Yun Q. Shi
Hyoung-Joong Kim
Stefan Katzenbeisser (Eds.)

LNCS 5041

Digital Watermarking

6th International Workshop, IWDW 2007
Guangzhou, China, December 2007
Proceedings

Springer

Lecture Notes in Computer Science 5041

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

Yun Q. Shi Hyoung-Joong Kim
Stefan Katzenbeisser (Eds.)

Digital Watermarking

6th International Workshop, IWDW 2007
Guangzhou, China, December 3-5, 2007
Proceedings

Springer

Volume Editors

Yun Q. Shi
New Jersey Institute of Technology
University Heights, Newark, NJ, USA
E-mail: shi@njit.edu

Hyoung-Joong Kim
Multimedia Security Lab, Science Campus
Korea University, Seoul, Korea
E-mail: khj@korea.ac.kr

Stefan Katzenbeisser
Philips Research Europe
Information and System Security
Eindhoven, The Netherlands
E-mail: skatzenbeisser@acm.org

Library of Congress Control Number: 2008941293

CR Subject Classification (1998): E.3, D.4.6, I.4.9, I.5.4, K.6.5

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN	0302-9743
ISBN-10	3-540-92237-7 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-92237-7 Springer Berlin Heidelberg New York

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12451173 06/3180 5 4 3 2 1 0

Preface

It is our great pleasure to present in this volume the proceedings of the 6th International Workshop on Digital Watermarking (IWDW), which was held in Guangzhou, China, during December 3–5, 2007. The workshop was hosted by the Sun Yat-sen University and sponsored by both the National Science Foundation of China and the Korea Institute of Information Security and Cryptology.

Since its birth in the early 1990s, digital watermarking has become a mature enabling technology for solving security problems associated with multimedia distribution schemes. Digital watermarks are now used in applications such as broadcast monitoring, movie fingerprinting, digital rights management, and document authentication, to name but a few. Still, many research challenges remain open, among them security and robustness issues, reversibility and authentication. Continuing the tradition of previous workshops, IWDW 2007 also featured—besides papers dealing with digital watermarking—contributions from other related fields, such as steganography, steganalysis and digital forensics.

The selection of the program was a challenging task. From 81 submissions (received from authors in 17 different countries) the Program Committee selected 27 as regular papers and 7 as posters. At this point we would like to thank all the authors who submitted their latest research results to IWDW and all members of the Program Committee who put significant effort into the review process, assuring a balanced program. In addition to the contributed papers, the workshop featured three invited lectures delivered by Bede Liu, Hany Farid and Ahmad-Reza Sadeghi; summaries of their lectures can be found in this proceedings volume.

We hope that you will enjoy reading this volume and that it will be a catalyst for further research in this exciting area.

Y.Q. Shi
H.-J. Kim
S. Katzenbeisser

Organization

Hosted by Sun Yat-sen University.
Sponsored by the National Science Foundation of China (NSFC) and the Korea Institute of Information Security and Cryptology (KIISC).

General Chairs

Huang, Jiwu	Sun Yat-sen University, China
Rhee, Min Surp	KIISC, Korea

Technical Program Chairs

Shi, Yun-Qing	New Jersey Institute of Technology, USA
Kim, Hyoung-Joong	Korea University, Korea
Katzenbeisser, Stefan	Philips, The Netherlands

Technical Program Committee

A. Piva	University of Florence, Italy
Sadeghi, Ahmad-Reza	University of Bochum, Germany
Kot, Alex	NTU, Singapore
Ho, Anthony TS	University of Surrey, UK
Macq, Benoit	UCL, Belgium
Jeon, Byeungwoo	SKKU, Korea
Chang, C.C	Feng-Chia University, Taiwan
Kuo, C.C. Jay	USC, USA
Zou, Dekun	Thomson, USA
Perez-Gonzalez, Fernando	University of Vigo, Spain
Xuan, Guorong	Tongji University, China
Lee, Heung-Kyu	KAIST, Korea
Noda, Hideki	Kyushu Institute of Tech, Japan
Lagendijk, Inald	Delft University of Technology, The Netherlands
Pitas, Ioannis	University of Thessaloniki, Greece
Dittman, Jana	University of Magdeburg, Germany
Dugelay, Jean-Luc	Eurecom, France
Bloom, Jeffrey	Thomson, USA
Pan, Jeng-Shyang	NKUAS, Taiwan
K. Sakurai	Kyushu University, Japan
Mihcak, Kivanc	Bogazici University, Turkey
Miller, Matt	NEC, USA
Barni, Mauro	University of Siena, Italy
Wu, Min	University of Maryland, USA
Goljan, Miroslav	SUNY Binghamton, USA

Kankanhalli, Mohan	NUS, Singapore
Memon, Nasir	Polytechnic University, USA
Sun, Qibin	Institute of Infocomm Research, Singapore
Voloshynovskiy, Sviatoslav	University of Geneva, Switzerland
Furon, Teddy	IRISA, France
Kalker, Ton	HP, USA
Ro, Yong-Man	ICU, Korea
Lu, Zheming	Sun Yat-sen University, China
Ni, Zhicheng	WorldGate Communications, USA

Organization Committee Chair

Liu, Hongmei	Sun Yat-sen University, China

Reviewer List

Au, Oscar
Barni, Mauro
Bloom, Jeffrey
Chang, C.C.
Chen, Chunhua
Chen, Wen
Choi, Su-Jeong
Cvejic, N.
Dittman, Jana
Dugelay, Jean-Luc
Fridrich, Jessica
Furon, Teddy
Goljan, Miroslav
Hagmüller, Martin
Ho, Anthony
Ho, Yo-Sung
Jeon, Byeungwoo
Kalker, Ton
Kang, Xiangui
Kankanhalli, Mohan
Katzenbeisser, Stefan
Kim, Hae Kwang
Kim, Hyung-Joong
Kuo, C.C. Jay
Lagendijk, Inald
Lee, Heung-Kyu
Lee, Kwangsoo
Li, Bin
Li, Wei
Liu, Hongmei
Lu, Zheming
Macq, Benoit
Maitra, Subhamoy
Memon, Nasir
Milller, Matt
Nam, Jeho
Ng, Tian-Tsong
Ni, Zhicheng
Noda, Hideki
Pan, Jeng Shyang
Perez-Gonzalez, Fernando
Piva, A.
Puech, William
Qiu, Guoping
Ro, Yong-Man
Sachnev, Vasiliy
Sadeghi, Ahmad-Reza
Sallee, Phil
Sun, Qibin
Tan, Shunquan
Tian, Guangsen
Tian, Jun
Tsai, Wen-Hsiang
Voloshynovskiy, Sviatoslav
Wang, Daoshun
Westfeld, Andreas
Won, Chee Sun
Wu, Min
Xiang, Shijun
Xuan, Guorong
Zhang, Xinpeng
Zou, Dekun

Table of Contents

Invited Lecture

Session I: Watermark Security

Session II: Steganalysis

Session III: Authentication

Session IV: Reversible Data Hiding

Session V: Robust Watermarking

Session VI: Poster Session

Session VII: Theory and Methods in Watermarking

Watermarking, a Mature Technology – Retrospect and Prospect

Bede Liu

Princeton University, USA
liu@Princeton.EDU

Abstract. Digital Watermarking has been proposed to manage digital rights, for authentication, to recover lost information, to monitor performance, and other applications. The efforts of many researchers from different disciplines have helped us to understand the basic issues and the challenges, and to help guiding the community toward actual applications. As digital watermarking matures as a technology, it is time to review what has been accomplished and to speculate what may be expected. In this talk, we will try to highlight the key development in digital watermarking, to examine the effectiveness in some applications, and to offer some thoughts regarding future development of digital watermarking.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, p. 1, 2008.

The Marriage of Cryptography and Watermarking — Beneficial and Challenging for Secure Watermarking and Detection

Ahmad-Reza Sadeghi

Horst Görtz Institute for IT Security
Ruhr-University Bochum, Germany
sadeghi@crypto.rub.de

Abstract. Multimedia applications deploy various cryptographic and watermarking techniques to maintain security. In this context, we survey the main work on two promising approaches for the secure embedding and detection of a watermark in an untrusted environment, and we point out some associated challenges.

In the former case we consider Zero-Knowledge Watermark Detection (ZKWMD) that allows a legitimate party to prove to a potentially untrusted verifying party that a watermark is detectable in certain content, without jeopardizing the security of the watermark. ZKWMD protocols are useful primitives for direct proofs of authorship (i.e., without online involvement of a trusted third party) and dispute resolving in distributed systems. In the latter case we consider a Chameleon-like stream cipher that achieves simultaneous decryption and fingerprinting of data, and can serve as the second line of defense for tracing illegal distribution of broadcast messages, termed as Fingercasting.

1 Motivation

Copyright protection is a significant prerequisite for intellectual achievements and the creation of original works. In addition to the legal framework that has protected authors since the early ages of book printing, technical measures have become paramount in the age of information technology.

Today, various technologies are combined to maintain multimedia security. Core technologies in this context are digital watermarking and cryptography. Many proposals for protecting digital works against misuse and illegal distribution apply robust watermarking methods as basic building blocks. Robust watermarking methods embed additional information, called watermark, into digital content in such a way that this information can be detected, even after the content has been manipulated. Ideally, the robustness property of the watermarking method should guarantee that the watermark cannot be removed without destroying the digital content. Moreover, cryptographic methods, primitives, and protocols are deployed for conditional access allowing the protection of digital content and the restriction of access to legitimate users.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 2–18, 2008.

In this paper we focus on the main research work on two promising approaches that combine cryptographic schemes with watermarking schemes, namely *Zero-Knowledge Watermark Detection* (ZKWMD) and *Chameleon-like* stream ciphers. Zero-Knowledge Watermark Detection concerns the problem of proving the presence of a watermark embedded into the underlying content to a potentially dishonest party without revealing sensitive information about the watermark. This approach has applications in direct proof of authorship and dispute resolving. Chameleon-like stream ciphers aim at jointly decrypting and watermarking the underlying content so that the plaintext (non-watermarked) content is never revealed. They can serve as a second line of defense in applications where a sender broadcasts the encrypted content to a large number of receiver devices. The resulting scheme is called *Fingercasting*.

In the following, we will consider some of the main work that has been done in both areas and discuss some challenges and future work.

2 Zero-Knowledge Watermark Detection

2.1 Motivation

In a typical application scenario, at some point in time, a watermark, carrying certain application-related information, is embedded into digital content. Later, a *prover* has to prove to a verifying party, the *verifier*, that the watermark is present in some, possibly modified, watermarked version of the original content. As a simple example consider a *dispute resolving scheme*, where an author embeds a watermark in his work before publication. If an authorship dispute arises, the author (being one of the disputants) has to prove to the dispute resolving party that his watermark is detectable in the disputed work [9].

Besides the limited robustness of watermarking methods, another problem arises from their *symmetric* nature: the watermark detection algorithm requires knowledge of the watermark and the same key that has been used in the embedding process. Proving the presence of such a watermark is usually done by revealing the required detection information (watermark and key) to the verifying party. However, at the same time, it allows the verifier to remove the watermark and, consequently, to undermine the security of the application.

This inherent need to give away critical information strongly limits the usability of symmetric watermarking schemes, since most applications at some point require the detection of a watermark by a — in reality — not fully trusted party. To tackle this problem, a variety of approaches were proposed: *Asymmetric watermarking* uses different keys for watermark embedding and detection. In such schemes, knowledge of the public detection key must not enable an adversary to remove the embedded watermark. There have been some proposals for asymmetric watermarking schemes based on different approaches such as properties of Legendre sequences [62], "one-way signal processing" techniques [37], or Eigenvectors of linear transformations [31]. Unfortunately, none of these schemes is sufficiently robust against malicious parties [30]. Furthermore, oracle or sensitivity attacks[1] pose a special threat to asymmetric watermarking schemes, because knowledge of the public detection key gives an attacker unlimited access to a detection oracle.

Another solution of this problem is known as *Zero-Knowledge Watermark Detection* (ZKWMD), which replaces the watermark detection process by a cryptographic protocol. The basic idea is to conceal the required detection input and to apply cryptographic techniques to perform detection on the concealed input, ideally, without disclosing any information. Thus, zero-knowledge watermark detection can improve the security of many applications, which rely on symmetric watermarking schemes.

2.2 Background: Interactive Proof Systems

Interactive proof systems are widely used as basic primitives in cryptographic systems. Since their introduction by Goldwasser, Micali and Rackoff [40], there has been a huge amount of research results and variants of proof systems. Here, we only informally summarize the most fundamental concepts relevant for this paper. For a detailed introduction and comprehensive theoretical treatise we refer the interested reader to [38].

An *interactive proof system* is a two-party protocol between two probabilistic interactive algorithms, a *prover* $\mathcal{P}$ and a *verifier* $\mathcal{V}$. The task of $\mathcal{P}$ is to prove some claim, represented by the common input x (public value known to all parties), to $\mathcal{V}$. Each party $\mathrm{X} \in \{\mathcal{P}, \mathcal{V}\}$ may have private inputs called *auxiliary input* aux_{X}. Further, each party has access to an exclusive[2] source of randomness, i.e., a random tape initialized with a random string $\mathrm{rand}_{\mathrm{X}}$. The fundamental properties of an interactive proof system are *completeness* and *soundness*. The former requires that a correct prover $\mathcal{P}$ can prove any correct statement to a correct verifier $\mathcal{V}$ while the latter requires that a cheating prover $\mathcal{P}^*$ cannot prove a wrong statement to an honest verifier.[3]

Proof systems can be categorized according to the general type of statements that can be proven: in a *proof of language membership* for a language L, a string x is given as common input to both parties. The objective of $\mathcal{P}$ is to convince $\mathcal{V}$ that x is a word of language L. In a *proof of knowledge* (POK) for a binary relation R, a string x is given as common input to both parties. Here, however, the objective of $\mathcal{P}$ is to prove to $\mathcal{V}$ that it *knows* a string w, called *witness*, for which $(x, w) \in R$ holds. In a proof of knowledge, the prover proves to the verifier that he has knowledge of "something", e.g., the solution of a certain hard cryptographic problem (e.g., the factorization of a very large integer). The simplest

[1] In an oracle attack (also known as sensitivity attack) [23, 49] the adversary utilizes a detector to iteratively modify the watermarked data until the watermark cannot be detected anymore.

[2] There are further types of interactive proof systems, where $\mathcal{P}$ and $\mathcal{V}$ have access to an additional *joint* source of randomness, referred to as the *common reference string.*

[3] The proof is not always absolute but rather probabilistic, i.e., there may be a tolerated success probability for a cheating prover.

way to prove this knowledge is to reveal it to the verifier. However, this is not desirable in many applications, since knowledge of a certain secret should be proven *without* disclosing any (partial) information about it. In cryptography, such straightforward proof systems are of little interest, as they are usually used to convince some *untrusted* party. Therefore, most cryptographic applications require (interactive) proof systems to reveal "no knowledge" to the verifier, except the fact that the proof's statement is valid. This additional requirement is known as the *zero-knowledge property* of interactive proof systems.[4] Interactive proof systems, revealing no knowledge except the validity of the proof's assertion, are called *zero-knowledge* proof systems. The additional zero-knowledge requirement makes the design of interactive proof systems more involved, and at first sight, it may not seem to be possible at all.[5]

Most (basic) interactive proof systems we are concerned with are of a *challenge and response* form: given the common input, the protocol consists of three moves. The prover $\mathcal{P}$ starts by sending a random variant of the common input to the verifier $\mathcal{V}$. Then $\mathcal{V}$ sends a random *challenge* to $\mathcal{P}$, and in the last move $\mathcal{P}$ answers the challenge with a corresponding *response*. The response matching the common input, the commitment, and the challenge are computed by means of $\mathcal{P}$'s *private auxiliary input* $aux_{\mathcal{P}}$. Finally, $\mathcal{V}$ verifies this response and, based on this verification, accepts or rejects the proof.

2.3 Informal Characterization of ZKWMD

The basic idea of Zero-Knowledge Watermark Detection (ZKWMD) is to conceal the input, which is required for detection, and to apply cryptographic techniques to prove that the detection criterion holds on the concealed input without disclosing any additional information. For this purpose *zero-knowledge proof systems* can be applied. These protocols should ideally fulfill the following requirements:

– *Hiding secret information:* The input required by the ZKWMD protocol does not reveal any additional information about the sensitive data (the watermark, the detection key, and the reference data).
– *Zero-knowledge:* A run of the protocol does not disclose any *new* information. Here, "new information" refers to information *beyond* the predicate (language membership or the knowledge relation imply a positive detection result) proven by the proof system and the information already leaked by the protocol input. Note, that knowledge of the detection result, although being no new knowledge, may pave the way for oracle attacks. This is no contradiction to the zero-knowledge property of the protocol because the

[4] Note that knowledge about the validity of an assertion, as given by an accepted run of the interactive proof (soundness), represents no new knowledge to the verifier. This is because the zero-knowledge property is only defined for *honest* provers that only prove correct statements by definition.

[5] A well-known result by Goldwasser, Micali and Wigderson [39] shows that, assuming the existence of one-way functions, zero-knowledge proof systems exist for every NPlanguage.

zero-knowledge property is required only for honest provers, which initiate the ZKWMD protocol only if the watermark is really detectable.

- *Soundness:* A dishonest prover should not be able to wrongly convince the verifying (detecting) party that the watermark, concealed in the input, is detectable in the given data.

Given a *fixed and concealed* watermark WM, cover-data W, secret watermarking key k_{wm}, and a given (possibly modified) stego-data W'', a prover can convince a verifier that the watermark is detectable in W'' relative to W under the (detection) key k_{wm} *without revealing any additional information about security critical quantities*, i.e., the watermark, watermarking key, or the original cover-data. More concretely, we consider a Zero-Knowledge Watermark Detection protocol for a certain watermarking scheme as an interactive proof system, where the following holds: The common input x consists of the stego-data W'' and *encodings* of WM, W, and k_{wm}. Note that the encoding aims at hiding secret information about the detection input and, therefore, its hiding property is a crucial issue of Zero-Knowledge Watermark Detection. The prover has additional private auxiliary input *aux*, which, in general, is some secret information about the common input x. The proof system is either a proof of language membership, such that membership of x in language L implies that the encoded watermark is detectable in the stego-data W'' by using the encoded detection key and the encoded cover-data. The other possibility is, that the proof system is a proof of knowledge such that the prover's knowledge $aux_{\mathcal{P}}$ about $x \in L_R$ implies that the encoded watermark is detectable in the stego-data W''. The proof protocol itself does not disclose additional knowledge, i.e., the proof protocol is (auxiliary-input) zero-knowledge.

2.4 Related Work

In this section we review some of the main literature on secure watermark detection that has been proposed so far. The idea of incorporating a symmetric watermark detection process into a cryptographic protocol was first introduced by Craver in [24] and Gopalakrishnan, Memon and Vora in [41]. Since then, a large number of proposals followed [4, 8, 25, 27, 51, 52, 63].

One of the first proposals relies on a blinding process induced by a *secret permutation* of the watermark [24, 25, 26], which serves as the underlying encoding. A generalized version of the scheme by Craver and Katzenbeisser has been introduced in [4]. The idea of Craver [24, 25] was to conceal the detection inputs by applying a permutation τ whereby the watermarking system allows a permuted watermark to be detected in equally permuted data, and, optionally, with an equally permuted key. Although the protocol can be proven to fulfill the zero-knowledge property in the case that the prover performs the protocol only on unmodified watermarked data W', it is susceptible to an oracle attack. If a dishonest verifier can issue adaptively modified stego-data W'' to the prover, he can recover τ after several invocations of the protocol.

It is possible to construct a detection protocol that relies on the possibility of performing an *ambiguity attack* [24]. Such attacks compute a watermark, which has never been embedded into a digital object W', but nevertheless can be detected therein. A drawback of this scheme is that it does not even encrypt the watermarks, but only hides the plaintext watermark in a list of "fake" watermarks.[6] The hiding requirement is only weakly fulfilled by this protocol, because it is not straightforward to make the fake watermarks sufficiently indistinguishable from the real watermark and, because it would be even impractical to achieve a reasonable security level.

A further protocol for watermark detection has been proposed by Gopalakrishnan, Memon and Vora [41] as a solution to the so called *watermarking decision problem*: Given certain stego-data decide whether an (RSA) encrypted watermark is present in this stego-data. The authors propose a multi-round challenge-response protocol for solving this problem for the blind version of the well-known watermarking scheme of Cox et al. [20]. However, no formal proof of soundness has been given for this protocol. It is not zero-knowledge since the verifier obtains a good estimation of the correlation value. Assuming that the common input does not leak the watermark, the latter contradicts the formal definition of zero-knowledge proof systems, and allows a dishonest verifier to mount more efficient sensitivity attacks in practice. Finally, non-probabilistic RSA encryption does not provide a good level of secrecy.

Yongliang et al. [51] propose a protocol, which combines the permutation-based approach of Craver and Katzenbeisser [25] with the approach of Adelsbach and Sadeghi [8]. However, the authors propose to run this protocol on permuted vectors to prevent leakage of the watermark location. There are several issues with this: first, permutation of the cover- and stego-data provides no sufficient secrecy and, second, the protocol does not include measures to guarantee that cover-data and stego-data are permuted consistently using the same permutation. The latter jeopardizes soundness of the protocol.

Zhao et al. [63] generalize the idea of Craver [24, 25] and define a secure equivalent operation based on adding noise (blinding) and permuting vectors. Based thereon, the authors construct a protocol and provide some formal proof that this protocol fulfills the zero-knowledge property. However, as the verifier learns the exact correlation value between the secret watermark and the stego-data, the protocol cannot be zero-knowledge. It also seems that, in their security proof, the authors falsely let the simulator choose the watermark and generate the cover-data therefrom. As the cover-data is part of the common input but the real plaintext watermark is not, and the fact that the verifier learns the correct correlation value (between the cover-data and the real watermark), this simulation cannot be indistinguishable from a true protocol run as the simulator does not know the correct correlation value.

Craver, Liu and Wolf [27] proposed an extension of Craver's protocol employing ambiguity attacks. In this recent proposal the authors aim to counter oracle attacks by embedding several ($l > 1$) legal watermarks and hiding them in a list of f fake watermarks. However, the same oracle attack still works by running

[6] Just imagine hiding your love letters in a box with other letters — obviously not a good choice.

the protocol $l+f$ times, each time deleting all but one watermark. Moreover, the authors propose a concrete method to generate the fake watermarks by decomposing the original image. To make these fake watermarks indistinguishable from legal watermarks, the fake watermarks should be whitened. To achieve this, the authors propose to permute the corresponding coefficients. However, the detection of permuted watermarks requires the stego-data W' to be permuted in the same way, which leaks information about the whitening permutation. This may allow an attacker to revert the whitening process and, thereby, to distinguish fake from legal watermarks very efficiently.

Recently, Yongliang et al. [52] proposed a minor variation of the protocol by Gopalakrishnan, Memon and Vora [41]. The differences are that the Rabin encryption scheme is used instead of RSA such that the coefficients are blinded with a value $\beta \cdot \gamma$, where β is a random number chosen by the prover and γ is a random number received from a trusted party. However, the resulting blinding is even weaker than the blinding proposed by Gopalakrishnan et al., since the same blinding factor is re-used for all coefficients. Furthermore, similar to the original protocol, this modified version leaks the correlation value to the verifier and therefore, the protocol is not zero-knowledge.

2.5 Strong Zero-Knowledge Watermark Detection Protocols

In this section we introduce strong ZKWMD protocols, which provide the following distinguishing features: (i) secret detection inputs are strongly hidden by means of commitment schemes (see below), and (ii) the actual protocol, proving the presence of the hidden watermark, is zero-knowledge. Strong ZKWMD have been proposed in [6, 7] for both, blind and non-blind normalized correlation-based detection statistics on securely encoded (committed) detection inputs using basic zero-knowledge proofs that prove basic arithmetic relations on committed integers. As such, they depend on the concrete detection statistic of the corresponding watermarking scheme. However, as stressed by Cox, Miller and Bloom [21], detection of most watermarking schemes is equivalent to some correlation-based detection statistic.[7]

Therefore, ZKWMD protocols are applicable to a large class of watermarking schemes. Furthermore, these protocols can be considered as representatives of a *generally applicable design methodology* for strong zero-knowledge watermark detection protocols: as most detection statistics can be computed by means of basic operations (such as $+, -, \cdot, /, exp$), for which efficient zero-knowledge subproofs exist, the corresponding zero-knowledge watermark detection protocols

[7] As an example assume the detection of a watermark vector WM in stego-data W'' based on the normalized correlation corr = $\text{CorrNorm}(m_{W''}, WM) = \frac{\sum_{i=1}^{n} m_{W''}[i] \cdot WM[i]}{|m_{W''}| \cdot |WM|}$ between the marking space vector $m_{W''}$ and the watermark vector WM where $|x|$ denotes the length of the vector x. The value CorrNorm is a measure of confidence for the presence of WM in the marking space vector $m_{W''}$, or the corresponding stego-data W'' respectively. The watermark is decided to be present in W'' if corr $\geq \delta$ for a predefined *detection threshold* δ. Usually one assumes δ to be a public parameter.

can be assembled. For encoding the common input to the ZKWMD protocol, *commitment schemes* are used. In the following we briefly explain some of the main cryptographic building blocks used to realize strong ZKWMD.

- **Commitment Schemes.** They are fundamental building blocks of cryptographic protocols and operate in two phases. A commitment scheme enables a party, *committer*, to commit itself to a message (*committing phase*), such that it cannot be changed after committing, while the committed value is hidden from the party who receives the commitment (*receiver*) [15, 28]. In the *opening phase*, the committer can reveal the committed message to the receiver, and typically the secret opening keys. The commitment scheme guarantees that a commitment can only be opened to the message that has actually been committed to. The security requirements are the *binding* (committing) property and the *hiding* (secrecy) property. Informally, the binding property requires that a dishonest committer cannot open a commitment in *two* different ways, and the hiding property requires that a commitment does not reveal *any* information about its content to the receiver. For concrete protocols we require commitment schemes with *homomorphic property.* An appropriate choice for a *homomorphic integer commitment scheme*[8] is the commitment scheme proposed by Damgård and Fujisaki (DF commitment) [29].[9] The DF commitment scheme is both, a modification and generalization of a commitment scheme previously proposed by Fujisaki and Okamoto [35, 36].
- **Zero-Knowledge Proofs for Relations between Committed Numbers.** In concrete ZKWMD protocols, such as [6, 7], several elementary zero-knowledge proof of knowledge protocols (sub-proofs) are applied. The most common ones are the following:

  Proving Knowledge of Opening Information is a basic zero-knowledge proof of knowledge, which proves that the prover knows how to open a certain commitment. For the commitment scheme of Damgård and Fujisaki such a protocol was introduced in [29]. *Proving Multiplication-Relation for Committed Numbers* is the next elementary zero-knowledge proof of knowledge, which, given three commitments c_a, c_b and c_c, allows the prover to prove that these commitments "contain" messages m_a, m_b and m_c such that $m_c = m_a \cdot m_b$ holds. *Proving the Square Relation* allows to prove in zero-knowledge, for two given commitments c_x, c_y as common input, the knowledge of the corresponding opening information to open these commitments to messages m_x, m_y such that $m_x = (m_y)^2$. This protocol has been introduced by Boudot [14] for Fujisaki-Okamoto commitments, but, according to [29], the protocol also works for DF commitments because they have the same structure. *Proving Equality* of committed messages was proposed by Boudot [14] for

[8] Integer commitment scheme means that the commitment scheme allows to commit to *integers of arbitrary size.*

[9] It is *statistically hiding* and *computationally binding under the (generalized) root assumption.*

Fujisaki-Okamoto commitments. According to [29], the same protocol also works for DF commitments. The protocol convinces the verifier that the prover can open two commitments c_x and c_y, given as common input, to the same message m. *Proving the "Greater or Equal Zero" Relation* is a protocol, which, on common input c_x, proves knowledge of opening information that opens c_x to some value $m_x \geq 0$. Recently, Lipmaa [50] proposed a highly optimized zero-knowledge proof of knowledge for the "greater or equal zero" relation.

2.6 A Note on Application of ZKWMD

ZKWMD protocols are "zero-knowledge proof *equivalents*" of the corresponding symmetric watermarking schemes. They have an equivalent input/output behavior, while hiding any information besides the detection result. Therefore, the ZKWMD faces the same limitations as the underlying watermarking scheme. This includes (i) robustness issues of the underlying symmetric watermarking scheme, (ii) susceptibility to ambiguity attacks, and (iii) susceptibility to oracle/sensitivity attacks.

Robustness issues are inherent to the underlying symmetric watermarking scheme and addressing them mostly requires a complete re-engineering of the watermarking scheme itself. Oracle attacks require a certain number of detector outputs on adaptively modified stego-data. As each run of the zero-knowledge watermark detection protocol requires active participation of the prover and the attacker, who only receives a binary detection output instead of the exact correlation value itself, oracle attacks are harder to perform for ZKWMD protocols.

What remains to be considered are measures for countering ambiguity attacks and how to apply these countermeasures to committed watermarks in the context of zero-knowledge watermark detection. For a more complete treatment of this topic, including detailed discussion of practical countermeasures, we refer the interested reader to [6, 7].

3 Fingercasting

3.1 Background

Encryption schemes are an essential component of many practical conditional access mechanisms. In this context Broadcast Encryption (BE) is a special type of encryption scheme that addresses a well-defined use case in a secure and efficient manner, namely secure distribution of digital content. More concretely a single sender distributes digital content to a large audience over a unidirectional broadcast channel such as a satellite transmission. A broadcast encryption scheme defines the secret information that the sender and the legitimate receivers obtain for encryption and decryption purposes.

There are various ways to attack Broadcast Encryption (BE) systems: Attacks on the hardware that stores cryptographic keys, e.g., to extract keys from a compliant device in order to develop a pirate device such as the DeCSS software that circumvents the Content Scrambling System [61], or alternatively, attacks on the decrypted content, e.g., when a legitimate user shares decrypted content with illegitimate users on a file sharing system such as Napster, Kazaa, or BitTorrent. Basically one can consider the following security requirements of a broadcasting sender: *confidentiality* and *traceability* of content and keys, and *renewability* of the encryption scheme. Confidentiality prevents illegal copies in the first place, whereas traceability is a second line of defense aimed at finding the origin of an illegal copy (content or key). The need for traceability originates from the fact that confidentiality may be compromised in rare cases, e.g., when some (legitimate) users illegally distribute their secret keys. Renewability ensures that after such rare events, the encryption system can recover from the security breach and continue operation.

In broadcasting systems deployed today, e.g., Content Protection for Pre-Recorded Media [1] or the Advanced Access Content System [2], confidentiality and renewability often rely on BE because it provides low transmission overhead while at the same time having a realistic receiver key size and an acceptable computational overhead. Traitor tracing enables the traceability of keys, whereas fingerprinting provides the traceability of content. Finally, renewability may be achieved by revoking the leaked keys. However, none of the mentioned cryptographic schemes covers all three security requirements. Some existing BE schemes do not provide the traceability of keys, whereas no practically relevant scheme provides traceability of content [34, 42, 44, 56]. Traitor tracing schemes only allow tracing of keys, but not of the content. In addition, the keys of traitors cannot be revoked [19, 57]. On the other hand fingerprinting schemes alone do not provide confidentiality [48]. The original Chameleon cipher [12] provides confidentiality, traceability, and a hint on renewability, but with a small constant bound for collusion resistance and, most importantly, without proof of security. Asymmetric schemes, which provide a certificate to each compliant device and accompany content with Certificate Revocation Lists (CRLs), lack in content traceability and may reach the limits of renewability when CRLs become too large to be processed by real-world devices. Finally, a trivial combination of fingerprinting and encryption leads to an unacceptable transmission overhead in the broadcast context because the broadcasting sender needs to sequentially transmit each fingerprinted copy.

3.2 Related Work

The original Chameleon cipher proposed by Anderson and Manifavas is 3-collusion-resistant [12]. A collusion of up to three malicious users has a negligible chance of creating a good content copy that does not incriminate them. Each legitimate user knows the seed of a Pseudo-Random Sequence (PRS) and a long table filled with random keywords. Based on the sender's master table, each receiver obtains a slightly different copy of the table where individual bits in the keywords are modified in a characteristic way. Interpreting the PRS as a sequence of addresses in the table, the sender adds the corresponding keywords in the master table bitwise modulo 2 in order to mask the plaintext word. The receiver applies the same operation to the ciphertext using its table copy, thus embedding the fingerprint.

The original cipher, however, has some inconveniences. Most importantly, it has no formal security analysis and bounds of the collusion resistance are fixed by the constant number 3. In addition, the original scheme limits the content space (and keywords) to strings with characteristic bit positions that may be modified without perceptibly altering the content.

The original Chameleon cipher was inspired by work from Maurer [54, 55]. His cipher achieves information-theoretical security in the bounded storage model with high probability. In contrast, the Chameleon scheme only achieves computational security. The reason is that the master table length in Maurer's cipher is super-polynomial. As any adversary would need to store most of the table to validate guesses, the bounded storage capacity defeats all attacks with high probability. However, Maurer's cipher was never intended to provide content traceability or renewability, but only confidentiality.

Ferguson et al. [33] discovered security weaknesses in a randomized stream cipher similar to Chameleon.

Ergun, Kilian, and Kumar proved that an averaging attack[10] with additional Gaussian noise defeats any watermarking scheme [32]. Their bound on the minimum number of different content copies needed for the attack asymptotically coincides with the bound on the maximum number of different content copies to which the watermarking scheme of Kilian et al. [47] is collusion-resistant.

Recently there has been a great deal of interest in joint fingerprinting and decryption [16, 17, 48, 53, 58]. Basically, we can distinguish three strands of work. The first strand of work applies to the original Chameleon cipher in different application settings. Briscoe et al. [16] introduce *Nark*, which is an application of this cipher in the context of Internet multicast. However, they neither enhance the original cipher nor analyze its security. The second strand of work tries to achieve joint fingerprinting and decryption by either trusting network nodes to embed fingerprints (Watercasting in [17]) or doubling the size of the ciphertext by sending differently fingerprinted packets of content [58]. The third strand of work proposes new joint fingerprinting and decryption processes [48, 53] but at the price of replacing encryption with scrambling, which does not achieve ciphertext indistinguishably and has security concerns.

Recently, in [3] the authors presented for the first time a security proof for Chameleon ciphers, thus providing an appropriate foundation for the recent applications of these ciphers, e.g., [16]. Furthermore, they give an explicit criterion to judge the security of the Chameleon cipher's key table. It is a combination of (i) a new Chameleon cipher based on the *finger*printing capabilities (collusion-resistance and frame-proofness) of a well-known class of watermarking schemes [22, 47] and (ii) an arbitrary broad*cast* encryption scheme, which explains the name *Fingercasting* of their approach. The basic idea is to use the Chameleon cipher for combining decryption and fingerprinting. To achieve renewability, they use a BE scheme with revocation capabilities in order to provide fresh session

[10] Computing the average of given copies of the fingerprinted data.

keys as input to the Chameleon cipher. To achieve traceability, the receivers' key tables are fingerprinted such that they embed a fingerprint into the content during decryption. To enable higher collusion resistance than the original Chameleon cipher, the scheme is tailored to emulate any watermarking scheme whose coefficients follow a probability distribution that can be disaggregated into additive components. As one can emulate a Spread Spectrum Watermarking Scheme [47] in this Fingercasting approach, its collusion resistance is — at least asymptotically — the best one can hope for. The authors mentioned that one might as well instantiate it with any other watermarking scheme with additive components.

In [18], Celik et al. have proposed a blind detection algorithm for their new Chameleon cipher that emulates Spread Spectrum Watermarking. However, they significantly change the properties of the master table as apposed to [3] by switching from entries with uniform distribution to entries with normal distribution and by using real numbers instead of elements of a group. Nevertheless, they do not analyze the impact of these changes on the cipher's security. Hence, the analysis of chosen ciphertext security in [3] may not apply to their cipher.

However, the proposed solution in [3] does not apply a special collusion-resistant code, but derives a limited resistance against collusion attacks from the underlying Spread Spectrum Watermark. In [46] the authors make the first step towards tackling this problem by proposing a construction that provides collusion-resistance against a large coalition in a secure watermark embedding setting. For this they incorporate a variant of the collusion resistant random code of Tardos [13, 60], which currently is the code with the best asymptotic behavior, into a Fingercasting framework. Through statistical analysis they show that the combination is feasible but for a small subset of possible Fingercasting system parameters.

Joint decryption and fingerprinting has significant advantages compared to existing methods such as transmitter-side or receiver-side Fingerprint Embedding (FE) [48]. Transmitter-side FE is the trivial combination of fingerprinting and encryption by the sender. The transmission overhead is in the order of the number of copies to be distributed, which is prohibitive in broadcast applications. Receiver-side FE happens in the user's receiver. After the distribution of a single encrypted copy of the content, a secure receiver based on tamper-resistant hardware is trusted to embed the fingerprint *after* decryption. This saves bandwidth on the broadcast channel. However, perfect tamper-resistance cannot be achieved under realistic assumptions [10, 11]. An adversary may succeed in extracting the keys of a receiver and subsequently decrypt without embedding a fingerprint.

3.3 High-Level Overview of Fingercasting Scheme from [3]

We give a high-level overview of the scheme proposed in [3]. To fingercast content, the center uses the BE scheme to send a fresh and random session key to each non-revoked receiver. The session key serves as the source of randomness of the Chameleon encryption algorithm. It is used as the seed of a Pseudo-random

Sequence Generator (PRSG) generating a pseudo-random sequence, which represents a sequence of addresses in the master table of the Chameleon cipher. The center encrypts the content with the master table entries to which the addresses refer. Each receiver has a unique receiver table that differs only slightly from the master table. During decryption, these slight differences in the receiver table lead to slight, but characteristic differences in the content copy (i.e., fingerprints).

4 Conclusion and Future Work

In this paper we surveyed two promising technologies for secure fingerprinting of broadcast messages (Fingercasting) and secure watermark detection (Zero-Knowledge Watermark Detection).

These approaches, and multimedia applications in general, deploy various cryptographic and watermarking techniques to protect digital data. The design, security analysis and security proof of such protocols requires a suitable formal framework and reasonable security definitions, in particular of watermarking schemes (as, e.g., aimed in [45] and [5]). Modern cryptography already uses established formal models and definitions for information-theoretic and computational security. More investigations are needed with regard to watermarking schemes, and those schemes which compose watermarking and cryptographic techniques to cover the subtle aspects essential for reasonable formal security definitions, analysis and abstraction of watermarking schemes.

Another strand of future work is to apply the zero-knowledge watermark detection (ZKWMD) framework to further watermarking schemes having better robustness or resistance to the computation of false-positives. As mentioned before, most known watermarking schemes use correlation-based detection and consequently the corresponding zero-knowledge watermark detection protocols can be constructed. Moreover, new advancement in cryptographic research such as more efficient proof protocols and techniques would result in more efficient zero-knowledge watermark detection. Further, investigating ZKWMD for a border range of applications is desirable.

Another interesting lone of research is investigating solutions for Fingercasting based on more general fingerprinting schemes with a more flexible (lager) collusion-resistance. The recent proposal in this area provides only limited collusion-resistance since it is only feasible for a small subset of possible Fingercasting system parameters.

Acknowledgments. We are very grateful to André Adelsbach and Ulrich Huber for the joyful joint research, which constitutes the major part of this paper. Further, we would like to thank Stefan Katzenbeisser for inspiring discussions and joint work. Also many thanks go to Frederik Armknecht and Christian Wachsmann for their helpful comments on the early drafts of this survey.

References

1. 4C Entity, LLC. CPPM specification—introduction and common cryptographic elements. Specification Revision 1.0, January 17 (2003), `http://www.4centity.com/data/tech/spec/cppm-base100.pdf`
2. AACS Licensing Administrator. Advanced access content system (AACS): Introduction and common cryptographic elements. Specification Revision 0.90, April 14 (2005), `http://www.aacsla.com/specifications/AACS_Spec-Common_0.90.pdf`
3. Adelsbach, A., Huber, U., Sadeghi, A.-R.: Fingercasting—joint fingerprinting and decryption of broadcast messages. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 136–147. Springer, Heidelberg (2006)
4. Adelsbach, A., Katzenbeisser, S., Sadeghi, A.-R.: Watermark detection with zero-knowledge disclosure. ACM Multimedia Systems Journal, Special Issue on Multimedia Security (2003)
5. Adelsbach, A., Katzenbeisser, S., Sadeghi, A.-R.: A computational model for watermark robustness. In: Camenisch, J., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 145–160. Springer, Heidelberg (2007)
6. Adelsbach, A., Rohe, M., Sadeghi, A.-R.: Overcoming the obstacles of zero-knowledge watermark detection. In: Proceedings of ACM Multimedia Security Workshop, pp. 46–55 (2004)
7. Adelsbach, A., Rohe, M., Sadeghi, A.-R.: Towards multilateral secure digital rights distribution infrastructures. In: ACM DRM Workshop 2005 (2005)
8. Adelsbach, A., Sadeghi, A.-R.: Zero-knowledge watermark detection and proof of ownership. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 273–288. Springer, Heidelberg (2001)
9. Adelsbach, A., Sadeghi, A.-R.: Advanced techniques for dispute resolving and authorship proofs on digital works. In: Proceedings of SPIE Security and Watermarking of Multimedia Contents V, vol. 5020 (2003)
10. Anderson, R.J.: Security Engineering: A Guide to Building Dependable Distributed Systems, 1st edn. John Wiley, Chichester (2001)
11. Anderson, R.J., Kuhn, M.: Tamper resistance—a cautionary note. In: Tygar, D. (ed.) USENIX Electronic Commerce 1996, pp. 1–11. USENIX (1996)
12. Anderson, R.J., Manifavas, C.: Chameleon—a new kind of stream cipher. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 107–113. Springer, Heidelberg (1997)
13. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Transactions on Information Theory 44(5), 1897–1905 (1998)
14. Boudot, F.: Efficient proofs that a committed number lies in an interval. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 431–444. Springer, Heidelberg (2000)
15. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. Journal of Computer and System Sciences 37(2), 156–189 (1988)
16. Briscoe, B., Fairman, I.: Nark: Receiver-based multicast non-repudiation and key management. In: ACM EC 1999, pp. 22–30. ACM Press, New York (1999)
17. Brown, I., Perkins, C., Crowcroft, J.: Watercasting: Distributed watermarking of multicast media. In: Rizzo, L., Fdida, S. (eds.) NGC 1999. LNCS, vol. 1736, pp. 286–300. Springer, Heidelberg (1999)
18. Celik, M.U., Lemma, A.N., Katzenbeisser, S., van der Veen, M.: Secure embedding of spread spectrum watermarks using look-up-tables. In: ICASSP 2007 [43] (2007)
19. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (1994)

20. Cox, I., Kilian, J., Leighton, T., Shamoon, T.: A secure, robust watermark for multimedia. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, pp. 175–190. Springer, Heidelberg (1996)
21. Cox, I., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, San Francisco (2002)
22. Cox, I.J., Kilian, J., Leighton, T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing 6(12), 1673–1687 (1997)
23. Cox, I.J., Linnartz, J.-P.M.G.: Some general methods for tampering with watermarks. IEEE Journal on Selected Areas in Communications 16(4), 587–593 (1998)
24. Craver, S.: Zero knowledge watermark detection. In: Pfitzmann (ed.) [59], pp. 101–116.
25. Craver, S., Katzenbeisser, S.: Copyright protection protocols based on asymmetric watermarking: The ticket concept. In: Communications and Multimedia Security Issues of the New Century, pp. 159–170. Kluwer Academic Publishers, Dordrecht (2001)
26. Craver, S., Katzenbeisser, S.: Security analysis of public-key watermarking schemes. In: Proceedings of SPIE, Mathematics of Data/Image Coding, Compression and Encryption IV, with Applications, vol. 4475, pp. 172–182 (2001)
27. Craver, S., Liu, B., Wolf, W.: An implementation of, and attacks on, zero-knowledge watermarking. In: Fridrich, J.J. (ed.) IH 2004. LNCS, vol. 3200, pp. 1–12. Springer, Heidelberg (2004)
28. Damgård, I.: Commitment schemes and zero-knowledge protocols. In: Damgård, I.B. (ed.) EEF School 1998. LNCS, vol. 1561, pp. 63–86. Springer, Heidelberg (1999)
29. Damgård, I., Fujisaki, E.: A statistically-hiding integer commitment scheme based on groups with hidden order. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 125–142. Springer, Heidelberg (2002)
30. Eggers, J.J., Su, J.K., Girod, B.: Asymmetric watermarking schemes. In: Sicherheit in Netzen und Medienströmen, September 2000, Springer Reihe, Informatik Aktuell (2000)
31. Eggers, J.J., Su, J.K., Girod, B.: Public key watermarking by eigenvectors of linear transforms. In: Proceedings of the European Signal Processing Conference (2000)
32. Ergün, F., Kilian, J., Kumar, R.: A note on the limits of collusion-resistant watermarks. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 140–149. Springer, Heidelberg (1999)
33. Ferguson, N., Schneier, B., Wagner, D.: Security weaknesses in a randomized stream cipher. In: Dawson, E., Clark, A., Boyd, C. (eds.) ACISP 2000. LNCS, vol. 1841, pp. 234–241. Springer, Heidelberg (2000)
34. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
35. Fujisaki, E., Okamoto, E.: Statistical zero knowledge protocols to prove modular polynomial relations. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 16–30. Springer, Heidelberg (1997)
36. Fujisaki, E., Okamoto, T.: Statistical zero-knowledge protocols to prove modular relations. IEICE Transactions on Fundamentals, E82-A(1):81–92 (Jan 1999)
37. Furon, T., Duhamel, P.: An asymmetric public detection watermarking technique. In: Pfitzmann [59], pp. 88–100
38. Goldreich, O.: Foundations of Cryptography, volume Basic Tools. Cambridge University Press, Cambridge (2001)

39. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM 38(3), 690–728 (1991)
40. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. In: Proceedings of the 17th Annual Symposium on Theory of Computing (STOC), Providence, RI, USA, May 1985, pp. 291–304. ACM Press, New York (1985)
41. Gopalakrishnan, K., Memon, N., Vora, P.: Protocols for watermark verification. In: Multimedia and Security, Workshop at ACM Multimedia, pp. 91–94 (1999)
42. Halevy, D., Shamir, A.: The LSD broadcast encryption scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
43. IEEE International Conference on Acoustics, Speech and Signal Processing—ICASSP 2007, Honolulu, Hawaii, USA, April 15–20, 2007. IEEE Computer Society, Los Alamitos (2007)
44. Jho, N.-S., Hwang, J.Y., Cheon, J.H., Kim, M.-H., Lee, D.H., Yoo, E.S.: One-way chain based broadcast encryption schemes. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 559–574. Springer, Heidelberg (2005)
45. Katzenbeisser, S.: Computational security models for digital watermarks (April 2005)
46. Katzenbeisser, S., Skoric, B., Celik, M., Sadeghi, A.-R.: Combining tardos fingerprinting codes and fingercasting. In: Information Hiding conference 2007 (2007)
47. Kilian, J., Leighton, F.T., Matheson, L.R., Shamoon, T.G., Tarjan, R.E., Zane, F.: Resistance of digital watermarks to collusive attacks. Technical Report TR-585-98, Princeton University, Department of Computer Science, July 27 (1998), `ftp://ftp.cs.princeton.edu/techreports/1998/585.ps.gz`
48. Kundur, D., Karthik, K.: Video fingerprinting and encryption principles for digital rights management. Proceedings of the IEEE 92(6), 918–932 (2004)
49. Linnartz, J.-P.M.G., van Dijk, M.: Analysis of the sensitivity attack against electronic watermarks in images. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 258–272. Springer, Heidelberg (1998)
50. Lipmaa, H.: On diophantine complexity and statistical zero-knowledge arguments. In: Laih, C.S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 398–415. Springer, Heidelberg (2003)
51. Liu, Y., Gao, W., Yao, H., Song, Y.: Secure watermark verification scheme. In: Chen, Y.-C., Chang, L.-W., Hsu, C.-T. (eds.) PCM 2002. LNCS, vol. 2532, pp. 477–484. Springer, Heidelberg (2002)
52. Liu, Y., Yang, X., Yao, H., Huang, T., Gao, W.: Watermark detection schemes with high security. In: ITCC (2), pp. 113–117. IEEE Computer Society Press, Los Alamitos (2005)
53. Luh, W., Kundur, D.: New paradigms for effective multicasting and fingerprinting of entertainment media. IEEE Communications Magazine 43(5), 77–84 (2005)
54. Maurer., U.: Conditionally-perfect secrecy and a provably-secure randomized cipher. Journal of Cryptology 5(1), 53–66 (1992)
55. Maurer, U.M.: A provably-secure strongly-randomized cipher. In: Damgard, I. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 361–373. Springer, Heidelberg (1990)
56. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
57. Naor, M., Pinkas, B.: Threshold traitor tracing. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 502–517. Springer (1998)

58. Parviainen, R., Parnes, P.: Large scale distributed watermarking of multicast media through encryption. In: Steinmetz, R., Dittmann, J., Steinebach, M. (eds.) Communications and Multimedia Security (CMS 2001), IFIP Conference Proceedings, pp. 149–158. International Federation for Information Processing, Communications and Multimedia Security (IFIP), vol. 192. Kluwer, Dordrecht (2001)
59. Pfitzmann, A. (ed.): IH 1999. LNCS, vol. 1768. Springer, Heidelberg (2000)
60. Tardos., G.: Optimal probabilistic fingerprint codes. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC), pp. 116–125 (2003)
61. D. S. Touretzky. Gallery of CSS descramblers. Webpage, Computer Science Department of Carnegie Mellon University (2000) (November 17, 2005), `http://www.cs.cmu.edu/~dst/DeCSS/Gallery`
62. van Schyndel, R.G., Tirkel, A.Z., Svalbe, I.D.: Key independent watermark detection. In: Proceedings of the IEEE International Conference on Multimedia Computing and Systems, vol. 1 (1999)
63. Zhao, X., Dai, Y., Feng, D.: A generalized method for constructing and proving zero-knowledge watermark proof systems. In: Cox, I.J., Kalker, T., Lee, H.-K. (eds.) IWDW 2004. LNCS, vol. 3304, pp. 204–217. Springer, Heidelberg (2004)

Detecting Photographic Composites of People

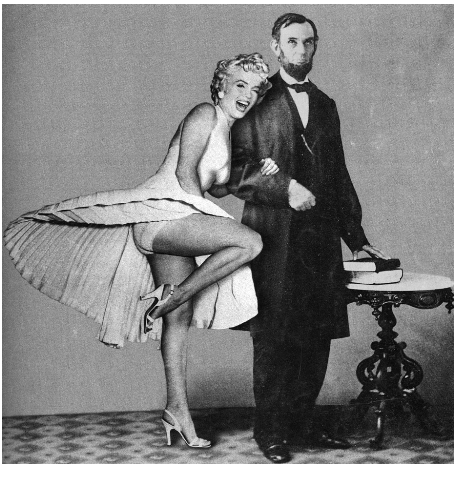
Micah K. Johnson and Hany Farid

Dartmouth College
Hanover, NH 03755, USA
{kimo,farid}@cs.dartmouth.edu
http://www.cs.dartmouth.edu/~{kimo,farid}

Abstract. The compositing of two or more people into a single image is a common form of manipulation. We describe how such composites can be detected by estimating a camera's intrinsic parameters from the image of a person's eyes. Differences in these parameters across the image are used as evidence of tampering.

Keywords: Digital Tampering, Digital Forensics.

1 Introduction



Composite of Marilyn Monroe and Abraham Lincoln (*by Jack Harris*).

From the creation of tabloid covers to political advertisements to family portraits, the compositing of two or more people into a single image is a common form of manipulation. Shown on the right, for example, is a composite of actress Marilyn Monroe (1926-1962) and President Abraham Lincoln (1809-1865).

Over the past few years the field of digital forensics has emerged to detect various forms of tampering. Forensic techniques have been developed for detecting cloning [2, 13]; splicing [11]; re-sampling artifacts [1, 14]; color filter array aberrations [15]; disturbances of a camera's sensor noise pattern [10]; and lighting inconsistencies [4,6,7]. Here we describe a new forensic technique specifically designed to detect composites of people. This approach estimates a camera's principal point from the image of a person's eyes. Inconsistencies in the principal point are then used as evidence of tampering.

In authentic images, the principal point is near the center of the image. When a person is translated in the image as part of creating a composite, the principal point is moved proportionally. Differences in the estimated principal point across

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 19–33, 2008.

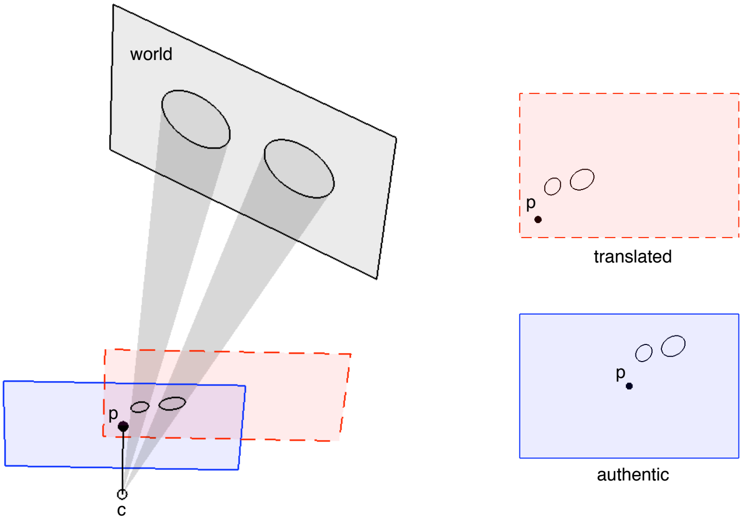


Fig. 1. Shown on the left is the projection of two circles (eyes) from a world plane onto an image plane (blue, solid line), with camera center c and principal point p. Also shown is a translated image plane (red, dashed line) which is equivalent to translating the image of the eyes in the original image plane (as shown on the right). Note that the translation results in the movement of the principal point.

the image can therefore be used as evidence of tampering. Shown in Fig. 1, for example, is the projection of two circles (eyes) from a world plane onto an image plane. The point c denotes the camera center and the point p denotes the principal point (the projection of c onto the image plane). Also shown in this figure is a translated image plane which is equivalent to translating the image of the eyes in the original image plane. Note that the translation results in the movement of the principal point away from the image center.

We describe how to estimate a camera's principal point from the image of a pair of eyes. We then show how translation in the image plane is equivalent to a shift of the principal point. Inconsistencies in the principal point across an image are used as evidence of tampering. We show the efficacy of this approach on synthetic and real images and visually plausible forgeries.

2 Methods

In general, the mapping between points in 3-D world coordinates to 2-D image coordinates is described by the projective imaging equation:

$$\boldsymbol{x} = P\boldsymbol{X}, \tag{1}$$

where the matrix P is a 3×4 projective transform, the vector $\boldsymbol{X}$ represents a world point in homogeneous coordinates, and the vector $\boldsymbol{x}$ represents an image

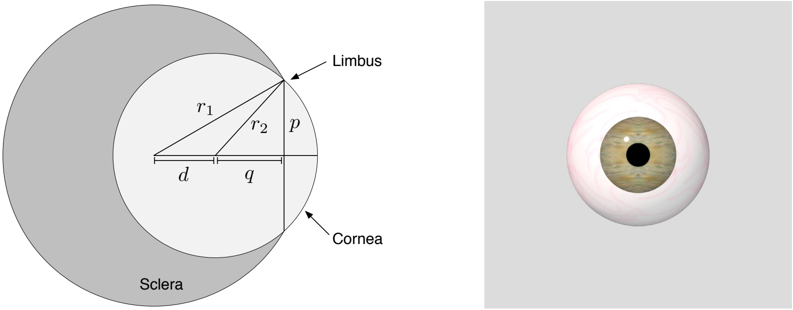


Fig. 2. A 3-D model of a human eye consisting of two spheres (left) and a synthetic eye rendered according to the model (right)

point in homogeneous coordinates. If all the world points $\boldsymbol{X}$ are coplanar, then the world coordinate system can be defined such that the points lie on the $Z = 0$ plane. In this case, the projective transformation P reduces to a 3×3 planar projective transform H, also known as a homography:

$$\boldsymbol{x} = H\boldsymbol{X}, \tag{2}$$

where the world points $\boldsymbol{X}$ and image points $\boldsymbol{x}$ are represented by 2-D homogeneous vectors.

We first describe how the homography H can be estimated from an image of a person's eyes and show how this transform can be factored into a product of matrices that embody the camera's intrinsic and extrinsic parameters. We then show how translation in the image plane can be detected from inconsistencies in the estimated camera's intrinsic parameters.

2.1 Homography Estimation

The homography H between points on a world plane and its projection on the image plane can be estimated if there is known geometry in the world: parallel lines, orthogonal lines, regular polygons, or circles [3,9]. We will focus primarily on the known geometry of a pair of eyes (circles) to estimate the homography.

A simple 3-D model for an eye consists of two spheres [8]. The larger sphere, with radius $r_1 = 11.5$ mm, represents the sclera and the smaller sphere, with radius $r_2 = 7.8$ mm, represents the cornea, Fig. 2. The centers of the spheres are displaced by a distance $d = 4.7$ mm [8]. The limbus, the boundary between the iris and the sclera, is defined by the intersection of two spheres – a circle with radius $p = 5.8$ mm.

With the assumption that the two circular limbi are planar, the homography H, Equation (2), can be estimated from a single image of a pair of eyes. Intuitively, the limbi will be imaged as ellipses (except when the eyes are directly facing the camera) and the distortion of the ellipses away from circles will be related to the pose and position of the eyes relative to the camera. We therefore seek the transform that aligns the image of the limbi to circles.

Points on the limbus in world coordinates satisfy the following implicit equation of a circle:

$$f(\boldsymbol{X};\boldsymbol{\alpha}) = (X_1 - C_1)^2 + (X_2 - C_2)^2 - r^2 = 0, \tag{3}$$

where $\boldsymbol{\alpha} = (\; C_1 \;\; C_2 \;\; r \;)^T$ denotes the circle center and radius. Consider a collection of points, $\boldsymbol{X_i}$, $i = 1, \ldots, m$, each of which satisfy Equation (3). Under an ideal pinhole camera model, the world point $\boldsymbol{X_i}$ maps to the image point $\boldsymbol{x_i}$ as:

$$\boldsymbol{x_i} = H\boldsymbol{X_i}, \tag{4}$$

where H is the 3×3 homography.

The estimation of H can be formulated in an orthogonal distance fitting framework. Let $E(\cdot)$ be an error function on the parameter vector $\boldsymbol{\alpha}$ and the unknown homography H:

$$E(\boldsymbol{\alpha}, H) = \sum_{i=1}^{m} \min_{\hat{\boldsymbol{X}}} \left\| \boldsymbol{x_i} - H\hat{\boldsymbol{X}} \right\|^2, \tag{5}$$

where $\hat{\boldsymbol{X}}$ is on the circle parametrized by $\boldsymbol{\alpha}$. This error embodies the sum of the squared errors between the data, $\boldsymbol{x_i}$, and the closest point on the model, $\hat{\boldsymbol{X}}$. One circle provides five constraints on the nine unknowns of H. In order to estimate H, at least one other circle is required. With two circles, the above error function takes the form:

$$\begin{aligned}\hat{E}(\boldsymbol{\alpha}_1, H_1, \boldsymbol{\alpha}_2, H_2) &= E(\boldsymbol{\alpha}_1, H_1) + E(\boldsymbol{\alpha}_2, H_2) \\ &\quad + \omega \left(\|\boldsymbol{H}_1 - \boldsymbol{H}_2\|^2 + (r_1 - 5.8)^2 + (r_2 - 5.8)^2 \right), \end{aligned} \tag{6}$$

where w is a scalar weighting factor. The first two terms are the individual error functions, Equation (5), for the two circles. The remaining terms constrain the transforms for both circles to be the same,[1] and the radius to be equal to 5.8 mm. This error function is minimized using non-linear least squares via the Levenberg-Marquardt iteration [6]. When the two circles are co-planar with respect to the camera plane, the eyes will be imaged as circles regardless of where they are in the world coordinate system. In this case, the principal point cannot be uniquely determined, and is assumed to be at the image center.

2.2 Camera Calibration

Once estimated, the homography H can be decomposed in terms of its intrinsic and extrinsic camera parameters [16,18]. The intrinsic parameters consist of the focal length f, principal point (c_1, c_2), skew σ, and aspect ratio α. The extrinsic parameters consist of a rotation matrix R and translation vector $\boldsymbol{t}$ that define

[1] The notation $\boldsymbol{H}_i$ expresses the matrix H_i as a vector.

the transformation between the world and camera coordinate systems. Since the world points lie on a single plane, H can be decomposed in terms of the intrinsic and extrinsic parameters [3] as:

$$H = \lambda K \left(\boldsymbol{r}_1 \; \boldsymbol{r}_2 \; \boldsymbol{t} \right), \tag{7}$$

where λ is a scale factor and the 3×3 intrinsic matrix K is:

$$K = \begin{pmatrix} \alpha f & \sigma & c_1 \\ 0 & f & c_2 \\ 0 & 0 & 1 \end{pmatrix}. \tag{8}$$

For simplicity, we will assume that the skew (σ) is zero and that the aspect ratio (α) is 1. Under these assumptions, the matrix K is:

$$K = \begin{pmatrix} f & 0 & c_1 \\ 0 & f & c_2 \\ 0 & 0 & 1 \end{pmatrix}. \tag{9}$$

The camera's intrinsic components can be estimated by decomposing H according to Equation (7). It is straightforward to show that that $\boldsymbol{r}_1 = \frac{1}{\lambda} K^{-1} \boldsymbol{h}_1$ and $\boldsymbol{r}_2 = \frac{1}{\lambda} K^{-1} \boldsymbol{h}_2$ where $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are the first two columns of the matrix H. The constraint that $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are orthogonal (they are columns of a rotation matrix) and have the same norm (unknown due to the scale factor λ) yields two constraints on the unknown matrix K:

$$\boldsymbol{r}_1^T \boldsymbol{r}_2 = \boldsymbol{h}_1^T (K^{-T} K^{-1}) \boldsymbol{h}_2 = 0, \tag{10}$$

$$\boldsymbol{r}_1^T \boldsymbol{r}_1 - \boldsymbol{r}_2^T \boldsymbol{r}_2 = \boldsymbol{h}_1^T (K^{-T} K^{-1}) \boldsymbol{h}_1 - \boldsymbol{h}_2^T (K^{-T} K^{-1}) \boldsymbol{h}_2 = 0. \tag{11}$$

With only two constraints, it is possible to estimate the principal point (c_1, c_2) or the focal length f, but not both [18]. As such, we will assume a known focal length.

