography*. John Willey & Sons. Inc., 1996.
26. A. Westfeld and A. Pfitzmann. "Attacks on steganographic systems." *Proc. Information Hiding Workshop*, LNCS 1768, Springer-Verlag, New York, pp. 61-75, 1999.
27. A. Westfeld. "F5—a steganographic algorithm: High capacity despite better steganalysis." *Proc. Information Hiding Workshop*, LNCS 2137, Springer-Verlag, Berlin, pp. 289-302, 2001.
28. P. Wayner. *Disappearing Cryptography*. 2nd ed., Morgan Kaufmann, San Francisco, 2002.
29. M. Wu and B. Liu, "Watermarking for image authentication," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, October 1998, pp. 437–441.

Improved Differential Energy Watermarking for Embedding Watermark*

Goo-Rak Kwon, Seung-Won Jung, Sang-Jae Nam, and Sung-Jea Ko

Department of Electronics Engineering, Korea University
5-1 Anam-Dong, Sungbuk-Ku, Seoul 136-701, Korea
Tel.: +82-2-3290-3228
grkwon@dali.korea.ac.kr

Abstract. Digital watermarking is increasingly demanded for protecting or verifying the original image ownership. In this paper, we propose an improved differential energy watermarking using adaptive differential energy watermarking (ADEW) with cross binding wavelet tree (CBWT). The ADEW embeds a secret bit string which is obtained by the error correction code (ECC). Thus, the proposed method not only takes advantage of the ADEW's error resilience but corrects a secret bit string by using ECC after error occurrence. Through experiments, we compare the proposed method with conventional DEW approaches and the proposed method shows the appropriateness for robust watermarking.

1 Introduction

With the rapid spread of computer networks and the further development of multimedia technologies, the copyright protection of digital contents such as audio, image and video, has been one of the most serious problems because digital copies can be identical to the original. Over the last decade, watermarking has been developed to a large extent for protection copyright of digital contents [1]. To be valid for copyright protection, an image watermarking system should meet the below requirements [2,3,4]:

- Transparency: The embedded watermark should degrade the perceptual quality of host media to a minimal degree.
- Robustness: Any attack that maintains the host image quality acceptable cannot erase the embedded watermark. The attacks fall into two categories: noise-type and geometric distortion. Noise-type distortions include lossy compression, filtering, dithering, re-sampling, and digital-analog conversion, and geometric distortions include scaling, cropping, flipping and rotation.
- Security: A watermarking system should be secure in a sense that an unauthorized party is unable to remove the watermark even with full knowledge of the watermarking technique. The security of the system should rely

* This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul Industry-Academy-Research Cooperation Project.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 377–386, 2006.

on the use of cryptographic keys rather than obscuring the watermarking technique.
– Adequate complexity: This issue is critical especially for real-time applications.

A tradeoff for these requirements is usually necessary. Since Cox et al. [5] proposed a novel watermarking strategy using spred-spectrum technique, there have been many researches inspired by methods of image coding and compression. These works are robust to image noise and spatial filtering, but show severe problems to geometric distortions. To solve these problems, Langelaar et al. [6] introduced a blind method called differential energy watermarking. A macroblock (MB) with 16×16 size which consists of discrete cosine transform (DCT) blocks with 8×8 size is divided into two parts to embed a watermark bit. High frequency DCT coefficients in the compressed bit stream are selectively discarded to produce an energy difference in the two parts of the same macroblock. In the same method, Wang [7] proposed a watermarking scheme that embeds a watermark into a pair of trees using Wavelet Transform (WT). The total energies of each tree are selectively discarded according the watermark bit until the discarded energy is below the other. However, this scheme has a serious problem. When the energy difference between two trees is large, it cannot be avoided damaging the image quality. In fact, most transformed image has the pairs of trees with a high energy difference. In this paper, we propose an adaptive wavelet tree based blind watermarking scheme. The wavelet coefficients of the image are grouped into two pairs of wavelet trees which are bound crosswise. Each watermark bit which is encoded by ECC [4] is embedded into two pairs of wavelet trees. To embed a watermark bit into the two pairs which are composed in CBWT, the energy of each tree is selectively discarded by ADEW.

The rest of this paper is organized as follows. In Section 2, the background on the wavelet trees and the DEW is described. In Section 3, the proposed watermarking algorithm is explained in detail. In Section 4. the experimental results are shown. Finally we present the conclusion of our experiments in Section 5.

2 Background

2.1 WT and Wavelet Tree

For convenience, we use 4-level wavelet transform of a 32×32 image as example. With 4-level decomposition, there are 13 subbands as shown Fig. 1.(a). The wavelet coefficients are grouped according to wavelet trees except the coefficients of high frequency bands ($L1^1$, $L1^2$, and $L1^3$) which contain little energy. We use the coefficients in sub-band $L4^1$, $L4^2$, and $L4^3$ as roots to form wavelets trees [9]. Thus, the total number of trees is equal to the number of the coefficients in subband $L4^1$, $L4^2$, and $L4^3$. In this example there are $3{\times}2^2 = 12$ trees. Each tree has 21 coefficients corresponding to the same spatial location as shown in Fig. 1.(b).

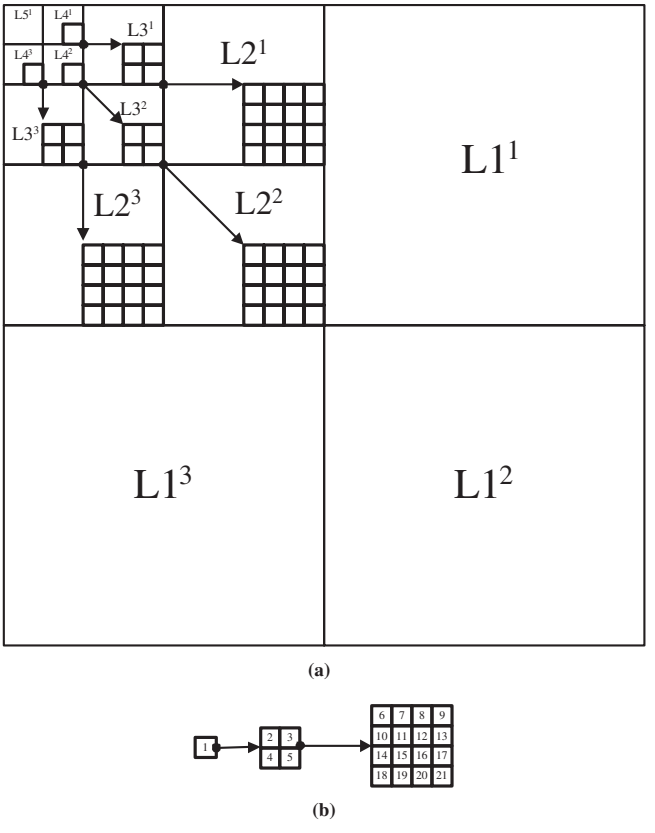


Fig. 1. WT and wavelet tree: (a) 4-level WT. (b) Wavelet tree with order.

2.2 The DEW Algorithm

The DEW algorithm [6] embeds watermark bits into an JPEG, JPEG2000, and MPEG stream (or any other block DCT based video compression system) by enforcing energy difference between certain groups of 8×8 DCT blocks of the I-frames to represent either a '1' or a '0' watermark bit. The energy difference is enforced by selectively removing high frequency components from the DCT blocks. The 8×8 DCT blocks of an I-frames are first randomly shuffled using a secret seed. In Fig. 2.(a), the complete procedure to calculate the energy difference D is illustrated for $n = 16$ nonshuffled 8×8 DCT blocks. The white triangularly shaped areas illustrate the subsets over which the energies are calculated for a particular choice of the cutoff index $index_c = 27$. At the right a blowup of one 8×8 DCT block is presented. This process serves two purposes. First, the seed serves as a secret key without which one cannot extract the watermark properly. Second, the process is done to avoid having a group of blocks where an unbalanced energy content exists. The DEW algorithm also has several interesting properties. The complexity is relatively low because it

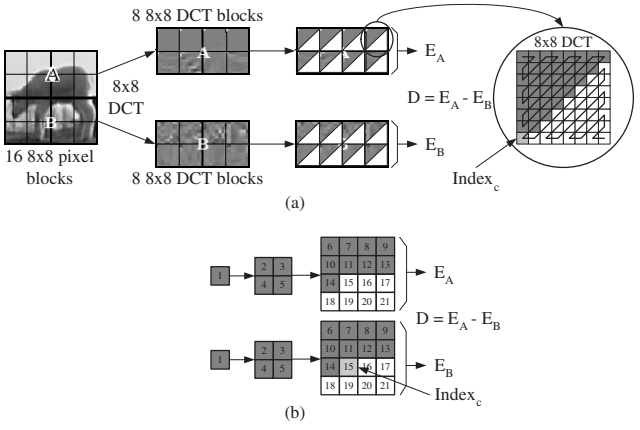


Fig. 2. The procedure of the DEW: (a) The conventional DEW using DCT. (b) The proposed method using WT (see Fig 1).

embeds the watermark at the DCT coefficient level and thus only variable length decoding of the bit stream is needed for the watermark decoding and detection process. It also has sufficient robustness because a full decoding and reencoding is needed to completely remove the watermark from the stream and it had been shown that transcoding a watermarked 8Mbps MPEG stream down to 6Mbps only introduce a 7% watermark bit error rate. The watermark payload is also sufficiently high, up to 0.42Kbps for a stream encoded at 8Mbps. The visual impact of the watermarking process is also negligible.

In the same process, Fig. 2.(b) shows that the proposed method has two wavelet trees to represent either a '1' or a '0' watermark bit using the energy difference and is calculated for the cutoff index $index_c = 15$. Since watermark requires much more additional information for the copyright protection, we propose an efficient watermarking technique that the superiority of the technique can not only has much watermark in payload but also preserve the image quality.

3 Proposed Watermarking Scheme

Figure 3 illustrates the overall block diagram of the proposed watermarking system. Figure 3(a) shows the proposed watermark encoder. The watermark is changed to a PN-Code which is a watermark bit of ± 1. The original image is transformed into wavelet coefficients using WT. CBWT groups the wavelet coefficients into two pairs of trees shown in Section 3.1. The PN-code is embedded into the two pairs by using ADEW. The proposed CBWT and ADEW are later described in Sections 3.1 and 3.2, respectively. Figure 3(b) represents the extraction procedure in the watermark decoder. Here, W is the number of inserting watermark bits and W' is the number of extracted watermark bits. After the adaptive extraction extracts W', the detection process compares W' with the

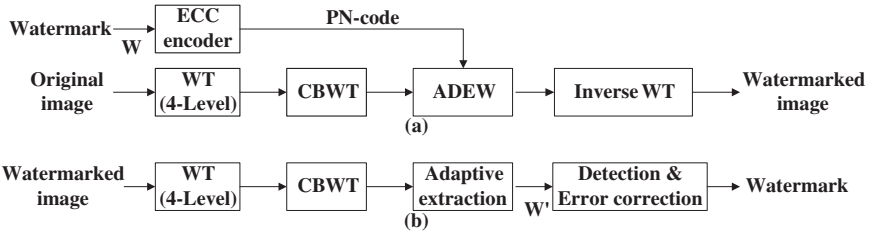


Fig. 3. Block diagram of proposed watermarking system: (a) Watermark encoder. (b) Watermark decoder.

original W. The normalized correlation between W and W' will be explained in Sect. 4. The watermark from CBWT is extracted by adaptive extraction. W' is detected and corrected by error correction [11]. The detection process distinguishes the watermark in watermarked image by using the similarity between the original watermark and the extracted watermark.

3.1 Proposed CBWT

Before explaining CBWT in detail, we introduce the energy difference, D, for both the conventional method and the proposed method. $D = |E_A - E_C|$ is used for the Wang's method and $D = |(E_A + E_B) - (E_C + E_D)|$ for the proposed method. The energy rate, R_E, of the Wang's method and the proposed CBWT is shown in Fig. 4. In the Wang's method, R_E is given by

$$R_E = \begin{cases} \frac{E_A}{E_C}, & if\ E_A \geq E_C \\ \frac{E_C}{E_A}, & otherwise. \end{cases} \tag{1}$$

In the proposed method, R_E is given by

$$R_E = \begin{cases} \frac{E_A+E_B}{E_C+E_D}, if(E_A + E_B) \geq (E_C + E_D) \\ \frac{E_C+E_D}{E_A+E_B}, otherwise. \end{cases} \tag{2}$$

Note that the highest R_E is about 18 in the Wang's method (see Fig. 4(a)). Figure 4(b) shows that the proposed method has the peak value that is close to 4. Here, R_E, which is related to D, is an important factor in determining the number of coefficients that should be eliminated in DEW. If the energy difference of wavelet trees is high, a lot of wavelet coefficients should be removed, which results in the degradation of the original image. On the other hand, if the energy difference of wavelet trees is low, a lot of wavelet coefficients which should be removed can be minimized. Thus, the original image is slightly damaged.

Figure 5 shows an example of the proposed CBWT. In this case, we use 4-level WT with 13 subbands. We use the coefficients in subbands $L4^1_{HL}$, $L4^1_{HH}$, and $L4^1_{LH}$ as roots to form wavelet trees [9]. CBWT binds the four trees crosswise which are located adjacent to two pairs. By binding the four trees crosswise into two pairs, the energy difference of two pairs can be reduced.

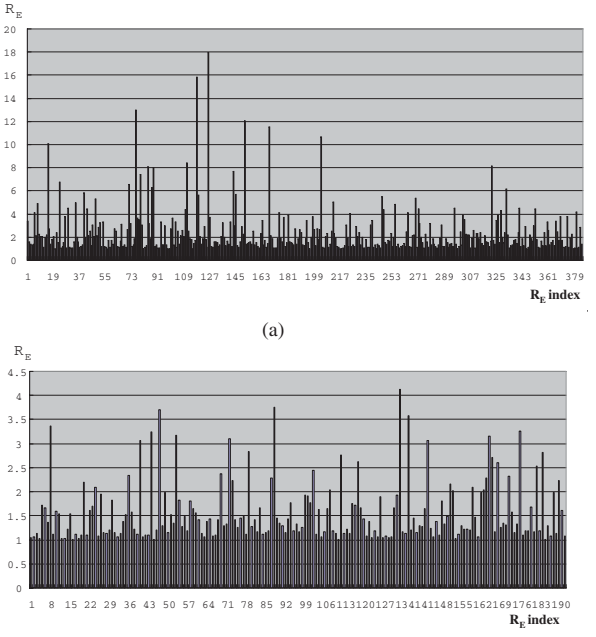


Fig. 4. The energy rate: (a) Wang's method. (b) Proposed method. (Test image is "LENNA" with size of 256×256)

3.2 Proposed ADEW and Adaptive Extraction

To embed a watermark bit into the two pairs which are composed in CBWT, the energy of each tree is selectively discarded by DEW. However, if DEW is applied on the two pairs which have the high energy difference, the image should be damaged. Therefore, to preserve the image, we propose the ADEW in Fig. 6(a). The procedure of ADEW is as follows:

1. Calculate each energy of two pairs.
2. Obtain D of two pairs and compare D with threshold T.
3. If $D > T$, skip to embed W in CBWT.
4. Otherwise, embed W in CBWT.

Figure 6(b) shows the adaptive extraction as follows:

1. Calculate each energy of two pairs.
2. Obtain D of two pairs and compare D with threshold T.
3. If $D > T$, skip to extract W from CBWT.
4. Otherwise, extract W from CBWT.

In the embedding watermark process, the total energy of discarding trees is reduced until the remained energy is below the other's to use the reference threshold, T_r, with experimental statistics. In this paper, the reference threshold

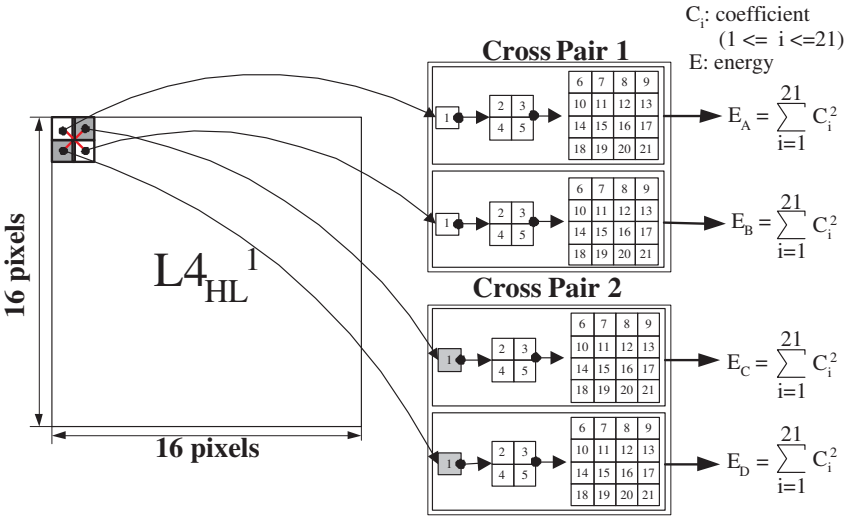


Fig. 5. An example of the proposed CBWT and the calculation of the energy by using wavelet coefficients

corresponding to PSNR = 35dB and PSNR = 40dB are, respectively, $T_r = 9$ and $T_r = 14$. In the same way, the adaptive extraction is applied when the energy difference of them is smaller than the threshold. Here, T_r provides a tradeoff between robust watermarking and quality of the watermarked image.

4 Experimental Results

In order to evaluate the performance of the proposed method in terms of watermark capacity, robustness, and visual quality impact, we tested the extracted watermark using CBWT and adaptive DEW.

For experiments, the proposed method was compared with the Wang's method using 100 randomly collected images. The spatial resolution of collected images is 256×256.

Figures 7(b) and (c) show that the watermark is embedded into "LENNA" test image by Wang's and proposed methods. The proposed method outperforms the Wang's method in image quality while preserving the same watermark payload.

In Tables 1 and 2, we compared the proposed method with that in [7] using the 100 randomly collected images. Here, #Img. represents the number of images where the watermark was correctly detected and Corr. indicates the degree of the correlation between the original watermark and the extracted watermark in (3). The normalized correlation coefficient, ρ_1, should be increased or decreased according to the number of bits in W and the degree of the attack. In the conventional method, ρ_1 is defined as,

$$\rho_1 = \frac{W \cdot W'}{\sqrt{W' \cdot W'}}. \tag{3}$$

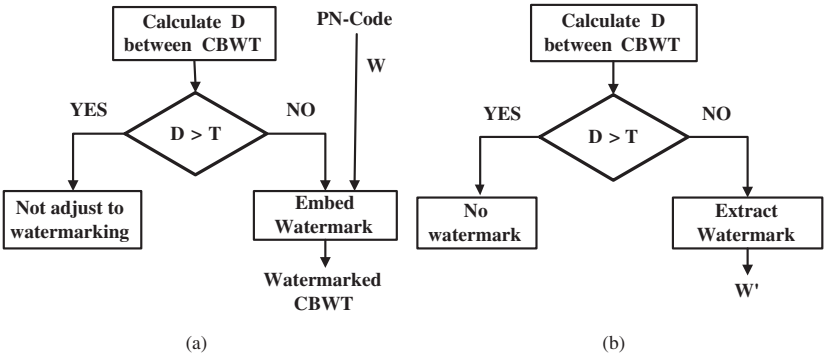


Fig. 6. The proposed ADEW: (a) ADEW. (b) Adaptive extraction.

Fig. 7. Test image: (a) Original LENNA. (b) Watermarked LENNA with PSNR = 30.05dB. (Wang's method [7]) (c) Watermarked LENNA with PSNR = 40.08dB. (Proposed method).

The choice of the normalized correlation coefficient (ρ_1=0.22) depends on the desired false positive probability. In our experiment, to protect the copyright, the excessive skip process in ADEW should be avoided. Therefore, we use 184bits for watermark in payload. The details of the attacks used and the corresponding results are as follows:

- Median and Gaussian filter: After these attacks, the collected images are blurred or unsharpened on the edges.
- Noise addition: The attacker may apply one or more watermark using the same wavelet tree technique.
- SPIHT [8]: The performance of the proposed method is better than that of Wang's method in terms of PNSR and correlation.
- JPEG [10]: Wang's method can not detect the W in watermarked image.
- Removal of bitplane: This attack is done by removing LSBs of wavelet trees.
- Rotation and Scaling: In Table 2, these have less effect on the W extraction on the test images.
- Pixel shift: This affects most watermark extraction in most collected images.

Through simulations, we can find that the proposed method is more robust against signal processing and geometric attacks.

Table 1. Performance under signal processing attacks

	Ref. [7]		Proposed method	
	#Img.	Corr.	#Img.	Corr.
No attack	100	1.00	100	1.00
Median 2×2	93	0.35	97	0.65
Median 3×3	90	0.31	97	0.63
Median 4×4	83	0.26	97	0.64
Gaussian filter	96	0.64	98	0.69
Noise addition	96	0.64	98	0.87
SPIHT				
- Bitrate = 0.3	21	0.13	59	0.70
- Bitrate = 0.5	76	0.27	87	0.71
- Bitrate = 0.7	85	0.27	94	0.78
JPEG (QF = 30)	37	0.15	98	0.70
JPEG (QF = 40)	75	0.23	97	0.73
JPEG (QF = 50)	83	0.26	98	0.81
JPEG (QF = 70)	93	0.57	98	0.84
JPEG (QF = 90)	100	1.00	100	0.92

Table 2. Performance under geometric distortion attacks

	Ref. [7]		Proposed method	
	#Img.	Corr.	#Img.	Corr.
Removal of bitplane 1	100	1.00	100	1.00
Removal of bitplane 2	100	1.00	100	0.98
Removal of bitplane 3	100	0.99	100	0.92
Removal of bitplane 4	92	0.52	98	0.92
Removal of bitplane 5	30	0.11	97	0.70
Pixel shift 2	84	0.28	95	0.67
Pixel shift 3	92	0.34	96	0.64
Pixel shift 4	89	0.29	96	0.73
Pixel shift 5	83	0.28	95	0.62
Rotation 0.25°	90	0.37	91	0.58
Rotation 0.5°	89	0.29	91	0.60
Rotation 0.75°	83	0.26	91	0.55
Rotation 1°	82	0.24	91	0.56
Rotation -0.25°	91	0.32	92	0.60
Rotation -0.5°	83	0.23	92	0.60
Rotation -0.75°	82	0.24	92	0.59
Rotation -1°	32	0.16	92	0.58
Scaling 0.5×	91	0.48	93	0.66
Scaling 0.8×	90	0.41	93	0.63

5 Conclusions

This paper has proposed CBWT and ADEW based on WT. The watermark strength and spread of the watermark into CBWT are controlled by wavelet tree energy. The proposed method is highly robust to most popular intentional attacks. Simulation results show that the proposed method not only has the transparency and the robustness for copyright protection but also preserves the image quality which is very important in some applications.

Acknowledgments

This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul Industry-Academy-Research Cooperation Project.

References

1. Langelaar, G. C., Setyanwn, I., and Lagendijk, R. L.:Watermarking digital image and video data, a state-of-the-art overview. IEEE Signal Processing Mag.. (2000) 20–46
2. Special Issue on Digital Watermarking. IEEE Signal Process. Mag.. **17** (2000)
3. Kaewkamnerd, N. and Rao, K. R.:Wavelet based image adaptive watermarking scheme. Electron. Lett.. **36** (2000) 312–313
4. Zeng, W. and Liu, B.:A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images. IEEE Trans. Image Processing. **8** (1999) 1534–1548
5. Cox, I. J., Kilian, J., Leighton, F. T., and Shamoon, T.:Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Processing. **6** (1997) 1673–1687
6. langelaar, G. C. and Langendijk, R. L.:Optimal differential energy watermarking of DCT encoded images and video. IEEE Journal on Selected Areas in Comm.. **12** (1998) 525–539
7. Wang, S.-H. and Lin, Y.-P.:Wavelet tree quantization for copyright protection watermarking. IEEE Trans. Image Processing. **13** (2004) 154–165
8. Said, A. and Pearlman, W. A.:A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. Circuits Syst. Video Technol.. **6** (1996) 243–250
9. Shapiro, J. M.:Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Processing. **41** (1993) 3445–3462
10. Wallace, G. K.:The JPEG still picture compression standard. Commun. ACM. (1991)
11. Hamming, R. W.:Error detecting and error correcting codes. The Bell System Technical Journal. **29** (1950) 147–160

A Colorization Based Animation Broadcast System with Traitor Tracing Capability

Chih-Chieh Liu[1], Yu-Feng Kuo[2], Chun-Hsiang Huang[2], and Ja-Ling Wu[1,2]

[1] Graduate Institute of Networking and Multimedia,
National Taiwan University
[2] Department of Computer Science and Information Engineering,
National Taiwan University
{ja, pjacky, bh, wjl}@cmlab.csie.ntu.edu.tw

Abstract. Distributing video contents via broadcasting network mechanisms has become a promising business opportunity for the entertainment industry. However, since content piracy is always a serious problem, broadcasted contents must be adequately protected. Rather than implementing sophisticate key-management schemes for access control, an animation broadcast system based on colorization techniques is proposed. In the proposed system, gray-level animation video sequences are delivered via broadcast mechanisms, such as multicast, to reduce the overhead in server processing and network bandwidth. Moreover, color seeds labeled with fingerprint codes are delivered to each client through low-bandwidth auxiliary connections and then used to generate high-quality full-color animations with slight differences between versions received by each client-side device. When a user illegally duplicates and distributes the received video, his identity can be easily found out by examining features extracted from the pirated video. The proposed scheme also shows good resistance to collusion attacks where two or more users cooperate to generate an illegal copy in expectation of getting rid of legal responsibility. The proposed scheme exhibits advantages in network bandwidth, system performance and content security.

Keywords: animation broadcasting, traitor tracing, colorization.

1 Introduction

With the establishment of broadband-network infrastructures and the proliferation of network usages, the entertainment industry is exploiting business opportunities related to delivering digital contents over network connections. Among all proposals, on-demand audio/video entertainments [1, 2] may be the most welcomed solution. Digital contents customized according to user preferences and DRM (digital rights management) regulations can be delivered to each subscriber via virtual or physical leased lines immediately after the service provider receives corresponding requests. However, the on-demand approach undoubtedly imposes heavy traffic on backbone networks when appealing contents are provided. Furthermore, the server will be occupied in order to generate many customized versions of the same content. Conventional traitor-tracing schemes [3, 4] where consumer-specific information

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 387 – 396, 2006.

should be embedded into each piece of content in the server side naturally fits this content-delivery model but also adds inevitable overhead to workload of server.

On the contrary, delivering audio/video entertainments over broadcast channel is another feasible alternative. At the cost of convenience provided by on-demand delivery, content broadcast can greatly alleviate the overhead on both network bandwidth and server performance. However, content broadcasting also introduce new problems. For example, protecting intellectual property rights of content owners is not easy. Currently, broadcast encryption schemes based on complicated key management mechanisms have been proposed, such as [5–7]. However, broadcast encryption schemes often introduce considerable overheads in either network traffic or client storage. Furthermore, once decrypted, received content becomes unprotected and may be distributed arbitrary since the broadcasted content is not customized for individual client at all. Though advanced broadcast encryption schemes can guard the rights of content owners by revoking keys of pirated devices, this type of protection is based on the elimination of pirated devices, rather than protecting each piece of content.

After taking the trade-offs between on-demand content delivery and content broadcast into consideration, an animation broadcast system based on colorization techniques is proposed. In this scheme, gray-level animation video sequences are delivered via broadcast network mechanisms like multicast in the IP network, and color seeds used for rendering full-color videos are transmitted over a low-bandwidth auxiliary channel. In other words, versions of full-color animations possessing features that can be used to identify the illegal user are generated by individual client-side rendering device. The proposed scheme can even resist the collusion attacks that more-than-one users maliciously produce a pirated copy out of their own video sequence.

This paper is organized as follows. Section 2 illustrates the architecture of the proposed animation broadcast system and introduces major modules, including color seeds generation, content delivery via the broadcast channel and the auxiliary channel, rendering full-color animations in the client side and the corresponding traitor tracing mechanism. Section 3 shows experimental results and system performances of the proposed scheme. Section 4 gives some discussions about security issues. Conclusions and future directions of our research are given in Section 5.

2 The Colorization-Based Animation Broadcast System

2.1 Colorization Techniques for Images and Videos

Colorization is a computer-assisted process that adds colors to grayscale images or movies. The work of colorization needs two inputs: one is a grayscale image or video that needs to be colorized; the other is the chrominance side-information. The chrominance information may be interactively provided by user scribbling [8], as well as extracted from images or video with similar color layouts [9, 10]. In conventional colorization techniques targeting on general images or video, the process of searching

for good color seeds is usually modeled as an optimization problems so that minimally distorted images can be reproduced, such as introduced in [8]. Furthermore, edge detection also plays a very important role since it is designate to identify the intensity discontinuity and mark out the object edges. In [11], adaptive edge detection schemes and elaborated color-seed searching algorithms together result in better visual quality of generated images.

In our applications, color seeds are obtained by analyzing the full-color version of frames in video sequences, rather than relying on information from user interventions or analysis based on other colorful images. Since the focus of this paper is to demonstrate how the traitor-tracing capability for broadcast video can be readily implemented based on colorization schemes and without loss of generality, frames in animation video sequences are taken as our test contents. Because frames in animations often consist of clear edges and less gradient areas, full-color video with satisfying visual quality can be rendered by simple colorization schemes. Consequently, the modules corresponding to edge detection and color-seeds finding in general video colorization schemes are simplified for the ease of implementation. Furthermore, each frame in an animation sequence is assumed to be compressed and delivered independently.