For notational simplicity we solve for the components of $Q = K^{-T} K^{-1}$, which contain the desired coordinates of the principal point and the assumed known focal length:

$$Q = \frac{1}{f^2} \begin{pmatrix} 1 & 0 & -c_1 \\ 0 & 1 & -c_2 \\ -c_1 & -c_2 & c_1^2 + c_2^2 + f^2 \end{pmatrix}. \tag{12}$$

In terms of Q, the first constraint, Equation (10), takes the form:

$$\begin{aligned} h_1 h_2 + h_4 h_5 - (h_2 h_7 + h_1 h_8) c_1 - (h_5 h_7 + h_4 h_8) c_2 \\ + h_7 h_8 (c_1^2 + c_2^2 + f^2) = 0, \end{aligned} \tag{13}$$

where h_i is the i^{th} element of the matrix H in row-major order. Note that this constraint is a second-order polynomial in the coordinates of the principal point, which can be factored as follows:

$$(c_1 - \alpha_1)^2 + (c_2 - \beta_1)^2 = \gamma_1^2, \tag{14}$$

where:

$$\alpha_1 = (h_2h_7 + h_1h_8)/(2h_7h_8), \tag{15}$$

$$\beta_1 = (h_5h_7 + h_4h_8)/(2h_7h_8), \tag{16}$$

$$\gamma_1^2 = \alpha_1^2 + \beta_1^2 - f^2 - (h_1h_2 + h_4h_5)/(h_7h_8). \tag{17}$$

Similarly, the second constraint, Equation (11), takes the form:

$$h_1^2 + h_4^2 + 2(h_2h_8 - h_1h_7)c_1 + 2(h_5h_8 - h_4h_7)c_2 - h_2^2 - h_5^2 \\ + (h_7^2 - h_8^2)(c_1^2 + c_2^2 + f^2) = 0, \tag{18}$$

or,

$$(c_1 - \alpha_2)^2 + (c_2 - \beta_2)^2 = \gamma_2^2, \tag{19}$$

where:

$$\alpha_2 = (h_1h_7 - h_2h_8)/(h_7^2 - h_8^2), \tag{20}$$

$$\beta_2 = (h_4h_7 - h_5h_8)/(h_7^2 - h_8^2), \tag{21}$$

$$\gamma_2^2 = \alpha_2^2 + \beta_2^2 - (h_1^2 + h_4^2 - h_2^2 - h_5^2)/(h_7^2 - h_8^2) - f^2. \tag{22}$$

Both constraints, Equations (14) and (19) are circles in the desired coordinates of the principal point c_1 and c_2, and the solution is the intersection of the two circles.[2]

For certain homographies, however, this solution can be numerically unstable. For example, if $h_7 \approx 0$ or $h_8 \approx 0$, the first constraint becomes numerically unstable. Similarly, if $h_7 \approx h_8$, the second constraint becomes unstable. In order to avoid these instabilities, an error function with a regularization term is introduced.

We start with the following error function to be minimized:

$$E(c_1, c_2) = g_1(c_1, c_2)^2 + g_2(c_1, c_2)^2, \tag{23}$$

where $g_1(c_1, c_2)$ and $g_2(c_1, c_2)$ are the constraints on the principal point given in Equations (13) and (18), respectively. To avoid numerical instabilities, a regularization term is added to penalize deviations of the principal point from the image center $(0, 0)$ (in normalized coordinates). This augmented error function takes the form:

$$E(c_1, c_2) = g_1(c_1, c_2)^2 + g_2(c_1, c_2)^2 + \Delta(c_1^2 + c_2^2), \tag{24}$$

where Δ is a scalar weighting factor. This error function is a nonlinear least-squares problem, which can be minimized using a Levenberg-Marquardt iteration. The image center $(0, 0)$ is used as the initial condition for the iteration.

[2] In fact, there can be zero, one, two or an infinite number of real solutions depending on the configuration of the circles.

2.3 Translation

The translation of two circles (eyes) in the image is equivalent to translating the camera's principal point. In homogeneous coordinates, translations are represented by multiplication with a translation matrix T:

$$\boldsymbol{y} = T\boldsymbol{x}, \tag{25}$$

where:

$$T = \begin{pmatrix} 1 & 0 & d_1 \\ 0 & 1 & d_2 \\ 0 & 0 & 1 \end{pmatrix}, \tag{26}$$

and the amount of translation is (d_1, d_2). The mapping from world $\boldsymbol{X}$ to (translated) image coordinates $\boldsymbol{y}$ is:

$$\begin{aligned} \boldsymbol{y} &= TH\boldsymbol{X} \\ &= \lambda TK \left(\boldsymbol{r}_1 \; \boldsymbol{r}_2 \; \boldsymbol{t} \right) \boldsymbol{X} \\ &= \lambda \hat{K} \left(\boldsymbol{r}_1 \; \boldsymbol{r}_2 \; \boldsymbol{t} \right) \boldsymbol{X}, \end{aligned} \tag{27}$$

where

$$\hat{K} = \begin{pmatrix} f & 0 & c_1 + d_1 \\ 0 & f & c_2 + d_2 \\ 0 & 0 & 1 \end{pmatrix}. \tag{28}$$

Therefore, translation in image coordinates is equivalent to translating the principal point. Assuming the principal point in an authentic image is near the origin [17], large deviations from the image center, or inconsistencies in the estimated principal point across the image, can be used as evidence of tampering.

3 Results

We tested our technique for estimating the principal point from images of eyes on synthetically generated images, real images, and visually plausible forgeries. In all of these results, the principal point was estimated by minimizing Equation (24). Because of the regularization term used in this solution, we found that the estimated principal point was biased towards the image center $(0, 0)$. For purely aesthetic purposes, we rescaled the norm, n, of the estimated principal point by $3n^{1.7}$, where the form of this correction was chosen empirically. Throughout, we will refer to normalized image coordinates where the image center is $(0, 0)$ and the horizontal and vertical coordinates are normalized so that the maximum of the dimensions is in the range $[-1, 1]$.

3.1 Synthetic

Shown in Fig. 3 are two examples of synthetically generated heads rendered using the `pbrt` environment [12]. Also shown in this figure is a magnified view

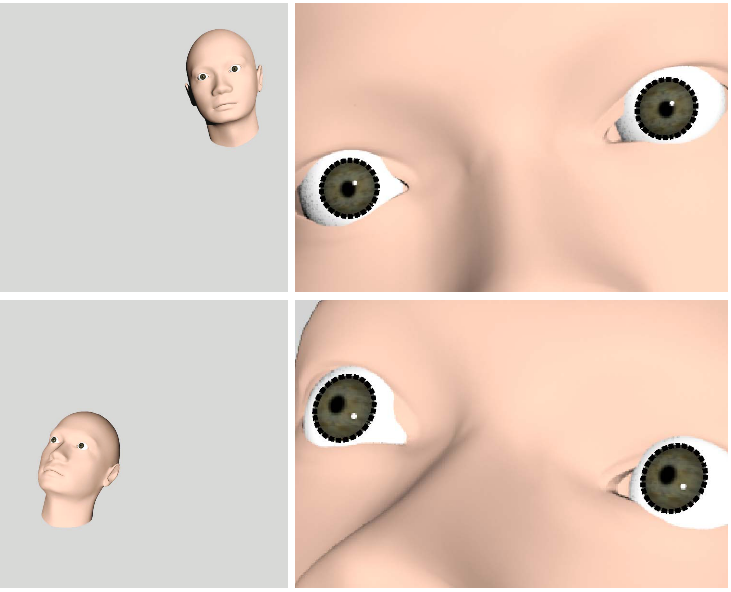


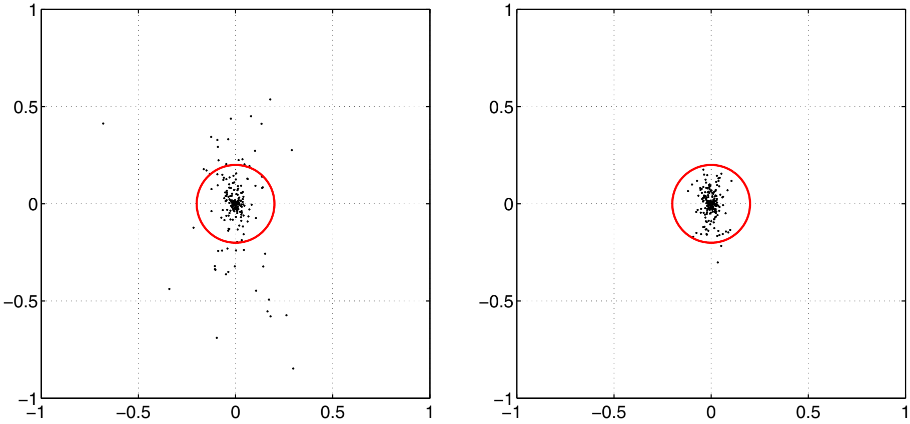
Fig. 3. A 3-D model of a head in two different locations and orientations (left), and a magnified view of the eyes with the extracted boundaries of the limbi (right)

of the eyes whose shape conformed to the model described in Section 2.1. The eyes were rendered with a full head model to provide a notion of size, though only the shape of the limbi are used in the subsequent analysis. Each image was rendered at 2400 × 2400 pixels and the radius of the limbus ranged from 24 to 34 pixels.

In the first set of simulations, the head model was rotated to 8 different orientations (ranging from -15 to 15 degrees from parallel) and 27 different locations, for a total of 216 images. Two sample images are shown in Fig. 3. The elliptical shapes of the limbi in each image were automatically extracted, Fig. 3. The homography H was estimated as described in the previous section, with a regularization term $\Delta = 0.0$ or $\Delta = 1.0$, Equation (24). The actual principal point in these synthetically generated images is the image center: $(0, 0)$ in normalized coordinates.

Shown in the left panel of Fig. 4 are the unconditioned estimates ($\Delta = 0.0$) and in the right panel are the conditioned estimates ($\Delta = 1.0$). Note that the conditioning significantly improves the accuracy of the estimation: without conditioning 80.6% (174/216) of the estimates are within 0.2 units of the origin (red circle), and with conditioning 99.1% (214/216) are within 0.2 units of the origin.

In the second set of simulations, the head model was positioned at the center of the world coordinate system and rotated to 252 different orientations. The rotations ranged from -30 to 30 degrees about each axis. Shown in Fig. 5 are four



$\varDelta = 0.0$ $\varDelta = 1.0$

Fig. 4. Estimates of the principal point in normalized coordinates for 216 synthetically generated images, Fig 3. Shown on the left are the unconditioned estimates ($\varDelta = 0.0$) and on the right are the conditioned estimates ($\varDelta = 1.0$). Note that the conditioning significantly improves the accuracy of the estimation – the actual principal point is the origin $(0, 0)$. A circle at a threshold of 0.2 units is drawn for reference.

sample images. To simulate tampering, the head in each image was translated to various locations and the principal point was estimated at each location. Superimposed on the images in Fig. 5 are level curves denoting the deviation of the estimated principal point from the origin as a function of spatial position in the image. The inner curve denotes a distance of 0.2, and each subsequent curve denotes an increment of 0.1. With a threshold of 0.2, translations of the head outside of the inner-most curve can be detected as fake. Note that the level curves are typically asymmetric and depend on the orientation of the head.

In the third set of simulations, the 252 images from the previous experiment were translated in the image by random amounts such that the displacement was greater than 0.2 units in normalized coordinates (240 pixels in the original 2400×2400 image). Shown in the left panel of Fig. 6 are the estimated principal points for the original 252 images, and in the right panel are the estimated results for 1260 translated images. In both cases, the conditioned estimator ($\varDelta = 1.0$) was used. Of the 252 authentic images, 99.2% had an estimated principal point less than 0.2 units from the origin, and of the 1260 translated images, 94.6% had an estimated principal point greater than 0.2 units from the origin.

3.2 Real

Shown in Fig. 7 are four of 15 images taken with a Nikon D200 10-megapixel camera set to record at the highest quality JPEG format. At a radius of 21 pixels, the size of the eyes in these images was slightly smaller than in the above simulations. The principal point was estimated using the conditioned estimator,

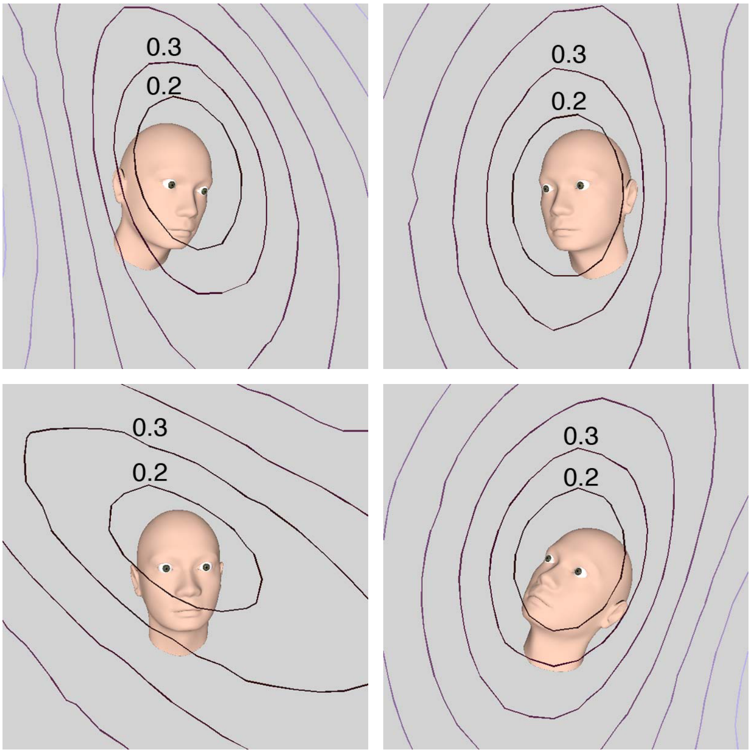


Fig. 5. A 3-D model of a head at four different orientations. The superimposed level curves show the deviation of the estimated principal point from the origin as a function of spatial position in the image.

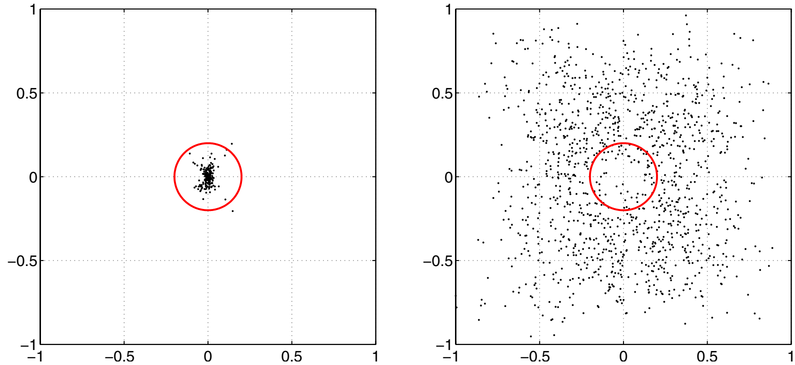


Fig. 6. Estimates of the principal point in normalized coordinates for 252 authentic images (left) and 1260 doctored images (right), Fig. 5. The actual principal point is the origin $(0, 0)$, and a circle at a threshold of 0.2 units is drawn for reference.

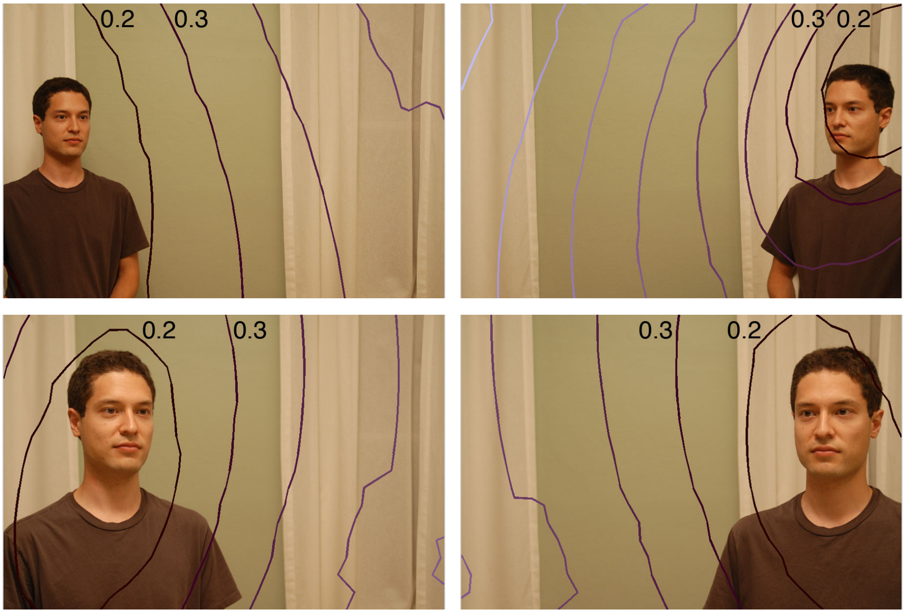


Fig. 7. Shown in each panel are authentic images with superimposed level curves showing the deviation of the estimated principal point from the true principal point as a function of spatial position in the image

Equation (24), with $\Delta = 1.0$. The average deviation from the calibrated principal point[3] was 0.15 units (in normalized coordinates) with a maximum distance of 0.24, a minimum distance of 0.05 and a standard deviation of 0.06 units. The eyes in each of the four images were translated to various locations in the image to simulate tampering. The level curves in Fig. 7 show the deviation of the estimated principal point from the true principal point, as a function of spatial position in the image. With a threshold of 0.2 units, translations in the image outside of the innermost curve are classified as fake.

Shown in Fig. 8 are four images acquired from Flickr, a popular image sharing website. The images were captured with different cameras at different focal lengths. The focal length was extracted from the metadata in the image and used to estimate the principal point. In each image, the '+' marker denotes the image center $(0, 0)$ and the white dots denote the principal points estimated from different people. The circle in each image has a radius of 0.2 and is centered at the average of the principal points. Note that in each of the four images, the estimated principal points fall within the circle, indicating relative agreement in the positions of the camera's principal point.

[3] The camera was calibrated to determine the actual principal point which at $(-0.028, 0.022)$ is close to the origin. The Camera Calibration Toolbox for Matlab `http://www.vision.caltech.edu/bouguetj/calib_doc` was used for this calibration.

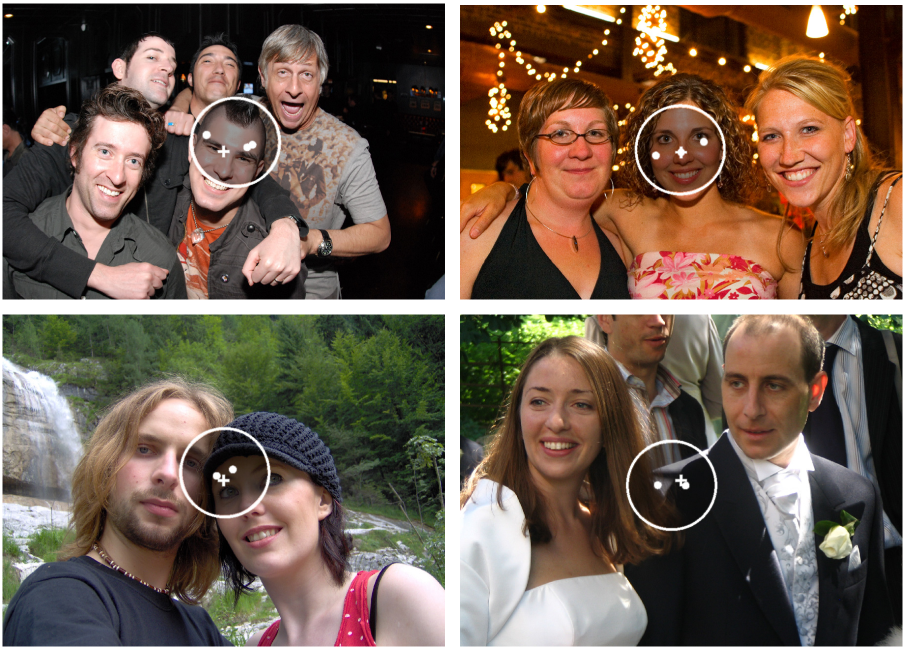


Fig. 8. Four authentic images. In each image, the '+' marker denotes the image center $(0, 0)$, and the dots denote the estimated principal points from each person (the eyes from only three people in the top left image were visible). The circle with radius 0.2 units is centered at the average of the principal points.

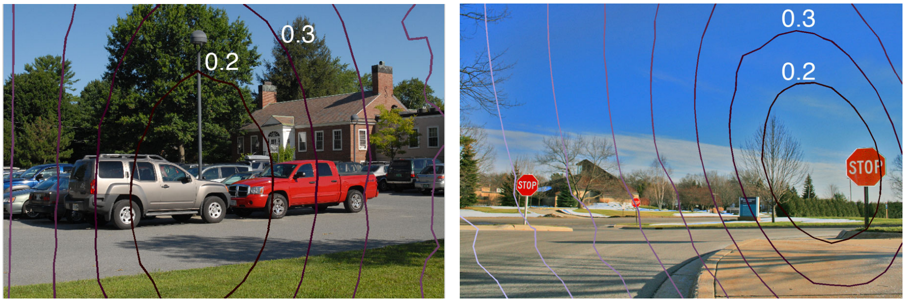


Fig. 9. Shown in each panel are authentic images with superimposed level curves showing the deviation of the estimated principal point from the true principal point as a function of spatial position in the image. The principal point for the left image was determined from the left-most car's wheels, and for the right image, the known geometry of the stop sign.

Since the homography can be estimated from other known geometries [5], the estimation of the principal point is not limited to images of the eyes. Shown in Fig. 9, for example, are results from a car's wheels, and the known geometry of a stop sign.

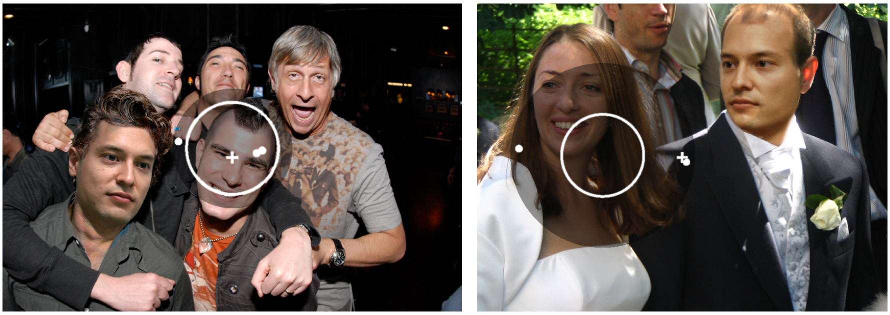


Fig. 10. Two forgeries made by combining images from Fig. 7 and Fig. 8. The '+' marker denotes the image center $(0, 0)$, and the dots denote the estimated principal points from each person. The circle with radius 0.2 units is centered at the average of the principal points. Notice that the estimated principal points are inconsistent with one another.

3.3 Forgeries

We created two forgeries by combining images from Fig. 7 and Fig. 8. As shown in Fig. 10, the principal points estimated from the forged heads are inconsistent with the other principal point(s) in the image.

4 Discussion

When creating a composite of two or more people it is often necessary to move a person in the image relative to their position in the original image. When done well, this manipulation is rarely visually obvious. We have shown how to detect such manipulations by estimating the camera's principal point (the projection of the camera center onto the image plane) from the image of a person's eyes. This approach relies on estimating the transformation from world to image coordinates and then factoring this transformation into a product of matrices containing intrinsic and extrinsic camera parameters. With a known focal length, the principal point can be determined from the intrinsic matrix. Inconsistencies in the estimated principal point can then be used as evidence of tampering.

We have shown the efficacy of this technique on simulated and real images. The major sensitivity with this technique is in extracting the elliptical boundary of the eye. This process will be particularly difficult for low-resolution images, but with a radius of $20 - 30$ pixels reasonably accurate estimates can be made from a person's eyes.

We expect this technique, in conjunction with a growing body of forensic tools, to be effective in exposing digital forgeries.

Acknowledgments

This work was supported by a Guggenheim Fellowship, a gift from Adobe Systems, Inc., a gift from Microsoft, Inc., a grant from the United States Air Force (FA8750-06-C-0011), and by the Institute for Security Technology Studies at Dartmouth College under grant 2005-DD-BX-1091 from the Bureau of Justice Assistance and Award Number 2006-CS-001-000001 from the U.S. Department of Homeland Security. Points of view or opinions in this document are those of the author and do not represent the official position or policies of the U.S. Department of Justice, the U.S. Department of Homeland Security, or any other sponsor.

References

1. Avcıbaş, İ., Bayram, S., Memon, N., Sankur, B., Ramkumar, M.: A classifier design for detecting image manipulations. In: International Conference on Image Processing, ICIP 2004, vol. 4, pp. 2645–2648 (2004)
2. Fridrich, J., Soukal, D., Lukáš, J.: Detection of copy-move forgery in digital images. In: Proceedings of Digital Forensic Research Workshop (August 2003)
3. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
4. Johnson, M.K., Farid, H.: Exposing digital forgeries by detecting inconsistencies in lighting. In: ACM Multimedia and Security Workshop (2005)
5. Johnson, M.K., Farid, H.: Metric measurements on a plane from a single image. Technical Report TR2006-579, Department of Computer Science, Dartmouth College (2006)
6. Johnson, M.K., Farid, H.: Exposing digital forgeries through specular highlights on the eye. In: 9th International Workshop on Information Hiding, Saint Malo, France (2007)
7. Johnson, M.K., Farid, H.: Exposing digital forgeries in complex lighting environments. IEEE Transactions on Information Forensics and Security (in press, 2007)
8. Lefohn, A., Caruso, R., Reinhard, E., Budge, B., Shirley, P.: An ocularist's approach to human iris synthesis. IEEE Computer Graphics and Applications 23(6), 70–75 (2003)
9. Liebowitz, D., Zisserman, A.: Metric rectification for perspective images of planes. Computer Vision and Pattern Recognition, 482–488 (1998)
10. Lukáš, J., Fridrich, J., Goljan, M.: Detecting digital image forgeries using sensor pattern noise. In: Proceedings of the SPIE, vol. 6072 (2006)
11. Ng, T.-T., Chang, S.-F.: A model for image splicing. In: IEEE International Conference on Image Processing (ICIP), Singapore (October 2004)
12. Pharr, M., Humphreys, G.: Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann, San Francisco (2004)
13. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting duplicated image regions. Technical Report TR2004-515, Department of Computer Science, Dartmouth College (2004)
14. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing 53(2), 758–767 (2005)

15. Popescu, A.C., Farid, H.: Exposing digital forgeries in color filter array interpolated images. IEEE Transactions on Signal Processing 53(10), 3948–3959 (2005)
16. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf cameras and lenses. IEEE Journal of Robotics and Automation RA-3(4), 323–344 (1987)
17. Willson, R.G., Shafer, S.A.: What is the center of the image? Journal of the Optical Society of America A 11(11), 2946–2955 (1994)
18. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1330–1334 (2000)

(In)Security of an Efficient Fingerprinting Scheme with Symmetric and Commutative Encryption of IWDW 2005

Raphael C.-W. Phan[1] and Bok-Min Goi[2,⋆]

[1] Laboratoire de sécurité et de cryptographie (LASEC),
Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
raphael.phan@epfl.ch

[2] Centre for Cryptography and Information Security (CCIS),
Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Malaysia
bmgoi@mmu.edu.my

Abstract. We analyze the security of a fingerprinting scheme proposed at IWDW 2005. We show two results, namely that this scheme (1) does not provide *seller security*: a dishonest buyer can repudiate the fact that he redistributed a content, and (2) does not provide *buyer security*: a buyer can be framed by a malicious seller.

Keywords: Watermarking, fingerprinting, security issues, combination of data hiding and cryptography, buyer-seller, repudiation, framing.

1 Introduction

Two of the most celebrated applications of watermarking are *copyright protection* and *piracy protection*. For this, a robust watermarking scheme is employed to embed the content owner's mark to prove his ownership; and to embed a mark (so called a fingerprint) of the content buyer so that the content binds to the buyer and any dishonest buyer who later redistributes this content can be traced.

An interesting body of literature in watermarking has formed around the design and analysis of buyer-seller watermarking (BSW) schemes, which are typically protocols that allow marks identifying both the seller (it is commonly assumed that the owner is the seller) and the buyer to be embedded into the content, so that copyright and piracy protection can be provided. In addition to ensuring this basic *seller security*, BSW schemes also provide *buyer security* [34], i.e., an honest buyer is assured that he cannot be framed by malicious sellers.

Related Work. It turns out that designing secure BSW schemes is more subtle than first thought. For instance, the original proposal that highlighted the need to provide buyer security in [34], was shown inadequate in [25] since the seller

⋆ The second author acknowledges the Malaysia eScience grants (01-02-01-SF0032, 01-02-01-SF0060).

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 34–44, 2008.

knows the final copy of the fingerprinted content and may well have redistributed this himself.

Meanwhile, a few subsequent BSW schemes proposed with different additional features like anonymity [20], without trusted third parties (TTP) [10] and extension for multiple purchases [11] were later found to have security problems [10,19,18]. A few more recent schemes can be found in [24,37,38].

BSW schemes typically employ techniques from both watermarking and cryptography. See [13,21,33] for cautions when integrating the two fields.

This Paper. We show the first known analysis of a recent BSW scheme proposed by Yong and Lee at IWDW 2005 [37]. Our results indicate that this scheme does not provide seller security and buyer security, properties that are desired by any basic BSW scheme.

Section 2 gives the preliminaries and notations used throughout this paper. We describe the Yong-Lee BSW scheme in Section 3, and then present our attacks in Section 4. Section 5 gives some concluding remarks.

2 Preliminaries

We list here basic requirements of a secure anonymous buyer-seller watermarking scheme (the interested reader can refer to [25,37] for details):

- **Traceability.** The buyer who has illegally redistributed watermarked contents can be traced.
- **Non-Repudiation.** The guilty buyer cannot deny having illegally redistributed copies of the content.
- **Non-Framing.** No one can accuse an honest buyer.
- **Privacy: Anonymity and Unlinkability.** Without obtaining an illegally distributed copy, the seller cannot identify the buyer. Also, the purchases of honest buyers should not be linkable even by a collusion of all sellers, registration center and other buyers.

Note that in any BSW scheme, it is assumed that the underlying watermarking scheme used for embedding is collusion-tolerant and robust.

2.1 Cryptographic Preliminaries

In a *public key cryptosystem* [26], each party A possesses a pair of public-private keys (y_A, x_A) obtainable from a certificate authority or registration center RC. For convenience, we let $y_A \equiv g^{x_A} \bmod p$ [26], where p is a large prime and g is a generator of the multiplicative group $\mathbb{Z}_p^*$ of order $(p-1)$. Also, unless otherwise specified, all arithmetic operations are performed in $\mathbb{Z}_p^*$. Any party can encrypt a message for A using y_A, but only A can decrypt this message with x_A. This ensures *confidentiality.* Furthermore, A can sign a message by encrypting it with x_A, denoted as $sign_{x_A}(M)$, so that anybody can verify by using y_A that the message really originated from A. This provides *authentication* and *non-repudiation.* Note however that it is common knowledge not to use the same key-pair for both encryption and signature.

Both the seller and the buyer have registered with the registration center RC, and have their own pair of keys which are (y_A, x_A) and (y_B, x_B), respectively. Note that the RC also has its own public-private key pair (y_{RC}, x_{RC}).

2.2 Notations

For ease of explanation, we use the following common notations for BSW schemes:

S	the seller who owns and sells the digital content X
B	the buyer who buys the digital content
RC	registration center who can issue certificates
J	the judge
$\otimes$	fingerprint embedding (watermarking) operation
X	original content with t elements $(x_1, x_2, ..., x_t)$
X'	fingerprinted content, where $X' = X \otimes F$ for a fingerprint F
$H(\cdot)$	collision-resistant hash function
$E_U(x)$	public-key encryption of x under party U's public key
$Enc_K(x)$	symmetric-key encryption of x under secret key K
$CEnc_K(x)$	commutative symmetric-key encryption of x under secret key K

3 The Yong-Lee Anonymous BSW Scheme

We describe the anonymous BSW scheme by Yong and Lee proposed at IWDW 2005 [37]. As is common for this type of scheme, it consists of three phases; i.e. *registration*, *fingerprinting* and *identification.* For better illustration, we depict the registration phase and fingerprinting phase in Fig. 1.

Registration. This phase involves two parties: the buyer B and registration center RC. Both are assumed to have public and private key pairs, i.e., x_I is the private key of party I while its public key is $y_I = g^{x_I}$. Certificates issued by RC are signed by its private key x_{RC}, and can be publicly verified by anyone using RC's public key y_{RC}.

1. B randomly chooses two secret values $x_1, x_2 \in \mathbb{Z}_p^*$ such that $x_1 + x_2 = x_B \in \mathbb{Z}_p^*$. Then B sends $(y_B, y_1 = g^{x_1}), E_{RC}(x_2)$ to RC, and convinces via zero knowledge to RC of its possession of x_1.
2. RC decrypts $E_{RC}(x_2)$ and computes $y_2 = g^{x_2}$ and checks that $y_1 y_2 = y_B$. If verified, it returns to B a certificate $Cert(y_1)$ which states the correctness of y_1 and the registration of B.

Repeating this phase several times allows B to obtain several different pairs (y_1, x_1) which it will use as its unlinkable and anonymous key pairs.

Buyer, B		**Registration Center,** RC
Randomly select:		
$x_1, x_2 \in_R \mathbb{Z}_p^*$ s.t. $x_1 + x_2 = x_B$		
Compute y_1 and encrypt x_2:		
$y_1 = g^{x_1}$, $E_{RC}(x_2)$.	$\xrightarrow{y_B, y_1, E_{RC}(x_2)}$	Decrypt $E_{RC}(x_2)$ using x_{RC}.
		Compute: $y_2 = g^{x_2}$.
		Check: $y_1 \cdot y_2 \overset{?}{=} y_B$.
	$\xleftarrow{Cert(y_1)}$	If pass, return $Cert(y_1)$.

(a) Registration Phase

Buyer, B		**Seller,** S
	$\xrightarrow{y_1, Cert(y_1), payment}$	Verify $Cert(y_1)$ (using y_{RC}).
		If pass, generate and embed:
		$F_B^i = \{f_B^{i,1}, f_B^{i,2}, \dots, f_B^{i,t}\}$,
		$X^i = \{x^{i,1}, x^{i,2}, \dots, x^{i,t}\}, i = \{0, 1\}$,
		$X_B^i = \{x_B^{i,1}, x_B^{i,2}, \dots, x_B^{i,t}\}$ where
		$x_B^{i,j} = x^{i,j} \otimes f_B^{i,j}$.
		Generate 2 secret keys K_0 and K_1:
		$K_i = \{k_{i,1}, k_{i,2}, \dots, k_{i,t}\}, i = \{0, 1\}$.
		Encrypt and obtain:
		$\mathcal{X}_B^i = \{\mathbf{x}_B^{i,1}, \mathbf{x}_B^{i,2}, \dots, \mathbf{x}_B^{i,t}\} = Enc_{K_i}(X_B^i)$.
		Encrypt K_0 and K_1 (using K_S):
		$C_i = \{CEnc_{K_S}(k_{i,1}), CEnc_{K_S}(k_{i,2}), \dots, CEnc_{K_S}(k_{i,t})\}$
		$= \{c_{i,1}, c_{i,2}, \dots, c_{i,t}\}$.
Randomly generate:	$\xleftarrow{\mathcal{X}_B^0, \mathcal{X}_B^1, C_0, C_1}$	
$L_B = \{l_1, l_2, \dots, l_t\}$		
for $l_j = \{0, 1\}$.		
Then construct:		
$C' = \{c'_1, c'_2, \dots, c'_t\}$		
where $c'_j = c_{l_j, j}$.		
Encrypts C' (using K_R):		
$D_1 = \{d_1, d_2, \dots, d_{\frac{t}{2}}\}$ and		
$D_2 = \{d_{\frac{t}{2}+1}, \dots, d_t\}$, where		
$d_i = CEnc_{K_R}(c'_i)$.	$\xrightarrow{D_1}$	Decrypt D_1 (using K_S):
		$U_1 = \{u_1, u_2, \dots, u_{\frac{t}{2}}\}$, where
Decrypt U_1 (using K_R).	$\xleftarrow{U_1}$	$u_i = CEnc_{K_S}^{-1}(d_i) = CEnc_{K_R}(k_{l_j, j})$.
Then, decrypt 1^{th} $t/2$ frames:		
$\mathbf{x}_B^{l_j, j}$ for $j = \{1, 2, \dots, \frac{t}{2}\}$.		
Generate:		
$T_B = E_J(L_B)$ and $sig_{x_1}(T_B)$.	$\xrightarrow{T_B, sig_{x_1}(T_B), D_2}$	Verify (using y_1) $sig_{x_1}(T_B)$.
		If verified, it decrypts D_2 (using K_S):
Decrypt U_2 (using K_R).	$\xleftarrow{U_2, sig_{x_S}(T_B)}$	$U_2 = \{u_{\frac{t}{2}+1}, \dots, u_t\}$.
Then, decrypt 2^{nd} $t/2$ frames:		
$X_B = \{x_B^{l_1,1}, x_B^{l_2,2}, \dots, x_B^{l_t,t}\}$.	$\xrightarrow{TN_B = E_J(H(X_B), H(X_B) \oplus H(L_B))}$	Record $Rec_B =$
		$\langle y_1, Cert(y_1), F_B^0, F_B^1, T_B, sig_{x_1}(T_B), TN_B \rangle$.

(b) Fingerprinting Phase

Fig. 1. Yong-Lee Anonymous BSW Scheme

Fingerprinting. This phase involves two parties: the buyer B and the seller S.

1. B sends y_1, $Cert(y_1)$ and *payment* to S as a purchase request for the digital content X.

2. On receiving this, S verifies $Cert(y_1)$ and generates two fingerprints F_B^0 and F_B^1 for B, i.e.,
$$F_B^i = \{f_B^{i,1}, f_B^{i,2}, \ldots, f_B^{i,t}\}, i = \{0, 1\}.$$

3. S generates two identical copies of the digital content X^0 and X^1, and splits each copy into t frames, i.e.,
$$X^i = \{x^{i,1}, x^{i,2}, \ldots, x^{i,t}\}, i = \{0, 1\}.$$

4. S then embeds F_B^i into each of the t frames of X^i for $i = \{0, 1\}$, by using the specific embedding construction in [14], to obtain
$$X_B^i = \{x_B^{i,1}, x_B^{i,2}, \ldots, x_B^{i,t}\}, i = \{0, 1\},$$
where
$$x_B^{i,j} = x^{i,j} \otimes f_B^{i,j} \quad , i = \{0, 1\}, j = \{1, \ldots, t\}.$$

5. S generates two secret key vectors K_0 and K_1. Each key vector consists of t randomly selected keys:
$$K_i = \{k_{i,1}, k_{i,2}, \ldots, k_{i,t}\}, i = \{0, 1\}.$$

6. S encrypts each of the t frames of X_B^i ($i = \{0, 1\}$) using each of the t keys of K_i, using symmetric key encryption $Enc_K(\cdot)$. This produces two encrypted digital content vectors $\mathcal{X}_B^0$ and $\mathcal{X}_B^1$ of frames, such that
$$\begin{aligned}\mathcal{X}_B^i &= \{\mathbf{x}_B^{i,1}, \mathbf{x}_B^{i,2}, \ldots, \mathbf{x}_B^{i,t}\}\\ &= Enc_{K_i}(X_B^i)\\ &= Enc_{k_{i,j}}(x_B^{i,j}), i = \{0, 1\}, j = \{1, \ldots, t\}.\end{aligned}$$

7. S randomly selects a secret key K_S and encrypts the two key vectors K_0 and K_1 via commutative encryption $CEnc_K(\cdot)$, producing two encrypted key vectors C_0 and C_1, i.e.
$$\begin{aligned}C_i &= \{c_{i,1}, c_{i,2}, \ldots, c_{i,t}\}\\ &= \{CEnc_{K_S}(k_{i,1}), CEnc_{K_S}(k_{i,2}), \ldots, CEnc_{K_S}(k_{i,t})\}, i = \{0, 1\}.\end{aligned}$$
S sends $(\mathcal{X}_B^0, \mathcal{X}_B^1, C_0, C_1)$ to B.

8. B randomly generates a t-bit integer $L_B = \{l_1, l_2, \ldots, l_t\}$ for $l_j = \{0, 1\}, j = \{1, \ldots, t\}$, restricted to the fact that L_B should not be all 0 or all 1. It then constructs a new encrypted vector $C' = \{c'_1, c'_2, \ldots, c'_t\}$ where $c'_j = c_{l_j,j}$. To elaborate, this means that each c'_j is either $c_{0,j}$ or $c_{1,j}$ depending on the bit l_j of L_B.

9. B randomly chooses a secret key K_R and encrypts C' via commutative encryption to obtain an encrypted vector that it halves into two consecutive parts $D_1 = \{d_1, d_2, \ldots, d_{\frac{t}{2}}\}$ and $D_2 = \{d_{\frac{t}{2}+1}, \ldots, d_t\}$, where
$$\begin{aligned} d_i &= CEnc_{K_R}(c'_i) \\ &= CEnc_{K_R}(CEnc_{K_S}(k_{l_j,j})) \\ &= CEnc_{K_S}(CEnc_{K_R}(k_{l_j,j})). \end{aligned}$$
   B sends D_1 to S.

10. S decrypts D_1 with K_S to get the vector $U_1 = \{u_1, u_2, \ldots, u_{\frac{t}{2}}\}$, where
$$\begin{aligned} u_i &= CEnc^{-1}_{K_S}(d_i) \\ &= CEnc^{-1}_{K_S}(CEnc_{K_S}(CEnc_{K_R}(k_{l_j,j}))) \\ &= CEnc_{K_R}(k_{l_j,j}). \end{aligned}$$
   S sends U_1 to B.

11. B now obtains $t/2$ decryption keys by decrypting each u_i with key K_R, and can thus decrypt the first $t/2$ frames of the encrypted digital content $\mathbf{x}_B^{l_j,j}$ for $j = \{1, 2, \ldots, \frac{t}{2}\}$.

12. B generates $T_B = E_J(L_B)$ and a signature $sig_{x_1}(T_B)$. These are evidence for resolving piracy disputes in future. B sends $(T_B, sig_{x_1}(T_B), D_2)$ to S.

13. S verifies $sig_{x_1}(T_B)$ with y_1. If verified, it decrypts D_2 with K_S to obtain the vector $U_2 = \{u_{\frac{t}{2}+1}, \ldots, u_t\}$, where u_i is similar to that in Step (10.). S sends $(U_2, sig_{x_S}(T_B))$ to B.

14. B now obtains the remaining $t/2$ decrypting keys by decrypting each u_i of U_2 with key K_R, thus it can decrypt the remaining $t/2$ frames of $\mathcal{X}_B^{l_j,j}$ for $j = \{\frac{t}{2}+1, \ldots, t\}$. Hence, B now has the complete fingerprinted content X_B, i.e.
$$X_B = \{x_B^{l_1,1}, x_B^{l_2,2}, \ldots, x_B^{l_t,t}\}.$$
   B sends $TN_B = E_J(H(X_B), H(X_B) \oplus H(L_B))$ to S.

15. S records $Rec_B = \langle y_1, Cert(y_1), F_B^0, F_B^1, T_B, sig_{x_1}(T_B), TN_B\rangle$ in its database.

Identification. This phase involves three parties: the seller S, the judge J and the registration center RC.

1. After finding an illegally redistributed digital content, S extracts the fingerprint from it. S then sends $\mathcal{X}_B^0$ and $\mathcal{X}_B^1$ with the transaction record Rec_B to the judge J.

2. J decrypts T_B and TN_B and checks that L_B corresponds to $\mathcal{X}_B$, and that T_B was signed by B. It verifies the presence of frames of either F_B^0 or F_B^1 in X_B based on L_B. If all are verified, it sends y_1 to RC and asks for the identity of B, and informs S.

4 Insecurity of the Yong-Lee BSW Scheme

Attacking the Seller Security. The security of the seller is captured by the notion of *traceability* and *non-repudiation.*

Nevertheless, we show how the seller security can be defeated by a malicious buyer. The attack follows.

1. B performs an entire **fingerprinting** protocol session with S, thus in the end B has the content X_B and S has recorded $Rec_B = \langle y_1, Cert(y_1), F_B^0, F_B^1, T_B, sig_{x_1}(T_B), TN_B\rangle$ in its database.
2. B initiates another **fingerprinting** protocol session with S, this time requesting for some other digital content X'. During the protocol, B proceeds normally, except that it reuses the y_1, $Cert(y_1)$, T_B, $sig_{x_1}(T_B)$, TN_B from the previous session. It is clear that S will correctly verify y_1 from $Cert(y_1)$, and T_B from $sig_{x_1}(T_B)$. Furthermore S cannot check TN_B since it is encrypted for only J to decrypt.
3. Thus in the end B obtains the fingerprinted X'_B and S records $Rec'_B = \langle y_1, Cert(y_1), F_B^{'0}, F_B^{'1}, T_B, sig_{x_1}(T_B), TN_B\rangle$ in its database.
4. B can repeat this as many times as it wishes. Now B can pirate all the fingerprinted content X'_B it received from its sessions with S except for the first, X_B.
5. When S discovers that X'_B has been redistributed and initiates the **identification** protocol, B can counter that it only bought once from S, for the digital content X_B. It can argue that the other X'_B have nothing to do with him, but that S reused y_1, $Cert(y_1)$, T_B, $sig_{x_1}(T_B)$, TN_B to frame him for distributing X'_B.
6. The judge J cannot reach a conclusion in favour of S because TN_B will not correspond to X'_B since it corresponds only to X_B.

This attack shows to some extent a failure of *traceability* since B cannot be judged guilty for redistributing X'_B. This also shows a failure of *non-repudiation* because the only part that binds to B for which B cannot repudiate is $T_B = E_J(L_B)$, which is independent of the digital content bought by B.

Attacking the Buyer Security. The security of the buyer is captured by the notion of *non-framing.* Additionally, when privacy is desired then this is captured by *anonymity* and *unlinkability.*

We demonstrate two cases for which *non-framing* can be violated. The first follows, by exploiting T_B.

1. S guesses all possible values of L_B and for each guess checks if $T_B = E_J(L_B)$. Since L_B is only a 32-bit vector, this requires just 2^{32} trials.

2. S does the **fingerprinting** protocol steps 3 and 4 for X', where the old fingerprints F_B^0 and F_B^1 are reused, and embedded into any other content X' for which S wants to frame B. This gives $X_B'^0$ and $X_B'^1$.
3. Since L_B has been obtained, S knows the fingerprinting pattern chosen by B. So S can embed the same pattern into any other content X'. Denote the fingerprinted content as X_B'.
4. S computes $TN_B' = E_J(H(X_B'), H(X_B') \oplus H(L_B))$.
5. S initiates the **identification** protocol to frame B for pirating X_B', by sending $X_B'^0$ and $X_B'^1$ together with transaction record $Rec_B' = \langle y_1$, $Cert(y_1)$, F_B^0, F_B^1, T_B, $sig_{x_1}(T_B)$, $TN_B' \rangle$ to the judge J.
6. J decrypts T_B and TN_B' and will correctly verify that L_B corresponds to X_B', and that T_B was signed by B. It will also correctly detect in X_B' the presence of the fingerprinting pattern based on L_B. Thus, this will cause J to agree that B has pirated X_B', and it will send y_1 to RC to ask for the identity of B, and informs S.

The second attack below also violates *non-framing* in the sense that even if B was dishonest and redistributed X_B, it should only be held guilty for X_B and not for any other content X_B' for which it did not redistribute. This is in line with the common legal system. If this is violated, it is still unfair to B; for instance if X_B is some inexpensive content whose copyright is claimed by S only for a brief period thus B might feel it is ok to redistribute among friends after some time. However, once X_B is obtained by S it can frame B for redistributing some other very expensive content X_B' and for which it holds copyright indefinitely. The attack follows.

1. S does not know the fingerprinting pattern based on L_B that was selected by B to be embedded into content X to form X_B. However, S does have the copies of X_B^0 embedded with F_B^0, and of X_B^1 embedded with F_B^1.

   Proceeding frame by frame in sequence, S compares each frame of X_B with each frame of X_B^0 and of X_B^1. Since each frame is processed independently (like in electronic code book way), S will successfully obtain the fingerprinting pattern L_B.
2. The rest of the attack steps is similar to the steps 2 to 6 of the first attack above.

Our first attack exploits the fact that L_B can be bruteforced in practice, and that T_B can be used for verifying these guesses. Even if L_B is too long to be bruteforced in practice (but this is not the case for the Yong-Lee scheme), our second attack still applies. It exploits the fact that the seller S knows the fingerprint set $\{F_B^0, F_B^1\}$ used to embed into the content thus it can know the fingerprinting pattern chosen by the buyer B by simple frame comparison once a copy of the fingerprinted content X_B is available. In both attacks, the major flaw we exploit is the same for which we exploited in our attack on Seller Security in the previous subsection: that the only thing that binds to the buyer B is $sig_{x_1}(T_B)$, which is independent of the content bought by B. This allows the seller S to transplant the same fingerprinting pattern to any other content for as many times as it wishes to frame B.

5 Concluding Remarks

The Yong-Lee BSW scheme attempts to eliminate the inefficiency of some existing BSW schemes by using symmetric key encryption and commutative encryption. The flaws that we have demonstrated on this scheme do not stem from the use of these encryption methods, but exploits the fact that the scheme was not sufficiently binding a buyer to the content. This causes a buyer to repudiate and thus get away with illegal redistribution of bought content, breaking seller security. This also makes it easier for a seller to transplant a buyer's fingerprint to other contents for framing, thus breaking buyer security.

Our results show that the Yong-Lee scheme does not offer the security for which it is designed to provide, and therefore leaves doubts on the design of this scheme, considering the state of the art of BSW schemes thus far, and the fact that the Yong-Lee BSW scheme is a fairly recent proposal that should have taken the state of the art into its design consideration. We caution against simple fixes that patch our attacks in this paper since experience has shown that the break-and-fix cycle loops indefinitely, for instance see [17,18,19,30,31,32] where attacks were applied to protocols [8,9,10,11,20,22,6,36] that improved on existing ones. We suggest instead, that if BSW schemes are required, to consider other schemes like [24,38] that have not yet been shown to fall to any attacks that counter their design goals.

Acknowledgement

We are grateful for suggestions by anonymous referees of IWDW '07 to give emphasis in our discussion on whether the Yong-Lee scheme can be fixed to counter our attacks. This has been duly treated in the Concluding Remarks above. Part of this work was done while the first author was attending the smoothly run IACR-sponsored Eurocrypt 2007 conference in Barcelona. We thank God for His many blessings.

References

1. Bao, F., Deng, R.H., Feng, P.: An Efficient and Practical Scheme for Privacy Protection in the E-commerce of Digital Goods. In: ICICS 2003. LNCS, vol. 2836, pp. 162–170. Springer, Heidelberg (2001)
2. Blakley, G., Meadows, C., Purdy, G.B.: Fingerprinting Long Forgiving Messages. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 180–189. Springer, Heidelberg (1986)
3. Boneh, D., Shaw, J.: Collusion-secure Fingerprinting for Digital Data. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 452–465. Springer, Heidelberg (1995)
4. Anderson, R.: Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley Publishing, U.S (2001)
5. Brickell, E.F., Yacobi, Y.: On Privacy Homormorphisms. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 117–125. Springer, Heidelberg (1988)

6. Byun, J.W., Lee, D.H., Lim, J.: Efficient and Provably Secure Client-to-Client Password-based Key Exchange Protocol. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 830–836. Springer, Heidelberg (2006)
7. Chaum, D.: An Improved Protocol for Demonstrating Possession of Discrete Logarithms and some Generalizations. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 127–141. Springer, Heidelberg (1988)
8. Chang, C.C., Chung, C.Y.: An Enhanced Buyer-Seller Watermarking Protocol. In: Proceedings of ICCT 2003, pp. 1779–1783 (2003)
9. Cheung, S.C., Leung, H.F., Wang, C.: A Commutative Encrypted Protocol for the Privacy Protection of Watermarks in Digital Contents. In: Proceedings of HICSS-37 (January 2004)
10. Choi, J.-G., Sakurai, K., Park, J.H.: Does It Need Trusted Third Party? Design of Buyer-Seller Watermarking Protocol without Trusted Third Party. In: Zhou, J., Yung, M., Han, Y. (eds.) ACNS 2003. LNCS, vol. 2846, pp. 265–279. Springer, Heidelberg (2003)
11. Choi, J.-G., Park, J.H.: A Generalization of an Anonymous Buyer-Seller Watermarking Protocol and Its Application to Mobile Communications. In: Cox, I., Kalker, T., Lee, H.-K. (eds.) IWDW 2004. LNCS, vol. 3304, pp. 232–243. Springer, Heidelberg (2005)
12. Choi, J.-G., Park, J.H., Kwon, K.R.: Analysis of COT-based Fingerprinting Schemes: New Approaches to Design Practical and Secure Fingerprinting Scheme. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 253–265. Springer, Heidelberg (2004)
13. Cox, I.J., Doerr, G.J., Furon, T.: Watermarking is Not Cryptography. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 1–15. Springer, Heidelberg (2006)
14. Cox, I.J., Kilian, J., Leighton, T., Shamoon, T.: Secure Spread Spectrum Watermarking for Images, Audio and Video. IEEE Trans. on Image Processing 6(12), 1673–1678 (1997)
15. Domingo-Ferrer, J.: Anonymous Fingerprinting based on Committed Oblivious Transfer. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 43–52. Springer, Heidelberg (1999)
16. Domingo-Ferrer, J.: Anonymous Fingerprinting of Electronic Information with Automatic Identification Redistributors. IEE Electronics Letters 43(13), 1303–1304 (1998)
17. Goi, B.-M., Phan, R.C.-W., Chuah, H.-T.: Cryptanalysis of Two Non-Anonymous Buyer-Seller Watermarking Protocols for Content Protection. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part I. LNCS, vol. 4705, pp. 951–960. Springer, Heidelberg (2007)
18. Goi, B.-M., Phan, R.C.-W., Siddiqi, M.U.: Cryptanalysis of a Generalized Anonymous Buyer-Seller Watermarking Protocol of IWDW 2004. In: Enokido, T., Yan, L., Xiao, B., Kim, D.Y., Dai, Y.-S., Yang, L.T. (eds.) EUC-WS 2005. LNCS, vol. 3823, pp. 936–944. Springer, Heidelberg (2005)
19. Goi, B.-M., Phan, R.C.-W., Yang, Y., Bao, F., Deng, R.H., Siddiqi, M.U.: Cryptanalysis of Two Anonymous Buyer-Seller Watermarking Protocols and An Improvement for True Anonymity. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 369–382. Springer, Heidelberg (2004)
20. Ju, H.S., Kim, H.J., Lee, D.H., Lim, J.I.: An Anonymous Buyer-Seller Watermarking Protocol with Anonymity Control. In: Lee, P.J., Lim, C.H. (eds.) ICISC 2002. LNCS, vol. 2587, pp. 421–432. Springer, Heidelberg (2003)

21. Katzenbeisser, S.: On the Integration of Watermarks and Cryptography. In: Kalker, T., Cox, I., Ro, Y.M. (eds.) IWDW 2003. LNCS, vol. 2939, pp. 50–60. Springer, Heidelberg (2004)
22. Kim, J., Kim, S., Kwak, J., Won, D.: Cryptanalysis and Improvement of Password-Authenticated Key Exchange Scheme between Clients with Different Passwords. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3043, pp. 895–902. Springer, Heidelberg (2004)
23. Kuribayashi, M., Tanaka, H.: A New Anonymous Fingerprinting Scheme with High Enciphering Rate. In: Pandu Rangan, C., Ding, C. (eds.) INDOCRYPT 2001. LNCS, vol. 2247, pp. 30–39. Springer, Heidelberg (2001)
24. Lei, C.-L., Yu, P.-L., Tsai, P.-L., Chan, M.-H.: An Efficient and Anonymous Buyer-Seller Watermarking Protocol. IEEE Trans. on Image Processing 13(12) (December 2004)
25. Memon, N., Wong, P.W.: A Buyer-Seller Watermarking Protocol. IEEE Trans. on Image Processing 10(4) (April 2001)
26. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, U.S (1997)
27. Pfitzmann, B., Sadeghi, A.R.: Coin-Based Anonymous Fingerprinting. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 150–164. Springer, Heidelberg (1999)
28. Pfitzmann, B., Schunter, M.: Asymmetric Fingerprinting. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 84–95. Springer, Heidelberg (1996)
29. Pfitzmann, B., Waidner, M.: Anonymous Fingerprinting. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 88–102. Springer, Heidelberg (1997)
30. Phan, R.C.-W., Goi, B.-M.: Cryptanalysis of an Improved Client-to-Client Password-Authenticated Key Exchange (C2C-PAKE) Scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 33–39. Springer, Heidelberg (2005)
31. Phan, R.C.-W., Goi, B.-M.: Cryptanalysis of the N-Party Encrypted Diffie-Hellman Key Exchange using Different Passwords. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 226–238. Springer, Heidelberg (2006)
32. Phan, R.C.-W., Goi, B.-M.: Cryptanalysis of Two Provably Secure Cross-Realm C2C-PAKE Protocols. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 104–117. Springer, Heidelberg (2006)
33. Phan, R.C.-W., Ling, H.-C.: Flaws in Generic Watermarking Protocols based on Zero-Knowledge Proofs. In: Cox, I., Kalker, T., Lee, H.-K. (eds.) IWDW 2004. LNCS, vol. 3304, pp. 184–191. Springer, Heidelberg (2005)
34. Qiao, L., Nahrstedt, K.: Watermarking Schemes and Protocols for Protecting Rightful Ownership and Customer's Rights. Journal of Visual Communication and Image Representation 9(3), 194–210 (1998)
35. Trappe, W., Wu, M., Liu, K.: Collusion-resistant Fingerprinting for Multimedia. In: Proceedings of IEEE ICASSP 2002, pp. 3309–3312 (2002)
36. Yin, Y., Bao, L.: Secure Cross-Realm C2C-PAKE Protocol. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 395–406. Springer, Heidelberg (2006)
37. Yong, S., Lee, S.-H.: An Efficient Fingerprinting Scheme with Symmetric and Commutative Encryption. In: Barni, M., Cox, I., Kalker, T., Kim, H.-J. (eds.) IWDW 2005. LNCS, vol. 3710, pp. 54–66. Springer, Heidelberg (2005)
38. Zhang, J., Kou, W., Fan, K.: Secure Buyer-Seller Watermarking Protocol. IEE Proceedings - Information Security 153(1), 15–18 (2006)

Attack Analysis for He & Wu's Joint Watermarking/Fingerprinting Scheme

Hans Georg Schaathun

University of Surrey
Department of Computing

Abstract. We introduce two novel collusion attacks against digital fingerprinting using additive spread-spectrum watermarks. These attacks demonstrate that the He-Wu fingerprinting system is considerably less secure against collusive attacks than suggested by the original paper. In addition to causing error rates above 85% at the decoder with as few as 8 colluders, one of our two attacks give copies with less distortion (measured by Euclidean distance) than the fingerprinted copies originally distributed.

1 Introduction

Unauthorised copying is a major worry for many copyright holders. As digital equipment enables perfect copies to be created on amateur equipment, many are worried about lost revenues, and steps are introduced to reduce the problem. Technology to prevent copying has been along for a long time, but it is often controversial because it not only prevents unauthorised copying, but also a lot of the legal and fair use.

A different approach to the problem is to deter potential offenders by application of forensic technology. These solutions do not prevent copying, but in the event that illegal copies are detected, they allow identification of the offenders, who can then be prosecuted. If penalties are sufficiently high, potential pirates are unlikely to accept the risk of being caught.

One forensic solution is digital fingerprinting, first proposed by Wagner [6]. Each copy of the copyrighted file is marked by hiding a fingerprint identifying the buyer. Illegal copies can then be traced back to one of the legitimate copies and the guilty user be identified. Obviously, the marking must be made such that the user cannot remove the fingerprint without ruining the file. Techniques to hide data in a file in such a way are known as robust watermarking. All references to watermarking (WM) in this paper refers to robust watermarking.

A group of users can compare their individual copies and observe differences caused by the different fingerprints embedded. By exploiting this information they can mount so-called *collusive attacks*. There is a growing literature on collusion-secure fingerprinting, both from mathematical and abstract and from practical view-points.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 45–59, 2008.

The goal of this paper is to make a critical review of the security of a recently proposed system by He and Wu [2], and to discuss the efficiency of various collusive attacks. In particular, we consider variations over the so-called minority choice attack, which has been largely ignored by the watermarking community and by [2] in particular. The He-Wu scheme serves as a case study for the evaluation of this attack.

We will discuss the relevant fingerprinting models in Section 2, and present the GRACE fingerprinting system of [2] in Section 3. In Section 4, we introduce our novel attacks, and in Section 5, we report our simulations and analysis. Section 6 presents our conclusions.

2 Fingerprinting Models

Digital fingerprinting is often viewed as a layered system. In the fingerprinting (FP) layer, each user is identified by a codeword $\mathbf{c}$, i.e. an n-tuple of symbols from a discrete q-ary alphabet. If there are M codewords (users), we say that they form an $(n, M)_q$ code.

In the watermarking (WM) layer, the copyrighted file is divided into n segments. When a codeword $\mathbf{c}$ is embedded, each symbol of $\mathbf{c}$ is embedded independently in one segment.

2.1 Digital Watermarking

A wide range of different WM techniques have been proposed. Following past works [2,8], we will limit our study to non-adaptive, additive watermarks. It is commonly argued that in most fingerprinting applications, the original file will be known by the decoder, so that non-blind detection can be used [2].

We view the copyrighted file as a signal $\mathbf{x} = (x_1, \ldots, x_N)$, called the *host signal*, of real or floating-point values x_i. Since we use non-adaptive, additive watermarking, the message (or customer ID) to be embedded is mapped to a watermark signal $\mathbf{w} = (w_1, \ldots, w_N)$ which is independent of $\mathbf{x}$. The watermarked copy is the signal $\mathbf{y} = \mathbf{x} + \mathbf{w}$. The watermark $\mathbf{w}$ is designed such that $\mathbf{y}$ and $\mathbf{x}$ are perceptually equivalent.

The adversary, the copyright pirates in the case of fingerprinting, will try to disable the watermark by creating an attacked copy $\mathbf{z}$ which is perceptually equivalent to $\mathbf{y}$, but where the watermark cannot be correctly interpreted.

The watermark decoder takes the attacked signal $\mathbf{z}$. A non-blind decoder will subtract the original host signal, to obtain an attacked watermark signal $\mathbf{v} = \mathbf{z} - \mathbf{x}$. We will assume a correlation decoder, which returns the message associated with the watermark $\mathbf{w}$ which maximises the correlation $\mathbf{w} \cdot \mathbf{v}$.

A WM system can be used for fingerprinting, either alone (e.g.) [8] or in conjunction with a fingerprinting code [2]. In [8], each user u is associated with a watermark $\mathbf{w}_u$ where each sample w_i is chosen independently using a normal distribution.

In the model of [2], each user u is associated with a codeword $\mathbf{c}_u = (c_1^{(u)}, \ldots, c_n^{(u)})$ from an $(n, M)_q$ FP code. In the WM layer, q orthogonal watermarking signals $\mathbf{w}_s = (w_1^{(s)}, \ldots, w_m^{(s)})$ are used, where $w_i^{(s)} = \pm 1$ and $m \cdot n = N$. Their decoder views the two layers together, in a way equivalent to using soft-decision decoding. A watermark $\mathbf{w}'_u$ associated with user u, by concatenating the watermarks associated with the code symbols, i.e.

$$\mathbf{w}'_u = \mathbf{w}_{c_1^{(u)}} || \mathbf{w}_{c_2^{(u)}} || \mathbf{w}_{c_3^{(u)}} || \ldots || \mathbf{w}_{c_n^{(u)}} .$$

Based on this model, [2] present two novel improved solutions which we will describe in Section 3, namely group-based joint coding and embedding, and subsegment permutation.

2.2 Cut-and-Paste Attacks and Collusion-Secure Codes

One of the most studied attacks on digital fingerprinting is the *cut-and-paste* attack (aka. interleaving attack). A collusion of copyright pirates can compare their copies, and they will find that some segments differ. Such segments have ostensibly been used to hide a part of the fingerprint. By cutting and pasting segments from different copies, the colluders can produce a copy with a hybrid fingerprint distinct from those of the colluders.

Collusion-secure codes are designed to permit correct tracing in the presence of the cut-and-paste attack. A substantial literature is evolving on this topic. Usually an abstract mathematical model is used. The interface to the watermarking layer is defined as a marking assumption. The most commonly used assumption originates in [1], and says that every position (segment) of a hybrid fingerprint will match at least one of the fingerprints of the colluders.

Various strategies are available to colluders employing a cut-and-paste attack. Common strategies include the following.

Equal shares. Divide the segments into groups groups of equal size, one for each colluder. All segments in a group are copied from the corresponding colluder.

Random. For each segment, choose a colluder fingerprint uniformly at random and copy the segment therefrom.

Minority choice. For each segment, the value occurring the fewest times in the corresponding segment of the colluders' fingerprints is used.

Majority choice. For each segment, the value occurring the most times in the corresponding segment of the colluders' fingerprints is used.

Experimental work tends to focus on equal shares or random strategies. However, it is well known in the theoretical literature, that minority choice is most effective when closest neighbour or correlation decoding is used. This is because minority choice minimises the average similarity between colluder fingerprints and the hybrid fingerprint.

2.3 Attacks on Joint Watermarking/Fingerprinting

There are important differences between the models commonly used for watermarking and for fingerprinting. Some of these differences result in different views on possible attacks, which we consider below.

Collusion-secure codes use discrete alphabets, whereas the host and watermark signals are numerical. Hence, in the joint model, a collusion attack can produce hybrid fingerprints which are numerical functions of the colluder fingerprints. A much studied example is the *averaging* attack, where the colluders produce an unauthorised copy by taking the average of the copies they have.

Example 1. *In the joint model, a collusion seeing* $+1$ *and* -1*, can produce a hybrid with a* 0*. In the (more abstract) FP model, the two symbols seen are just arbitrary symbols, say* A *and* B*, and the pirates can only output either* A *or* B*. The average of* A *and* B *is not defined.*

The Marking Assumption relies on each segment being a unit, and that the pirates cut and paste entire segments together. However, in most joint systems, the each watermark has to use a large number of samples to be sufficiently robust. Thus, the colluders could equally well cut and paste individual samples, effectively mounting a cut-and-paste attack on the WM layer.

The averaging attack, as well as a range of non-linear attacks, were studied in [8]. The attacks considered can be described as follows, where P is the set of fingerprints held by the collusions and $\mathbf{z}$ is the hybrid fingerprint produced by the attack.

$$\begin{aligned}
\text{Average:} &\quad \bar{z}_i = \frac{1}{t}\sum_{\mathbf{y}\in P} y_i.\\
\text{Minimum:} &\quad z_i^{\min} = \min_{\mathbf{y}\in P} y_i.\\
\text{Maximum:} &\quad z_i^{\max} = \max_{\mathbf{y}\in P} y_i.\\
\text{Median:} &\quad z_i^{\mathrm{med}} = \operatorname*{median}_{\mathbf{y}\in P} y_i.\\
\text{Midpoint (MinMax):} &\quad z_i^{\mathrm{mid}} = (z_i^{\min} + z_i^{\max})/2.\\
\text{Modified negative:} &\quad z_i^{\mathrm{modneg}} = z_i^{\min} + z_i^{\max} - z_i^{\mathrm{mid}}.\\
\text{Randomised negative:} &\quad z_i^{\mathrm{rndneg}} = \begin{cases} z_i^{\min}\text{with probability } p,\\ z_i^{\max}\text{with probability } 1-p,\end{cases}
\end{aligned}$$

It was assumed in [8], that p for the randomised negative attack be independent of the signals $\{y_i\}$. None of the above attacks adapt to the signal.

There are two important characteristics for the evaluation of fingerprinting attacks.

Success rate. The attack succeeds when the watermark decoder does not return any of the colluders. Obviously, we want to maximise the rate of success.

Distortion. The unauthorised copy has to pass as the original, so it should be as close as possible to the unknown signal $\mathbf{x}$ perceptually.

It is natural to expect low distortion from the average, median, and midpoint attacks. The pirate collusion is likely to include both positive and negative fingerprint signals. Consequently, these attacks are likely to produce a hybrid which is closer to the original sample than any of the colluder fingerprints. On the contrary, the maximum, minimum, and randomised negative attacks would tend to give a very distorted hybrid, by using the most distorted version of each sample. This is experimentally confirmed in [8].

Not surprisingly, the most effective attacks are the most distorting. The most effective attack according to [8] is the randomised negative, but the authors raise some doubt that it be practical due to the distortion.

2.4 Evaluation Methodology

The performance of existing fingerprinting schemes and joint WM/FP schemes have been analysed experimentally or theoretically. Very few systems have been studied both experimentally and theoretically. In the cases where both theoretical and experimental analysis exist, there is a huge discrepancy between the two.

It is not surprising that theoretical analyses are more pessimistic than experimental ones. An experimental simulation (e.g. [2]) have to assume one (or a few) specific attack(s). An adversary who is smarter (or more patient) than the author and analyst may very well find an attack which is more effective than any attack analysed. Thus, the experimental analyses give lower bounds on the error rate of the decoder, by identifying an attack which is good enough to produce the stated error rate.

The theoretical analyses of the collusion-secure codes of [1,5,3] give mathematical upper bounds on the error rate under any attack provided that the appropriate Marking Assumption holds. Of course, attacks on the WM layer (which is not considered by those authors) may very well break the assumptions and thereby the system. Unfortunately, little work has been done on theoretical upper bounds for practical, joint WM/FP schemes.

In any security application, including WM/FP schemes, the designer has a much harder task than the attacker. The attacker only needs to find one attack which is good enough to break the system, and this can be confirmed experimentally. The designer has to find a system which can resist every attack, and this is likely to require a complex argument to be assuring.

This paper will improve the lower bounds (experimental bounds) for the He-Wu joint WM/FP system, by identifying adaptive, non-linear attacks, which are more effective than those originally studied. These attacks are likely to be effective against other joint schemes as well.

3 GRACE Fingerprinting

We gave a brief overview of joint WM/FP in Section 2.1. The He-We solution is based on two additional features, namely Group-Based Joint Coding and Embedding Fingerprinting (GRACE), and subsegment permutations, which we introduce below.

Group-based fingerprinting assumes that one can divide the users into groups such that users are more likely to collude within a group than across several groups. A group could for instance be a limited geographical area, assuming that organising a world-wide collusion is more difficult and/or expensive than colluding with your town-mates.

3.1 The Design

In the FP layer, we use a q-ary $[n, k]$ code C, with the so-called c-traceability property [4]. The code is linear of dimension k with codewords of length n. Each user is associated with one codeword (fingerprint). Let $D \subset C$ be a $[n, 1, n]$ repetition code, that is a subcode generated by a codeword of Hamming weight n. The system supports q^{k-1} disjoint groups of size q. The total number of users is $M = q^k$.

Fingerprints are assigned to users so that fingerprints in the same coset $D + \mathbf{x}$ in C belong to the same group. Consequently, the minimum Hamming distance between two fingerprints in the same group is equal to n, which is the maximum possible.

Each group is also associated with a codeword $\mathbf{c}' \in D$ over an alphabet of q^{k-1} symbols. Thus, each user will have a user fingerprint $\mathbf{c} \in C$ and a group fingerprint $\mathbf{c}' \in D$. Both these fingerprints will be embedded on top of each other in the WM layer.

The embedding uses $2q^{k-1}$ orthogonal sequences $\{\mathbf{u}_1, \ldots, \mathbf{u}_q; \mathbf{a}_1, \ldots, \mathbf{a}_{q^{k-1}}\}$. Consider a user associated with fingerprints

$$\mathbf{c} = (c_1, c_2, \ldots, c_n) \in C,$$
$$\mathbf{c}' = (c'_1, c'_2, \ldots, c'_n) \in A.$$

A WM signal $\mathbf{w}_u$ is constructed by concatenating n segments s_i defined as

$$s_i = \sqrt{1-\rho}\mathbf{u}_{c_i} + \sqrt{\rho}\mathbf{a}_{c'_i},$$

where ρ is used to adjust the relative energy of group and user information. The experiments in [2] used $\rho = 1/7$. The WM signal $\mathbf{w}$ is added to the host signal $\mathbf{x}$ as usual.

In the actual implementation tested in [2], the fingerprinting code C was a $[30, 2, 29]$ Reed-Solomon Code over $\mathsf{GF}(32)$. This code is 5-traceability code, meaning that it is collusion-secure under the Marking Assumption of [1] for collusions of size 5 or less. In the watermarking layer, 64 orthogonal sequences of length 1000 were used, requiring a total of 30 000 samples for embedding.

3.2 GRACE Decoding

The GRACE decoder will first identify suspicious groups. Suppose group j is assigned $\mathbf{c}' \in D$. Define the associated group WM signal $\mathbf{g}_j$ as

$$\mathbf{g}_j = \mathbf{a}_{c'_1}||\mathbf{a}_{c'_2}||\ldots||\mathbf{a}_{c'_n}.$$

A group j is deemed to be suspicious if $\mathbf{g}_j \cdot \mathbf{v} > \tau$, where τ is some threshold. Let $\mathcal{S}$ be the set of users who belong to a suspicious group. The decoder will return the user u solving

$$\max_{u \in \mathcal{S}} \mathbf{w}_u \cdot \mathbf{v}.$$

3.3 Subsegment Permutation

The second feature of the construction in [2], is that the segments are divided into subsegments, and the entire set of subsegments is permuted according to a secret key. The effect of this is that a collusion cannot mount a segment-wise cut-and-paste attack, because they have no way to identify the subsegments belonging to the same segment.

Observe that the subsegment permutation has no effect on sample-wise attacks. The only attacks it can counter are those using information about the segment structure. The only affected attack in this paper is the segment-wise cut-and-paste attack.

3.4 Original Performance Analysis

He and Wu analysed their scheme based on simulations of averaging and cut-and-paste attacks in combination with noise. Cut-and-paste was more effective than averaging. They did not analyse the performance absent noise. We quote their results for the cut-and-paste attack (using subsegment permutations) at a WNR of 0dB.

Drawing the pirates randomly from two groups, they had an error rate $\epsilon \approx 0.00$ up to 30 pirates. Drawing the pirates randomly across all groups, they had $\epsilon < 0.01$ up to 26 pirates and $\epsilon \approx 0.25$ at 30 pirates. The experiments without subsegment permutations had higher error rates.

4 The Novel Attacks

Let $\mathbf{w}_u$ be the watermark identifying user u, and let $\mathbf{v} = \mathbf{z} - \mathbf{x}$ be the hybrid watermark generated by the collusion. The correlation decoder calculates the heuristic

$$h_u = \mathbf{v} \cdot \mathbf{w}_u = \sum_{i=1}^{N} v_i \cdot w_i^{(u)},$$

for each u and returns the user(s) u with the largest h_u.

In order to avoid detection, the pirates should attempt to minimise $\max_{u \in P} h_u$. Without complete knowledge of the original host $\mathbf{x}$ and the watermark signals used, an accurate minimisation is intractable. However, attempting to minimise $\bar{h} = \operatorname{avg}_{u \in P} h_u$ is a reasonable approximation, and this can be done by minimising sample by sample, $\operatorname{avg}_{u \in P} v_i \cdot w_i^{(u)}$.

4.1 The Minority Extreme Attack

If only two values occur in the watermark signals, the minority choice attack can be applied directly. The pirates seeing to different values of sample i, will use the one occurring the fewest times. Using GRACE, however, four different values occur, $\pm\sqrt{\rho} \pm \sqrt{1-\rho}$. Gaussian fingerprints [8] could give any number of distinct sample values.

Example 2. *Suppose the four possible values occur with the following frequencies*

$$\begin{aligned} a_1 &= x_i - \sqrt{1-\rho} - \sqrt{\rho} : 7 \textit{ times,} \\ a_2 &= x_i - \sqrt{1-\rho} + \sqrt{\rho} : 1 \textit{ times,} \\ a_3 &= x_i + \sqrt{1-\rho} - \sqrt{\rho} : 0 \textit{ times,} \\ a_4 &= x_i + \sqrt{1-\rho} + \sqrt{\rho} : 2 \textit{ times.} \end{aligned}$$

The minority choice a_2 will make a positive contribution to the correlation for 8 out of 10 colluder fingerprints, those with a_1 or a_2. If the pirates on the other hand choose a_4, 8 fingerprints will have negative correlation and only two will have positive correlation.

Remark 1. *In the case of GRACE, the colluders can in many cases recover x_i and remove the watermark completely from a sample. This is always true if they see four distinct values, as they can take the average of the minimum and the maximum. It is also true if they know ρ and see a_2 and a_3 or a_1 and a_4. Again x_i is the average of the two values.*

Since the remark is only applicable when there is a finite number of possible values, we focus on the idea of the example. When the average is close to the minimum, we choose the maximum, and vice versa. In mathematical notation, we write

$$\text{Minority Extreme (MX):} \qquad z_i^{\text{MX}} = \begin{cases} z_i^{\min} \text{ if } z_i^{\text{avg}} > z_i^{\text{mid}}, \\ z_i^{\max} \text{ if } z_i^{\text{avg}} < z_i^{\text{mid}}, \end{cases}$$

and we shall see later that the experimental performance is good.

4.2 The Moderated Minority Extreme Attack

The above attack is expected to give distortion similar to that of the randomised negative, as it always chooses an extreme value. This problem has a simple fix.

Consider the difference $Z = z_i^{\text{avg}} - z_i^{\text{mid}}$. If $|Z|$ is small, it probably makes little difference to $\bar{h}$ whether $z_i = z_i^{\text{max}}$ or $z_i = z_i^{\text{min}}$. However, the distortion caused is likely to be the same regardless of Z. A solution is to use the average value when $|Z|$ is small, and the minority extreme attack when $|Z|$ is large. In other words,

$$\text{Moderated Minority Extreme (MMX):} \quad z_i^{\text{MMX}} = \begin{cases} z_i^{\text{min}} \text{ if } Z > \theta, \\ z_i^{\text{avg}} \text{ if } \theta > Z > -\theta, \\ z_i^{\text{max}} \text{ if } Z < -\theta, \end{cases}$$

where θ is some threshold. Again, the experimental analysis in the next section will confirm our intuition.

5 Experimental Analysis

We have tried to replicate the algorithms from [2] as closely as possible. The authors did not specify how the orthogonal sequences are selected. To simplify coding, we used slightly longer sequences, 1024 bits rather than 1000, which should slightly improve the performance. These sequences were drawn randomly from a simplex code, mapping $0 \mapsto -1$.

For simplicity, we measure the distortion as the squared Euclidean distance (or power) $||\mathbf{y} - \mathbf{x}||^2$. This is intended to be a general estimate independent of the actual type of medium, and give a relative impression of distortion for the various attacks compared to the distortion in the fingerprinted copies. An exact measure would require a perceptual model, and would restrict the analysis to specific media.

We consider two different cases.

Two groups. The pirate collusion is formed by t pirates drawn uniformly at random from the first two groups.

Any group. The pirate collusion is formed by t pirates drawn uniformly at random from any group.

Group-based systems are only intended to have advantages in the two-group case, but [2] claim good performance in both cases, so we will discuss both.

We stress that we study exclusively the collusive attacks, and no additional noise attacks have been considered. This is for the simple reason that the two novel attacks are so effective that additional noise would not make any difference to the error rates.

We did not implement subsegment permutations, such that the performance under segment-wise attacks of our implementation is inferior to that of He-Wu's original implementation. However, the other attacks, being sample-wise, would be superfluous.

5.1 The Group Detection Threshold

The first question we investigate is about the group decoding threshold τ. There is no recommendation for τ to be found in [2], so we have made a set of simulations, presenting the group decoding heuristics in Table 1.

Table 1. Comparison of Group Decoding Heuristics for the attacks studied in [2]. The minimum, mean, and maximum are calculated per sample over all guilty groups. Innocent groups have a heuristic of 0 throughout. Averaging 500 samples per data point.

(a) Any group

	Segment-Wise			Averaging		
t	Min.	Mean	Max.	Min.	Mean	Max.
2	5207.18	5991.31	6775.45	6014.53	6014.53	6014.53
5	1365.46	2491.35	3791.40	2322.21	2514.18	3014.23
10	353.75	1343.31	2745.63	1161.11	1362.37	2240.94
20	10.84	779.62	2181.33	580.55	783.44	1595.36
30	0.00	593.75	1913.50	387.04	592.55	1315.92

(b) Two group

	Segment-Wise			Averaging		
t	Min.	Mean	Max.	Min.	Mean	Max.
2	8347.58	8754.75	9161.91	8777.97	8777.97	8777.97
5	4255.84	6153.87	8051.89	4351.83	6165.48	7979.13
10	4026.72	5805.53	7584.35	4495.81	5840.37	7184.93
20	4507.42	5805.53	7103.65	4774.47	5805.53	6836.60
30	4629.72	5805.53	6981.35	4976.50	5805.53	6634.56

Remark 2. *Observe that all innocent groups have a heuristic of 0. This is because all the sequences* $\mathbf{u}_i$ *and* $\mathbf{a}_i$ *are orthogonal. Calculating the correlation between the hybrid fingerprint and an innocent one, we find that each segments of the hybrid is a linear combination (either a single signal or an average of signals) of signals orthogonal to that of the innocent fingerprint.*

As we can see, a small positive threshold τ, would succeed in eliminating all innocent groups and retain almost all guilty groups under segment-wise cut-and-paste and under averaging.

One of the claims in [2] was that the cut-and-paste attack is more effective when it is applied to entire segments than individual samples. On the whole, our simulations confirm this, but one point should be noted. Comparing Tables 1 and 2, we see that for 20 and 30 pirates, the gap between group decoding heuristics of innocent and guilty groups is smaller, hence less noise would have to be introduced to cause errors in the group decoding step. The group decoding heuristics of innocent groups are more spread when the attack is sample-wise.

We choose a group decoding thresholds of $\tau = 350$ for initial simulations, which is sufficient to exclude all innocent groups with high probability under the attacks studied so far. We will reconsider the threshold in Section 5.4.

Table 2. Group Decoding Heuristics under sample-wise cut-and-paste with equal shares. Averaging 500 samples per data point.

	Guilty groups			Innocent groups		
t	Min.	Mean	Max.	Min.	Mean	Max.
2	5979.84	6047.89	6115.94	-254.15	-1.47	249.97
5	2168.74	2514.83	3163.88	-309.56	-1.03	303.33
10	936.94	1352.15	2287.28	-320.20	-2.00	318.50
20	317.09	784.58	1657.26	-296.10	0.55	294.48
30	112.82	597.55	1420.28	-276.91	0.52	276.96

5.2 The Minority Extreme Attack

Table 3 show group decoding heuristics under the minority extreme attack. The first we note is that 5-7 pirates drawn from all groups can completely wreck the group decoding. With five pirates, the mean group decoding heuristics are approximately equal for innocent and guilty groups. With seven pirates, all guilty groups have lower heuristic than any innocent group. In other words, if we require the system to be secure against collusions spanning 7 groups, we have to abandon the group decoding.

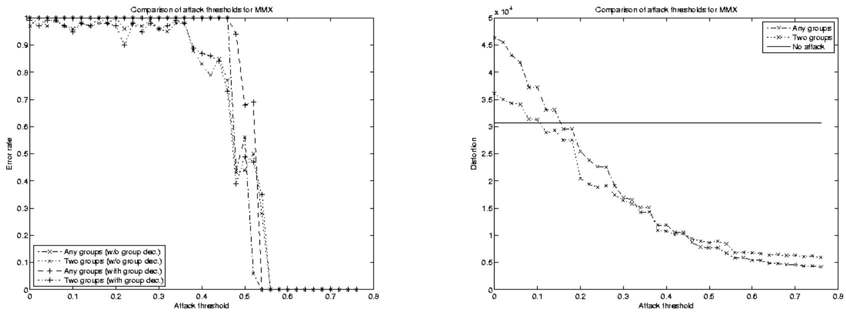
Table 3. Group Decoding Heuristics under the Minority Extreme attack. Averaging 500 samples per data point.

(a) t random pirates from any group

	Guilty groups			Innocent groups		
t	Min.	Mean	Max.	Min.	Mean	Max.
2	5909.77	5982.16	6054.54	-249.38	0.50	250.87
5	-273.61	57.31	371.23	-258.34	60.38	652.04
7	-2118.88	-1176.22	-742.62	-326.51	94.43	680.24
10	-2889.77	-1554.41	-1005.88	-355.37	129.77	577.65
20	-5374.96	-2472.27	-1434.85	-164.68	206.50	616.22
30	-5408.57	-2247.93	-985.22	-135.92	236.44	635.33

(b) t random pirates from two groups

	Guilty groups			Innocent groups		
t	Min.	Mean	Max.	Min.	Mean	Max.
5	2880.62	6169.04	9457.47	-254.89	52.38	519.69
7	3569.96	5943.76	8317.55	-228.04	72.44	504.76
10	1869.13	4576.03	7282.93	-196.99	102.61	507.94
20	913.69	4524.88	8136.07	-175.08	144.09	720.30
30	976.80	4797.54	8618.28	-193.41	161.30	823.89



(a) Error rate

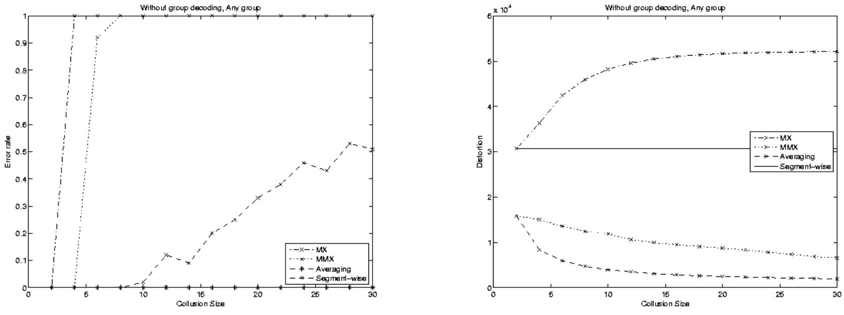
(b) Distortion

Fig. 1. Comparison of different thresholds θ for the Moderated Minority Extreme attack with 10 pirates. Each data point is the average of 100 samples. Distortion is shown without group decoding. Error rates without group decoding, and with $\tau = 350$.

5.3 The Moderated Minority Extreme Attack

Figure 1 shows a comparison of error rates and distortion levels for the MMX attack with different thresholds θ. Most interestingly, we note that for $\theta \geq 0.16$, we have less distortion than a fingerprinted copy prior to attack. For $\theta \leq 0.36$ we have more than 95% errors. If the pirates are drawn randomly across all groups, we get even more errors. Quite conservatively, we choose a threshold of $\theta = 0.4$ for our further simulations.

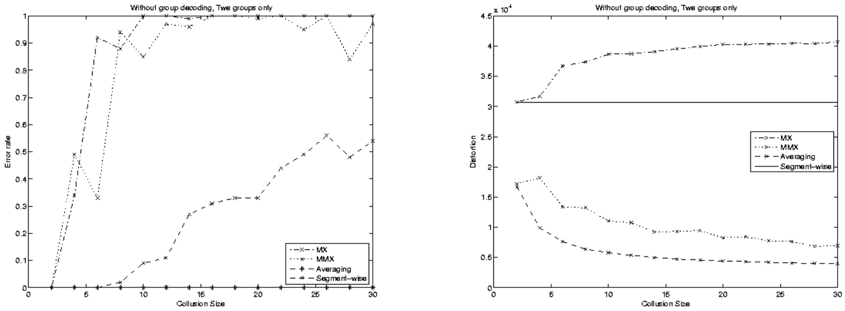
Comparison of the MX and MMX ($\theta = 0.4$) attacks with the attacks of [2] are shown in Figures 2, 3, and 4. The MX and MMX not only give much higher error rates than the old attacks, but they completely wreck the system with as few as eight pirates, giving error rates above 85%. This is a pure collusive attack, with no additional noise added.



(a) Error rate

(b) Distortion

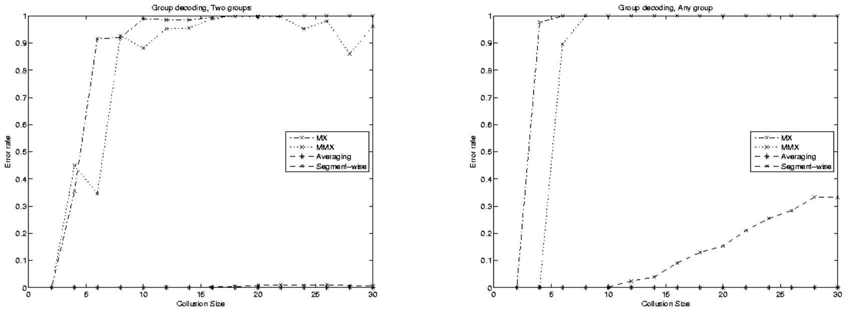
Fig. 2. Comparison of different attacks with pirates drawn from any group. Averaging 100 samples per data point. The decoder uses no group decoding.



(a) Error rate

(b) Distortion

Fig. 3. Comparison of different attacks with pirates drawn from two groups only. Averaging 100 samples per data point. The decoder uses no group decoding.



(a) Two groups

(b) Any group

Fig. 4. Comparison of error rates under different attacks using group decoding with $\tau = 350$. Averaging 1000 samples per data point.

The distortion is very much as expected. The distortion of the MX attack increase in the number of pirates, whereas the MMX and averaging attacks give decreasing distortion. Clearly the MMX attack give more distortion than the averaging attack, but yet much less than the fingerprinting caused in the first place.

5.4 Group Decoding Revisited

Figure 5 compares different group decoding thresholds against the MX and MMX ($\theta = 0.4$) attacks with seven pirates drawn from two groups. We observe that the group decoding threshold makes little difference under the MX attack. The sharp increase in errors starting at $\tau \approx 5500$ is due to all groups being excluded with high probability.

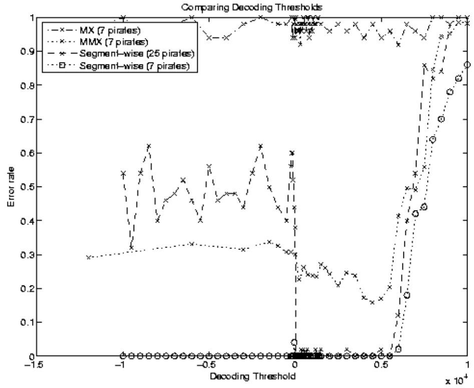
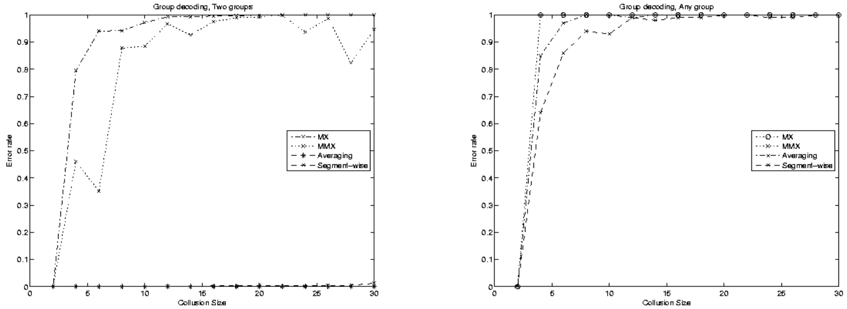


Fig. 5. Comparison of different group decoding thresholds τ under various attacks. Pirates drawn from two groups only. Using 50 samples per data point (1000 samples for MMX).



(a) Two groups

(b) Any group

Fig. 6. Comparison of error rates under different attacks using group decoding with $\tau = 4500$. Averaging 1000 samples per data point.

The figure shows that we should have $0 < \tau < 5500$, but otherwise the threshold seems to matter relatively little. There appears to be a slight dip in the error rate under the MMX attack around $\tau = 4500$, but the full simulation (Figure 6) for $t = 2 \dots 30$ failed to confirm this. The error rates for $\tau = 350$ and $\tau = 4500$ were practically indistinguishable in the two-group case (over 1000 samples per value of t). In the any-group case, $\tau = 350$ gave clearly the better performance.

6 Conclusion

We have taken the idea of minority choice attacks into the more practical model of joint WM/FP, and demonstrated that it can be effective. The result is an adaptive collusion attack which easily breaks the He-Wu joint WM/FP scheme (with the suggested parameters) while giving a hybrid copy which is less distorted than the original. One of the referees pointed out that He & Wu used Gaussian sequences in some of the tests. However, little detail was given for this variant, and we have not included it in our study at this stage.

The attack is quite general, and would be expected to give similar results for other fingerprinting schemes based on additive watermarking with correlation decoding (T statistic). Additional experiments would be needed to confirm this.

Later experiments have shown that the proposed attack is not effective against Gaussian fingerprints with Z statistic decoding and preprocessing [8]. However, the general point remains, that an adaptive attack more effective than the randomised attacks of [8] is likely to exist. To identify such adaptive attacks for decoding of [8] is an interesting open question.

It is interesting to note that past works on binary fingerprinting codes in abstract models, e.g. [1,3,5], never rely solely on closest neighbour or correlation decoders. Other combinatorial ideas are used in the inner code in order to resist attacks such as minority choice.

It would be interesting to construct joint WM/FP systems based on the collusion-secure codes from [3,5]. It remains an open question whether it is possible to construct a joint WM/FP scheme which is both practical and secure against optimised attacks.

References

1. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Trans. Inform. Theory 44(5), 1897–1905 (1998); presented in part at CRYPTO 1995
2. He, S., Wu, M.: Joint coding and embedding techniques for multimedia fingerprinting. IEEE Trans. Information Forensics and Security 1, 231–248 (2006)
3. Schaathun, H.G., Fernandez-Muüoz, M.: Boneh-Shaw fingerprinting and soft decision decoding. In: Information Theory Workshop, Rotorua, NZ (2005)
4. Staddon, J.N., Stinson, D.R., Wei, R.: Combinatorial properties of frameproof and traceability codes. IEEE Trans. Inform. Theory 47(3), 1042–1049 (2001)
5. Tardos, G.: Optimal probabilistic fingerprint codes. Journal of the ACM (2005); In: part at STOC 2003 (to appear), `http://www.renyi.hu/~tardos/fingerprint.ps`
6. Wagner, N.R.: Fingerprinting. In: Proceedings of the 1983 Symposium on Security and Privacy (1983)
7. Wu, M., Trappe, W., Wang, Z.J., Liu, K.J.R.: Collusion resistant fingerprinting for multimedia. IEEE Signal Processing Magazine (2004)
8. Zhao, H., Wu, M., Wang, Z.J., Liu, K.J.R.: Nonlinear collusion attacks on independent fingerprints for multimedia. IEEE Trans. Image Proc., 646–661 (2005)

Multiple Watermarking in Visual Cryptography

Hao Luo[1], Zhe-Ming Lu[2], and Jeng-Shyang Pan[3]

[1] Harbin Institute of Technology, Harbin 150001, P.R. China
[2] Sun Yat-sen University, Guangzhou 501725, P.R. China
[3] Kaohsiung University of Applied Sciences, Kaohsiung 807, Taiwan

Abstract. This paper proposes a scheme to hide two watermarks in visual secret sharing of a gray level image. One watermark is embedded during secret image halftoning, and the other is hidden during the halftone image encryption. A non-expansion visual secret sharing model is used to encrypt the halftone image. In watermark extraction, only Exclusive-OR operation is needed, and basic properties of conventional visual cryptography technique are still preserved. Experimental results show the effectiveness of our schemes.

1 Introduction

Visual cryptography is a paradigm introduced by Naor and Shamir [1] to encrypt a secret image into two or more than two random noise-like transparency images (also called shares, shadows). Via stacking some qualified shares, the secret content is revealed by human visual system (HVS). The key advantage of visual cryptography compared with traditional cryptography algorithms lies in that no computations or prior knowledge are required in secret image decryption. Generally, a visual secret sharing scheme is designed based on a $\{k, n\}$-threshold framework. That is, a secret image is encrypted into n transparencies distributed to n participants. Only when k or more than k transparencies are stacked, content of the secret image is visible. So far researchers have developed a lot of visual secret sharing (VSS) schemes for halftone [2], gray-scale [3, 4] and color [5, 6] images encryption. In most available VSS techniques, secret image decryption is just by a simple mechanism of stacking some transparencies. However, in some recent publications [7, 8] the original intent of performing the decryption by HVS with stacking transparencies is sacrificed by requiring a computer processing to reconstruct the image. Although they lose the ability of decryption by visual means, such encryption schemes have a large number of potential applications in the Internet world.

Digital watermarking [9] is a powerful technique for multimedia copyright protection, tamper detection, content authentication, etc. Researchers have proposed plenty of watermarking approaches based on the host multimedia of images, videos, audios, 3D meshes, etc. In this paper, we focus on incorporate watermarking techniques in visual cryptography. Namely, two different watermarks are embedded in transparencies when sharing a gray level image.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 60–70, 2008.

In fact, embedding extra data in transparencies has many potential applications. For example, these data may be some affiliation information of the secret image such as authors, processing time, content annotation and so on. Furthermore, these affiliation information can be used for transparency authentication, i.e., to judge a transparency is whether a legal one or not. Motivated by this, Fang et al [10] presented a scheme which hiding some extra data during secret image sharing. Fang et al's scheme has following properties. (1) As the input secret image is a halftone image, it can not be directly applied for gray level images sharing. (2) Only one watermark is embedded in two transparencies. (3) The secret sharing model used is the conventional $\{2, 2\}$ VSS. Size of transparency images is expanded to four times of the secret image. Consequently, it will greatly places heavy loading for the limited network bandwidth and the storage space.

Our proposed method aims to improve the inefficiencies of Fang et al's method in the above mentioned aspects. (1) The input secret image can be a gray level image instead of a halftone image. Thus our scheme is appropriate for more practical scenarios. (2) There are two watermarks instead of one are embedded and thus the hidden data capacity is enhanced. (3) A non-expansion visual secret sharing model instead of the conventional $\{2, 2\}$ VSS model is employed, i.e. size of transparencies is equal to that of the secret image. Therefore, less time and space are needed for transparencies transmission and storage.

In particular, in our scheme the input image is halftoned into a two-tone image first, and meanwhile one watermark is embedded during the halftoning processing. Second, the halftone image is encrypted based on a non-expansion visual secret sharing model and the second watermark is embedded during the encryption processing. The second watermark can be extracted using Exclusive-OR (XOR) operations on two transparency images.

The rest of this paper is organized as follows. Section 2 briefly reviews the conventional $\{2, 2\}$ VSS. Section 3 extensively describes the multiple watermarks embedding and extraction procedure. Experimental results and some discussions are given in Section 4, and Section 5 concludes the whole paper.

2 Conventional $\{2, 2\}$ VSS

In the conventional $\{2, 2\}$ VSS, a secret image is encrypted in two transparencies. In this model, six 2×2 codewords as shown in Fig. 1 (from c_1 to c_6) are involved. In order to produce random-noise like transparencies, each codeword contains two white pixels and two black pixels. Each secret pixel is encrypted into two 2×2 codewords. Fig. 2 illustrates the encryption and decryption strategy of conventional $\{2, 2\}$ VSS. Each white (black) pixel has six pairs of encryptions modes. In particular, a white pixel is encrypted by one of six combinations of (c_1, c_1), (c_2, c_2), (c_3, c_3), (c_4, c_4), (c_5, c_5), (c_6, c_6). Each combination appears with an equal probability. In a combination, the former codeword is assigned to one transparency T_1, and the latter is held by the other T_2. Similarly, a black pixel is encrypted by another six sets of combinations (c_1, c_2), (c_2, c_1), (c_3, c_4), (c_4, c_3), (c_5, c_6), (c_6, c_5). Hence, a $K_1 \times K_2$ secret image can be encrypted and shared in two $2K_1 \times 2K_2$ transparencies. An example of the $\{2, 2\}$ VSS is shown in Fig. 3. The halftone secret image "Cameraman" is of 256×256 pixels, and the two transparencies and their stacking results are of 512×512 pixels. Note in Fig. 3 the secret image and all the resultant images have been scaled down to the same size. It is clear that the size of the transparencies is four times of the secret image. Hence, a heavy loading for the limited network bandwidth and the storage space will be leaded by the conventional $\{2, 2\}$ VSS.

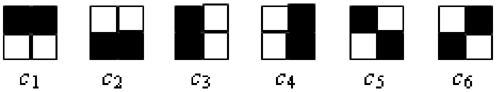


Fig. 1. Codewords of the conventional $\{2, 2\}$ VSS

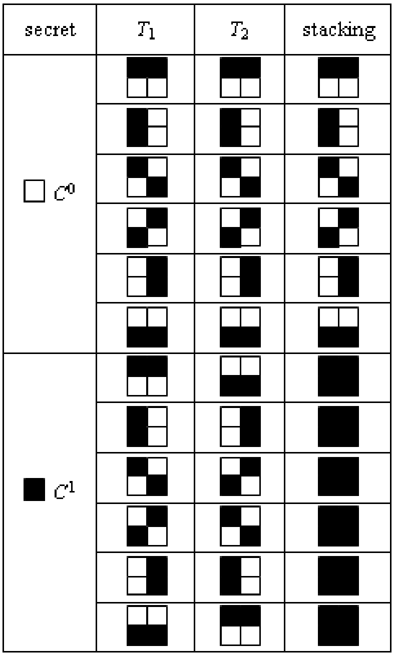


Fig. 2. Encryption and decryption strategy of the conventional $\{2, 2\}$ VSS

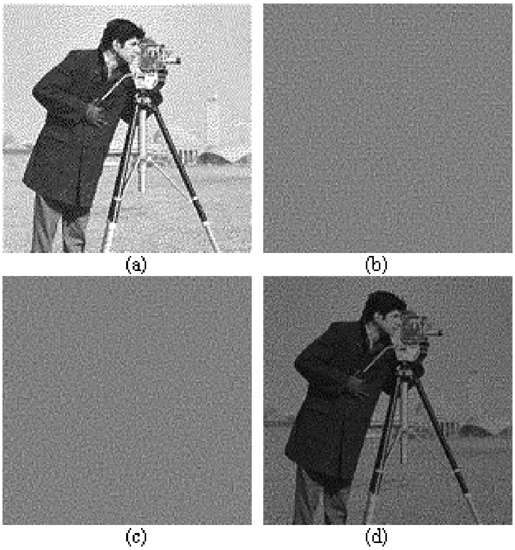


Fig. 3. Example of the conventional $\{2, 2\}$ VSS, (a) the secret image, (b) the transparency T_1, (c) the transparency T_2, (d) the stacking result of T_1 and T_2

3 Proposed Scheme

Suppose the secret image S is of the size $K_1 \times K_2$ pixels. As the scheme is based on a non-expansion VSS, both of the transparencies are of the size $K_1 \times K_2$ pixels. Suppose the two watermarks W_1 and W_2 are both binary images. The block diagrams of the proposed method are shown in Fig. 4.

In watermark embedding, the secret image is transformed into a halftone image using digital halftoning technique. Popular halftoning methods can be classified into two categories, i.e., ordered dithering and error diffusion. In our case, error diffusion is adopted because visual quality of a halftone image produced by it is better than that obtained by ordered dithering. The watermark W_1 is embedded during error diffusion halftoning the secret image. Specifically, we randomly select some pixels positions according to a pseudo random number generator with a key, and then replace their pixel values using the watermark W_1 sequence. The induced error is feeded back and diffused to the neighbor area. In this way, the introduced image quality distortion is not very evident.

After that, the watermarked halftone image WH is encrypted into two transparencies based on a non-expansion VSS. The second watermark embedding is incorporated in the non-expansion visual secret sharing processing. Thus two watermarked transparencies T_1 and T_2 are produced. Actually, there is an embedding order in the embedding processing, i.e., W_1 is first inserted and next W_2, and then WH is last encrypted.

In secret image decryption and watermark extraction, the basic properties of the conventional visual cryptography are still well maintained, including perfect

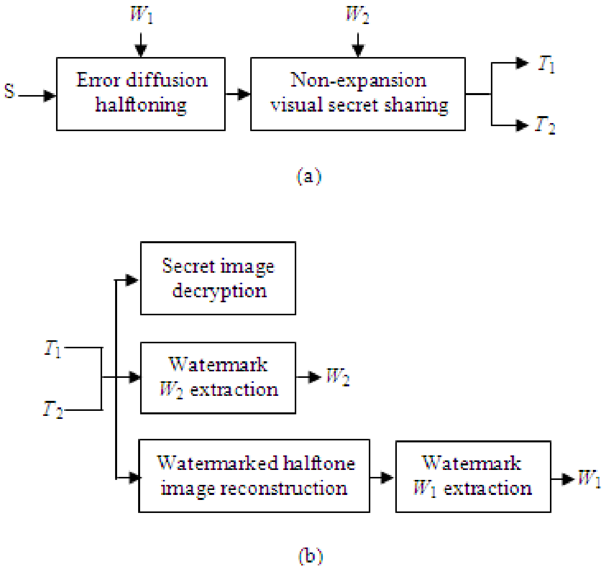

W_1
W_2
S
Error diffusion halftoning
Non-expansion visual secret sharing
T_1
T_2
(a)
Secret image decryption
T_1
T_2
Watermark W_2 extraction
W_2
Watermarked halftone image reconstruction
Watermark W_1 extraction
W_1
(b)

Fig. 4. Block diagrams of our scheme, (a) watermark embedding and secret image encryption, (b) watermark extraction and secret image decryption

security, simple decryption mechanism, etc. The second watermark is extracted by XOR operation of the two transparencies. To extract the first watermark, the halftone image must be precisely reconstructed in advance, and then the operation is reduced to retrieve pixel values in some positions selected by the pseudo random number generator with the same key.

3.1 Watermark Embedding in Halftoning

The first watermark W_1 embedding is incorporated into a halftone image generation, with details described below.

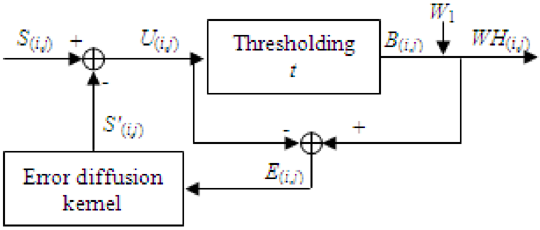


Fig. 5. Watermark (W_1) embedding in error diffusion halftoning

The secret image S is used as the input image of error diffusion halftoning. As shown in Fig. 5, when halftoning a continuous-tone image line by line sequentially, the past error is diffused to the current pixel. $S_{(i,j)}$ represents the current processing pixel and $S'_{(i,j)}$ is the diffused error sum added up from the neighboring processed pixels. $B_{(i,j)}$ is the binary output at position (i, j). $U_{(i,j)}$ is the updated output and $E_{(i,j)}$ is the difference between the $U_{(i,j)}$ and the $B_{(i,j)}$. $WH_{(i,j)}$ is the final watermarked output at position (i, j). The relationships of these variables are described from Eq. (1) to Eq. (5).

$$U_{(i,j)} = S_{(i,j)} - S'_{(i,j)} \tag{1}$$

$$S'_{(i,j)} = \sum_{m,n \in R} E_{(i-m,j-n)} \times K_{(m,n)} \tag{2}$$

$$E_{(i,j)} = WH_{(i,j)} - U_{(i,j)} \tag{3}$$

$$B_{(i,j)} = \begin{cases} 0 & \text{if} \quad U_{(i,j)} < t \\ 1 & \text{if} \quad U_{(i,j)} \geqslant t \end{cases} \tag{4}$$

$$WH_{(i,j)} = \begin{cases} 1 - B_{(i,j)} & \text{if} \quad B_{(i,j)} \neq W_1 \\ B_{(i,j)} & \text{if} \quad B_{(i,j)} = W_1 \end{cases} \tag{5}$$

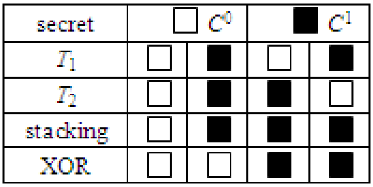
The parameter t in Eq. (4) is a threshold usually set as 0.5 (0 and 1 denote a black and a white pixel respectively). From Fig. 1, it is easily to find that different error diffusion kernels result in different visual quality of halftone images. There are three wide used error diffused kernels, i.e., Floyed-Steinberg kernel [12], Jarvis kernel [13] and Stucki kernel [14]. In this research Floyed-Steinberg kernel is adopted.

The watermark W_1 embedding is shown in Eq. (5). If the current pixel position belongs to those positions selected in advance, the output pixel value is determined by the watermark bit, instead of the thresholding value. Thus, the watermarked halftone image WH is obtained.

3.2 Watermark Embedding in Encryption

In this subsection, the second watermark W_2 is embedded in the two transparencies during encrypting the watermarked halftone image WH. One of the key factors of visual cryptography techniques is the pixel expansion, which is corresponding to the number of subpixels contained in each transparency. Ito et al. [11] proposed a non-expansion VSS model, which is based on a probabilistic principle. The size of the transparencies is equal to that of the secret image for there is no image size expansion. In Ito et al.'s model, only two single pixels called codewords are involved, as shown in Fig. 7. Similar to the conventional $\{2, 2\}$ VSS, there are also two choices for a white pixel and a black pixel encryption.

The encryption and decryption strategy of this non-expansion VSS is shown in Fig. 6. In particular, when encrypting a white (black) pixel, we randomly choose a column from C_0 (C_1), and assign them to the two transparencies respectively. In this paper this model is utilized for WH encryption.



secret	□ C^0		■ C^1	
T_1	□	■	□	■
T_2	□	■	■	□
stacking	□	■	■	■
XOR	□	□	■	■

Fig. 6. Encryption and decryption strategy of Ito et al's non-expansion VSS

Fig. 7. Codewords of Ito et al's non-expansion VSS

The second watermark W_2 embedding and secret sharing principle is illustrated in Fig. 8, in which four pixels (two pixels $P^1_{(i,j)}$ and $P^1_{(i,j+1)}$ belongs to T_1 and two corresponding $P^2_{(i,j)}$ and $P^2_{(i,j+1)}$ belongs to T_2) are considered at a time. We defined the group of these four pixels as an embedding unit. Each embedding unit can be used for 1 bit watermark embedding and 2 bit secret image data encryption. To encrypt a WH with the size of $K_1 \times K_2$ pixels, a watermark W_2 with up to $(K_1 \times K_2)$ bits can be hidden in two transparencies. There is also an embedding and encryption order of this processing.

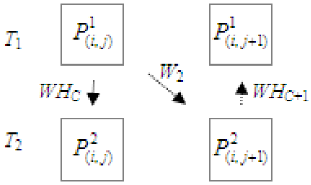


Fig. 8. Watermark (W_2) embedding in secret sharing

In particular, W_2 is hidden in $P^1_{(i,j)}$ and $P^2_{(i,j+1)}$ first, and then one bit WH_c is encrypted in $P^1_{(i,j)}$ and $P^1_{(i,j+1)}$, at last another $WH_c + 1$ bit is encrypted in $P^2_{(i,j+1)}$ and $P^1_{(i,j+1)}$. Details are described as follows.

Randomly select a codeword from the codebook (Fig. 7) and assign it to $P^1_{(i,j)}$, and then $P^2_{(i,j+1)}$ is determined by $P^1_{(i,j)}$ and the current watermark bit W_2 according to Eq. (6).

$$P^2_{(i,j+1)} = \begin{cases} P^1_{(i,j)} & \text{if} \quad W_2 = 0 \\ 1 - P^1_{(i,j)} & \text{if} \quad W_1 = 1 \end{cases} \tag{6}$$

Thus $P^1_{(i,j)}$ and $P^2_{(i,j+1)}$ are uniquely fixed.

In addition, $P^2_{(i,j)}$ is determined according to $P^1_{(i,j)}$ and the current WH bit. Similarly, $P^1_{(i,j+1)}$ is determined according to $P^2_{(i,j+1)}$ and next WH bit, as shown in Eq. (7) and Eq. (8).

$$P^2_{(i,j)} = \begin{cases} 1 - P^1_{(i,j)} & \text{if} \quad WH_c = 0 \\ P^1_{(i,j)} & \text{if} \quad WH_c = 1 \end{cases} \tag{7}$$

$$P^1_{(i,j+1)} = \begin{cases} 1 - P^2_{(i,j+1)} & \text{if} \quad WH_{c+1} = 0 \\ 1 - P^2_{(i,j+1)} & \text{if} \quad WH_{c+1} = 1 \end{cases} \tag{8}$$

Repeat above procedures for all embedding units, and in this way two watermarked transparencies are produced.

3.3 Secret Image Decryption and Watermark Extraction

Although two watermarks are embedded in the two transparencies, the secret content is revealed by stacking the two transparencies.

The hidden data W_2 can be retrieved using simple XOR operations, as Eq. (9).

$$W_2 = \mathrm{XOR}(P^1_{(i,j)}, P^2_{(i,j+1)}) \tag{9}$$

The watermark W_2 and W_1 extraction operations are two independent processes. As shown in Fig. 6, we find that the encrypted values can be recovered by XOR operation, which is quite essential for WH reconstruction. The watermarked halftone image WH must be losslessly reconstructed in advance for further $W1$ extraction. To each embedding unit, two watermark bits WH_c and WH_{c+1} are extracted as Eq. (10) and Eq. (11).

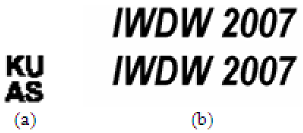
$$WH_c = \mathrm{XOR}(P^1_{(i,j)}, P^2_{(i,j)}) \tag{10}$$

$$WH_{c+1} = \mathrm{XOR}(P^1_{(i,j+1)}, P^2_{(i,j+1)}) \tag{11}$$

Repeating the above operations for all embedding units, and thus WH is precisely reconstructed. After that, we only need to find the selected pixels positions using the pseudo random number generator with the same key. Retrieve the pixel values in these positions and rearrange it into a binary sequence. Thus the watermark W_1 is extracted.

4 Experimental Results and Discussions

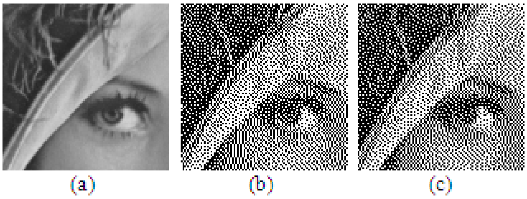
The 512 × 512 gray level Lena image is selected as the secret image, as shown in Fig. 11(a). Different from the watermark W_1, the watermark W_2 can be of half size of the secret image. In other words, to the 512 × 512 Lena image, the watermark W_2 can be a $(512 \times 512)/2$ bits message. In the experiment, two binary images 32×32 "KUAS" and 256×512 "IWDW2007" are selected as test watermarks W_1 and W_2 respectively, as shown in Fig. 9(a) and Fig. 9(b).



(a) (b)

Fig. 9. Watermark images, (a) W_1 "KUAS", (b) W_2 "IWDW2007"

Fig. 10 shows the visual distortion of the watermarked halftone image introduced by watermark W_1 embedding. Fig. 10(a) is the center part of the gray level Lena image. Fig. 10(c) and Fig. 10(b) are the corresponding parts of the halftone Lena images with and without "KUAS" embedded, respectively. From them, we find that when a small amount of data is embedded, the introduced visual quality distortion is acceptable. Actually, in order to preserve a better visual quality of WH, the first watermark message should not be very large.



(a) (b) (c)

Fig. 10. Visual distortion introduced by watermark W_1 embedding, (a) the gray level Lena, (b) the halftone Lena without watermark embedded, (c) the halftone Lena with watermark "KUAS" embedded

Fig. 11 demonstrates the experimental results of embedding "KUAS" and "IWDW2007" in visual secret sharing the gray level Lena image. There is no meaningful information can be distinguished from the two transparencies

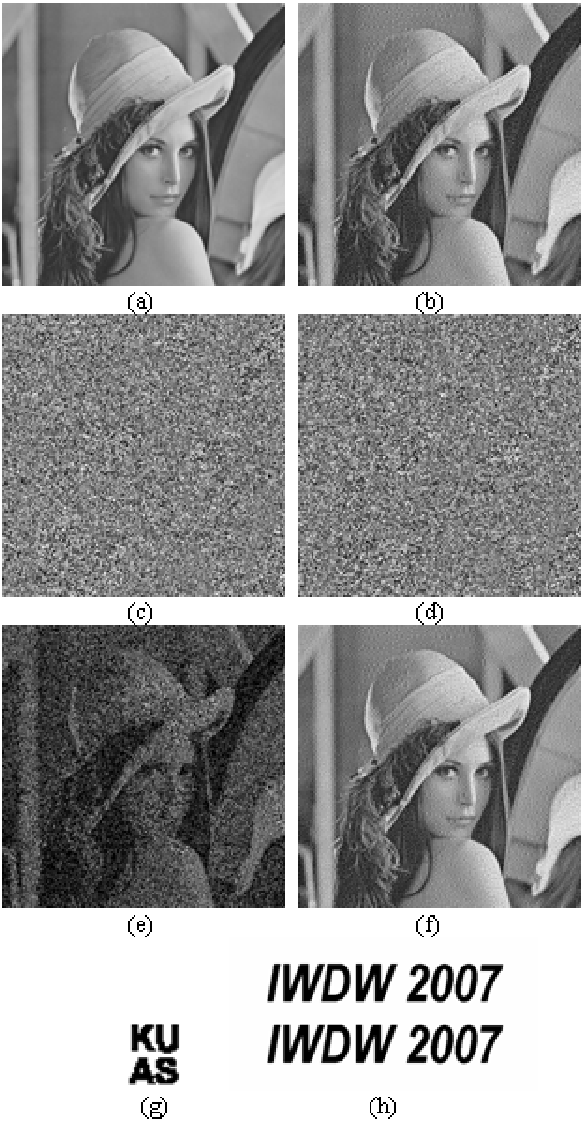


Fig. 11. Experimental results of the proposed scheme, (a) the secret image Lena, (b) the halftone Lena with watermark "KUAS" embedded, (c) the transparency image T_1, (d) the transparency image T_2, (e) the stacking results of T_1 and T_2, (f) the reconstructed halftone Lena with watermark "KUAS" embedded, (g) the extracted watermark "KUAS", (h) the extracted watermark "IWDW2007"

(Fig. 11(c) and Fig. 11(d)) for they looks like random-noise, and so as to guarantee the perfect security of the secret information.

In the experiment, on one hand, transparency holders can directly stack the two transparencies to reveal the secret content (Fig. 11(e)). On the other hand, a more distinct secret image (Fig. 11(f)), i.e., the watermarked halftone image WH, can be obtained by XOR computations of the two transparencies. Besides, lossless reconstruction of WH is the compulsory operation to extract the watermark "KUAS". Fig. 11(g) and Fig. 11(h) are the extracted hidden watermarks W_1 and W_2.

5 Conclusions

This paper proposes a scheme to hide two watermarks in transparencies of visual cryptography. With the use of error diffusion halftoning and a non-expansion VSS model, two sets of confidential messages can be embedded in two transparency images. Watermark extraction is simple for only XOR computations of two transparencies are needed. Experimental results demonstrate our scheme is effective and practical.

References

[1] Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)

[2] Zhou, Z., Arce, G.R., Crescenzo, G.D.: Halftone visual cryptography. IEEE Trans. on Image Processing 15(8), 2441–2453 (2006)

[3] Lin, C.C., Tsai, W.H.: Visual cryptography for gray-level images by dithering techniques. Pattern Recognition Letters 24, 349–358 (2003)

[4] Blundo, C., Santis, A.D., Naor, M.: Visual cryptography for grey level images. Information Processing Letters 75(6), 255–259 (2000)

[5] Jin, D., Yan, W.Q., Kankanhalli, M.S.: Progressive Color Visual Cryptography. Journal of Electronic Imaging 14(3) (2005)

[6] Hou, Y.C.: Visual cryptography for color image. Pattern Recognition 36(7), 1619–1629 (2003)

[7] Chang, C.-C., Yu, T.-X.: Sharing a secret gray image in multiple images. In: IEEE Proceedings of the First International Symposium on Cyber Worlds, pp. 230–237 (2002)

[8] Lukac, R., Plataniotis, K.N.: Colour Image Secret Sharing. IEE Electronics Letters 40(9), 529–530 (2004)

[9] Pan, J.S., Huang, H.C., Jain, L.C.: Intelligent Watermarking Techniques. World Scientific Publishing Company, Singapore (2004)

[10] Fang, W.P., Lin, J.C.: Visual Cryptography with Extra Ability of Hiding Confidential Data. Journal of Electronic Imaging 15(2) (2006)

[11] Ito, R., Kuwakado, H., Tanka, H.: Image size invariant visual cryptography. IEICE Trans. Fundamentals E82-A(10), 2172–2177 (1999)

[12] Floyd, R., Steinberg, L.: An adaptive algorithm for spatial gray scale. SID. Int. Symp. Dig. Tech. Papers, 36-37 (1975)

[13] Jarvis, J.F., Judice, C.N., Ninke, W.H.: A survey of techniques for the display of continuous-tone pictures on bilevel displays. Computer Graphics Image Process 5, 13–40 (1976)

[14] Stucki, P.: MECCA - A multiple error correcting computation algorithm for bilevel image hardcopy reproduction, Research Report RZ1060, IBM Res. Lab., Zurich, Switzerland (1981)

Scalable Security and Conditional Access Control for Multiple Regions of Interest in Scalable Video Coding

Yeongyun Kim, Sung Ho Jin, and Yong Man Ro

Image and Video Systems Laboratory, Information and Communications University,
119 Munjiro, Yuseong-gu, Deajeon, 305-732, Korea
yro@icu.ac.kr

Abstract. In this paper, we propose an encryption method to secure the multiple regions of interest (ROIs) of scalable video coding (SVC) bitstream from malicious attacks and a key generation scheme to control the protected ROIs conditionally. To establish the encryption method, we analyze the encryption requirements to maintain spatial, temporal, and SNR scalability of SVC. Depending on the given user's condition such as the status of the payment, the proposed conditional access control provides keys to access various types of the secured scalable videos ranging from the lowest to the highest quality. Since the conditional access control needs multiple ROIs as secured videos, the key generation scheme should support the multiple ROIs that are secured with the proposed encryption method. Experiments are performed to verify the proposed methods and results show that the proposed schemes could provide an effective access control of the secured multiple ROIs in SVC.

1 Introduction

Scalable video coding (SVC) is an ongoing standard that can support spatial, temporal, and quality scalability. Also, it is considered as one of the new video coding methods that can replace the conventional transcoding methods without re-encoding original video to adapt different network conditions or user devices [1].

Recently, flexible macroblock ordering (FMO) in SVC supports an independent decoding of the region of interest (ROI) offering ROIs from a picture of video sequences in order to meet user's preferences and desires. No encryption method of the ROI in SVC has been proposed so far. The demand of a new encryption method has been increased to accomplish the secured ROIs.

As developing scalable coding methods of images and videos [2], [3], access control emerges to provide the secured scalable contents with appropriate users who have a variety of rights or licenses [4], [5]. Access control plays an important role to obtain and manage scalable media. It can be used for providing different portions of secured scalable contents. Low-quality content can be available for free, which leads consumers to pay for higher quality contents. In our previous work, we proposed the conditional access control scheme in SVC full picture [6].

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 71–86, 2008.

On the other hand, an independent ROI, which supposes to be usedin cellular phones, is considered as a scalable video content. This means that a conditional access control is needed within ROI. To support a scalable security in multiple ROIs, current scheme is not sufficient enough to fully achieve access control.

In this paper, we propose an encryption method and key generation scheme for a conditional access control scheme of multiple ROIs which are currently the standard element in SVC. The experimental results showed that the proposed method can achieve the conditional access control for each ROI effectively.

2 Overview of Multiple ROIs in SVC

2.1 SVC Bitstream Structure

SVC supports spatial, temporal and SNR (or quality) scalabilities for users who have variable network environment and devices. In SVC, spatial scalability is emerged by layered coding, and temporal scalability is established by hierarchical B picture structure [1]. For SNR scalability, the fine-granular scalability (FGS) and the coarsegranular scalability (CGS) are provided. Encoded SVC bitstream consists of a base layer and multiple enhancement layers. The base quality of original video can be obtained from the base layer. In addition, the enhancement layers are added on the base layer to obtain an enhanced quality [1]. Figure 1 shows a hierarchical structure of the scalable layers in terms of spatial, temporal, and SNR scalability in SVC.

In order to effectively consume the SVC bitstream in diverse networks, encoded bitstream is composed of network abstraction layer (NAL) units. NAL units are classified into the video coding layer (VCL) NAL unit, which is a transport entity containing at least a coded slice data, and non-VCL NAL unit, that includes additional information such as parameter sets and supplemental enhancement information (SEI) [1]. A NAL unit consists of two parts: the NAL header and the raw byte sequence payload (RBSP). The NAL header includes information of RBSP.