2.2 Generating the Gray-Level Frame and Color Seeds

Fig. 1 shows the server-side operations of the animation broadcast system. A full-color animation frame is firstly converted to its YUV representation. Then, the intensity value of each pixel in the Y component of this frame is uniformly quantized to reduce the number of consisting intensities (i.e. number of bins in the color histogram of the quantized gray-level image), as described by:

$$I'(x,y)=[I(x,y)/Q],\ 0 \leqq x \leqq M-1,\ 0 \leqq y \leqq N-1 \tag{1}$$

where $I'(x,y)$ and $I(x,y)$ are intensity values of the pixel with coordinates (x,y) in the Y component of an M by N frame before and after performing an uniform quantization. Q is the employed quantization step, which is empirically set as 8 in our experiments. Instead of performing complicated edge-detection operations, neighboring pixels in the quantized gray-level image are labeled as connected components based on whether their quantized intensity values are the same. In other words, neighboring pixels sharing similar intensities in the original frame will be clustered into the same connected component. Note that there may be many small connected components in the resulting labeled image. To reduce the number of connected components (so as to reduce the transmission overhead), tiny connected components consisting of less than K pixels will be incorporated into nearby connected component if the difference between their quantized intensity values is less than a threshold value T_1 ◦ In the experiments of this paper, K and T_1 are empirically set as 16 and 32, respectively. Then, assume that there are finally C connected components in this frame. The initial color seeds can be calculated as:

$$S=\{(u_1, v_1),\ldots,(u_C, v_C)\} \tag{2}$$

where u_i and v_i are the mean values of consisting pixels in the i-th connected component in the U and the V color domains. When there are totally L users, color seeds required by the frame that will be delivered to user i can be constructed by:

$$S_i = S + a \cdot P_i \quad 0 \leq i \leq L-1 \tag{3}$$

where P_i is a vector sequence consisting of C pairs of pseudo-random binary values, and a is simply a weighting factor.

As shown in Fig. 1, a corresponding gray-level frame and L color seeds corresponding to each client-side user will be generated. Each sequence of color seeds is slightly different from other sequences but all of them can be used to produce animation frames of similar visual quality.

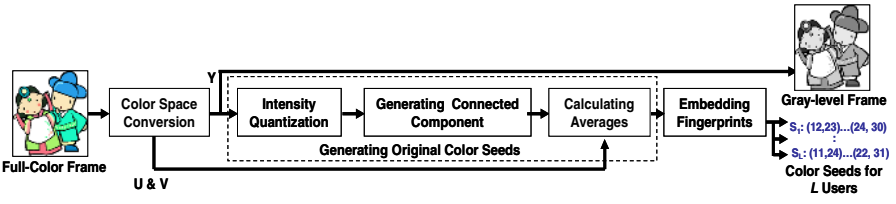


Fig. 1. Generating the gray-level frame and color seeds in the server side

2.3 Animation Broadcasting

The gray-level frame will be delivered to all the users with a lightweight broadcast channel to reduce the overhead on network traffic. In modern network implementations, broadcast functionality is readily provided, e.g. the multicast in IP networks. In this paper, gray-level animations are assumed to be of no commercial values for content pirates, thus no specialized protections are provided. Nevertheless, security mechanisms like access control schemes or digital watermarking can be easily incorporated into the proposed architecture to protect the rights of the gray-level version of animation.

To render the full-color animation video in the receiving end, color seeds for each user are delivered via a low-bandwidth auxiliary channel. Animations displayed on each user's display will share similar visual quality, but the rendered frames are in fact of minute difference in colors.

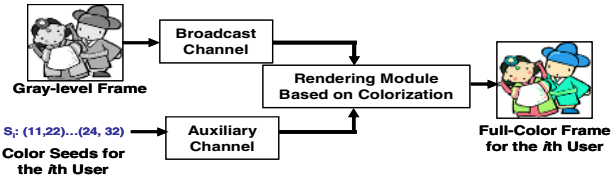


Fig. 2. Delivering the gray-level frame and color seeds and rendering for individual users

2.4 Tracing Traitors

When pirated full-color animation videos are discovered, the traitor tracing mechanism will be invoked to find out the responsible pirate. Identifying features will be extracted from the discovered frames and compared with relevant data reserved by the content provider. The similarity between a suspect frame and a reference frame previously delivered to a certain user is calculated by:

$$Similarity = \frac{\sum_{i=1}^{C} f^i\left(\sum_{(x,y)\in CC_i} D^U(x,y), \sum_{(x,y)\in CC_i} D^V(x,y)\right)}{C} \tag{4}$$

$$f^i(a,b) = \begin{cases} 1, & if \quad a > N_i/2 \quad or \quad b > N_i/2 \\ 0, & otherwise \end{cases} \tag{5}$$

$$D^U(x,y) = \begin{cases} 1, & if \quad \left|U_s(x,y) - U_r(x,y)\right| < T_2 \\ 0, & otherwise \end{cases} \tag{6}$$

$$D^V(x,y) = \begin{cases} 1, & if \quad \left|V_s(x,y) - V_r(x,y)\right| < T_2 \\ 0, & otherwise \end{cases} \tag{7}$$

where $U_s(x,y)$ and $U_r(x,y)$ stand for the values of pixel (x, y) in the U color space of the suspected frame and the reference frame. $V_s(x,y)$ and $V_r(x,y)$ are the counterparts of $U_s(x,y)$ and $U_r(x,y)$ in the V color space. T_2 is a threshold value determining whether two chrominance colors will be regarded as similar and is empirically set as 2. CC_i stands for the i-th connected component and N_i is the total number of pixels in CC_i. If more than one half pixels in a connected component of a suspected frame are regarded as similar to their counterparts in the reference frame, the similarity value will be increase by $1/C$. Therefore, according to the similarity measure, the traitor who illegally distributed the received video can be unambiguously identified according to the distribution of similarity values.

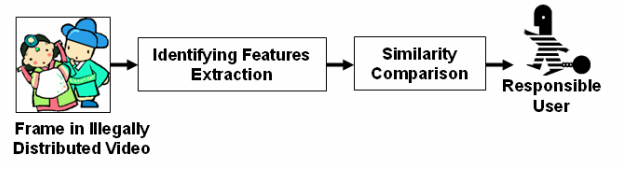


Fig. 3. The traitor tracing mechanism of the proposed broadcast system

3 Experimental Results

In the following experiments, frames adopted from animation movies are used to evaluate the effectiveness of the proposed scheme. Fig. 4 shows all the original frames. In the beginning, the "Chicken Little" frame in Fig. 4(a) is used to demonstrate the visual quality and the compression ratio of the colorization scheme. Table 1 shows the file sizes and PSNR values when JPEG compressions of different quality settings are performed. Table II are corresponding performance using the proposed colorization scheme. Note that the Y component is compressed with JPEG compression quality 100 and the color seeds are compressed by simple VLC coding. In the colorization-based scheme, since the Y component is delivered to users by broadcast channels, this overhead can be neglected when the number of receiving users is large. According to Table I and Table 2, it is clear that, under the condition that similar visual quality is achieved, a user of the traitor-tracing enabled scheme based on colorization receives less amounts of side information as compared with sending each user a compressed and customized version.

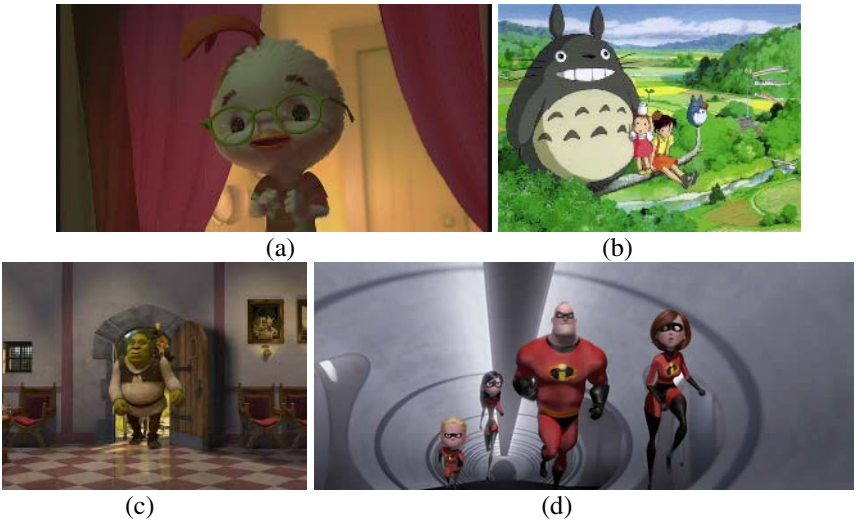


Fig. 4. Test animation frames (a) Chicken Little, (b) My Neighborhood Totoro, (c) Shrek and (d) The Incredibles

Table 1. File sizes and visual qualities of JPEG-compressed "Chicken Little" frames

Frame Type	JPEG Quality	File Size (Bytes)	PSNR (dB)
Uncompressed	N/A	1,843,254	N/A
Full-Color with JPEG Compression	15	20,076	39.09
	20	20,366	39.36
	72	42,225	43.09
	75	45,967	43.53
	100	247,791	47.23

Table 2. File sizes and visual qualities of "Chicken Little" frames based on colorization

Frame Type	# of Color-Seed Pairs	Data Size (Bytes)	PSNR (dB)
Y-Only & JPEG	N/A	66,044	N/A
Color Seeds	1,940	3,119	39.09
	2,737	4,404	39.40
	9,852	12,759	43.07
	11,953	15,684	43.51

Fig. 5 depicts the frames reconstructed by the proposed colorization scheme. Configurations and results of this experiment are listed in Table 3. Though minute rendering differences from the original frame do exist, they are imperceptible under normal viewing conditions.

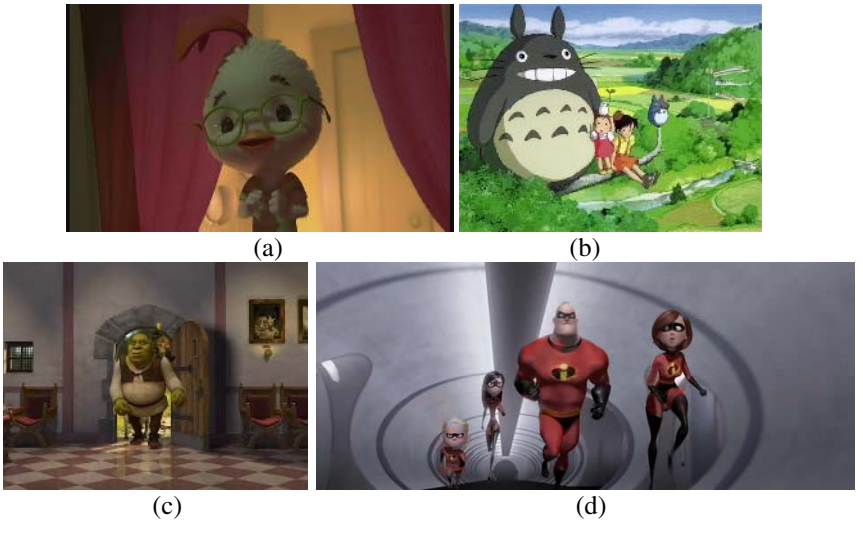


Fig. 5. Reconstructed animation frames

Table 3. Performance of the colorization scheme using different animation frames

Animations	Frame Size	# of Color-Seed Pairs	PSNR of Recs. Frame (dB)
Chicken Little	1024x600	1,940	39.09
Totoro	480x360	2,290	30.48
Shrek	560x416	1,759	38.27
The Incredibles	608x256	1,055	37.50

Figure 6 shows the visual quality of an animation clip consisting of 90 animation frames, and the frame rate is 30 frames per second. It clearly shows that the

fingerprinted color seeds still can be used to generate video of good visual quality. The differences of PSNR values between the animation frames generated using fingerprinted color seeds and those produced using original color seeds are less than 2 dB.

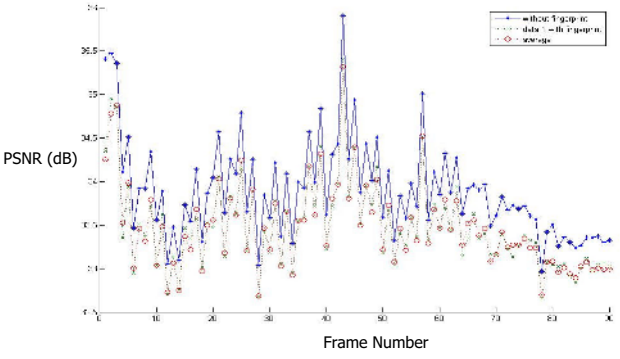


Fig. 6. The visual quality of a animation clip based on color seeds and fingerprinted color seeds

To test the traitor tracing capability of the proposed scheme, 128 frames are rendered based on the same gray-level frame and 128 sequences of color-seed pairs generated from Fig. 4(b), the "My Neighborhood Totoro". Assume that a pirated copy provided by the 10^{th} user is found. Fig. 7 shows the similarity measures, and it clearly identifies the user who is to be blamed as the source of pirate copies.

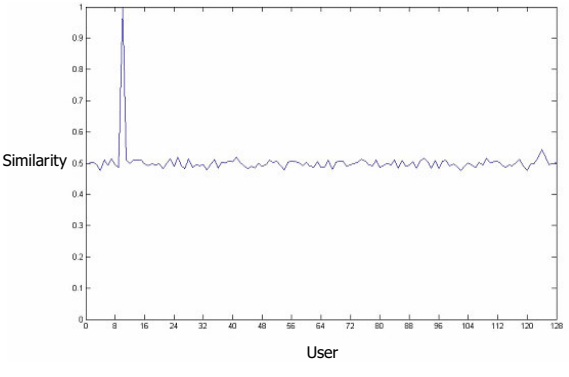
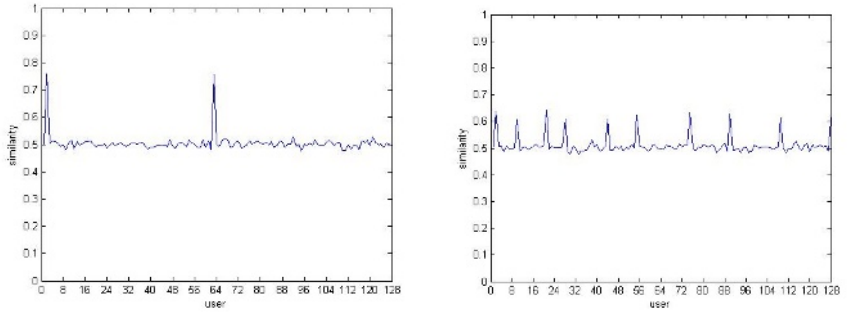


Fig. 7. The proposed scheme unambiguously detects the 10th user who illegally distribute the received animation video

In the literature of fingerprinting, robustness against the collusion attack is another important evaluation criterion. Fig. 8 (a) shows the experimental result that the pirated copy is obtained by averaging the video received by two colluders (the 2^{nd} and 64^{th} users). Fig. 8(b) shows the similarity measures when 10 users colluded together to generate a pirated copy. In both cases, the proposed traitor-tracing scheme can clearly identify the involved users.



(a) (b)

Fig. 8. The proposed scheme can successfully detect (a) 2 users and (b) 10 users who colluded together to generate the pirated version of animation video

4 Security Issues

In the proposed architecture of colorization-based animation broadcast system, the security lies in the availability of good color seeds. In this paper, a simplified colorization scheme for generating animation videos of acceptable quality is demonstrated for the purposes of easy implementation and proving of concepts. In fact, color seeds for high-quality entertainment animations must be computed by computers possessing great computational resources for significant computation time. Therefore, the cost of generating color seeds from a full-color animation movie by the pirate is much larger than buying a legal copy. Furthermore, the auxiliary channel must be adequately protected by security measures to avoid eavesdropping or interception of color seeds. Nevertheless, due to the low rate of data delivered via this channel, the overhead is relatively smaller than protecting on-demand video or conventional broadcast video. Finally, the rendering device in the client side shall be temper-proving to prevent the pirate from directly obtaining the color seeds.

5 Conclusions and Future Works

In this paper, a traitor-tracing enabled animation broadcast scheme based on colorization techniques is proposed. The proposed system can reduce the overhead in network traffic and server load as compared with on-demand video delivery and surpass the conventional video broadcast system in that only lightweight rendering module based on colorization is required in the client side. Our future works will be colorization-based video broadcast systems for general high-quality videos. Furthermore, colorization architectures compatible with videos compressed with important video standards are to be devised.

References

1. Sincoskie, W. D.: System Architecture for a Large Scale Video on Demand Service. Computer networks and ISDN systems, Vol. 22, Issue 2 (1991)
2. Viswanathan S. and Imielinski, T.: Metropolitan Area Video-on-Demand Service Using Pyramid Broadcasting, Multimedia Systems. Vol. 4, NO. 4 (1996)
3. Wu, M., Trappe, W., Wang, Z. J. and Liu, K. J. R.: Collusion Resistant Fingerprint for Multimedia. IEEE Signal Process. Mag., Vol. 21, No. 2 (2004).
4. Zhao H. V. and Liu K. J. R.: Fingerprint Multicast in Secure Video Streaming. IEEE Trans. On Image Processing, Vol.15, Issue 1(2006)
5. Fiat, A. and Naor, M.:Broadcast Encryption. Advances in Cryptography – CRYPTO 93' Proceeding, LNCS, Vol. 773 (1994)
6. Halevy D. and Shamir A.: The LSD Broadcast Encryption Scheme. Advances in Cryptology –CRYPTO '02, LNCS, Vol. 2442, (2002)
7. Lotspiech, J., Nusser S. and Pestoni F.:Anonymous Trust: Digital Rights Management Using Broadcast Encryption. Proceedings of the IEEE, Vol. 92, No. 6 (2004)
8. Anat, L., Dani,L., and Yair, W.: Colorization Using Optimization, Proc. of SIGGRAPH, (2004) pp.689-693
9. Erik R., Michael, A., Bruce G. and Peter S.: Color Transfer between Images. IEEE Computer Graphics and Applications (2001) pp. 34-41,
10. Welsh, T., Ashikhmin, M. and Mueller, K.: Transferring Color to Greyscale Images. ACM Transactions on Graphics (2002)
11. Huang, Y. C., Tung, Y. S., Chen, J. C., Wang S. W. and Wu, J. L.: An Adaptive Edge Detection Based Colorization Algorithm and Its Applications. Proceedings of the 13th annual ACM international conference on Multimedia (2005)

Adaptive Video Watermarking Utilizing Video Characteristics in 3D-DCT Domain

Hyun Park, Sung Hyun Lee, and Young Shik Moon

Department of Computer Science and Engineering, Hanyang University,
1271 Sa-Dong, Ansan, Kyunggi-Do 425-791, Korea
{hpark, sunghyon, ysmoon}@cse.hanyang.ac.kr

Abstract. In this paper, we propose an adaptive blind video watermarking method using video characteristics based on human visual system (HVS) in three-dimensional discrete cosine transform (3D-DCT) domain. In order to optimize the weight factors for watermarking, we classify the patterns of 3D-DCT cubes and the types of video segments by using the texture and motion information of 3D-DCT cubes. Then we embed an optimal watermark into the mid-range coefficients of 3D-DCT cubes by using the trained optimal weight factors. Experimental results show that the proposed method achieves better performance in terms of invisibility and robustness than the previous method under the various possible attacks such as MPEG compression, frame dropping, frame insertion and frame swapping to experimental videos.

1 Introduction

There are two kinds of approach for video watermarking method: embedding a watermark in compressed domain, embedding a watermark in uncompressed domain. Also, watermarking method in uncompressed domain is also classified into 2D and 3D watermarking [1-2]. In the field of watermarking using domain specific transformation such as DCT (discrete cosine transform), DWT (discrete wavelet transform), most 2D watermarking usually embeds a watermark by using motion estimation between successive frames. This method is relatively weak against temporal attacks and watermarks are easily detected by statistical comparing between successive video frames [3]. 3D watermarking embeds a watermark per 8 frames of successive video frames. S. J. Kim et al. proposed the method that embeds watermarks in sub-band of 3 levels 3D-DWT [4]. This method has the disadvantage that is to need the original video stream because of non-blind watermarking. J. H. Lim et al. proposed the method that embeds watermarks at the mid-range coefficients of 3D-DCT cubes [5]. This method has the disadvantage that videos with high motions are relatively weak against possible attacks such as temporal attach and MPEG compression attack.

Therefore, in order to solve the mentioned problems, we propose new adaptive blind video watermarking method using video characteristics based on HVS in 3D-DCT domain. To optimize the weight factors for water marking, we classify the patterns of 3D-DCT cubes and the types of video segments by using the texture and

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 397 – 406, 2006.

motion information of 3D-DCT cubes. After dividing the mid-range AC coefficients that are relatively robust against temporal attacks and properly reflect the HVS into two groups, we embed the optimal watermark the optimal watermark into the mid-range AC coefficients by using the statistical property that summation of two groups is zero approximately. We experiment the performance of our method by estimate the PSNR of watermarked video quality and detection ratio of embedded watermark under the various possible attacks such as MPEG compression, frame dropping, frame insertion and frame swapping.

2 Definition and Estimation of Video Characteristics

Through the statistical observation of many experimental video sequences, we can discover that the distribution of sorted AC coefficients of 3D-DCT cube has a distinct shape according to the amount of texture and motion information in 3D-DCT cube. Therefore, we define the video characteristics by using the mean of absolute value of sorted coefficients and the standard deviation of sorted AC coefficients. Then by using these video characteristics, we compute the optimal sensitivity table and classify the patterns of 3D-DCT cubes and the types of video segments to embed the optimal watermark.

2.1 Pattern Classification of 3D-DCT Cubes

After we apply the 3D-DCT on each 8*8*8 cube, each 3D-DCT cube has 512 coefficients [6]. Through randomly extracted 8*8*8 cubes, we can observe statistically the three distributions of AC coefficients as shown in Figure 1 when the each AC coefficients of 3D-DCT cubes are sorted in u-v-w order. i.e. (0,0,0), (1,0,0), (2,0,0),…, (0,1,0), (1,1,0),...,(0,0,1), (1,0,1),…, (8,8,8). Therefore, we classify 3D-DCT cubes into the three patterns according to the distinct distribution of AC coefficients sorted in u-v-w order.

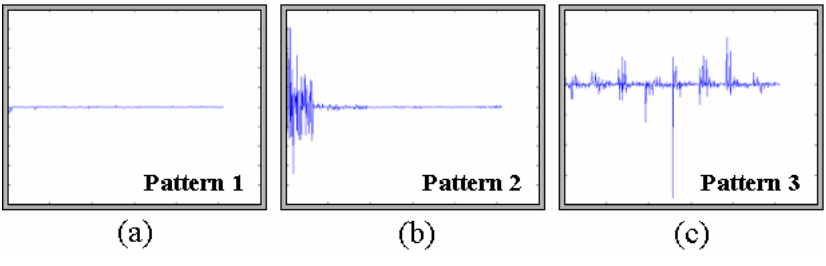


Fig. 1. The distribution of AC coefficients sorted in u-v-w order (a) cube with little textures and little motion (b) cube with high textures and little motion (c) cube with motion and textures. Row coordinate is AC coefficients sorted in u-v-w order and column coordinate is the index *k* of AC coefficients sorted in u-v-w order. The index *k* is from 1 to 511.

Statistically the cube with little texture and little motion has low coefficients as shown in Figure 1(a). The cube with high texture and little motion has the high average and variance in from 1st to 63th coefficient as shown in Figure 1(b). The cube with texture and motion has the high average and variance in from 64th to 511th coefficient as shown in Figure 1(c). 0th coefficient is dc component. To express these distribution characteristics of AC coefficients, we define the video characteristics by the mean of absolute value of sorted coefficients and the standard deviation of sorted AC coefficients.

$$\begin{cases} E_T = E\left\{ \left| F_{n,k}(u,v,w) \right| \right\}, & 1 \le k \le 63 \\ E_M = E\left\{ \left| F_{n,k}(u,v,w) \right| \right\}, & 64 \le k \le 511 \end{cases} \tag{1}$$

$$\begin{cases} \text{Pattern 1}, & \text{if } E_T < T_T \text{ and } E_M < T_M \\ \text{Pattern 2}, & \text{if } E_T \ge T_T \text{ and } E_M < T_M \\ \text{Pattern 3}, & \text{otherwise} \end{cases} \tag{2}$$

where $F_{n,k}(u,v,w)$ is k^{th} coefficient in n^{th} 3D-DCT cube, n is cube index and k is index of AC coefficient.

We use only the mean of absolute value of sorted coefficients to classify the 3D-DCT cubes into three patterns as shown in equation (1) and equation (2). In equation (2), two threshold values T_T and T_M are decided heuristically though randomly extracted 8*8*8 cubes in various experimental video sequences.

2.2 Type Classification of Video Segments

An input video sequence can be divided into a set of video segments, each of which is 8 frames long and converted into 3D-DCT cubes. According to the ratio of patterns of 3D-DCT cubes, we classify the video segment into three types so that the characteristics of the video segment can be reflected in the video watermarking.

Video type 1 is the video segment with little texture and little motion. Video type 2 is the video segment with high texture and little motion. Video type 3 is the video segment with both texture and motion. The rule for the type classification is summarized in equation (3).

$$\begin{cases} \text{Video Type 1}, & \text{if Pattern 1 is more than 50\%} \\ \text{Video Type 2}, & \text{if Pattern 2 is more than 50\%} \\ \text{Video Type 3}, & \text{otherwise} \end{cases} \tag{3}$$

3 Weight Factors for Adaptive Video Watermarking

Generally, a perceptually adaptive watermarking utilizing sensitivity inserts watermarks by using formula form such as the equation (4). Sensitivity explains the

perceptibility of change in individual coefficients of a 3D-DCT cube. i.e. a smaller value on sensitivity table indicates that HVS is more sensitive to this coefficient (or frequency).

To use the mentioned characteristics of HVS, we use also sensitivity table as quantization table used in compression field based on 3D-DCT [6-7]. Then, for improving the robustness, invisibility and capacity of watermarking, we are going to modify the sensitivity *t(u,v,w)* and global embedding strength αusing video characteristics.

$$\begin{cases} X^* = X + \alpha \cdot t(u,v,w) \cdot W, & \text{if } X > t(u,v,w) \\ X^* = X + W, & \text{otherwise} \end{cases} \tag{4}$$

where X is the coefficients of 3D-DCT cube, X^* is the watermarked coefficients, α is a proportional constant and W is the watermark.

3.1 Sensitivity Reflecting Video Characteristics

To compute the sensitivity tables reflecting video characteristics for the optimal watermarking, we utilize the mean of absolute value and the standard deviation of AC coefficients sorted in u-v-w order. Experimentally, it is known that these two variables reflect the amount of texture and motion information on 3D-DCT cubes very well. Therefore we use the summation of these variables as a factor to modify the initial sensitivity table. Equation (5) is the sensitivity reflecting the motion information and equation (6) is the sensitivity reflecting the texture information.

$$t_T(u,v,w) = t(u,v,w) \times (E_T + \sigma_T) \tag{5}$$

$$t_M(u,v,w) = t(u,v,w) \times (E_M + \sigma_M) \tag{6}$$

where *t(u,v,w)* is the initial sensitivity table, E_T is the mean of absolute coefficients which index is from 1st to 63th, E_M is the mean of absolute coefficients that index is from 64th to 511th, σ_T is the standard deviation of coefficients which index is from 1st to 63th, σ_T is the standard deviation of coefficients which index is from 64th to 511th.

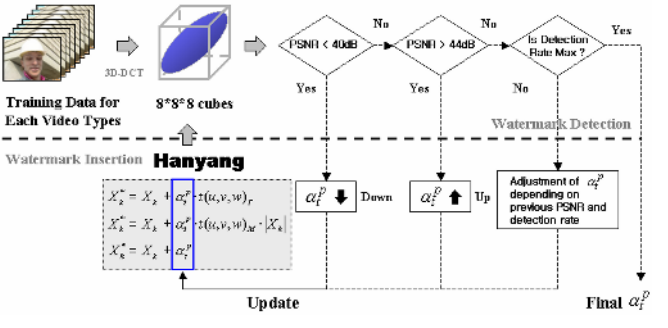


Fig. 2. Training process of proportional constants

3.2 Training of Proportional Constant Reflecting Video Characteristics1

We carry out supervised training to obtain the optimal proportional constants reflecting video characteristics. As shown in Figure 2, the proportional constants α_t^p is trained by estimating the PSNR of video and the detection ratio of watermark recursively. Initial α_t^p is 1, $p \in \{1,2,3\}$ is the pattern of 3D-DCT cubes and $t \in \{1,2,3\}$ is the type of video segments [6].

4 Watermark Insertion and Extraction Scheme

The proposed watermarking method embeds a watermark in the mid-range AC coefficients of 3D-DCT cube that are relatively robust against temporal attacks and properly reflect the HVS [8-9]. Before we embed an optimal watermark, we divide the AC coefficients of mid-range into two groups (odd index group and even index group) and then insert and extract the optimal watermark by using the statistical property that summation of two groups is zero approximately [5].

4.1 Watermark Insertion Scheme

As shown in Figure 3, the proposed watermark insertion scheme is composed of the following six steps; the step accomplishing the 3D-DCT to every 8*8*8 cube, the step classifying pattern of 3D-DCT cubes, the step classifying type of video segments, the step dividing the mid-range coefficients into two groups (odd index group A and even index group B), the step embedding watermark in divided mid-range coefficients and the step accomplishing the Inverse 3D-DCT.