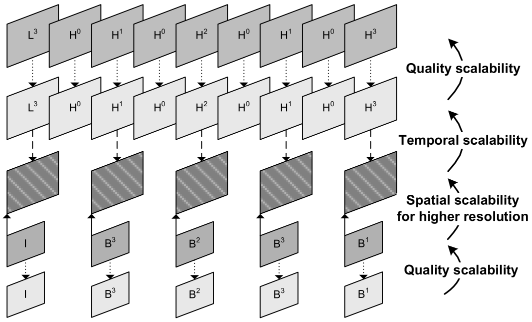

L^3 H^0 H^1 H^0 H^2 H^0 H^1 H^0 H^3
Quality scalability
L^3 H^0 H^1 H^0 H^2 H^0 H^1 H^0 H^3
Temporal scalability
Spatial scalability
for higher resolution
I B^3 B^2 B^3 B^1
Quality scalability
I B^3 B^2 B^3 B^1

Fig. 1. Hierarchical structure of the scalable layers in terms of the spatial, temporal, and quality scalability

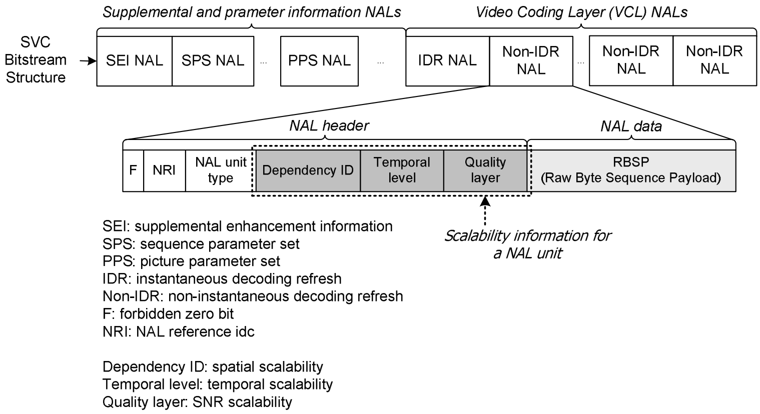

Supplemental and prameter information NALs
Video Coding Layer (VCL) NALs
SVC Bitstream Structure
SEI NAL
SPS NAL
...
PPS NAL
...
IDR NAL
Non-IDR NAL
...
Non-IDR NAL
Non-IDR NAL
NAL header
NAL data
F
NRI
NAL unit type
Dependency ID
Temporal level
Quality layer
RBSP (Raw Byte Sequence Payload)
Scalability information for a NAL unit
SEI: supplemental enhancement information
SPS: sequence parameter set
PPS: picture parameter set
IDR: instantaneous decoding refresh
Non-IDR: non-instantaneous decoding refresh
F: forbidden zero bit
NRI: NAL reference idc
Dependency ID: spatial scalability
Temporal level: temporal scalability
Quality layer: SNR scalability

Fig. 2. NAL unit syntax in SVC bitstream

The VCL NAL unit is the smallest unit that can be defined with the spatial, temporal, and quality scalability. Figure 2 shows the coded SVC bitstream structure. In the NAL header, scalability information such as dependency ID, temporal level, and quality layer is included. And VCL NAL data contains encoded slice data as RBSP.

To transmit the usable coded data to the users through the diverse network, SVC bitstream needs to be extracted. The bitstream extraction along with the spatial and temporal scalability is achieved by dropping the NAL unit. On the other hand, the extraction in terms of the quality scalability is achieved by cropping and dropping the NAL unit [1]. Especially, cropping the NAL unit can provide the detail adjustment of the bitstream at the transmittable bitrate in the diverse network.

2.2 Multiple ROIs Representation in SVC

The FMO forms a slice group with a set of macroblocks by using flexible manners [7]. The ROIs are defined by map type 2 of macroblock-to-slice-group-maps, known as “foreground and leftover,” which are supported by the FMO. As shown in Figure 3, the “foreground and leftover” map groups the macroblocks located in rectangular regions into slice groups 0 and 1, and the macroblocks belonging to the background are specified into slice group 2.

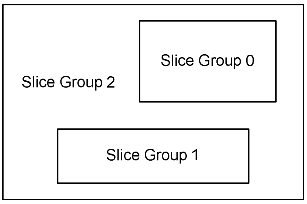


Fig. 3. “Foreground and leftover” type of FMO

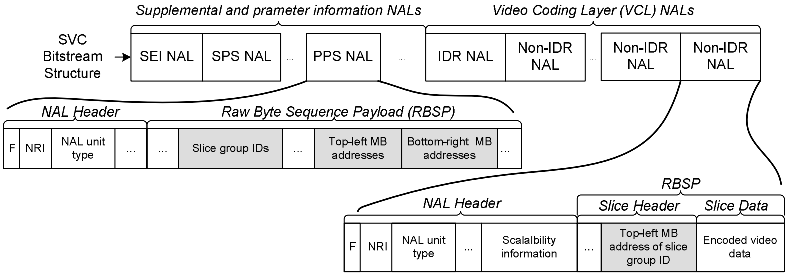


Fig. 4. SVC bitstream structure for the description of multiple ROIs

The PPS NAL in Figure 4 contains the geometric information of the ROI, e.g., the top-left and the bottom-right macroblock address of each slice group, as well as the slice group ID for each slice group [7]. Because each NAL unit contains one slice group and the slice header has the first macroblock address of the slice group, the slice group ID of the ROI identifies NAL units that belong to the ROI.

3 Scalable Security and Consumption of Multiple ROIs in SVC

3.1 System Layout for Scalable Secured Multiple ROIs in SVC

Figure 5 shows the system layout of the applications adopting the secured multiple ROIs in SVC. Described ROIs can be independently consumed in appropriate user devices such as Digital TV, PC, PDA, cellular phone, etc.

If a user demands a high quality video in terms of spatial, temporal, and quality scalability, SVC can support a suitable scalability of ROIs to the user. The required ROIs are encrypted so that only users who hold licenses to consume the protected ROIs can enjoy the available content. However, malicious attackers cannot access the ROIs because they have no keys. The proposed conditional access control scheme can be effectively adapted for the secured ROI. Moreover, the content provider can get appropriate values by our conditional access control scheme. In addition, the background area except for the ROIs has also slice group ID as the lowest number. Therefore, the background area can be encrypted as an ROI. Thus, the entire picture is encrypted and protected by using the proposed methods.

3.2 A Selective Encryption Method

The encryption should be performed in accordance with the scalabilities of SVC, which satisfies three requirements in [6]. The requirements are described below. Firstly, since SVC has base and enhancement layers, all layers should be encrypted for the robust SVC video security. In other words, all types of encoded video data such as texture, motion vector difference, and FGS in SVC should

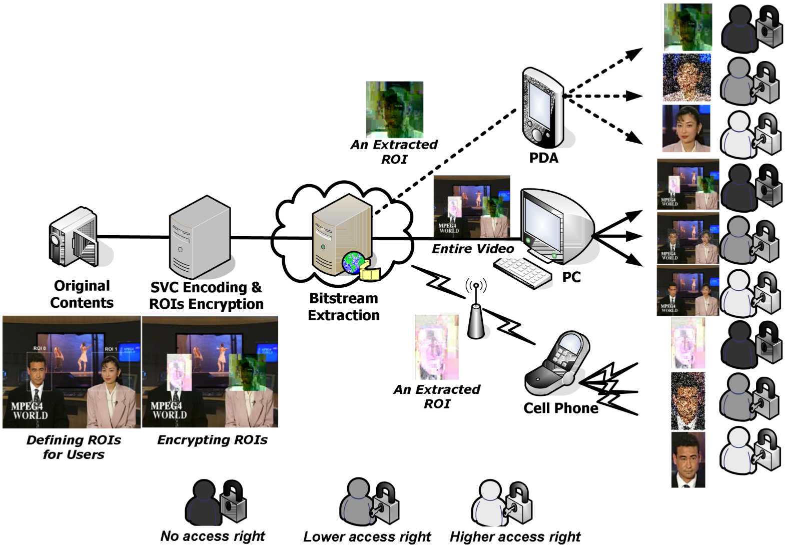


Fig. 5. Applications by using the secured multiple ROIs in SVC

be encrypted. In addition, encryption of texture, motion, and FGS data as no syntax element will achieve the format compliance for SVC file format. Secondly, we should consider the bitstream extraction that is mentioned in Subsection 2.1. In order to avoid repeatedly decrypting and encrypting the bitstream in the extraction stage, the encryption should be applied segment by segment based on the NAL unit. In addition, conditional access control is proposed to use different keys assigned to the NAL units which have different scalability information in [6]. Thus, we need to consider the conditional access in the ROI encryption in SVC. Thirdly, the encryption method should be lightweighted in terms of computational complexity. Since encoding and decoding of SVC have heavy computational complexity, a light-weighted encryption method could reduce an additional complexity and work as a real-time application. The proposed encryption method has to meet these requirements.

Figure 6 shows a block diagram that represents the proposed encryption method. Our encryption method is now as followed.

Input:

$U_{VCL\ NAL}$: VCL NAL unit

NAL_unit_key: A key available to encrypt and decrypt the VCL NAL unit of the required ROI in terms of spatial, temporal, and SNR scalability

slice_group_id: slice group id which can identify the required ROI

NAL_unit_type: NAL unit type identifier

dependency_id: spatial scalability identifier

temporal_level: temporal scalability identifier

quality_layer: SNR scalability identifier

```
Output:
    $E_{sign}$ : encrypted sign bitstream of texture, motion, FGS in the VCL NAL unit

Operation:
    Order $A$:  perform the order on $A$
    Order $A \Leftarrow B$:  perform the order on $A$ by using $B$

Method ENCRYPTION
  1. select VCL NAL unit of the required ROI, $U_{VCL\ NAL} \Leftarrow$ {slice_group_id, NAL_unit_type}
  2. Generate seed $\Leftarrow$ {NAL_unit_type, dependency_id, temporal_level, quality_layer}
  3. Generate random number stream $\Leftarrow$ seed
  4. select sign of texture, motion vector difference, FGS $\in U_{VCL\ NAL}$
  5. for $i = 1$ to the last data of $U_{VCL\ NAL}$ do
  6.     $E_{sign} \leftarrow$ random number stream xor sign
  7. Encrypt seed $\Leftarrow$ NAL_unit_key
  8. end
```

After selecting the VCL NAL unit of the required ROI to be encrypted in line 1 of the method, a seed initializing the pseudo-random number generator in Figure 6 is generated by using the NAL unit type, dependency id, temporal level, and quality layer. The different seed depends on the NAL unit type and the scalability information. That means the encryption is based on NAL units which have the same scalability information. Conditional access control can be achieved to encrypt VCL NAL units which have the same scalability information

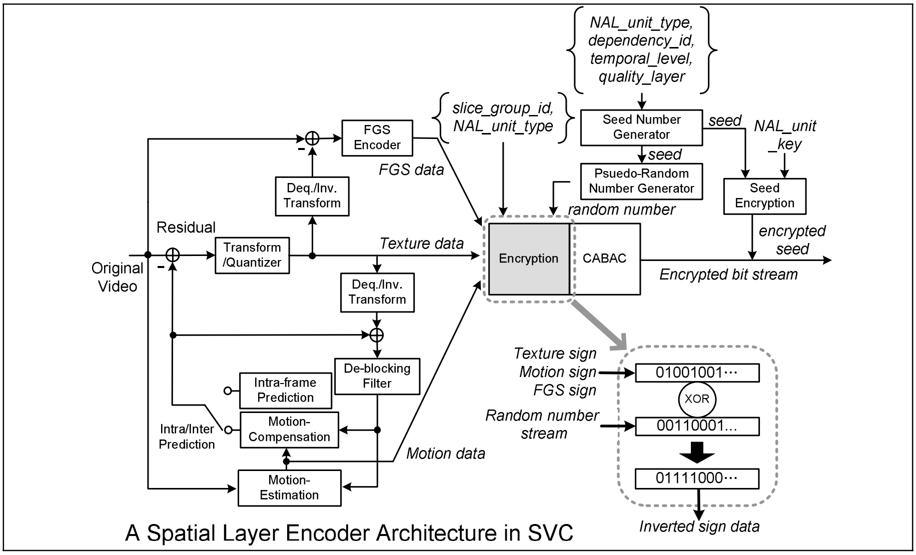


Fig. 6. Proposed encryption method for a spatial layer in SVC

belonging to the ROI by using the same seed. In addition, encryption applied to the NAL unit meets the second requirement as aforementioned above.

The seed is inserted into encrypted SVC bitstream with a NAL unit key using the conventional data encryption schemes in line 7 of the method. Here, a precondition regarding a particular cipher is out of this work. NAL unit keys are used for the security of generated seeds. These keys enable to access the NAL units. Thus, the encrypted seed needs to be transmitted with the SVC bitstream, since the decryption process requires the seed producing the same random stream via the same random number generator in the SVC decoder.

Finally, XOR-operation is performed to the generated random stream with sign of coded data in line 6 of the method. Encrypting sign data from texture, motion vector difference, and FGS do not affect the original coding efficiency resulted from the encoding process which has no encryption process because the sign data has no effect in determining the probability of a variable length code in the context-based adaptive binary arithmetic coding (CABAC) [8]. As a result, inverting the sign of each data using XOR-operation meets the third requirements as a light-weighted encryption. And the first requirement is met, because the signs of all data in the scalable layers are encrypted in our encryption method.

3.3 Consumption of the Scalable Secured Multiple ROIs

The scalable secured ROIs should be consumed by an authorized user who has the suitable accessibility right. The proposed conditional access control scheme enables the authorized user to access a higher layer as well as a lower layer of the required ROI.

For the proposed conditional access control scheme of the secured multiple ROIs in SVC bitstream, the scalabilities of the NAL unit should be deliberated as the fundamental unit. Spatial, temporal, and SNR scalabilities belonging to the ROI are enhanced by a form of a layer or level in SVC. Each layer or level becomes 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 [6]. The key encrypting and decrypting the NAL unit is denoted as NAL unit key in this paper. In Figure 2, we obtain scalability information such as *dependent ID, temporal level,* and *quality layer* from the NAL header.

Figure 7 shows an example of the access control scheme in SVC bitstream. It shows 2 ROIs (ROI 0, ROI 1), 2 spatial layers (Spatial 0, Spatial 1), 2 temporal levels (Temporal 0, Temporal 1), and 2 quality layers (SNR 0, SNR 1). If a user wants to access r ROIs, 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 7, *NAL unit key is* represented as *Key (ROI, spatial layer, temporal level, SNR layer).*

The number of keys needed to encrypt multiple ROIs in SVC bitstream can be written as,

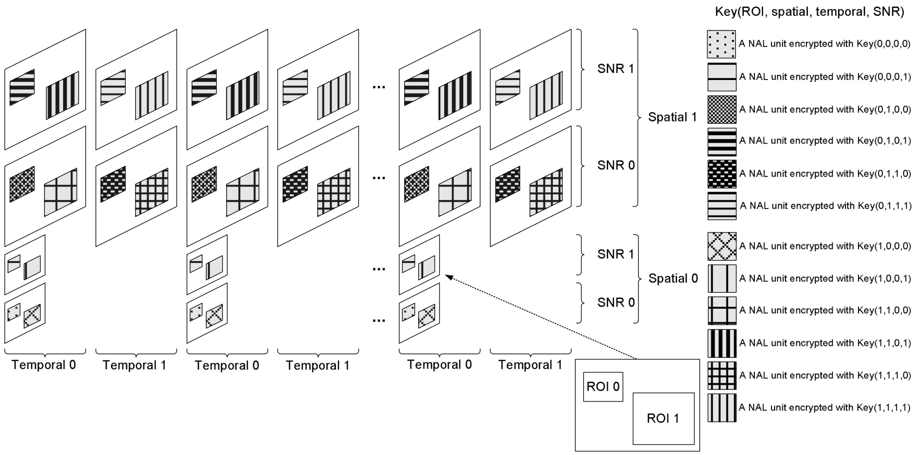

Key(ROI, spatial, temporal, SNR)
A NAL unit encrypted with Key(0,0,0,0)
A NAL unit encrypted with Key(0,0,0,1)
A NAL unit encrypted with Key(0,1,0,0)
A NAL unit encrypted with Key(0,1,0,1)
A NAL unit encrypted with Key(0,1,1,0)
A NAL unit encrypted with Key(0,1,1,1)
A NAL unit encrypted with Key(1,0,0,0)
A NAL unit encrypted with Key(1,0,0,1)
A NAL unit encrypted with Key(1,1,0,0)
A NAL unit encrypted with Key(1,1,0,1)
A NAL unit encrypted with Key(1,1,1,0)
A NAL unit encrypted with Key(1,1,1,1)
SNR 1
SNR 0
Spatial 1
SNR 1
SNR 0
Spatial 0
Temporal 0
Temporal 1
Temporal 0
Temporal 1
Temporal 0
Temporal 1
ROI 0
ROI 1

Fig. 7. NAL units belonging to each ROI encryption for conditional access control

$$Key_{Total} = \sum_{r=1}^{NR} \sum_{s=1}^{NS} (NQ_s \times NT_s), \tag{1}$$

where *NR* is the number of ROIs, *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.

From the proposed conditional access control scheme depicted in Figure 7, the key set to access the SVC bitstream with r ROIs, s spatial layer, t temporal level, and q SNR layer can be written as,

$$KEYSET_{r,s,t,q} = \left\{ Key(k,l,m,n) \left| \begin{array}{l} 0 \le k \le r, \\ 0 \le l \le s, \\ 0 \le m \le min(t, NT_s), \\ if \;\; l = s \;\; then \quad 0 \le n \le q, \;\; else \quad 0 \le n \le NQ_s \end{array} \right. \right\}, \tag{2}$$

A key set list for different access control depending on scalable layers in Figure 7 is shown in Table 1.

3.4 Key Generation for Consumption of the Scalable Secured Multiple ROIs

In general, key generation and management schemes are to provide decryption keys to users effectively. Especially, secured scalable media contents with multiple keys can cause the increase of complexity in both digital rights management server and terminal. Thus, key generation scheme on scalable coding such as MPEG-4 FGS to reduce the number of key used for encryption and decryption was proposed[9], [10]. In order to access a higher layer, all lower layers are necessary in scalable coding. In this case, a lot of keys should be provided. In [6],

Table 1. The *NAL unit key* set list needed to access the NAL unit which has certain scalabilities belonging to each ROI. It is calculated by Eq. (2) from the SVC example of Fig. 7.

Temporal layer ROI, Spatial and SNR layer			Temporal 0	Temporal 1
ROI 0	Spatial 0	SNR 0	*{Key(0,0,0,0)}*	*Not exist*
		SNR 1	*{Key(0,0,0,0),Key(0,0,0,1)}*	*Not exist*
	Spatial 1	SNR 0	*{Key(0,0,0,0),Key(0,0,0,1) Key (0,1,0,0) }*	*{Key(0,0,0,0),Key(0,0,0,1) Key(0,1,0,0),Key(0,1,1,0)}*

		SNR 1	*{Key (0,0,0,0),Key(0,0,0,1)
			Key (0,1,0,0),Key(0,1,0,1)}*
				*{Key(0,0,0,0),Key(0,0,0,1)
				Key(0,1,0,0),Key(0,1,0,1)
				Key(0,1,1,0),Key(0,1,1,1)}*

ROI 1	Spatial 0	SNR 0	*{Key(1,0,0,0)}*	*Not exist*
		SNR 1	*{Key(1,0,0,0),Key(1,0,0,1)}*	*Not exist*
	Spatial 1	SNR 0	*{Key(1,0,0,0),Key(1,0,0,1) Key (1,1,0,0) }*	*{Key(1,0,0,0),Key(1,0,0,1) Key(1,1,0,0),Key(1,1,1,0)}*

		SNR 1	*{Key (1,0,0,0),Key(1,0,0,1)
			Key (1,1,0,0),Key(1,1,0,1)}*
				*{Key(1,0,0,0),Key(1,0,0,1)
				Key(1,1,0,0),Key(1,1,0,1)
				Key(1,1,1,0),Key(1,1,1,1)}*

we proposed the key generation scheme to reduce to multiple keys for a picture in SVC. However, [6] is not enough to achieve conditional access control for the secured multiple ROIs. In this paper we propose key generation scheme to effectively provide authorized users with keys.

In [6], we defined *master key, type key, layer key,* and *access key*. The *master key* is the key assigned to one secured SVC 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 assigned to 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 decrypt all the NAL units to access extracted bitstream with a given access rights.

In case of having no ROIs, a coded bitstream have spatial, temporal, and SNR scalability and corresponding *type keys* need to be generated from a *master key*. On the other hand, if there are multiple ROIs in SVC, another type key should be generated for identifying each ROI. In [6], spatial scalability and SNR scalability are categorized into the same type, since the spatial and SNR layers are dependent. Here, another type from the ROIs is added. Figure 8 shows three types for scalable layers in Figure 7. Each ROI is included in type 1, spatial and SNR layers are included in type 2, and temporal layers are included in type 3.

The *master key*, K is assigned to a SVC contents which have multiple ROIs and three *type keys* (K_j) are generated as followed,

$$K_j = H(K \,||\, j), \tag{3}$$

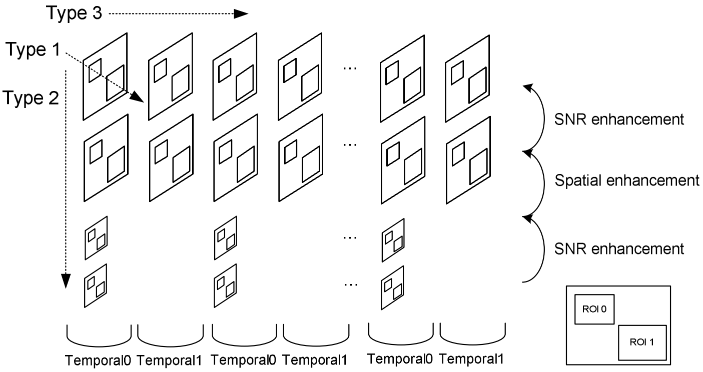


Fig. 8. Redefined the ROI type and 2 scalability types to generate the type key

where $H(.)$ is a cryptographic hash function, K is the *master key*, j represents the ROI type (j=*1*), spatio-SNR scalability type (j=2) and temporal scalability type (j=3). The operator, "||" denotes concatenation operation.

For the type of ROIs, the layer keys cannot be adapted for our key generation scheme. The relationship of the ROIs is independent, because multiple ROIs should be independently processed in the application. Thus, we define the *ROI key*. The *ROI key* K_{ROIr} for r-th ROI can be obtained as,

$$K_{ROIr} = H(K_j \mid\mid r), \quad for\ j = 1 \tag{4}$$

For the *layer key* of spatio-SNR layer and temporal level, the highest *layer key* is generated by hashing the *type key*. In Figure 8, in order to access a high layer belonging to the required ROI, lower layers of the ROI are needed. Thus, a *layer key* is generated by hashing a higher *layer key*. The *layer key* $K_{i,j}$ for i-th layer in j-th scalability type can be obtained as,

$$\begin{aligned} K_{i,j} &= \begin{pmatrix} H(K_{i+1,j}), \text{for } 1 \le i < n_j \\ H(K_j), \quad \text{for } i = n_j \end{pmatrix}, \quad for\ j = 2{,}3 \\ &= H^{n_j+1-i}(K_j), \text{for } 1 \le i \le n_j, \quad for\ j = 2{,}3 \end{aligned} \tag{5}$$

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 *ROI key* in Eq. (4) and the *layer key* in Eq. (5), the *NAL unit key* of Key(r,s,t,q) belonging to r-th ROI that is used to encrypt and decrypt the NAL unit for (s, t, q) scalabilities of r-th ROI, can be written as,

$$K(r,s,t,q) = K_{ROI_r} \mid\mid K_{a,2} \mid\mid K_{b,3}, \tag{6}$$

where K_{ROIr} is the ROI key for r-th ROI, $K_{a,2}$ is the *layer key* for layer a in type 2, $K_{b,3}$ is the *layer key* for layer b in the type 3, $a=\sum_{x=1}^{s-1} NQ_x+q$, and $b = t$.

Table 2. *Access keys* by the proposed algorithm from the SVC example of Fig. 7

Temporal layer ROI, Spatial and SNR layer			15 fps	30 fps
ROI 0	QCIF	Base	$K_{ROI_0} \,||\, K_{0,2} \,||\, K_{0,3}$	*Not exist*
		FGS	$K_{ROI_0} \,||\, K_{1,2} \,||\, K_{0,3}$	*Not exist*
	CIF	Base	$K_{ROI_0} \,||\, K_{2,2} \,||\, K_{0,3}$	$K_{ROI_0} \,||\, K_{2,2} \,||\, K_{1,3}$
		FGS	$K_{ROI_0} \,||\, K_{3,2} \,||\, K_{0,3}$	$K_{ROI_0} \,||\, K_{3,2} \,||\, K_{1,3}$
ROI 1	QCIF	Base	$K_{ROI_1} \,||\, K_{0,2} \,||\, K_{0,3}$	*Not exist*
		FGS	$K_{ROI_1} \,||\, K_{1,2} \,||\, K_{0,3}$	*Not exist*
	CIF	Base	$K_{ROI_1} \,||\, K_{2,2} \,||\, K_{0,3}$	$K_{ROI_1} \,||\, K_{2,2} \,||\, K_{1,3}$
		FGS	$K_{ROI_1} \,||\, K_{3,2} \,||\, K_{0,3}$	$K_{ROI_1} \,||\, K_{3,2} \,||\, K_{1,3}$

In this paper, *access key* is the *NAL unit key* for given a number of ROI and scalabilities to be accessed, and all the *NAL unit keys* which are needed to access to the given a number of ROI and scalabilities are generated by *access key*. Table 2 shows the *access keys* generated by the proposed key generation scheme. In Table 2, the *access keys* for the example of Figure 8 are generated.

In Table 2, for example, the ROI 1 has CIF, FGS layer quality, and 15 fps can be accessed with access key, $K_{ROI1}||K_{3,2}||K_{0,3}$. By Eq. (6), the *access key* can generate the *NAL unit key set* $\{K_{ROI1}||K_{0,2}||K_{0,3}$, $K_{ROI1}||K_{1,2}||K_{0,3}$, $K_{ROI1}||K_{2,2}||$ $K_{0,3}$, $K_{ROI1}||K_{3,2}||K_{0,3}\}$ which is needed to decrypt the bitstream.

By using our key generation scheme to achieve conditional access control for the scalable secured multiple ROIs, a single *master key* can *access keys* generated from the *master key*, and then all *NAL unit keys* to encrypt and decrypt the NAL units belonging to each ROI can be generated effectively. As compared both Table 1 and Table 2, the number of transmittable keys are significantly reduced in Table 2 rather than Table 1. And the previous work in [6] did not consider any ROI in SVC. In case of using key generation scheme in [6], *access keys* or *NAL unit keys* cannot be identified whether those keys are belonged to a certain ROI. In addition, security level of the key generation is increased by defining the *ROI keys* generated from the hash function.

4 Experimental Results

We have implemented the proposed methods in the JSVM 6.0 [1]. The "foreman" and "news" sequence as the test sequences of MPEG SVC are used in our experiment. The test sequences are encoded by 2 spatial layers {CIF, QCIF}, 2 temporal levels {15fps, 30fps} and 2 SNR layers {base quality, FGS quality}. The "foreman" sequence has an ROI. The size of the ROI is 80 × 64 (pixel by pixel) in QCIF resolution. And the "news" sequence has 2 ROIs. The sizes of the ROI 0, ROI 1 are

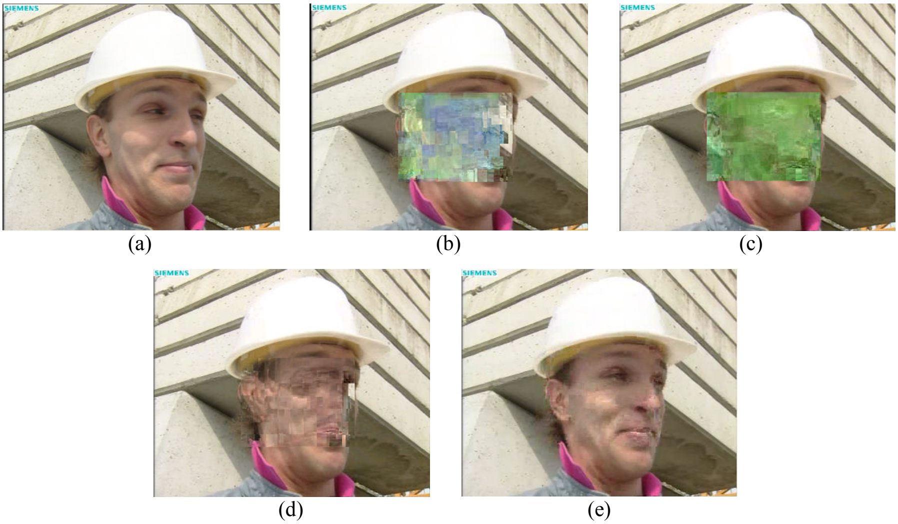


Fig. 9. Visual patterns comparison of encrypted video data in terms of texture, motion vector difference, and FGS: (a) original "foreman" video, (b) visual pattern of video in which texture, motion vector difference, and FGS are encrypted, (c) visual pattern of video in which texture is encrypted, (d) visual pattern of video in which motion vector difference is encrypted, and (e) visual pattern of video in which FGS is encrypted.

32 × 48, 48 × 48 in QCIF resolution respectively. The value of quantization parameter (QP) is 36 for all experiments. In this paper, the experiment deals mainly with the conditional access control with the scalable multiple ROIs secured by the proposed encryption/decryption method and key generation scheme.

Figure 9 shows visual patterns of the decoded "foreman" sequence which has no decryption keys. A SVC content belonging to the ROI consists of multi-layer structure and has different data types in terms of texture, motion vector difference and FGS in SVC. In order to obtain a secured ROI content, encryption should be performed for all data types belonging to the ROI.

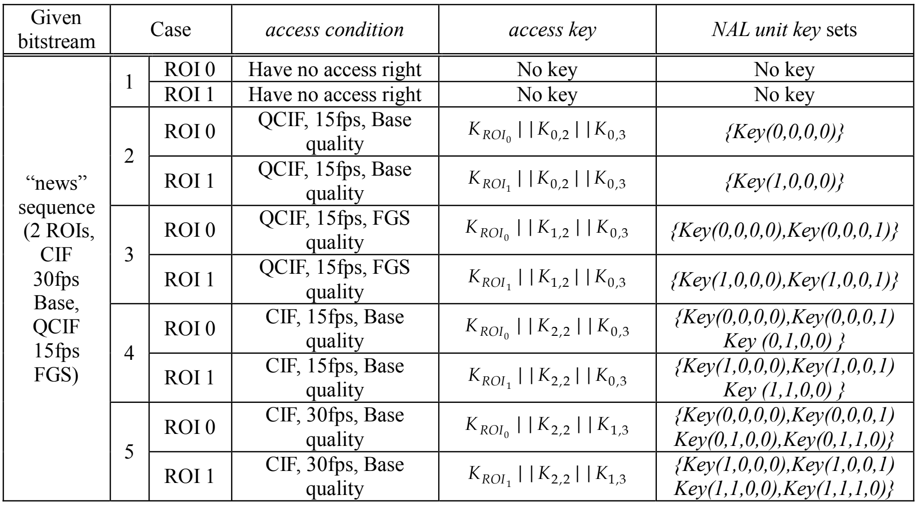
Table 3 shows PSNR values of the decoded results in Figure 9.

For the next experiment, we performed with the 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 accessi-

Table 3. Y, U, and V PSNR values of decoded pictures in Fig. 9

Encryption data	PSNR Y	PSNR U	PSNR V
None (original)	38.3728	43.9624	46.3036
Texture + Motion vector difference + FGS	20.2982	31.4153	28.0091
Texture	22.3170	33.6134	33.3702
Motion vector difference	29.3865	42.4271	42.6711
FGS	35.0670	40.8353	42.9685

Table 4. Access condition and corresponding key set to access



Given bitstream	Case		*access condition*	*access key*	*NAL unit key* sets
“news” sequence (2 ROIs, CIF 30fps Base, QCIF 15fps FGS)	1	ROI 0	Have no access right	No key	No key
		ROI 1	Have no access right	No key	No key
	2	ROI 0	QCIF, 15fps, Base quality	$K_{ROI_0} \| \| K_{0,2} \| \| K_{0,3}$	*{Key(0,0,0,0)}*
		ROI 1	QCIF, 15fps, Base quality	$K_{ROI_1} \| \| K_{0,2} \| \| K_{0,3}$	*{Key(1,0,0,0)}*
	3	ROI 0	QCIF, 15fps, FGS quality	$K_{ROI_0} \| \| K_{1,2} \| \| K_{0,3}$	*{Key(0,0,0,0),Key(0,0,0,1)}*
		ROI 1	QCIF, 15fps, FGS quality	$K_{ROI_1} \| \| K_{1,2} \| \| K_{0,3}$	*{Key(1,0,0,0),Key(1,0,0,1)}*
	4	ROI 0	CIF, 15fps, Base quality	$K_{ROI_0} \| \| K_{2,2} \| \| K_{0,3}$	*{Key(0,0,0,0),Key(0,0,0,1) Key (0,1,0,0) }*
		ROI 1	CIF, 15fps, Base quality	$K_{ROI_1} \| \| K_{2,2} \| \| K_{0,3}$	*{Key(1,0,0,0),Key(1,0,0,1) Key (1,1,0,0) }*
	5	ROI 0	CIF, 30fps, Base quality	$K_{ROI_0} \| \| K_{2,2} \| \| K_{1,3}$	*{Key(0,0,0,0),Key(0,0,0,1) Key(0,1,0,0),Key(0,1,1,0)}*
		ROI 1	CIF, 30fps, Base quality	$K_{ROI_1} \| \| K_{2,2} \| \| K_{1,3}$	*{Key(1,0,0,0),Key(1,0,0,1) Key(1,1,0,0),Key(1,1,1,0)}*

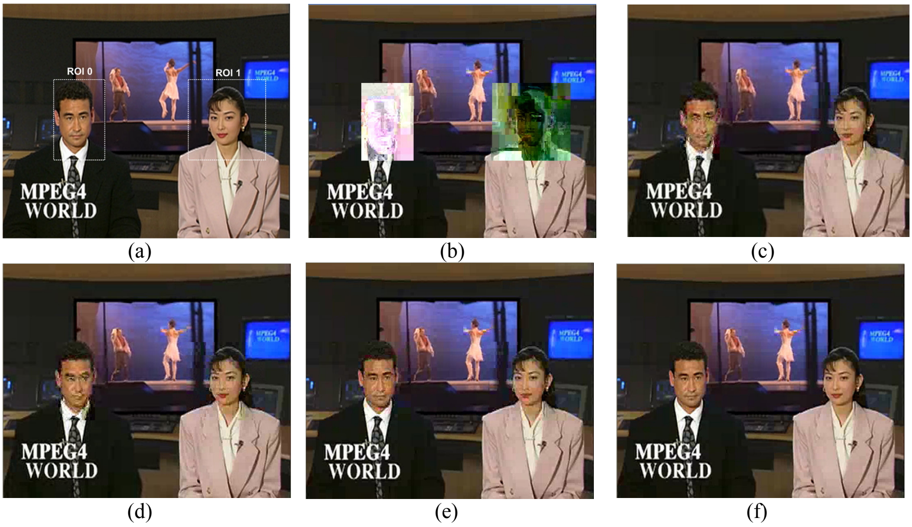


Fig. 10. Visual patterns of decoded video for the given SVC bitstream with the *access key* of the corresponding access right: (a) original “news” video, (b) case 1, (c) case 2, (d) case 3, (e) case 4, and (f) case 5

bility for its scalability. Table 4 shows five access rights and corresponding *access keys*.

Figure 10 shows the decoded video with different *access rights*. Case 5 shows that all layers are decrypted using an *access* key in accordance with the access

Table 5. PSNR results for the *entire access right quality* and the *corresponding access quality*

		visual quality for entire access right			*visual quality for corresponding access right*		
Case		PSNR Y	PSNR U	PSNR V	PSNR Y	PSNR U	PSNR V
1	ROI 0	33.9190	37.1669	37.5510	4.8930	20.6157	25.1187
	ROI 1	33.9190	37.1669	36.4454	4.3193	11.3005	20.3109
2	ROI 0	33.9190	37.1669	37.5510	17.8548	30.2833	28.4369
	ROI 1	32.4982	35.7145	36.4454	18.7248	30.9519	29.5293
3	ROI 0	33.9190	37.1669	37.5510	21.5029	32.2970	32.2730
	ROI 1	32.4982	35.7145	36.4454	21.8921	32.8761	32.0385
4	ROI 0	33.9190	37.1669	37.5510	28.5869	33.5918	33.2401
	ROI 1	32.4982	35.7145	36.4454	27.4841	32.0918	33.3516
5	ROI 0	33.9190	37.1669	37.5510	33.9190	37.1669	37.5510
	ROI 1	32.4982	35.7145	36.4454	32.4982	35.7145	36.4454

condition. The others, however, are partially decrypted or not decrypted at all. The visual effects show differently depending on the corresponding access rights.

Table 5 and Figure 11 show the visual quality for the entire access right and that for the corresponding access right of what a user has. The visual quality for entire access right is the maximum quality that can be achieved by the given bitstream. The visual quality for corresponding access right is the decoded quality of the video of each ROI that is decrypted using the corresponding *access key* shown in Table 4. In the experimental results, the visual quality for corresponding access right is always equal or lower than the visual quality for entire access right. Thus, to access a higher layer with a given access right will emerge the degradation of the video quality.

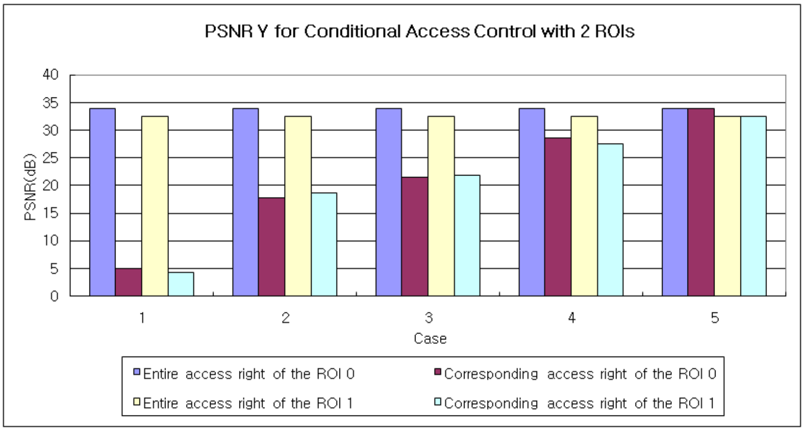


Fig. 11. Comparison between the visual qualities for entire access right and corresponding access right

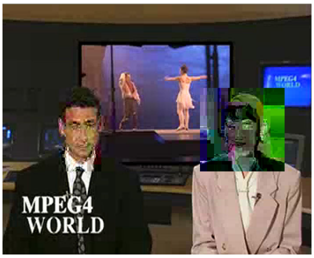


Fig. 12. Conditional Access control with different access rights of 2 ROIs; ROI 0 (QCIF, 15fps, base quality), ROI 1(No access right)

Table 6. Time efficiency of the encryption/decryption process

Video	Encryption(%)/ encoding	Decryption(%)/ decoding
Football	0.27	0.65
Foreman	0.27	0.37
Harbour	0.37	0.69
Akiyo	0.84	1.57
Bus	0.23	3.45

Figure 12 shows the independent conditional access control for multiple ROIs. A user has only one *access key* to decrypt the NAL unit belonging to the ROI 0 as "QCIF, 15fps, base quality." If a picture has multiple ROIs, a user has variable access right for each ROI and can access the ROI respectively.

Finally, Table 6 represents the time efficiency of the proposed encryption and decryption process for full size pictures. The efficiency experiment was performed by using various SVC test videos and the result shows that encryption and decryption time do not exceed more than 4% of the encoding and decoding time. The test was performed with Intel Pentium processor 2.80GHz and 1.50GB RAM.

5 Conclusion

In this paper, we proposed an effective encryption method for the scalable secured multiple ROIs and the conditional access control with the proposed key generation scheme. In order to encrypt the required ROI in SVC, we analyzed the characteristics of an ROI described in SVC stream and the requirements of the SVC encryption. Based on the analysis, the proposed encryption method protected the multiple ROIs selectively. In addition, we proposed conditional access control scheme using selective decryption in terms of the scalabilities of the encrypted NAL units and key generation scheme that reduces the number of NAL unit keys using access keys. Experimental results showed that the proposed encryption method and conditional access control scheme can provide the secured ROIs with users in an effective way.

References

1. ISO/IEC JTC 1/SC 29/WG 11: Joint Scalable Video Model (JSVM) 6.0 Reference Encoding Algorithm Description, N 8015, Switzerland (April 2006)
2. 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)
3. van der Schaar, M., Radha, H.: A Hybrid Temporal-SNR Fine-Granular Scalability for Internet Video. IEEE Trans. on Circuits and Systems for video Technology 11(3) (March 2001)
4. Liu, X., Eskicioglu, A.M.: Selective Encryption of Multimedia Contents in Distribution Network: Challenges and New Directions. In: IASTED International Conference on Communications, Internet and Information Technology (CIIT 2003), Scottsdale, AZ, November 17-19 (2003)
5. Zhu, B.B., Swanson, M.D., Li, S.: Encryption and Authentication for Scalable Multimedia, Current State of the Art and Challenges. In: Proc. SPIE Internet Multimedia Management Systems, October 2004, vol. 5601, pp. 157–170 (2004)
6. Won, Y.G., Bae, T.M., Ro, Y.M.: Scalable Protection and Access Control in Full Scalable Video Coding. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 407–421. Springer, Heidelberg (2006)
7. Bae, T.M., Thang, T.C., Kim, D.Y., Ro, Y.M., Kang, J.W., Kim, J.K.: Multiple Region-of-Interest Support in Scalable Video Coding. ETRI Journal 28(2) (April 2006)
8. Marpe, D., Schwarz, H., Wiegand, T.: Context-Based Adaptive Binary Arithmetic Coding in H.264/AVC Video Compression Standard. IEEE Trans. on Circuits and Systems for Video Technology 13(7), 620–636 (2003)
9. Zhu, B.B., Li, S., Feng, M.: A Framework of Scalable Layered Access Control for Multimedia. In: IEEE Int. Symposium on Circuit and Systems, May 2005, vol. 3, pp. 2703–2706 (2005)
10. Zhu, B.B., Feng, M., Li, S.: An Efficient Key Scheme for Layered Access Control of MPEG-4 FGS Video. In: IEEE Int. Conf. on Multimedia and Expo, June 2004, pp. 443–446 (2004)

Fusion Based Blind Image Steganalysis by Boosting Feature Selection

Jing Dong, Xiaochuan Chen, Lei Guo, and Tieniu Tan

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, P.O. Box 2728, Beijing, 10080
{jdong,xcchen,lguo,tnt}@nlpr.ia.ac.cn

Abstract. In this paper, a feature-level fusion based approach is proposed for blind image steganalysis. We choose three types of typical higher-order statistics as the candidate features for fusion and make use of the Boosting Feature Selection (BFS) algorithm as the fusion tool to select a subset of these candidate features as the new fusion feature vector for blind image steganalysis. Support vector machines are then used as the classifier. Experimental results show that the fusion based approach increases the blind detection accuracy and also provides a good generality by identifying an untrained stego-algorithm. Moreover, we evaluate the performance of our candidate features for fusion by making some analysis of the components of the fusion feature vector in our experiments.

Keywords: Blind image steganalysis, fusion, boosting.

1 Introduction

In the past few years, information hiding (e.g. steganography and watermarking) has drawn much attention. Steganography was dated from ancient Greece as the art of covert communication thousands of years ago [1]. The goal of steganography is to hide some secret messages into the cover image while no one else would notice the existence of the hidden information. Although the presence of embedded messages is often imperceptible to the human eyes, it may nevertheless change some statistics of an image. In contrast to steganography, steganalysis is then developed with the goal to detect whether a given medium contains embedded messages or not and furthermore to detect the length or even the exact content of the embedded messages. With the increasing demand of information security, we have witnessed the advance of both image steganography and steganalysis [2,3,4,5,6,7] during the past few years.

There are two kinds of image steganalysis techniques. One is called the specific steganalysis which is targeted at a particular known steganographic technique. The other is named blind (or universal) steganalysis which is effective over a wide variety of steganographic techniques. As the nature of image patterns and the steganographic algorithm are usually unknown beforehand, blind steganalysis techniques based on supervised learning schemes are more valuable in real world applications[4].

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 87–98, 2008.

The problem of blind steganalysis is approached by extracting a set of inherent features (e.g. some statistics) of images and then training a classifier on this data to distinguish the clean cover images from message-embedded stego images. The higher-order statistics of image and their other representations after image transformation are usually considered as the appropriate inherent features in the recent literature. *Farid et al.* [5] were perhaps the first to propose a universal supervised learning steganalysis scheme that uses a wavelet-like decomposition to build higher-order statistical models of natural images. These higher-order statistics appear to capture certain properties of "natural" images and are significantly altered when a message is embedded within an image. *Harmsen et al.* [6] used a three-dimensional feature obtained from the mass center (the first order moment) of histogram characteristic function (CF) for image steganalysis. The second and third order moments are also considered in this method. *Goljan et al.* [7] extracted the higher-order absolute moments of the noise residual from wavelet domain as features to capture the informative changes of the image before and after hiding information.

Since there are already many feature-based image steganalysis techniques proposed in the literature, and different features based on higher-order statistics are often slightly different in principle, one may ask:

- Is there any efficient strategy to combine different features together without bringing significant additional dimensional and computational complexity problems to classification?
- Among many kinds of higher-order statistics, which kind of statistical features works better than others? Is there any measurement?

To answer the two questions, a feature-level fusion based approach using Boosting Feature Selection (BFS) algorithm is proposed in this paper. In addition, we also evaluate the performance of different higher-order statistics used in fusion. The organization of this paper is as follows: Section 2 introduces the Boosting Feature Selection (BFS) algorithm. Section 3 describes the fusion approach with BFS for blind image steganalysis. Experimental results and comparison are presented in Section 4. Finally Section 5 concludes this paper.

2 Learning with Boosting

2.1 Boosting

Discrete AdaBoost [8] is a well-known learning algorithm used to boost the classification performance of a simple learning algorithm in a two-class classification problem. The basic idea of this algorithm is "boosting", which combines a set of weak classifiers to form a strong classifier. In the language of boosting, a weak classifier is called a weak learner. During the training stage, training samples are re-weighted according to the training error, and the weak learners trained later are focused on the miss-classification samples with higher weights. The final strong classifier is a weighted combination of the weak classifiers. The mathematic description is: Given a set of training data $(x_1, y_1), ...(x_M, y_M)$ with x_m a variable of the feature vector and $y_m = -1$ or $+1$ (here $+1$ denotes the positive samples and the -1 denotes the negative samples), one can define:

$$F(x) = \sum_{i=1}^{M} c_m f_m(x) \tag{1}$$

where each $f_m(x)$ is a classifier producing values ±1 and c_m are constants, and $sign(F(x))$ is the corresponding prediction results of final classification. The following is the outline of the Discrete AdaBoost algorithm (details may be found in [8]).

1. Start with weights $\omega_i = 1/N$, $i = 1, \cdots, N$,
2. Repeat for $m = 1, \cdots, M$,
   a. Fit the classifier $f_m(x) \in \{-1, 1\}$ using weights ω_i on the training data.
   b. Compute $err_m = E_\omega[1_{(y \neq f_m(x))}]$,$c_m = log((1 - err_m)/err_m)$.
   c. Set $\omega_i \longleftarrow \omega_i exp[c_m \cdot 1_{(y_i \neq f_m(x_i))}]$, $i = 1, \cdots, N$, and re-normalize so that $\sum_i \omega_i = 1$.
3. Output the classifier $sign[\sum_{m=1}^{M} c_m f_m(x)]$.

2.2 Boosting Feature Selection

The Boosting Feature Selection (BFS) algorithm proposed by *Tieu et al.*[9] was first to explicitly combine AdaBoost and ensemble feature selection together. The BFS algorithm differs from AdaBoost only by the way the weights are initialized and that each weak learner $f_m(x)$ is trained and does classification only based on one of the variables x_m in the input feature vector. The effective feature for classification is selected on the basis of lowest weighted error err_m for the given weighting ω_i. As the weighting changes, different input variables are selected for the current classifier $f_m(x)$. In mathematic description, one can denote:

$$f_m(x) = \beta_m b(x; \gamma_m) \tag{2}$$

where β_m is a multiplier and γ_m is the order of dimensionality of x in the whole input vector, and $b(.)$ denotes the m-*th* column of the input feature vector. In [8], Friedman et al. pointed out that one can solve for an optimal set of parameters through a "greedy" forward stepwise approach with updates:

$$\{\beta_m, \gamma_m\} \longleftarrow arg\min_{\beta,\gamma} E[y - F_{m-1}(x) - \beta b(x; \gamma)]^2 \tag{3}$$

for $m = 1, 2, ..., M$ in cycles until convergence, where $\{\beta_m, \gamma_m\}_1^{M-1}$ are fixed at their corresponding solution values at earlier iterations in the algorithm. In another word, depending on each γ_m, one can use a simple weak learner $f_m(x)$ in Eqn. (2) to form a powerful committee $F(x)$ in Eqn. (1). More specifically, we could obtain the most effective weak learner $f_m(x)$ to distinguish the positive from negative samples during iteration. The selected $f_m(x)$ also corresponds to an input variable x_m of the feature vector. Then after several iterations, a powerful classifier could be produced using only a manageable small sub-set of all input variables.

We notice that under the Boosting Feature Selection (BFS) framework, feature selection is applied directly on the feature space and an optimal subset of features is derived from these input variables. To the view of feature-level fusion, the fusion approach also operates in the feature space by gathering a combination of the input feature vectors. Therefore, we consider a fusion approach steganalysis by taking advantage of the BFS framework. If we provide several individual sets of steganalysis features for the BFS framework and employ it as a fusion tool, we could obtain a new fusion feature vector which consists of the re-combined set of features from the original feature sets. In addition, the final fusion feature vector also represents the set of de-correlated, redundancy-removed, "good" features which could distinguish the original image from the stego images. In the following section, we will describe our fusion based approach for blind image steganalysis in details.

3 Fusion for Blind Image Steganalysis by Boosting

3.1 Fusion Techniques

Information fusion has drawn much attention in recent years due to its ability to improve and enhance the accuracy and stability of certain system by combining some existing schemes. In general, most fusion techniques could be ascribed into three kinds: data-level fusion approach, feature-lever fusion approach and decision-level fusion approach [10].

To the best of our knowledge, little work has been done on fusion based image steganalysis. One piece of related work in *Kharazi et al.*[11] presents a case study on a decision-level fusion for image steganalysis by aggregating the outputs of multiple steganalysis systems. However, in the context of feature-level fusion for blind image steganalysis, our work is perhaps a first attempt. Traditional feature-level fusion techniques often combine all the individual feature vectors together into a sequence without any selection or dimensionality reduction. However, if the feature vector has high dimensionality, the computational complexity of learning and decision will increase. Thus, we need to keep the dimensionality low when we fuse different features together. Fig.1 shows the framework of our proposed scheme.

We select three types of higher-order statistics of existing steganalysis techniques in our approach, which will be introduced in details in the Section 3.2. Then the Boosting Feature Selection (BFS) is employed as the feature-level fusion strategy to combine these three sets of features and finally form the new fusion feature vector to train the classifier for blind image steganalysis.

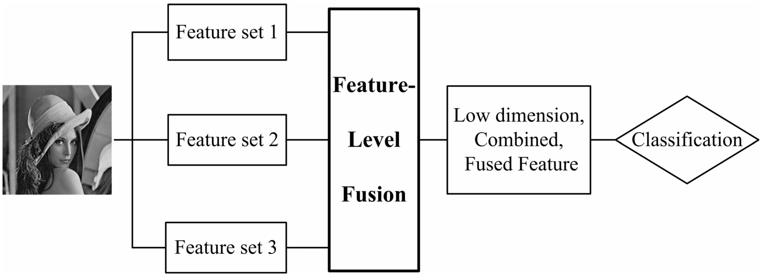


Fig. 1. The framework of the proposed feature-level fusion scheme

3.2 Choice of Candidate Features for Fusion

WHO features: *Farid et al.* [5] propose a useful blind steganalysis approach based on higher-order image statistics of wavelet decomposition. The decomposition employed in their approach is based on separable quadrature mirror filters (QMFs). This decomposition splits the frequency space into multiple scales and orientations. Given this image decomposition, a set of higher-order statistics is computed which includes the mean, variance, skewness and kurtosis of the sub-band coefficients at three orientations and at scales $i = 1, \cdots, n-1$. These statistics characterize the basic coefficient distributions. Another similar statistics set is based on the errors in an optimal linear predictor of coefficient magnitude where the mean, variance, skewness and kurtosis of the prediction error are obtained to form the error statistics. The combination of these two sets of coefficients statistics yields a total of $24(n-1)$ statistics that form a feature vector which is used to discriminate between images that contain hidden messages and those that do not. Here we use the particular 72-D feature vector proposed in their approach (denoted as WHO) as our first type of candidate features for fusion.

CF features: The second type of candidate features we used is the characteristic function moments, which refer to the moments of discrete characteristic function of the histogram. This type of features was used by, for example, *Harmsen* [6], *Xuan* [12] and *Shi* [13], and has been shown to be effective for image steganalysis. In our experiments, we employ the efficient 78-D feature vector proposed by *Shi et al.* [13] for fusion (denoted as CF). The CF features are obtained from the statistical moments of characteristic function of the test image after discrete wavelet transform (DWT). These moments can reflect the difference of the associated histograms, hence reflecting sensitively the changes caused by data hiding. The moments of the prediction error image are also included in the CF features.

EM features: Another type of candidate features we selected is the higher-order moments of empirical matrix of image. *Sullivan et al.* [14] pointed out that some characteristics of empirical matrix would be slightly changed after the embedding operation. This slight alteration of empirical matrix could be considered as an effective feature for blind image steganalysis. Here we choose the higher-order moments of the projection histogram (PH) of image empirical matrix proposed by *Chen et al.* [15] (denoted as EM). The EM features form a 108-D feature vector which is composed of two parts: the moments of the projection histogram (PH) of image and the moments of the characteristic function of PH.

3.3 Fusion Approach

We then apply the BFS algorithm to train the three candidate feature sets for fusion—the three feature vectors which we outlined above (denoted by WHO, CF and EM). After n iterations (the step of iteration could be assigned as long as it is sufficient to keep training error small enough), we would get a new subset of input candidate feature vectors as our fusion feature set with good performance for blind image steganalysis. Fig. 2 shows the overview of this fusion approach and the particular description of our approach is presented as follows.

Step 1. We directly cascade the three candidate feature vectors mentioned above to form a totally 258-D feature vector which is denoted as $\mathbf{X} = \{x_1, x_2, \cdots, x_M, M{=}72{+}78{+}108{=}258\}$.

Step 2. Then we implement the Boosting Feature Selection (BFS) algorithm to train the input vector $\mathbf{X}$. According to $f_m(x) = \beta_m b(x; \gamma_m)$, here every input variable x_m in $\mathbf{X}$ is also a weak learner f_m.

Step 3. The parameters of γ_m could be obtained in the first n iterations. We can also obtain the weak learner f_m, which depends on the corresponding solution values of the parameters γ_m in Eqn. (3). After n iterations, we could finally obtain a combination of the weak learners $\{f_m,\ m = 1, 2, \cdots, N\}$.

Step 4. We select those corresponding input variables $\{x_m\ ,\ m = 1, 2, \cdots, N\}$ to form the N-D, fused new feature vector $\mathbf{X_n}$ for the classification of blind image steganalysis. This fused feature vector $\mathbf{X_n}$ also represents an effective combination of the subset features from the input candidate features.

According to the BFS framework, higher-weighted weak learners would always be selected in the earlier iterations. In another word, the input feature with better discrimination would be more likely to be selected to form the fusion feature by BFS. Consequently, we could also evaluate which type of the candidate feature sets is better than the others based that which feature set provides the largest number of variable x_m in the fusion feature vector $\mathbf{X_n}$. For example, if the variables in $\mathbf{X_n}$ mostly consist of the variables of WHO feature vector, then a reasonable judgment could be made as follows: the WHO candidate features contribute more than the other two types of candidate features used in fusion.

3.4 Classifier

Considering the goal in the experiments is to utilize a unified classifier so as to form a common base for the performance comparisons of different features, we

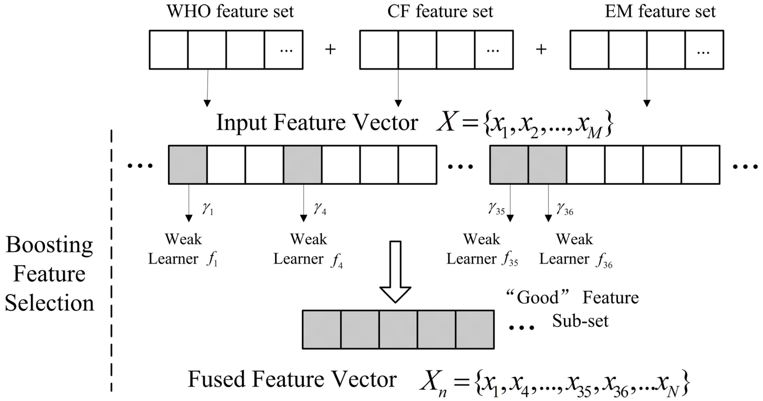


Fig. 2. Overview of the fusion approach for image steganalysis by Boosting Feature Selection

focus on the choice of the features rather than the classifier. Since Support Vector Machine (SVM) is considered as the optimal classifier and it is computationally efficient for large scale learning with small samples in the literature, the SVM^{light} [16] is applied as the classifier for our experimental testing and a non-linear kernel is chosen. All the comparisons are tested on the same database and classifier.

4 Experiments

In this section,we experimentally investigate how much improvement in blind steganalysis we can obtain by the proposed fusion approach. We also evaluate the effectiveness of the candidate features for fusion in the experiments.

4.1 Database

For this experiment,we choose a commonly used image database, the CorelDraw Database, and totally 1569 images from CorelDraw version 11 CD# 4 were collected as the original images. Also, six sets of stego images were generated by using the following six typical stego-algorithms:

a) SS: non-blind spread spectrum method proposed by *Cox et al.* [17]. (0.15bpp, 36dB)
b) Huang: 8 × 8 DCT block SS method proposed by *Huang et al.* [18]. (0.1bpp, 48dB)
c) LSB: generic LSB embedding method. (0.3bpp, 56dB)
d) Lie: adaptive LSB proposed by *Lie et al.* [19]. (0.3bpp, 51dB)
e) QIM: generic quantization index modulation method proposed by*Chen et al.* [20]. (0.11bpp, 47dB)
f) Rand: random ±1 embedding method [7]. (0.2bpp, 48dB)

To make our tests more convincing we just embed a small amount of secret data in these data hiding methods. The approximate average embedding rates and the PSNR are shown in brackets.

4.2 Detection Performance

In our experiment, we typically run AdaBoost for 75 iterations which are already sufficient to achieve a training error small enough. Therefore, after fusion, we obtain a 75-D new feature vector for the next classification. Then, we test our fusion features as well as the selected three types of candidate features (WHO, CF and EM) on the above database. The same training and testing procedures are used. All the experiments are repeated 5 times, and the average rate is recorded for each run.

We evaluate the system with each one of the six data hiding methods at a time. We randomly select 800 original images and their corresponding 800 stego images for training each time. The remaining 769 pairs are then used for testing. The true positive detection rate (TPR) is defined as the ratio of the number in correctly classified images out of the overall test images. The false positive rate (FPR) represents the ratio of wrongly classifying the plain images as stego ones. In practical applications of blind image steganalysis, we should keep FPR as low as possible while enhancing the TPR. The comparisons of the obtained TPR results of the four types of features (as described by WHO, CF, EM and Fusion) are listed in the first six rows in Table 1, where the FPR is fixed at 5.6%.

In order to evaluate the blind steganalysis ability of our proposed fusion based approach, we combine the six data hiding methods in a mixed mode. Similar to the above tests, 800 original images are randomly selected for training. But this time, their corresponding training stego images are not from one particular set of stego images but the mixed six sets of stego images. That means the whole training samples consist of 800 original images and 4800(6 × 800) stego images. Then the remaining 769 original images and corresponding 4614(6 × 769) stego images are used to test. The comparison of the obtained TPR results is also shown in the "Mixed" mode of Table 1. The corresponding ROC curves are shown in Fig. 3.

In addition, another experiment is designed to test the generality of the fusion based approach for blind image steganalysis. In this test, the training procedures are exactly the same as above but during testing, we replace one set of 769 stego images which were produced by the RAND stego-algorithm by another set of 769 stego images generated by an untrained stego-algorithm proposed in [21](The embedding rates and the PSNR are 0.15bpp and 52dB). The corresponding ROC curves are shown in Fig.4.

From Table 1 and Fig. 3, we can observe that the proposed fusion feature set performs better than the individual feature sets. In the "mixed" mode, the correct detection rate of the fusion features is significantly higher than the other features. That indicates our fusion method provides good performance on blind image steganalysis. Besides, Fig. 4 shows that the fusion approach has better generality for blind image steganalysis by enabling effective identification on the stego images which are generated by an untrained stego-algorithm. Moreover, compared with traditional feature-level fusion which cascades all the candidate features together, the proposed fusion feature vector keeps the dimensionality low (75-D vs. 258-D) and in the meantime enhances the performance of the universal steganalysis system.

Table 1. The detection accuracy of the proposed fusion approach in comparison with that of the WHO, CF and EM features with 5.6% false positives

Stego-algorithm	WHO	CF	EM	Fusion
SS	68.11%	87.52%	80.18%	**90.60%**
Huang	73.12%	83.13%	76.86%	**87.74%**
LSB	60.72%	92.45%	91.91%	**95.22%**
Lie	63.35%	91.68%	85.52%	**94.76%**
QIM	81.29%	96.14%	97.74%	**99.01%**
Rand	71.58%	94.07%	90.45%	**96.15%**
Mixed	70.36%	88.31%	86.74%	**95.44%**

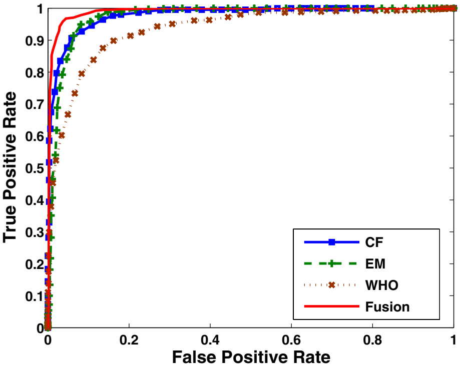


Fig. 3. The ROC curves for the "Mixed" mode, where the test image samples consist of randomly selected images from one original sample and the mixed six corresponding stego samples

4.3 Feature Evaluation

We make some analysis of our experimental results and obtain some more detailed information about the fusion feature vector. We calculate the numbers of the feature variable x_m which build up our fusion feature vector. We want to compare which type of candidate features for fusion (WHO, CF, and EM) contributes more and provides more variables to our 75-D fusion feature vector. Table 2 shows the statistical results. The figures in each row indicate the number of the selected variable x_m during AdaBoost training step from the corresponding type of candidate features. The percentages in brackets represent the

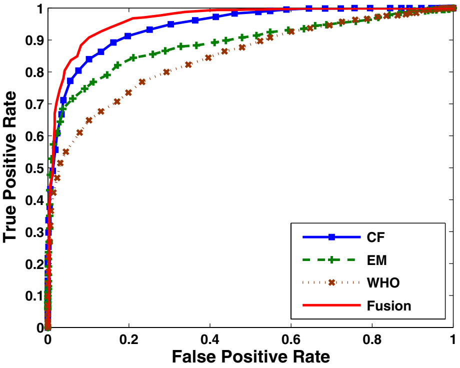


Fig. 4. ROC curves for the designed blind steganalysis when employed to identify the test samples where one set of them are generated by an untrained stego-algorithm

contribution of each type of candidate features to the fusion feature vector. For example, the results in Row 1 represent that the 75-D fusion feature vector for SS detection test consists of 11 variables (15.3% = 11/72) from WHO features, 39 variables (50% = 39/78) from CF features and 25 variables (23.1% = 25/108) from EM features.

Table 2. Feature evaluation by analyzing the components of the 75-D fusion feature vector. The percentages represent the contribution of each type of candidate features.

Stego-algorithm	WHO	CF	EM
SS	11-D (15.3%)	39-D (50%)	25-D (23.1%)
Huang	9-D (12.5%)	36-D (46.2%)	30-D (27.8%)
LSB	6-D (8.3%)	38-D (48.7%)	31-D (28.7%)
Lie	8-D (11.1%)	47-D (60.3%)	20-D (18.5%)
QIM	12-D (16.7%)	30-D (38.5%)	33-D (30.6%)
Rand	9-D (12.5%)	37-D (47.4%)	29-D (26.9%)
Mixed	7-D (9.7%)	43-D (55.1%)	25-D (23.1%)

From Table 2, we can observe that the fusion feature vector is statistically mostly made up of the sub-set of CF features. It can be concluded that the type of CF features made more contribution to the fusion features and hence made more contribution to the classification of the original and stego images. Therefore, we could reasonably consider that the type of CF features performs better than the other two types of candidate features in our approach. A similar conclusion would also be made between the WHO features and the EM features. These two conclusions are also consistent with our experiment results in Table 1.

5 Conclusion

In this paper, we have proposed a feature-level fusion based blind image steganalysis approach by Boosting Feature Selection. We have selected three types of typical higher-order statistics as our candidate features for fusion. The BFS algorithm is used as the feature-level fusion tool to obtain a sub-set of the candidate features as the final fusion feature vector. The experimental results have demonstrated that out proposed fusion approach improves the system performance without bringing significant additional dimensional and computational complexity and at the same time provides a satisfying blind image steganalysis ability. We have also evaluated the effectiveness of the candidate features for fusion by analyzing the composition of the fusion feature vector. In conclusion, the purpose of our fusion approach is two-folds: 1) As the first attempt for feature-level fusion of blind image steganalysis, it could enhance the performance of existing steganalysis techniques and could provide a satisfying generality for blind detection; 2) it also serves as a measurement of the effectiveness of different features for image steganalysis.

The proposed fusion based image steganalysis approach is by no means optimal though we have obtained satisfying experimental results. To achieve better performance, various improvements can be made and more representative features could be chosen for fusion. The approach in this paper is just an attempt and case study since a large number of steganalysis techniques have been proposed in the literature but no one is the best for application. We have provided an idea rather than a technique to take the advantages of AdaBoost, which is considered more likely as a classification method to be used for steganalysis as the fusion technique for the first time. And we believe that the applications of fusion techniques in image steganalysis are not limited in the example we have studied in this paper. For example, one could improve and expand his steganalysis method by including more different efficient features, or even to extend his approach to fuse different steganalysis methods such as estimation of the embedded message length. This idea may also be valuable in other forensic analysis area besides image steganalysis.

Acknowledgments. The work on this paper was supported by Nature Science Foundation of China (Grant No.60603011) and National Key Technology R&D Program (Grant No.2006BAH02A13).

References

1. Johnson, N.F., Jajodia, S.: Exploring steganography: Seeing the unseen. In: Computer, vol. 31, pp. 26–34. IEEE Computer Society, Los Alamitos (1998)
2. Provos, N.: Defending against statistical steganalysis. In: Proceedings of the 10th USENIX Security Symposium, pp. 323–336 (2001)
3. Fridrich, J., Goljan, M.: Practical steganalysis of digital images — state of the art. In: Security and Watermarking of Multimedia Contents, vol. SPIE-4675, pp. 1–13 (2002)

4. Tzschoppe, R., Aauml, R.B.: Steganographic system based on higher-order statistics. In: Proceedings of SPIE, Security and Watermarking of Multimedia Contents V, USA, vol. 5020 (2003)
5. Farid, H.: Detecting hidden messages using higher-order statistics and support vector machines. In: 5th International Workshop on Information Hiding (2002)
6. Harmsen, J.J., Pearlman, W.A.: Steganalysis of additive noise modelable information hiding. In: Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents VI, pp. 131–142 (2003)
7. Goljan, M., Fridrich, J., Holotyak, T.: New blind steganalysis and its implications. In: Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents VI, pp. 1–13 (2006)
8. Friedma, F., Hastie, T.: Additive logistic regression: a statistical view of boosting (1998)
9. Tieu, K., Viola, P.: Boosting image retrieval. In: IEEE Conf.on Computer Vision and Pattern Recognition, pp. 228–235 (2002)
10. Jain, A.K.: Score normalization in multimodal biometric systems. Pattern Recognition 38, 2270–2285 (2005)
11. Kharrazi, M.: Improving steganalysis by fusion techniques: A case study with image steganography. In: Tran. On Data Hiding and Multimedia Security, pp. 123–137 (2006)
12. Xuan, G., Shi, Y.Q.: Steganalysis based on multiple features formed by statistical moments of wavelet characteristics functions. In: Proc. Information Hiding Workshop, pp. 262–277 (2005)
13. Shi, Y.Q., et al.: Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network. In: ICME 2005, pp. 269–272 (2005)
14. Sullivan, K., et al.: Steganalysis for markov cover data with applications to images. IEEE Trans. Inf. Forensics Security 1, 275–287 (2006)
15. Chen, X.C., Wang, Y.H., Guo, L., Tan, T.N.: Blind image steganalysis based on statistical analysis of empirical matrix. In: ICPR (3) 2006, pp. 1107–1110 (2006)
16. Joachims, T.: Making large-scale svm learning practical, in adavances in kernel methods-support vector learning. In: Scholkopf, B., Burges, C. (eds.). MIT Press, Cambridge (1999)
17. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectru, watermarking for multimedia. IEEE Trans.Image Process 6(12), 1673–1687 (1997)
18. Huang, J., Shi, Y.Q.: Adaptive image watermarking scheme based on visual masking. Electron, Letter 34, 748–750 (1998)
19. Lie, W.N., Chang, L.C.: Data hiding in images with adaptive numbers of least significant bits based on human visual system. In: Proc. IEEE Int. Conf. Image Processing, pp. 286–290 (1999)
20. Chen, B., Wornell, G.W.: Digital watermarking and information embedding using dither modulation. In: Proceedings of IEEE MMSP, pp. 273–278 (1998)
21. Piva, A., Barni, M., Bartolini, E., Cappellini, V.: Dct-based watermark recovering without resorting to the uncorrupted original image. In: Proc. ICIP 1997, vol. 1, p. 520 (1998)

Steganalysis of Multi Bit Plane Image Steganography

Johann Barbier[1,2] and Kichenakoumar Mayoura[1]

[1] École Supérieure et d'Application des Transmissions,
Laboratoire de Virologie et Cryptologie,
BP 18, 35998 Rennes Cedex, France
kichenakoumar.mayoura@esat.defense.gouv.fr

[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. In this paper, we present an efficient specific steganalysis scheme for multi bit plane image steganography. This algorithm proposed by Nguyen, Yoon and Lee at IWDW'06 is applied to uncompressed images and embeds data into several bit planes of the canonical gray coding. It is designed to be robust against RS steganalysis and pixel difference histogram analysis. We show how to adapt RS analysis into a local analysis to design an efficient detector against multi bit plane image steganography. This method takes advantages from both the counter-measures introduced by the authors and the power of RS analysis approach. Finally, we designed a specific classifier to detect images embedded with Nguyen's algorithm. This detector has high detection rates, even for low embedding rates.

Keywords: Multi bit plane image steganography, Fisher discriminant, RS steganalysis.

1 Introduction

Different kinds of steganography algorithms have been developed so far. For each vehicle for information, it exists a mean to embed data and so using it as a subliminal channel. Most of time, the security ensured by cryptography is enough but in particular cases, hiding the very existence of the communication appears to be as important as the protection of the information itself. For instance, an agent who infiltrates the mafia would try to cipher but also to hide its communications with the Police. Steganography seems to be the solution for this new security need. Nowadays, images appear to be one of the most widespread means for hiding data. For many reasons, Least Significant Bit (LSB) algorithms, and particularly those for uncompressed images, are the most studied in steganography. This popularity is mainly due to their simplicity and a good capacity. Such algorithms were the first ones to be developed, but the number of attacks

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 99–111, 2008.

dedicated to such schemes lead us to think they are not the best way to hide data.

The Multi Bit Plane Image Steganography (MBPIS) in the Canonical Gray Coding (CGC) was proposed by Nguyen, Yoon and Lee [1] at IWDW'06. The authors claim that their data hiding method is secure against RS analysis and pixel difference histogram analysis. A similar approach has been developed by Agaian, Rodriguez and Perez [2]. The main common characteristics to such techniques are to embed information in multi bit planes, to change the initial coding domain and to take advantage of non-informative areas of the image in order to protect from classical steganalysis like RS analysis [3], χ^2 attacks [4] and Pairs Analysis [5,6]. But the immunity of MPBIS against some other classical steganalysis techniques [7,8] has not been tested so far. We present how to adapt RS analysis in order to design an efficient classifier to detect the use of MBPIS.

The paper is organized as follows. In the first section, we describe in details the Multi Bit Plane Image Steganography and present the counter-measures designed by the authors of [1] to be robust against RS analysis. Then, we briefly describe RS steganalysis and introduce our approach. We show how to convert RS analysis into a local steganographic detector against MBPIS. In the last section, we discuss about the immunity of MBPIS against RS analysis, present an illustration of the statistical features we pointed out and give some experimental results.

2 The Multi Bit Plane Image Steganography

The Multi Bit Plane Image Steganography (MBPIS) was proposed by Nguyen, Yoon and Lee [1] at IWDW'06. This algorithm is designed to be secure against several classical steganalysis methods like RS steganalysis. The main goal of this paragraph is to detail this steganography algorithm which is dedicated to uncompressed images. Two effects are expected by using MBPIS; first to avoid the human visual analysis and then the non-random changes of pixels values. One of its most important properties consists in locating the non-noisy areas of bit planes also called *flat areas.* In smooth regions of the cover image, pixels have similar values. The embedding process may add noise to non-noisy areas and therefore some steganalysis may succeed. Hence the flat areas are isolated and are not modified during the embedding process.

Another feature of the designed algorithm is to embed data into the Canonical Gray Coding (CGC). For that, an N-bit pixel value in Pixel Binary Code (PBC) $b_N b_{N-1} \dots b_1$ is converted into an N-bit pixel value in CGC, $g_N g_{N-1} \dots g_1$ as follows:

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

The inverse transformation is described below:

$$\begin{cases} b_N = g_N \\ b_i = g_i \oplus b_{i+1} \; 1 \leq i \leq N-1. \end{cases} \tag{2}$$

The flip of several bits of a CGC pixel involves a change of the pixel value scattered in a non-step range, *i.e.* changing a bit b_i of a pixel in the CGC representation causes a change of its value in the range $[1, 2^i - 1]$. The image is first decomposed into N CGC bit planes $B_N B_{N-1} \dots B_1$. In order to avoid visual detection caused by the degradation of the image, the number of bit planes i_{max} to embed is no more than 4. We embed the message in the cover medium from higher to lower bit planes (from $B_{i_{max}}$ to B_1).

The smooth areas for a certain bit plane B_i, are obtained by combining smaller smooth areas, of sizes $n \times n$ where $1 \leq n \leq$ (height or width of the image). For this, a sliding and non-overlapping window of size $n \times n$ goes through the image. Let us consider

$$W = \begin{bmatrix} p_1 \; p_2 \\ p_3 \; p_4 \end{bmatrix} \quad , \tag{3}$$

such a window, where p_i are pixel values. Then, we compute

$$\begin{bmatrix} |p_1 - p_1| \; |p_1 - p_2| \\ |p_1 - p_3| \; |p_1 - p_4| \end{bmatrix} = \begin{bmatrix} p_1^{'} \; p_2^{'} \\ p_3^{'} \; p_4^{'} \end{bmatrix} \quad . \tag{4}$$

Finally, if $\forall k$, $\lceil \frac{p_k^{'}}{2^i} \rceil \leq t$, where t is the threshold, then W is said to be *flat* and *non-flat* otherwise. This definition can be extended to more general windows of size $m \times n$. The secret message is then embedded into the non-flat areas of bit plane B_i, using a pseudo-random sequence. We describe now the embedding and the extracting algorithms.

2.1 Embedding Algorithm

```
Input : $l$-bit secret message $M$, an uncompressed image $I$
        $K_e$, $K_s$ the cryptographic and steganographic keys

Output : stego-image $I^{'}$ or failure

Parameters : the higher bit plane $i_{max}$, the threshold $t$
             the size $m \times n$ of the sliding window.

1. Transform $I$ into $I^{'}$ from PBC to CGC according to (1)
2. Decompose $I^{'}$ into N-bit planes
3. Compress and encrypt $M$ with $K_e$
4. Init the Pseudo-Random Generator with $K_s$
5. for $i$ from $i_{max}$ to 1
   – Find all $m \times n$ flat areas in bit plane $B_i$ with threshold $t$ according to (4)
   – Randomly embed the message in the bits of $B_i$ of the non-flat areas using the pseudo-random sequence
6. If some bits of the message has not been embedded return failure
7. Transform $I^{'}$ from CGC to PBC according to (2)
8. Return $I^{'}$
```

2.2 Extracting Algorithm

```
Input : a stego image $I^{'}$
        $K_e$, $K_s$ the cryptographic and steganographic keys

Output : the $l$-bit secret message $M$

Parameters : the higher bit plane $i_{max}$, the threshold $t$
             the size $m \times n$ of the sliding window.

1. Transform $I^{'}$ from PBC to CGC according to (1)
2. Decompose $I^{'}$ into N-bit planes
3. Init the Pseudo-Random Generator with $K_s$
4. for $i$ from $i_{max}$ to 1
   – Find all $m \times n$ flat areas in bit plane $B_i$ with threshold $t$ according to (4)
   – Extract the message $M$ in the non-flat areas of $B_i$ using the pseudo-random sequence
5. Decrypt $M$ with $K_e$ and decompress it
6. Return $M$
```

3 Adapting the RS Steganalysis

3.1 RS Steganalysis

RS steganalysis was introduced by Fridrich, Goljan and Du [3] for detecting the use of LSB steganography. This technique was designed for spatial domain but can also fit all kinds of LSB steganography. Moreover, RS steganalysis makes possible the estimation of the embedded message length. In this section, we describe briefly this technique in order to better understand the proposed adaptation. We only detail it for grayscale image but it can easily be generalized component by component for color images.

The uncompressed grayscale image is first divided into non-overlapping groups of n consecutive pixels. In practice $n = 4$. Now, let us consider such a group $G = (x_1, \ldots, x_n)$. The smoothness of this group is evaluated by applying the discrimination function f, defined by

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \ . \tag{5}$$

An invertible mapping F, called *flipping* and defined on $[0, 255]^n$ is also introduced. F is based on

$$F_1 : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255 \ , \tag{6}$$

$$F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256 \ , \tag{7}$$

and F_0 as the identity permutation on $[0, 255]$. Finally, F is defined as follows

$$\begin{aligned} F_M : [0,255]^n &\longrightarrow [-1,256]^n \\ x = (x_1, \ldots, x_n) &\longrightarrow F_M(x) = (F_{M(1)}(x_1), \ldots, F_{M(n)}(x_n)) \ , \end{aligned} \tag{8}$$

where M is called a *mask* and is a n-uple with values -1, 0 and 1. For each group G, we compute $F(G)$ and classify it into three types of pixel groups, R, S and U such that

$$\begin{aligned} G \in R &\Leftrightarrow f(F(G)) > f(G) \ , \\ G \in S &\Leftrightarrow f(F(G)) < f(G) \ , \\ G \in U &\Leftrightarrow f(F(G)) = f(G) \ . \end{aligned} \tag{9}$$

Then, we evaluate R_M, the proportion of groups in R for the mask M, S_M the relative number of groups in S for M. In the same way we compute R_{-M} and S_{-M}, where $-M$ is the *negative mask*, such that $[-M](i) = -M(i)$ for all i in $[1, n]$. The very hypotheses of RS steganalysis rely on the fact

$$R_M \cong R_{-M} \text{ and } S_M \cong S_{-M} \ , \tag{10}$$

for a non embedded image. For more details refer to [3]. Then, we compute R_M, S_M, R_{-M} and S_{-M} with an embedding rate from 0 to 100% and obtain the RS-diagram as shown in Fig. 1.

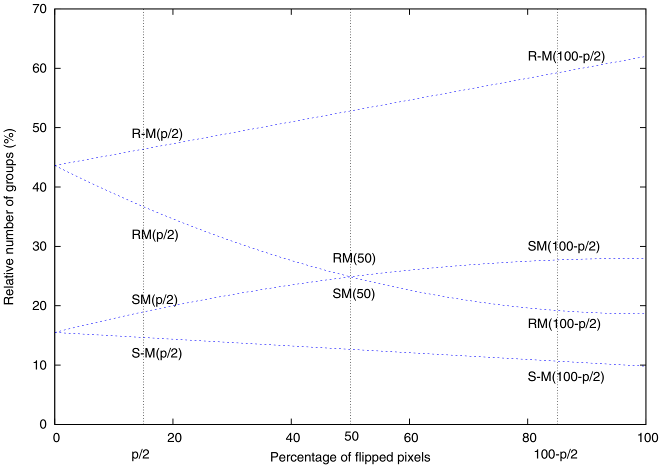


Fig. 1. RS-diagram of an image embedded with an embedding rate p using $M = [0110]$

Finally, for a given image with an embedding rate of p, we calculate $R_M(p/2)$, $S_M(p/2)$, $R_{-M}(p/2)$ and $S_{-M}(p/2)$ as our initial measurement, $R_M(1-p/2)$, $S_M(1-p/2)$, $R_{-M}(1-p/2)$ and $S_{-M}(1-p/2)$ when flipping all the LSBs and

$R_M(1/2)$, $S_M(1/2)$, $R_{-M}(1/2)$ and $S_{-M}(1/2)$ by randomizing the LSB plane. Given these measurements, we are able to interpolate the RS-diagram, considering quadratic curves. First, we rescale the x axis so that $p/2$ becomes 0 and $100-p/2$ becomes 1. Then, we calculate the coordinate of the intersection point as the root of the quadratic equation

$$2(d_1+d_0)x^2+(d_{-0}-d_{-1}-d_1-3d_0)x+d_0-d_{-0}=0\ , \tag{11}$$

where

$$\begin{aligned} d_0 &= R_M(p/2)-S_M(p/2)\ , \\ d_{-0} &= R_{-M}(p/2)-S_{-M}(p/2)\ , \\ d_1 &= R_M(1-p/2)-S_M(1-p/2)\ , \\ d_{-1} &= R_{-M}(1-p/2)-S_{-M}(1-p/2)\ . \end{aligned}$$

If x is the root whose the absolute value is smaller then p is given by

$$p=\frac{x}{x-\frac{1}{2}}\ . \tag{12}$$

3.2 Local RS Steganalysis

As the authors claim, MBPIS is secure against a straight forward application of the classical RS steganalysis. This point is discussed in the next section. Moreover, the MBPIS algorithm selects specific areas to hide data, that means not all pixels are available for embedding. Since the MBPIS algorithm is a multi bit plane algorithm, we define the embedding rate as the ratio $\frac{\text{size of the data}}{\text{size of the cover file}}$. One difficulty introduced by the MBPIS is that the capacity is only determined step by step during the embedding process, as the non-flat areas of a bit plane depend on the embedding in the higher bit planes.

As a multi bit plane algorithm, MBPIS changes not only the LSB bit plane but also higher order ones, so computing $R_M(1-p/2)$, $S_M(1-p/2)$, $R_{-M}(1-p/2)$ and $S_{-M}(1-p/2)$ can not be achieved when flipping only all the LSBs. We have also to take into account that MBPIS embeds from the higher bit planes to the lower ones, that implies the LSBs are most of the time unchanged. Evaluating the RS-diagram seems to be impossible if we apply RS analysis in a straight forward manner. In this section, we present how to adapt RS analysis focusing only on the non flat-areas. Such a choice is motivated by the facts that only non-flat areas contain hidden information and they can be determined without the knowledge of the stego key. Our approach can be seen as a local RS-analysis. We illustrate it with 8-bit grayscale images and precise how to generalize it to color images at the end of the section.

First, according to step 2 of the MBPIS extraction algorithm [1], we compute all the non-flat areas of size $m\times n$. These areas can be considered as groups of $m\times n$ pixels. Now let us denote by $\hat{G}=\begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{mn}\end{pmatrix}$ such a group. We evaluate the smoothness of $\hat{G}$ by applying the discrimination function $\hat{f}$ defined by

$$\hat{f}\left(\hat{G}\right) = \left(\left(\begin{matrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{mn} \end{matrix}\right)\right) \tag{13}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n-1} |x_{i,j+1} - x_{i,j}| + \sum_{j=1}^{n}\sum_{i=1}^{m-1} |x_{i+1,j} - x_{i,j}| \ .$$

We also define $\hat{F}$, the flipping by

$$\begin{aligned} \hat{F} : \mathcal{M}_{m\times n}\left([0,255]\right) &\longrightarrow \mathcal{M}_{m\times n}\left([-1,256]\right) \\ \hat{G} = \left(\begin{matrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{mn} \end{matrix}\right) &\longrightarrow \hat{F}\left(\hat{G}\right) \ , \end{aligned} \tag{14}$$

where

$$\hat{F}\left(\hat{G}\right) = \left(\begin{matrix} F_{M(1,1)}(x_{11}) & \dots & F_{M(1,n)}(x_{1n}) \\ \vdots & & \vdots \\ F_{M(m,1)}(x_{m1}) & \dots & F_{M(m,n)}(x_{mn}) \end{matrix}\right) \ ,$$

and M is a $m \times n$ matrix of values $-1, 0, 1$ and F_i are given by (6) and (7). For each group $\hat{G}$, we compute $\hat{F}(\hat{G})$ and classify it into three types of pixel groups, R, S and U such as

$$\begin{aligned} \hat{G} \in R &\Leftrightarrow \hat{f}(\hat{F}(\hat{G})) > \hat{f}(\hat{G}) \ , \\ \hat{G} \in S &\Leftrightarrow \hat{f}(\hat{F}(\hat{G})) < \hat{f}(\hat{G}) \ , \\ \hat{G} \in U &\Leftrightarrow \hat{f}(\hat{F}(\hat{G})) = \hat{f}(\hat{G}) \ . \end{aligned} \tag{15}$$

Then, we evaluate R_M, the proportion of groups in R for the mask M, S_M the relative number of groups in S for M. In the same way we compute R_{-M} and S_{-M}, where $-M$ is the *negative mask*, such as $[-M](i,j) = -M(i,j)$ for all i,j in $[1,m] \times [1,n]$. M is given by $M(i,j) = 1$ if and only if $i+j$ is even and $M(i,j) = 0$ otherwise. Additionally, we also calculate

$$\mathcal{Q}_R = \frac{R_M - R_{-M}}{R_M} = 1 - \frac{R_{-M}}{R_M} \ , \tag{16}$$

$$\mathcal{Q}_S = \frac{S_M - S_{-M}}{S_M} = 1 - \frac{S_{-M}}{S_M} \ . \tag{17}$$

As we are not able to compute exactly the entire classical RS-diagram but only a small part, we can not interpolate the RS-curves as explained above. Unfortunately, using the local RS analysis, we are not able to determine an estimation of the embedding rate. We only have a local vision of the RS-diagram. But the theory developed in [3] implies that the difference between R_M and R_{-M} and between S_M and S_{-M} locally increases with the embedding rate, as illustrated in the Fig. 1. To take advantage of that, we have introduced the relative differences $\mathcal{Q}_R$ and $\mathcal{Q}_S$. Finally, we map the analyzed image I to a statistical vector of 6 coordinates, such as

$$I \longrightarrow \mathcal{V}(I) = (R_M, R_{-M}, S_M, S_{-M}, \mathcal{Q}_R, \mathcal{Q}_S) \ . \tag{18}$$

This feature vector is the central point of our approach. As we do not care about the flat areas, these features are no more negligible and can be exploited to design a highly discriminating detector. For cover images we have

$$R_M \cong R_{-M} \ , \ S_M \cong S_{-M} \text{ and } \mathcal{Q}_R \cong \mathcal{Q}_S \cong 0 \ . \tag{19}$$

On the contrary, R_M,S_M, $|\mathcal{Q}_R|$, $|\mathcal{Q}_S|$ increase and R_{-M}, S_{-M} decrease with the embedding rate. We detect this statistical deviation using a linear Fisher discriminant as described in section 4.3. As the bit plane number i_{max} is the most changed by the algorithm, we only compute the statistical vector with only the flat areas of this bit plane. This technique can easily be generalized to color images. For each component RGB we compute the R and S values and keep the mean of them for R_M, R_{-M}, S_M and S_{-M}.

4 Experimental Results

4.1 RS Steganalysis Immunity

The MBPIS algorithm is designed to be secure against RS steganalysis. Actually, the discrimination function is more sensitive to embedding in smooth areas, since changing the value of a pixel in such an area increases the discontinuity of values inside a group of pixels. That is why non-flat areas have very small impact on RS analysis compared to the flat ones. Since MBPIS does not embed data into flat areas, the distortions introduced can not be detected with RS analysis if we consider all the areas.

To illustrate this, let us consider $\mathcal{H}$, the set of flat areas, $\mathcal{E}$, the MBPIS embedding algorithm. We evaluate now the relative variations, ΔR_M, ΔS_M, ΔU_M, of the RS coefficients when embedding with $\mathcal{E}$.

$$\Delta R_m = \left|\frac{\mathcal{P}r(\mathcal{E}(G) \in R) - \mathcal{P}r(G \in R)}{\mathcal{P}r(G \in R)}\right| = \left|\frac{\mathcal{P}r(\mathcal{E}(G) \in R)}{\mathcal{P}r(G \in R)} - 1\right| \ . \tag{20}$$

ΔS_M and ΔU_M are defined in the same way. Moreover, we have $\frac{\mathcal{P}r(\mathcal{E}(G) \in R)}{\mathcal{P}r(G \in R)}$

$$= \frac{\mathcal{P}r\,(G \in \mathcal{H})\,\mathcal{P}r\,(\mathcal{E}(G) \in R \,|\, G \in \mathcal{H}) + \mathcal{P}r\,(G \notin \mathcal{H})\,\mathcal{P}r\,(\mathcal{E}(G) \in R \,|\, G \notin \mathcal{H})}{\mathcal{P}r\,(G \in R)} \ ,$$

$$= \frac{\mathcal{P}r\,(G \in \mathcal{H})\,\mathcal{P}r\,(G \in R \,|\, G \in \mathcal{H}) + \mathcal{P}r\,(G \notin \mathcal{H})\,\mathcal{P}r\,(\mathcal{E}(G) \in R \,|\, G \notin \mathcal{H})}{\mathcal{P}r\,(G \in \mathcal{H})\,\mathcal{P}r\,(G \in R \,|\, G \in \mathcal{H}) + \mathcal{P}r\,(G \notin \mathcal{H})\,\mathcal{P}r\,(G \in R \,|\, G \notin \mathcal{H})} \ ,$$

as MBPIS does not embeds in flat areas, $\mathcal{E}(G) = G$.

$$\frac{\mathcal{P}r\left(\mathcal{E}(G)\in R\right)}{\mathcal{P}r\left(G\in R\right)} = \frac{1+\frac{\mathcal{P}r(G\notin\mathcal{H})}{\mathcal{P}r(G\in\mathcal{H})}\frac{\mathcal{P}r(\mathcal{E}(G)\in R\,|\,G\notin\mathcal{H})}{\mathcal{P}r(G\in R\,|\,G\in\mathcal{H})}}{1+\frac{\mathcal{P}r(G\notin\mathcal{H})}{\mathcal{P}r(G\in\mathcal{H})}\frac{\mathcal{P}r(G\in R\,|\,G\notin\mathcal{H})}{\mathcal{P}r(G\in R\,|\,G\in\mathcal{H})}} = \frac{1+\varepsilon}{1+\varepsilon'}\ ,$$

under the hypothesis that $\mathcal{P}r\left(G\in\mathcal{H}\right) \gg \mathcal{P}r\left(G\notin\mathcal{H}\right)$. The assumption holds for most of natural images as the flat areas correspond to the homogeneous areas and the non flat ones correspond to the outlines. Moreover, $\mathcal{P}r\left(G\in R\,|\,G\in\mathcal{H}\right)$ and $\mathcal{P}r\left(G\notin R\,|\,G\in\mathcal{H}\right)$ are constant for a given image and do not depend on $\mathcal{E}$. We conclude that ΔR_M is close to zero. The same reasoning can be made for ΔS_M and ΔU_M. Due to counter-measures introduced by MBPIS, the RS coefficients appear to be quasi-constant under classical RS steganalysis. The proposed method does not take flat areas into account, so it is equivalent to measure the relative variations $\Delta^{'} R_M$ and $\Delta^{'} S_M$ such as

$$\Delta^{'} R_M = \frac{\mathcal{P}r\left(\mathcal{E}(G)\in R\,|\,G\notin\mathcal{H}\right)}{\mathcal{P}r\left(G\in R\,|\,G\notin\mathcal{H}\right)}\ , \tag{21}$$

$$\Delta^{'} S_M = \frac{\mathcal{P}r\left(\mathcal{E}(G)\in S\,|\,G\notin\mathcal{H}\right)}{\mathcal{P}r\left(G\in S\,|\,G\notin\mathcal{H}\right)}\ , \tag{22}$$

which are not negligible and then, which are easier than the classical RS coefficients variations to detect.

4.2 Observing Features

In the remainder of the paper, we fixed the parameters of MBPIS. We tested our steganalysis with non-flat areas of size 2×2, the highest bit plane for embedding $i_{max} = 4$ and the threshold $t = 0$. To illustrate the features we pointed out in relation (18), we randomly embed the image of Fig. 2, and letting the embedding rate vary, we have plotted the local its RS-diagram.

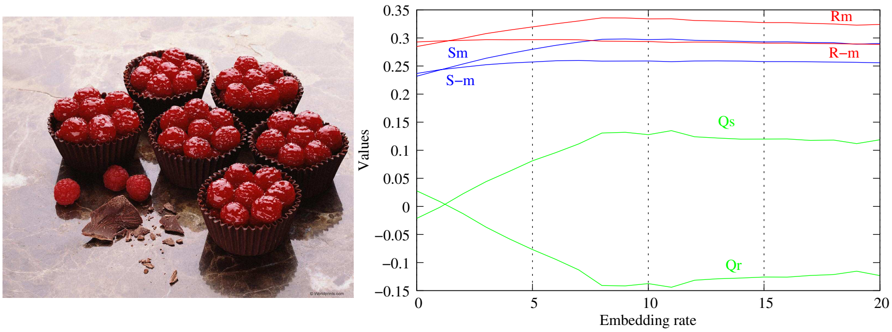


Fig. 2. A tested image and its local RS-diagram

4.3 Designing a Specific Detector

We use the same design for our detector as the one we presented in [9,10], as it is very simple and efficient. 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, but the following method can be easily adapted with learning sets of different cardinals.

First, for each set, we compute $\mathcal{V}_c = \{\mathcal{V}(I) | I \in \mathcal{C}\}$ as defined in relation (18), and $\mathcal{V}_s = \{\mathcal{V}(I) | I \in \mathcal{S}\}$ which are subsets of $\mathbb{R}^6$. 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, V_c and V_s. Finally, we compute the intraclass and interclass variance matrices W and B defined under our hypothesis by

$$B = \frac{1}{2}(g_c - g_s)(g_c - g_s)^{'}, \tag{23}$$

$$W = \frac{1}{2}\left(V_c + V_s\right). \tag{24}$$

The variance matrix, V is given by $V = B + W$.

The Fisher discrimination analysis [11,12] consists in finding a projection axis which discriminates the best $\mathcal{V}_c$ and $\mathcal{V}_s$ and so $\mathcal{C}$ and $\mathcal{S}$. This axis, (g_c, g_s), is defined by the vector

$$u = W^{-1}(g_c - g_s), \tag{25}$$

where $M = W^{-1}$ can be regarded as a metric. Actually, a new image, I represented by the point p will be said to belong to $\mathcal{C}$, 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 belongs $\mathcal{C}$ if and only if

$$p.u = pM(g_c - g_s) > T, \tag{26}$$

where T is the detection threshold. Another metric can also be considered, setting $M = V^{-1}$.

During the training step, we randomly chose 500 uncompressed images mainly from *Worldprint.com* database and 500 another ones randomly embedded with MBPIS and embedding rates from 3 to 25%. Experimentally, the average capacity is about 40%, but if the random message can not be embedded, we throw the image out and randomly choose an another one. The training set is composed of images of different sizes (from some Kilo-bytes to some Mega-bytes) and different color spaces (color and grayscale). Then, we calibrate our linear Fisher discriminant to obtain the best detection rates for the training set. We obtain the following discriminant vector and the associated barycenter g.

$$u = \begin{pmatrix} -8.327397E+01 \\ -6.080035E+02 \\ +1.224144E+02 \\ +4.573029E+02 \\ -5.263161E+01 \\ +2.452215E+01 \end{pmatrix} \quad g = \begin{pmatrix} \text{+2.910226E-01} \\ \text{+2.525373E-01} \\ \text{+2.976101E-01} \\ \text{+2.450667E-01} \\ \text{-2.384120E-02} \\ \text{+1.886609E-02} \end{pmatrix} . \tag{27}$$

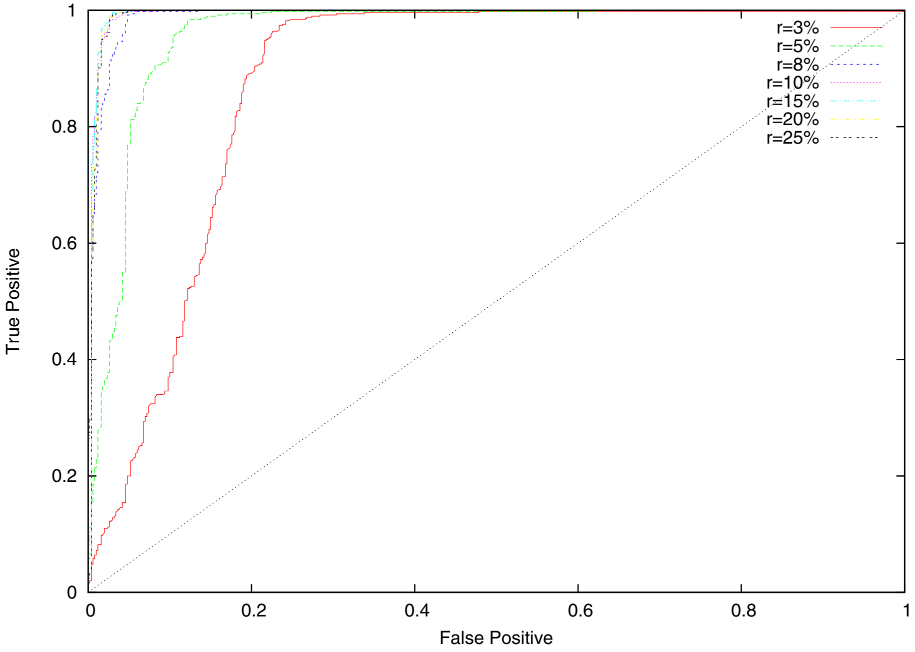


Fig. 3. ROC curves

4.4 Wild Detection Step

We randomly chose 500 uncompressed images and 500 another ones randomly embedded with MBPIS and embedding rates from 3 to 25%. Then, we submitted them to our classifier and obtained the detection rates according to Fig. 3.

It turns out that we are able to detect the use of MBPIS with a probability of detection higher than 62% in the worst case, *i.e.* with an embedding rate higher or equal to 1% and a set only composed of cover images. The obtained detection rates are summarized in Tab. 1.

Table 1. Detection rates

Embedding rate	False Positive	False Negative	Detection
0.03	0.1	0.622	0.639
0.03	0.15	0.356	0.747
0.05	0.048	0.232	0.86
0.05	0.1	0.066	0.917
0.08	0.01	0.27856	0.85586
0.08	0.05	0.00802	0.97097
0.1	0.01	0.14629	0.92192
0.1	0.02	0.03808	0.97097

5 Conclusion

We have presented a technique to efficiently detect Multi Bit Plane Image Steganography. Although this algorithm is secure against RS analysis, we adapted this classical analysis into an efficient local analysis scheme. We have taken advantage of the counter-measures introduced by the authors to protect MBPIS against RS analysis. Unfortunately, we are not able to estimate the embedding rate as in the classical RS analysis. Actually, the capacity of an uncompressed image according to MBPIS is message dependent as some of the flat areas of the bit plane B_i become flat after embedding in bit planes B_j, $j > i$. So, it is impossible to compute the coefficients needed for the quadratic interpolation and then, the length estimation. One main conclusion we can draw regarding the presented analysis is the following one. We suggest that making steganographic algorithm robust against a steganalysis by avoiding to embed into parts of the image which are significant for the considered analysis is absolutely not secure. In the same way, one strategy which can be performed by the analyzer is to focus only on areas effectively used by the steganographic algorithm and then adapt classical analysis without taking into account the avoided parts.

On one hand, embedding in multi bit planes preserves statistics of the LSBs and greatly improves the capacity of a steganography scheme, but on the other hand, even if statistical deviations are spread all over the bit planes, some of these bit plane statistics are less robust to embedding. Nevertheless, hiding data into non-smooth areas seems to be the best way to do. In this sense, JPEG steganography appears to be more adapted to easily separate informative and non-informative areas as DCT coefficients separate high frequencies and low frequencies in the frequency domain.

In future work we will try to improve our method in order to estimate the message length without the quadratic interpolation. We also plan to transform other classical steganalysis, as Pairs Analysis for instance, into local analysis in the same way.

Acknowledgment

We want to warmly thank Andreas Enge for his very useful advices and comments to improve the quality of the paper. He always took the time to provide us his help even during his holidays.

References

1. Nguyen, B., Yoon, S., Lee, H.K.: Multi bit plane image steganography. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 61–70. Springer, Heidelberg (2006)
2. Agaian, S., Rodriguez, B., Perez, J.: Stego sensitivity measure and multibit plane based steganography using different color models. In: Delp, E., Wong, P. (eds.) Proc.Security, Steganography, and Watermarking of Multimedia Contents VIII, February 2006, vol. 6072, pp. 279–290 (2006)

3. Fridrich, J., Goljan, M., Du, R.: Reliable detection of LSB steganography in grayscale and color images. In: Proc. ACM Workshop on Multimedia and Security, Ottawa, Canada, October 2001, pp. 27–30 (2001)
4. Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 61–76. Springer, Heidelberg (2000)
5. Dumitrescu, S., Wu, X., Wang, Z.: Detection of LSB steganography via sample pair analysis. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 355–372. Springer, Heidelberg (2003)
6. Lu, P., Luo, X., Tang, Q., Shen, L.: An improved sample pairs method for detection of LSB embedding. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 116–127. Springer, Heidelberg (2004)
7. Lyu, S., Farid, H.: Steganalysis using higher-order image statistics. IEEE Transactions on Information Forensics and Security 1 (2006)
8. Ker, A.: Improved detection of LSB steganography in grayscale images. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 97–115. Springer, Heidelberg (2004)
9. Barbier, J., Filiol, É., Mayoura, K.: Universal detection of JPEG steganography. Journal of Multimedia 2(2), 1–9 (2007)
10. Barbier, J., Filiol, É., Mayoura, K.: Universal JPEG steganalysis in the compressed frequency domain. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 253–267. Springer, Heidelberg (2006)
11. Saporta, G.: Probabilité, Analyse des Données et Statistiques. Technip (in french) (1990)
12. Raudys, S.: Statistical and Neural Classifiers: An Integrated Approach to Design. Advances in Pattern Recognition. Springer, Heidelberg (2001)
13. Fridrich, J., Goljan, M., Du, R.: Detecting LSB steganography in color and grayscale images. IEEE MultiMedia 8(4), 22–28 (2001)
14. Westfeld, A.: F5-a steganographic algorithm. In: Moskowitz, I. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)
15. Provos, N.: Defending against statistical steganalysis. In: 10th USENIX Security Symposium, Washington, DC, USA (2001)

Steganalysis of Enhanced BPCS Steganography Using the Hilbert-Huang Transform Based Sequential Analysis

Shunquan Tan[1,2], Jiwu Huang[2], and Yun Q. Shi[3]

[1] College of Information Engineering
Shenzhen University, Shenzhen, China, 518060
[2] School of Information Science and Technology
Sun Yat-sen University, Guangzhou, China, 510275
[3] New Jersey Institute of Technology, Newark, NJ, USA

Abstract. In this paper, we present a steganalytic method to attack the enhanced BPCS steganography (for short EBPCS) proposed by Niimi et al. The key element of the method is analysis of the bitplane-block complexity difference sequences between stego images and cover images based on Hilbert-Huang transform. The Hilbert transform based characteristic vectors are constructed via empirical mode decomposition of the sequences and the support vector machine classifier is used for classification. Experimental results have demonstrated effectiveness of the proposed steganalytic method in spatial domain images and JPEG2000 format images. According to our best knowledge, this method is the first successful attack of EBPCS and can be effectively used in spatial domain and frequency domain (especially DWT/JPEG2000) images.

1 Introduction

Steganalysis is the art of detecting hidden data in cover mediums. In this paper, we primarily deal with the steganalysis of enhanced BPCS steganography (EBPCS) [1]. EBPCS steganography is derived from traditional BPCS (Bit-Plane Complexity Segmentation steganography) [2], which is an important branch of bitplane based data hiding schemes. BPCS uses image segmentation based on a measure of complexity. The complexity is defined over a local region (referred as a block) within each bitplane. Such blocks can be classified as "informative" or "noise-like". By using the complexity measure, BPCS steganography embeds secret data into noise-like blocks considering that human eyes are insensitive to the alternation in those blocks.

The most remarkable character of BPCS steganography is its adaptability. Now BPCS steganography has demonstrated to be effective in embedding data into many classes of cover mediums including images and videos in spatial domain and frequency domain. Niimi et al. applied BPCS steganography to palette-based images by embedding secret information into the luminance channel of a

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 112–126, 2008.

palette-based image [3]. Ouellette et al. used a self-organizing neural network to re-order the index table of a indexed color image, so that similar colors in the index table are near each other with respect to their index values. Using this technique, they put forward a BPCS steganography used in indexed color images [4]. Spaulding et al. proposed a steganography method based on an embedded zerotree wavelet compression scheme and bit-plane complexity segmentation steganography. This is a BPCS steganography used in lossy compressed images [5]. Silvia et al. presented a steganographic algorithm based on BPCS which is robust against noisy channels [6]. Noda et al put forward a new steganographic method based on JPEG2000 lossy compression scheme and BPCS steganography. This is the first steganographic method used in JPEG2000 image format [7]. Furthermore, Noda et al. presented a BPCS based steganographic method using lossy compressed video which provides a natural way to convey a large amount of secret data via video stream [8].

Several steganalytic methods are proposed to analyze BPCS steganographies. In ICIP'04, Niimi et al proposed an attack method to BPCS [1]. This is the first and the most effective steganalytic method against BPCS steganography. By analysing the shape of the complexity histogram in an image, they pointed out the essential defect of traditional BPCS. In paper [9], Zhang et al. improved Niimi's method. They use statistic analysis technology to detect the discontinuities in the complexity histogram introduced by BPCS steganography. The steganalytic technique proposed in their paper is effective in attacking BPCS steganography in spatial domain and transform domain. Yu et al. put forward another steganalytic method against BPCS [10]. They claim that their method could detect the existence of secret message not only in spatial domain, but also in transform domain. But they didn't provide experimental results about the performance of their method when attacking BPCS in wavelet domain in their paper.

Enhanced BPCS steganography, for short EBPCS in this paper is proposed by Niimi et al [1] to act as a countermeasure of the essential defect of traditional BPCS. According to our best knowledge, there is still no report on steganalysis against EBPCS. In this paper, we propose a steganalytic method which can break EBPCS by utilizing Hilbert-Huang transform (HHT) based sequential analysis [11]. HHT is a new method for analyzing nonlinear and non-stationary sequence. Using support vector machine (SVM) [12] as classifier, our method achieves good performance. The proposed method makes the first successful attack on EBPCS steganography not only in spatial domain, but also in frequency domain (including JPEG2000 image domain). Our work open a new stage of the competition between steganography and steganalysis in BPCS steganography family.

In the next section, we give a brief review of EBPCS steganography. In Sect. 3, the proposed steganalytic method is described in detail. The experimental results and related discussions are shown in Sect. 4. The paper is concluded in Sect. 5.

2 Enhanced BPCS Steganography

2.1 Complexity Measure

The complexity measure BPCS uses is based on the border length of a block's binary pattern. In a binary pattern, the total length of the 0-1 border is equal to the summation of the number of 0-1 changes along the rows and columns inside the binary pattern. We assume that a block is a square of size $m \times m$. The region complexity measure is defined by the following equality:

$$\alpha = \frac{k}{2 \times m \times (m-1)} \tag{1}$$

where k is the total length of the 0-1 border in a bit-plane region.

2.2 BPCS and EBPCS

Define a 8×8 bit-plane block with low complexity as an "informative" block, a bit-plane block with relatively high complexity as a "noise-like" block. Split the secret message into a series of blocks each having 8 bytes of data. These blocks are regarded as 8×8 image patterns. The BPCS algorithm embeds these secret blocks into a cover image using the following steps:

1. Segment each bit-plane of a cover image into "informative" and "noise-like" blocks by using a threshold value (α). In [2] the author suggests to use $\alpha = 0.3$ for spatial domain images.
2. Group the bytes of the secret message into a series of secret blocks.
3. If a block is less complex than the threshold α, conjugate it to make it a more complex block. The conjugated block will be more complex than α according to the derivation in [2].
4. embed each secret block into the "noise-like" bit-plane blocks. The traditional BPCS steganography simply replaces all the "noise-like" blocks with a series of secret blocks.

The defect of BPCS is that all bits in a noise-like block is used for embedding secret data. In most cases the complexity of a stego block is different from that of the original noise-like one. This causes a change in the shape of the complexity histogram. When embedding secret data, blocks that are above the threshold are changed to obey a normal distribution and a discontinuity in the complexity histogram around the threshold becomes noticeable.

EBPCS presents a countermeasure to eliminate this defect. It uses only half of the bits in a block for secret data, which are termed PSD (2-value Pixels for Secret Data). The remaining half of the bits are used for adjusting complexity and are termed PAC (2-value Pixels for Adjusting Complexity). the locations of the PSD and PAC correspond to a checkerboard pattern. Define noise-like patterns as $0.5 - \delta \leq \alpha(P) \leq 0.5 + \delta$, where P is a bit-plane block and δ is a constant coefficient. Let P^i, $(i = 1, 2, \cdots, N)$ be the noise like blocks having the size of $m \times m$. The original complexity histogram is denoted by $h_{ORG}(c)$. The complexity histogram after embedding is denoted by $h_{EMB}(c)$ with initial value "0".

The EBPCS algorithm is given for each noise-like region as follow [1]:

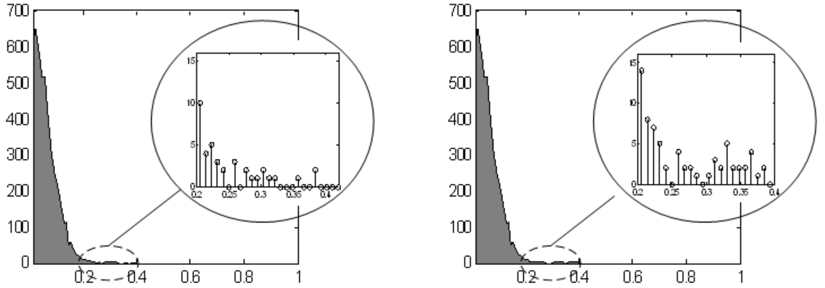
1. Initialization: $C_{org} = \alpha(P^i)$, $e = 0$.
2. Let k_c and K_s be $C_{org} + e$ and $C_{org} - e$. Select k_c as the target complexity if $h_{ORG}(k_c) > h_{EMB}(k_c)$. Otherwise select k_s if $h_{ORG}(k_s) > h_{EMB}(k_s)$. If k_c and k_s do not satisfy the conditions, the value of e is changed by:
$$e \leftarrow e + \frac{1}{2 \times m \times (m-1)} \tag{2}$$
and the above conditions are re-checked. The final target complexity is denoted by C_t.
3. Embed secret data into the PSD of P^i using the same method as in BPCS. In the PAC of P^i, we denote the set of bits having the property that the complexity of P^i becomes larger by reversing its value as B^+, and the set of bits having the inverse property as B^-. Draw out a bit from B^+ or B^- and reverse its value in turn, until $\alpha(P^i)$ is equal to C_t or B^+/B^- is empty.
4. Change the complexity histogram after embedding by the following equation:
$$h_{EMB}(\alpha(P^i)) \leftarrow h_{EMB}(\alpha(P^i)) + 1 \tag{3}$$

3 Proposed Steganalytic Method Using the HHT Based Sequential Analysis

3.1 Analysis of Correlation between Adjacent Bitplane Blocks

By and large, EBPCS do keep the complexity histogram of the object image, but we can still find some clues in the statistical characteristics of bit-plane block complexity in nature images to attack EBPCS. It is noted that for nature images, there exists strong correlation between adjacent blocks in a bit plane. EBPCS embedding impairs the correlation and introduces tail wave effect in the complexity difference histogram of a stego image, which is notable when embedding rate is high. Fig. 1 gives a demonstration. It shows the complexity difference histograms of LSB plane of a natural image before and after embedding (with embedding rate= 50%). The shape of the histograms is close to a generic Gaussian histogram, and the differences between Fig. 1(a) and Fig. 1(b) in interval [0, 0.2], which concentrates most of the energy in the histogram, are inconspicuous. Whereas, it can be observed that when we magnify the histograms in interval [0.2, 0.4], the energy in this interval after embedding, despite its very low value, is obviously higher than that before embedding.

The tail wave effect is due to the fact that EBPCS can not adjust each stego-block's complexity to the original one exactly. Denote the complexity of P^i after PSD embedding by C_0, the object complexity by C_t and the complexity in adjustment process by $C_t^{'}$. C_0 obeys a normal distribution with mean $\mu = 0.5$ [1].



(a) From the image before embedding (b) From the image after embedding

Fig. 1. Bit-plane block complexity difference histogram of a nature image before and after embedding

In order to eliminate the discontinuity in the complexity histogram around the threshold, the complexity of P^i moves from C_0 near the mean value μ to C_t whose value is more close to the threshold than μ in most cases. EBPCS reverses the values of the bits in B^+ or B^- to adjust the complexity. So the step size of the complexity adjustment after each reversal depends on the values of the horizontal and vertical neighbors of the reversed bit. Its value is set inside the interval $[\frac{1}{2\times m\times(m-1)}, \frac{2}{m\times(m-1)}]$. The adjustment process is divided into two parts: coarse adjustment and fine tuning. Coarse adjustment moves $C_t^{'}$ from C_0 to interval $[C_t - \frac{1}{m\times(m-1)}, C_t + \frac{1}{m\times(m+1)}]$. Fine tuning keeps reversing the bits in B^+ or B^-, until $C_t^{'}$ is equal to C_t or either one of B^+ and B^- is empty. Denote the number of the bits in B^+ and B^- by $|B^+|$ and $|B^-|$. $|B^+| \approx |B^-|$ when the complexity of P^i is near μ. It is noted that the approximately equal relationship between $|B^+|$ and $|B^-|$ is broken in coarse adjustment process. One of them is much less than the other in the beginning of the fine tuning process. Give the case that $C_t < C_0$. Since most of the bits in B^- are drawn out to reverse, $|B^-| \ll |B^+|$ when fine tuning process start. So B^- becomes empty first if the fine tuning process fails to make $C_t^{'}$ equal to C_t, which means that $C_t^{'}$ falls over to the right side of C_t and $C_t^{'} \in [C_t + \frac{1}{2\times m\times(m-1)}, C_t + \frac{1}{m\times(m+1)}]$. Similarly, we can deduce that when $C_t > C_0$, $C_t^{'}$ falls over to the left side of C_t and $C_t^{'} \in [C_t - \frac{1}{2\times m\times(m-1)}, C_t - \frac{1}{m\times(m+1)}]$ if $C_t^{'} \neq C_t$.

The deductions above lead to an assertion:

Assert 1. *In a bit-plane complexity histogram, μ=0.5, Let P denote probability. If $\mu < c_1 < c_2$ or $\mu > c_1 > c_2$, then:*

$$P(h_{EMB}(c_1) = h_{ORG}(c_1)) > P(h_{EMB}(c_2) = h_{ORG}(c_2)) \tag{4}$$

The assertion means that the complexity histogram bins close to μ is filled up to the original height earlier than those far from μ. Since the complexity with a filled bin can not be appointed to target complexity, EBPCS has to find

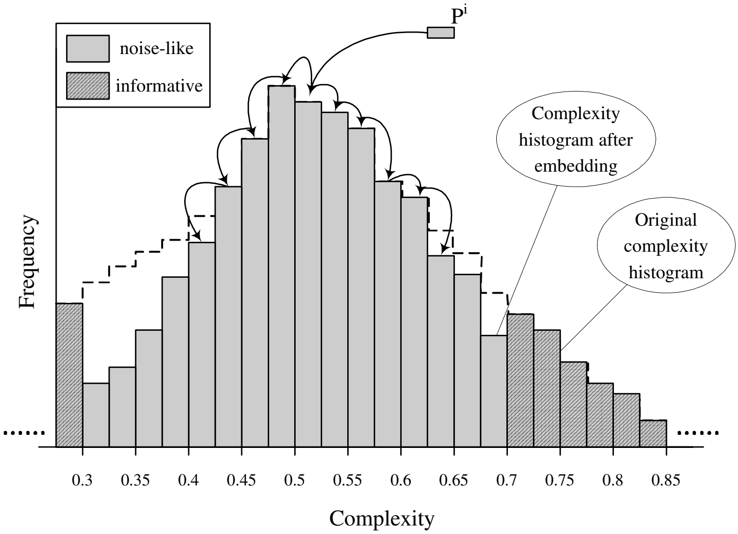


Fig. 2. Diagram of the process of setting the target complexity

another bin with space. The value of the result complexity is always far from the original one, especially in the late period of EBPCS embedding process. Fig. 2 shows an example. In EBPCS embedding process, the bins inside the interval $[0.425, 0.625]$ have been filled. Thus when processing a noise-like block P^i with complexity 0.5, the bins inside that interval are ignored. $k_c = 0.65$ and $k_s = 0.425$ beside the interval $[0.425, 0.625]$ are the candidates we can choose. Since $C_{org} - k_s = 0.1 < k_c - C_{org} = 0.125$, we select $C_t = k_s = 0.425$ as the target complexity, whose corresponding bin is far from the original one and introduce tail wave effect.

3.2 Steganalysis Based on HHT of the Difference Sequence of Bitplane-Block Complexity

It's hard to detect the low energy difference introduced by EBPCS only according to the shape of complexity histogram or complexity difference histogram especially when embedding rate is low. But we know that the long-range jumps of complexity mainly occur in the later period of EBPCS embedding process. So abnormal complexity differences are assembled in a local EBPCS scanning area. A sliding window with fixed size can be used to move along the EBPCS scanning area. Any abnormal complexity difference in the window will be detected and used as the evidence of EBPCS embedding.

To find out the abnormality in the bitplane-block complexity difference sequence of a given window, we employ HHT which is already successfully used in JPEG2000 steganalysis [11]. The basis of HHT is derived from a sequence itself, hence the HHT-based analysis has an excellent track record when applied to non-stationary sequences. HHT consists of empirical mode decomposition (EMD) and Hilbert spectral analysis. EMD reduces the data into a collection of intrinsic mode functions (IMF) defined as any function satisfying the following conditions:

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 EMD method is given as follow:

Algorithm 2. *Assume $X(t)$ is one discrete random series.*

```
1. Initialize: $r_0(t) = X(t)$ (the residual) and $i = 1$ (index number of IMF).
2. Extract the $i^{th}$ IMF:
   (a) Initialize: $h_0(t) = r_{i-1}(t), j = 1$.
   (b) Extract regional extrema(minima/maxima) of $h_{j-1}(t)$.
   (c) Compute upper envelope and lower envelope functions $u_{j-1}(t)$ and
       $l_{j-1}(t)$ by interpolating respectively local minima and local maxima of
       $h_{j-1}(t)$.
   (d) Compute $m_{j-1}(t) = \frac{u_{j-1}(t)+l_{j-1}(t)}{2}$ as the estimation of the local mean
       of $h_{j-1}(t)$.
   (e) Update $h_j(t) = h_{j-1}(t) - m_{j-1}(t)$.
   (f) If stopping criterion is satisfied, $h_j(t)$ is already an IMF. So set
       $imf_i(t) = h_j(t)$. Otherwise set $j = j + 1$ and go to step (2b).
3. Update residual $r_i(t) = r_{i-1}(t) - imf_i(t)$.
4. Repeat step 1-3 with $i = i + 1$ until the number of extrema in $r_i(t)$ is less
   than two.
```

After the EMD process, the random sequences is represented as the sum of finite IMFs. We denote them as $\{imf_i(t), i \in N\}$. We can compute Hilbert transform of $imf_i(t)$ by:

$$imf_i^H(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{imf_i(u)}{t-u} \mathrm{d}u \tag{5}$$

where P indicates the Cauchy principal value. The analytic signal of $imf_i(t)$ can be represented as a complex signal:

$$a(t)e^{j\theta(t)} = imf_i(t) + j \cdot imf_i^H(t) \tag{6}$$

where

$$a(t) = [(imf_i(t))^2 + (imf_i^H(t))^2]^{\frac{1}{2}} \tag{7}$$

$$\theta(t) = \arctan(\frac{imf_i^H(t)}{imf_i(t)}) \tag{8}$$

are the amplitude and the phase of this analytic signal. With Eq. (6), the instantaneous frequency of $imf_i(t)$ is defined as

$$\omega(t) = \frac{d\theta}{dt} \tag{9}$$

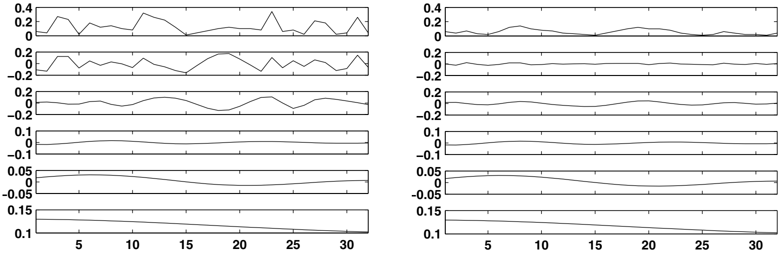
The time-frequency distribution of amplitude is designated as Hilbert amplitude spectrum. It gives a time-frequency-amplitude distribution of an IMF.

Different from Fourier transform and wavelet transform, the basis of EMD is derived from the sequence itself. Hence the analysis may be applied to non-stationary data. It can extract frequency components in each location from high frequency to low frequency, so the high frequency components which contaminate a low frequency signal and bring its nonlinear distortion are extracted before the low frequency component is extracted. Consequently, the nonlinear distortion of the low frequency component is relieved.

Let $X_{\triangle}^{A}(t)$ denotes a bitplane-block complexity difference sequence in a fixed size window with abnormal long-range jumps and $imf_i^A(t)$ is its i^{th} IMF. Similarly, $X_{\triangle}^{N}(t)$ denotes a normal bitplane-block complexity difference sequence in a window and $imf_i^N(t)$ is its i^{th} IMF. From Sect. 3.1, we know that the long-range jumps of complexity introduce abnormal complexity difference in the window. Thus the energy in high frequency distilled from $X_{\triangle}^{A}(t)$ via EMD should be higher than that from $X_{\triangle}^{N}(t)$. Fig.3 gives a demonstration. Fig.3(a), from top to bottom is the $X_{\triangle}^{A}(t)$ in a selected window area from a stego image produced by embedding secret data in 15% of "noise-like" blocks of a randomly selected image, its four levels of IMFs and a residual produced by EMD. Fig.3(b), from top to bottom is the $X_{\triangle}^{N}(t)$ from the same window area before embedding, its IMFs and a residual. It can be observed that $imf_1^A(t)$ and $imf_2^A(t)$ have higher variation amplitudes compared with $imf_1^N(t)$ and $imf_2^N(t)$. Whereas, the differences of the IMFs in low frequency and the residuals of two sequences are inconspicuous, which shows that the energy differences of $X_{\triangle}^{A}(t)$ and $X_{\triangle}^{N}(t)$ mainly exist in the highest two IMFs. This conclusion is also validated by us in many other experiments.

3.3 Steganalysis Based on EMD and SVM

Given an test image, EMD is applied to the bitplane-block complexity difference sequence of each window and Hilbert transform is applied to the first two IMFs of the results. We use the mean of the sampling values of the amplitude of one IMF's analytic signal as the statistical feature of the IMF and call it the IMF's characteristic energy. We can use the vector comprising the characteristic energies of $imf_1(t)$ and $imf_2(t)$ of each complexity difference sequence as a characteristic vector, which constitutes the entry of the SVM classifier. Giving a training set of characteristic vectors built from bitplane-block complexity difference sequences with and without abnormal long-range jumps , the



(a) From a sequence with abnormal long-range jumps

(b) From a normal sequence

Fig. 3. IMFs and residual of $X_{\triangle}^{A}(t)$ and $X_{\triangle}^{A}(t)$

classifier is designed based on them using two-class linear separable SVM [12]. The construction of SVM classifier is equal to solve the following primal problem:

$$\begin{array}{c} \min_{w,b,\xi} \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ \text{subject to } y_i(w^T\phi(x_i)+b) \geq 1-\xi_i, \\ \xi_i \geq 0, i=1,\ldots,l. \end{array} \tag{10}$$

Its dual is

$$\begin{array}{c} \min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ 0 \leq \alpha_i \leq C, i=1,\ldots,l, \\ \text{subject to } y^T\alpha = 0. \end{array} \tag{11}$$

where e is the vector of all ones, C is the upper bound and $C > 0$, Q is a positive semidefinite matrix of $l \times l$, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ is the kernel. Here training vectors x_i are mapped into a higher dimensional space by the function ϕ.