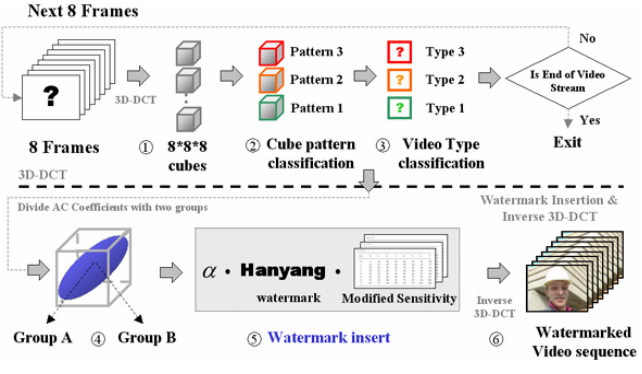


Fig. 3. Proposed watermark insertion scheme

After step 2 and 3, we decide a constant β that indicates the quantity of embedding watermark according to the video characteristics. Here, equation (7) is used for cube with little texture and little motion such as cube pattern 1, equation (8) is used for cube with high texture and little motion such as cube pattern 2 and equation (9) is

used for cube with texture and motion such as cube pattern 3. In equation (9), we add the absolute value of AC coefficients for reflecting characteristics of high AC coefficients.

$$\beta = \alpha_t^p, \qquad \textit{if cube is pattern 1} \tag{7}$$

$$\beta = \alpha_t^p \cdot t_T(u,v,w), \qquad \textit{if cube is pattern 2} \tag{8}$$

$$\beta = \alpha_t^p \cdot t_M(u,v,w) \cdot \left| X_{n,k} \right|, \quad \textit{if cube is pattern 3} \tag{9}$$

where α_t^p is a proportional constant, p is the cube pattern, t is the type of video segment, $t(u,v,w)$ is the sensitivity and $X_{n,k}$ is the mid-range AC coefficients that is selected by equation (10).

In step 4, we select the mid-range AC coefficients $X_{n,k}$ by equation (10) in state of arranging the coefficients of 3D-DCT cube in *u, v, w* order. Then, we divide the mid-range AC coefficients into two groups (odd index group A and even index group B) by equation (11).

$$\begin{aligned} X_{n,k} = \{ & F_{n,k}(u,v,w), \textit{for all } F_{n,k}(u,v,w) \textit{ such} \\ & \textit{that } u+v+w = s,\ s \in \{9,10,11,12\} \} \end{aligned} \tag{10}$$

where $F_{n,k}(u,v,w)$ is k^{th} coefficient in n^{th} 3D-DCT cube, n is cube index and k is index of coefficient.

$$\begin{cases} X_{n,k} \in \textit{Group A}, & \textit{if } k = \textit{odd} \\ X_{n,k} \in \textit{Group B}, & \textit{if } k = \textit{even} \end{cases} \tag{11}$$

In step 5, we select the embedding equation for watermarking among the equation (12), (13) according to the watermark bit. Then, we embed watermarks in mid-range coefficients by using the constant β that is decided depending on pattern of cube and type of video segment.

$$\begin{cases} A_{n,k}^* = A_{n,k} + \beta \\ B_{n,k}^* = B_{n,k} - \beta \end{cases}, \quad \textit{if watermark bit} = 1 \tag{12}$$

$$\begin{cases} A_{n,k}^* = A_{n,k} - \beta \\ B_{n,k}^* = B_{n,k} + \beta \end{cases}, \quad \textit{if watermark bit} = 0 \tag{13}$$

where $*$ indicates the watermarked coefficient, $A_{n,k}$ is the mid-range coefficient in group A and $B_{n,k}$ is the mid-range coefficient in group B.

4.2 Watermark Extraction Scheme

The proposed watermark extraction scheme is composed of the following three steps; the step accomplishing 3D-DCT for every 8*8*8 cubes, the step dividing the

mid-range coefficients into two groups and the step extracting watermarks using the difference of two groups.

$$\begin{cases} watermark\ bit\ = 1, & if\ \ \sum(A_{n,k} - B_{n,k}) > 0 \\ watermark\ bit\ = 0, & if\ \ \sum(A_{n,k} - B_{n,k}) < 0 \end{cases} \tag{11}$$

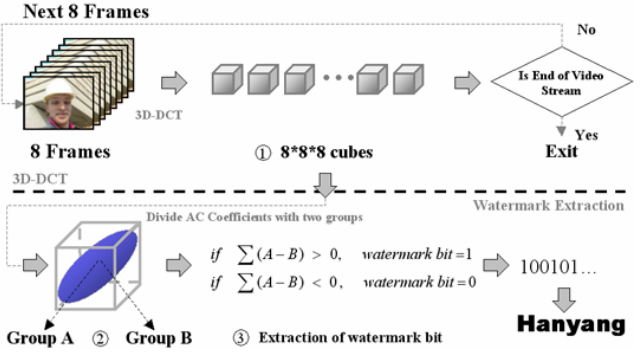


Fig. 4. Proposed watermark extraction scheme

5 Experimental Results

In this paper, we compare proposed method with the Lim's method that use the 3D-DCT in uncompress domain. We estimate the performance of our method on the invisibility and robustness of watermarking. We use the PSNR (peak signal to noise ratio) to estimate the performance of invisibility and the detection ratio of watermarks to estimate the performance of robustness.

5.1 Invisibility Experiment

Table 1 shows the result of watermark invisibility test. Generally, if the PSNR of video sequence is higher than 50dB, the video sequence is regarded as the original video sequence and if the PSNR of video sequence is higher than 40dB, it is known that video quality is superior. In our method, the minimum value of PSNR is higher than 40dB on the full stream of each experimental video and the average values of PSNR are 42~45dB on each experimental video.

Table 1. Average PSNR of each experimental video

	Miss America	Table Tennis	Foreman	Football
Average PSNR	44.84	42.42	44.48	42.32

5.2 Robustness Experiment

First, we must select a suitable threshold T_r to decide whether a certain watermark is exists in the video sequence. Therefore, we do experiments for confirming the empirical threshold of the detection ratio R and we can assume $T_r = 0.75$ based on the probability density function of detection ratio R [10]. Figure 5 shows the probability densities of detection ratios on the non-watermarked and the watermarked signals. We decide the detection of watermark if the detection ratio is greater than 0.75 (75%).

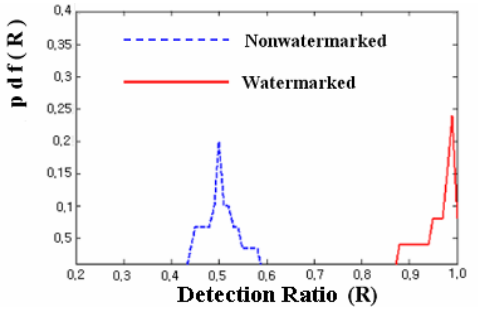


Fig. 5. Probability density of detection ratio on non-watermarked and watermarked signal

Table 2 shows the result of watermark robustness test against MPEG-2 compression attack at 5Mbps and 2Mbps. As shown table 2, the proposed method is sufficiently more robust than the previous method against MPEG-2 compression attack.

Table 2. Average detection rate under MPEG-2 attack

	Miss America		Table Tennis	
	Previous method	Proposed method	Previous method	Proposed method
5Mbps	99.99%	100%	95.83%	99.40%
2Mbps	97.55%	99.88%	92.95%	93.81%
	Foreman		Football	
	Previous method	Proposed method	Previous method	Proposed method
5Mbps	97.49%	99.96%	94.44%	99.17%
2Mbps	84.39%	96.90%	78.37%	88.45%

Table 3 shows the performance improvement of watermark robustness test against MPEG-2 compression attack.

Table 3. Performance improvement under MPEG-2 attack

	Miss America	Table Tennis	Foreman	Football
5Mbps	0.01%	3.57%	2.47%	4.73%
2Mbps	2.23%	2.86%	12.51%	10.08%

Table 4 shows the result of watermark robustness test against MPEG-2 coding attack and temporal dropping attack.

Table 4. Detection under MPEG-2 attack and frame dropping attack

	Miss America		Table Tennis	
	Previous method	Proposed method	Previous method	Proposed method
5Mbps+1 frame dropping	Detect	Detect	Detect	Detect
5Mbps+2 frame dropping	Detect	Detect	Not Detect	Detect
	Foreman		Football	
	Previous method	Proposed method	Previous method	Proposed method
5Mbps+1 frame dropping	Not Detect	Detect	Not Detect	Detect
5Mbps+1 frame dropping	Not Detect	Detect	Not Detect	Detect

Table 5 shows the result of watermark robustness test against MPEG-2 coding attack and temporal insertion attack.

Table 5. Detection under MPEG-2 attack and frame insertion attack

	Miss America		Table Tennis	
	Previous method	Proposed method	Previous method	Proposed method
5Mbps + 1 frame insertion	Detect	Detect	Detection	Detect
5Mbps + 2 frame insertion	Detect	Detect	Not Detect	Not Detect
	Foreman		Football	
	Previous method	Proposed method	Previous method	Proposed method
5Mbps + 1 frame insertion	Not Detect	Detect	Not Detect	Detect
5Mbps +2 frame insertion	Not Detect	Detect	Not Detect	Detect

Table 6 shows the result of watermark robustness test against MPEG-2 coding attack and temporal swapping attack.

Table 6. Detection under MPEG-2 attack and frame swapping attack

	Miss America		Table Tennis	
	Previous method	Proposed method	Previous method	Proposed method
5Mbps + 2 frame swapping	Detect	Detect	Detect	Detect
	Foreman		Football	
	Previous method	Proposed method	Previous method	Proposed method
5Mbps +2 frame swapping	Detect	Detect	Detect	Detect

6 Conclusions

In this paper, we propose the adaptive blind video watermarking scheme utilizing the video characteristics. To optimize the weight factors for watermarking, we classify the pattern of 3D-DCT cubes and the type of video segments by using the texture and motion information of 3D-DCT cubes. Then we embed an optimal watermark into the mid-range coefficients of 3D-DCT cubes by using the trained optimal weight factors. In experimental results, detection ratio is average 99.63% against 5Mbps MPEG compression attack and average 94.76% against 2Mbps MPEG compression attack. The minimum value of PSNR is higher than 40dB on the full stream of each experimental video and the average values of PSNR are 42~45dB on each experimental video. Therefore, the proposed method achieves better performance in terms of invisibility and robustness of watermarking and especially achieves better robustness than previous method on videos with high textures and high motion.

Since the proposed method relies on the temporal and spatial synchronization, it is assumed that the grouping of 8*8*8 cubes is identical for the watermark insertion and the detection. Therefore, this method may be vulnerable to de-synchronization attacks such as scaling, frame dropping or frame insertion. This problem may be overcome by adding some spatiotemporal synchronization methods, which will be future work.

Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication) of Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

[1] Swanson, M. D., Zhu, B., Tewfik, A.: Data Hiding for Video in Video and Other Applications, IEEE Int. Conf. on Image Processing, Oct. 1997
[2] Hartung, F., Girod, B.: Watermarking of Uncompressed and compressed Video, Signal Processing, Vol.66, no.3, pp.283-301, 1998
[3] Ge, Q. M., Lu, Z. M., Niu, X. M.: Oblivious Video Watermarking Scheme with Adaptive Embedding Mechanism, Int. Proc. Machine Learning and Cybernetics, 2003
[4] Kim, S. J., Kim, T. S., Kwon, K. R., Ahn, S. H., Lee, K. I.: Digital Watermarking Based on Three-Dimensional Wavelet Transform for Video Data, LNCS 3768, 2005
[5] Lim, J. H., Kim, D. J., Kim, H. T., Won, C. S.: Digital Video Watermarking Using 3D-DCT and Intra-Cubic Correlation, Proc. of SPIE, vol.4314, pp.64-72, 2001
[6] Lee, M. C., Chan, R. K. W., Adjeroh, D. A.: Quantization of 3D-DCT Coefficients and Scan Order for Video Compression, Journal of Visual Communication and Image Representation, 1997
[7] Watson, A. B.: DCT Quantization Matrices Visually Optimized for Individual Images, Human Vision, Visual Processing, and Digital Display IV, Proc. SPIE 1913, pp. 202-216, 1993
[8] Moon, J. Y., Ho, Y. S.: A Video Watermarking Algorithm Based on the Human Visual System Properties, ISCIS 2003
[9] Wu, G., Zhuang, Y., Wu, F., Pan, Y.: A Novel Watermarking Scheme Based on Video Content, LNCS 3334, 2004
[10] Wu, G., Zhuang, Y., Wu, F., Pan, Y.: Self-Adaptive MPEG Video Watermarking based on Block Perceptual Features, Int. Proc. Machine Learning and Cybernetics, Vol. 6, 2004

Scalable Protection and Access Control in Full Scalable Video Coding

Yong Geun Won, Tae Meon Bae, and Yong Man Ro

IVY Lab., Information and Communication University (ICU),
119, Munjiro, Yuseong-gu, Deajeon, 305-714, Korea
yro@icu.ac.kr

Abstract. In this paper, we propose an encryption algorithm to protect scalable video coding (SVC) bitstream and a method for conditional access control to consume the encrypted SVC bitstream. To design an encryption algorithm, we analyzed the encryption requirements to support scalability function in the scalable video and proposed an effective encryption method developed in the network abstraction layer (NAL). In addition, conditional access control scheme and key management scheme are proposed to consume the SVC bitstream protected with proposed method. Experiments were performed to verify the proposed method and results showed that proposed algorithms could provide an effective access control of scalable video as well as support a video encryption with scalability function.

1 Introduction

The development of video applications in cooperation with various transmission networks has lead to diverse usage environments for video consumption. For any given usage environment, video adaptation is a vital issue for guaranteeing the quality of service (QoS). When a video is supposed to be adapted to a certain usage environment, conventional non-scalable video coding scheme likely requires decoding/re-encoding of the video to meet the usage environments. Therefore the non-scalable contents could give a load to the adaptation server in both computational complexity and hardware capacity.

Until now, various scalable coding schemes designed for efficient adaptation have been proposed. JPEG2000 is a wavelet based scalable image coding [1]. MPEG-4 FGS is a scalable video coding which provides SNR and temporal scalability [2]. Unfortunately, the previous scalable video coding scheme, MPEG-4 FGS, has limitations for coding efficiency and scalability type. Joint video team (JVT) of MPEG and ITU-T has developed a new scalable video coding scheme, which provides three scalabilities such as spatial, temporal, and SNR with guaranteed adequate coding performance. Early 2005, JVT announced joint scalable video model (JVSM) for SVC, and has currently continued to develop.

Recently, diverse video application and easy access through networks brings us secure consumption as well as adaptation. To encrypt a video, a region related with contents characteristics is used for high security [3][4]. In scalable contents encryption, the enhancement structure for the scalability was considered.[5]

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 407 – 421, 2006.

Base-layer encryption could provide enough security of the whole bitstream because scalable contents are enhanced from the base layer. However Base-layer encryption causes loss of the scalability when the video is adapted.

When a user's access right is not enough to consume the whole bitstream, SVC protection scheme should not allow the access. To accomplish this objective, SVC encryption should consider multi-layer structure.

In this paper, we analyzed SVC bitstream structure in encoding process and derived encryption requirements to support scalability of contents. Based on this, we propose an efficient encryption algorithm which can conserve SVC characteristics despite adaptation. Also we propose a conditional access control method to consume the SVC bitstream protected by proposed scheme. We performed experiments with SVC standard software, called Joint Scalable Video Model (JSVM), and verified the usefulness of the proposed method.

2 SVC Scalable Protection and Consumption

2.1 Overview of Scalable Video Coding

SVC provides spatial, temporal, and SNR scalability with high coding efficiency. In SVC, the spatial scalability is achieved by layered coding and the temporal scalability is achieved by hierarchical B picture structure. In the case of SNR scalability, fine granular scalability (FGS) and coarse granular scalability (CGS) are employed. SVC bitstream consist of a base layer and several enhancement layers. By decoding the base layer, the lowest quality of original video can be obtained. Enhancement layers are added on the base layer to get a better quality [6]. Figure 1 shows the enhancement structure for spatial, temporal, SNR scalability in SVC.

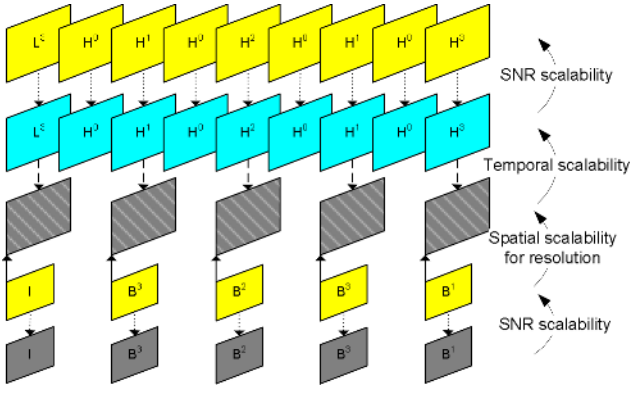


Fig. 1. Spatial, Temporal, SNR, enhancement structure in SVC bitstream

In order to deliver SVC bitstream in diverse transport environments, video coding layer (VCL) and network abstraction layer (NAL) are separated in joint video team (JVT). NAL provides abstracted transport information about VCL data within JVT standards [7][8]. In SVC, each slice could be a NAL unit, and its size is variable.

Figure 2 shows NAL types and the NAL unit structure. In the NAL header, scalability information is included as shown in the shaded part in Figure 2. And NAL data contains encoded video data.

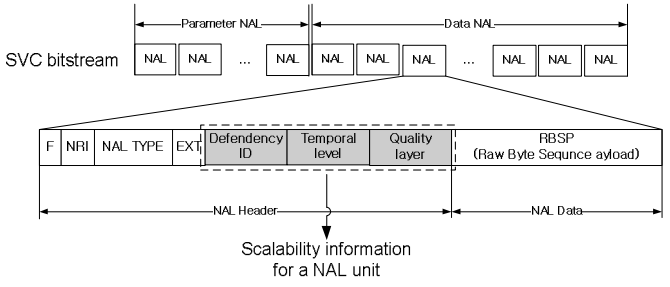


Fig. 2. NAL unit syntax in SVC bitstream

To adapt the video according to the environment, corresponding bitstream needs to be extracted. In SVC, NAL unit is the smallest unit through which spatial, temporal, SNR scalability can be defined. Scalability information is written in the NAL header. So the protection of content will be performed based on NAL unit.

As is seen, extraction for spatial and temporal scalability is achieved by dropping the NAL unit which is fundamental unit. On the other hand, the extraction that controls SNR scalability is achieved by cropping the NAL unit. Figure 3 shows the extraction in SVC. Figure 3 (a) shows extraction by NAL unit dropping to control spatial and temporal scalability while Figure 3 (b) shows extraction by NAL unit cropping to control SNR scalability.

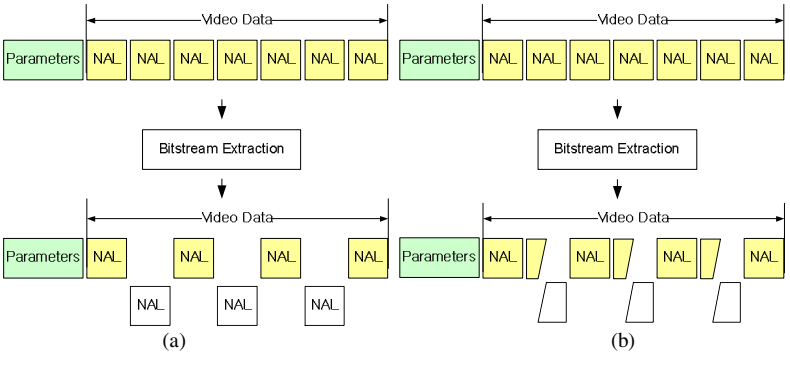


Fig. 3. Bitstream extraction for the SVC bitstream, (a) NAL unit dropping for spatial and temporal scalability, (b) NAL unit cropping for SNR scalability

2.2 Scalable Protection for SVC

Based on the characteristics of SVC mentioned above, we analyzed the SVC contents and derived requirement to achieve SVC protection with full scalability. The

requirements for SVC encryption are as follows: First, Not only encrypting the base layer but also encrypting the enhancement layer is needed so that the protected SVC bitstream can be accessed in a scalable manner. Therefore encryption should be performed at all layers in the SVC bitstream. In SVC, each enhancement layer has different kinds of data such as texture, motion vector, and FGS data. Thus, the encryption algorithm should apply all types of data in the SVC bitstream.

Second, SVC protection should employ an encryption scheme that can be applicab le in bitstream extraction process. Since bitstream extraction could be performed outsi de a trusted server such as the decoder, it should be done in NAL units without decryp tion and encryption process. In addition, protected NAL unit is cropped for bitstream extraction in SNR enhancement layers. SVC bitstream encryption should consider the se two kinds of bitstream extraction.

Third, the encryption scheme should be light-weighted. Since SVC has complex encoding and decoding process, light-weighted encryption algorithm is needed to avoidadditional complexity. Light-weighted algorithm could satisfy a real-time application as well.[9]

From the requirement described above, we design the protection scheme in SVC. The proposed encryption algorithm in this paper is performed directly to video data and so video decoding without decryption shows a noisy display. The proposed algorithm is also applicable to bitstream extraction because the NAL unit encryption is employed. Moreover the proposed protection scheme is fast and easy to implement, because it randomly inverts the sign bit of the data before the context adaptive binary arithmetic coding(CABAC) stage.

For the encryption in NAL unit, we need to distinguish data NAL unit from other NAL units. As Fig. 2 shows, using *NAL_TYPE* in the NAL header, we can get the data NAL unit information [6]. The video data of the NAL unit has three data types such as texture, motion vector, and FGS data. Among them, texture data is more effective for the encryption. However, texture data encryption alone could not provide full enhancement layer protection. Because texture data in the enhancement layer is predicted from the base layer, full bitstream protection can not be achieved when base layer is decrypted. Therefore motion vector and FGS data should also be encrypted. Because FGS data could be truncated in arbitrary point to meet the given bit rate, the proposed SVC encryption should support these requirements as well.

Figure 4 shows SVC coding scheme in a layer. As see in Figure 4, texture, motion, FGS data are transmitted to CABAC. Before the CABAC, FGS, texture, and motion data are divided into sign and absolute value. The sign values of the texture, FGS, and motion data are inverted with a random value which comes from pseudo random number generator as a lightweight encryption. Moreover, the proposed encryption algorithm doesn't affect coding efficiency. Because it inverts only sign bit of the data. The *Seed* which is used for random value generation is previously generated and inserted into encrypted SVC bitstream using conventional data encryption algorithm such as DES(data encryption algorithm) or AES(advanced encryption algorithm) [10]. By separating *Seed* and key, we can get different random values with the same key. The seed needs to be transmitted with SVC bitstream. Because current SVC syntax has no place to put the seed, a new syntax for containing the seed is required.

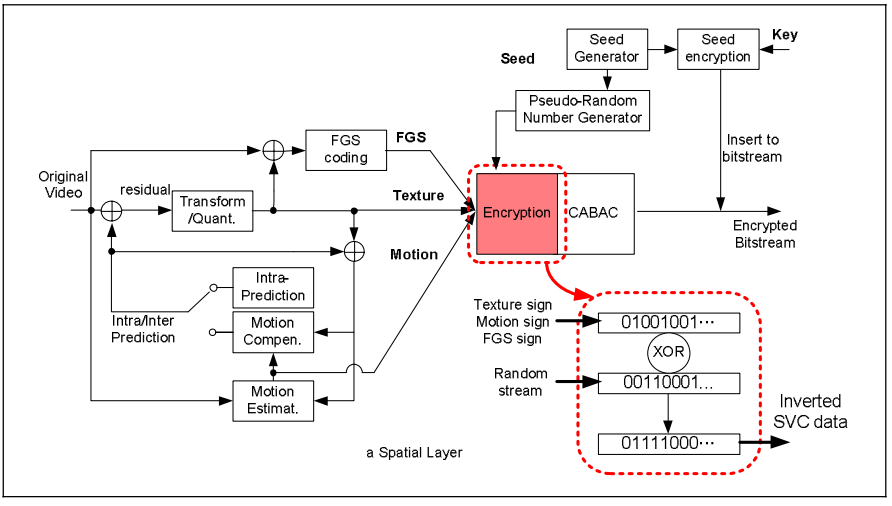


Fig. 4. Proposed encryption scheme for one layer in SVC

2.3 Encrypted SVC Bitstream Consumption

The encrypted SVC bitstream with the proposed method is able to be consumed in a form of a scalability converted bitstream by the bitstream extraction process. The protected SVC bitstream should be consumed by a trusted user who has the rights for accessibility. The proposed algorithm makes it possible for the trusted user who has rights to access a higher layer to also access a lower layer. This can be achieved by the proposed conditional access control.

For the access control in SVC bitstream, we considered the fundamental unit of scalability. Spatial, temporal, and SNR scalabilities are enhanced by a form of a layer or level in SVC. Each layer or level is a fundamental unit for the scalability. A layer or level is classified by NAL unit which can be considered as a fundamental unit for the encryption. A NAL unit is encrypted with different keys depending on the scalability to be achieved. The key encrypting and decrypting the NAL unit is denoted as *NAL unit key* in this paper. As seen in Fig. 2, we can obtain scalability information such as *dependent_ID, temporal_level*, and *quality_layer* from the NAL header.

Figure 5 shows an example of the access control scheme. It has two spatial layers (spatial 0, spatial 1), two temporal levels (temporal 0, temporal 1), and two quality layers (SNR 0, SNR 1). If a user wants to access s spatial layers, t temporal layers, and q SNR layers, a set of the keys is needed to decrypt the encrypted NAL units. As shown in Figure 5, *NAL unit key* is represented as *Key*(*spatial layer, temporal level, SNR layer*).

The number of keys needed to encrypt SVC bitstream can be written as

$$Key_{Total} = \sum_{s=1}^{NS} (NQ_s \times NT_s), \tag{1}$$

where NS is the number of spatial layers, NQ_S is the number of SNR layers in s-th spatial layer, and NT_S is the number of temporal levels in s-th spatial layer. Note that SNR and temporal scalabilities are related with corresponding spatial layer.

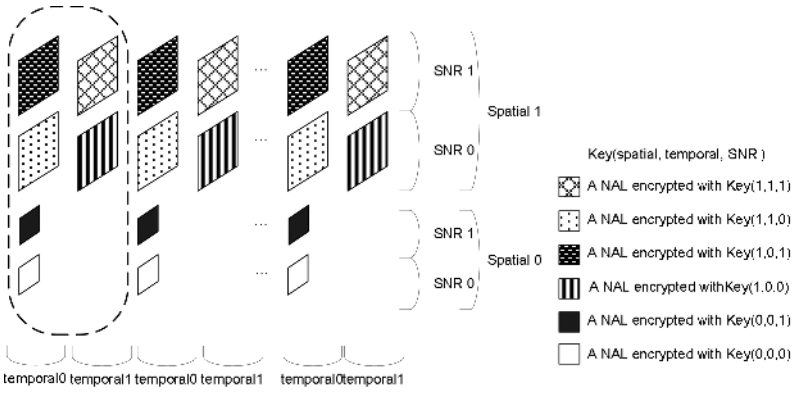


Fig. 5. NAL unit encryption for conditional access control

From the proposed conditional access control scheme depicted in Fig. 5, the key set to access the SVC bitstream with s spatial layer, t temporal level, and q SNR layer can be written as

$$KEYSET_{s,t,q} = \left\{ Key(l,m,n) \;\middle|\; \begin{array}{l} 0 \le l \le s, \\ 0 \le m \le \min(t, NT_s), \\ if \quad l = s \quad then \quad 0 \le n \le q, \;\; else \quad 0 \le n \le NQ_s \end{array} \right\}, \tag{2}$$

An example of key set list for different access control depending on the scalability is shown in Table 1.

Table 1. The *NAL unit key* set list needed to access SVC bitstream with certain scalability. It is calculated by Eq. (2) from the SVC example of Fig. 5.

Spatial and SNR layer \ Temporal layer		Temporal 0	Temporal 1
Spatial 0	SNR 0	*{Key(0,0,0)}*	*Not exist*
	SNR 1	*{Key(0,0,0),Key(0,0,1)}*	*Not exist*
Spatial 1	SNR 0	*{Key(0,0,0),Key(0,0,1) Key (1,0,0) }*	*{Key(0,0,0),Key(0,0,1) Key(1,0,0),Key(1,1,0)}*
	SNR 1	*{Key (0,0,0),Key(0,0,1) Key (1,0,0),Key(1,0,1)}*	*{Key(0,0,0),Key(0,0,1) Key(1,0,0),Key(1,0,1) Key(1,1,1),Key(1,1,0)}*

2.4 Key Management in Bitstream Consumption

Accessing SVC bitstream with multiple keys mentioned above would cause the increase of complexity in both contents provider and terminal. Previously, key management scheme to reduce the number of key used in encryption was proposed [11][12]. In the scalable contents, accessing higher layer needs all lower layers. In this paper, we employed the key management scheme to reduce to multiple keys mentioned above.

To do so, we define *master key, type key, layer key*, and *access key*. The *master key* is the key assigned to one video bitstream, which further generates *type keys*. The *type key* is the key showing scalability type, which further generates the highest *layer key* in its type. The *layer key* is the key assigning a layer in a given scalability type, further generates lower *layer keys*. The *access key* is a key which could generate all *NAL unit keys* which is needed to decrypt all NAL units to access extracted bitstream with a given access rights.

SVC has spatial, temporal, and SNR scalability and corresponding *type keys* need to be generated from a *master key*. When SVC bitstream is enhanced in spatial, e.g., from one spatial layer S_0 to enhanced spatial layer S_1, all SNR layers in spatial layer S_0 are needed. This means that the spatial and SNR layers are dependent. Therefore, in this paper, spatial scalability and SNR scalability are categorized into the same type, which means the same *type key* is used to generate the *layer keys* in spatial and SNR scalability. Figure 6 shows two types in the SVC bitstream for example. As is seen, spatial and SNR layers are included in type 1 and temporal layers are included in type 2.

The *master key*, K is assigned to a SVC contents and two *type keys* (K_j) are generated as follows,

$$K_j = H(K \parallel j), \tag{3}$$

where $H(\cdot)$ is a cryptographic hash function, K is the *master key*, j represent spatio-SNR scalability type (j=1) or temporal scalability type(j=2), and $\parallel$ denotes concatenation operator.

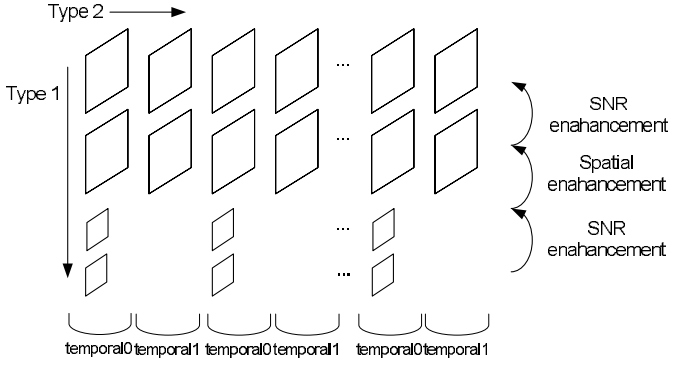


Fig. 6. Redefined two scalability types to categorize the *type key*

For the *layer key*, highest *layer key* is generated by hashing the *type key*. As is seen in SVC structure, to access a higher layer in SVC content, lower layers are needed. So a *layer key* is generated by hashing higher *layer key*. The *layer key* $K_{i,j}$ for i-th layer in j-th scalability type can be obtained as,

$$\begin{aligned} K_{i,j} &= \begin{pmatrix} H(K_{i+1,j}), \text{for } 1 \le i < n_j \\ H(K_j), \quad \text{for } i = n_j \end{pmatrix} \\ &= H^{n_j+1-i}(K_j), \text{for } 1 \le i \le n_j, \end{aligned} \tag{4}$$

where n_j is the number of layer in the j-th scalability type. $H^m(x)$ is a cryptographic hash function of x applied m times.

Using the *layer key* in Eq. (4), therefore, the *NAL unit key* of Key(*s,t,q*), that is used to encrypt/decrypt the NAL unit for (*s, t, q*)scalabilities can be written as

$$K(s,t,q) = K_{a,1} \parallel K_{b,2}\,, \tag{5}$$

where $K_{a,1}$ is the *layer key* for layer *a* in type 1, $K_{b,1}$is the *layer key* for layer *b* in the type 2, $a = \sum_{x=1}^{s-1} NQ_x + q$, and $b = t$.

In this paper, *access key* is the *NAL unit key* for given scalabilities to be accessed, All the *NAL unit keys* which are needed to access to the given scalabilities are generated by *access key*. Table 2 shows the *access keys* generated by the proposed key management scheme. As is seen in Table 2, the *access keys* for the example of Fig.5 are generated.

Table 2. *Access keys* by the proposed algorithm from the SVC example of Fig. 5

Spatial And SNR layer \ Temporal layer		15 fps	30 fps
QCIF	Base	$K_{0,1} \parallel K_{0,2}$	Not exist
	FGS	$K_{1,1} \parallel K_{0,2}$	Not exist
CIF	Base	$K_{2,1} \parallel K_{0,2}$	$K_{2,1} \parallel K_{1,2}$
	FGS	$K_{3,1} \parallel K_{0,2}$	$K_{3,1} \parallel K_{1,2}$

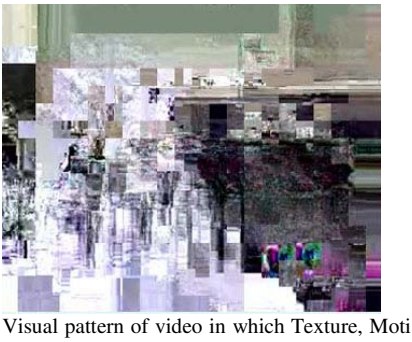
Table 3. *NAL unit key* set generated by the *access key* from the SVC example of Fig. 5

Spatial and SNR layer \ Temporal layer		15 fps		30 fps	
		access key	*NAL unit key* set	*access key*	*NAL unit key* set
QCIF	Base	$K_{0,1} \parallel K_{0,2}$	$K_{0,1} \parallel K_{0,2}$	Not exist	Not exist
	FGS	$K_{1,1} \parallel K_{0,2}$	$K_{0,1} \parallel K_{0,2}$ $K_{1,1} \parallel K_{0,2}$	Not exist	Not exist
CIF	Base	$K_{2,1} \parallel K_{0,2}$	$K_{0,1} \parallel K_{0,2}$ $K_{1,1} \parallel K_{0,2}$ $K_{2,1} \parallel K_{0,2}$	$K_{2,1} \parallel K_{1,2}$	$K_{0,1} \parallel K_{0,2}$ $K_{1,1} \parallel K_{0,2}$ $K_{2,1} \parallel K_{0,2}$ $K_{2,1} \parallel K_{1,2}$
	FGS	$K_{3,1} \parallel K_{0,2}$	$K_{0,1} \parallel K_{0,2}$ $K_{1,1} \parallel K_{0,2}$ $K_{2,1} \parallel K_{0,2}$ $K_{3,1} \parallel K_{0,2}$	$K_{3,1} \parallel K_{1,2}$	$K_{0,1} \parallel K_{0,2}$ $K_{1,1} \parallel K_{0,2}$ $K_{2,1} \parallel K_{0,2}$ $K_{3,1} \parallel K_{0,2}$ $K_{2,1} \parallel K_{1,2}$ $K_{3,1} \parallel K_{1,2}$

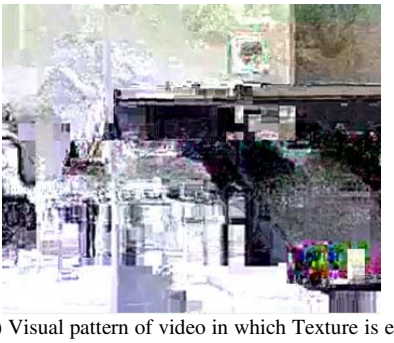
As is seen in Table 2, for example, the video of CIF, base layer quality, and 15fps can b e accessed with access key of $K_{2,1}||K_{0,2,}$ By Eq. (4), the *access key* can generate the *NAL unit key set* { $K_{0,1}||K_{0,2,}$ $K_{1,1}||K_{0,2,}$ $K_{2,1}||K_{0,2,}$ } which is needed to decrypt the bitstream. Table 3 shows the *NAL unit key* set generated by the *access key* for the example of Fig. 5.

As mentioned in the proposed method above, the SVC bitstream can be encrypted by a single *master key* and one can access the video with *access key* generated from the *master key*.

(a) Origianl video



(b) Visual pattern of video in which Texture, Moti on vector and FGS are encrypted



(c) Visual pattern of video in which Texture is en crypted

(d) Visual pattern of video in which Motion vector is encrypted

(e) Visual pattern of video in which FGS is encry pted.

Fig. 7. Visual patterns of decoded video corrupted by no decryption key for respective texture, motion vector, and FGS encrypted videos

3 Experimental Results

We have implemented the proposed methods in the JSVM 2.0 [6]. The "BUS" sequence that is the test sequence of MPEG SVC is used in the experiment. The test sequence is encoded by 2 spatial layers (CIF, QCIF), 2 temporal levels (15fps, 30fps) and 2 SNR layers (base quality, FGS quality). Two kinds of experiments are performed: One is the experiment to verify the proposed SVC bitstream encryption method which meets the SVC encryption requirements mentioned in section 2. The other is the experiment for the conditional access control for SVC bitstream protected by the proposed method.

For the first experiment, figure 7 shows visual pattern of video without decryption key for the video encrypted by proposed method. SVC content has multi-layer structure and different data types for texture, motion vector and FGS. To protect entire SVC contents, encryption should be performed for all data types.

As seen in Fig. 7, texture encryption is effective. However, texture data in the enhancement layer are predicted from the base layer, thereby the protection of enhanced layer only is not enough in the case that base layer is decrypted. To solve the problem, motion vector is encrypted in addition. Figure 8 shows visual effect of the decoded result when motion vector and texture are encrypted and the base layer is decrypted.

(a) Visual pattern of video with Texture only encryption

(b) Visual pattern of video with Texture and motion vector encryption

Fig. 8. Visual patterns of decoded video corrupted by no decryption key for respective texture and texture+motion vector encrypted videos when base-layers are not encrypted

Table 4 shows PSNR of the decoded result in Fig. 8.

Table 4. PSNR of decoded video for texture encryption and texture + motion vector encryption when base-layers are not encrypted

Encryption data	PSNR Y	PSNR U	PSNR V
Texture	21.7585	37.3057	38.0141
Texture + motion vector	18.8713	35.7341	35.4734

The proposed SVC encryption algorithm is able to work during bitstream extraction. As mentioned above, there are two kinds of bitstream extraction; NAL unit dropping and NAL unit cropping.

Figure 9 shows the case when NAL units for FGS layer are cropped to satisfy a target bitrate. Especially, case (a) and case (b) show NAL unit dropping extraction for FGS data. Because FGS data is residual between original data and reconstructed data, encryption is relatively less effective compared with the effect in case of spatial or temporal encryption.

(a) Visual pattern of video when all FGS layers a re enhanced in the FGS encryption case

(b) Visual pattern of video when all FGS layers are enhanced in the FGS decryption case

(c) Visual pattern of video when 50% of FGS layers are enhanced in the FGS encryption case

(d) Visual pattern of video when 50% of FGS layers are enhanced in the FGS decryption case

(e) Visual pattern of video when FGS layers are n ot enhanced in the FGS encryption case

(f) Visual pattern of video when FGS layers are not enhanced in the FGS decryption case

Fig. 9. SNR enhancement layers which consist of FGS data could be extracted by cropping NAL unit. (a), (c), (e) is the cases that FGS data is encrypted. (b), (d), (e) is the cases that FGS data is not encrypted.

Table 5 shows the PSNR of decoded results for bitstream extraction for Fig. 9. PSNR results show that FGS enhancement leads to higher PSNR of decoded sequence when FGS is decrypted whereas when FGS is not decrypted, FGS enhancement lowers the PSNR of decoded sequence.

Table 5. PSNR results for each FGS extraction case

FGS data is encrypted cases				FGS data is decrypted cases			
FGS enhancement	PSNR Y	PSNR U	PSNR Y	FGS enhancement	PSNR U	PSNR V	PSNR V
100%	24.0170	35.8473	35.2499	100%	28.9284	38.6394	39.3413
50%	25.0704	36.1773	35.6802	50%	27.0394	38.0490	38.5241
0%	25.8685	36.9999	37.0809	0%	25.8685	36.9999	37.0809

In the second experiment, we performed conditional access control to verify the proposed method. For the experiment, we set the arbitrary *access condition* and five different *access rights*. Each *access right* permits the accessibility for its scalability. Table 6 shows five access rights and corresponding *access key*s.

Table 6. Access condition and corresponding key set to access

Given bits tream	Case	*access right*	*access key*	*NAL unit key* sets
Two layer SVC (CIF 30fps Base + QCIF 15fps, FGS)	1	Don't have access right	No key	No key
	2	QCIF, 15fps, Base quality	$K_{0,1} \| K_{0,2}$	*{key(0,0,0)}*
	3	QCIF, 15fps, FGS quality	$K_{1,1} \| K_{0,2}$	*{key(0,0,0), key(0,0,1)}*
	4	CIF, 15fps, Base quality	$K_{2,1} \| K_{0,2}$	*{key(0,0,0), key(0,0,1), key(1,0,0)}*
	5	CIF, 30fps, Base quality	$K_{2,1} \| K_{1,2}$	*{key(0,0,0), key(0,0,1), key(1,0,0) ,key(1,1,0)}*

Figure 10 shows the decoded results with different *access rights*. Case 5 shows that all layers are decrypted and rightly displayed using an *access key* for access condition. The other cases, however, are partially decrypted or not decrypted at all. The visual effects look different depending on the access right.

Table. 7 and Figure 10 show the visual quality for correct access right and that for wrong access right. The visual quality for correct access right is the maximum quality that can be achieved for a given access right as shown in Table 6. It is the decoded quality of the extracted bitstream with the scalability for corresponding access right. The visual quality for wrong access right is the decoded quality of the video of two layers SVC (CIF 30fps Base + QCIF 15fps, FGS) that is decrypted using the *access key* shown in Table 6, which is wrong to access two layers of SVC. As experimental results show, the visual quality for wrong access right is always equal or lower than the visual quality for correct access right. Therefore trying to access a higher layer with a given access right causes the degradation of the video quality.

Case 1

Case 2

Case 3

Case 4

Case 5

Fig. 9. Visual patterns of decoded video for the given bitstream with *access key* of corresponding access right

Table 7. PSNR results for *access right quality* and *forced access quality*

	visual quality for correct access right			*visual quality for wrong access right*		
Case	PSNR Y	PSNR U	PSNR V	PSNR Y	PSNR U	PSNR V
1	Not accessible	Not accessible	Not accessible	8.7364	25.2576	28.4450
2	27.3808	38.3680	38.4299	18.3491	34.6364	34.0368
3	28.3098	39.6567	40.0774	18.4750	35.7254	35.1707
4	27.0986	38.0180	38.4488	22.1120	36.0828	36.0237
5	26.7798	37.9623	38.4204	26.7798	37.9623.	38.4204

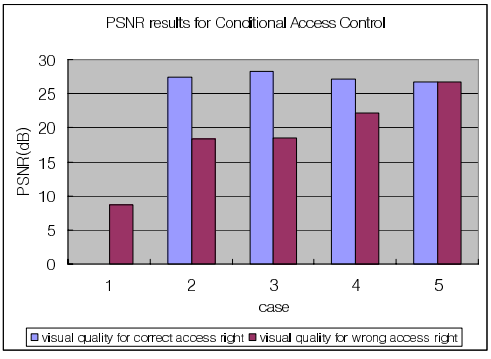


Fig. 10. Comparison between the visual qualities for correct access right and wrong access right

4 Conclusion

In this paper, we proposed an efficient encryption algorithm for SVC bitstream and corresponding conditional access control. For the SVC encryption, we analyzed the requirements of the SVC encryption. Based on the analysis, we proposed SVC encryption method. Also, we proposed conditional access control using selective decryption for encrypted SVC bitstream and key management scheme that reduces the number of key used in the encryption. Experimental results show that the proposed algorithms satisfy SVC encryption requirements and provide conditional access control supporting full scalability.

Reference

1. Information Technology – JPEG2000 Image Coding System Part1 : Core Coding system, ISO/IEC 15444-1:2000 ISO/IEC JTC/SC 29/WG 1 N1646R, March 2000.
2. Mihaela van der Schaar, Hayder Radha : A Hybrid Temporal-SNR Fine-Granular Scalability for Internet Video : IEEE Transaction on circuits and systems for video Technology Vol. 11, no. 3, march (2001)
3. Iskender Agi, Li Gong, : An Empirical Study of Secure MPEG Video Transmissions : in proc. Internet society symposium. Network & Distributed system security, Feb. (1996) 137-144,
4. Xiliang Liu, Ahmet M. Eskicioglu, : Selective Encryption of Multimedia Contents in Distribution Network: Challenges and New Directions : IASTED International Conference on Communications, Internet and Information Technology (CIIT 2003), Scottsdale, AZ, November (2003) 17-19,
5. Bin B. Zhu, Mitchell D. Swanson, and Shipeng Li, : Encryption and Authentication for Scalable Multimedia: Current State of the Art and Challenges," Proc. SPIE Internet Multimedia Management Systems V, vol. 5601, Oct. (2004) 157-170
6. ISO/IEC JTC 1/SC 29/WG 11N 7084 : Joint Scalable Video Model (JSVM) 2.0 Reference Encoding Algorithm Description. April (2005), Buzan, Korea
7. Thomas Stockhammer, Miska M. Hannuksela, Stephan Wenger : H.26L/JVT coding network abstraction layer and ip-based transport : IEEE ICIP, Vol. 2, (2002) 485-488,

8. Thomas Wiegand, Gary J. Sullivan, Ajay Luthra, Aharon Gill: Text of ISO/IEC FDIS 14496-10: Information Technology – Coding of audio-visual objects – Part 10: Advanced Video Coding. ISO/IEC FDIS 14496-10: (2003)
9. C. Shi, B. Bhargava : A fast MPEG video encryption algorithm, Proc. ACM multimeida98, Sep (1998) 81-88,.
10. Rapha¨el Grosbois, Pierre Gerbelot, and Touradj Ebrahimi : Authentication and access control in the JPEG 2000 compressed domain : Proc. of the SPIE 4472, Jul (2001) 95-104
11. Bin B. Zhu, Shipeng Li, Min Feng : A Framework of Scalable Layered Access Control for Multimedia, Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on 23-26 May (2005) Page(s):2703 - 2706 Vol. 3
12. B. B. Zhu, M. Feng, and S. Li, : An Efficient Key Scheme for Layered Access Control of MPEG-4 FGS Video, IEEE Int. Conf. on Multimedia and Expo, June (2004) 443-446

A Wavelet-Based Fragile Watermarking Scheme for Secure Image Authentication

HongJie He[1], JiaShu Zhang[1], and Heng-Ming Tai[2]

[1] Sichuan Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu, Sichuan, 610031 China
[2] The Electrical Engineering Department, the University of Tulsa, Tulsa, OK 74104, USA

Abstract. This paper proposes a wavelet-based fragile watermarking scheme for secure image authentication. In the proposed scheme, the embedded watermark is generated using the discrete wavelet transform (DWT), and then the improved security watermark by scrambling encryption is embedded into the least significant bit (LSB) of the host image. The proposed algorithm not only possesses excellent tamper localization properties and greater security against many attacks, but also demonstrates a new useful feature that can indicate whether the modification made to the image is on the contents or the embedded watermark. If only the watermark is modified, the authenticity of the image is assured, instead of being declared as a counterfeit. Experimental results illustrate the effectiveness of our method.

Keywords: fragile watermarking; discrete wavelet transform (DWT); the vector quantization attack; the transplantation attack.

1 Introduction

Fragile watermarks are designed to protect the authenticity and integrity of digital images by detecting changes in an image [1-2]. Other than having the property of thwarting a wide spectrum of attacks including vector quantization attack and transplantation attack [3-6] for secure communication applications, fragile watermarking schemes typically have the functionalities for image authentication and tamper localization [3-9]. However, the feature of distinguishing whether the tampering is on image contents or on embedded watermarks, which might be important to practical applications, has not been addressed in the literature.

For digital image, the modification of its contents and of the watermark is not the same. Content alteration destroys the integrity and authenticity of the image, while the tampered watermark does not affect the authenticity of the image. Therefore, the verification process in a watermarking system should be able to detect and localize exactly where the contents are tampered. At the same time, it also should authenticate the image if the alteration is only on the watermark. This task is called the tamper discrimination. For example, if only the watermark is modified, the verification algorithm should indicate that the image is authentic

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 422–432, 2006.

and can be used as desired rather than regarded it as a fake. Unfortunately, the current fragile watermarking algorithms have the tamper localization capability, but do not have the capability of distinguishing these two alterations. As a result, attacker can forger the digital media by tampering the embedded watermark only, not contents, so as to confuse the system and to make the image fail the verification process. The mere existence of such a flaw indicates a weakness in the schemes.

To overcome this problem, we present a wavelet-based fragile watermarking scheme for image authentication and tamper discrimination. In proposed algorithm, the embedded watermark is generated using the discrete wavelet transform (DWT), and then the improved security watermark scrambled by scrambling encryption is embedded into the LSB of the image data. The strategy of a scrambling encryption can not only extent the ability to discriminate the watermark tampering from the content tampering, but also increase the security against VQ attack and transplantation attack. In addition, the proposed algorithm possesses excellent tamper localization properties. All the aforementioned features will be described in later sections and validated by theoretical analysis and experimental results.

2 Proposed Fragile Watermarking Scheme

2.1 The Embedded Watermark Generation

This work intends to improve localization accuracy and security of the watermarking scheme as well as to provide as much information of the altered image as possible. To achieve these goals, certain processes must be taken into account in the watermark generation process. Here we select the low frequency wavelet coefficients from the 2-D one-level DWT to produce a low-frequency compressed image by a 4-bit non-uniform scalar quantization. Then the watermark is formed from the binary version of this compressed image. At the same time, we employ a scrambling encryption scheme to enhance the security of the watermarking algorithm. Details of the watermark generation process are described as follows.

Step 1: Perform a one-level DWT after setting the LSB of the original $m \times n$ image X to zero and extract the low-frequency coefficients denoted as LL.

The DWT has an advantage of achieving both spatial and frequency localization. In other words, each wavelet coefficient represents image content local in space and frequency [10]. Fig.1 shows the mapping between spatial block and DWT coefficients. Choosing the proper the wavelet basis such as DB1, a low-frequency coefficient in one-level DWT mostly depends on the spatial block size of 2×2. More specifically, a low-frequency coefficient LL_{ij} depends on the pixel values in the block $X((i-1)\times 2+1 : i\times 2\ ,\ (j-1)\times 2+1 : j\times 2)$, noted as $X_{ij}(2)$. Where $i = 1, \cdots, m/2$ and $j = 1, \cdots, n/2$.

Step 2: Apply a 4-bit non-uniform scalar quantization to LL using a secret key k_1 and obtain the image Q. That is

$$Q = f(LL, k_1). \tag{1}$$

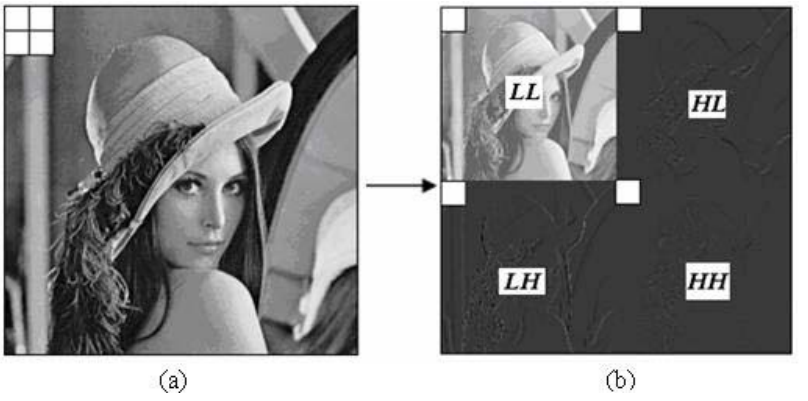


Fig. 1. The mapping between spatial block and DWT coefficients (a) "Lena" image, (b) the one-lever DWT of "Lena"

The (i, j) element of Q is defined as

$$Q_{ij} = \begin{cases} a, min + aq + \delta_a \leq LL_{ij} < min + (a+1)q + \delta_{a+1} \\ 15, LL_{ij} = max \end{cases} \tag{2}$$

Where min and max represent the minimum value and the maximum value in LL, $i = 1, \cdots, m/2$ and $j = 1, \cdots, n/2$. $q = \lceil (max - min)/16 \rceil$ is the uniform quantization step length, and $\lceil x \rceil$ denotes the smallest integer larger than or equal to x. The random sequence $\{\delta_a, a = 0, 1, \cdots, 15\}$is derived from k_1 and has value between $-q/4$ and $q/4$. Fig. 2 depicts such non-uniform scalar quantization. We name $Q = [Q_{ij}]$the LL-band scalar quantization matrix.

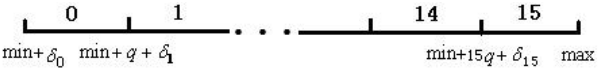


Fig. 2. Non-uniform scalar quantization scheme

Step 3: Given the secret key k, the LL-band scalar quantization matrix Q is scrambled encryption to produce the encrypting matrix E, depicted as:

$$E = P(Q, k) \tag{3}$$

Where, $P(.)$ means scramble encryption function.

Step 4: Convert each E_{ij} into a four-bit binary, i.e., $E_{ij} = [b_3 b_2 b_1 b_0]_2$ and form a binary matrix.

$$B_{ij} = \begin{bmatrix} b_3, b_2 \\ b_1, b_0 \end{bmatrix} \tag{4}$$

Step 5: By assembling B_{ij} together according to its position, we obtain the embedded watermark image $W = [B_{ij}]$ generated by the host image X;

Note thatB_{ij} is a binary block with size of 2×2. As a result, the embedded watermark image W is a same size with original image.

2.2 Watermark Embedding

The embedding procedure is similar to Wong's and other fragile watermarking techniques that exhibit invisibility and tamper localization [1-9]. We insert the generated embedded watermark image W into the LSB of the pixels in X. That is,

$$Y = \lfloor X/2 \rfloor \times 2 + W \tag{5}$$

Where, Y is the watermarked image.

Clearly, the proposed method, through scrambling, embeds the watermark into the LSB of the image pixels. In this manner the watermark derived from a block $X_{ij}(2)$ is not embedded into the same block; rather, it is randomly placed the LSB of other block. This introduces block-wise non-deterministic dependency among all blocks in the image. Therefore our method improves the robustness to thwart VQ attack and transplantation attack [3, 9]. In addition, our algorithm can achieve excellent tamper localization due to a block size of 2×2. It is worth mentioning that the strategy of scrambling encryption can also extend the new capacity to discriminate tampers on the content or watermark.

2.3 Authentication Algorithm

In the verification procedure, the watermark W' is first extracted from LSB of each pixel of the target image Y^*. And then the reconstructed encrypting matrix E' is computed using W' according to an inverse procedure to the step 4 in the watermark generation. The reconstructed LL-band scalar quantization matrix Q' is obtained using the correct key k,

$$Q' = P^{-1}(E', k) \tag{6}$$

Where $P^{-1}(.)$ is the inverse function of $P(.)$. Next according to the secret key k_1, we apply the steps 1 and 2 in watermark generation to compute the LL-band scalar quantization matrix Q^* from Y^*. By calculating the difference matrix,

$$\triangle Q = |Q^* - Q'| \tag{7}$$

Tamper localization and tamper discrimination can be achieved by viewing the difference matrix. Consider the case where the tested image was not altered, this implies $Q^* = Q$ and $Q' = Q$. Hence we have $\triangle Q = 0$. The proposed method, through scrambling, embeds the watermark into the LSB of the image pixels. As a result, the $\triangle Q$ displays randomly distributed isolated points if the watermark is altered. If $\triangle Q$ contains clustered regions, then the image contents in those regions are altered. According to the predefined threshold T, we can localize the area of alterations on image content with high probability. The approach is described as follows.

Let t_{ij} denotes the nonzero number in $N_8(\triangle Q_{ij})$ which is a set formed by eight adjacent pixels of $\triangle Q_{ij}$[11] and T is the predefined threshold (Details will be discussed in the following sub-section). Tamper discrimination and localization can be achieved by detecting each/every pixel in $\triangle Q$.

(1) If $\triangle Q = 0$, then corresponding to image block $Y_{ij}^*(2)$ would be considered as authentic;
(2) If $\triangle Q \neq 0$and $t_{ij} \geq T$, it would be considered as tamper on the content of $Y_{ij}^*(2)$.
(3) If $\triangle Q \neq 0$and $t_{ij} < T$, it would be thought of tamper on watermark, the $Y_{ij}^*(2)$ is genuine.

Obviously, according to the predefined threshold T, the proposed algorithm can not only distinguish alterations between the content and watermark, but also accurately localize the regions of alterations on image content. Consequently, the value of threshold T becomes a pivotal issue to be solved.

2.4 The Threshold T

In this sub-section, we will discuss how the threshold is selected from the theory of probability. Tamper discrimination aims at verifying the authenticity of the content of image block $Y_{ij}^*(2)$, whose corresponding $\triangle Q_{ij}$ unequal to zero. The nonzero pixels $\triangle Q_{ij}$ is resulted by tampering some watermarks or the content of the $Y_{ij}^*(2)$. Thus, tamper discrimination can be formulated as a binary hypothesis test as follows:

- H_0: the content of image block $Y_{ij}^*(2)$ is tampered, i.e.,$Y_{ij}^*(2)$ is not authentic.
- H_1: there are alterations on watermarks in the tested image, while the content of $Y_{ij}^*(2)$ is authentic.

In order to decide on the valid hypothesis, t_{ij} is compared with a suitably selected threshold T. For a given threshold T, the system performance can be measured in terms of the probability of false acceptance $P_{fa}(T)$ (i.e., the probability to consider it as authentic when the content of image block is tampered) and the probability of false rejection $P_{fr}(T)$ (i.e., the probability to reject it when the content of image block is authentic).

$$P_{fa}(T) = P\{t_{ij} < T|H_0\} \tag{8}$$

$$P_{fr}(T) = P\{t_{ij} \geq T|H_1\} \tag{9}$$

In the ideal case, a threshold T should exist such that both $P_{fa}(T)$ and $P_{fr}(T)$ are zero. In order to calculate $P_{fa}(T)$ and $P_{fr}(T)$, the fundamental theorem in probability and statistics can be used.

Theorem 1: if t obeys binomial distribution $t \sim B(n, p)$, n, p as the parameters, then the probability of t less than given T is:

$$P(t < T) = \sum_{i=0}^{T-1} C_n^i P^i (1-p)^{n-i} \tag{10}$$

Where C_n^i denotes the combination that i elements selected from n elements. For the proof of this theorem, the reader is referred to any book of probability theory [12].