After the training process applied to the SVM classifier, We can judge that a given image contains secret data embedded by EBPCS if the trained SVM classifier picks out a sequence with abnormal long-range jumps in the image. The steganalysis algorithm for JPEG2000 format image is given as follow:

Algorithm 3. *Assume I is a JPEG2000 format image for detection.*

1. *Extract the wavelet coefficient tree W from I. Build a bitplane-block complexity difference sequence for each subband in W and use a fixed size window to split it. The result set of the sub-sequences is denoted as $\mathbb{X}$.*
2. *Given a sub-sequence $X_{\triangle}(t)$ in $\mathbb{X}$, Apply EMD to it and get the first two IMFs: $imf_1(t)$ and $imf_2(t)$.*
3. *Calculate characteristic vector $\{P_{\sigma_1}, P_{\sigma_2}\}$ of $imf_1(t)$ and $imf_2(t)$ according to Sect. 3.3*
4. *Calculate the output of decision function:*

$$Output = sgn(\sum_{i=1}^{l} y_i\alpha_i K(x_i, \{P_{\sigma_1}, P_{\sigma_2}\}) + b). \tag{12}$$

If Output is $+1$*, then I contains secret message embedded by EBPCS steganography.*

4 Experimental Results and Analysis

4.1 Samples for Train and Test

To evaluate the proposed steganalytic method in spatial domain images and JPEG2000 format images, the CorelDraw image database[1] is selected as the experimental image set. This image database contains all kinds of images: natural scene, architecture, animals, indoor, outdoor, etc. Fig. 4 gives some sample images. The size of each image in the database is set to 768×512 or 512×768. The image set contains 1096 spatial domain images in BMP format. JPEG2000 format images are generated using JASPER[2]. The JASPER encoder parameters are set as follow: number of resolution layers at 2; number of quality layers at 5; size of code block at 32×32; compression ratios associate with quality layers at 2:1, 4:1, 8:1, 16:1, 32:1 . In EBPCS embedding process, The first two LSB planes of spatial domain images are used for embedding secret messages. For JPEG2000 images, only the first two LSB planes in each subband are used. The secret message is simulated by a binary bit stream obeying uniform distribution. The embedding rates of EBPCS for spatial domain images and JPEG2000 images are set to 50%, 30%, 15%. The threshold of spatial domain EBPCS is set to $\alpha_0 = 0.3$. The threshold of JPEG2000 EBPCS is set to $\alpha_0 = 0.475$. We randomly choose 3/4 of the images for training purpose. The remaining 1/4 of the images are used for testing purpose.

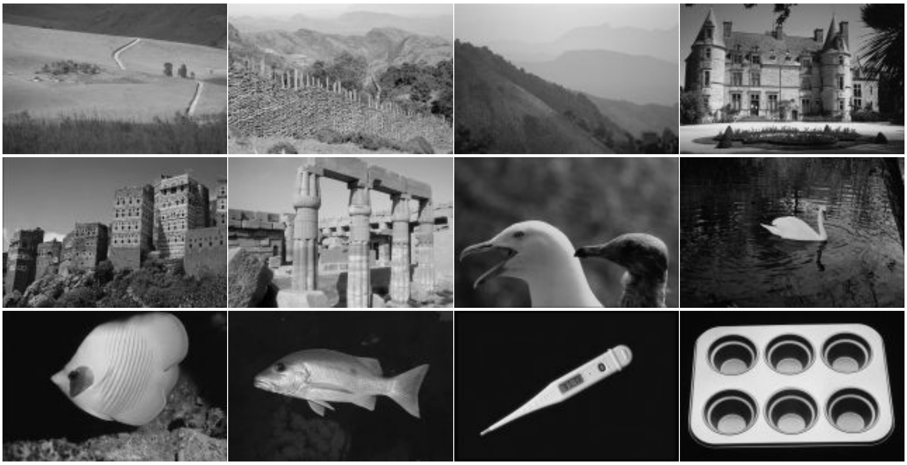


Fig. 4. Sample images from CorelDraw image database

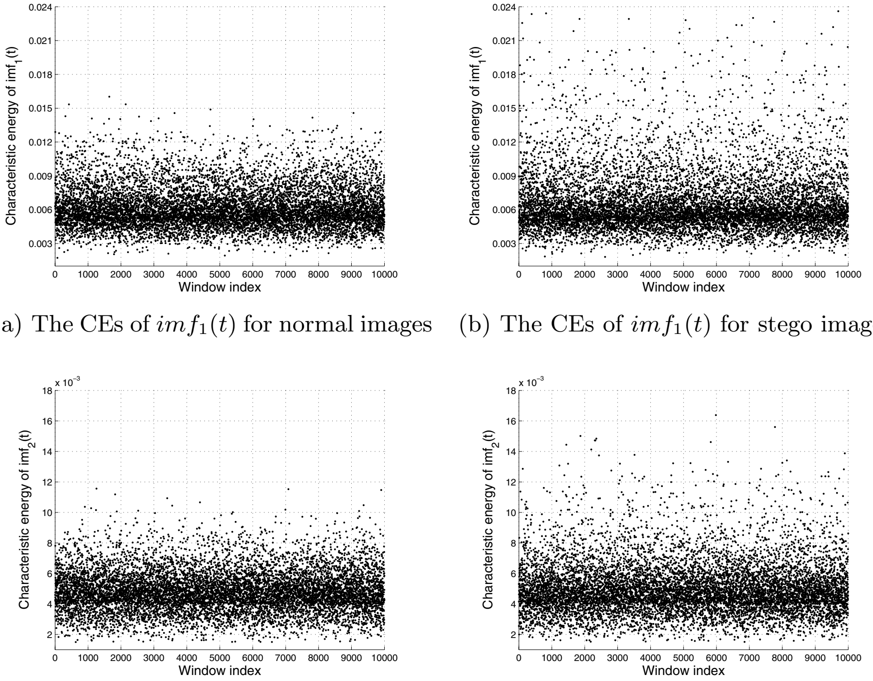
[1] http://www.corel.com

[2] http://www.ece.uvic.ca/˜mdadams/jasper/

4.2 Analysis of Experimental Results

Figure 5 shows the characteristic energies of $imf_1(t)$ and $imf_2(t)$ for spatial domain images. For clarity, we only show the characteristic energies of the first ten thousand fixed size window areas in these images. They are ordered by their positions in the EBPCS scanning sequence. Figure 5(a) and 5(b) show the characteristic energies of $imf_1(t)$ and $imf_2(t)$ for normal images, respectively. Similarly, fig. 5(c) and 5(d) show those of $imf_1(t)$ and $imf_2(t)$ for stego images. From Fig. 5 we can observe that the derivations of the distribution of the characteristic energies of $imf_1(t)$ and $imf_2(t)$ for stego spatial domain images are higher than the ones for normal images. As a result, there are much more peculiar points with high energy in Fig. 5(c) and 5(d) than in Fig. 5(a) and 5(b). Though the number of the peculiar points in Fig. 5(c) and 5(d) is small compared with the massive normal points. They are enough for our steganalytic algorithm because just a single fixed size window area with peculiar characteristic energies can be used to certify that the host image is a stego image.

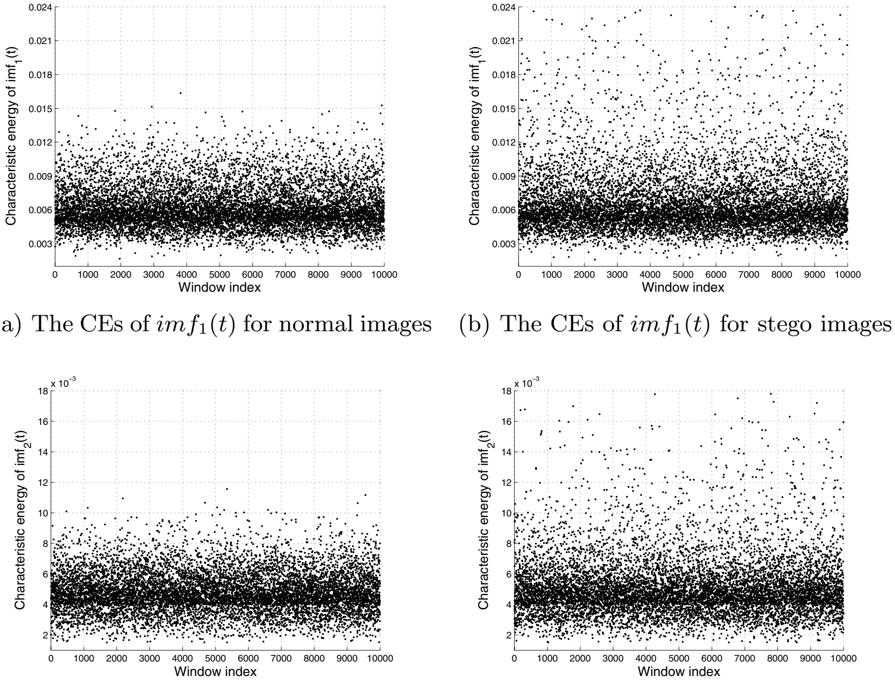
Figure 6 shows the characteristic energies of $imf_1(t)$ and $imf_2(t)$ of the first ten thousand fixed size window areas in JPEG2000 images. Similarly, they are



(a) The CEs of $imf_1(t)$ for normal images (b) The CEs of $imf_1(t)$ for stego images

(c) The CEs of $imf_2(t)$ for normal images (d) The CEs of $imf_2(t)$ for stego images

Fig. 5. The characteristic energies (for short CE) of $imf_1(t)$ and $imf_2(t)$ for spatial domain images



(a) The CEs of $imf_1(t)$ for normal images (b) The CEs of $imf_1(t)$ for stego images

(c) The CEs of $imf_2(t)$ for normal images (d) The CEs of $imf_2(t)$ for stego images

Fig. 6. The characteristic energies (for short CE) of $imf_1(t)$ and $imf_2(t)$ for JPEG2000 images

also ordered by their positions in the EBPCS scanning sequence. Figure 6(a) and 6(b) are for normal images. On the other side, fig. 6(c) and 6(d) are for stego images. Like what Fig. 5 shows, the derivations of the distribution of the characteristic energies of $imf_1(t)$ and $imf_2(t)$ for stego JPEG2000 images are also higher than the ones for normal JPEG2000 images. Furthermore, in Fig. 6(c) and 6(d) we can find out that the derivations for stego JPEG2000 images are higher than the corresponding ones for stego spatial domain images, which means that our steganalytic algorithm should get better performance when attacking stego JPEG2000 images. Our experimental results also verify this conclusion.

Figure 7 shows the ROC curve drawn from the testing result using trained SVM classifier for spatial domain images. It can be seen that the constructed SVM classifier can distinguish the stego spatial domain images from the normal spatial domain images with high accuracy. It shows a high true positive rate even when the corresponding false positive rate is relatively low for stego images when embedding rate is 30% or 50%. Furthermore, the proposed steganalytic method still can get a relatively high embedding rate when the embedding rate is as low as 15%. The graph demonstrate that the proposed steganalytic method has good reliability when attacking spatial domain EBPCS steganography.

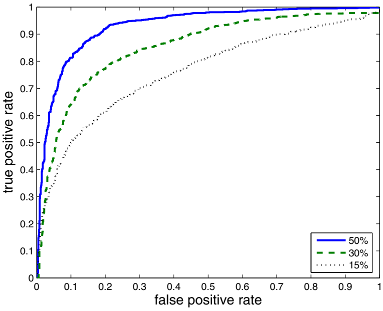


Fig. 7. ROC of the test result for spatial domain images

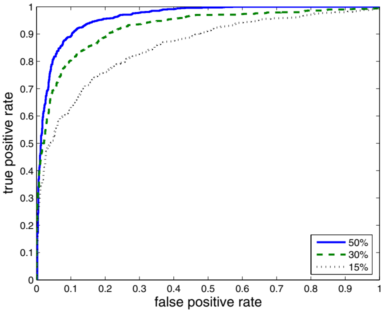


Fig. 8. ROC of the test result for JPEG2000 images

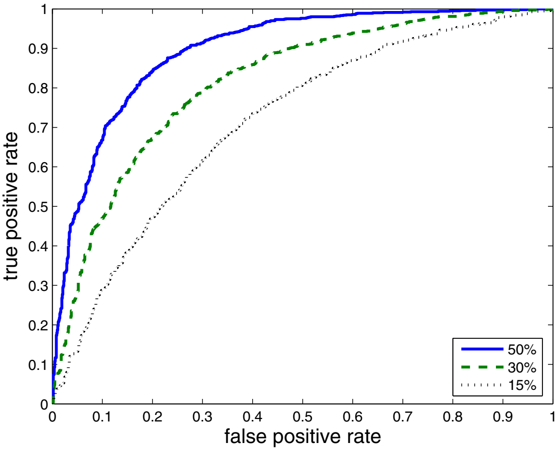


Fig. 9. ROC of the test result for mixture images

Figure 8 shows the ROC curve drawn from the testing result using trained SVM classifier for JPEG2000 images. Like the test result for spatial domain images, the constructed SVM classifier can distinguish the stego JPEG2000 images from the normal JPEG2000 images with high accuracy. The further analysis to Fig. 7 and 8 shows that the classifier achieves better performance with JPEG2000 images, which verify the conclusion we draw in the last paragraphs. This is because compared with the only eight bit planes in a spatial domain image, wavelet decomposition in JPEG2000 encoding process generates much more coefficient bit-planes. The more the bit-planes are, the more bit-plane blocks can be used to construct the SVM classification boundary, which is essential for the proposed steganalytic method.

In order to show the universality of the proposed steganalytic algorithm, we use a mixture of the characteristic vectors built from spatial domain images and

JPEG2000 images as training set and testing set for the SVM classifier. Figure 9 shows the ROC curve drawn from the classify result using trained SVM classifier for mixture testing set. We can see that the classifier has good reliability even when the format of a cover image is unknown, although the false positive rate arises compared with that of the classifier trained with known image format.

5 Conclusion

In this paper, we present a steganalytic method of EBPCS steganography based on the HHT based analysis of the bitplane-block complexity difference sequences in fixed size window areas. The key element of the method is the HHT-based analysis of the bitplane-block complexity difference sequences of stego images and cover images. The Hilbert transform based characteristic vectors are constructed via empirical mode decomposition of the sequences and the support vector machine classifier is used in classification. Experimental results have demonstrated effectiveness of the proposed steganalytic method in spatial domain images and JPEG2000 format images. The proposed method can be efficiently used in steganalysis of EBPCS in spatial domain and frequency domain images, and can be easily extended to other cover mediums.

References

1. Niimi, M., Ei, T., Noda, H., et al.: An attack to BPCS-steganography using complexity histogram and countermeasure. In: Proceedings - International Conference on Image Processing, ICIP, vol. 5, pp. 733–736 (2004)
2. Kawaguchi, E., Eason, R.O.: Principle and applications of BPCS-steganography. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 3528, pp. 464–473 (1999)
3. Niimi, M., Eason, R.O., Noda, H., Kawaguchi, E.: A method to apply BPCS-steganography to palette-based images using luminance quasi-preserving color quantization. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E85-A, 2141–2148 (2002)
4. Ouellette, R., Noda, H., Niimi, M., et al.: Topological ordered color table for BPCS-steganography using indexed color images. In: roceedings of SPIE - The International Society for Optical Engineering, vol. 3971, pp. 502–509 (2000)
5. Spaulding, J., Noda, H., Shirazi, M.N., et al.: BPCS steganography using EZW lossy compressed images. Pattern Recognition Letters 23(13), 1579–1587 (2002)
6. Torres-Maya, S., Nakano-Miyatake, M., Perez-Meana, H.: An image steganography systems based on bpcs and iwt. In: Proceedings of the 16th IEEE International Conference on Electronics, Communications and Computers, CONIELECOMP 2006, p. 51 (2006)
7. Noda, H., Spaulding, J., et al.: Bit-plane decomposition steganography combined with JPEG2000 compression. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 295–309. Springer, Heidelberg (2003)
8. Noda, H., Furuta, T., Niimi, M., Kawaguchi, E.: Application of BPCS steganography to wavelet compressed video. In: 2004 International Conference on Image Processing, ICIP 2004, pp. 2147–2150 (2004)

9. Zhang, X., Wang, S.: Statistical analysis against spatial BPCS steganography. 17, 1625–1629 (2005)
10. Yu, X., Tan, T., Wang, Y.: Reliable detection of BPCS-steganography in natural images. In: Third International Conference on Image and Graphics, pp. 333–336 (2004)
11. Tan, S., Jiwu Huang, Z.Y.: Steganalysis of JPEG2000 lazy-mode steganography using the hilbert-huang transform based sequential analysis. In: Proceedings - International Conference on Image Processing (ICIP 2006), Atlanta, October 8-11 (2006)
12. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/cjlin/libsvm`

Weaknesses of MB2

Christian Ullerich and Andreas Westfeld

Technische Universität Dresden, Institute for System Architecture,
D-01062 Dresden, Germany
{christian.ullerich,westfeld}@mail.inf.tu-dresden.de

Abstract. Model-based steganography is a promising approach for hidden communication in JPEG images with high steganographic capacity and competitive security. In this paper we propose an attack, which is based on coefficient types that can be derived from the blockiness adjustment of MB2. We derive 30 new features to be used in combination with existing blind feature sets leading to a remarkable reduction of the false positive rate (about 10:1) for very low embedding rates (0.02 bpc). We adapt Sallee's model-based approach for steganalysis where the Cauchy model itself is used to detect Cauchy model-based embedded messages. We apply a gradient aware blockiness measure for improved reliability in the detection of MB1. We evaluate our proposed methods based on a set of about 3000 images.

1 Introduction

LSB steganography is easily detected (cf., e. g., [1,2,3] for spatial images and [4,5,6,7] for JPEG images) because it equalises the frequencies of pairs of values that only differ in the least significant bit (LSB). The model-based approach for steganography tries to model the deterministic part of the carrier medium to enable steganographic application of the indeterministic part.

Sallee modelled the marginal distribution of the DCT coefficients in JPEG images by the generalised Cauchy distribution [8]. In contrast to LSB steganography, the pairs of values are not equalised with this model-based approach. Instead, the embedded message is adapted to the generalised Cauchy distribution of each AC DCT subband in the JPEG carrier file. This adaptation is implemented as arithmetic decoding. Arithmetic coding transforms unevenly distributed bitstreams into shorter, uniform ones. Conversely, the arithmetic decoding can take a uniformly distributed bitstream (the message to be embedded) to produce a bitstream that is adapted to given probabilities of 0 and 1 according to the present generalised Cauchy distribution. In case the chosen distribution fits to the JPEG file, the first order statistics is preserved after embedding the adapted bitstream into the LSBs of the coefficients. This procedure is known as MB1 today.

A first hint that the generalised Cauchy distribution does not completely match the histogram of DCT coefficients was given by Böhme and Westfeld [9]. They showed that there are non-conforming pairs that considerably deviate from

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 127–142, 2008.

the modelled distribution (outliers). After embedding with MB1, these outliers are — dependent on the embedding rate — scarcer or disappear completely. Although only first order statistics was considered, this attack achieves fairly reliable detection rates.

It is obvious that higher order statistics are more powerful. One weak property of MB1 is that block artefacts increase with growing size of the payload. MB2 was developed to overcome this weakness [10]. It embeds the message in the same way as MB1 does but offers only half the capacity of MB1 to the user. The other half of the non-zero DCT coefficients are reserved for blockiness reduction. Early assessment by Fridrich showed good security compared to MB1 [11]. In recent analysis, however, the tables have been turned [12,13].

This paper is organised as follows: In Sect. 2, we consider the chosen model of MB2 and consequently of MB1. As the aforementioned work of Böhme and Westfeld has shown, it is possible to launch an attack based on the model conformity of the low precision bins. To strengthen the assumption that the specific model used in the embedding scheme is incorrect for JPEG images, we apply the Cauchy model itself to detect model-based embedded messages with Sallee's model-based approach for steganalysis [10]. Sect. 3 presents an MB2 specific finding that the blockiness reduction used in MB2 increases the detection reliability of MB2 steganograms using the same distance measures. In Sect. 4, we use another blockiness measure to classify steganograms, which was proposed by Westfeld [14] to compensate for visible distortions and remove a watermark in the first BOWS contest [15]. Sect. 5 defines coefficient types derived from case discrimination of the blockiness adjustment used in MB2. These are applied in an MB2 specific strong attack. In Sect. 6 we propose 30 new features to be used with existing blind attacks and present our experimental results. The paper is concluded in Sect. 7.

2 Model-Based Steganalysis

Sallee proposed not only the use of models for steganography but also for steganalysis [10]. He demonstrated it by detecting JSteg steganograms.

The approach done is that two submodels are necessary. One represents the cover media and the other the changes caused by the embedding function. Those two submodels are connected by a parameter β, which identifies the distance of the current media to the two submodels. If for example this model is applied to a cover medium, the β value should express that the submodel of the cover explains the presented media best. If, on the other hand, the model is applied to a steganogram, than β should express that the submodel, which describes the changes of the embedding, explains the presented media best. A rather easy differentiation is possible if $\beta = 0$ represents a cover medium and $\beta > 0$ a steganogram.

Since the embedding with MB1 and MB2 causes different changes than JSteg one submodel needs to be changed. The submodel that represents the cover media can be used as it is, because all three algorithms embed in JPEG images.

Even though MB1 uses individual histograms for its embedding model the focus on the global histogram shall be sufficient for this analysis. The difference between JSteg and MB1 regarding the usage of high precision bins is that MB1 uses bin 1 for embedding while JSteg does not. The coefficient value zero is not used in either algorithm. In order to model the embedding process it is necessary to model the possible changes of the coefficient values. Arguing for a bin size *step* = 2 there is only one possible other value in which a coefficient value c can be changed into. This value is denoted with $\tilde{c}$ and can be calculated with $\tilde{c} = \mathrm{sign}(c)(|c| + 1 - 2((|c| + 1) \bmod 2)) = \mathrm{sign}(c) \left(4 \left\lfloor \frac{|c|+1}{2} \right\rfloor - (|c| + 1)\right)$.

The density function of the global histogram is approximated by a generalised Cauchy distribution with parameters σ and π. The cumulative density function (cdf) F of the used generalised Cauchy distribution is defined as

$$F(x|\sigma, \pi) = \frac{1}{2} \left(1 - \mathrm{sign}(x) \left(\left(1 + \left|\frac{x}{\sigma}\right|\right)^{1-\pi} - 1\right)\right) . \tag{1}$$

Using the cdf F, the probability P of a coefficient value c can be calculated as follows

$$P(c|\sigma, \pi) = F(c + 0.5|\sigma, \pi) - F(c - 0.5|\sigma, \pi) . \tag{2}$$

The combined model for the probability of a coefficient value is

$$\tilde{P}(c|\sigma, \pi, \beta) = \begin{cases} P(c|\sigma, \pi) & \text{if } c = 0, \\ (1 - \beta)P(c|\sigma, \pi) + \beta P(\tilde{c}|\sigma, \pi) & \text{else.} \end{cases} \tag{3}$$

The combined model consists of two submodels. One of them $(P(c|\sigma, \pi))$ models the probabilities of odd coefficient values greater or equal than the probabilities of the even ones. The other submodel $(P(\tilde{c}|\sigma, \pi))$ does it the other way round. It is to be expected that β is zero if the log likelihood is maximised with

$$\hat{\sigma}, \hat{\pi}, \hat{\beta} = \arg \min_{\sigma, \pi, \beta} \left[- \sum_{c} h_c \log \tilde{P}(c|\sigma, \pi, \beta)\right] \tag{4}$$

using the frequencies h_c of the corresponding coefficient values. Interestingly, if this is applied to cover images most of the resulting β are greater than zero (96 %), which means that the submodel of the cover media does not correctly approximate the global histogram and that $\beta = 0$ does not represent cover images.

This finding itself does not show the use for steganalysis. Fitting the model to MB1 steganograms with a high embedding rate results in a decreasing β — mostly still positive — and an increasing deviation. Fitting it to MB2 steganograms results also in a decreasing β but an even greater deviation.

Knowing that β changes with embedding, a simple measure can be used to classify steganograms. Let γ be the measure, then $\gamma = |\beta_{\mathrm{s}}(1) - \beta_{\mathrm{s}}(0)|$. $\beta_{\mathrm{s}}(0)$ denotes the resulting β if the model is fitted to the image that is to be classified. The β value after embedding anew (or for the first time) with maximal message

Table 1. Detection reliability of model based steganalysis with different JPEG quantisation. For the reliability the absolute value is decisive. We present signed values in the table where it might be interesting to see between which JPEG qualities the sign change occurs, i. e., where the model fits best.

	JPEG quality q						
	0.99	0.95	0.9	0.8	0.7	0.6	0.5
MB1							
$\beta_{\mathrm{s}}(0)$	0.430	−0.074	−0.295	−0.468	−0.572	−0.656	−0.706
BW attack	0.421	0.527	0.581	0.614	0.668	0.701	0.711
γ	0.501	0.389	0.414	0.498	0.539	0.558	0.603
LDA M_4	0.934	0.494	0.593	0.736	0.836	0.867	0.907
LDA M_6	0.936	0.562	0.698	0.856	0.912	0.925	0.945
QDA M_4	0.960	0.840	0.897	0.936	0.947	0.949	0.956
QDA M_6	0.982	0.908	0.958	0.990	0.988	0.981	0.984
MB2							
$\beta_{\mathrm{s}}(0)$	0.462	0.193	0.032	−0.178	−0.327	−0.462	−0.559
BW attack	0.394	0.341	0.380	0.387	0.439	0.508	0.548
γ	0.488	0.404	0.484	0.609	0.681	0.712	0.758
LDA M_4	0.935	0.665	0.390	0.206	0.432	0.584	0.707
LDA M_6	0.946	0.735	0.638	0.628	0.641	0.650	0.721
QDA M_4	0.964	0.907	0.895	0.938	0.950	0.951	0.957
QDA M_6	0.968	0.922	0.923	0.965	0.964	0.964	0.956

$M_4 = \{\beta_{\mathrm{s}}(0), \beta_{\mathrm{s}}(1)_{\mathrm{MB2}}, \beta_{\mathrm{cal}}(0), \beta_{\mathrm{cal}}(1)_{\mathrm{MB2}}\}$,
$M_6 = \{\beta_{\mathrm{s}}(0), \beta_{\mathrm{s}}(1)_{\mathrm{MB1}}, \beta_{\mathrm{s}}(1)_{\mathrm{MB2}}, \beta_{\mathrm{cal}}(0), \beta_{\mathrm{cal}}(1)_{\mathrm{MB1}}, \beta_{\mathrm{cal}}(1)_{\mathrm{MB2}}\}$

length is denoted $\beta_{\mathrm{s}}(1)$. Using γ, the detection reliability of MB2 steganograms with maximum embedded message length of 630 JPEG images (840×600, $q = 0.7$, 0.38 bpc[1]) is $\rho = 0.681$.[2] The results for other image qualities are listed in Table 1. Using $\beta_{\mathrm{s}}(0)$ alone, the reliability is still $\rho = 0.327$. If the Cauchy model had been appropriate, the detection reliability should have resulted in $\rho \approx 0$.

Another first order statistics attack is the one by Böhme and Westfeld (BW attack) [9]. It is, apart from the extreme qualities ($q = 0.99, 0.5$), somewhat more powerful. But in contrast to model-based steganalysis that considers only the global histogram, the BW attack estimates the generalised Cauchy distribution parameters for all 63 AC subbands. The parameter p_{lim} in the BW attack was set to 0.0001. This value is near the overall optimum for our image set. Besides, this parameter is rather robust to changes.

For a classification it is recommended to consider the calibrated[3] version of the images as well. We denote these as $\beta_{\mathrm{cal}}(\cdot)$. A set of the four model parameters

[1] Bits per non-zero coefficient.

[2] $\rho = 2A - 1$ for the area A under the ROC curve.

[3] The calibrated image is calculated by cutting the first four rows and columns of the JPEG image in the spatial domain and recompressing it with the original quantisation table.

$\beta_{\mathrm{s}}(0)$, $\beta_{\mathrm{s}}(1)$, $\beta_{\mathrm{cal}}(0)$, and $\beta_{\mathrm{cal}}(1)$ can be used to train a classifier.[4] (Table 5 gives an overview of all feature sets used throughout the paper.) For each quality, two classifiers (linear and quadratic discriminant analysis) were trained with 60 images and then applied to another 570 images resulting in the detection reliabilities shown in Table 1. The best result is achieved with the quadratic discriminant analysis, which is almost independent of JPEG quality and embedding method MB1 or MB2.

3 Blockiness Differences

Block artefacts increase in JPEG images when steganography is applied. Since MB1 can be easily detected by a blockiness measure [10], a deblockiness routine is implemented in MB2 to reduce the blockiness to its original value. This blockiness measure, denoted B^1, is the sum of the absolute differences of the pixel values $g_{i,j}$ along block borders (vertical and horizontal).

$$B^\lambda = \frac{\sum\limits_{i=1}^{\lfloor (M-1)/8 \rfloor} \sum\limits_{j=1}^{N} \left| g_{8i,j} - g_{8i+1,j} \right|^\lambda + \sum\limits_{i=1}^{M} \sum\limits_{j=1}^{\lfloor (N-1)/8 \rfloor} \left| g_{i,8j} - g_{i,8j+1} \right|^\lambda}{N \lfloor (M-1)/8 \rfloor + M \lfloor (N-1)/8 \rfloor} \tag{5}$$

Prior to summation the absolute differences can be exponentiated by λ. This means that B^1 is the sum of absolute differences. When B^1 and B^2 are applied to a cover and a resulting MB1 steganogram, B^1 as well as B^2 increase ("+" in Fig. 1).

If, on the other hand, they are applied to a cover and an MB2 steganogram, B^1 scarcely shows any change and B^2 decreases ("×" in Fig. 1). The blockiness reduction stops when the blockiness of the cover image is not exceeded any more or no suitable coefficients for compensation are left. In the rare latter cases, the blockiness B^1 increases for MB2. This is especially true for images with initially low blockiness. In the figure, the images are sorted by the respective blockiness measure of the cover. On the top we have the images with the least blockiness. The conclusion of this finding is that through embedding, the small differences are increased in a much bigger scale than the greater ones while both small and great differences are reduced to the same extend during blockiness adjustment.

Let B^2_{c} denote the blockiness of the cover image, B^2_{s1} the one of the MB1 steganogram and B^2_{s2} of the MB2 steganogram. If the changes of the blockiness values are compared with each other, it shows that in most cases the difference $|B^2_{\mathrm{s2}} - B^2_{\mathrm{c}}|$ is greater than $|B^2_{\mathrm{s1}} - B^2_{\mathrm{c}}|$ (cf. Fig. 1 right). So the blockiness reduction in MB2 can increase the reliability for a detection solely based on blockiness measures. This shows that the chosen blockiness reduction is inadequate for the changes the embedding function makes.

[4] When both calibrating and embedding are applied to an image (as with the calculation of $\beta_{\mathrm{cal}}(1)$) then firstly the image is calibrated and afterwards the embedding takes place.

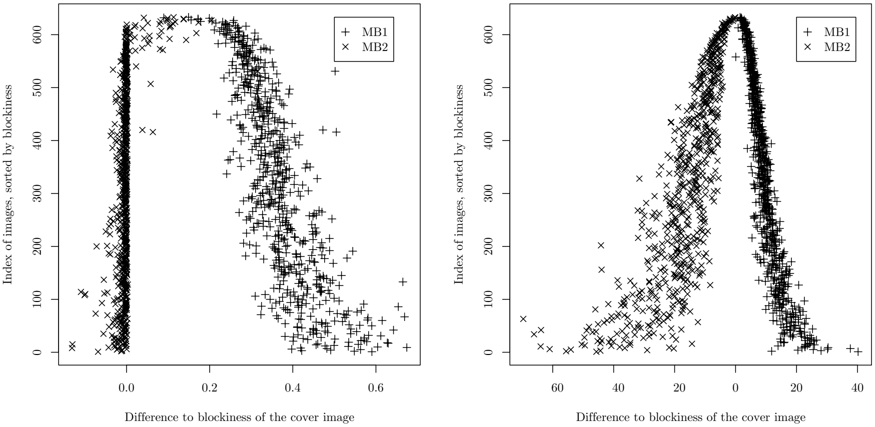


Fig. 1. Change of the blockiness measure B^1 (left) and B^2 (right). Descending ordering of cover blockiness values, i. e., the lowest image index shows the greatest blockiness value.

In our experimental set-up we consider blockiness values from four images: the blockiness $B^2_\mathrm{s}(0)$ of the given JPEG image, the blockiness $B^2_\mathrm{s}(1)$ for the image with about 95 %[5] of the MB2 capacity embedded (0.38 bpc), and two further values ($B^2_\mathrm{cal}(0)$, $B^2_\mathrm{cal}(1)$) calculated after calibration (directly after calibration and after embedding 95 %, respectively). The measure for classification is

$$m^\lambda = 1 - \frac{B^\lambda_\mathrm{s}(0) - B^\lambda_\mathrm{s}(1)}{B^\lambda_\mathrm{cal}(0) - B^\lambda_\mathrm{cal}(1)} \, . \tag{6}$$

For cover images the numerator and denominator should be equal so that $m^2 = 0$. For a steganogram we expect values greater than zero. We applied m^2 to 2900 images,[6] which resulted in the reliabilities shown in Table 2 (m^2). A moderate improvement can be achieved when the four values are used in a linear discriminant analysis (cf. Table 2, m^2 and LDA_{m^2}).

There are several options to extend the features for the LDA. One is to use both blockiness measures B^1 and B^2, resulting in 8 features. Another option is to use two more images, the original and the calibrated, with 95 % MB1 embedded, resulting in 12 features. These feature sets are denoted LDA^8_B and LDA^{12}_B in Table 2. The detection reliability of both steganogram types are considerably improved with these feature sets.

Even though the blockiness B^1 is adjusted by MB2 to its original value the embedding is still detectable using B^2 (i. e., using the same pixel value differences).

[5] Only 95 % is embedded because in most cases embedding 100 % of the MB2 capacity causes the blockiness adjustment to fail. (But not always: cf. the image to the right in Fig. 2.).

[6] Of each image we had a cover and an MB1 or MB2 steganogram with 0.02 bpc payload.

Table 2. Overview of detection reliabilities of blockiness measures. We used 2900 images for m and m_{gr}. The training set for the LDA contained 1000 images to classify the remaining 1900.

	Detection reliability ρ for 0.02 bpc											
	Simple blockiness						Gradient sensitive blockiness					
	m^1	m^2	LDA_{m^1}	LDA_{m^2}	LDA^8_B	LDA^{12}_B	m^1_{gr}	m^2_{gr}	$\text{LDA}_{m^1_{\text{gr}}}$	$\text{LDA}_{m^2_{\text{gr}}}$	$\text{LDA}^8_{B_{\text{gr}}}$	$\text{LDA}^{12}_{B_{\text{gr}}}$
MB1	0.319	0.198	0.382	0.386	0.426	0.435	0.309	0.166	0.410	0.200	0.448	0.463
MB2	0.008	0.269	0.019	0.284	0.475	0.483	0.131	0.157	0.155	0.161	0.405	0.445

The comparison with MB1 shows that B^1 gives better detection results than B^2. Consequently, it would be worse to exchange B^1 by B^2 in the blockiness reduction of MB2.

4 Gradient Aware Blockiness

Out of the variety of blockiness measures the one that is aware of gradients along block borders seems promising as well. It can be used to remove watermarks and simultaneously create a high peak signal to noise ratio [14]. In contrast to the above mentioned blockiness measure, the gradient aware blockiness B^λ_{gr} uses four instead of only two adjacent pixel values across block borders (cf. [14]):

$$B^\lambda_{\text{gr}} = \frac{\sum\limits_{i=1}^{\lfloor (M-2)/8 \rfloor} \sum\limits_{j=1}^{N} |g_{8i-1,j} - 3g_{8i,j} + 3g_{8i+1,j} - g_{8i+2,j}|^\lambda}{N \lfloor (M-2)/8 \rfloor + M \lfloor (N-2)/8 \rfloor} + \frac{\sum\limits_{i=1}^{M} \sum\limits_{j=1}^{\lfloor (N-2)/8 \rfloor} |g_{i,8j-1} - 3g_{i,8j} + 3g_{i,8j+1} - g_{i,8j+2}|^\lambda}{N \lfloor (M-2)/8 \rfloor + M \lfloor (N-2)/8 \rfloor} . \quad (7)$$

B^λ measures a non-zero blockiness for smooth areas with a gradient. Thus, the MB2 blockiness reduction reduces the small differences along the block borders, which results in a visible block artefact that was not there before. The idea of B^λ_{gr} is to measure the deviation from the gradient along block borders. So for smooth areas with a gradient, B^λ_{gr} is still zero.

MB1 steganograms linearly increase B^1_{gr} with growing length of the embedded message. For MB2 steganograms two extreme cases show up. In the first case, when there is a clear and smooth gradient in the image, B^1_{gr} slightly decreases with growing message length as long as the blockiness adjustment of MB2 is successful. When the adjustment fails, B^1_{gr} and B^2_{gr} increase significantly, as can be seen in Fig. 2. The second case — represented by the image to the right — is similar to the one of the MB1 steganogram, i. e., B^1_{gr} increases linearly with the embedded message length. Note that Fig. 2 shows the *gradient aware* measure

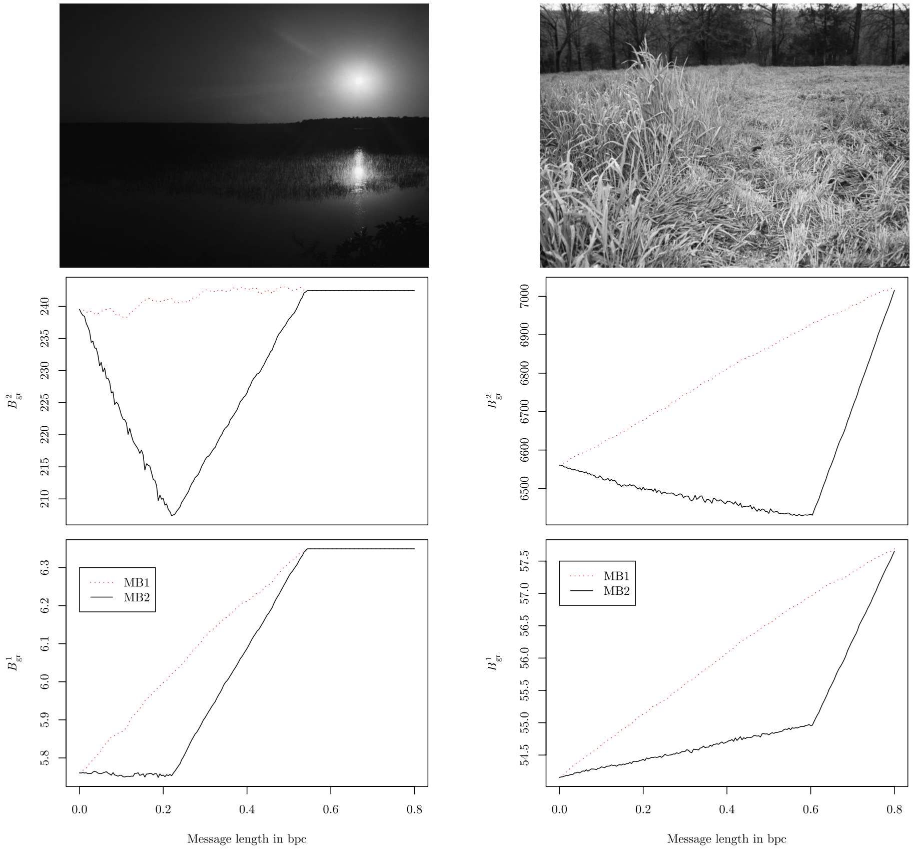


Fig. 2. Difference of blockiness changes depending on the image content. Photos courtesy of USDA Natural Resources Conservation Service.

(cf. Equ. (7)), while MB2 compensates for the measure in Equ. (5). The quadratic blockiness B^2_{gr} shows analogies to the blockiness that is not gradient aware. While B^2_{gr} increases in most cases when embedding with MB1, it decreases mostly when embedding with MB2.[7] Different is though, that the blockiness adjustment does not amplify the change in the quadratic gradient aware blockiness as it does with the simple blockiness. Only 40 % of the images show that the change of B^2_{gr} with MB2 embedding is greater than with MB1.

The measure for gradient aware classification is similarly defined to m (cf. Equ. (6)):

$$m^{\lambda}_{\text{gr}} = (B^{\lambda}_{\text{gr,s}}(0) - B^{\lambda}_{\text{gr,s}}(1)) - (B^{\lambda}_{\text{gr,cal}}(0) - B^{\lambda}_{\text{gr,cal}}(1))\,. \tag{8}$$

[7] Out of 2000 images (840×600, $q = 0.8$, 0.38 bpc) B^1_{gr} decreases zero (MB1) respectively 26 (MB2) times and B^2_{gr} decreases six (MB1) respectively 1936 (MB2) times.

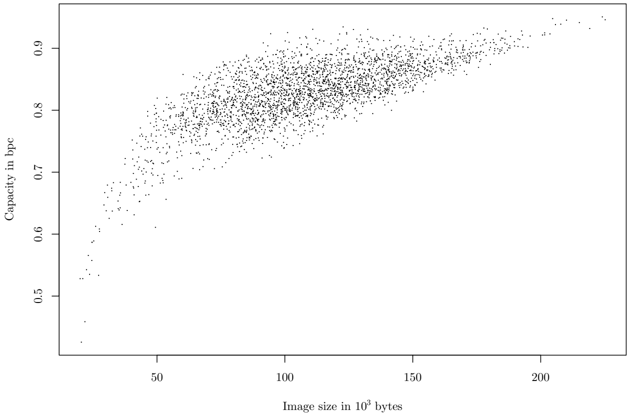


Fig. 3. Dependency between image size and maximum MB1 capacity

For $\lambda = 1$, smaller values of m^1_{gr} indicate a cover and larger values a steganogram. On the contrary and because of the inverse changing direction of B^2_{gr}, larger values of m^2_{gr} indicate a cover and smaller values a steganogram.

We can enhance the feature set the same way as for the simple blockiness measure to 8 and 12 features. The results are given in Table 2.

Comparing the results of the simple blockiness and the gradient aware it turns out that the simple one generates better results for MB2 steganograms (with less features) while the gradient sensitive generates better results for MB1 steganograms.

A classification solely based on the image and its calibrated version showed a very low reliability only ($\rho = 0.034$). A closer look at the histogram of the small differences of neighbouring pixels at block borders might improve the results. Instead of using the blockiness itself we use the histogram h_d of the first six difference values $d = 0, \ldots, 5$ from the direct and the calibrated image. Those twelve features used in an LDA result in a detection reliability of $\rho = 0.117$ for 0.02 bpc.[8] This is an improvement, but compared to the results with the six images this is far less.

Figure 2 raises another interesting point. The image with the obvious smooth gradient has a low steganographic capacity of only 0.54 bpc while the other image has a capacity of over 0.8 bpc. Also, images with visible gradients have a smaller image size (in bytes) because of the sparse non-zero DCT coefficients. In Fig. 3 the dependency between the image size and the MB1 capacity of 2900 images ($q = 0.8$, 840×600) is displayed. Note that the presented capacity is already relative to the number of non-zero coefficients.

[8] The LDA was trained with 1900 images and another 1000 images of size 840×600 with $q = 0.8$ were classified.

5 Coefficient Types

The blockiness adjustment in MB2 generates additional modifications which can be used for an attack. Deduced from the blockiness adjustment a differentiation of coefficients is suggested. The reason for this is that neither the coefficients used for embedding nor the coefficients that do not decrease the blockiness are altered during the blockiness adjustment. Regarding the blockiness reduction three sets of coefficient types can be defined. These three are the set of

Fixed coefficients F, characterised by the fact that they can not be altered because of model restriction even though they would decrease the blockiness if changed. The set of

Different coefficients D are the ones that can be altered and if so they would decrease the blockiness. The remaining set of

Indifferent coefficients I could be altered or not but if they are altered the blockiness would increase.

In order to categorise the coefficients a blockiness minimal image needs to be created. This is done with a routine that is included in the MB2 algorithm, which is also used for blockiness reduction. To generate the blockiness minimal image, the (given) image is firstly transformed into the spatial domain (after the dequantisation). Then the opposing pixels along block borders are set to their mean value. Subsequently the resulting image is transformed into the frequency domain and quantised. Thus, the generated image is the unique blockiness minimal image for the given image.

It is advisable to separate the coefficient types further into the disjoint sets DC, AC1 and AC0 of coefficients. AC0 coefficients are the AC coefficients that are zero. The separation is useful because MB1 and MB2 do not use DC and AC0 coefficients for embedding. Having the blockiness minimal image at hand the indifferent coefficients can be enumerated. Doing this we get indifferent AC1, DC and AC0 coefficients. To separate D from F we need to take a closer look at the restrictions of the blockiness adjustment. MB1 and MB2 only modify coefficients within their bin in order to keep the low precision bins unchanged so the receiver of the steganogram can calculate the same model of the image. Thus, if a coefficient value needed to be altered into another low precision bin in order to decrease the blockiness it is an element of F because it must not be altered that way to keep the extraction of the model parameters correct. This again could happen with DC, AC1 and AC0 coefficients. The remaining coefficients form the set D. They differ between the image and the blockiness minimal image but they could be altered without changing low precision bins and if so cause the blockiness to decrease.

In the case of MB1 a shrinkage of F and I is expected while D should expand. With MB2 the cardinality of D should decrease because the longer the embedded message length the more coefficients are changed in order to decrease the blockiness, which can only be done by coefficients contained in D. Thus, F expands because elements of D can only be moved into I or F. Since the blockiness is increasing during embedding the cardinality of I can not increase significantly

Table 3. Overview of detection reliabilities of the coefficient types attack. The best result was achieved with the SVM using the C-classification with a polynomial kernel (second degree). For m_{type} 2900 images are used to specify the detection reliability. The training set for LDA and SVM were 2300 images and another 630 images were classified.

	Detection reliability ρ for 0.02 bpc				
	m_{type}	LDA(T^6)	LDA(T^{18})	SVM(T^6)	SVM(T^{18})
MB1	0.019	0.364	0.513	0.390	0.563
MB2	0.630	0.645	0.808	0.678	0.838

and so F needs to expand. Empirical analysis shows that the cardinality of I increases in most cases (92 %) too.

Let T_{s} be the number of just one coefficient type or a combination of coefficient types and T_{cal} its calibrated version. Then the following measure can be used to mount a specific attack to model-based steganography:[9]

$$m_{\text{type}} = \frac{T_{\text{s}}(-1) - T_{\text{s}}(0)}{T_{\text{s}}(-1) - T_{\text{s}}(1)} - \frac{T_{\text{cal}}(-1) - T_{\text{cal}}(0)}{T_{\text{cal}}(-1) - T_{\text{cal}}(1)} \, . \tag{9}$$

For $T = |D|$ a detection reliability of $\rho = 0.572$ can be achieved for MB2 steganograms with 0.02 bpc. This can be further improved if not only the number of *different* coefficients but a combination of all fixed, i. e., fixed AC1, AC0

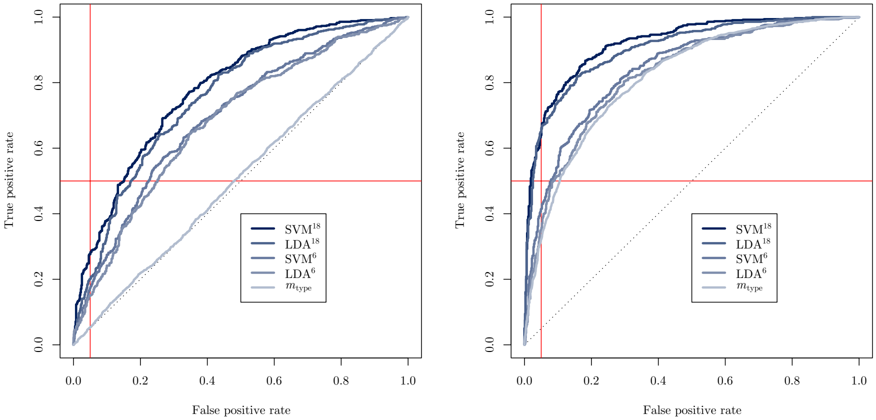


Fig. 4. ROC curves for coefficient types attack on MB1 (left) and MB2 images (right) with 0.02 bpc embedding rate

[9] The values of $T_{\text{s}}(-1)$ and $T_{\text{cal}}(-1)$ are calculated after embedding 95 % of MB1 capacity with MB1 and the values $T_{\text{s}}(1)$ and $T_{\text{cal}}(1)$ are calculated after embedding 95 % of MB2 capacity with MB2 regardless of the image. $T_{\text{s}}(0)$ and $T_{\text{cal}}(0)$ are calculated without any additional embedding.

and DC coefficients, and the indifferent AC1 coefficients is selected as the basis for the classification (cf. Table 5). The combination of those leads to a detection reliability of $\rho = 0.630$.

To advance this approach of combining several coefficient types an LDA can be used. Given the cardinality of some types the cardinality of the complements can be calculated. Thus, we reduce the number of types we use in the LDA. If $|F_{\mathrm{AC0}}|$, $|D|$, and $|I_{\mathrm{AC0}}|$ for coefficients of the six images: image (A), calibrated image (B), A + 95 % MB1, A + 95 % MB2, B + 95 % MB1, and B + 95 % MB2 (cf. Table 5), are used, we have 18 features for the LDA. The detection reliability can be increased as depicted in Fig. 4 and Table 3: LDA(T^{18}). (The superscript index states the number of features.) If only the image and its calibrated version are used, the detection reliability is $\rho = 0.645$: LDA(T^{6}).

6 Another “Blind” Attack

In the earlier sections some hints for the integration of new features for known blind classifiers have been given. In this section we focus on the actual combination of new features with existing ones. The most promising single feature set for the detection of MB2 is the use of three coefficient types of six images, which

Table 4. Detection reliability for feature combinations for a new blind or specific attack. The 23 represents the attack by Fridrich with 23 features, the 274 the one with 274 features [13] and the 324 the one by Shi et al. [12] with the 324 features. The 81 and 193 features are a splitting of the 274 features, where the 81 represent the calibrated averaged Markov features. The training set was 2300 images and the classification was done by LDA based on 630 images ($q = 0.8$, 840×600, message length is 0.02 bpc).

		Additional	Number of features				
		features	23	324	274	81	193
ρ	MB1	—	0.181	0.597	0.698	0.585	0.516
		T^{6}	0.379	0.706	0.740	0.652	0.600
		T^{18}	0.510	0.759	0.770	0.700	0.675
		$B_{\mathrm{gr}}^{(12)}$, T^{18}	0.596	0.793	0.791	0.743	0.708
	MB2	—	0.187	0.659	0.759	0.666	0.518
		T^{6}	0.734	0.895	0.885	0.859	0.793
		T^{18}	0.842	0.919	0.930	0.903	0.886
		$B_{\mathrm{gr}}^{(12)}$, T^{18}	0.873	0.924	0.937	0.919	0.909
$\mathrm{FPR}_{0.5}$	MB1	—	0.348	0.133	0.077	0.133	0.136
		T^{6}	0.259	0.067	0.055	0.096	0.125
		T^{18}	0.172	0.052	0.047	0.066	0.092
		$B_{\mathrm{gr}}^{(12)}$, T^{18}	0.140	0.050	0.041	0.060	0.080
	MB2	—	0.370	0.105	0.042	0.086	0.147
		T^{6}	0.052	0.016	0.012	0.020	0.030
		T^{18}	0.020	0.012	0.006	0.017	0.012
		$B_{\mathrm{gr}}^{(12)}$, T^{18}	0.019	0.008	0.006	0.011	0.011

Table 5. Overview of used images and features for the different classifications described in the paper

				Images used					
Where	Row	Denotion	Feature	Image (A)	Calibrated image (B)	A + 95 % MB1	A + 95 % MB2	B + 95 % MB1	B + 95 % MB2
Table 1		$\beta_s(0)$	β	×					
		γ	β	×			×		
		M_4	β	×	×		×		×
		M_6	β	×	×	×	×	×	×
Table 2	for MB1	m^1	B^1	×	×	×		×	
		m^2	B^2	×	×	×		×	
	for MB2	m^1	B^1	×	×		×		×
		m^2	B^2	×	×		×		×
		LDA_B^8	B^1	×	×	×		×	
			B^2	×	×	×		×	
		LDA_B^{12}	B^1	×	×	×	×	×	×
			B^2	×	×	×	×	×	×
	for MB1	m_{gr}^1	B_{gr}^1	×	×	×		×	
		m_{gr}^2	B_{gr}^2	×	×	×		×	
	for MB2	m_{gr}^1	B_{gr}^1	×	×		×		×
		m_{gr}^2	B_{gr}^2	×	×		×		×
		$\mathrm{LDA}_{B_{\mathrm{gr}}}^8$	B_{gr}^1	×	×	×		×	
			B_{gr}^2	×	×	×		×	
		$\mathrm{LDA}_{B_{\mathrm{gr}}}^{12}$	B_{gr}^1	×	×	×	×	×	×
			B_{gr}^2	×	×	×	×	×	×
Tables 3&4		m_{type}	$\lvert F_{\mathrm{AC0}}\rvert + \lvert F_{\mathrm{AC1}}\rvert + \lvert F_{\mathrm{DC}}\rvert + \lvert I_{\mathrm{AC1}}\rvert$	×	×	×	×	×	×
		T^6	$\lvert F_{\mathrm{AC0}}\rvert$	×	×				
			$\lvert D\rvert$	×	×				
			$\lvert I_{\mathrm{AC0}}\rvert$	×	×				
		T^{18}	$\lvert F_{\mathrm{AC0}}\rvert$	×	×	×	×	×	×
			$\lvert D\rvert$	×	×	×	×	×	×
			$\lvert I_{\mathrm{AC0}}\rvert$	×	×	×	×	×	×
		$B_{\mathrm{gr}}^{(12)}$, T^{18}	B_{gr}^1	×	×	×	×	×	×
			B_{gr}^2	×	×	×	×	×	×
			$\lvert F_{\mathrm{AC0}}\rvert$	×	×	×	×	×	×
			$\lvert D\rvert$	×	×	×	×	×	×
			$\lvert I_{\mathrm{AC0}}\rvert$	×	×	×	×	×	×

is labelled as LDA(T^{18}) in Table 3. But this method of detection includes the use of the algorithms MB1 as well as MB2 to generate further images, which are needed for the feature set of the classification. Combining these six features with the L1 and L2 norm of the gradient aware blockiness of those six images, the detection reliability can be increased.

Adding those 30 features to the 81 averaged calibrated Markov features proposed in [13], the detection reliability can be increased significantly, as can be seen in Table 4 in the last rows. The improvement when the 81 are replaced by all 274 features is only small for MB2 but larger for MB1.

So far one could argue that the described feature set belongs more likely to a specific attack, because the embedding algorithms MB1 and MB2 are used for the extraction of features. If those features are ignored and only the features of the image and its calibrated version are used, the detection reliability drops noticeably. Even though it decreases, the detection reliability for images with a quality of 80 %, a size of 840 × 600 and an embedding rate of 0.02 bpc is significantly higher for 274+T^6 than for 274 features alone. The blind attack with 280 features reaches a detection reliability for MB2 steganograms of $\rho = 0.885$ and for MB1 steganograms of $\rho = 0.740$. The false positive rate for a detection level of 50 % is 5.5 % for MB1 and 1.2 % for MB2. Table 3 has shown that an SVM with the right parameters can moderately increase the detection reliability. However, these parameters have to be determined for each feature set again. An SVM with standard parameters and radial kernel as used in [13] does not provide any better (rather worse) results than those already listed in Table 4. Besides the detection reliability ρ, Table 4 presents the false positive rates at 50 % detection rate $\mathrm{FPR}_{0.5}$. According to Ker's criterion [16], a good attack has at most 5 % false positives at 50 % true positive rate. The proposed attacks reduce the false positive rates by a factor of 1.7. . . 2.7 for MB1 and 7. . . 19 for MB2.

7 Conclusion and Further Work

MB2 is detectable by blockiness measures. It is even more reliably detected by the change to the number of *fixed*, *different*, and *indifferent* coefficients, which are discriminated by the blockiness reduction of MB2. Our proposed set of 30 new features used in combination with current blind feature sets can increase the reliability for a very low embedding rate (0.02 bpc) from $\rho \approx 0.7$ to $\rho \approx 0.9$ while decreasing the false positive rate from $\mathrm{FPR}_{0.5} \approx 0.1$ to $\mathrm{FPR}_{0.5} \approx 0.01$ (cf. Table 4).

Another important finding is that additional changes applied to a steganogram to reduce steganographic artefacts increase once more the reliability of an attack. This was true for histogram preserving compensations in Outguess [17]; this is also true for the blockiness compensation in MB2 [12,13]. A much better approach to reduce detectability could be informed embedding combined with a more sophisticated model for DCT coefficients, taking into account different qualities of the image. The fitness of a model for a particular dataset can be assessed in a model-based steganalysis as we did for the Cauchy model in this paper.

Acknowledgement

The work on this paper was supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under the research grant number 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. Fridrich, J., Goljan, M., Soukal, D.: Higher-order statistical steganalysis of palette images. In: Delp, E.J., Wong, P.W. (eds.) Security, Steganography and Watermarking of Multimedia Contents V (Proc. of SPIE), San Jose, CA, vol. 5020, pp. 178–190 (2003)
2. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Delp, E.J., Wong, P.W. (eds.) Security, Steganography and Watermarking of Multimedia Contents VI (Proc. of SPIE), San Jose, CA, vol. 5306, pp. 23–34 (2004)
3. Dumitrescu, S., Wu, X., Wang, Z.: Detection of LSB steganography via sample pair analysis. IEEE Trans. of Signal Processing 51, 1995–2007 (2003)
4. Yu, X., Wang, Y., Tan, T.: On estimation of secret message length in Jsteg-like steganography. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), pp. 673–676 (2004)
5. Zhang, T., Ping, X.: A fast and effective steganalytic technique against Jsteg-like algorithms. In: Proceedings of the 2003 ACM Symposium on Applied Computing (SAC 2003), Melbourne, Florida, USA, March 9-12, 2003, pp. 307–311. ACM Press, New York (2003)
6. Lee, K., Westfeld, A., Lee, S.: Category Attack for LSB steganalysis of JPEG images. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 35–48. Springer, Heidelberg (2006)
7. Lee, K., Westfeld, A., Lee, S.: Generalised Category Attack—improving histogram-based attack on JPEG LSB embedding. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 378–391. Springer, Heidelberg (2008)
8. Sallee, P.: Model-based steganography. In: Kalker, T., Ro, Y.M., Cox, I.J. (eds.) IWDW 2003. LNCS, vol. 2939, pp. 154–167. Springer, Heidelberg (2004)
9. Böhme, R., Westfeld, A.: Breaking Cauchy model-based JPEG steganography with first order statistics. In: Samarati, P., Ryan, P., Gollmann, D., Molva, R. (eds.) ESORICS 2004. LNCS, vol. 3193, pp. 125–140. Springer, Heidelberg (2004)
10. Sallee, P.: Model-based methods for steganography and steganalysis. International Journal of Image and Graphics 5(1), 167–190 (2005)
11. Fridrich, J.J.: Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In: [18], pp. 67–81
12. Shi, Y.Q., Chen, C., Chen, W.: A Markov process based approach to effective attacking JPEG steganography. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 249–264. Springer, Heidelberg (2007)
13. Pevný, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. In: Delp, E.J., Wong, P.W. (eds.) SPIE 2007, Security, Steganography, and Watermarking of Multimedia Contents IX, Proceedings of the SPIE, San José, January 2007, vol. 6505 (2007)

14. Westfeld, A.: Lessons from the BOWS contest. In: MM&Sec 2006: Proceeding of the 8th Workshop on Multimedia and Security, pp. 208–213. ACM Press, New York (2006)
15. ECRYPT: BOWS, Break our watermarking system (2006), `http://lci.det.unifi.it/BOWS`
16. Ker, A.D.: Improved detection of LSB steganography in grayscale images. In: [18], pp. 97–115
17. Fridrich, J.J., Goljan, M., Hogea, D.: Attacking the Outguess. In: Proc. of ACM Multimedia and Security Workshop 2002, MM&Sec06, Juan-les-Pins, France, pp. 967–982. ACM Press, New York (2002)
18. Fridrich, J.J. (ed.): IH 2004. LNCS, vol. 3200. Springer, Heidelberg (2004)

Effect of Different Coding Patterns on Compressed Frequency Domain Based Universal JPEG Steganalysis

Bin Li[1,2], Fangjun Huang[1], Shunquan Tan[1], Jiwu Huang[1], and Yun Q. Shi[2]

[1] School of Information Science and Technology,
Sun Yat-sen University, Guangzhou 510275, China
isshjw@mail.sysu.edu.cn
[2] New Jersey Institute of Technology, Newark, NJ 07102, USA
shi@njit.edu

Abstract. Current steganalytic schemes for JPEG steganography are in favor of extracting features in DCT (Discrete Cosine Transform) domain and/or spatial domain. A recently proposed compressed frequency domain based universal steganalytic algorithm [21] showed concern over the statistics of Huffman compressed stream. The authors claimed that it was very effective in detecting stego images embedded by JPEG steganographic tools, including JPHide, F5 and OutGuess. Even though only several bytes were embedded, the scheme was still able to work, thus demonstrating astonishing steganalysis capability. By carefully controlled studies on the factors which may have impact on the new steganalytic method, we find out the truly cause of the powerfulness of this "payload-independent" steganalyzer. Experimental results reveal that different coding patterns used in cover and stego images, rather than the Avalanche Criterion [24] explained by its authors, have led to great detection efficiency. When the same coding pattern is employed in both cover and stego images, the performance of the newly devised steganalyzer has greatly dropped. Implication from this paper is that we should ensure the difference between the cover and stego images is only caused by data embedding itself in steganography and steganalysis.

1 Introduction

Steganography is the practice of secret communication by hiding information in cover medium without drawing suspicion. JPEG image is one of such media popular with steganographists since it is a widely used format for storing image data in our daily life. Several steganographic schemes, such as Jsteg [1], JPHide [2], F5 [3], OutGuess [4], MB1 [5], MB2 [6], PQ [7] and MME [8], hide secret message bits in JPEG image by altering its quantized DCT (Discrete Cosine Transform) coefficients. Discovering the existence of the covert communication is the purpose of steganalysis. If the steganalytic method can determine whether or not a medium contains hidden information with a success rate better than random guessing, the steganographic algorithm is considered to be broken. Similar to cryptography and cryptanalysis, steganography and steganalysis develop with each other.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 143–157, 2008.

Jsteg [1] and JPHide [2] are JPEG steganographic tools at the early stage. They replace the least significant bit (LSB) of DCT coefficients with secret message bit and they are detectable by the chi-square attack [9] and the generalized chi-square attack [10,11]. The F5 algorithm [3] can resist the chi-square attack by decreasing the absolute values of the coefficients. Additionally, it employs the matrix encoding technique to improve the embedding efficiency. However, it does not preserve the first order statistic well, especially introducing more zero coefficients so that the histogram of DCT coefficients is shrunk to the center. Another LSB based steganographic tool, OutGuess [4], preserves the global histogram of DCT coefficients to counter the chi-square attack, but it increases block discontinuity in spatial domain. Fridrich et al. [12] presented a calibration method to estimate the statistics of the JPEG cover image by decompressing, cropping and re-compressing. It is effective in detecting F5 as well as OutGuess [13,14]. Later, Sallee [5] designed the model-based steganography, named by MB1. The embedding of MB1 keeps the low precision histogram unchanged by fitting a generalized Cauchy distribution model. The improved version defending against Fridrich's blockiness attack is known as MB2 [6]. By observing the phenomenon of the distribution of DCT coefficients over-fitting the generalized Cauchy distribution, Böhme et al. [15] successfully launched an attack against MB1. This method is also theoretically effective for attacking MB2.