• Under the hypothesis H_0: If the content of $Y_{ij}^*(2)$ is tampered randomly, the corresponding Q_{ij}^* is integer within [0, 15] with identical probability. That is, the probability of Q_{ij}^* to be unchanged is about one of sixteen parts in this case. Consequently the probability to detect the modification on the content of $Y_{ij}^*(2)$ is approximate 15/16. Suppose the image content of $N_8(\triangle Q_{ij})$ is tampered at random, it could be concluded as $t_{ij} \sim B(8, 15/16)$ from the theory of probability. According to Theorem 1, the probability of t_{ij} less than T:

$$P(t_{ij} < T|H_0) = \sum_{t=0}^{T-1} C_8^t (15/16)^t (1 - 15/16)^{8-t} \tag{11}$$

That is:

$$P_{fa}(T) = \sum_{t=0}^{T-1} C_8^t (15/16)^t (1 - 15/16)^{8-t} \tag{12}$$

• Under the hypothesis H_1: Suppose $\triangle w$ is the number of watermarks in tampered regions, we can obtain the reconstructed LL-band scalar quantization matrix $Q_\prime$ according to formula (6), where the altered bits obey a uniform distribution. Therefore, the nonzero probability to each pixel in $Q_\prime$ is same and equal to,

$$P_{\triangle w} = (\triangle w/4)/((m/2) \times (n/2)) = \triangle w/(m \times n) \tag{13}$$

In this condition, the probability that t_{ij} is less than given T is:

$$P(t_{ij} < T|H_1) = \sum_{t=0}^{T-1} C_8^t (P_{\triangle w})^t (1 - P_{\triangle w})^{8-t} \tag{14}$$

Therefore,

$$P_{fr}(T) = 1 - \sum_{t=0}^{T-1} C_8^t (P_{\triangle w})^t (1 - P_{\triangle w})^{8-t} \tag{15}$$

In watermarking algorithms for authentication, the goal of the attack is not make the authentication watermark unreadable, but to try to make the change undetectable [5], therefore the number of changed watermarks is not many, i.e. $P_{\triangle w}$ is small. According to the formulas (12) and (15), Fig.3 shows the curve of $P_{fa}(T)$ using a symbol "□" and six curves of $P_{fr}(T)$ with different values $P_{\triangle w}$. As can be seen from Fig.3, with increasing T, the $P_{fa}(T)$ increases but the $P_{fr}(T)$ reduces. For a given threshold T, the larger $P_{\triangle w}$ leads to the larger $P_{fr}(T)$.

For detecting the content tampers with high probability such as more than 99%, the threshold T is not more than $5 (P_{fa}(5) = 8.7 \times 10^{-4})$. Fig.4 shows the theoretical result of $P_{fr}(5)$ using a symbol "□" with different values $P_{\triangle w}$. Two experimental results of $P_{fr}(5)$, using the gray image of "Lena" and "woman"

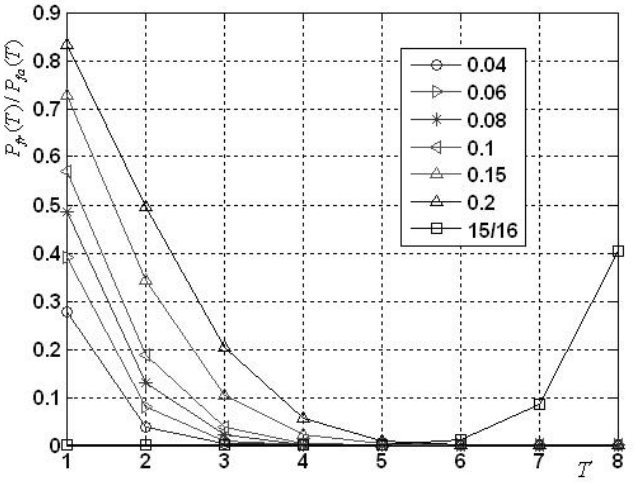


Fig. 3. One $P_{fa}(T)$ curve and six $P_{fr}(T)$ curves

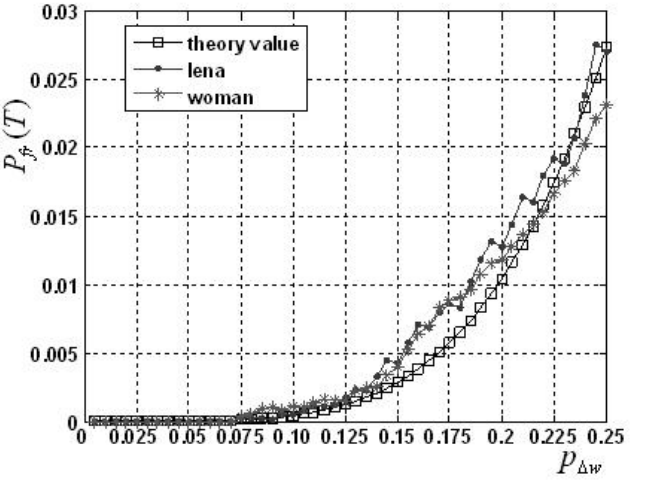


Fig. 4. $P_{fr}(T)$ results in comparison of theory and experimental with $T = 5$

with size of 256×256, are shown in Fig.4. As can be seen from them, the value of $P_{fr}(5)$ is almost zero when $P_{\triangle w} < 0.075$. Increasing the number of tampered watermarks to $P_{\triangle w} = 0.25$ caused the probability of false rejection $P_{fr}(T)$ to increase to nearly 0.03. These results indicate that our algorithm can discriminate tampers on image content or watermarks with high probability when the altered watermark information is less than 1/5.

3 Performance Analysis and Simulation Results

Now, we will demonstrate the effectiveness of the proposed approach with experimental results and discuss the performance of our algorithm. In simulation, the tested images are grayscale images with different size, the pixels values are within [0, 255].

We demonstrate the results of our method using the 240×320 image shown in Fig.5 (a), while the watermarked image is shown in Fig. 5(b). The power signal-to-noise rate (PSNR) between the watermarked image and the original one is 51.1023dB. Fig. 5(c) shows the difference matrix extracted from watermarked image. Clearly, all pixels equal to zero and the tested image would be considered as authentic.

Fig. 5. Original and watermarked images: (a) original image; (b) watermarked image; (c) difference matrix

3.1 Discrimination Tampers

To illustrate the effectiveness on the tamper discrimination property of the proposed method, several experiments were carried out using the watermarked image of Fig.5 (b). The types of tamper are the manipulation occurred on:

- Tamper 1: Both the image content and the watermark—- a fake plate number "$R237JAD$" is pasted on the watermarked image of Fig.5 (b). This tampered image is shown in Fig.6 (a).
- Tamper 2: The image content —- we replace the 7 most significant bits (MSBs) of each pixel in Fig.5 (b) with that in Fig.6 (a). The resulting image is depicted in Fig.6 (b).
- Tamper 3: The watermark—- we replace the LSB of each pixel in Fig.5 (b) with that in Fig.6 (a). That is, we alter the watermark of the test image. The tampered image is shown in Fig.6 (c).

Figs. 6(d), 6(e) and 6(f) are the difference images of three corresponding tampers, respectively. Tamper localization and tamper discrimination can be achieved by viewing the difference images. As can be seen from Fig. 6(a), altered plate is located. The isolated dots spread all over the image indicate that the embedded watermark was changed. On the other hand, as shown in Fig. 6(c),

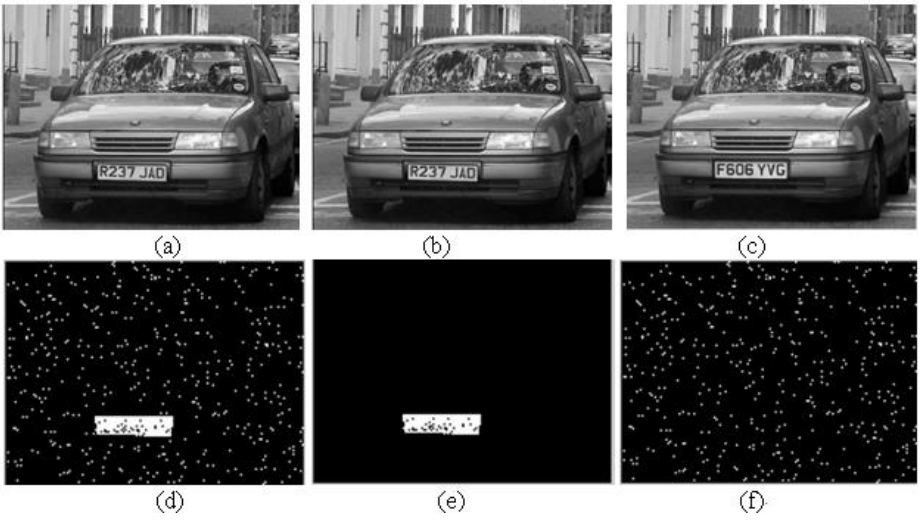


Fig. 6. Tampered images and corresponding difference matrixes (a), (b) and (c) are the tampered images of three corresponding tampers, respectively; (d), (e) and (f) are the difference matrixes of three corresponding tampers, respectively

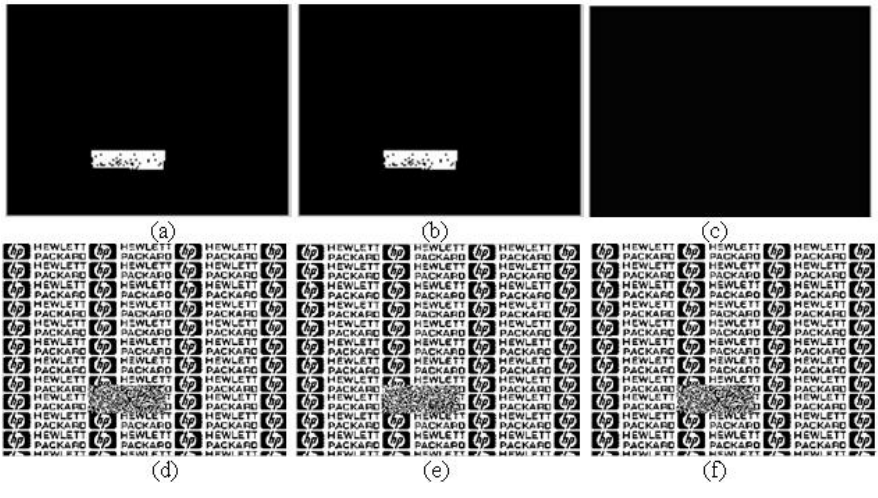


Fig. 7. Authentication results in comparison of tamper discrimination (a), (b) and (c) are authentication results by the proposed algorithm; (d), (e) and (f) are the results by Wong's algorithm [7]

only spread isolated dots appears. This implies that the manipulation on the image is only restricted to the watermark. Thus the image content is genuine.

According to the predefined threshold T =5, the authentication results of three tampers above are shown in Figs. 7(a), (b) and (c), respectively. In the (a)

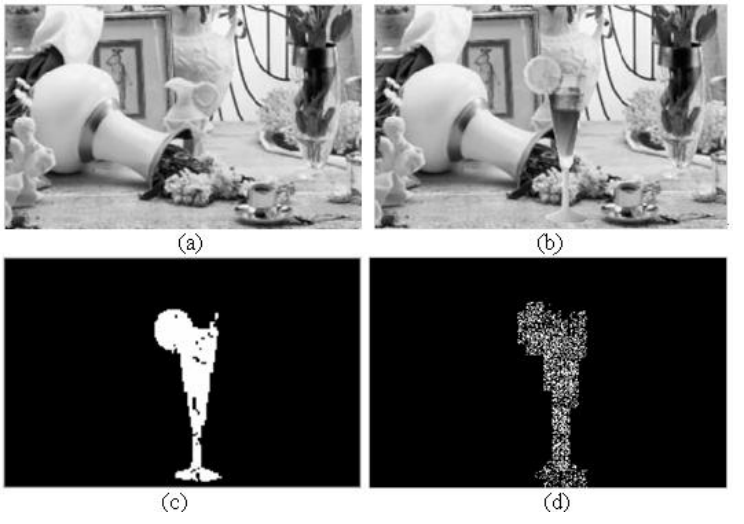


Fig. 8. Results in comparison of tamper localization accuracy (a) watermarked image; (b) tampered image; (c) authentication result by proposed method; (d) authentication result by Wong [7]

and (b), they exhibit the same altered regions and it suggests that the proposed method can localize the image content modification whether watermark in the tampered regions is altered or not. In (c), there were not non-zero pixels and it means that corresponding image content is genuine.

The Wong's and other LSB-modification schemes from the literature embed the watermark derived from a block into the LSB in the same block. Thus, no matter whether where the tampering is on the image contents or the watermark, the alternations shown in the decoded image are still confined in that block. This results in the indistinguishable detection results. Using the same test image and the same modification, Figs. 7(d), 7(e) and 7(f) show the results of three tampers using Wong's algorithm [7]. It can locate where the alteration of the image is. But we cannot tell what kind of manipulations being made on the marked image; and they might be declared fake, even though Fig. 6 (c) contains genuine digital contents.

3.2 Localization Accuracy

In this experiment, we test the localization accuracy of the proposed algorithm. In current block-wise schemes, the tamper localization accuracy is a block size of 8×8; whereas our algorithm can achieve an accuracy of 2×2. Fig.7 (a) is a watermarked image generated by the proposed algorithm. Using "Photoshop", a cup is placed in (a) and the tampered image is shown in Fig.7 (b). With the predefined threshold T=5, the authentication result by our algorithm is shown in Fig.7 (c). Using the same method, Fig. 7 (d) shows the authentication result by Wong [7]. The above results demonstrate that our authentication algorithm is more accurately than the exiting block-wise schemes such as Wong's.

4 Conclusion

A wavelet-based fragile watermarking algorithm for secure image authentication has been presented. In proposed algorithm, the embedded watermark is generated using the discrete wavelet transform (DWT), and then the improved security watermark scrambled by chaotic systems is embedded into the LSB of the image data. This results in much improved protection of the watermarking system. Simulation results have been given to demonstrate that the proposed method exhibits excellent tamper localization and discrimination properties. To discuss on future work in conclusion.The further work is dedicated to develop a secure fragile watermarking scheme with tamper recovery.

Acknowledgments. This work is partially supported by the Program for New Century Excellent Talents in University of China (NCET-05-0794), the Sichuan Youth Science and Technology Foundation (03ZQ026-033 and 51430804QT2201), and Application Basic Foundation of Sichun Province, China (2006 J13-10).

References

1. M.M. Yeung and F. Mintzer, An invisible watermarking technique for image verification, Proc. IEEE Int. Conf. Image Processing, 1997, vol. 2, pp. 680-683.
2. P. Wong, A public key watermark for image verification and authentication, Proc. IEEE Int. Conf. Image Processing, Chicago, IL, 1998, pp. 425 - 429.
3. Barreto, P., Kim, H., Rijmen, V., Toward secure public-key block-wise fragile authentication watermarking, IEE Proceedings-Vision, Image and Signal Processing, 2002.2 (149), pp: 57-62.
4. M.Holliman and N. Memon, Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes, IEEE Trans. Image Processing, vol. 9, no. 3, pp. 432-441, March 2000.
5. J. Fridrich, Security of fragile authentication watermarks with localization, Proc. SPIE, vol. 4675, Security and Watermarking of Multimedia Contents, San Jose, CA, Jan., 2002, pp. 691-700.
6. F.Deguillaume, S.Voloshynovskiy, T.Pun, Secure hybrid robust watermarking resistant against tampering and copy attack, Signal Processing 83(2003):2133-2170.
7. P. Wong and N. Memon, Secret and public key image watermarking schemes for image authentication and ownership verification, IEEE Trans. Image Processing, vol. 10, pp. 1593-1601, 2001.
8. Suthaharan, S., Fragile image watermarking using a gradient image for improved localization and security, Pattern Recognition Letters, 2004(25),pp:1893-1903.
9. Yinyin Yuan, Chang-Tsun Li., Fragile Watermarking Scheme Exploiting Non-deterministic Block-wise Dependency, In proceeding of 17th International Conference on Pattern Recognition (ICPR'04).
10. Justin K R, Hyeokho Choi ,et al, Bayesian Tree-structured image modeling using wavelet-domain hidden markov models, IEEE Transactions on Image Processing, 2001, 10(7):1056 - 1068.
11. Rsfael C.Gonzalez, Richard E.Woods, Digital image processing (second edition), Publishing House of Electronics Industry (in Chinese), BEIJING,2003.3
12. Li Yuqi, The Theory of Probability and Statistics, Beijing: National Defense Industry Press, (in Chinese), 2001.

Secure Watermark Embedding Through Partial Encryption

Aweke Lemma, Stefan Katzenbeisser, Mehmet Celik, and Michiel van der Veen

Philips Research Europe
High Tech Campus 34
NL-5656 AE Eindhoven, The Netherlands
{aweke.lemma, stefan.katzenbeisser, mehmet.celik,
michiel.van.der.veen}@philips.com

Abstract. Secure watermark embedding allows to securely embed a watermark into a piece of content at an untrusted user device without compromising the security of the watermark key, the watermark or the original. In this paper, we show how secure embedding can be achieved by using traditional watermarking schemes in conjunction with partial encryption techniques, which were primarily developed to facilitate fast encryption of media content. Based on this concept, we develop two new efficient secure embedding mechanisms, one for the MASK watermarking scheme operating on baseband audio and one for a spread spectrum watermarking scheme operating on MPEG-2 encoded video streams.

1 Introduction

In the past few years we have experienced a clear shift from classic content distribution channels, such as CDs or DVDs, towards electronic content distribution (ECD). Even though electronic distribution offers new business possibilities for content providers, the risk of un-authorized mass re-distribution largely limited the widespread adoption of digital distribution channels. Digital Rights Management (DRM) systems try to minimize the risk of copyright infringements by using cryptographic techniques to securely distribute content to client devices and enforce proper usage. Encryption, however, can only offer a partial solution to the problem of unauthorized distribution. Eventually, the content has to be decrypted and presented to the user in (analog) clear-text form, from which copies can easily be made and re-distributed.

Forensic tracking watermarks [13]—which may be used in place of or in conjunction with traditional DRM/encryption methods—allow to enforce usage rights beyond the digital domain. In a forensic tracking system, each copy of the distributed content is watermarked with a unique transaction tag, which links that copy either to a particular user or to a specific device. When an unauthorized copy is found, the embedded watermark (carrying the transaction tag) uniquely identifies the source of the copy, and allows to trace the user who has redistributed the content. Even though forensic tracking in itself does not prevent unauthorized re-distribution, the risk of being caught acts as a strong deterrent.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 433–445, 2006.

In current forensic tracking systems, forensic watermarks are embedded into the content directly by a trusted distribution server before the content is released onto a distribution network. This model, however, severely limits the applicability of forensic watermarks in forthcoming content distribution models:

- Integrating forensic tracking watermarks into large-scale ECD systems brings challenges with regard to security, system complexity, and bandwidth usage. As the ECD server needs to embed a unique watermark into each copy of the content, both the server load and the bandwidth requirements for content transmission scale linearly with the number of users. In large-scale content distribution applications, the watermark embedder at the server side turns out to be a major performance bottleneck. In addition, as the content is personalized for each user, distribution requires a point-to-point channel between the ECD server and the client, prohibiting the use of broadcasting, multicasting and caching, which significantly reduce the bandwidth usage for content transmission.
- In addition to the above-mentioned performance problems, server-side watermark embedding is unsuitable in forthcoming content distribution systems which employ a clear separation between content providers and license brokers. For example, in the OMA DRM model [2], content is allowed to float in a network freely in encrypted form. Once a party wishes to access the content, it purchases a license from a clearance center and obtains a decryption key. Due to the absence of a central distribution server, server-side watermark embedding is not applicable in this scenario.

These limitations could be circumvented if the untrusted client devices themselves perform watermark embedding. The major obstacle to be solved is that watermark embedders require knowledge of a secret watermarking key, which, once exposed to an attacker, allows to effectively remove watermarks. Thus, watermark embedding at the client must be done in a way which does not compromise the security of the keys; in addition, neither the watermark nor the original content should be available for the client. In the sequel, we will call client-side watermark embedding systems achieving these security properties *secure watermark embedding*. The use of secure client-side embedding can overcome both above mentioned limitations: it shifts the computational burden of watermark embedding to the client, allows to use broadcasting techniques to distribute encrypted content, and facilitates distribution models where no central server is involved in the actual purchase phase.

Secure watermark embedding transmits to the client an encrypted version of the original content together with some helper data, which implicitly encodes the watermark to be embedded. The client can use this personalized helper data to decrypt a watermarked version of the content that was sent to him. Still, the client cannot extract the watermark out of the helper data or obtain the original content in the clear.

In this paper, we show how secure watermark embedding can be realized by utilizing concepts of partial encryption [12], which have primarily been developed in the past in order to speed up the encryption process of media files by selectively

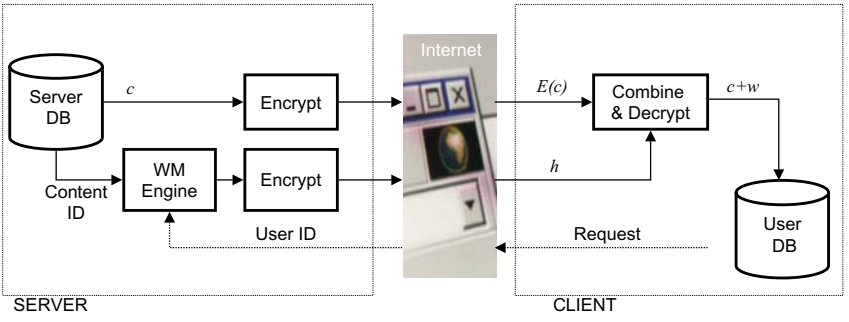


Fig. 1. Electronic Content Distribution utilizing secure watermark embedding

encrypting only the perceptually most relevant parts. To use partial encryption for secure watermark embedding, we encrypt the perceptually most relevant parts of a piece of content and give the client helper data which allows him to decrypt the content in a slightly different way; the differences induced by the changed decryption process represent the watermark. In this paper, we show how this general methodology can be applied to baseband audio and MPEG-2 compressed video streams.

The rest of the paper is organized as follows. In Section 2 we discuss in greater detail the concept of and the requirements for practical secure watermark embedding; Section 3 reviews existing client-side watermark embedding solutions with regard to the requirements. In Section 4 we outline our general methodology for secure embedding, while Sections 5 and 6 discuss two concrete implementations of the methodology for baseband audio and MPEG-2 encoded video streams. Finally, Section 7 concludes the work.

2 Secure Client-Side Watermark Embedding

Figure 1 illustrates the concept of secure watermark embedding in the context of electronic content distribution in greater detail. When a client wants to retrieve a piece of content c, he contacts a distribution server, who ships an encrypted version $E(c)$. At a later state, some party (not necessarily the same server) generates a watermark representing the identity of the user and computes helper information h, implicitly coding the personalized watermark. This helper information is subsequently shipped to the client, who can use h to decrypt a copy of the content which is watermarked by w (denoted by $c+w$ in the figure); however, the helper information h does not allow him to infer either c or the watermark directly.

We can identify the following requirements for practical secure watermark embedding techniques:

– *Low bandwidth overhead.* The transmission overhead induced by the secure watermark embedding mechanism should be as small as possible. In particular:

  - The employed encryption algorithm should operate in a space efficient manner, i.e., the size of $E(c)$ should be similar to the size of c. This is especially relevant as content is usually transmitted in (lossy) compressed form. The chosen encryption algorithm E should thus ideally operate directly on compressed content.
  - The bandwidth required for the transmission of the helper data h should be considerably smaller than the one required for transmitting $E(c)$.
- *Security.* Transmitting $E(c)$ and the helper data h must not compromise the security of either c or w. In particular, h must not reveal to the client more information about the original and the watermark than it is already leaked by the watermarked work itself.
- *Content independence.* Ideally, h should be independent of the content c. This enables the use of secure watermark embedding in flexible distribution models that split the content distribution from the license acquisition process. Furthermore, it allows to pre-compute helper data for a particular set of clients (which may allow to implement live video broadcasting solutions in which the computationally intensive process of helper data generation can be done offline).

3 Related Work

Secure watermark embedding has only recently gained attention in the scientific community. With current technology, client-side watermark embedding is typically performed in a dedicated piece of hardware within consumer electronic devices (see [11,10] for a framework). However, this solution has the apparent drawback that it requires a dedicated hardware installed base, cannot be easily integrated in legacy applications and is not easily updatable. Thus software solutions are clearly preferable.

In broadcast environments, Crowcroft et al. [4] and Parviainen et al. [9] proposed a client-side watermark insertion technique based on stream switching. In their method, they chop the content stream into small chunks and broadcast two version of the stream, watermarked with different watermarks. Each chunk is encrypted by a different key. Clients are given a different set of decryption keys that allow them to selectively decrypt chunks of the two broadcast streams such that each client obtains the full stream. The way the full stream is composed out of the two broadcast versions encodes the watermark. However, this solution does not meet the bandwidth requirements stated above, as the amount of data needed to be broadcast to the clients is twice as large as the content itself.

Emmanuel et. al. [5] proposed a client-side embedding method in which a pseudorandom mask is blended over each frame of a video; each client is given a different mask, which, when subtracted from the masked broadcast video, leaves an additive watermark in the content. The scheme has security problems, as a constant mask is used for all frames of a video, which can be estimated by averaging attacks. Subsequently the estimated mask can be subtracted from the encrypted video in order to obtain a perceptually acceptable and watermark-free version of the content.

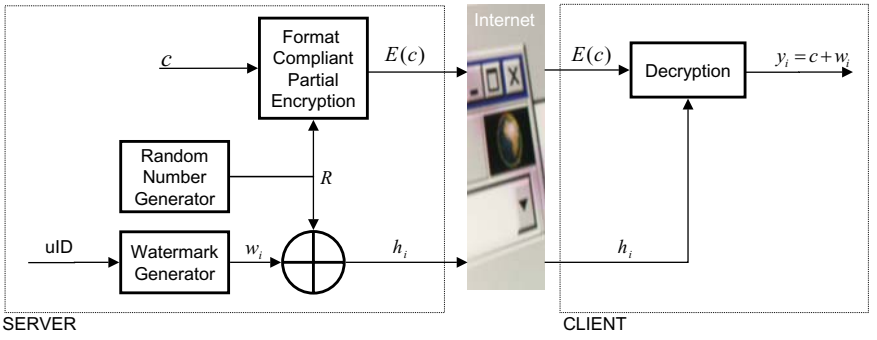


Fig. 2. Secure watermark embedding using partial encryption

Anderson et. al. [3] designed a special stream cipher, called Chameleon, which allows, by appropriate design of encryption keys, to decrypt Chameleon-encrypted content in slightly different ways. Thus, the special design of the cipher allows to leave a key-dependent trace in the decrypted data stream. Kundur and Karthik [6] were the first to use techniques from partial encryption together with Chameleon in order to fingerprint digital images. Their method is based on encrypting the signs of DCT coefficients in an image; during decryption some signs are left unchanged, which leaves a detectable fingerprint in the image. As the sign bits of DCT coefficients are perceptually significant, the partially encrypted version of the content is heavily distorted. However, as some DCT coefficients are left scrambled during decryption, the watermark can be visible; visibility of the watermark must be traded in for optimal detection.

Recent work by Adelsbach et. al. [1] showed how to generalize the Chameleon cipher in order to be able to embed spread spectrum watermarks. However, the work still only considers uncompressed baseband signals.

4 Secure Embedding Through Partial Encryption

In this section, we show how secure watermark embedding can be realized through partial encryption. As mentioned above, we choose a partial encryption scheme and encrypt perceptually important parts of the content, while preserving the content file format. Finally, we provide the client with helper data, which allows him to access a personalized, slightly modified version of the content. The remaining unique signature (difference between the original and the reconstructed version) can later be used as a forensic watermark to trace back the origin of the content. The concept is schematically depicted in Figure 2.

Note that in our approach we only perform partial encryption of the content c (for example, as opposed to [1]). Typically, in DRM applications partial encryption of the content is sufficient, as the content itself is not confidential (it can be accessed by every legitimate user). For the security analysis of a forensic tracking watermarking architecture one has to assume that an attacker posesses

at least the same information as a legitimate user. Thus, the applied encryption scheme only needs to protect those parts of the content that potentially help an attacker to derive an un-watermarked copy. In addition, partial encryption has the advantage that the encrypted files can be viewed or listened on a normal playback device. Even though the content is severely distorted, the user gets a first impression on how the decrypted content will look like. Thus, the partially encrypted content can serve as a low-quality preview.

In greater detail, the proposed system works as follows:

- **Server:** The server performs the following operations:
  1. The server reads an input content c,
  2. chooses perceptually significant features of c,
  3. and encrypts those features using a format compliant partial encryption scheme; this process yields to a perceptually unacceptable distorted content $E(c)$, which can be safely released into the public. The features are chosen in such a way that it is hard to reconstruct, using techniques of signal processing, a perceptually acceptable estimate of c out of the encryption $E(c)$.
  4. For each user i, the server generates a watermark w_i and chooses helper information h_i, which can be applied to $E(c)$ in order to undo the distortions of the encryption process and to leave a detectable watermark w_i. The helper information h_i is constructed in such a way that knowledge of h_i does not allow the client to infer the watermark. In addition, knowledge of h_i does not facilitate obtaining an un-watermarked copy of the content.
  5. Finally, the server sends h_i to the client.
- **Client:** The client performs the following operations:
  1. The client acquires the content $E(c)$ from the public domain and
  2. receives the helper information h_i from the server via a one-to-one link.
  3. Finally, the client applies h_i to the distorted content $E(c)$ in order to obtain his personalized copy of the content y_i. This process produces a perceptually acceptable, but watermarked output signal, $y_i = h_i(E(c)) = c + w_i$.

In the following sections, we show how this general concept can be applied to baseband audio and MPEG-2 encoded video streams by discussing two proof-of-concept implementations.

5 Baseband Audio

In this section, we show how the MASK [7] audio watermark embedder can be implemented safely at an untrusted client device. To facilitate our discussion, we first present a brief summary of the MASK watermarking system, and subsequently show how this system is implemented in the context of secure watermark embedding.

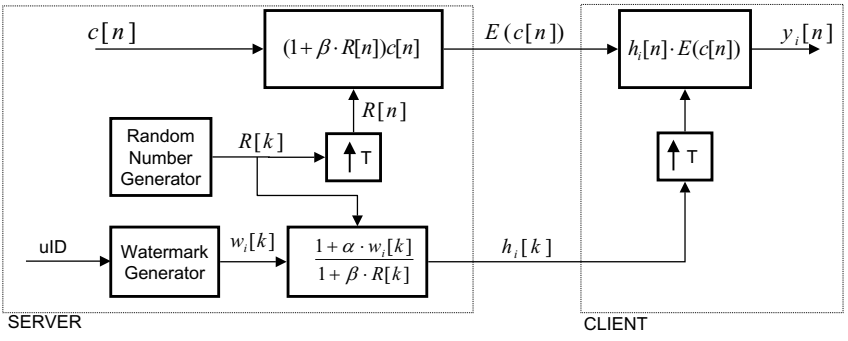


Fig. 3. MASK watermarking system based secure watermarking scheme

The MASK Watermarking System

In MASK, a watermark is embedded by modifying the envelope of the host signal. More specifically, given the host signal $c[n]$ and the watermark signal $w_i[n]$, the watermarked content $y_i[n]$ is given by

$$y_i[n] = c[n] + \alpha[n]w_i[n]c[n], \tag{1}$$

where the watermark signal $w_i[n]$ is chosen in such a way that it predominantly modifies the short time envelope of the signal, and the gain function $\alpha[n]$ is controlled by a psychoacoustic model of the human auditory system. The MASK system has been extensively tested and has proven to combine good audibility quality with high robustness. For more details on the implementation and on the robustness tests, we refer to [7].

Joint Decryption and Watermarking

Figure 3 shows the secure embedding framework for MASK. Encryption of the original content is achieved by modulating the host signal with a piece wise stationary random sequence $R[k]$ such that the resulting audio is perceptually annoying to listen to. Let T be the interval (frame) over which $R[k]$ remains constant and let $c_k[n]$, $0 \leq n \leq T-1$, represent the k-th frame of the content signal. We encrypt the k-th frame by

$$E(c_k[n]) = (1 + \beta[k]R[k])c_k[n], \tag{2}$$

where the weighting coefficient $\beta[k]$ is chosen in such a way that the condition $1 + \beta[k]R[k] \neq 0$ is always satisfied.

For one client i, the server first generates the MASK watermark signal $w_i[k]$ that is linked to the identity of the client (for the process of payload encoding we refer to [7]). The watermark signal $w_i[k]$ is made to vary gracefully in order to minimize audible artifacts in the watermarked content. The typical waveform of $w_i[k]$ is shown in the upper part of Figure 4. Finally the server computes a helper signal h_i for user i, which is given by

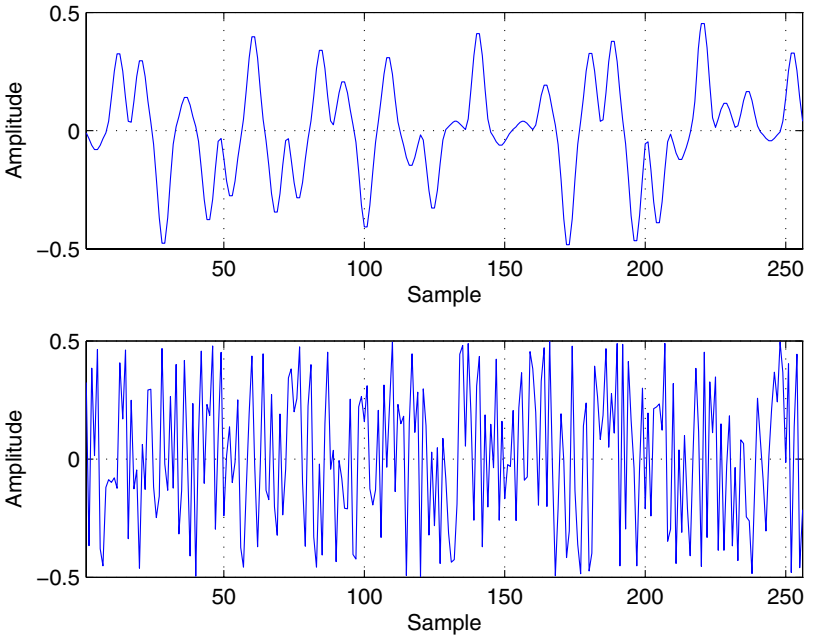


Fig. 4. Typical wave shapes of $w_i[k]$ (top) and $R[k]$ (bottom)

$$h_i[k] = \frac{1 + \alpha[k]w_i[k]}{1 + \beta[k]R[k]}, \tag{3}$$

and distributes this signal to client i.

On the client side, joint decryption and watermarking is achieved by taking the product between the helper data $h_i[k]$ and the encrypted frame content $E(c_k)$. More specifically, for each frame k, the client computes the watermarked frame signal $y_{i,k}[n]$ by

$$y_{i,k}[n] = h_i[k]E(c_k[n]). \tag{4}$$

Substituting the values of $h_i[k]$ and $E(c_k[n])$ from (3) and (2), respectively, we obtain

$$y_{i,k}[n] = (1 + \alpha[n]w_i[k])c_k[n]. \tag{5}$$

From the last equation we see that the client is left with a MASK-watermarked version of the content. The MASK watermarking system is extensively studied in different papers [8,7,14] and has been shown to combine excellent audibility/robustness tradeoff. Thus, in this paper, we do not consider such details, interested readers are advised to visit the above references.

Effect of the Spreading Factor on Robustness and Security

Note that in the above, we have assumed that the random number $R[k]$ remains constant for a period of T samples. If we let q represent the number of audio

channels, this means that a single random number is provided for every $T \times q$ audio samples. This in turn implies that the size overhead introduced by the helper data is linearly related to the "spreading" factor T. In MASK system (crf. [7]), T represents the so-called watermark symbol period. It reflects the granularity of the watermark symbol repetition. If the audio clip is long-enough the symbol period does not affect the robustness significantly because the total number of samples per a single watermark symbol remains unchanged. To be more specific, let T_1 and T_2 be two spreading factors, L_w be the length of the watermark sequence and $L_s \gg L_w \times \max(T_2, T_1)$ be the length of the audio clip. Then, in the audio, the watermark sequence will be repeated $r_1 = L_s/(L_w * T_1)$ times for the case of T_1 and $r_2 = L_s/(L_w * T_2)$ times for the case of T_2. The repetition of each watermark symbol is given by $T_1 \times r_1$ for the first case and by $T_2 \times r_2$ for the second case. After substituting the values of r_1 and r_2, both of the above products simplify to L_s/L_w. This shows that if L_s is large enough, the level of averaging used to extract each symbol is independent of the spreading factor and thus robustness is not significantly affected. However, the spreading factor T introduces tradeoff between security and size overhead. That is, repeating $R[k]$ over several samples leaks information. We defer the security analysis for a future work.

Experimental Results

We have tested the system depicted in Figure 3 using different stereo audio streams sampled at 44.1 kHz. For the test, we have chosen $T = 64$ samples, $\beta[k] = \beta = 0.9$ and $\alpha[k] = \alpha = 0.15$. The encrypted audio $E(c)$, though still recognizable, is graded as extremely annoying to listen to, whereas the watermarked output signal y_i is perceptually indistinguishable from the original one. In the implementation, the helper data was coded in 8 bits float, thus for the transmission of the helper data a side channel with capacity of at least

$$C_{CH} = \frac{8 * 44100}{T} \text{ bps}$$

is required. For $T = 64$, this equals to 5.5 kbps. Compared to a bitrate of a typical compressed audio stream (about 128 kbps), this amounts to an overhead of approximately 6%.

6 MPEG-2 Compressed Video

In this section, we show how the general methodology of joint watermarking and decryption can be applied to MPEG-2 compressed streams. Again, we first describe the employed watermarking scheme and subsequently detail how it is used in conjunction with a partial encryption scheme.

Watermarking Scheme

We use an additive spread spectrum watermark which modulates the luminance DC values of all I-frames present in the MPEG-2 stream. Recall that in MPEG-2,

each frame is divided into $N \times M$ macroblocks, each having 16×16 pixels; a macroblock is further subdivided into four 8×8 luminance blocks. Let $c_k[x, y]$, $1 \leq x \leq 2N$ and $1 \leq y \leq 2M$, denote the luminance DC values of all image blocks of the k-th I-frame. As a carrier for the watermark, a pseudorandom bit pattern of size $N \times M$, where each value is either $+1$ or -1, is created. To encode a payload, the pattern is shifted circularly both in the horizontal and the vertical direction to obtain a watermark w_i of size $N \times M$. From w_i, we obtain a $2N \times 2M$ matrix w_i' by

$$w_i' = w_i \otimes \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},$$

where $\otimes$ denotes the Kronecker product. The watermark w_i' is used to modulate the luminance values $c_k[x, y]$ to obtain the watermarked content

$$y_k[x, y] = c_k[x, y] + \alpha w_i'[x, y],$$

where α controls the watermark embedding strength. This embedding method has the effect that the upper two DC values in a macroblock will be modulated with the watermark, whereas the lower two values are left unchanged (and will be used in the detection process to minimize the influence of the host signal on the watermark detection result).

For watermark detection, the stream is decompressed and a constant number of consecutive frames is averaged; a blockwise DCT transform is applied to this averaged frame. In each macroblock, the upper two (watermarked) luminance DC coefficients are added, from which the lower two (unchanged) coefficients are subtracted. This way, the averaged frame is condensed to an $N \times M$ matrix, which is finally correlated with circular shifts of the watermark pattern w_i. If sufficient correlation exists, the watermark is assumed to be present; the shift with which the highest correlation has been achieved codes the payload. Note that for simplicity of explanation, we have used a constant watermark for all I-frames. However, the system can be easily changed to support embedding of different watermarks in subsequent I-frames.

Joint Decryption and Watermarking

To encrypt an MPEG-2 stream, we produce for each I-frame a random $2N \times 2M$ matrix $r_{i,k}$ with entries in the range of $(-2^l, 2^l)$ and add its elements to the luminance DC coefficients

$$E(c_k[x, y]) = c_k[x, y] + r_{i,k}[x, y].$$

Depending on the value of l, this results in more or less severe visible artifacts in the stream; the visual effect of this partial encryption method is illustrated in Figure 5. Part (a) of the figure shows a frame of the video, while (b) illustrates the effect of the chosen partial encryption: due to the noise in DC values, severe blocking artifacts are introduced.

For secure watermark embedding, the client is given the encrypted version of the stream as well as (for each I-frame) the $2N \times 2M$ matrix $h_{i,k} = r_{i,k} - \alpha w_i'$

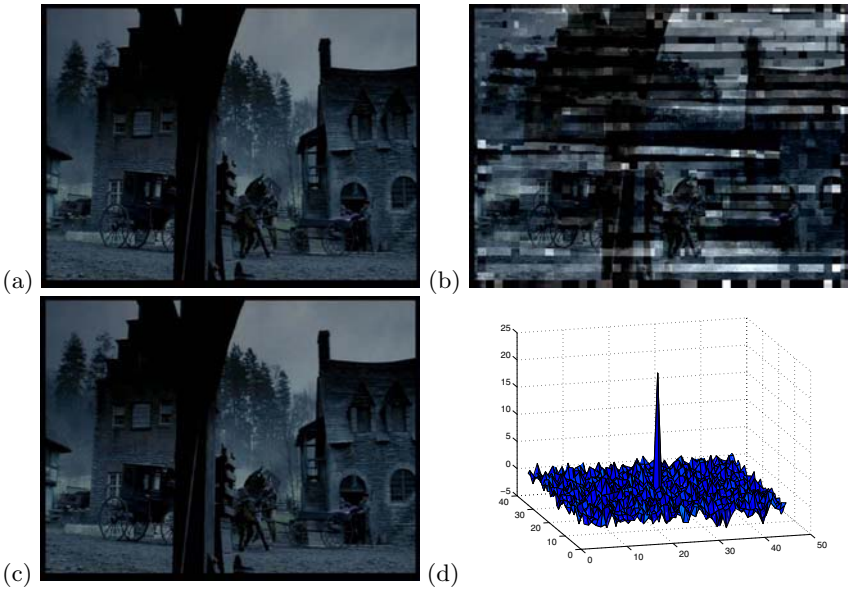


Fig. 5. Illustration of the proposed combined watermarking and decryption system: (a) an original frame of a MPEG-2 compressed movie, (b) the corresponding encrypted frame, (c) the reconstructed watermarked frame and (d) the watermark detection result

as helper information, which he subtracts from the DC luminance coefficients to obtain the watermarked content:

$$\begin{aligned} y_k[x,y] &= E(c_k[x,y]) - h_{i,k}[x,y] \\ &= c_k[x,y] + \alpha w_i'[x,y]. \end{aligned} \tag{6}$$

Figure 5(c) shows that using equation (6), the visual artifacts can be completely removed in the joint decryption and watermarking step. Still, the watermark can be reliably detected by a correlation detector, see part (d) of the figure.

Experimental Results

We have tested the system on several MPEG-2 compressed movies; results for four different clips are summarized in Table 1. First, we can note that embedding the watermark only marginally increases the size of the compressed content (about 0.05%). The encryption step has a noticeably effect on the size of the content, as it is adding uniformly distributed noise. Depending on the strength of the noise (i.e., the value l) we can observe an increase in the content size of about $0.5 - 0.8\%$. The size of the helper data which needs to be sent to the client in addition to the content scales linearly with the content size: for each luminance DC value of the content, one l-bit value needs to be transmitted. For $l = 3$, this amounts to a helper data size of about 1.1% of the content, whereas for $l = 3.5$, we obtain an overhead of about 1.3%.

Table 1. Performance of the combined watermarking and decryption system

clip	original size (bytes)	watermark overhead	overhead for $l = 3$		overhead for $l = 3.5$	
			encryption	helper data	encryption	helper data
A	13,561,344	0.05%	0.53%	1.11%	0.77%	1.30%
B	14,998,551	0.03%	0.45%	1.10%	0.70%	1.29%
C	12,808,526	0.03%	0.48%	1.10%	0.73%	1.28%
D	15,007,249	0.02%	0.25%	1.12%	0.45%	1.30%

7 Conclusions and Future Work

In this paper, we considered secure watermark embedding algorithms, which allow to securely insert a watermark at an untrusted client device without compromising the security of the watermark key, the watermark or the original content. To implement the functionality, we perform a partial encryption of the content and give the client helper information, which allows to decrypt a slightly different version of the content; the differences between the original and the reconstructed version constitute a forensic watermark. In particular, we discussed two proof-of-concept implementations, one for the MASK watermarking scheme operating on baseband audio and one for a simple additive spread spectrum watermark operating on MPEG-2 compressed video streams. We showed that partial encryption can overcome the major current obstacle of secure watermark embedding, namely limit the size of the helper data needed to be transmitted between the server and the client. In the current paper, we have mainly concentrated on efficiency aspects of secure watermark embedding and have not thoroughly addressed security issues of the employed partial encryption (i.e., the exact relation between the difficulty of a successful cryptanalysis and the complexity of watermark removal). We leave this, as well as the investigation of different partial encryption methods, for future work.

References

1. A. Adelsbach, U. Huber, and A.-R. Sadeghi. Fingercasting—joint fingerprinting and decryption of broadcast messages. In *11th Australasian Conference on Information Security and Privacy*, 2006.
2. Open Mobile Allowance. OMA digital rights management. `http://www.openmobilealliance.org`.
3. R. J. Anderson and C. Manifavas. Chameleon—a new kind of stream cipher. In *FSE '97: Proc. of the 4th Int. Workshop on Fast Software Encryption*, pages 107–113, London, UK, 1997. Springer-Verlag.
4. J. Crowcroft, C. Perkins, and I. Brown. A method and apparatus for generating multiple watermarked copies of an information signal. WO Patent No. 00/56059, 2000.
5. S. Emmanuel and M.S. Kankanhalli. Copyright protection for MPEG-2 compressed broadcast video. In *ICME 2001. IEEE Int. Conf. on Multimedia and Expo.*, pages 206–209, 2001.

6. D. Kundur. Video fingerprinting and encryption principles for digital rights management. *Proceedings of the IEEE*, 92(6):918–932, 2004.
7. A.N. Lemma, J. Aprea, W. Oomen, and L. van de Kerkhof. A temporal domain audio watermarking technique. *IEEE Transactions on Signal Processing*, 51(4):1088–1097, 2003.
8. Aweke Negash Lemma, Javier Aprea, Werner Oomen, and Leon v.d. Kerkhof. A robustness and audibility analysis of a temporal envelope modulating audio watermark. In *IEEE DSP/SPE workshop proceedings*, Gallaway Gardans, GA, USA, October 13-16 2002.
9. R. Parviainen and P. Parnes. Large scale distributed watermarking of multicast media through encryption. In *Proceedings of the International Federation for Information Processing, Communications and Multimedia Security Joint working conference IFIP TC6 and TC11*, pages 149–158, 2001.
10. P. Tomsich and S. Katzenbeisser. Copyright protection protocols for multimedia distribution based on trusted hardware. In *Protocols for Multimedia Systems (PROMS 2000)*, pages 249–256, 2000.
11. P. Tomsich and S. Katzenbeisser. Towards a robust and de-centralized digital watermarking infrastructure for the protection of intellectual property. In *Electronic Commerce and Web Technologies, Proceedings (ECWEB 2000)*, volume 1875 of *Springer Lecture Notes in Computer Science*, pages 38–47, 2000.
12. A. Uhl and A. Pommer. *Image and Video Encryption, From Digital Rights Management to Secured Personal Communication*. Springer, 2005.
13. M. van der Veen, A. Lemma, and A.A.C. Kalker. Electronic content delivery and forensic tracking. *Multimedia Systems*, 11(2):174–184, 2005.
14. Michiel van der Veen, Aweke Lemma, and Ton Kalker. Watermarking and fingerprinting for electronic music delivery. In *SPIE Workshop 2004*, San Jose, CA, USA, 2004.

A Rotation-Invariant Secure Image Watermarking Algorithm Incorporating Steerable Pyramid Transform

Jiangqun Ni*, Rongyue Zhang, Jiwu Huang, Chuntao Wang, and Quanbo Li

Department of Electronic and Communication Engineering, Sun Yat-Sen University
Guangzhou 510275, P.R. China
Phn: 86-20-84036167
issjqni@mail.sysu.edu.cn

Abstract. Robustness and security are the key issues in the development of image watermarking algorithm. A new rotation invariant security image watermarking algorithm based on steerable pyramid transform is proposed in this paper. The algorithm is characterized as follows: (1) the rotation invariance and robust watermarking are achieved concurrently on the same transform domain; (2) the rotation synchronization is obtained through template matching by using steerable pyramid transform, which satisfies the shiftability in orientation condition; (3) the watermarks are embedded into an oriented subband at angle θ, which can be interpolated with base filter kernels and used as a key in watermark detection to increase the security of watermark; (4) the watermark detector is designed based on the steerable vector HMM model. High robustness is observed against StirMark attacks and their joint attacks.

1 Introduction

With the popularity of Internet, the copyright protection, authentication and tamper proofing of digital media are becoming increasingly important. And thus digital watermarking, especially for image and video, has become the domain of extensive research. The DWT is playing an increasingly important role in the development of watermarking algorithm [1][2], due to its good spatial-frequency characteristics and its wide applications in image/video coding standards. One of the drawbacks of standard wavelet transforms is in that they are sensitive to the orientation of the input image. If the image is rotated, then in the wavelet domain, the wavelet coefficients change completely. Actually, the wavelet coefficients of the rotated image are not just be simply rotated, but are also modified. One way to remedy this situation is to replace the standard wavelet decomposition with the steerable pyramid transform proposed by Simoncelli and Freeman [3][4]. The steerable pyramid is a linear multiscale, multi-orientation image decomposition where the basis functions are directional derivative operators. One can convolve the image with a range of oriented filter kernels tuned to cover all orientations of interests in the image, where the oriented filter kernels can be interpolated with a fixed set of basis kennels to avoid high computational cost[4].

* Corresponding author.

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 446 – 460, 2006.

According to J.Cox[2], there exists two fundamental attributes for watermarking system, i.e., robustness and security. Robustness means the resistance against common signal distortions, such as geometrical distortions and other signal processing operations. While the security means the resistance against malicious and intentional modification of the watermark signal.

For robustness of watermarking, one of the major challenges is to increase its performance to resist against geometrical attacks such as rotation, scaling, translation, cropping and shearing. These geometrical distortions cause the loss of geometrical synchronization that is necessary in watermark detection and decoding [2]. Although some significant progresses have been made recently [5][6][7], the existing approaches generally require an extensive computational load, or need to work in multiple domains (such as DWT and DFT) and are not robust to JPEG compression. There exist two different approaches to resisting geometrical attacks, i.e., the blind and non-blind ones. For the non-blind approach, due to the availability of the original image, the loss of synchronization caused by geometrical distortions can be recovered efficiently. While the blind one, which does not use the original image in watermark extraction, has wider applications but is obviously more challenging. Three major approaches for the blind solutions have been reported in the literatures. The first approach hides the watermark signal in the invariant domain of the host signal (invariant with respect to rotation, scaling, translation and etc.). In [8], Ruanaidh *et al.* proposed a watermarking scheme based on transform invariants via applying Fourier-Mellin transform to the magnitude spectrum of the original image. However, the resulting stego-image quality is poor due to interpolation errors. The second approach exploits the self-reference principle based on an auto-correlation function (ACF) or the Fourier magnitude spectrum of a periodical watermark [9]. Unfortunately, the proposed watermarking scheme is generally vulnerable to loss coding operation such as JPEG compression. The third approach incorporates the template for watermark synchronization. In [7], Kang and Huang proposed a DWT-DFT composite watermarking scheme, where the messages and templates are embedded into DWT and DFT domain, respectively. Relatively high robustness is observed against both affine transformation and JPEG compression. However the proposed watermarking scheme is required to work in multiple domains.

In this paper, a new robust image watermarking algorithm based on the steerable pyramid transform is proposed, where the rotation synchronization and robust watermarking against JPEG compression are concurrently obtained on the same transform domain. The rotation synchronization is achieved through the template matching, which is just the operation of interpolation with greatly reduced complexity of computation. Under the framework of steerable pyramid, the watermarks can be embedded into an oriented subband at angle θ, which can also be interpolated with base filter kernels and used as a key in watermark detection to increase the security of watermark.

The vector HMM model developed in [10] is also extended to steerable wavelet domain, and the resulting HMM based watermark detector achieves significant improvement in performance compared to the conventional correlation detector. Simulation results demonstrate that the proposed watermarking algorithm is robust against joint Stirmark attacks (the joint attacks of rotation and JPEG compression, additive noise, median cut, etc).

The remainder of the paper is organized as follows. In section 2, the framework of steerable pyramid is briefly reviewed. And the theory behind efficient rotation

synchronization and watermark security in steerable pyramid transform domain is established. By replacing the standard wavelet transform with steerable pyramid, the WD-HMM model is extended and reviewed in section 3. Section 4 gives the overall framework of the proposed rotation invariant secure image watermarking algorithm. The simulation results and analysis are presented in section 5. Finally, we draw the conclusion in section 6.

2 The Steerable Pyramid Transform for Image Watermarking

The standard wavelet transforms have only limited oriented resolution. If the image is rotated, then in the wavelet domain the wavelet coefficients change completely. To solve these problems, a variant on standard wavelet transform, i.e., the steerable pyramid, was proposed by Simoncelli [3]. This transform satisfies the shiftability in orientation.

2.1 The Steerable Pyramid

The steerable pyramid is based on a set of steerable filter kernels which could be tuned to cover all orientations of interest in the image. Given the 2-D, circularly symmetric Gaussian function G:

$$G(x, y) = e^{-(x^2+y^2)} \tag{1}$$

where the scaling and normalization constants have been set to 1 for convenience.

The first partial derivatives in x and y of the Gaussian function constitute a set of filter kernels, i.e.,

$$\begin{aligned} G^0(x, y) &= \frac{\partial}{\partial x} G(x, y) = -2xe^{-(x^2+y^2)}, \\ G^{90}(x, y) &= \frac{\partial}{\partial y} G(x, y) = -2ye^{-(x^2+y^2)}. \end{aligned} \tag{2}$$

It is straight to show that a filter $G^\theta(x, y)$ at arbitrary orientation θ can be interpolated with G^0 and G^{90}:

$$G^\theta(x, y) = \cos(\theta)G^0(x, y) + \sin(\theta)G^{90}(x, y) \tag{3}$$

Let $f(x, y)$ denotes the original image, then

$$R^0(x, y) = (f * G^0)(x, y) \tag{4}$$

is the image filtered for vertical features and

$$R^{90}(x, y) = (f * G^{90})(x, y) \tag{5}$$

for horizontal features. For image features at orientation θ, we further have

$$R^\theta(x, y) = \cos(\theta)R^0(x, y) + \sin(\theta)R^{90}(x, y) \tag{6}$$

For large number of orientations, Eq. (6) is computationally far less expensive than filtering with $G^\theta(x, y)$ directly for an equal number of orientations.

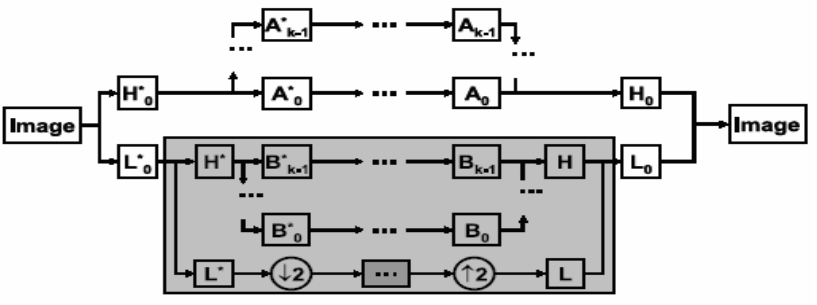


Fig. 1. Block diagram of steerable pyramid

Steerable pyramid is a recursive multi-scale and multi-orientation decomposition. Fig.1 shows its block diagram, where H and L are high-pass and low-pass filters, respectively. And the B_k 's are the oriented steerable filters for a resolution scale. As a bonus, the steerable pyramid representation is also translation-invariant. It is noted that second and third derivatives of Gaussian function give rise to 3 and 4 oriented subbands Without loss of generality, we use two oriented subbands per resolution scale in this paper, i.e., $k = 2$ (unless mentioned otherwise).

2.2 Rotation Synchronization Using Steerable Pyramid

Using the steerable pyramid above, we now develop the theory and framework for rotation synchronization.

***Proposition 1*:** Suppose that $f(x, y)$ and $f^{\theta}(x, y)$ are the original image and its rotated version by θ, respectively. $G^{\theta}(x, y)$ and $R^{\theta}(x, y)$ are the filter kernel and response at angle θ. R^{θ_1,θ_2} represents the response of $G^{\theta_2}(x, y)$ to $f^{\theta_1}(x, y)$. And $T[\circ,\theta]$ denotes the rotation operator such that for any image $f(x, y)$, $T[f(x, y),\theta]$ is $f(x, y)$ rotated by θ anti-clockwise. Then for $\forall\theta$ and a steerable pyramid with two oriented subbands, $R^{0,0}(x, y) = T\left[R^{\theta,\theta}(x, y),-\theta\right]$ and $R^{0,90}(x, y) = T\left[R^{\theta,\theta+90}(x, y),-\theta\right]$.

***Proof*:** Recall that the first derivative of Gaussian function results in a set of steerable filter, i.e., $G^{0}(x, y)$ and $G^{90}(x, y)$.

$$\begin{aligned} &R^{\theta,\theta}(x, y) = f^{\theta}(x, y) * G^{\theta}(x, y) = f^{\theta}(x, y) * \left[G^{0}(x, y)\cos(\theta) + G^{90}(x, y)\sin(\theta)\right] \\ &= f\left[x\cos(\theta) + y\sin(\theta), y\cos(\theta) - x\sin(\theta)\right] * \left[-2xe^{-(x^2+y^2)}\cos(\theta) - 2ye^{-(x^2+y^2)}\sin(\theta)\right]. \end{aligned} \tag{7}$$

Rotate the $R^{\theta,\theta}(x, y)$ by $(-\theta)$, we have

$$\begin{aligned} x &= x'\cos(-\theta) + y'\sin(-\theta), \\ y &= y'\cos(-\theta) - x'\sin(-\theta). \end{aligned} \tag{8}$$

Substitute (8) into (7) and have some trigonometric manipulations, we further have

$$T\left[R^{\theta,\theta}(x, y), -\theta\right] = f(x', y')\left[-2x'e^{-(x'^2+y'^2)} - 2y'e^{-(x'^2+y'^2)}\right]. \tag{9}$$

Therefore, we obtain $T(R^{\theta,\theta}(x, y), -\theta) = R^{0,0}(x, y)$. Similarly, we have $T(R^{\theta,90+\theta}(x, y), -\theta) = R^{0,90}(x, y)$. The result can also be extended to the cases where 3 or 4 steerable filters are used [3].