The steganalytic methods mentioned above are of one category which is called specific steganalysis. It is used to break one targeted steganographic algorithm and seldom applicable to others. Another category which draws more and more attention nowadays is the universal steganalysis, or called blind steganalysis. It can detect several steganographic algorithms, but it is less accurate than the specific ones. Based on the techniques used in pattern recognition, the universal steganalysis consists of two major steps. The first step is to extract features which can accurately capture the statistical changes by steganography. The second step is to use a classifier, such as Fisher linear discriminant (FLD), neural network (NN) or support vector machine (SVM), to distinguish the extracted features from cover and stego objects. The "universal" property is mainly contributed by good features, which reflect the statistics deviated by steganographic schemes and are usually changed along with the size of embedding payload. In [16], Farid proposed a steganalyzer with features extracting from a multi-scale decomposition of an image and its prediction error. It provides a satisfactory detection rate. Avcibas et al. [17] obtained features on correlation between the contiguous bit planes of spatial domain representation of image. As [16], Avcibas et al.'s method is effective to JPEG steganography as well as spatial steganography. The calibrated DCT feature based steganalyzer developed by Fridrich [18] is very powerful in attacking serval JPEG steganography. More satisfactory detection results come from the steganalyzer proposed by Shi et al. [19], which utilizes the Markov process to explore the correlation within DCT coefficients. The steganalyzer merging the Markov based and the calibrated DCT based features performs even better [20]. Among the existing universal steganalytic schemes, extracting features directly in DCT domain (e.g. [18,19,20]) appears to be more sensitive to JPEG steganography than in spatial domain (e.g. [16,17]).

Nonetheless, the state-of-the-art steganalytic techniques cannot work well when embedding rate is extremely low.

Recently, Barbier et al. [21] have shown a new point of view on JPEG steganalysis. Their feature extraction was conduct in compressed frequency domain. By examining the Huffman compressed binary stream, a steganalyzer with high and constant detection rate, regardless of the size of embedding payload, was obtained in their setting. It outperforms all existing universal steganalytic algorithms when the embedding rate is low. Besides, the steganalyzer is able to detect a steganographic algorithm not used during the training stage. In this sense, the new steganalytic method seems really "universal". The compressed frequency domain based features also led to a specific steganalyzer [22].

In this paper, we take a close look on this compressed frequency domain based universal steganalytic scheme. Several factors, which may have impact on the scheme but are not clearly mentioned in [21], are carefully examined here. We find out the detection results have been exaggerated by wrong selection of coding patterns. If the same coding pattern is used in both cover and stego images, the detection performance drops.

The rest of this paper is organized as follows. In Sect. 2, we briefly review the basics of JPEG compression and the compressed frequency domain based steganalysis. Investigations through extensive controlled experiments are presented in Sect. 3 to demonstrate that the powerfulness of the newly devised steganalyzer is mainly contributed by different coding patterns. We discuss the obtained results and implications in Sect. 4. Conclusions are drawn in the last section.

2 Steganalysis in JPEG Compressed Frequency Domain

2.1 The Basics of JPEG Compression

The JPEG standard defines four types of operation modes: **sequential**, **progressive**, **hierarchical** and **lossless** [23]. The first three modes are lossy schemes using DCT-based coding and the last one uses a predictive method for lossless compression. In the DCT-based coding scheme, the source image samples are firstly grouped into non-overlapped 8×8 blocks and each block is transformed by the forward Discrete Cosine Transform into a set of 64 values referred to as DCT coefficients. One of these values is referred to as the DC coefficient and the other 63 as the AC coefficients. In the next step, each of the 64 coefficients is quantized using one of 64 corresponding values from a quantization table and then rounded. Then, the previous quantized DC coefficient is used to predict the current quantized DC coefficient, and the difference is encoded. The 63 quantized AC coefficients are converted into a one dimensional zig-zag sequence and undergo run-length encoding. The differential encoding and run-length encoding are done within a single left-to-right and top-to-bottom scan in sequential mode, while in multiple scans in progressive mode. Finally, the entropy coding is used to achieve additional compression losslessly by encoding the quantized DCT coefficients according to their statistical characteristics. One of the two entropy coding, Huffman encoding and arithmetic encoding, can be used. The Huffman encoding is more widely applied because of the patent restriction on arithmetic encoding. The output data of the entropy encoder is the JPEG raw binary stream.

2.2 The Compressed Frequency Domain Based Steganalysis

Barbier et al. noticed that the entropy of the JPEG raw binary stream (e.g. Huffman compressed stream) would change after the quantized DCT coefficients were modified by JPEG steganographic tools. They proposed a universal JPEG steganalytic method based on the statistics of the compressed data stream. The idea of the algorithm is as follows [21]. Let I be the binary stream output by Huffman encoding. Assume the length of the stream is m and let b_j be the j-th bit. The entropy $H(I)$ of 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 of "0" in the stream. It can be computed as

$$P(I) = \frac{1}{m}\sum_{j=1}^{m}(1 - b_j) \tag{2}$$

Hiding message may introduce a variation of entropy, which is related to $P(I)$. In the observation of [21], $P(I)$ follows a Gamma distribution where the peak located at smaller than 0.5 for non-stego images, while $P(I)$ follows a normal distribution centered at 0.5 for stego images regardless of the embedding rate[1]. In other words, the hypothesis in this scenario is that the mean of $P(I)$ would move to 0.5 after only several coefficients are modified. The authors of [21] claimed that this phenomenon could be explained by the Avalanche Criterion [24], though no further solid proof was given. However, if this observation were true, it would bring great impact to steganography and steganalysis. Detecting the DCT-based JPEG steganography in extremely low embedding rate would no longer be "mission impossible".

In the practical steganalytic scheme, the stream is divided into blocks of equal size and the Hamming weight is computed for each block to measure $P(I)$ in a finer way. A random variable X is used to describe the outcome values of Hamming weight. The observed probability density function $\hat{p}(x)$ of X and the i-th order moment $M_i(I)$ (e.g. $i \in \{1, 2, .., 5\}$) are computed. Then the Kullbak-Leibler (KL) distance or relative entropy [25] between the $\hat{p}(x)$ and the average probability function $p(x)$, which is experimentally computed by averaging many cover images, is derived. So is the KL distance between $p(x)$ and $\hat{p}(x)$. The KL distances and $M_i(I)$ are selected as features. The authors used the gray and color images of different sizes (probably of different quality factors) downloaded from the Internet in their experiments. JPHide, F5 and OutGuess were selected to verify their approach. An FLD based classifier was trained to distinguish between cover and stego images. The authors also claimed that the performance of this steganalyzer would be independent of the embedding rate. Even an incredible low embedding rate, such as 10^{-6} of the cover image size, would be detectable.

[1] See Fig. 2 in [21] for illustration.

3 Investigation on Different Coding Patterns for JPEG Steganalysis

It is very unusual that the statistics of the compressed stream of stego images do not depend on the embedding rate. In [21], even though the embedding rate is low, the entropy difference between the cover and stego images is obvious. This would be a formidable challenge for current JPEG steganography at first glance. It is important to investigate this compressed frequency domain based steganalyzer.

3.1 Different JPEG Operation Modes and Huffman Tables

Progressive mode is claimed to be used in [21]. However, there is no option for JPHide, F5 and OutGuess to choose the operation mode. They all use **sequential mode** by default. In other words, the operation mode of the cover image and the stego image might be different. Besides, there are two ways to obtain Huffman tables in a practical JPEG encoder. One way is to use the predefined Huffman tables which are suggested in the annex of the JPEG standard. Although there is no default tables defined in the standard, they are used as a default option to save compression time in many JPEG encoders, such as the implementation by IJG[2]. The other way is to use the optimized Huffman tables which are adaptively derived from each image based on its statistics. It makes the entropy compression more efficient in trade of additional computation time in encoding. Progressive mode definitely uses optimal Huffman tables for each scan while the sequential mode may choose predefined Huffman tables or optimal Huffman tables.

JPHide and OutGuess are implemented based on IJG codec. JPHide employs the **sequential mode with optimized Huffman tables** while OutGuess makes use of the **sequential mode with predefined Huffman tables**. The JPEG encoder of F5 is written in Java but it borrows a great deal of codes and structures from IJG. And the setting for F5 is the **sequential mode with predefined Huffman tables**.

Different JPEG operation modes and different types of Huffman tables may be highly probable of increasing or decreasing the entropy of Huffman compressed data stream. In the following subsections, we will show the role of the operation mode and the Huffman table, as well as the steganographic algorithms and some other factors, in the newly devised steganalytic scheme with comprehensive experiments.

3.2 Experimental Environment

Since we cannot obtain the same cover image set as in [21], we generate JPEG images by ourselves. In return, we can set different parameters such as the quality factor, operation mode and Huffman table in the process of JPEG compression so that the truly cause of the entropy deviation of the Huffman compressed stream

[2] Independent JPEG Group. `http://www.ijg.org`

can be revealed. There are 1338 uncompressed TIFF format color images of size 384×512 (or 512×384) in UCID [26] image database. All these images are used in our experiments. We apply the publicly available JPEG implementation library IJG version 6b to generate cover image sets. Three kinds of coding pattern are used. They are:

1. Sequential mode with predefined Huffman tables, denoted as "**FIX**" in the context.
2. Sequential mode with optimized Huffman tables, denoted as "**OPT**".
3. Progressive mode, denoted as "**PRO**".

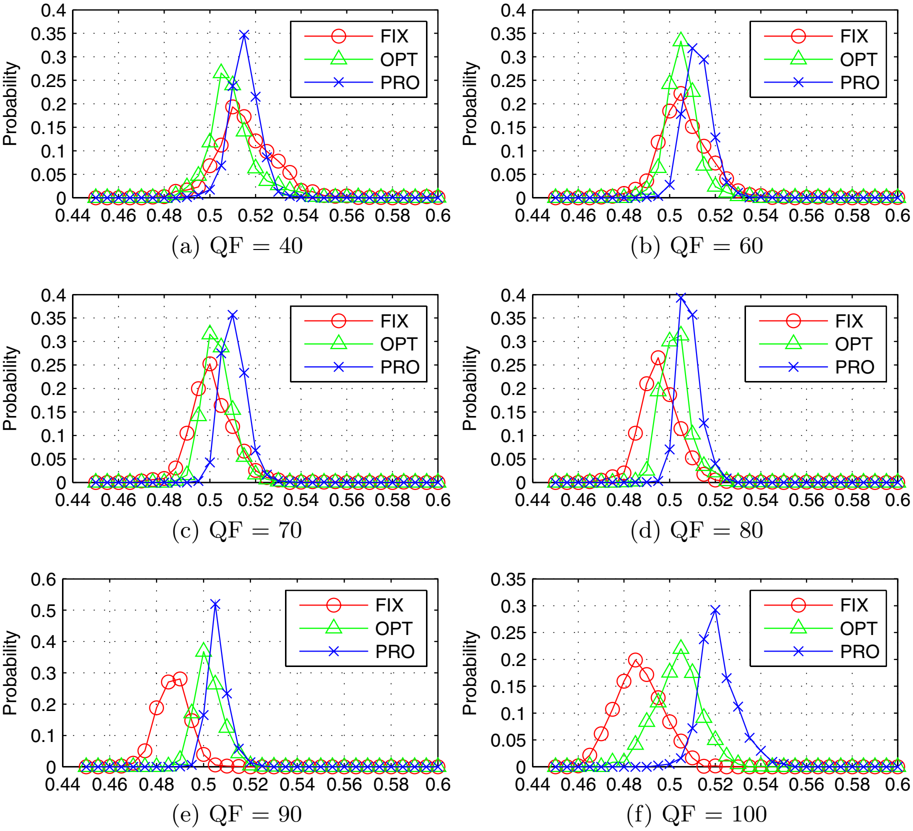
We can easily convert images within these coding patterns by Matlab JPEG Toolbox [27] which is also based on IJG version 6b. Under this circumstance, for example, an image compressed with the coding pattern PRO by IJG is exactly the same as the image compressed with the coding pattern FIX or OPT by IJG and then converted by JPEG Toolbox to PRO. The stego images with different coding patterns we generated are similarly derived in this way.

3.3 Results of Investigation

The role of coding pattern. The primary task is to find out if there is any difference in entropy within three coding patterns. We compress the TIFF format images into JPEG using the quality factor (QF) varying from 40 to 100. With coding pattern FIX, OPT and PRO respectively, three kinds of cover image sets are generated for each QF. No message is embedded. The probability of "0" in Huffman compressed stream, denoted as $P(I)$, is computed for each image. Distribution of $P(I)$ within 1338 images compressed by each coding pattern are compared in the Fig. 1. We can observe that the distribution of $P(I)$ is different in three coding patterns, though three image sets contain all the cover images without embedding any bit. So we can conclude that the entropy of the Huffman compressed stream can be changed by coding pattern conversion. Furthermore, the difference within the coding patterns become obvious with larger QFs.

The role of quality factor. We plot the results of image sets with the same coding pattern but using different quality factors in Fig. 2. As we can see, the distribution curves shift from right to left as QF increases in coding pattern FIX (QF=100 is an exception). For most of the higher QFs, the peaks of the curves concentrate on 0.5 in coding pattern OPT. Although the distribution curves in coding pattern PRO shift from right to left, the peaks are never located at less than 0.5. These results prove that not only the coding pattern but also the quality factor has effect on $P(I)$. Lowering the QF tends to make more zero coefficients, thus the distribution curve shifts.

In addition, we do not get a Gamma distribution, which can model the sum of normal distributions, for the cover image set as in [21]. It is possible that the cover image set in [21] might be composed of images with different coding patterns and different quality factors. This is consist with the fact that the images downloaded from the Internet are of great diversity.



(a) QF = 40

(b) QF = 60

(c) QF = 70

(d) QF = 80

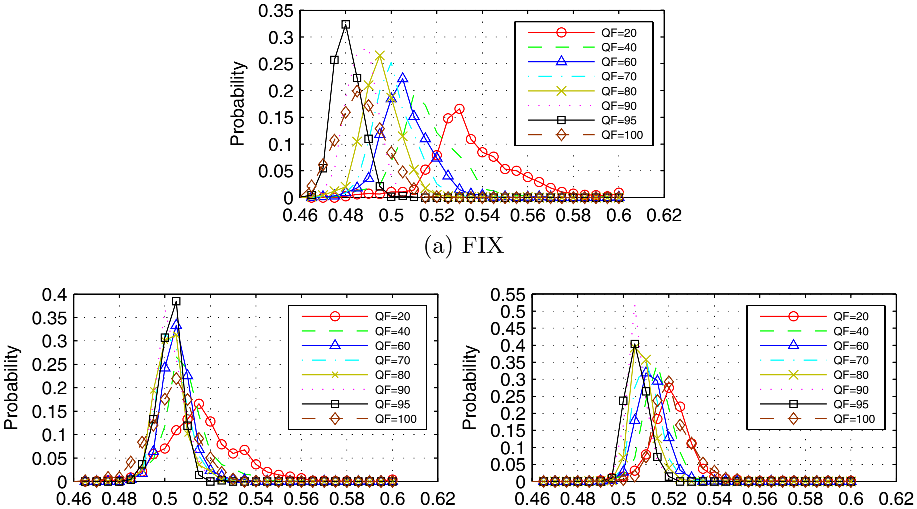
(e) QF = 90

(f) QF = 100

Fig. 1. Distribution of $P(I)$ within 1338 cover images with different coding patterns using a unique quality factor (QF): (a) QF=40, (b) QF=60, (c) QF=70, (d) QF=80, (e) QF=90 and (f) QF=100. Red solid with ○: FIX, green solid with △: OPT, blue solid with ×: PRO. (The horizontal axis and the vertical axis represent the value of $P(I)$ and its probability, respectively. They are the same for Fig. 2 to Fig. 5).

The role of double JPEG compression. F5 and OutGuess decompress the JPEG cover image and then re-compress it with a user specified quality factor before embedding. If the QF of the cover image and the stego image are not equal, it would probably introduce serious double JPEG compression artifacts, which means the histogram of coefficients might not resemble a generalized Cauchy distribution or generalized Gaussian distribution any more. This phenomenon confuses some existing universal steganalyzers by improper training, as it is pointed out in [28,29]. If the image is compressed with the same QF twice, some coefficients are also changed due to the rounding and truncation of inverse DCT in decompression and forward DCT in re-compression.

In an attempt to find out the effect of double JPEG compression on the entropy of Huffman compressed stream, we generate three types of double JPEG



(a) FIX

(b) OPT

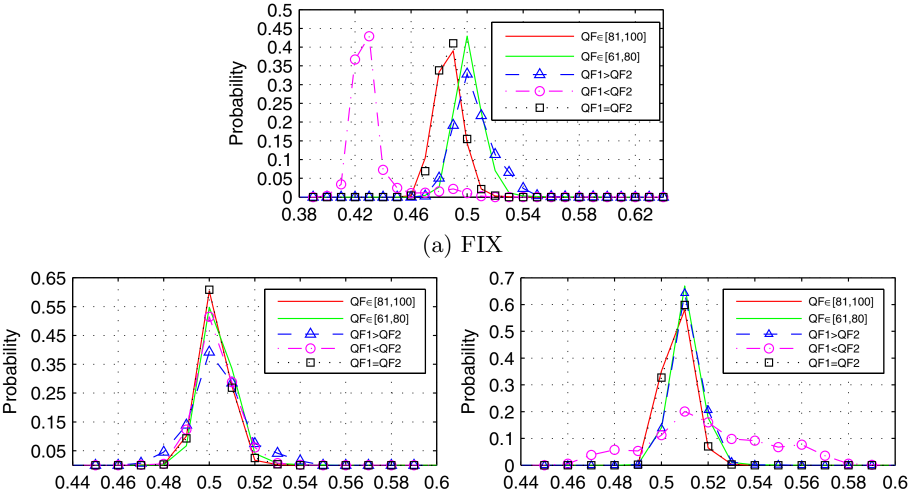
(c) PRO

Fig. 2. Distribution of $P(I)$ within 1338 images for cover image sets using different quality factors with coding pattern (a) FIX, (b) OPT and (c) PRO. Red solid with ○: QF=20, green dash: QF=40, blue solid with △: QF=60, cyan dash and dot: QF=70, yellow solid with ×: QF=80, magenta dot: QF=90, black solid with □: QF=95, brown dash with ◇: QF=100.

compressed image sets. The first type is to compress the images with the first QF (QF1) randomly and uniformly chosen from 81∼100 and re-compress it with a second QF (QF2) randomly and uniformly chosen from 61∼80. This would make QF2 smaller than QF1. The second type is to compress the images twice with QF1 from 61∼80 and QF2 from 81∼100. The third type is to compress the image twice with the same QF selected from 81∼100. All three coding patterns are applied in each type of the double JPEG compressed image and the distribution of $P(I)$ for each set are computed. We compare the single JPEG compressed image sets with the double JPEG compressed image sets for each coding pattern in Fig. 3.

Note that in the case of QF1>QF2, difference between single compressed and double compressed images can be found in coding pattern FIX and OPT, while in the case of QF1<QF2, large statistic deviation is made in coding pattern FIX and PRO. When QF1=QF2, the effect of double JPEG is not obvious in all three coding patterns.

The role of steganography. Now we will find out the real role of steganography in deviating the statistics of entropy. In order to eliminate the impact of QF, we set the same QF for the cover and stego images. Here we set QF=80 for illustration. JPHide, F5 and OutGuess are used. We define the embedding rate in terms of bpc (bits per non-zero DCT AC coefficient) as in [18]. Some of the



(a) FIX

(b) OPT

(c) PRO

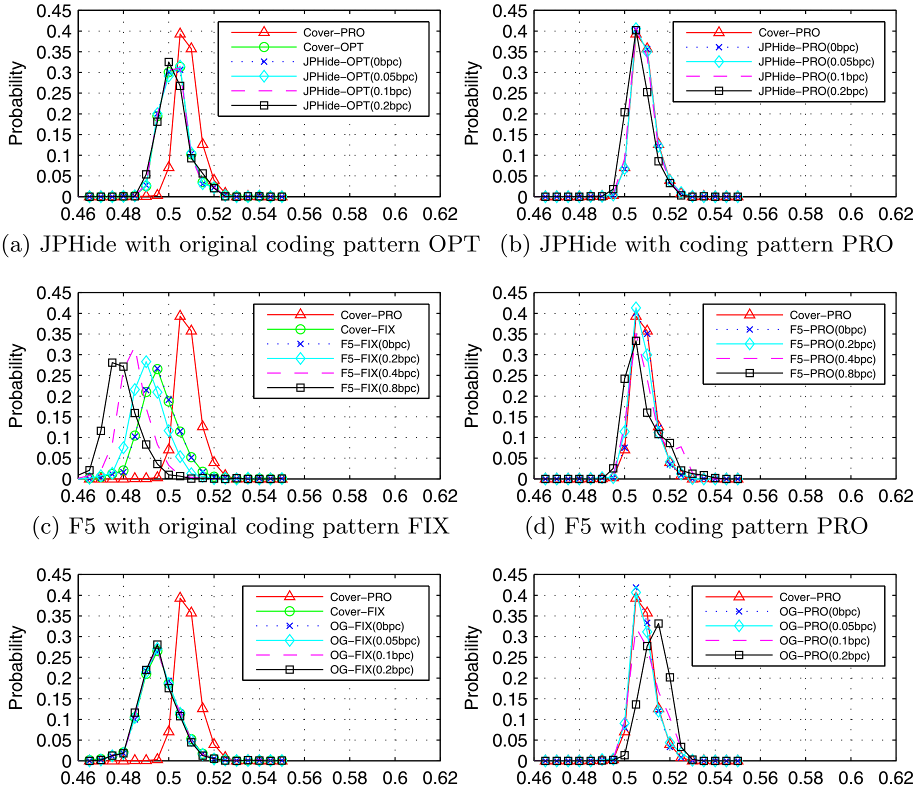
Fig. 3. Distribution of $P(I)$ within 1338 images for image sets with coding pattern (a) FIX, (b) OPT and (c) PRO. Red solid: single JPEG compression with QF$\in [81, 100]$, green solid: single JPEG compression with QF$\in$ $[61, 80]$, blue dash with $\triangle$: double JPEG compression with QF1$\in$ $[81, 100]$ and QF2$\in$ $[61, 80]$, magenta dash and dot with $\bigcirc$: double JPEG compression with QF1$\in$ $[61, 80]$ and QF2$\in$ $[81, 100]$, black dot with $\square$: double JPEG compression with QF1=QF2$\in$ $[81, 100]$.

images cannot be embedded more than 0.2 bpc by JPHide and OutGuess, or 0.8 bpc by F5, thus we embed 0.05, 0.1 and 0.2 bpc for JPHide and OutGuess, and 0.2, 0.4 and 0.8 bpc for F5, approximately corresponding to 25%, 50% and 100% of the maximal embedding capacity of each steganographic algorithm. We also embed a 0-byte[3] message for JPHide and F5, and 2-byte[4] for Outguess, in which case we denote as 0 bpc.

We start with using the coding pattern PRO for cover images as stated in [21]. The stego images are generated by the original coding pattern of the steganographic tools, that is OPT in JPHide, FIX in F5 and OutGuess. The distributions of $P(I)$ are computed and shown in Fig. 4(a), (c) and (e). We can observe that all the distribution curves of the stego can be separated well from the curves of the cover with PRO. It makes no exception for the stego images in 0 bpc. The curves of the cover image sets with the original coding pattern of the steganographic software are also shown for comparison. For JPHide and OutGuess, all the curves of stego are almost overlapped with the curves of the cover with the original coding pattern of the steganographic software, no matter what the

[3] The size of the secret message file for embedding is 0 byte. Nonetheless, the information of the file's size may be embedded (e.g. F5 embeds a 4-byte length header containing the file's size). This would make several bits changed.

[4] Since OutGuess does not support 0-byte embedding, we embed a 2-byte message instead.



(a) JPHide with original coding pattern OPT (b) JPHide with coding pattern PRO

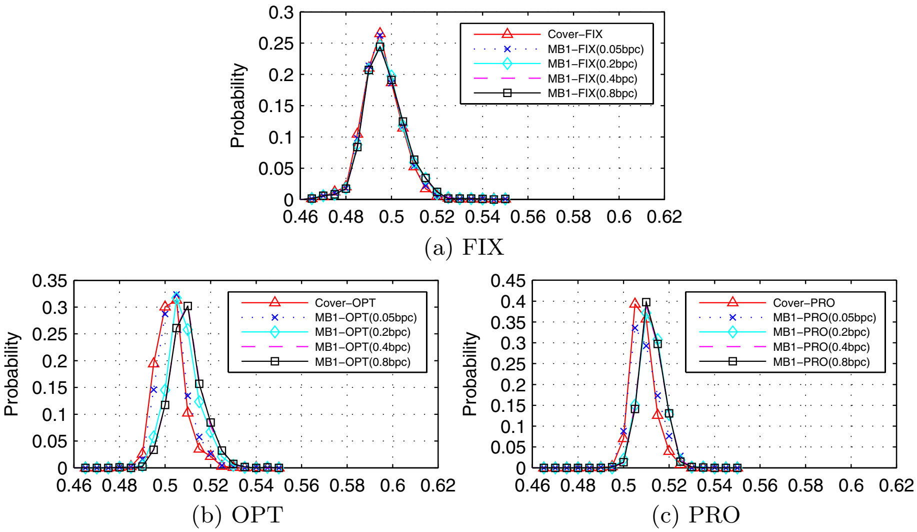
(c) F5 with original coding pattern FIX (d) F5 with coding pattern PRO

(e) OutGuess with original pattern FIX (f) OutGuess with coding pattern PRO

Fig. 4. Distribution of $P(I)$ within 1338 images for JPHide, F5 and OutGuess stego images with the original coding pattern of the steganographic tools and coding pattern PRO. Red solid with △: cover with the coding pattern PRO, green solid with ○: cover with the original coding pattern of the steganographic software, blue dot with ×: embedding rate of 0bpc, cyan solid with ◇: 0.05bpc for JPHide and OutGuess or 0.2bpc for F5, magenta dash: 0.1bpc for JPHide and OutGuess or 0.4bpc for F5, black solid with □: 0.2bpc for JPHide and OutGuess or 0.8bpc for F5.

embedding rate is. More specifically, the shape of the distribution curves of JPHide are like normal distribution centered at 0.5, which are very similar to Fig. 2 in [21]. For F5, the curves shift from right to left as embedding rate increases. We attribute this phenomenon to the "histogram shrinkage" by F5, in which case the number of zero DCT coefficients increases along with the embedding rate. For OutGuess, the number of zero coefficients is not changed and hence the curves do not shift much, just as the case of JPHide. But they are not centered at 0.5 as JPHide since FIX instead of OPT is used in OutGuess.

If the coding pattern PRO is used in both cover and stego images, the distribution curves of cover and stego images are very close, as it is illustrated in Fig. 4(b), (d) and (f). And the difference of $P(I)$ between stego and cover slightly



(a) FIX

(b) OPT

(c) PRO

Fig. 5. Distribution of $P(I)$ within 1338 images for MB1 with coding pattern: (a) FIX, (b) OPT and (c) PRO. Red solid with △: cover, blue dot with ×: embedding rate of 0.05bpc, cyan solid with ◇: 0.2bpc, magenta dash: 0.4bpc, black solid with □: 0.8bpc.

increases with the embedding rate. The curves of JPHide and F5 shift from right to left, indicating a decrease of "0" in the Huffman compressed stream, whereas the curves of OutGuess shift from left to right. It means that in PRO the extracted feature does not work consistently within these steganographic algorithms.

We also do an extra test for MB1, which is implemented in Matlab by its author and free of double JPEG compression issue, with the embedding rate of 0.05, 0.2, 0.4 and 0.8 bpc. There is an option in MB1 implementation to choose the coding pattern of stego images. Therefore, the same coding pattern is used for the cover and stego images. The distribution of $P(I)$ within 1338 images for each coding pattern are shown in Fig. 5. In coding pattern FIX, the distribution curves of the cover and stego images are hard to differentiate. In pattern OPT and PRO, the curves shift a little from left to right as embedding rate increases. The trend of the shifting distribution curves is very similar to the case of OutGuess, since MB1 does not modify the zero coefficients as well.

A Fisher linear discriminant based classifier with the extracted features as described in Sect. 2.2 is used to differentiate the cover and stego images in all the combination of patterns. The divided block size of the Huffman binary stream is set to 8 bytes. 200 cover images and 200 stego images are used for testing while the remaining images are used for training. We display the rate of detection accuracy of each classification with different embedding rate in Table 1. When the cover and stego images are using the same coding pattern, the detection rate is no better or just slightly better than random guessing, as we highlight the results in bold fonts. An outlier is F5 with both cover and stego in FIX, and

Table 1. Detection accuracy (in percentage) for JPHide, F5, OutGuess and MB1

JPHide		Stego								
		FIX			OPT			PRO		
		0bpc	0.1bpc	0.2bpc	0bpc	0.1bpc	0.2bpc	0bpc	0.1bpc	0.2bpc
Cover	FIX	**52.75**	**58**	**67.5**	65.5	66	70.75	85.75	85	86.25
	OPT	63.75	65.25	71.75	**53.5**	**53.75**	**59.5**	81.75	81.75	83.75
	PRO	83.25	81.75	86.5	83.25	82.25	81.5	**51.25**	**54.25**	**66.75**

F5		Stego								
		FIX			OPT			PRO		
		0bpc	0.4bpc	0.8bpc	0bpc	0.4bpc	0.8bpc	0bpc	0.4bpc	0.8bpc
Cover	FIX	**53.75**	**84**	**93.75**	62.5	60	62.5	83.75	79.75	76.5
	OPT	62.5	92.25	96.75	**49**	**57.25**	**66.5**	81.75	75.25	72.25
	PRO	84	99	99.25	81.25	83	88.5	**51**	**55.75**	**61.75**

OutGuess		Stego								
		FIX			OPT			PRO		
		0bpc	0.1bpc	0.2bpc	0bpc	0.1bpc	0.2bpc	0bpc	0.1bpc	0.2bpc
Cover	FIX	**51.75**	**53**	**62**	63	64.75	68	83	85.5	89.5
	OPT	62.75	63.25	66.5	**51**	**51.75**	**55.5**	81.5	72.25	88.75
	PRO	83.25	82.5	84.75	83.25	83.25	83.25	**56.5**	**57**	**70.75**

MB1		Stego								
		FIX			OPT			PRO		
		0.05bpc	0.4bpc	0.8bpc	0.05bpc	0.4bpc	0.8bpc	0.05bpc	0.4bpc	0.8bpc
Cover	FIX	**52.75**	**65.25**	**67**	67.25	77.5	78	85.75	88.75	88.5
	OPT	61.25	64.75	65.5	**55**	**73.25**	**74.25**	79.25	87.5	88.25
	PRO	81.5	81.75	82	81.75	84.25	82.75	**53.75**	**60.75**	**62.5**

the detection accuracy reaches 93.75% for fully embedding. Another exception is MB1 with both cover and stego in OPT, the detection accuracy reaches 74.25% with 0.8 bpc. When the cover and stego images are in different patterns, the classification accuracy rates, which more or less increase or decrease with the embedding rate, are always higher than 65% and some even exceed 80% with 0 bpc. It can be explained by the fact that the features are mainly influenced by different coding patterns and minor influenced by data embedding.

The role of image size and color. Finally we test for 1124 color images of size 1200×1600 taken by digital camera. Experiments on gray images converted from UCID are also conducted. Similar results are obtained in these two cases. Besides, none of the testing steganographic algorithms changes the image size and the number of color components, so we do not address the details here.

4 Discussions

All our observations are consist with the conclusion that the high detection rate achieved in [21] are mainly contributed by the different coding patterns between the cover and stego images. In fact, the entropy of Huffman compressed stream is related to Huffman tables, the frequency of occurrence of differential encoding codes (for DC coefficients) and the frequency of occurrence of run-length encoding codes (for AC coefficients). The number and location of zero coefficients greatly affect the run-length encoding codes. Hence, if the steganographic algorithm keeps the Huffman tables untouched and the number and location of zero coefficients unchanged, the entropy of the Huffman stream is not deviated much, as the case of OutGuess and MB1 with both of the cover and stego images in pattern FIX. F5 introduces more zero coefficients and thus it is detectable by the compressed frequency domain based steganalysis when the embedding rate exceeds 0.2 bpc even if the same pattern is used in both of the cover and the stego. Another phenomenon is that the peak of distribution curve is always near 0.5 when OPT is applied, as it is illustrated in Fig. 2(b), 3(b) and 4(a). It can be explained by the Huffman encoder making the entropy achieve its maximum by equalizing the number of "0" and "1" in the binary stream. As a result, it is more difficult to discriminate the stego from the cover in this coding pattern solely depend on the statistics of Huffman compressed stream.

The coding patterns confuse the compressed frequency domain based universal steganalyzer, which would make much more false alarm or miss detection because of the coding pattern conversion. It is very similar to the situation that the double JPEG compression would confuse the steganalyzer due to improper training and testing procedure. The confusion tests designed by Kharrazi et al. [28] and Shi et al. [29] have demonstrated the effect of double JPEG compression on the existing universal steganalytic schemes. What we learn from the investigation here is that one should make sure that the feature extracted from the cover and stego objects should truly reflect the data embedding itself.

Extracting features from the statistics of Huffman compressed stream is creative and showing a new way for steganalysis. It also may be applicable to digital forensics. But the entropy of the Huffman bitstream is certainly not a good feature because it is too coarse to describe the alteration of the DCT coefficients. Other statistics of Huffman compressed stream should be considered and may lead to an effective steganalyzer, since the run-length encoded data do reflect the relation between DCT coefficients in the zig-zag order within the 8×8 block.

From the viewing angle of steganographists, practical steganographic tool should avoid leaving obvious trace on the stego object. One example is that a seldom appeared comment "JPEG Encoder Copyright 1998, James R. Weeks and BioElectroMech." was inserted into the JPEG file header in the previous release of F5. Later, the author fixed this bug by adding a switch to change or remove the default comment. Another example is from the leading edge work by Westfeld [30]. Weaknesses are found in several steganographic tools due to message encryption and encoding. These traces will improve the performance of steganalyzer when the embedding rate is low.

5 Conclusions

In this paper, we carefully examine the factors which may have impact on the steganalyzer proposed in [21]. Our contributions are as follows:

1. Our observations show that the Avalanche Criterion [24] is misleading to explain the "powerfulness" of the "payload-independent" compressed frequency domain based steganalytic scheme. Instead, the real cause is the different coding pattern between cover and stego images.
2. Double JPEG compression artifacts also have influence on this steganalyzer.
3. If both cover and stego images are using the same coding pattern and the same quality factor, modifying a few DCT coefficients in small amplitude, especially not changing the number and the location of zero coefficients, cannot be detectable well by this steganalytic scheme.
4. We assert that the difference between cover and stego, other than being caused by the data embedding itself, should be avoided in both steganalysis and steganography. On the one hand, cover-stego difference, such as double JPEG compression artifacts or coding pattern conversion, may confuse a feature based universal steganalyzer. On the other hand, this implication can be used in designing a practical steganographic tool to enhance its security.

Acknowledgement. This work was supported by NSFC (90604008, 60633030), NSF of Guangdong(04205407), 973 Program (2006CB303104), China Scholarship Council.

References

1. http://www.nic.funet.fi/pub/crypt/steganography/jpeg-jsteg-v4.diff.gz
2. Latham, A.: Steganography: JPHIDE and JPSEEK (1999), http://linux01.gwdg.de/~alatham/stego.html
3. Westfeld, A.: F5 – A steganographic algorithm (high capacity despite better steganalysis). In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)
4. Provos, N.: Defending against statistical steganalysis. In: 10th USENIX Security Symposium (2001)
5. Sallee, P.: Model-based steganography. In: Kalker, T., Cox, I., Ro, Y.M. (eds.) IWDW 2003. LNCS, vol. 2939, pp. 154–167. Springer, Heidelberg (2004)
6. Sallee, P.: Model-based methods for steganography and steganalysis. International Journal of Image and Graphics 5, 167–189 (2005)
7. Fridrich, J., Goljan, M., Soukal, D.: Perturb quantization steganography using wet paper codes. In: Dittman, J., Fridrich, J. (eds.) Proceedings ACM Multimedia and Security Workshop, pp. 4–15. ACM Press, New York (2004)
8. Kim, Y., Duric, Z., Richards, D.: Modified matrix encoding for minimal distortion steganography. In: Information Hiding: 8th International Workshop (2006)
9. Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 61–75. Springer, Heidelberg (2000)
10. Provos, N., Honeyman, P.: Detecting steganographic content on the Internet. CITI Technical Report, pp. 01–11 (2001)
11. Westfeld, A.: Detecting low embedding rates. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 324–339. Springer, Heidelberg (2003)
12. Fridrich, J., Goljan, M., Hogea, D.: New methodology for breaking steganographic techniques for JPEGs. In: Delp III, E.J., Wong, P.W. (eds.) Proceedings SPIE, Electronic Imaging, Security, Steganography of Multimedia Contents V, San Jose, CA, January 20-24, vol. 5020, pp. 143–155 (2003)

13. Fridrich, J., Goljan, M., Hogea, D.: Steganalysis of JPEG images: Breaking the F5 algorithm. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 310–323. Springer, Heidelberg (2003)
14. Fridrich, J., Goljan, M., Hogea, D.: Attacking the OutGuess. In: Proceedings ACM Workshop on Multimedia and Security (2002)
15. Böhme, R., Westfeld, A.: Breaking Cauchy model-based JPEG steganography with first order statistic. In: Samarati, P., Ryan, P.Y.A., Gollmann, D., Molva, R. (eds.) ESORICS 2004. LNCS, vol. 3193, pp. 125–140. Springer, Heidelberg (2004)
16. Farid, H.: Detecting steganographic messages in digital images. Technical Report, TR2001-412, Dartmouth College, Computer Science (2001)
17. Avcibas, I., Memon, N., Sankur, B.: Image steganalysis with binary similarity measures. In: Proceedings of the IEEE International Conference on Image Processing, Rochester, New York, vol. 3, pp. 45–648 (2002)
18. Fridrich, J.: Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 67–81. Springer, Heidelberg (2004)
19. Shi, Y.Q., Chen, C., Chen, W.: A Markov process based approach to effective attacking JPEG steganography. In: 8th Information Hiding Workshop (2006)
20. Pevny, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. In: Proc. SPIE Electronic Imaging, Photonics West (2007)
21. Barbier, J., Filiol, E., Mayoura, K.: Universal JPEG steganalysis in the compressed frequency domain. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 253–267. Springer, Heidelberg (2006)
22. Barbier, J., Filiol, E., Mayoura, K.: New features for specific JPEG Steganalysis. Transactions on Engineering, Computing and Technology 16, 72–77 (2006)
23. Wallace, G.K.: The JPEG still picture compression standard. Communication of ACM 34(4), 30–44 (1991)
24. Feistel, H.: Cryptography and computer privacy. Scientific American 228(5), 15–23 (1973)
25. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley Interscience, New York (1991)
26. Schaefer, G., Stich, M.: UCID – An uncompressed colour image database. In: Proceedings SPIE, Storage and Retrieval Methods and Applications for Multimedia, San Jose, USA, pp. 472–480 (2004)
27. Sallee, P.: Matlab JPEG Toolbox (2003), `http://redwood.ucdavis.edu/phil/demos/jpegtbx/jpegtbx.htm`
28. Kharrazi, M., Sencar, H.T., Memon, N.: Benchmarking steganographic and steganalysis techniques. In: Delp III, E.J., Wong, P.W. (eds.) Proceedings SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII, January 16-20, vol. 5681, pp. 252–263 (2005)
29. Shi, Y.Q., Chen, C., Chen, W., Kaundinya, M.P.: Effect of recompression on attacking JPEG steganographic schemes – An experimental study. In: IEEE International Symposium on Circuits and Systems (2007)
30. Westfeld, A.: Steganalysis in the presence of weak cryptography and encoding. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 19–34. Springer, Heidelberg (2006)

Steganalysis Versus Splicing Detection

Yun Q. Shi[1], Chunhua Chen[1], Guorong Xuan[2], and Wei Su[1]

[1] New Jersey Institute of Technology, Newark, NJ, USA 07102
{shi,cc86}@njit.edu
[2] Tongji University, Shanghai, China 200092

Abstract. Aiming at detecting secret information hidden in a given image using steganographic tools, steganalysis has been of interest for years. In particular, universal steganalysis, not limited to attacking a specific steganographic tool, is of extensive interests due to its practicality. Recently, splicing detection, another important area in digital forensics has attracted increasing attention. Is there any relationship between steganalysis and splicing detection? Is it possible to apply universal steganalysis methodologies to splicing detection? In this paper, we address these intact and yet interesting questions. Our analysis and experiments have demonstrated that, on the one hand, steganography and splicing have different goals and strategies, hence, generally causing different statistical artifacts on images. However, on the other hand, both of them make the touched (stego or spliced) image different from the corresponding original (natural) image. Therefore, natural image model based on a set of carefully selected statistical features under the machine learning framework can be used for steganalysis and splicing detection. It is shown in this paper that some successful universal steganalytic schemes can make promising progress in splicing detection if applied properly. A more advanced natural image model developed from these state-of-the-art steganalysis methods is thereafter presented. Furthermore, a concrete implementation of the proposed model is applied to the Columbia Image Splicing Detection Evaluation Dataset, which has achieved an accuracy of 92%, indicating a significant advancement in splicing detection.

Keywords: Steganography, steganalysis, splicing detection, tampering detection, digital forensics, natural image model.

1 Introduction

Covert communications emerged at the very beginning of our human history. From human hair to papyrus, many articles were ever used as carriers to pass secret messages. Nowadays, digital images have become an important channel to bear a large amount of stego information. Moreover, the non-stationarity of images makes image steganography, the art and science of invisible communication, hard to break. Consequently, active resarch on image steganography has resulted in abundant publications. Accordingly, researchers have made efforts

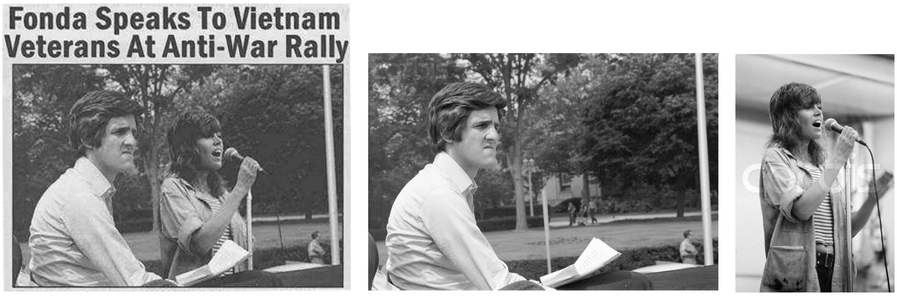
Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 158–172, 2008.

to develop image steganalysis schemes, which is to detect the very presence of hidden messages in a given image. According to the targeted steganographic tools, steganalysis methods may be classified into two categories, i.e., specific steganalysis and universal steganalysis [1]. Universal steganalysis aims to attack any steganographic tool, known or unknown in advance, while specific steganalysis is designed to detect some particular steganographic tool.

Images are not only able to carry stego information, but also able to convey fake information. Sometimes people maliciously manipulate images to forge a scene that actually never exists to mislead the observers on purpose. Taking a look at Fig. 1(a), one can see that in this picture, John Kerry stands by Jane Fonda, who was speaking to Vietnam veterans at an anti-war rally in 1970's. Appearing at the beginning of 2004, this picture enraged some Vietnam veterans. It was substantiated later to be a composite of Fig. 1(b) and Fig. 1(c).

As its name implies, image splicing is a simple process of cropping and pasting regions from the same or different images to form another image without post-processing such as edge smoothing. Image splicing is one of the simple and commonly used image tampering schemes and often used as an initial and important step for image tampering. With modern image processing techniques, image splicing is hardly caught by the human visual system (HVS). Therefore, image splicing detection is of fundamental importance in image tampering detection and is urgently called for. People need to tell if a given image is spliced or not without any a priori knowledge. In other words, the splicing detection should be blind in nature.

Since both image steganalysis and splicing detection are critical for digital forensics and information assurance, in this paper, we take the lead to conduct a comparison study between universal steganalysis and splicing detection. Analytical research and extensive experiments have demonstrated that, those approaches used in steganalysis can promisingly make progress in splicing detection applications if properly applied. This study leads to our conclusion that lessons learnt from steganalysis can shed light to splicing detection. Moreover, we raise and respond to some questions. Although the questions may have not been answered satisfactorily, our responses are expected to be thought-provoking.



(a) (b) (c)

Fig. 1. An altered image and two original authentic images [2]

The rest of this paper is organized as follows. A general comparison between image steganalysis and splicing detection, and the concept of natural image model are contained in Section 2. Machine learning framework is presented in Section 3. In Section 4, some successful universal steganalysis approaches are reviewed and natural image models used in these approaches are applied to splicing detection, and their performances are reported. In Section 5, a newly developed natural image model and one of its concrete versions designed for the available Columbia Image Splicing Detection Evaluation Dataset [3] are given to boost splicing detection capability. Further discussion on steganalysis and splicing detection with respect to natural image model is given in Section 6 and conclusion is drawn in Section 7, respectively.

2 Natural Image Model for Steganalysis and Splicing Detection

In this section, we first study the difference and then similarity between steganalysis and splicing detection. At the end, we prompt a question for discussion.

Obviously, steganalysis and splicing detection have different motivations and objectives. Specifically, steganalysis is to deter the secret communication, while splicing detection is to authenticate a given image by determining if it has been spliced. To look further, it is necessary to study their counterparts, i.e., steganography and splicing.

Steganography encodes information bits (possibly encrypted) and then embeds them into the cover image. Splicing is to replace one or more parts of a host image with fragment(s) from the same host image or other source images. Therefore, the statistical artifacts left with steganography are likely different from those caused by splicing.

Because steganography is to transmit secret data to the receiver, it is essential to hide the data in a digital image without drawing suspicion. Therefore, it is important for steganography to reduce the difference between a cover image and its stego image. Furthermore, steganography must ensure the embedded message "readable" to the receiver. However, the receiver of information conveyed by a spliced image is assumed to be human eyes instead of a machine, and therefore, it is critical to make the host image and the cut-and-pasted image fragment(s) look accordant. In other words, it is essential to make sure a spliced image perceived by human eyes looks like a non-spliced (natural) image, though the image content has been changed. That is, once the HVS observes a spliced image as non-spliced, this specific splicing operation is considered to be successful.

To resist steganalytic attacks, steganography often embeds data in a cover image as widely as possible, while splicing just touches part(s) of the host image. In this sense, steganography is more global while splicing is more local in nature. Consequently, the relative change between a cover image and its stego image is small. Differently, splicing generally changes the content of a host image. Therefore, the relative change between a host image and its spliced version is larger, more dramatic, and more local though.

From information assurance point of view, both image steganography and splicing are to disguise something: they try to make their touched images look like intact. In other words, they try to impress observers that these stego images or spliced images are natural images.

Since these stego images and spliced images are ever touched, the steganographic embedding and splicing operation shall cause disturbance on the smoothness, regularity, continuity, consistency, and/or periodicity of the images. Therefore, they do cause some statistical artifacts and are thus detectable using certain statistical attack approaches. As shown in the next a few sections, if we have a well designed natural image model, which can separate stego images (with hidden data and therefore unnatural) and spliced images (with inconsistent image fragment(s) and therefore unnatural) from natural images, both steganography and splicing are detectable by machine learning schemes.

How to measure the strength of change brought to the cover image (for steganography) or host image (for splicing) is an interesting question. We may have subjective measurement and objective measurement. The HVS generally provides subjective measurement. For steganography, we may use BPP (bits per pixel) and/or MSE (mean square error) (or PSNR, peak signal to noise ratio) as objective measures. For splicing, it is hard to measure the change strength. MSE (or PSNR) may be an objective measure candidate. Other possible measurements include the ratio of the perimeter of the cut-and-pasted image fragment(s) to that of the host image and the ratio of the area of the cut-and-pasted image fragment(s) to that of the host image. More study in this regard seems necessary.

3 Machine Learning for Steganalysis and Splicing Detection

As mentioned above, a well designed natural image model may separate stego images or spliced images from natural images. An image model consists of a set of features, or a feature vector, characterizing the statistical behavior of a given image. With a dataset comprising both natural images and non-natural (stego or spliced) images, universal steganalysis or splicing detection can be carried out under the machine learning framework. The dataset and classifier used in our experimental study are described in this section.

3.1 Image Database for Splicing Detection

The Columbia Image Splicing Detection Evaluation Dataset [3] is used in our experimental work by courtesy of digital video and multimedia lab (DVMM), Columbia University. This dataset is created by DVMM for benchmarking the blind passive image splicing detection algorithms. There are 933 authentic and 912 spliced image blocks in this dataset, each of which is of the same size 128×128. The following points are emphasized while creating the dataset: “content diversity”, “balanced authentic and spliced image block distribution”, and "localized detection", i.e., "the block is kept at a reasonable size 128×128 to ensure that sufficiently accurate statistical features can be estimated using the empirical data of each block". Furthermore, "the process of creating spliced images in reality is simulated with two types of operations - crop-and-paste along object boundaries vs. crop-and-paste of horizontal (or vertical) strips. Image objects and strips can be from the same image or two separate source images. Objects spliced together can be the same or different types - smooth or textured". Some authentic and spliced image blocks from [3] are shown in Fig. 2. This image block dataset is open for downloading. Although this dataset may be "greatly improved in many ways, and should be considered as a preliminary effort addressing the increasingly important topic of benchmarking", it has been constructed scientifically and is undoubtedly helpful to research in this area. For more information about [3], readers are referred to the technical report [4].

3.2 Classifier, Classification, and Result Analysis

Machine learning has been rather successfully used in quite many universal steganalysis [5, 6, 7, 8, 9, and 10] and some blind splicing detection [11, 12, 13, 14, 15, and 16]. From machine learning point of view, both universal steganalysis and blind/universal splicing detection consist of the following two parts: feature extraction, training and using trained classifier for testing.

In supervised machine learning, the support vector machine (SVM) has been widely used as the classifier. The SVM codes of Matlab are downloaded from [17]. Specifically, we use the radial basis function (RBF) kernel in our experiments.

In each experiment, randomly selected 5/6 of the authentic images and 5/6 of the spliced images are used to train a SVM classifier. The remaining 1/6 of these images are used to test the trained classifier. The receiver operating characteristic (ROC) curve is thereafter obtained to demonstrate the trained classifier's performance. Two numerical methods can also be used to show the classifier's performance. One is to calculate the area under the ROC curve (AUC). Readers are referred to [18] for more information of ROC and AUC. Another is to obtain true negative (TN) rate, true positive (TP) rate, and accuracy of the trained classifier. To reduce the effect caused by variation incurred in selection of training/testing images, we individually conduct each random experiment 20 times and report their arithmetic average.

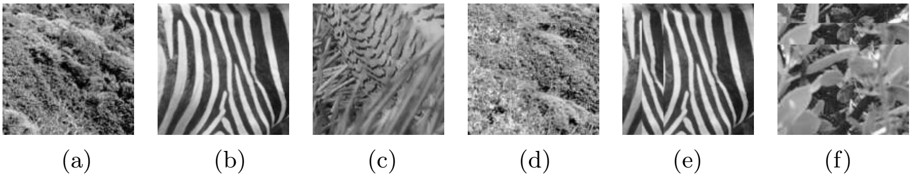

(a) (b) (c) (d) (e) (f)

Fig. 2. Some sample authentic images ((a), (b), and (c)) and spliced images ((d), (e), and (f)) used in this experimental work (source: [3])

4 Applying Natural Image Models Created in Universal Steganalysis to Splicing Detection

Since splicing detection emerges later than steganalysis, it is expected that splicing detection can "borrow" something from steganalysis. In this section, four natural image models, i.e., feature vectors used in universal steganalysis schemes are introduced. We then apply these models to Dataset [3] and use the trained SVM as classifier to detect spliced images. Experimental results are reported. The consideration of applying these four steganalysis schemes in our study is that they are applicable to non-JPEG images (image blocks in [3] are in BMP format). Furthermore, we do have these algorithms available in our past work. By no means it is meant that there are only these four universal steganalyzers.

4.1 Review: Some State-of-the-Art Universal Steganalysis Methods

In [5], the first four order statistical moments of wavelet coefficients and their prediction-errors of nine high frequency wavelet subbands (resulted from three-level wavelet decomposition) are used to form a 72-dimensional (72-D) feature vector for steganalysis. This image model (thereafter Lyu and Farid's) is shown in Fig. 3(a).

A steganalysis scheme utilizing statistical moments of characteristic functions of the test image, its prediction-error image, and all of their wavelet subbands is proposed in [6], where a 78-D feature vector is used as the image model. The block diagram of this image model (thereafter Shi et al.'s) is given in Fig. 3(b).

In [7], a steganalysis system based on 2-D Markov chain of thresholded prediction-error image is proposed. Image pixels are predicted with their neighboring pixels, and the prediction-error image is generated by subtracting the prediction value from the pixel value and then thresholded with a predefined threshold (set as 4). The empirical transition matrices of Markov chain along the horizontal, vertical, and main diagonal directions serve as features. The block diagram of this image model (thereafter Zou et al.'s) is shown in Fig. 3(c). Please note that those parts within dot lines in Fig. 3(c) are not used in [7]. Instead, these parts are used in [8].

Another effective universal steganalyzer is proposed in [9], which combines statistical moments of 1-D and 2-D characteristic functions extracted from the image pixel 2-D array and the multi-size block discrete cosine transform (MBDCT) 2-D arrays. This scheme greatly improves the capability of attacking steganographic methods applied to texture images, which has been shown to be a tough task [9]. In addition, this scheme can also be used as an effective universal steganalyzer for non-texture images. The block diagram of this method (thereafter Chen et al.'s) is shown in Fig. 3(d) and Fig. 3(e).

4.2 Applying These Natural Image Models to Splicing Detection

Although image steganography and image splicing result in different statistical artifacts, both of them cause disturbances on the correlation between image pixels. Since both artifacts deviate from natural image model, both steganography

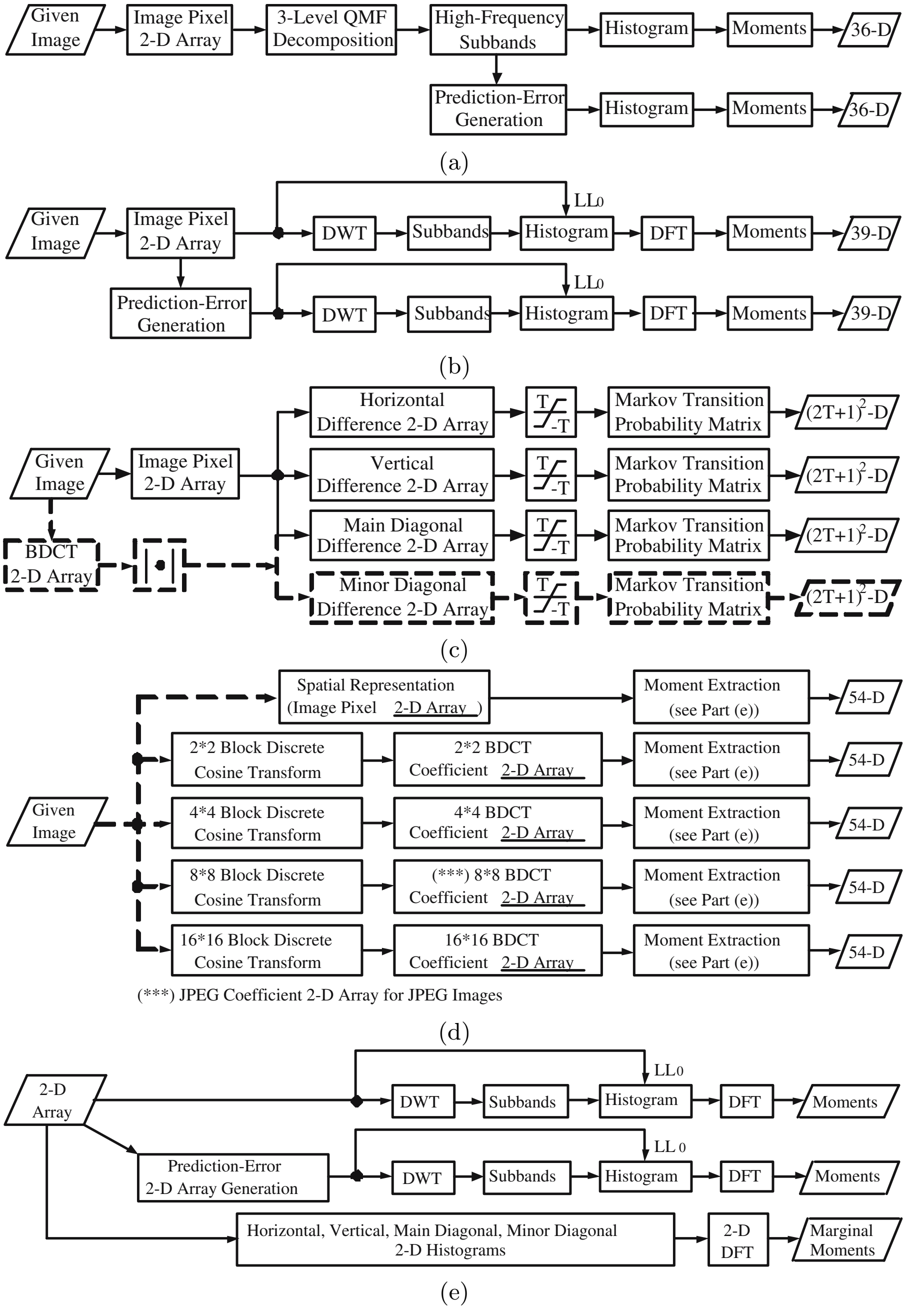

Given Image
Image Pixel 2-D Array
3-Level QMF Decomposition
High-Frequency Subbands
Prediction-Error Generation
Histogram
Moments
36-D
(a)
LL0
DWT
Subbands
DFT
39-D
(b)
Horizontal Difference 2-D Array
Vertical Difference 2-D Array
Main Diagonal Difference 2-D Array
Minor Diagonal Difference 2-D Array
BDCT 2-D Array
T
-T
Markov Transition Probability Matrix
$(2T+1)^2$-D
(c)
Spatial Representation (Image Pixel 2-D Array)
2*2 Block Discrete Cosine Transform
4*4 Block Discrete Cosine Transform
8*8 Block Discrete Cosine Transform
16*16 Block Discrete Cosine Transform
2*2 BDCT Coefficient 2-D Array
4*4 BDCT Coefficient 2-D Array
(***) 8*8 BDCT Coefficient 2-D Array
16*16 BDCT Coefficient 2-D Array
Moment Extraction (see Part (e))
54-D
(***) JPEG Coefficient 2-D Array for JPEG Images
(d)
2-D Array
Prediction-Error 2-D Array Generation
Horizontal, Vertical, Main Diagonal, Minor Diagonal 2-D Histograms
2-D DFT
Marginal Moments
(e)

Fig. 3. Feature generation block diagrams. (a) Lyu and Farid's method [5]; (b) Shi et al.'s method [6]; (c) Zou et al.'s method [7]; (d) General block diagram of Chen et al.'s method [9]; (e) Moment generation block diagram of [9].

Table 1. Detecting spliced images using four natural image models established in universal steganalysis (standard deviation in parentheses)

	Lyu and Farid's	Shi et al.'s	Zou et al.'s	Chen et al.'s
TN Rate	69.42% (4.34%)	76.07% (2.41%)	75.10% (3.47%)	86.32% (3.02%)
TP Rate	78.22% (4.14%)	75.59% (3.85%)	77.43% (4.24%)	87.83% (2.95%)
Accuracy	73.78% (2.01%)	75.83% (2.36%)	76.25% (3.20%)	87.07% (2.28%)
AUC	0.7892 (0.0206)	0.8162 (0.0247)	0.8232 (0.0244)	0.9292 (0.0170)

and splicing can be detected by using natural image model and machine learning framework. Therefore it is possible to apply natural image model established in steganalysis to splicing detection applications via machine learning methodology.

To evaluate the performances of those four natural image models applied to splicing detection, we directly use these methods to extract features from image blocks in Dataset [3], then train the SVM classifier, and finally use the trained classifier to detect spliced images. Experimental results are given in Table 1.

It is observed from Table 1 that all these four steganalysis approaches can be used to detect spliced images with quite good performances. Thus we have shown that the natural image models established in these steganalysis approaches can be straightforwardly applied to detection of spliced images.

5 An Advanced Natural Image Model to Boost Splicing Detection Capability

Steganalysis methods presented in Section 4 can be summarized as follows. An image model (i.e., a feature vector) is used to represent an image in the statistical sense. This model describes characteristics of different image classes, i.e., non-stego images and stego images. A classifier is then used to separate these classes. These natural image models have demonstrated their promising capability on splicing detection.

In this section, we present a more advanced natural image model to separate spliced images from natural images. In this model, two types of features are used: moments (of characteristic functions) based features and Markov process based features. This model further combines features derived from the image pixel 2-D array and features derived from the multi-size block discrete cosine transform (MBDCT) coefficient 2-D arrays.

5.1 General Framework of This Natural Image Model

This natural image model is shown in Fig. 4. That is, we consider the spatial representation of the given test image (i.e., the image pixel 2-D array, or referred to as image 2-D array for short in this paper) and extract statistical moments of characteristic functions and Markov transition probabilities from this image 2-D array. Furthermore, we apply block discrete cosine transform (BDCT) with a set of block sizes to the test image, resulting in a set of BDCT coefficient 2-D

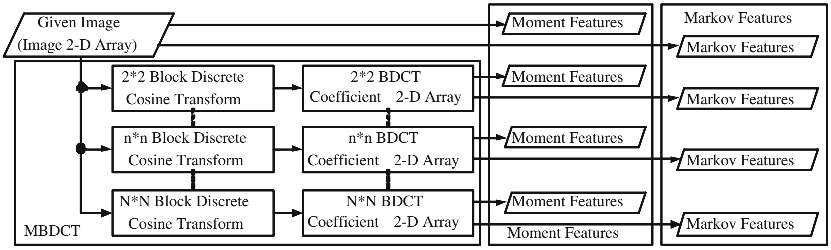


Fig. 4. A natural image model

arrays (i.e., MBDCT coefficient 2-D arrays, or MBDCT 2-D arrays for short in this paper). From these MBDCT 2-D arrays, we also extract statistical moments of characteristic functions and Markov transition probabilities as features.