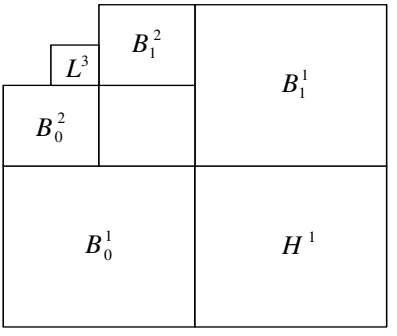


Fig. 2. Two scale steerable pyramid model decomposition for *k*=2

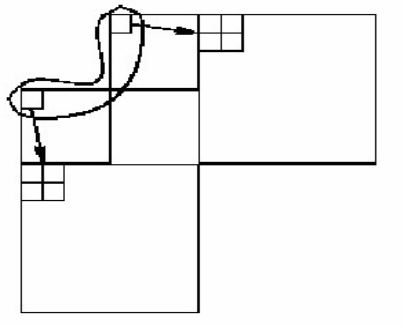


Fig. 3. Steerable wavelet vector HMM (two levels)

Fig.2 shows a two-stage steerable pyramid decomposition for *k*=2, where the L^j and H^1 denote the low-pass bank at scale *j* and high-pass band at scale 1, respectively; and B_k^j represents the *k*'s ($k = 0,1$; 0 and 1 corresponds to the oriented subband at 0^0 and 90^0) seeable pass-band at scale *j*. **Proposition 1 implies the principle for rotation synchronization using steerable pyramid.** The designed template is placed in B_0^1 for highest oriented resolution. For a rotated image $f^{\theta}(x, y)$, if the template is detected in BT, where $BT = T\left[B_0^1\cos(\theta) + B_1^1\sin(\theta), \quad -\theta\right]$, then the rotated image is recovered with angle θ. The detailed template design and efficient matching scheme are included in section 4.

2.3 The Security of Watermarking

One of the objectives for watermarking systems design is security, i.e., the embedded watermark should only be accessible by authorized parties, and be undetectable by

unauthorized users in general. The framework of steerable pyramid provides the necessary security for watermarking system design to some extent.

For a *J*-level decomposition of steerable pyramid $\left\{H^1, L^{J+1}, B_0^j, B_1^j, j = 1 \cdots J\right\}$, rather than the subband B_0^j and B_1^j at scale *j*, two oriented subband at angle θ_1 and θ_2 are randomly selected:

$$\begin{aligned} B_{\theta_1}^j &= B_0^j \cos(\theta_1) + B_1^j \sin(\theta_1), \\ B_{\theta_2}^j &= B_0^j \cos(\theta_2) + B_1^j \sin(\theta_2). \end{aligned} \tag{10}$$

The watermark signals are embedded according to:

$$\hat{B}_{\theta_1}^j = B_{\theta_1}^j + \alpha_1 w_1^j \ and \ \hat{B}_{\theta_2}^j = B_{\theta_2}^j + \alpha_2 w_2^j \tag{11}$$

According to (10), the $\hat{B}_0^j$ and $\hat{B}_1^j$ can be reconstructed from $B_{\theta_1}^j$ and $B_{\theta_2}^j$ via

$$\begin{aligned} \hat{B}_0^j &= \left[\hat{B}_{\theta_1}^j \sin(\theta_2) - \hat{B}_{\theta_2}^j \sin(\theta_1)\right] \Big/ \left[\cos(\theta_1)\sin(\theta_2) - \cos(\theta_2)\sin(\theta_1)\right], \\ \hat{B}_1^j &= \left[\hat{B}_{\theta_1}^j \cos(\theta_2) - \hat{B}_{\theta_2}^j \cos(\theta_1)\right] \Big/ \left[\sin(\theta_1)\cos(\theta_2) - \sin(\theta_2)\cos(\theta_1)\right]. \end{aligned} \tag{12}$$

where $j = 1, \cdots, J$. And finally based on $\left\{H^1, L^{J+1}, \hat{B}_0^j, \hat{B}_1^j, j = 1 \cdots J\right\}$ the watermarked image is obtained via inverse steerable pyramid transform.

The angle θ for oriented subband can be randomly selected from a large space, ranging from $0 \sim 2\pi$. Combined with other secure schemes, such as the random selection of vector tree and Direct Sequence Spread Spectrum (DSSS) [10], the watermark signal can be hidden in a secret multi-scale and multi-orientation pyramid transform domain, making it difficult for the hostile attack that seeks to remove or destroy the watermark at specific locations.

3 The Steerable Wavelet HMM Model

The approach in [10] can be extended to develop the steerable wavelet domain vector HMM for robust watermarking system design. For a steerable pyramid with two oriented subband as shown in Fig.2, each coefficient $w_{j,i}$ in B_k^j has its hidden state $s_{j,i}$ ($1 \le j \le J$, $j = J$ denotes the coarsest scale). Given $s_{j,i} = m$, $w_{j,i}$ is modeled with a zero-mean Gaussian $g(0, \sigma_{j,i}^{(m)})$. Also if a two-states HMM is adopted, the pdf of $w_{j,i}$ is given by

$$f_j(w) = p_j^{(1)} g(w; \sigma_j^{(1)}) + p_j^{(2)} g(w; \sigma_j^{(2)}) \tag{13}$$

where $p_j^{(1)} + p_j^{(2)} = 1$, and $p_j^{(1)}, p_j^{(2)}$ in (13) represent the probability that $w_{j,i}$ is small or large (in statistical sense), respectively.

The steerable pyramid HMM model captures the energy dependency across scale by using Markov chain to describe the probability of hidden state transition from the parent node to its four child nodes, i.e.,

$$A_j = \begin{pmatrix} p_j^{1\to 1} p_j^{1\to 2} \\ p_j^{2\to 1} p_j^{2\to 2} \end{pmatrix}, \; j = 1, 2, \cdots, J-1. \tag{14}$$

where $p_j^{m'\to m}$ represents the probability that child node is in state m given that its parent node is in state m'. Let $\mathrm{p}_j = \left(p_j^{(1)} p_j^{(2)}\right)$ and $\mathrm{p}_j = \mathrm{p}_{j+1}\mathrm{A}_j$, then

$$\mathrm{p}_j = \mathrm{p}_J \mathrm{A}_{J-1}\mathrm{A}_{J-2}...\mathrm{A}_j, \; j = 1, 2, \cdots, J-1. \tag{15}$$

Therefore, the steerable wavelet HMM model is completely defined by a set of parameters:

$$\theta = \{\mathrm{p}_J, \mathrm{A}_{J-1}, ..., \mathrm{A}_1; \sigma_j^{(m)}, (j = 1, ..., J, m = 1, 2)\}. \tag{16}$$

Taking into account the cross-orientation dependency of steerable wavelet coefficients, the steerable vector HMM model is developed as shown in Fig.3. Denote the subband coefficients at orientation d (d =0,1), scale j and location i as $w_{j,i}^d$, the grouping operation results in vectors of coefficients: $\mathbf{w}_{j,i} = \left(w_{j,i}^{(0)} w_{j,i}^{(1)}\right)^T$. For vector WD-HMM model, we have

$$f_j(\mathbf{w}) = p_j^{(1)} g(\mathbf{w}, \mathbf{C}_j^{(1)}) + p_j^{(2)} g(\mathbf{w}; \mathbf{C}_j^{(2)}) \tag{17}$$

where $g(\mathbf{w}; \mathbf{C})$ denotes the zero-mean multivariate Gaussian density with covariance matrix $\mathbf{C}$, i.e.,

$$g(\mathbf{w}; \mathbf{C}) = \frac{1}{\sqrt{(2\pi)^n \left|\det(\mathbf{C})\right|}} \exp(-\mathbf{w}^T \mathbf{C}^{-1} \mathbf{w}), \tag{18}$$

where n is the numbers of orientations (in this case n=2).

Therefore, the steerable pyramid coefficients of image can be modeled by a vector HMM model with a set of parameters:

$$\Theta = \{\mathrm{p}_J, \mathrm{A}_{J-1}, ..., \mathrm{A}_1; \mathbf{C}_j^{(m)}, (j = 1, ..., J, m = 1, 2)\} \tag{19}$$

4 Watermarking Embedding and Detection

The rotation-invariant secure image watermarking scheme based on steerable pyramid transform (SPT) is given in this section, which includes the efficient template matching for rotation synchronization, the strategy of watermarking security, and the steerable vector HMM based watermarking scheme.

4.1 Embedding of Template and Watermark

Fig.4 shows the overall structure for the proposed template and watermark embedding scheme.

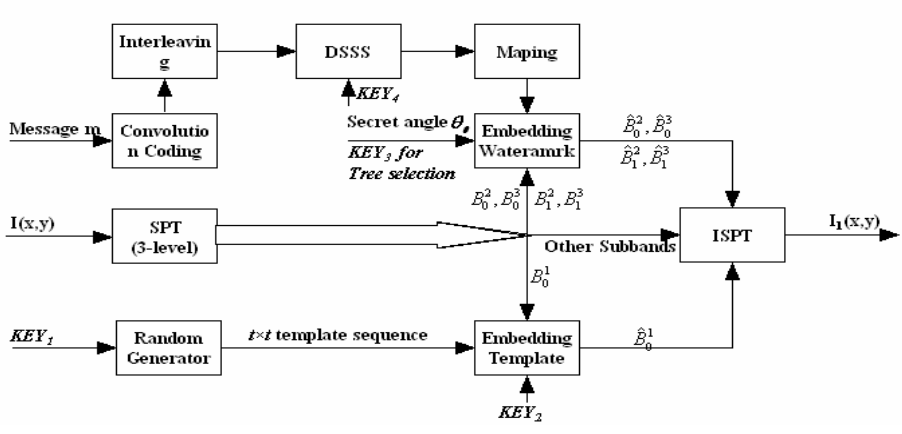


Fig. 4. Embedding scheme of the template and watermark

4.1.1 Watermark Coding

For watermark message **m** , the watermark coding includes:

1. Convolution coding for watermark **m**: let $\mathbf{m} = \{m_i; i = 1,2,...,L, m_i \in [0,1]\}$ be the watermark message, where L is the length of the message. Then **m** is coded with the 1/3 convolution code to generate a sequence $\mathbf{m}_c$ with the length of L_c;
2. Interleaving of $\mathbf{m}_c$ to generate $\mathbf{m}_I$;
3. DSSS for $\mathbf{m}_I$: The interleaved message $\mathbf{m}_I$ is DSSS (Direct Sequence Spread Spectrum) with PN sequence **p** of length N_p, which is generated with a secret key KEY_4. The coded watermark message is $\mathbf{w}=\{w_i; w_i \in \{-1,+1\}, i = 1,2,..., L_c * N_p\}$.

4.1.2 Template Design and Embedding

As the filter bank in Fig.1 is non-orthogonal and near PR (perfect reconstruction), and uses a set of non-separated filters of 9×9 for the steerable band-pass filter B_k , there exists some error spreading around its neighbor if some coefficients in oriented subband B_j^k are modified. Fig.5 shows the effect of error spreading, where (1) the image is decomposed with 4 scale steerable pyramid and the coefficient at (32,32) in B_0^1 is modified by –50; (2) the image is reconstructed with the modified coefficients; (3) the image is decomposed into pyramid again and the coefficients around (32,32) is more or less modified. As the radius of the region where the coefficients are significantly affected is about 6 pixels, we design a template of 32×32 with lattice structure. The lattice points of the template are spaced apart with 10 pixel alone x and y axis to alleviate the effect of error spreading.

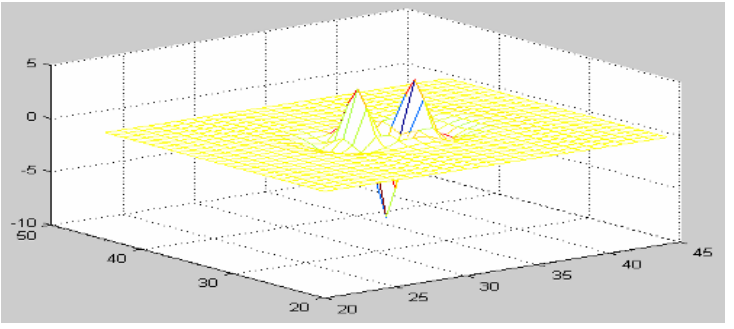


Fig. 5. The effect of error spreading in oriented subband

A PN sequence $\{t(i)\}$ of length 32×32 is generated with key KEY_1, and used to construct the template described above. Also the template is place in the center of B_0^1 to avoid the cropping attack. Let $w(u_i, v_i)$ be the coefficient in B_0^1 which is used to host the template bit $t[i]$, then the template are embedded according to:

$$w'(u_i, v_i) = w(u_i, v_i) + \beta * t(i), t(i) \in \{+1. -1\} \; and \; i \in [1,1024] \subset Z \,. \tag{20}$$

where β is a global parameter to adjust the embedding strength.

4.1.3 Watermark Embedding

For the coded watermark message $\mathbf{w} = [w_i]$ of length $L_c * N_p$, the watermark embedding process is as follows:

1. The original image $I(x, y)$ is decomposed into L-level steerable pyramids (L=3 in our scheme), where scale 1 for template and scale 2-3 for watermark message;
2. As described in section 2.3, rather than the band $\{B_0^j, B_1^j; j = 2,3\}$, the watermark messages are embedded into randomly selected oriented subband $\{B_{\theta_1}^j, B_{\theta_2}^j; j = 2,3\}$ to increase the security of watermarking. The θ_1 and θ_2 are assigned to be θ_0 and $(\theta_0 + 90)$ in our work for good visual quality and high capacity. And θ_0 is used as the key in watermark detection;
3. Under the framework of HMM model, the carrier of watermark signal is vector tree. In the interest of resisting against JPEG attack, the watermark is only embedded into the coarsest 2 scales ($j = 2,3$). The resulting vector tree includes 10-nodes as shown in Fig.3;
4. Each vector tree is used to embedded 1 bit for the coded message $\mathbf{w} = [w_i]$, and the optimal strategy for 1-to-10 mapping $\mathbf{k}$ is designed based on the principle given in [10]. Fig.6 shows the mapping rule, where the dot and circle stand for same and inverse version of the to-be-embedded bit w_i, respectively;

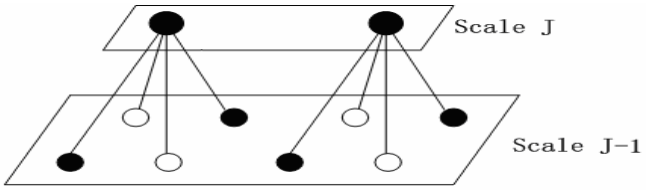


Fig. 6. Optimal mapping strategy for each vector tree

5. The t^{th} mapped pattern $\mathbf{k}(t)$ for w_t is embedded into the vector tree according to

$$\mathbf{x}'(t,i) = \mathbf{x}(t,i) + \beta * a(t,i) * \mathbf{k}(t,i) \tag{21}$$

where $\mathbf{x}(t,i)$ is the i^{th} node of the t^{th} vector tree, $a(t,i)$ is its corresponding HVS masking weight and β is the global adjustment factor for embedding strength [10].

6. After all coded message bits are embedded into $L_c * N_p$ vector trees which are randomly selected with the secret key KEY_3, the inverse steerable pyramid transform is performed to obtain the watermarked image $I'(x, y)$.

4.2 Rotation Synchronization Via Template Matching

For a rotated watermarked image, the rotation synchronization should be achieved before the watermark can be extracted. Let $I^{\theta}(x, y)$ be the rotated image, and the template $t[i]$ generated with key KEY_1 is contained in the oriented subband B_0^1 of original image $I(x, y)$, if the template can be detected via a correlation detector in $T\left[B_{\theta}^1, \ -\theta\right]$, then the rotated image is synchronized with angle θ. Incorporating the steerable pyramid transform, two optimized strategies are developed for efficient template matching, which are described as follows:

A. Efficient template matching

1. For a matching angle α and the template $t[i]$ of 32×32, the correlation can be computed via (22):

$$t'[i] = B_{\alpha}^1\left[u_i \cos(\alpha) + v_i \sin(\alpha), \quad v_i \cos(\alpha) - u_i \sin(\alpha)\right], i = 1, 2, \cdots, 1024.$$

$$Cor(\alpha) = \frac{\sum_{i=1}^{1024} t'[i]t[i]}{\sqrt{\sum_{i=1}^{1024} t'^2[i]}\sqrt{\sum_{i=1}^{1024} t^2[i]}} \tag{22}$$

where (u_i, v_i) is the location of $t[i]$ in B_0^1.

2. It is noted that, instead of rotating each pixel of B_{α}^1 by $-\alpha$, only 32×32 pixels in B_{α}^1 are accessed to compute the $Cor(\alpha)$ using Eq. (22). Therefore the computation load is greatly reduced.

B. Coarse to fine template searching

1. The template searching is begun with a coarse stage, where a relatively large step $\Delta\alpha_c$ (e.g., $\Delta\alpha_c = 0.5$) is used to compute the $Cor(\alpha)$ with $\alpha = \alpha + \Delta\alpha_c$. A coarsely recognized rotation angle θ_c is obtained.
2. Based on the roughly estimated θ_c , a fine searching is carried out with an accurate step $\Delta\alpha_f$ (say, $\Delta\alpha_f = 0.1$) around the neighbor of θ_c .

4.3 Steerable HMM-Based Watermark Detection

Assume the watermark signal is embedded into secure oriented subband $\left\{B_{\theta_0}^j, B_{\theta_0+90}^j\right\}_{j=2,3}$ with key θ_0 .

1. Generate the secure oriented subbands $\left\{B_{\theta_0}^j, B_{\theta_0+90}^j\right\}_{j=2,3}$ of $I(x, y)$ from the oriented subbands $\left\{B_0^j, B_1^j\right\}_{j=2,3}$ of $I^\theta(x, y)$ according to

$$\begin{aligned} B_{\theta_0}^j &= T[B_0^j \cos(\theta+\theta_0) + B_1^j \sin(\theta+\theta_0), -\theta]_{j=2,3} , \\ B_{\theta_0+90}^j &= T[B_0^j \cos(90+\theta+\theta_0) + B_1^j \sin(90+\theta+\theta_0), -\theta]_{j=2,3} . \end{aligned} \tag{23}$$

2. The recovered oriented subbands $\left\{B_{\theta_0}^j, B_{\theta_0+90}^j\right\}_{j=2,3}$ are used to construct the posterior vector HMM model with parameters set Θ . The vector trees to carry the watermark signal are determined with the key KEY_3.
3. The watermark is detected with a vector HMM detector as described in [10]

$$\ln f_x(\mathbf{T}_Z^t - \mathbf{a}^t * \mathbf{k}_0 \mid \Theta) > \ln f_x(\mathbf{T}_z^t - \mathbf{a}^t * \mathbf{k}_1 \mid \Theta) , \tag{24}$$

   where $f_x(\circ)$ is the likelihood function, $\mathbf{T}_z^t$ stands for the selected vector tree, and $\mathbf{k}_0$ and $\mathbf{k}_1$ are the mapping pattern for bit 0 and 1, respectively (only 1 bit message is embedded into each selected tree). Eq.(24) implies that the HMM detector outputs the pattern with maximum likelihood, which is taken as the detected watermark;
4. Based on the detected sequence, the operations of de-mapping, de-DSSS, and convolution decoding are implemented to get the decoded watermark signal $\hat{\mathrm{w}}$.

5 Simulation Results and Analysis

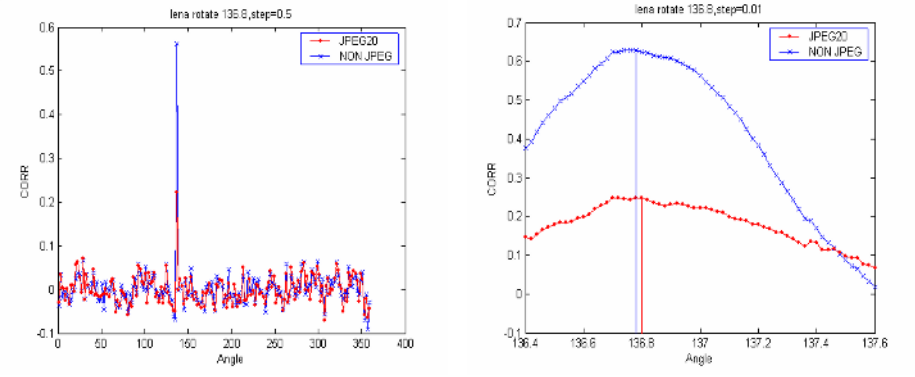
In our experiments, 5 standard 512*512*8b images with different texture characteristic are tested. The images are firstly decomposed into 3-level pyramids with the steerable pyramid transform; the 32*32 template and 60-bit meaningful watermark messages are embedded in the B_0^1 and $\left\{B_\theta^j, B_{\theta+90}^j\right\}_{j=2,3}$, respectively. Here, θ is the parameter to describe the secure oriented subband. Fig.7 shows the steerable pyramid based watermarked image with template.

Fig. 7. Watermarked images with template: (a) baboon (PSNR=35.91dB); (b) fishingboat (PSNR=38.98dB); (c) f16 (PSNR=37.38dB); (d) lena (PSNR=40.60dB); (e) peppers (PSNR=37.86dB)

Fig.8 shows the performance of rotation synchronization under JPEG attacks with steerable pyramid transform, where the watermarked image "lena" in Fig.7 is rotated by 136.8^o and the coarse to fine template matching scheme developed in section 4.2 is employed. For the case without JPEG attack ("x" mark line), the searching step is set to be 0.5^o at the coarse stage and the estimated angle is 137^o, as shown in Fig.8 (a). Then based on the roughly estimated angle, the fine searching is implemented with step of 0.01^o, and the finally recognized rotation angle is 137.78^o, as shown in Fig.8 (b). The 2-stage searching process needs only 820 searching operations and takes 0.5s when it is implemented with Matlab6.5 on a P4-2.4G PC, which shows good searching resolution and efficiency. The searching process under JPEG20 attack ("$\bullet$" mark line) is also given in Fig.8, which demonstrates that the proposed rotation synchronization scheme is robust to deep JPEG attack. Similar results are observed with other test images.

To evaluate the secure watermarking strategy described in section 2.3, the secret angle θ is set to 12.36^o, and two oriented subband at 12.36^o and 102.36^o ($=\theta+90^0$) are generated to embed watermark signal. Fig.9 gives the security performance under different orientation angle θ, where the "x" mark, "$\bullet$" mark and "o" mark line stand for BER performance after HMM detector, De-DSSS and Viterbi decoding, respectively. Fig.9 shows that the farther away the oriented subband derivates from the secure angle θ, the higher the detection BER would be. The watermark signal can only be recovered from those oriented subbands near the secret angle θ ($\Delta\theta \approx 15^o$). Considering the fact that the secure angle θ used in watermark detection runs from 0 to 2π, when combined with other strategies such as the random selection of vector tree and the key to generate the PN for DSSS, the proposed scheme greatly increases the security of watermarking to hostile attacks.

The watermarked images in Fig.7 are attacked with StirMark4.0 [12], and then the proposed steerable HMM based detectors are employed to detect the watermark signals. Table 1 shows the performance under Stirmark attacks, which are very robust against Stirmark attacks. Due to the fact that the steerable pyramid transform is near PR with a reconstruction PSNR of 50dB, the objective quality (PSNR) of the watermarked image with steerable pyramid is relatively lower than that when other PR transforms, such as DCT and 9/7 biorthogonal wavelet, are used.



(a) Estimation at coarse stage (b) Estimation at fine stage

Fig. 8. Template matching

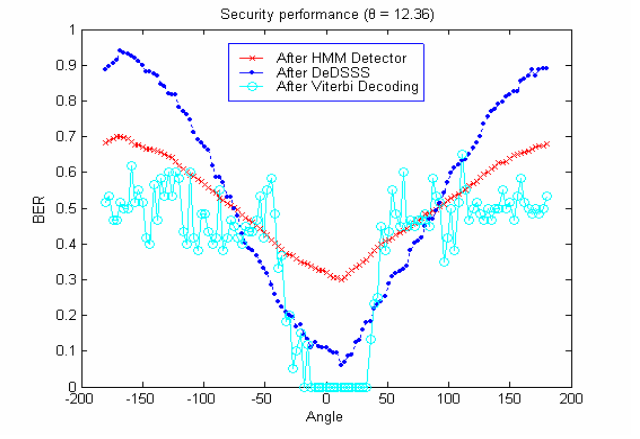


Fig. 9. Security performance under different orientation angle θ

Table 1. Performance of steerable vector HMM-based detector under StriMark attack

Images / Attacks	lena	baboon	F16	fishingboat	peppers
PSNR(dB)	40.60	35.91	37.38	38.98	37.86
JPEG	10~100	15~100	12~100	13~100	10~100
AddNoise	1~5	1~3	1~3	1~2	1~6
MedianCut	2~5	Fail	2~5	2~3	2~5
Gaussian	Ok	Ok	Ok	Ok	Ok
Sharpening	Ok	Ok	Ok	Ok	Ok

Table 2. Performance under joint StriMark attacks: Rotation 30.5° + (JPEG, Additive noise, MedianCut, or filter)

Attacks \ Images	lena	baboon	f16	fishingboat	peppers
PSNR(dB)	40.60	35.91	37.38	38.98	37.86
JPEG	13~100	22~100	19~100	20~100	17~100
Additive noise	1~4	Fail	1	Fail	1~4
MedianCut	2~3	Fail	2~3	Fail	2~3
Gaussian	Ok	Ok	Ok	Ok	Ok
Sharpening	Ok	Fail	Ok	Fail	Ok

In addition, the performance against the joint attacks is also investigated. The images in Fig.7 are first rotated by 30.5° and then attacked with the StirMark4.0 such as JPEG compression, addtitive noise, median cut, and filter (Gaussian and sharpening). After the rotation synchronization is obtained with template matching, the watermark is extracted from the secure oriented subbands based on steerable HMM model. The performance with the proposed watermarking algorithm is given in Table 2, which demonstrates high robustness against joint Stirmark attacks.

6 Conclusion

In this paper, we present a rotation-invariant secure image watermarking algorithm based on the steerable pyramid transform. The rotation synchronization is obtained through efficient template matching via steerable pyramid transform, which satisfies shiftability in orientation condition. By embedding the watermark signal into randomly selected oriented subband at angle θ, which can be interpolated with base filter kernels and used as a key in watermark detection, the security of the proposed watermarking system is greatly increased. The HMM model is also extended to develop a steerable wavelet HMM model for robust watermarking system design. Under the framework of steerable pyramid transform, the rotation synchronization, robustness and security of watermarking are concurrently obtained in the same transform domain. And simulation results with the proposed algorithm demonstrate high robustness against StirMark attacks (rotation, JPEG compression, additive noise, median cut, etc.) and Joint StirMark attacks (the Joint attack of rotation and JPEG compression, etc).

Acknowledgments

The authors appreciate the supports received from NSFC (60325208, 90604008), 973 Program (2006CB303104) and NSF of Guangdong (04205407).

References

1. P.Meerwald and A.Uhl, "A Survey of Wavelet-Domain Watermarking Algorithms," *Proceeding of SPIE*, Security and Watermarking of Multimedia Contents III, Vol.4314, San. Jose, CA, USA, Jan. 2001.
2. J.Cox, M.L.Miller and J.A.Bloom, Digital Watermarking, *Morgan Kaufmann*, 2001
3. E.P.Simoncelli, W.T.Freeman, E.H.Adelson and D.J.Heeger, "Shiftable Multi-scale Transform," *IEEE Trans. on Information Theory*, Vol.38(2), pp.587-607, March 1992.
4. W.T.Freeman and E.H.Adelson, "The Design and Use of Steerable Filters," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.13, No.9, pp.891-906, Sep.1991.
5. F.Hartung and M.Kutter, "Multimedia Watermarking Techniques," *Proc. of IEEE*, Vol.87, pp.1079-1107, July 1999.
6. C.Y.Lin, M.Wu, J.A.Bloom, J.Cox, M.L.Miller and Y.M.Lui, "Rotation, Scale and Translation Resilient Watermarking for Images," *IEEE Trans. on Image Processing*, ol.10, pp.767-782, May 2001.
7. X.G. Kang, J.W. Huang, Y.Q.Shi and Y.Lin, "A DWT-DFT Composite Watermarking Scheme Robust to Both Affine Transform and JPEG Compression," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.13, No.8, pp: 776-786, Aug. 2003.
8. J.Ruanaidh and T.Pun, "Rotation, Scale and Translation Invariant Spread Spectrum Digital Image Watermarking," *Signal Processing*, Vol.6, No.3, pp.303-317, 1998.
9. M.Kutter, "Watermarking Resistance to Translation, Rotation and Scaling," *Proc. of SPIE: Media Systems Applications*, Vol.3528, pp:423-431, 1998.
10. J.Ni, R.Zhang, J.Huang and C.Wang, "A Robust Multi-bit Image Watermarking Algorithm Based on HMM in Wavelet Domain," *Lecture Notes in Computer Science: Proc.of IWDW 2005*, Vol. 3710, pp: 110-123, Springer-Verlag.
11. C. Fei, D. Kundur and R. H. Kwong, "Analysis and Design of Watermarking Algorithms for Improved Resistance to Compression," *IEEE Trans. on Image Processing*, Vol. 13, No. 2, pp. 126-144, Feb. 2004.
12. StirMark, [Online], Available: http://www.cl.cam.ac.uk/~fapp2/watermarking /stirmark/.

Error Resilient Image Authentication Using Feature Statistical and Spatial Properties

Shuiming Ye[1, 2], Qibin Sun[1], and Ee-Chien Chang[2]

[1] Institute for Infocomm Research, A*STAR, Singapore, 119613
[2] School of Computing, National University of Singapore, Singapore, 117543
shuiming, qibin@i2r.a-star.edu.sg, changec@comp.nus.edu.sg

Abstract. The pervasive distribution of digital images triggers an emergent need of authenticating degraded images by lossy compression and transmission. This paper proposes a robust content-based image authentication scheme for image transmissions over lossy channels. Content-based image authentication typically assesses authenticity based on a distance measure of feature differences between the testing image and its original. Commonly employed distance measures such as the Minkowski measures may not be adequate for content-based image authentication since they do not exploit statistical and spatial properties of the feature differences. This proposed error resilient scheme is based on a statistics- and spatiality-based measure (*SSM*) of feature differences. This measure is motivated by an observation that most malicious manipulations are localized whereas acceptable manipulations cause global distortions. Experimental results show that *SSM* is better than previous used measures in distinguishing malicious manipulations from acceptable ones, and the proposed *SSM*-based scheme is robust to transmission errors and other acceptable manipulations, and is sensitive malicious image modifications.

Keywords: Image Authentication, Error Resilience, Feature Distance Measure, Image Transmission, Digital Watermarking, Digital Signature.

1 Introduction

Image transmission is always affected by transmission errors due to environmental noises, fading, multi-path transmission and Doppler frequency shift in wireless channel [1], or packet loss due to congestion in Internet network [2]. Normally errors under a certain level in images would be tolerable and acceptable. Therefore, it is desirable to authenticate images even if there are some uncorrectable but acceptable errors. That is, image authentication should be robust to acceptable transmission errors besides other acceptable image manipulations such as smoothing, brightness adjusting, compressing or noises, and be sensitive to malicious content modifications such as object addition, removal, or position modification.

A straightforward way of image authentication is to treat images as data, so that data authentication techniques can be used for image authentication. Several approaches to authenticate data stream damaged by transmission errors have been proposed. Golle et al. proposed an approach based on efficient multi-chained packet

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 461 – 472, 2006.

signature [2]. Perrig et al. proposed to use an augmented signature chain of packets [3] for one packet. However, treating images as data stream during authentication does not take advantage of the fact that images are tolerable to some certain degree of errors, and the computing payload would be very large. Therefore, it is not suitable for these data approaches to be applied directly to image authentication.

In order to be robust to acceptable manipulations, several content-based image authentication schemes have been proposed [4, 5, 6]. Content-based authentication typically measures feature distortion in some metrics, so authenticity fuzziness would be introduced in these approaches which may make the authentication result useless. Furthermore, transmission errors would damage the encrypted signatures or embedded watermarks. Therefore, previous techniques would fail if the image is damaged by transmission errors.

The objective of this paper is to authenticate images received through lossy transmission when there are some uncorrectable transmission errors. It aims to distinguish the images damaged by transmission errors from the images modified by the malicious users. It focuses on the development of error resilient image authentication schemes incorporated with error correcting code, image feature extraction, transmission error statistics, error concealment, and perceptual distance measure for image authentication. The results of this paper would provide a way to authenticate images even if there are uncorrectable transmission errors in them. The proposed authentication scheme produced good robustness against transmission errors and some acceptable manipulations, and it was sensitive to malicious modifications. Moreover, the perceptual distance measure proposed for image authentication should improve the system performance of content based image authentication schemes. Therefore, many acceptable manipulations, which were detected as malicious modifications in previous schemes, would be correctly verified by the proposed scheme based on this measure.

2 Proposed Error Resilient Image Authentication

2.1 Statistics- and Spatiality-Based Measure for Image Authentication

Content-based image authentication typically measures authenticity by the distance between a feature vector from the received image and the corresponding vector from the original image, and compares it with a preset threshold to make a decision [7, 8]. The distance metric commonly used is the Minkowski metric $d(X, Y)$ [9] is defined as:

$$d(X,Y) = \left(\sum_{i=1}^{N} | x_i - y_i |^r\right)^{1/r} \tag{1}$$

where X, Y are two N dimensional feature vectors, and r is a Minkowski factor. Particularly, when r is set as 2, it is the Euclidean distance; when r is 1, it is Manhattan distance (or Hamming distance for binary vectors).

The Minkowski metric is not suitable for content-based image authentication. The reason is that the Minkowski metric treats each feature independently which does not exploit statistical and spatial properties of image features. That is to say, it does not exploit statistical or spatial properties of image features.

Many features used in content-based image authentication to represent image content are composed of localized feature value of the image such as edge [10, 4], block DCT coefficients based features [5, 7, 11], highly compressed version of the original image [12], or block intensity histogram [8]. Therefore, the image authentication scheme based on the Minkowski metric may not be able to distinguish the tampered images with small local objects modified from the images by acceptable manipulations such as lossy compression. On the other hand, even if the Minkowski metric distances are the same, the feature difference under typical acceptable modifications and malicious ones are still distinguishable when the feature contains spatial information such as edge or block DCT coefficient based features.

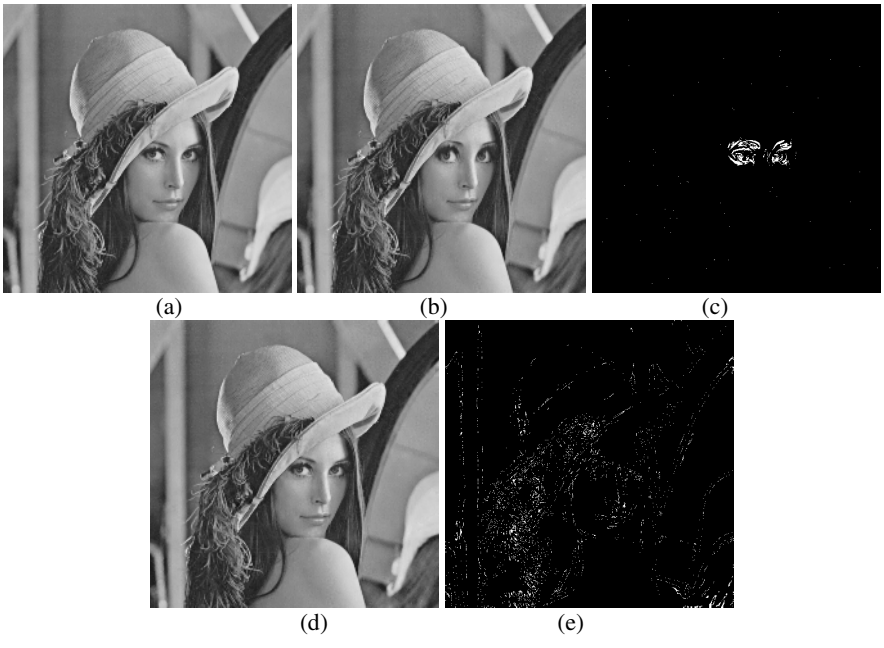


Fig. 1. Discernable patterns of edge feature differences caused by acceptable image manipulation and malicious modification: (a) original image; (b) tampered image; (c) feature difference of (b); (d) error-concealed image; (e) feature difference of (d)

Feature differences are the differences between the feature extracted from the original image and the feature extracted from the testing image. The feature difference distribution pattern is determined by the way the manipulations act on the image content. Image contents are typically represented by objects and each object is usually represented by spatially clustered image pixels. Therefore, the feature to represent the content of the image would inherit some spatial relations. That is to say, the feature contains spatial information and this kind of feature is often used in content-based image authentication. For example, the Hamming distance measures of Fig. 1(b) and Fig. 1(d) are almost the same, but yet, one could argue that Fig. 1(b) is probably distorted by malicious tampering since the feature differences concentrate on the eyes.

A malicious manipulation of an image is always concentrated on modification of small number of objects in the image, changing the image to a new one which carries a different visual meaning to the observer. If the contents of an image are modified, the features may be altered around the objects, and the altered feature points are mostly to be connected with each other. Therefore, the feature differences introduced by a meaningful tampering would be typically spatially concentrated.

On the contrary, acceptable image manipulations such as image compression, contrast adjustment, and histogram equalization introduce distortions globally into the image. Although the feature differences may likely to cluster around objects, they are not as prominent as malicious manipulations. In addition, many objects spread out spatially in the image. Hence, the feature differences are likely to be evenly distributed with little connectedness. The distortion introduced by transmission errors would also be evenly distributed since the transmission errors are randomly introduced into the image [13].

Based on the discernable feature difference pattern, three observations were summarized after examining many instances of feature differences of various image manipulations:

1) The feature differences by most acceptable operations are evenly distributed spatially, whereas the differences by malicious operations are locally concentrated.
2) The maximum connected size of modifications caused by acceptable operations is small whereas the one by tampering operations is large.
3) Even if the size of the maximum connected component is fairly small, the image could have been tampered with if those small components are spatially concentrated.

These observations are supported by our intensive experiments or other literatures listed mentioned previously [4]. The above observations not only refute the suitability of Minkowski metric to be used in image authentication, but also provide hints as to how a good distance function would work: it should exploit the statistical and spatial properties of feature differences.

Based on these observations, a perceptual distance measure is proposed for image authentication with the assumption that the feature contains spatial information. The distance measure is based on the differences of the two feature vectors from the testing image and from the original image. Two measures are used to exploit statistical and spatial properties of feature differences including the kurtosis (*kurt*) of grouped feature difference distribution and the maximum connected component size (*mccs*) in the feature difference map.

To facilitate discussions, we write the feature value x_i to be the feature value at spatial location i, and write X as a N-dimension feature vector, for example, $N=WH$ when using edge feature (W and H are the width and height of the image). The feature difference vector δ is defined as the difference between feature vector X of the testing image and feature vector Y of the original image:

$$\delta_i = | x_i - y_i | \tag{2}$$

where δ_i is the difference of features at spatial location i.

The proposed Statistics- and Spatiality-based Measure (*SSM*) is calculated by sigmoid membership function based on both *mccs* and *kurt*. Below is the definition of

the proposed measure. Given two feature vectors X and Y, the proposed feature distance measure $SSM(X, Y)$ is defined as follows:

$$SSM(X,Y) = \frac{1}{1+e^{\alpha(mccs\cdot kurt\cdot\theta^{-2}-\beta)}} \tag{3}$$

The measure $SSM(X, Y)$ is derived from the feature difference vector δ defined in Eq. (2). The parameter α decides the changing rate at $mccs\cdot kurt\cdot\theta^{-2}=\beta$. β is the average $mccs\cdot kurt\cdot\theta^{-2}$ value of a set of malicious attacked images and acceptable manipulated images. In this paper, the acceptable manipulations are contrast adjustment, noise addition, blurring, sharpening, compression and lossy transmission (with error concealment), and the malicious tampering operations are object replacement, addition and removal. During authentication, if the measure $SSM(X, Y)$ of an image is smaller than 0.5 (that is, $mccs\cdot kurt\cdot\theta^{-2}<\beta$), the image is identified as authentic, otherwise it is unauthentic.

The kurtosis describes the shape of a random variable's probability distribution based on the size of the distribution's tails. It measures how fat or thin the tails of a distribution are relative to a normal distribution. Therefore, it could be used to distinguish feature difference distribution of the malicious manipulations from that of the acceptable manipulations.

Let us partition the spatial locations of the image into neighborhoods, and let N_i be the i-th neighborhood. That is, N_i is a set of locations that are in a same neighborhood. For example, by dividing the image into blocks of 8×8, we have a total of $WH/64$ neighborhoods, and each neighborhood contains 64 locations. Let D_i be the total feature distortion in the i-th neighborhood N_i:

$$D_i = \sum_{j\in N_i} \delta_j \tag{4}$$

We can view D_i as a sample of a distribution D. The $kurt$ in the Eq. (3) is the kurtosis of the distribution D. It can be estimated by:

$$kurt(D) = \frac{\sum_{i=1}^{N}(D_i-\mu)^4}{N\sigma^4} - 3 \tag{5}$$

where N is the number of all samples used for estimation. μ and σ is the estimated mean and standard deviation of D, respectively.

The maximum connected component size ($mccs$) is calculated from morphological operators. Firstly, the isolated points in feature differences are removed and then broken segments are joined by morphological dilation [14]. The maximum connected component size ($mccs$) is then achieved using connected components labeling on the binary feature map based on 8-connected neighborhood [14].

Since images may have different number of objects or details than others, the ratios of the number of extracted feature points to image dimension may be different. Therefore, a normalizing factor θ is introduced:

$$\theta = \frac{\mu}{W\cdot H} \tag{6}$$

where W and H are the width and height of the image respectively. The normalized factor θ introduced can make the proposed measure suitable for most natural scene images despite of the different amount of feature points detected among different images.

2.2 Image Signing

The image signing procedure is outlined in Fig. 2. Firstly, edge of the original image is extracted using the fuzzy reasoning based edge detecting method [15].Then the edge feature is divided into 8×8 blocks, and edge point number in each block is encoded by error correcting code (ECC) [6]. BCH(7,4,1) is used to generate one parity check bit (PCB) for ECC codeword (edge point number) of every 8×8 block. The signature is generated by encrypting the concatenated ECC codewords using a private key. Finally, the PCB bits embedded into the DCT coefficients of the image. In our implementation, the PCB bits are embedded into the middle-low frequency DCT coefficients using the same quantization based watermarking as in [11].

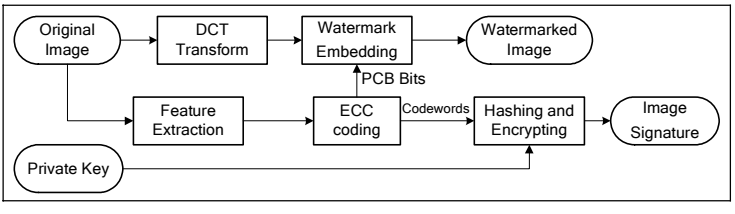


Fig. 2. Signing process of the proposed error resilient image authentication scheme

Let the total selected DCT coefficients form a set **P**. For each coefficient c in **P**, it is replaced with c_w which is calculated by:

$$c_w = \begin{cases} Q\,\mathrm{round}(c/Q), \text{ if } \mathrm{LSB}(\mathrm{round}(c/Q)) = w \\ Q\left(\mathrm{round}(c/Q) + \mathrm{sgn}\left(c - Q\,\mathrm{round}(c/Q)\right)\right), \text{ else} \end{cases} \tag{7}$$

where w (0 or 1) is the bit to be embedded. Function round(x) returns the nearest integrate of x, sgn(x) returns the sign of x, and function LSB(x) returns the least significant bit of x. Eq. (7) makes sure that the LSB of the coefficient is the same as the watermark bit.

Embedding should not affect the signature verification process, since the watermarking procedure would introduce some distortions. In order to exclude the effect of watermarking from feature extraction, a compensation operator C_w is adopted before feature extraction and watermarking. C_w is designed according to the watermarking algorithm. In this paper, we have:

$$\begin{cases} I_c = C_w(I) \\ I_w = f_e(I_c) \end{cases} \tag{8}$$

$$C_w(I) = \mathrm{IDCT}\left\{\mathrm{IntQuan}\left(d_i, 2Q, \mathbf{P}\right)\right\} \tag{9}$$

where d_i is the i-th DCT coefficient of I, and IDCT is inverse DCT transform. $f_e(I)$ is the watermarking function based on quantization using Eq. (7), and I_w is the final watermarked image. The IntQuan(c, **P**, Q) function is defined by:

$$\text{IntQuan}(c, Q, \mathbf{P}) = \begin{cases} c, \text{ if } c \notin \mathbf{P} \\ Q \text{ round}(c/Q), \text{ else} \end{cases} \tag{10}$$

Function $C_w'(I)$ is designed according to the watermarking algorithm, which uses $2Q$ to pre-quantization the DCT coefficients before feature extraction and watermarking. This function (with quantization step $2Q$) makes sure that whatever the watermarks are, the watermark embedding procedure (with quantization step Q) will not affect the extracted feature: from Eq. (7), (9) and (10), we can get $C_w'(I_w) = C_w'(I)$, thus $f_e(I_w) = f_e(I)$, i.e., the feature extracted from the original image I is the same as the one from the watermarked image I_w.

2.3 Image Authenticity Verification

The image verification procedure can be viewed as an inverse process of the image signing procedure, as shown in Fig. 3. Firstly, error concealment is carried out if transmission errors are detected. The feature of image is then extracted using the same method used in image signing procedure and watermarks of the image are also extracted. If there are no uncorrectable errors in ECC codewords, the authentication is based on bit-wise comparison between the decrypted feature and the feature extracted from the image [6]. Otherwise, image authenticity is calculated by the *SSM* based on differences between the feature extracted and the watermark decrypted. Finally, if the image is identified as unauthentic, the attacked areas are then detected.

For wavelet-based images, edge directed filter-based error concealment algorithm proposed in [13] is adopted. For DCT-based JPEG images, a content-based error concealment proposed in [16] is used. It is efficient and advisable for error concealment to be applied before image authentication at the end of receiver since the edge feature of the error-concealed image is much closer to the original one than the damaged image [13, 16]. As a result, the image content authenticity of the error concealed image is higher than that of the damaged image which is validated in our experiments of the error resilient image authentication. On the other hand, in view of applications, the visual information received by the users is generally already error-concealed after transmission over lossy channels.

Given that the image is to be authenticated, we repeat feature extraction described in content signing procedure. The corresponding PCB bits (PCB_W) of all 8×8 blocks (one bit/block) of the image are extracted from the embedded watermarks. Then the feature set extracted from the image is combined with the corresponding PCBs to form ECC codewords. If all codewords are correctable, we concatenate all codewords and cryptographically hash the result sequence. The final authentication result is then concluded by bit-by-bit comparison between these two hashed sets. If there are uncorrectable errors in ECC codewords, image authenticity is calculated based on the proposed distance measure. The two feature vectors in the proposed measure are PCB_W from watermarks and the recalculated PCB bits (PCB_F) from ECC coding of the re-extracted image feature set. If the distance measure between PCB_W and PCB_F is smaller than 0.5 ($SSM(PCB_W, PCB_F)<0.5$), the image is authentic. Otherwise, the image is unauthentic.

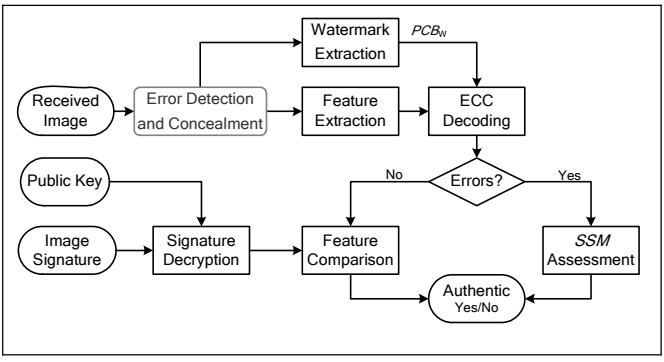


Fig. 3. Image authentication process of the proposed error resilient image authentication scheme

If the image is deduced to be unauthentic, the tampered areas can be detected. Attack location is an important part of the authentication result since the detected attacked areas give the users a clear figure where the image contents are tampered maliciously. The attack areas are detected using information combining watermarks and image feature. Firstly, the difference map is calculated between PCB_W and PCB_F. Morphological operations are used to compute connected areas, removing the isolated pixels and small connected areas. After these operations, the difference map is masked with the union of the watermark and feature. The masking operation can refine the detected areas by concentrating them with the objects in the tampered image or in the original image. The areas in the difference map which do not belong to an object are removed which may be a false alarm of some noises or acceptable image manipulations.

3 Experimental Results and Discussions

In our experiments, different images were used including testing images of JPEG and JPEG2000: *Actor, Barbara, Bike, Airplane, Fruits, Girl, Goldenhill, Lena, Mandrill, Monarch, Pepper, Woman*, and so on. The dimensions of these images differ among 512×512, 640×512, 640×800, and 720×576. *Daubechies* 9/7 wavelet filter is used for the wavelet transformation which is used in JPEG2000 standard [17]. The acceptable manipulations are contrast adjustment, noise addition, blurring, sharpening, compression and lossy transmission, and the malicious tampering operations are object replacement, addition and removal.

Some acceptable distortions and malicious attacks were introduced into the original images for robustness evaluation. The proposed *SSM* was compared with *Hamming* (Minkowski Metric with r=1 for binary feature) as shown in Fig. 4. Pratt's *Figure of Merit* (*FoM*) [18] was also used for comparison since it is good at evaluating distances of images when using edge based features. In Fig. 4, the last three columns are images maliciously tampered from the original image *Lena*, by enlarging the eyes, modifying multiple objects in the image, or adding a small spot in the face. The others are images from some acceptable manipulations including Gaussian noise introduction, auto contrast adjustment, median filtering, sharpening, transmission

errors, and error concealment. Note that the *SSMs* were all below 0.5 for acceptable manipulations and all above 0.5 for maliciously attacked images. On the contrary, the Hamming and Figure of Merit (*FoM*) measures of maliciously attacked images were among the range of acceptable manipulations especially the measures of the attacked image in which there was a small local object changed (last column). The results show that the proposed *SSM* was able to distinguish the malicious manipulations from acceptable ones, i.e., identify lossy transmission as acceptable and was sensitive to malicious manipulations. On the contrary, the Hamming and *FoM* measures were not sensitive to small localized object modification. The results indicate that the proposed *SSM* is more suitable for content-based image authentication than Hamming and *FoM* measures.

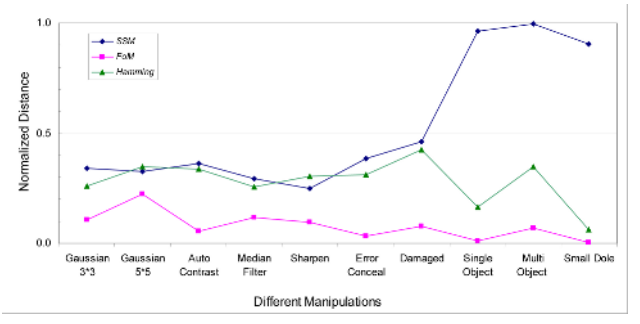


Fig. 4. Comparison of distinguish ability of different distance measures: only the proposed measure can successfully distinguish malicious manipulations from acceptable ones

The transmission errors in wireless networks were simulated based on the *Rayleigh* model [16] which is commonly used for wireless networks with bit error rate (BER) at 10^{-4}. Fig. 5 shows the evaluation results of the system robustness of the proposed error resilient image authentication scheme based on the proposed *SSM* including *PSNR* and *SSM* measures of the images damaged by transmission errors and error-concealed images. 60% of the damaged images in our experiments were verified as

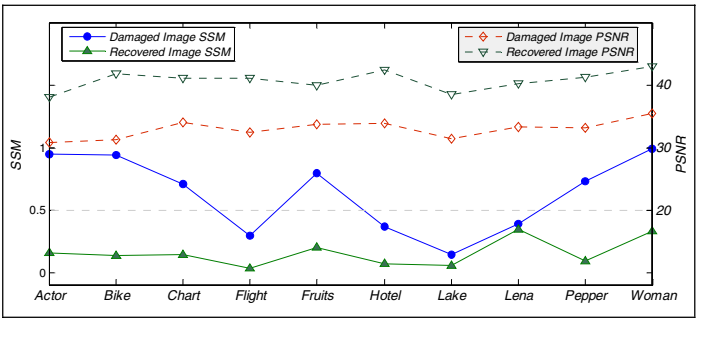


Fig. 5. Authentication performance improvement by error concealment: *PSNR* (dB) and *SSM* of damaged images and error-concealed images

unauthentic and 40% of them were authentic. On the contrary, all error-concealed images were verified as authentic. The results confirm that it is effective and advisable for error concealment to be applied before image authentication. The reason that the authenticities of the recovered images were better than those of the damaged images may be the image quality improvement by using error concealment on the damaged images. For example, the recovered image had much better objective qualities than the damaged images (evaluated by *PSNR*). This quality improvement made features of the error-concealed images closer to those of the original images than damaged images, so that the image authenticities (evaluated by *SSM*) of the error-concealed images were much larger than the damaged images.

Table 1. Robustness against acceptable image manipulations

Manipulations	*Histogram Normalizing*	*Brightness Adjustment*	*Contrast Adjustment*	*JPEG Compression*	*JPEG2000 Compression*
Parameter	Auto	-40	Auto	10:1	1bpp
SSM	0.159	0.159	0.262	0.017	0.057

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

Fig. 6. Detected possible attack locations which are concentrated on objects in images: (a) *Lena* with small spot added (0.916); (b) attacked areas detected of (a); (c) attacked image *Lena* (logo added, flower on cap deleted, and eyes enlarged) (0.983); (d) attacked areas detected of (c)

Our scheme was also tested with other acceptable manipulations such as image contrast adjustment, histogram equalization, compression and noises addition. The results are shown in Table 1, with the parameter for each manipulation. The *SSM* values of these images were all less than 0.5, i.e., all these images can pass the authentication. These results validate that the proposed scheme is not only designed to be robust to transmission errors, but also robust to general acceptable manipulations.

An important aspect of our *SSM*-based authentication scheme is that it is sensitive to the malicious content tampering. We tampered the previous watermarked *Lena* image and tested the ability of our system to detect and highlight the doctoring. All the attacked images were detected and possible attacked areas were located. The examples of the attack location results are shown in Fig. 6. These results indicate that the ability of our system to detect tampering is good even in the presence of very small area modified (Fig. 6a), or multiple tampered areas (Fig. 6c).

4 Conclusions

An error resilient image authentication scheme using statistics and spatiality based measure (*SSM*) was developed, which was robust to transmission errors in JPEG or JPEG200 images. Many acceptable manipulations, which were incorrectly detected as malicious modifications by the previous schemes, were correctly classified by the proposed scheme. These results support the observation that the feature difference patterns under typical acceptable image modifications and malicious ones are distinguishable. These results indicate that the statistical and spatial properties of the image feature are helpful and useful in distinguishing acceptable image manipulations from malicious content modifications. The proposed *SSM* would improve system performance for content-based authentication schemes which use features containing spatial information. Furthermore, the proposed error resilient scheme based on *SSM* can improve the trustworthiness of digital images damaged by transmission errors by providing a way to distinguish them from digital forgeries. A limitation of the proposed measure is that it is suitable only for schemes using features containing spatial information. Further work would be needed to expand the use of the proposed measure by exploiting new discernable patterns in feature differences when the features contain no spatial information.

References

1. V. Erceg and K. Hari, “Channel Models for Fixed Wireless Applications”, *IEEE 802.16 Broadband Wireless Access Working Group*, 2001.
2. P. Golle and N. Modadugu, “Authenticating Streamed Data in the Presence of Random Packet Loss”, In *Proceedings of the Symposium on Network and Distributed Systems Security*, pp.13-22, 2001.
3. A. Perrig, R. Canetti, D. Song, and J. D. Tygar, “Efficient and Secure Source Authentication for Multicast”, in Proceedings of *Network and Distributed System Security Symposium*, 2001, pp.35-46.
4. M.P. Queluz, “Authentication of Digital Images and Video: Generic Models and a New Contribution”, *Signal Processing: Image Communication,* vol.16, pp. 461-475, 2001.

5. C. Y. Lin and S.F. Chang, “A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation”, *IEEE Transaction on Circuits and Systems of Video Technology*, vol.11, pp. 153-168, 2001.
6. Q. Sun and S.F. Chang, “Semi-fragile Image Authentication using Generic Wavelet Domain Features and ECC”, *IEEE International Conference on Image Processing (ICIP)*, Rochester, USA, Sep. 2002.
7. C. W. Wu, “On the Design of Content-Based Multimedia Authentication Systems”, *IEEE Transactions on Multimedia*, Vol. 4, No. 3, pp.385-393, September 2002.
8. M. Schneider and S.F. Chang, “A Robust Content-based Digital Signature for Image Authentication”, in *Proceedings of International Conference on Image Processing (ICIP)*, Vol.3, pp.227 - 230, 1996.
9. B. Li, E. Chang, and Y. Wu, “Discovery of a Perceptual Distance Function for Measuring Image Similarity,” ACM Multimedia Journal Special Issue on Content-based Image Retrieval, vol. 8, no. 6, pp.512-522, 2003.
10. J. Dittmann, A. Steinmetz, and R. Steinmetz, "Content-based Digital Signature for Motion Pictures Authentication and Content-fragile Watermarking", *IEEE International Conference on Multimedia Computing and Systems*, Vol.2, pp.209-213, 1999.
11. Q. Sun, S. Ye, L.Q. Lin, and S.F. Chang, “A Crypto Signature Scheme for Image Authentication over Wireless Channel”, *International Journal of Image and Graphics*, Vol. 5, No. 1, pp.1-14, 2005.
12. E.C. Chang, M.S. Kankanhalli, X. Guan, Z.Y. Huang, and Y.H. Wu, “Robust Image Authentication Using Content-based Compression”, *ACM Multimedia Systems Journal*, Vol. 9, No. 2, pp. 121-130, 2003.
13. S. Ye, Q. Sun, and E.C. Chang, “Edge Directed Filter based Error Concealment for Wavelet-based Images”, *IEEE International Conference on Image Processing*, Singapore, 2004.
14. R. Jain, R. Kasturi and B. G. Schunck, “Machine Vision”, McGraw Hill, New York, 1995.
15. W. Chou, “Classifying Image Pixels into Shaped, Smooth and Textured Points”, *Pattern Recognition*, Vol. 32, No. 10, pp.1697-1706, 1999.
16. S. Ye, X. Lin and Q. Sun, “Content Based Error Detection and Concealment for Image Transmission over Wireless Channel”, *IEEE International Symposium on Circuits and Systems* (*ISCAS*), Thailand, May 2003.
17. M. Boliek (ed.), “JPEG 2000 Final Committee Draft”, *ISO/IEC FCD1.5444-1,* Mar. 2000.
18. Y. Yu and S. T. Acton, "Speckle Reducing Anisotropic Diffusion", *IEEE Transaction on Image Processing*, vol. 11, no. 11, Nov. 2002, pp.1260-1270.

Author Index