5.2 Multi-size Block Discrete Cosine Transform 2-D Arrays

As in [9], BDCT with a set of block sizes is used in this novel natural image model. This is to utilize the complementary decorrelation capabilities contributed by BDCT's with various block sizes. The splicing procedure changes the frequency distribution of the host images. Coefficients of the BDCT's can reflect these changes. It is noted that the pattern in which the correlation changes is various and complicated due to different host images (contents), different cut-and-pasted image fragments (contents and shapes), and different possible positions of those image fragments. Therefore, one cannot expect to catch this kind of changes effectively by using only one single-block-size BDCT. With various block sizes, the MBDCT coefficients can perceive the changes of frequency distribution in a variety of ways and hence the spliced images can be distinguished from natural images with features extracted from these MBDCT 2-D arrays.

The application of an n×n BDCT is described as follows. Firstly, the given image is divided into non-overlapping nn blocks. Then, 2-D discrete cosine transform (DCT) is applied to each block independently. Finally, we obtain a 2-D array consisting of all the BDCT coefficients of all these blocks.

With each individual block size, we obtain one BDCT 2-D array. From the image 2-D array or each BDCT 2-D array, we can generate corresponding features.

5.3 Moment Based Features

As shown in Fig. 3(e), the moment feature extraction procedure is similar to that described in [9]. That is, moment based features consist of moments of 1-D characteristic functions (discrete Fourier transform (DFT) of the first-order histograms) of the image 2-D array or MBDCT 2-D array, its prediction-error 2-D array, and all of their wavelet subbands, and marginal moments of the 2-D characteristic functions (2-D DFT of the second-order histograms) of the image 2-D array or MBDCT 2-D array.

Wavelet analysis, prediction-error, characteristic function, and 2-D histogram are key points of moment based features. Wavelet analysis has been widely used in digital image processing applications owing to its superior multi-resolution and space-frequency analytical capability. Wavelet transform is suitable to catch transient or localized changes in spatial and frequency domains and hence good for splicing detection, which has demonstrated its efficiency in steganalysis applications [5, 6, 9]. A prediction-error 2-D array is used to reduce the influence caused by diversity of image contents and to simultaneously enhance the statistical artifacts introduced by splicing. Moments of characteristic functions have been shown more effective than moments of histograms [6]. By measuring the intensity change of pixels (or coefficients) with respect to their neighbors, 2-D histograms can reflect the statistical effects of splicing artifacts more efficiently than 1-D histograms, which consider one pixel (or coefficient) at a time.

5.4 Markov Process Based Features

Moment based features are a kind of measures reflecting the statistical changes caused by image splicing. Introduced in this section, Markov based features are another kind of measures able to reflect those statistical changes. Combining these two kinds of features, this novel natural image model can be enhanced to detect spliced images more effectively.

The Markov feature extraction procedure is similar to that described in [8]. At first, we form difference 2-D arrays (difference arrays for short) from the given image and/or coefficient 2-D array. These difference 2-D arrays are modeled by Markov process and then the transition probability matrix is calculated for each difference array. The entries of all the transition probability matrices are utilized as features to build up another part of the natural image model. In addition, a thresholding technique is developed to greatly reduce the dimensionality of the transition probability matrices, and hence the dimensionality of feature vectors, thus making the computational complexity manageable. The general block diagram of Markov feature extraction is shown in Fig. 3(c), in which four difference arrays are formed from an image 2-D array or BDCT 2-D array (rounded magnitude). This procedure is also similar to that used in [9].

By simply predicting an image pixel or a BDCT coefficient using its immediate neighbor, it is expected that the disturbances caused by splicing can be emphasized by the prediction-error, i.e., the difference between an element and its neighbor in an image or BDCT 2-D array.

For a given image or BDCT 2-D array (each of its element is the magnitude of a coefficient rounded into an integer), we can form four difference arrays, i.e., the horizontal, vertical, main diagonal, and minor diagonal difference 2-D arrays. The former three are called horizontal, vertical, and diagonal prediction-error image in [7]. The minor diagonal difference 2-D array is given by

$$F_m(i,j) = x(i,j+1) - x(i+1,j), \tag{1}$$

where $i \in [0, S_i - 2]$, $j \in [0, S_j - 2]$, S_i, S_j denote the 2-D array's dimensions in the horizontal direction and vertical direction, respectively, and $x(i,j)$ is either an image pixel value or a rounded magnitude of BDCT coefficient.

Proverbially, there exists correlation between pixels/coefficients in a 2-D array and a difference array. Therefore, we can use Markov process to model these difference arrays. According to the theory of random process, a transition probability matrix can characterize these Markov processes.

In our natural image model, we use the so-called one-step transition probability matrix to characterize those difference arrays [19]. To further reduce computational complexity, we resort to a thresholding technique, i.e., the value of an element in a difference array is represented by T or -T, respectively, if it is either larger than T or smaller than -T, resulting in a transition probability matrix of (2T+1)×(2T+1), the elements of which are given by

$$p\{F_h(i,j)=n|F_h(i+1,j)=m\}=\frac{\sum_{i,j}\delta(F_h(i+1,j)=m,F_h(i,j)=n)}{\sum_{i,j}\delta(F_h(i+1,j)=m)}, \tag{2}$$

$$p\{F_v(i,j)=n|F_j(i,j+1)=m\}=\frac{\sum_{i,j}\delta(F_v(i,j+1)=m,F_v(i,j)=n)}{\sum_{i,j}\delta(F_v(i,j+1)=m)}, \tag{3}$$

$$p\{F_d(i,j)=n|F_d(i+1,j+1)=m\}=\frac{\sum_{i,j}\delta(F_d(i+1,j+1)=m,F_d(i,j)=n)}{\sum_{i,j}\delta(F_d(i+1,j+1)=m)}, \tag{4}$$

$$p\{F_m(i+1,j)=n|F_m(i,j+1)=m\}=\frac{\sum_{i,j}\delta(F_m(i,j+1)=m,F_m(i+1,j)=n)}{\sum_{i,j}\delta(F_m(i,j+1)=m)}, \tag{5}$$

where $m, n \in \{-T, \cdots, 0, \cdots, T\}$, and

$$\delta(A = m, B = n) = \begin{cases} 1 & \text{if } A = m \ \& \ B = n \\ 0 & \text{otherwise} \end{cases} . \tag{6}$$

Note that all the elements of the transition probability matrix are used as features to form our natural image model.

5.5 A Concrete Implementation on Dataset [3]

Thus far in this section, we have presented the general framework of the novel natural image model. Here, we provide a concrete implementation of this model on dataset [3] with only 933 authentic image blocks and 912 spliced image blocks, each of which is 128×128 in size. These facts have determined this concrete implementation. The most obvious constraint is that the feature vector's dimensionality [20] is bonded by the limited number of image blocks in dataset [3].

In this implementation, the BDCT block sizes are selected as 2×2, 4×4, and 8×8, partially because this choice (power of 2) is of computational benefits in implementing BDCT. As a result, for each given test image, we have the image pixel 2-D array, and 2×2, 4×4, and 8×8 BDCT 2-D arrays.

Haar wavelet is applied due to its simplicity in implementation. Furthermore, to balance splicing detection capability and computational complexity, we only conduct one-level wavelet decomposition and thus have five subbands (the image 2-D array is considered as the LL0 subband). Compared to three-level DWT decomposition which results in 13 subbands, the number of subbands reduces by 62%. For each wavelet subband, we compute the first three order moments. When calculating marginal moments from the second-order histograms, we only use horizontal and vertical 2-D histograms and obtain the first three order marginal moments. As a result, 42 moment features are obtained from each given image 2-D array or BDCT 2-D array. Since we have one image pixel 2-D array and three derived BDCT 2-D arrays, 168 moment features for each given image are formed in this specific implementation.

To limit the dimensionality of Markov features, in this specific implementation, we only use two difference arrays (i.e., horizontal and vertical difference 2-D arrays) and we only generate these two types of difference arrays from the 8×8 BDCT 2-D array of the given image. With threshold T = 4, a transition probability matrix of dimensionality 9×9 = 81 is formed for each direction and the derived Markov features are of dimensionality 162 consequently.

To sum up, 330 features are used to represent this novel natural image model.

5.6 Experimental Results with This Concrete Implementation

The averaged ROC curve of 20 experiments applying our proposed natural image model to dataset [3] is given in Fig. 5, where the ROC curves of experiments reported in Section 4.2 are also included. This concrete implementation of the novel approach has achieved TN rate 91.52% (2.19%), TP rate 92.86% (1.72%), accuracy 92.18% (1.30%), and averaged AUC 0.9537 (0.0112) (standard deviation in parentheses). Compared to the accuracies 72%, 80%, and 82% achieved by [11, 12, and 13] over the same image dataset [3], this novel natural image model has made a significant advancement in splicing detection.

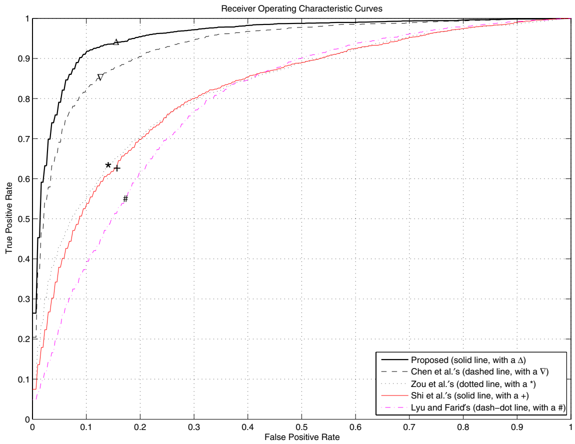


Fig. 5. The ROC curves of applying the natural image model to [3]

Table 2. Test results on real images (✓ - correct, × - wrong)

Experiment	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Fig. 1(a)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Fig. 1(b)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Fig. 1(c)	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓

5.7 Detecting Real Images

In Section 1, we have given an altered image and its two associated originals. We used the trained classifier resulted in the 20 random experiments mentioned in Section 5.6 to test these three images. The 20 test results are shown in Table 2. That is, among these 60 image-tests, 57 provide correct classification. These results are rather encouraging. No doubt, however, much more efforts need to be made for image splicing/tampering detection research in the future.

6 Discussion

We have presented a novel splicing detection scheme in Section 5. This scheme constructs a natural image model derived from some effective steganalysis approaches to distinguish spliced images from natural authentic images.

We can describe the relationship of steganalysis and splicing detection using Fig. 6. With a well designed natural image model, although stego images and spliced images occupy different regions in the high dimensional feature space, both of them are separable from natural images. In other words, if the image model is advanced enough, both image steganalysis and splicing detection can be successfully performed using this natural image model provided the corresponding and sufficient image datasets are available.

As discussed in this paper, splicing detection can learn lessons from steganalysis. Can steganalysis learn something from splicing detection? The answer seems positive. It is expected that both steganalysis and splicing detection can learn from each other and advance each other. It is noticed that the term "model" is sometimes confusing. This is because it has been used widely. An example is in image compression. There is a so-called model-based image compression methodology [21, 22]. An image of human face can be modeled by a number of triangle regions. When a person smiles or is angry, the shape of triangle changes.

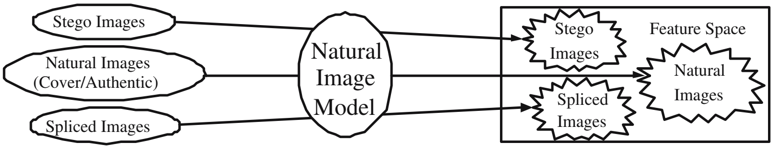


Fig. 6. Mapping images to feature space

Thus, if one can determine the coordinates of end points of these triangles, one can efficiently code the human face in videophone sequences. In this paper, we are talking about a statistical model, which represents the statistical behaviors of natural images, whose pixels are known to be correlated to some degree. With the machine learning framework working on a scientifically constructed image dataset, the statistical model is expected to be able to detect splicing.

7 Conclusion

In this paper, we have conducted a comparison study between steganalysis and splicing detection and proposed an advanced natural image model for splicing detection. We can summarize our study as follows.

1) Although different in target and application, image steganography and splicing do have some common aspects. One of such aspects is that both of them cause the touched images to deviate from natural images statistically, although the resultant statistical artifacts are different. Hence, steganography and splicing can be detected by natural image model (a set of natural image features).

2) Our experimental works have shown that applying the natural image models established in several state-of-the-art universal steganalysis schemes to splicing detection with a scientifically well-designed dataset for splicing detection has resulted in promising splicing detection capability.

3) An advanced natural image model, developed from the state-of-the-art steganalysis schemes, has been presented and applied to splicing detection. Its performance has demonstrated significant advancement in splicing detection.

4) Lessons learnt from steganalysis can be applied to splicing detection. It is expected that steganalysis can learn something from splicing detection as well.

Acknowledgement. Credits for using Columbia Image Splicing Detection Evaluation Dataset are given to DVMM Laboratory of Columbia University, CalPhotos Digital Library, and the photographers, whose names are listed in [4].

References

1. Kharrazi, M., Sencar, H.T., Memon, N.: Image Steganography: Concepts and Practice. In: Lecture Note Series, Institute for Mathematical Sciences, National University of Singapore (2004)
2. http://www.snopes.com/photos/politics/kerry2.asp
3. Columbia DVMM Research Lab: Columbia Image Splicing Detection Evaluation Dataset (2004), http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSpliced-DataSet/AuthSplicedDataSet.htm
4. http://www.ee.columbia.edu/dvmm/publications/04/TR_splicingDataSet_ttng.pdf
5. Lyu, S., Farid, H.: Detecting Hidden Messages Using Higher-order Statistics and Support Vector Machines. In: Information Hiding Workshop, Noordwijkerhout, Netherlands (2002)

6. 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. In: International Conference on Multimedia and Expo, Amsterdam, Netherlands (2005)
7. Zou, D., Shi, Y.Q., Su, W., Xuan, G.: Steganalysis Based on Markov Model of Thresholded Prediction-Error Image. In: International Conference on Multimedia and Expo, Toronto, ON, Canada (2006)
8. Shi, Y.Q., Chen, C., Chen, W.: A Markov Process Based Approach to Effective Attacking JEPG Steganography. In: Information Hiding Workshop, Old Town Alexandria, VA, USA (2006)
9. Chen, C., Shi, Y.Q., Xuan, G.: Steganalyzing Texture images. In: International Conference on Image Processing, St. Antonio, TX, USA (2007)
10. Fridrich, J.: Feature-based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes. In: Information Hiding Workshop, Toronto, ON, Canada (2004)
11. Ng, T.-T., Chang, S.-F., Sun, Q.: Blind Detection of Photomontage Using Higher Order Statistics. In: IEEE International Symposium on Circuits and Systems, Vancouver, BC, Canada (2004)
12. Fu, D., Shi, Y.Q., Su, W.: Detection of Image Splicing Based on Hilbert-Huang Transform and Moments of Characteristic Functions with Wavelet Decomposition. In: Shi, Y. Q., Jeon, B. (eds.) Digital Watermarking, Proceeding of 5th International Workshop on Digital Watermarking, Jeju Island, Korea (2006)
13. Chen, W., Shi, Y.Q., Su, W.: Image Splicing Detection Using 2-D Phase Congruency and Statistical Moments of Characteristic Function. In: Delp, E. J., Wong, P.W. (eds.) Security, Steganography and Watermarking of Multimedia Contents IX, Proceeding. of SPIE, San Jose, CA, USA (2007)
14. Farid, H.: A Picture Tells a Thousand Lies. New Scientist 179(2411), 38–41 (2003)
15. Ng, T.-T., Chang, S.-F.: A Model for Image Splicing. In: IEEE International Conference on Image Processing, Singapore (2004)
16. Bayram, S., Avcibas, I., Sankur, B., Memon, N.: Image Manipulation Detection. Journal of Electronic Imaging 15(4) (2006)
17. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines, `http://www.csie.ntu.edu.tw/cjlin/libsvm`
18. Fawcett, T.: Roc Graphs: Notes and Practical Considerations for Researchers, `http://home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf`
19. Leon-Garcia, A.: Probability and Random Processes for Electrical Engineering, 2nd edn. Addison-Wesley Publishing Company, Reading (1994)
20. Vapnik, V.N.: The Nature of Statistical Learning Theory, 2nd edn. Springer, Heidelberg (1999)
21. Shi, Y.Q., Sun, H.: Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards. CRC Press, Boca Raton (1999)
22. Forchheimer, R., Kronander, T.: Image Coding from Waveforms to Animation. IEEE Transactions on Acoustics, Speech and Signal Processing 37(12), 2008–2023 (1989)

Authenticating Binary Text Documents Using a Localising OMAC Watermark Robust to Printing and Scanning

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
c.culnane@surrey.ac.uk

Abstract. In this paper we propose a new authentication and localisation scheme to produce a watermark which can be embedded in a limited capacity binary text document and that will work in a print and scan environment. The scheme utilises Message Authentication Codes (MAC), specifically OMACs, which create a cryptographic fixed-length summary of a document. An OMAC must be truncated to a form part of our watermark and is used during authentication. The remainder of the watermark is used during localisation. We have created over 2,000,000 watermarks in controlled experiments to evaluate their ability to authenticate a document and localise any changes. In addition, we have embedded an authenticating watermark into seven different documents and authenticated them after printing and scanning.

1 Introduction

In our previous work [1] we increased the capacity available and robustness when embedding in binary text documents robust to printing and scanning. As we stated in our previous work, our goal is to increase the capacity sufficient to enable us to embed a meaningful authenticating watermark. We aim to only embed an authenticating watermark because it is not appropriate to embed a copyright based watermark since text documents can easily be OCR'd (Optical Character Recognition) or even manually re-typed to remove the watermark. There have been recent developments in using natural language watermarking [2], which change the sentence structure and sometimes actual words in order to create the watermark. These still suffer from easy attack, since someone is permitted to use an alternative synonym and this presents the problem of making changes to users' documents. In legal, medical and creative writing it would be unacceptable to make such changes.

The use of Message Authentication Codes (MACs) in authenticating text is not new. In [3] Villán et. al. proposed an embedding system providing far greater capacity, providing between 1 and 3 bits of capacity per character. However, the robustness of the scheme to printing and scanning relies on being able to use the same co-ordinate system for embedding and detection. It is unclear how the same co-ordinate system can be used after the distortions of printing and scanning.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 173–187, 2008.

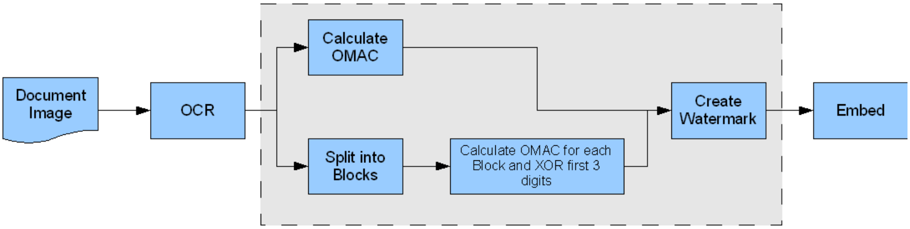


Fig. 1. Embedding Process

Creating an authenticating watermark for a binary text document robust to printing and scanning presents a unique set of problems. One problem is the distortion that occurs during printing and scanning and the other is what information can be represented in the limited capacity watermark and used as a basis of authentication. The way to counter the distortions is to abstract the image to a more constant value. This can be achieved by using OCR to convert the image of the document back into text. It does not matter if the position of the letter has moved a few pixels in some direction, or if a few pixels have been flipped from black to white or vice-versa. The OCR process is flexible enough to counter most of the distortions seen during printing and scanning. Whilst this can still be susceptible to some distortion, modern OCR packages allow user input in order to accurately correct any mistakes and to learn from those mistakes. The area most susceptible to distortion is the recognition of whitespace. It is for that reason that during the processing of the OCR output we strip all the whitespace. The removal of the whitespace will have no impact on the meaning of the document. Figure 1 is an overview of our authentication system and involves performing an OCR of the document as the first part of the embedding process.

The first contribution of the paper is the creation of an authenticating and localising watermark which can be embedded within a document with limited

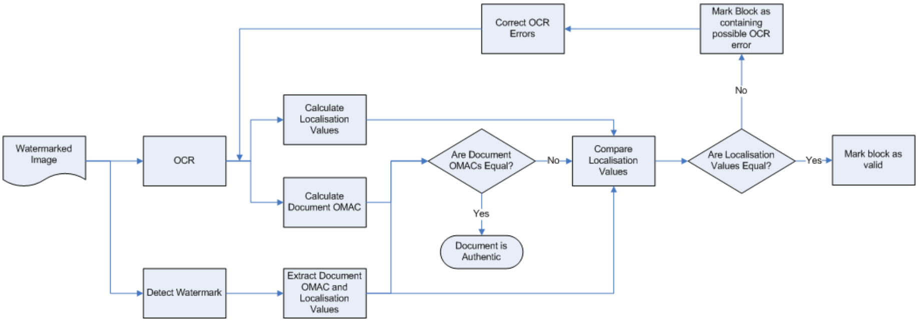

Correct OCR Errors
Mark Block as containing possible OCR error
No
Calculate Localisation Values
Watermarked Image
OCR
Are Document OMACs Equal?
No
Compare Localisation Values
Are Localisation Values Equal?
Yes
Mark block as valid
Calculate Document OMAC
Yes
Document is Authentic
Detect Watermark
Extract Document OMAC and Localisation Values

Fig. 2. Flow Chart for the Detection Process

capacity. This creation process is highlighted by the dotted box in Figure 1 and detailed in Section 3. Section 4 describes the structure of the watermark. Once a watermark is created it can be embedded into a document. This embedding process is summarised in Section 2 and based on our previous work.

The detection process is outlined in Figure 2. In contrast to traditional image authentication methods [4] that localise the document and then authenticate the localised blocks, we authenticate the document as a whole and subsequently attempt to localise any changes, and this will be an iterative process. We have to take this approach in order to comply with the capacity limitations.

Section 5 explains how we localise any errors during the detection process and this is another contribution of the paper. Section 6 summarises the experiments we conducted on authenticating documents and localising any errors.

2 Watermarking

In [1] we proposed a method for watermarking binary documents in a print and scan robust way. This was an extension of our work in [5] which was based on the work by Zou and Shi in [6]. Data is embedded by creating sets of word spaces, the whitespace between words, and making changes to the widths of these spaces to create a detectable difference between two sets. In [1] we further increased the capacity by embedding in a continuous line, instead of the inefficient line by line method used previously. Even with the improvements the average capacity is still only 98 bits. The combination of limited capacity and the need to handle printing and scanning are constraining factors in the identification of an authenticating watermark. However, in this paper we have succeeded to work within these constraints and developed a new authentication and localisation scheme.

3 What Is an OMAC

An OMAC is a One-key Message Authentication Code (MAC) equivalent to a CMAC (Cipher-based MAC) [7]. A MAC takes a message of arbitrary length and a secret key, it outputs a fixed length tag (MAC) that can be used to verify the integrity and authenticity of the message. Due to the MAC being created with a symmetric key, MACs do not offer any proof of non-repudiation as seen in digital signatures.

3.1 Alternative Authentication Methods

The traditional approach to authentication is to create a hash. A cryptographic hash function is a way of getting a fixed length output, called a digest, from an arbitrary length input in a manner whereby an attacker cannot create a second message that generates the same output or find a previously unseen message from a given digest. The two most common cryptographic hashes are SHA-1 and MD5. MD5 was shown to have security weaknesses many years ago and is rarely used in cryptography today. SHA-1 is still very much in use although recent research has demonstrated potential weaknesses, see [8]. We cannot use either algorithm because of the size constraints. MD5 produces 128 bit hashes, whilst SHA-1 produces 160 bit hashes. Both of which easily exceed our 100 bit capacity requirement. SHA-1 is a cryptographic hash function, but it is not equivalent to an OMAC or indeed a HMAC. The SHA-1 algorithm is well known and as such anyone can generate a SHA-1 hash of anything. Typically a SHA-1 hash is encrypted using the private key from an asymmetrical encryption system. When someone wants to authenticate the message they decrypt the block using the public key, re-calculate the hash and compare them. SHA-1 can be combined with an HMAC which is similar to an OMAC. The reasons for not using such a scheme are given in Section 3.2. We also considered the use of Cyclic Redundancy Checks (CRCs) although these are easily attacked and are used more for error detection as opposed to authentication. The advantage of CRCs was the range of output sizes they provided 16, 32 and 64 bits. However, their lack of security meant they were no comparison to an OMAC.

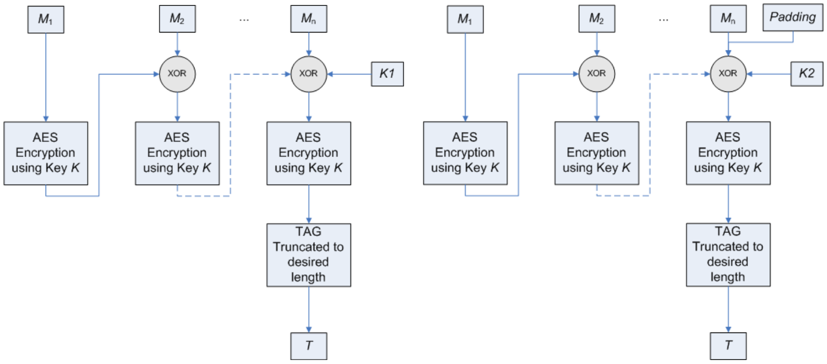
3.2 Selection of OMAC as Opposed to HMAC or SHA-1

We selected the OMAC algorithm for a number of reasons. Its equivalent, the CMAC, has been approved by NIST (National Institute of Standards and Technology, USA). There is a Java implementation of OMAC in the public domain and there are no known patent claims over the methods used. The OMAC can use any block based cipher and does not require the use of a cryptographic hashing algorithm. This allows us some flexibility in selection of the block based cipher. OMACs use a cipher block to encrypt the data as it is processed. This means someone without the key cannot create an OMAC and hence the key is required during authentication. There is never any decryption, thus allowing the OMAC to be truncated. This is not something that would be possible in a system that required decryption. The HMAC relies on a cryptographic hash function and typically uses SHA-1, however, some security concerns have been raised about SHA-1 [8] and truncating it may create greater weaknesses. The SHA-1 algorithm is due to be phased out in 2010 [9] and therefore its use in a new system is not prudent. Thus, an OMAC implementation based on an AES key is appropriate for our use because we will need the flexibility of truncation.

3.3 Overview of How OMACs Are Calculated

As stated above OMACs are based on the use of a block cipher, in our case AES. This block cipher requires a key K and operates on a block whose length is b, in this case 128 bits. Two subkeys $K1$ and $K2$ are derived from K. Both $K1$ and $K2$ are the same length as the block. During the process of generating the subkeys a pre-determined bit string is used. This bit string is based on the block length. In our case, 128 bit length, the bit string is defined as follows: $R_{128} = 0^{120}10000111$. The message from which to generate an OMAC is denoted as M and is $Mlen$ long, in terms of bits. The output of the OMAC generation is termed a TAG and denoted by T.

The full notation of the OMAC/CMAC algorithm is available from [7]. Our aim below is to provide an overview as opposed to a full definition.


M1
M2
...
Mn
XOR
XOR
K1
AES Encryption using Key K
AES Encryption using Key K
AES Encryption using Key K
TAG Truncated to desired length
T
M1
M2
...
Mn
Padding
XOR
XOR
K2
AES Encryption using Key K
AES Encryption using Key K
AES Encryption using Key K
TAG Truncated to desired length
T

(a) Last Message Block is Complete **(b)** Last Message Block is Incomplete

Fig. 3. Illustration of the OMAC/CMAC generation process (Images adapted from [7])

SubKey Generation. A string of '0' bits of the same length as the AES block b is encrypted using the AES key K, the output is denoted as L. If the most significant bit of L is a 0 the subkey $K1$ is equal to L bit shifted to the left by 1, otherwise $K1$ is equal to $L1$ bit shifted to the left by 1 and XOR'd with R_{128}. (Bit shifting involves removing the left most bit and appending a 0 zero bit to the right.) The generation of $K2$ is in essence the same except instead of using $L1$ we use $K1$. As such, if the most significant bit of $K1$ is a 0 the subkey $K2$ is equal to $K1$ bit shifted to the left by 1. Otherwise, $K2$ is equal to $K1$ bit shift to the left one and XOR'd with R_{128}.

MAC Generation. Given that we have already obtained K, $K1$ and $K2$, if the length of the message M is 0 then $n = 1$ otherwise $n = Ceiling(Mlen/b)$. The $Ceiling$ refers to the first integer not less than the expression. So $Ceiling(1.2)$ is 2 whilst the $Ceiling(3)$ is 3. The value of n is used to break the message up into blocks $M_1, ..., M_n$. The last block of the message is then processed as follows: if it is a complete block, contains 128 bits, M_n is made equal to M_n XOR $K1$. Otherwise M_n is made equal to M_n padded to the correct length XOR $K2$. Details of the padding are available in [7].

C_0 is defined as the output from the next step. It is initially set to be a bit string of zeros equivalent to the block length. For $i = 1$ to n, let C_i be equal to the encryption of C_{i-1} XOR M_i. In essence each message block is XOR'd with the output from the previous block's encryption and encrypted again. T is a truncation of the bit string to the desired length, taking the most significant bits.

Figure 3a and Figure 3b show the process in a graphical format. These images were adapted from [7].

4 Watermark Structure

Figure 4 is a diagram of the basic watermark structure. We use OMACs in two ways, firstly to authenticate the document and secondly to localise any errors

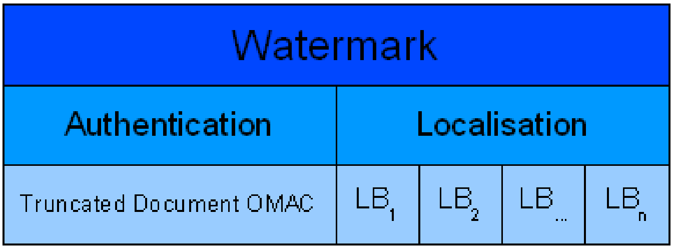


Fig. 4. Watermark Structure Diagram

found. We recognise that the OCR process may not be flawless and as such we want to be able to provide a guide as to where any error may have occurred. It is possible that OCR errors could occur during the embedding phase. We assume that should any such an errors occur it is likely they will re-occur during detection and therefore not cause a problem. An OMAC is generated for the whole document and then another OMAC is created for each localisation block. See Section 5 for more details on this. The principle of our system is that the secret key used in the OMAC will not be distributed, but held in a central location or shared between two trusted parties in advance. If someone wishes to verify a document they either email the document or bring it for verification. If the key was released it would allow anyone to change the document, generate another OMAC and embed it. Thus, defeating the security. It is important to record and potentially limit the number of times a document can be verified due to the truncation of the OMAC.

As can be seen from Figure 4 the watermark capacity is split in two. One half for authentication and one half for localisation. The authentication half contains the OMAC calculated over the entire document truncated to fit into the available size. The localisation half is further split into localisation blocks, each of which is 4 bits long.

4.1 Why Split the Watermark in Half?

How the watermark is divided needs to decided in advance and the same for all watermarks created within that system. We do not have the capacity within the watermark to embed structural information such as where the division between authentication and localisation takes place. We choose to split the watermark in half as we considered localising any errors as important as authentication and are willing to accept the reduced security of a smaller truncated watermark. The impact of reducing the localisation size is that localisation blocks become larger and it becomes more difficult for a user to find and correct any OCR errors.

5 Localisation

As was mentioned above the OCR may not be flawless and there may be a need to correct any OCR errors if the authentication fails. In order to do this we need to localise where the error is, to allow the user to quickly find and correct any errors. The biggest challenge to localising authentication errors is the small amount of capacity we have available. As was discussed in Section 4 we devote half our capacity to localisation. This is still a very small amount, 50 bits over approximately 50 lines. The method we propose is a balance between successful localisation and reduced space.

5.1 Division of the Document into Localisation Blocks

Once the document image has been OCR'd we have a text document containing the lines of OCR'd text. We read this document in and create an array of strings, each item in the array containing one line. We then count the number of lines we have and the available capacity. We allocate 4 bits per localisation block. We experimented with allocating 5 bits but there was no noticeable improvement in results. We divide the available capacity by 4 (the number of bits per block) and this gives us the number of localisation blocks we can create. We then divided the number of lines in the document by the number localisation blocks, giving us how many lines will be in each block. As we process each line during the OMAC calculation we also create the localisation blocks and calculate an OMAC per localisation block. We can express this process using the following equations:

$$L_b = (W_c/2)/4$$
$$L_s = D_l/L_b$$

where D_l is the number of lines in the document, W_c is the capacity of the watermark, L_b is the number of localisation blocks, and L_s is the number of lines in a localisation block.

Obviously, we cannot store the whole OMAC for each localisation block, since we only have 4 bits of capacity. We could just choose to store one of the OMAC digits, for example the first or last digit. However, this would not be the most efficient use of the 4 bits of capacity. It also would be prone to errors, whereby two different OMACs are treated as the same. We therefore decided to XOR the first three digits. We experimented with XORing all the digits but again saw no improvement. This is most likely due to the inherent limits of the XOR operation.

5.2 Example

Let $D_l = 50$ and $W_c = 100$, then we can calculate the following:

$$L_b = (100/2)/4 = 12$$
$$L_s = 50/12 = 5$$

Note that the rounding on the number of blocks is always down. It is better to have unused bits than not enough space to embed the final localisation block data. Also the rounding is always up on the block size calculation. This is to ensure all lines are part of some block.

The following is an example of an OMAC value for a localisation block: 11624770511598628777770661211885184331 (and corresponding to 128 bits and we only have 4 bits available). If we then take the first three digits and XOR them we get the following (we XOR the first two, then the XOR the result of that with the third):

Table 1. OMAC values to be XOR'd and XOR value

Decimal	Binary
1	0001
1	0001
6	0110
	0110

Suppose we change one character in the first block (an 's' to a 't') we then obtain the following OMAC value for that block: 3782238743673572791382206625495005089. This gives use the following XOR output:

Table 2. OMAC values to be XOR'd and XOR value after character change

Decimal	Binary
3	0011
7	0111
8	1000
	1100

Thus, during detection when comparing the two XOR values, they will not be equal and the block will be marked as having changed.

5.3 Localisation of a Document

Before the watermark is created an XOR value for each localisation block is calculated. These are then combined into the complete watermark, using the structure discussed in Section 4. During the detection an XOR value of the OMAC for each localisation block is calculated (see Calculation of Localisation Values box in Figure 2). The values stored in the watermark are retrieved and are compared with the computed value. If they are not the same the block is marked as potentially containing an error. The system is not perfect, since it is possible that a block cannot be localised, this can occur because there is not a 1:1 mapping between possible digit combinations and XOR values. This is due to the capacity limitations. However, a localisation block should never be indicated as changed when it has not changed. Also the OMAC for the entire document will still function correctly even if the localisation fails.

6 Experimentation and Analysis

Our experiments were done in three stages: we examined the theoretical limits of using a truncated OMAC, conducted a number of controlled experiments, and performed print and scan tests.

6.1 Theoretical Analysis

How secure is the authentication? The level of security provided by an OMAC has been proved using cryptographic proofs in [10], it is beyond the scope of this paper to re-examine those proofs. However, those proofs are based on using a full length OMAC. As we stated in Section 4 we use a truncated OMAC and hence need to evaluate the impact of such a reduction in the security.

It should be noted that a weakness of the OMAC is if an attacker has access to difference OMACs generated with the same key as was shown in [11]. We could protect against this attack by using a different key for each OMAC.

Birthday Paradox and Birthday Attack. The birthday paradox is part of probability theory, it states that given a group of 23 or more people the probability of two of them having the same birthday is at least 50%. This is important as it forms the basis of the birthday attack [12]. The birthday attack is a cryptographic attack that determines how many times we will need to brute force attack a hash before we find a collision. In our case a collision is when two different documents produce the same OMAC.

If our OMAC has a output length of m bits then to a collision we will need to calculate $2^{m/2}$ different OMACs in order to find a collision. This allows us the calculate a quantitive measure of how secure our OMAC is depending on the length of the truncated OMAC. For example, with a length of 50 bits we would need to calculate $2^{50/2}$ OMACs in order to have a high probability of finding a collision. However, a collision may be found within fewer calculations.

6.2 Controlled Experiments

As part of our evaluation of the authentication and localisation schemes we conducted five controlled experiments outside of the print and scan environment. These involved making controlled changes to the text in order to evaluate the effectiveness of the authentication and localisation schemes. The experiments resulted in over 2,000,000 tests.

In each of the experiments we took the equivalent of a page of text and created a watermark for it. We then made controlled changes to the document, recalculated the watermark and compared the OMACs. We expected the OMACs to be different and the aim was to localise the changes. We made the following five types of changes:

1. Single Character Change
2. Dual Character Change with Same Character Change
3. Dual Character Change with Same Size Change

Table 3. Experimentation Results

	Number Hash Values Calculated	Collisions	Localisation Errors
Experiment Name			
Single Character Change	424080	0	27648
Dual Character Same Character Change	424080	0	82625
Dual Character Same Size Change	424080	0	79928
Dual Character Random Change	424080	0	83759
Dual Character Same Size Oposite Change	424080	0	80887
Total	2120400	0	354847

4. Dual Character Change with Random Character Change
5. Dual Character Change with Opposite Size Change

Further details of each experiment and their results are given below. When making changes to characters we used characters with ASCII values between 33 and 126 (inclusive). This is basically the range of human readable characters, excluding whitespace. The sample document consisted of 19 pangrams repeated to form a document consisting of 4560 characters (excluding whitespace). We choose pangrams to ensure we had a range of all characters. Therefore, 424080 watermarks were calculated, (126-33)*4560, when conducting each of the following experiments.

Single Character Change. This experiment is designed to mimic the small changes we are likely to see during OCRing process. It involves iterating through each character in the document and changing it to each one of the characters within the predetermined range, as shown in Figure 5. After each change the watermark is re-calculated. It is unlikely that someone could meaningfully attack a document based on changing just one character, unless it was a number. Table 3 shows that during this experiment there were no collisions. However, there were a number of localisation errors. The number of localisation errors should be viewed in the context of having performed over 424000 tests. The error was correctly localised in over 93% of the cases. The failure of the localisation method in no way affects the authentication process.

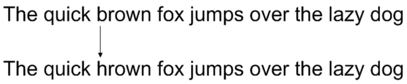

The quick brown fox jumps over the lazy dog
The quick hrown fox jumps over the lazy dog

Fig. 5. Single Character Change Diagram

Dual Character Change with Same Character Change. This experiment is to test how well the method works with two seemingly unconnected changes. This experiment also involves iterating through each character in the document and changing it to each of the characters in the predetermined range. For each change we also randomly select another character from the document and change that to the same character. For example both the characters ‘b’ and ‘o’ change to ‘h’ in Figure 6. This also allows us to further test that the localisation method can pick up changes in two different locations. Table 3 shows that during this experiment there were no collisions. There were more localisation errors, but this is due to more blocks being changed and thus the chance of localisation errors increases. Even so the localisation method worked in over 80% of the tests.

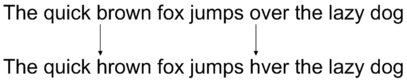

The quick brown fox jumps over the lazy dog
The quick hrown fox jumps hver the lazy dog

Fig. 6. Dual Character Same Change Diagram

Dual Character Change with Same Size Change. In this experiment we iterate through the characters in the document changing it to each of the possible characters. We calculate the size of the change then pick another character at random and make the same size change to it. For example in Figure 7 both changes increase by six. It is possible that in some cases the character that is picked at random may become a non-human readable character since it will be outside of the ASCII range. We choose to allow this since Java uses Unicode to represent characters and as such the value will not be outside of the acceptable range of characters. Again, Table 3 shows that there were no collisions. The number of localisation errors was similar in size to the one found in the Dual Character Change with Same Character Change.

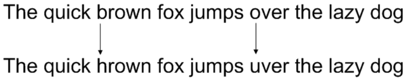

The quick brown fox jumps over the lazy dog
The quick hrown fox jumps uver the lazy dog

Fig. 7. Dual Character Same Size Change Diagram

Dual Character Change with Random Character Change. This experiment is an extension of the second experiment with more randomness. Figure 8 shows an example of the type of change undertaken. We iterate through each of the characters and change it to each of the possible characters. We then pick another character at random and change that to another random character. We can see from Table 3 that no collisions occurred and the localisation errors remained in the same region as the other similar experiments.

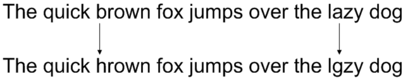

The quick brown fox jumps over the lazy dog
The quick hrown fox jumps over the lgzy dog

Fig. 8. Dual Character Random Change Diagram

Dual Character Change with Opposite Size Change. This is designed to test a deliberate attack on the system. By making an opposite change it means that the byte values will remain balanced. Since one is increased and another is decreased by the same amount. This experiment follows the same lines as the one described in Dual Character Change with Same Size Change except we make the opposite change to the second character. Figure 9 illustrates the type of change we make. Table 3 shows that again there were no collisions. The number of localisation errors was inline with the other experiments. The localisation errors are not critical, since they are used as a way of helping the user to correct any OCR errors or to see where any attack may have taken place.

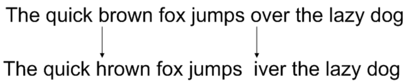

The quick brown fox jumps over the lazy dog
The quick hrown fox jumps iver the lazy dog

Fig. 9. Dual Character Opposite Size Change Diagram

6.3 Print and Scan Experiments

The above experiments demonstrated that a watermark based on OMACs successfully works given our capacity constraints. We also conducted experiments to evaluate the new method in a print and scan environment by embedding authenticating watermarks into seven documents, each formatted using a different font. The document images were at 300dpi and the font size in each case was 12pt. The fonts tested were: Arial, Arial Narrow, Comic Sans, MS Sans Serif, Tahoma, Times New Roman and Verdana. These were the same fonts as were used in our previous work [1]. The documents were printed and scanned on a HP PSC 2110 and we used ReadIris Pro 9 to do the OCRing. We could not include the Courier New font because it could not be OCR'd correctly. This was due to the large whitespaces in the document causing it to be fragmented into columns by the OCR Package.

Table 4 shows the results from the printing and scanning experiment. The first, second and third pass reflect the iterative nature of the process. Once a document has been authenticated we do not authenticate it again in any further passes. The values in the cells indicate whether the document was authenticated (**Valid**), contained OCR errors (e.g. 1 OCR) or contained noise (Noise). The noise is a result of the printing and scanning process and as is mentioned in [1] needs to be removed manually.

Table 4. Print and Scan Results

				Pass		
Font	W_c	D_l	L_s	**First**	**Second**	**Third**
Arial	105	50	4	Noise	**Valid**	
Arial Narrow	122	48	3	1 OCR	**Valid**	
Comic Sans	82	50	5	1 OCR	**Valid**	
MS Sans Serif	108	51	4	Noise	2 OCR	**Valid**
Tahoma	100	47	4	**Valid**		
Times New Roman	114	49	4	3 OCR	1 OCR	Not Valid
Verdana	87	47	5	1 OCR	**Valid**	

The Tahoma document was authenticated without the need for any corrections. The MS Sans Serif document and the Arial document both contained noise that had to be manually removed in order to correctly recover the watermark. Once the noise had been removed the Arial document authenticated correctly. The MS Sans Serif document contained two OCR errors each error was correctly localised to four lines in the 51 line document. After correcting those errors the document authenticated.

Comic Sans, Arial Narrow and Verdana documents all contained OCR errors that were correctly localised to within L_s lines, as shown in Table 4. Once these OCR errors had been corrected, they too authenticated. Even though all the OCR errors were corrected the Times New Roman document could not be validated due to a watermarking error. One bit was flipped due to distortion and that resulted in an incorrect OMAC being extracted from the watermark. This prevented the document from authenticating. This highlights the fact that should the watermark be lost or damaged it operates in a fail-safe manner and does not authenticate the document. These results show that it is an iterative process to remove the OCR errors before a document can be authenticated. Further work is needed to counter the problems caused by detection errors.

7 Conclusion

In this paper we have introduced an authentication and localisation scheme that can be embedded in a limited capacity watermark. The watermarks are typically less than 100 bits. We authenticate the document as a whole and cannot authenticate individual localisation blocks because this would require storing the entire OMAC for each localisation block which would exceed our capacity limitations. The ability to localise errors to within at most 5 lines allows the scheme to handle the distortions introduced during printing and scanning.

capacity is split in two, half for authentication and half for localisation. The larger the capacity in the document the greater the level of authentication security and the fewer the lines in the localisation block. The size of the watermark is calculated dynamically and therefore is adaptive to the capacity of the document. We could further emphasise security or localisation by changing how we split the watermark structure. For a more secure authentication the localisation can be reduced and a larger document OMAC embedded. Likewise if localisation is more important then authentication can be reduced.

We conducted over 2,000,000 million controlled tests and successfully authenticated in all of them. It should be noted that this to be expected since we would need to conduct $2^{50/2}$ tests in order to have a high probability of finding a collision. In over 83% of cases the changes were correctly localised. We have demonstrated that the authentication of the document is still sound, even with a truncated OMAC. We also demonstrated that even if the localisation fails it in no way affects the authentication.

We used the same authentication and localisation scheme in a print and scan environment. We successfully localised all of the documents and authenticated six out of the seven. The one failure was caused by a watermark detection error.

8 Future Work

As we discussed above the watermark is split in a 50-50 manner, however, that may not be ideal for specific applications. It was sufficient as a proof of concept, but we would like to develop the structure further so greater emphasis can be placed on authentication or localisation depending on the requirements of the user.

There are a number of different potential applications for the scheme we have proposed. The two most significant are authenticated archiving and authenticated message transfer. In authenticated archiving, the goal is to be able to archive the document and then at a later date authenticate that it is genuine. This is applicable to legal and medical situations. In this scenario there would be a trusted third party that would create the watermark and provide authentication services. None of the interested parties would have access to the AES key, therefore providing protection from malicious changes.

Authenticated message transfer aims to provide an authenticated link between two parties who trust each other. This is applicable to situations where printed text documents need to be sent through potentially hostile locations. It is particularly relevant to Emergency Procedure Plans and Work Procedures. These are often sent via couriers and there is currently no way of authenticating that the document received has not been changed. The two parties would need to agree on a shared AES key in advance. This could be done using established protocols currently used for generating session keys. Alternatively, the AES Cipher could be replaced with a Password Based Encryption (PBE) Cipher which generates the encryption key from a password. The two users could then exchange the password.

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

References

1. Culnane, C., Treharne, H., Ho, A.T.S.: Improving multi-set formatted binary text watermarking using continuous line embedding. In: IEEE International Conference on Innovative Computing, Information and Control (ICICIC 2007) (to appear, 2007)
2. Topkara, M., Topkara, U., Atallah, M.J.: Words are not enough: sentence level natural language watermarking. In: MCPS 2006: Proceedings of the 4th ACM international workshop on Contents protection and security, pp. 37–46. ACM Press, New York (2006)

3. Villán, R., Voloshynovskiy, S., Koval, O., Deguillaume, F., Pun, T.: Tamperproofing of electronic and printed text documents via robust hashing and data-hiding. In: Delp III, E.J., Wong, P.W. (eds.) Security, Steganography, and Watermarking of Multimedia Contents IX, February 2007. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, vol. 6505, p. 65051T (2007)
4. Miller, M.L., Cox, I.J., Bloom, J.A.: Digital Watermarking, 1st edn. Morgan Kaufmann, San Francisco (2002)
5. Culnane, C., Treharne, H., Ho, A.T.S.: A new multi-set modulation technique for increasing hiding capacity of binary watermark for print and scan processes. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 96–110. Springer, Heidelberg (2006)
6. Zou, D., Shi, Y.Q.: Formatted text document data hiding robust to printing, copying and scanning. In: IEEE International Symposium on Circuits and Systems (ISCAS 2005) (2005)
7. Dworkin, M.: Recommendation for block cipher modes of operation: The cmac mode for authentication. Special Publication 800-38B, NIST (May 2005)
8. Wang, X., Yin, Y.L., Yu, H.: Finding Collisions in the Full SHA-1. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621. Springer, Heidelberg (2005)
9. NIST. Nist brief Comments on Recent Cryptographic Attacks on Secure Hashing Functions and the Continued Security Provided by SHA-1. Internet (August 2004), `http://csrc.nist.gov/hash_standards_comments.pdf`
10. Iwata, T., Kurosawa, K.: Omac: One-Key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129–153. Springer, Heidelberg (2003)
11. Mitchell, C.J.: Partial key recovery attacks on XCBC, TMAC and OMAC. In: Smart, N. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796. Springer, Heidelberg (2005)
12. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc., New York (1996)

A General Data Hiding Framework and Multi-level Signature for Binary Images

Huijuan Yang and Alex C. Kot

School of Electrical & Electronic Engineering
Nanyang Technological University, Singapore 639798
{ehjyang,eackot}@ntu.edu.sg

Abstract. This paper reviews techniques for binary images data hiding and watermarking and proposes a general framework for authentication purposes. Finally, a multi-level signature scheme is presented, which is effective to detect malicious tamperings conducted to an image.

1 An Overview

Watermarking or data hiding on binary images can be classified as *fragile* and *robust* watermarks depending on different applications. It can be classified as *spatial* and *transform* domain techniques based on the domain where data hiding occurs. In addition, it can be classified as *pixel-flipping* and *object*-based techniques depending on whether the watermark embedding is done in pixel level or object level. Specifically, a pixel-flipping technique encodes a message bit by flipping a pixel from white to black or vice versa, whereas an object-based technique encodes a message bit by modulating the spaces or features among characters and words. Classification of the data hiding/watermarking techniques for binary images is shown in Fig. 1.

Most data hiding or watermarking techniques are for gray or color images due to the wide range of pixel values. Slightly perturbing the pixel values does not

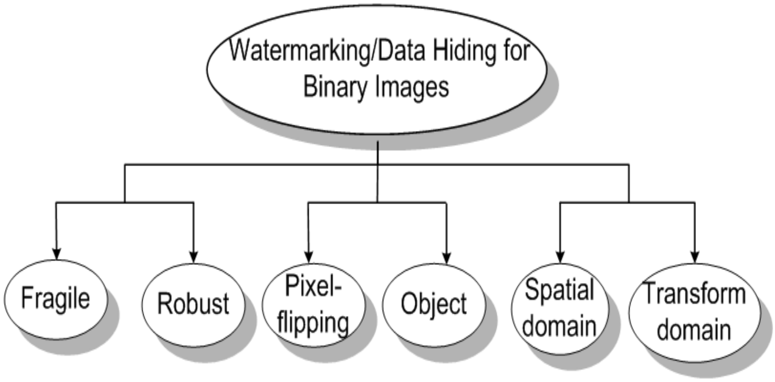


Fig. 1. Classification of Data Hiding/Watermarking Techniques for Binary Images

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 188–202, 2008.

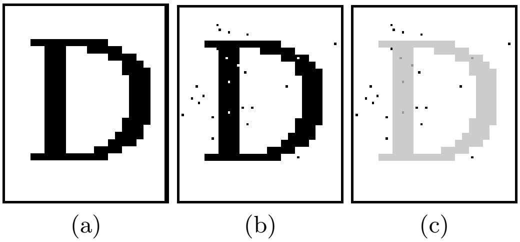


Fig. 2. Effects of Arbitrarily Flipping Pixels in Binary Images. (a) the original letter "D", (b) the letter "D" with 24 pixels being flipped and (c) the difference image.

cause obtrusive distortions under normal viewing conditions. On the contrary, binary images only have one bit-plane, whereby each pixel takes one of the two possible colors, e.g., "1" represents black and "0" represents white. Arbitrarily flipping pixels in the non-edge regions in binary images creates dramatic difference as illustrated in Fig. 2. In the difference image, pixels in black and dark gray represent the pixels that are flipped from white to black, and from black to white, respectively.

In the past few years, many spatial domain techniques are proposed for document watermarking and data hiding [1]. Among these methods, the line, word and character shifting methods [2], [3], [4], [5], [6], [7], [8] have low capacity. Typically, these methods (especially for the line and word shifting methods [2], [3]) are designed mainly for copyright protection purposes such that the watermark can survive the printing-photocopying-scanning process. The inter-word spaces coding [6], [7], [9] and inter character spaces coding [4], [5], [8] methods achieve larger capacity compared with that of the line/word shifting method, but the robustness has decreased. The feature-based method [10] encodes the watermark by modifying the length of selected runs of a stroke. The security issue has not been sufficiently addressed in these object-based methods to be used for authentication purposes. Moreover, most of these techniques maybe vulnerable to the attacks of deliberate retyping and reformatting the documents.

A data hiding scheme that makes use of a key and a weight matrixes achieves high data embedding rate (i.e., the number of bits embedded for each flipping) [11], but the randomness in choosing the data hiding locations results in poor visual quality of the watermarked images. Further improvements are made in [12] by considering the edge pixels only. Generally, the watermark in spatial domain can be embedded by 1) incorporating the odd-even feature of a group of pixels [13], [14], [15], [16], [17], [18]; 2) employing pairs of contour patterns [19]; 3) mapping the block features to the message bit [20] or enforcing the modulo value of the element-wise computation of the image, key and weight matrixes [11], [12], [21]; 4) enforcing the ratios of black versus white pixels [22], or enforcing the neighborhood pixel ratios [23], and 5) enforcing the relationship or modulating the white spaces in neighboring lines, words, characters and segments [2], [3], [4], [5], [6], [7], [8], [9].

Recently, significant efforts have been made in finding good “flippable” locations to carry the information data [12], [13], [14], [15], [16], [17], [18], [19], [20], [24], [25], [26]. These methods include: defining a visual distortion table to assess the “flippability” of a pixel [13], [14], [16], [17], [18]; defining pairs of contour edge patterns [19] and just using edge locations [12]. Random shuffling techniques have been employed in [13], [14], [16], [17] to equalize the uneven embedding capacity of binary images. Adaptively finding the “Connectivity-Preserving” patterns to carry the watermark data also addresses the “uneven-embeddability” of the images [24], [25]. Employing the denoise patterns [15], [20] achieves good visual effects of the watermarked image due to the denoising effects of the patterns. A low capacity can be expected for the images of high resolutions due to the decrease in the number of denoise patterns. Pairs of contour edge patterns that are dual to each other are employed to find suitable locations for data hiding in [19], a minimum length of contour segments is required such that the capacity is not high.

Traditional distortion measures such as mean square error (*MSE*), signal to noise ratio (*SNR*) and peak signal to noise ratio (*PSNR*) are no longer suitable for measuring distortion in binary images since the relationships among pixels in binary images such as connectivity and smoothness are not considered. Flipping the same amount of pixels in different locations of the same binary image may result in visually quite different watermarked images even if *MSE*, *SNR* or *PSNR* are same. Methods for evaluating the distortions caused by flipping pixels in binary images are proposed in [27], [28], [29] and the method proposed in [27] is subsequently employed in determining the “embeddable” locations for binary images data hiding [30].

The edge portions are chosen as the data hiding locations in the run-length based method [31]. Choosing the edge locations alone does not guarantee the good visual quality of the resultant watermarked image, which has been improved by taking the characteristic of the blocks into consideration [32]. Watermarking for binary message under the attack of modulo two addition noise is considered in [33], the probabilities of watermark missing and false alarm as a function of the distortion constraints and the length of the watermark are derived. Employing the code replacement based on the minimum distance of two codes [34] results in visible distortions for large number of authentication bits or small number of card holders. The changeable pixels in a block are chosen by computing the distance and weight matrixes in [35]. Errors in one row may be propagated to its subsequent rows due to the row by row processing.

To gain high security, public/private key encryption algorithms are incorporated for binary images authentication [16], [17], [18], [24], [25], [36], [37]. How to counter against the “Parity Attack” is a key problem for the algorithms that employ the odd-even enforcement to embed one bit of data [16], [17], [18], [24], [25], [37]. Chaining the blocks in the shuffled domain and embedding the image fingerprint computed in one block into the next block [16] help alleviate the attack. However the last several blocks still suffer the “Parity Attack”. Watermarking on JBIG2 text images [17] is done by embedding watermark in one of the instances, namely the data-bearing symbol using the pattern-based method proposed in [16]. Recently, a list of 3 × 3 patterns with symmetrical center pixels are employed to choose the data hiding locations in [18] which is similar to [24], [25]. Flipping the center pixels in some patterns may break the "connectivity" between pixels or create an erosion and protrusion [18]. Hence the visual quality of the watermarked image is difficult to control. In addition, employing the fixed 3 × 3 block scheme to partition the image leads to small embedding capacity.

Embedding information on a selection channel of the cover object while keeping the receiver with no information on the selection rule is discussed in [38], [39]. The wet paper codes allow a high utilization of pixels with high flippability score, thus improve the embedding capacity significantly. Localization of tampering for binary images is seldom addressed in the literature due to the uneven distribution of the "flippable" pixels and low capacity of binary images. A watermark of 7 bits is used to represent each alphanumeric character in [40]. The watermark for restoring or detecting tampering for each character is embedded in another character determined by a random or cyclic key. Accurate segmentation and recognition of each character play an important role in implementing this scheme. The low accuracy of the tampering localization is achieved due to employing a larger sub-image size [36]. Moreover, the uniform white or black areas that do not participate in the data hiding process are easily tampered. We recently proposed a two-layer data authentication scheme with tampering localization capability [37]. The localization accuracy has been improved due to the small macro-block is employed. The dynamic block-identifier is effective in detecting the changes made to the macro-block.

Data hiding for binary images based on real transforms is known to be difficult due to the quantization errors introduced in the pre/post-thresholding and binarization process [41], which renders the visual quality of the resultant watermarked image more obtrusive. An early investigation on watermarking in transform domain is presented in [41], in which the watermark is embedded in spatial domain by using the line and word shifting method to achieve good visual quality while the watermark detection is done in frequency domain to achieve high robustness. A further development in [42] is to embed the watermark on *DC* components of discrete cosine transform (*DCT*) with the aid of a pre-blurring and post-biased binarization process. The Interlaced Morphological Binary Wavelet Transform (*IMBWT*) is employed for data hiding in [43]. The coefficients obtained from the transform contain the transitions in horizontal, vertical and diagonal directions (i.e., the edges in vertical, horizontal and diagonal directions), which can be employed for data hiding applications.

2 A General Framework

Many data hiding techniques for binary images have been proposed in recent years. In this paper, we propose a typical framework for watermark embedding, extraction and authentication for binary images, which is depicted in Fig. 3.

The watermark embedding process starts with processing a binary image Y based on blocks, e.g., in [11], [12], [13], [14], [15], [17], [18], [20], [21], [24], [25], [26], [30], [34], [35], or pixels scanned in raster scan order, i.e., in row by row and column by column sequence [19], [31]. The features ($\mathcal{B}_f$) of the block or segment computed by $\mathcal{C}_f(Y)$ can be the odd-even features (modulo 2) [13], [14], [15], [17], [18], [30], [31], other modulo features [11], [12], [21], block signatures [20], "add" and "delete" patterns [19], etc. The kth payload watermark $\mathcal{W}_p$ is embedded in $E_m(Y, \mathcal{B}_f, \mathcal{W})$ by flipping the "embeddable" pixels (if needed) to enforce $\mathcal{W}(k) = \mathcal{B}_f(m)$, where $\mathcal{W} = \mathcal{W}_p$ and m is the block index, $m, k \in Z$ and Z denotes the nonnegative integer set. Finally, the watermarked image Y_w is obtained. The "embeddable" blocks or segments and "embeddable" pixels are determined by $\mathcal{D}_e(Y, \mathcal{R})$, where $\mathcal{R}$ is the determination rules. Specifically, $\mathcal{R}$ for determining the "embeddable" blocks or segments can be: a block having at least one "embeddable" pixel [24], or having a pixel with low visual impact [18]; each block in the shuffled domain [13], [14]; a segment containing "add" or "delete" patterns [19]; each non-completely black or white block after embedding the data [11], [30]; and each non-completely black or white block after embedding the data that contains the edge pixels [12], etc.. While $\mathcal{R}$ for assessing the "flippability" of a pixel can be: visual distortion table [13], [14], [17], [18]; "Connectivity-Preserving" criterion [24]; contour edge pattern pairs [19], distortion measure [30] and pixels [11] or edge pixels [12] chosen by a random key matrix, etc.. Finally, $\mathcal{R}$ for determining the "embeddable" pixel in a block or segment can be: a pixel with the highest flippablity score [13], [14] or low visual impact [17], [18], [30]; first "embeddable" pixel in the block [24]; changeable pixel in "add" or "delete" patterns [19], and pixels [11] or edge pixels [12] chosen by a random key matrix, etc..

To ensure the authenticity and integrity of Y, the salient features of the image or intermediate image, e.g., Y_1, can be generated by $\mathcal{X}_f(Y)$. For the methods whose "embeddable" locations can be determined for both embedding and extraction processes, e.g., in [18], [24], [25], one way to generate the intermediate image Y_1 is to clear out the "embeddable" locations of Y. Y_1 is subsequently fed to a hash function such as *SHA-1* to generate the hash value $\mathcal{H}_o = Hash(Y_1)$. $\mathcal{H}_o$ is encrypted with an encryption algorithm $E_k()$ to obtain the message authentication code or cryptographic signature of Y, i.e., $\mathcal{W}_s = E_k(\mathcal{H}_o, K_s)$, where $\mathcal{K}_s$ is the private key of the authorized user or owner. $\mathcal{W}_s$ is subsequently combined with $\mathcal{W}_p$ to generate the authenticator watermark $\mathcal{W}_r = \mathcal{C}_m(\mathcal{W}_s, \mathcal{W}_p)$. One choice of the operation $\mathcal{C}_m$ is concatenating $\mathcal{W}_s$ with $\mathcal{W}_p$, i.e., $\mathcal{W}_r = \mathcal{W}_s \parallel \mathcal{W}_p$, where "$\parallel$" denotes "concatenation" operation. $\mathcal{W}_r$ is embedded in the "embeddable" locations of Y by $E_m(Y, \mathcal{B}_f, \mathcal{W})$ to enforce $\mathcal{W}(k) = \mathcal{B}_f(m)$, where $\mathcal{W} = \mathcal{W}_r$, finally the watermarked image Y_w is obtained. $\mathcal{W}_p$ can be a visual pattern such as a binary logo image, a random sequence, a fingerprint or a handwritten signature image obtained from the authorized user or owner. Bitwise accuracy is required in the verification of the watermark.

For the watermark extraction, Y_w will go through similar process to determine the extraction locations by $\mathcal{D}_e(Y_w, \mathcal{R}')$. $\mathcal{R}'$ can be the same rules used in the embedding process, e.g., in [18], [19], [24], [25], or some pre-defined rules, e.g., each block of Y_w in the shuffled domain [13], [14]; each non-completely black or white block [11], [30] and each even, non-completely black or white block [12]. Compute the features of blocks $\mathcal{B}'_f$ by $\mathcal{C}_f(Y_w)$, thereafter, the watermark data $\mathcal{W}'_p$ or $\mathcal{W}'_r$ can be extracted by $E_x(\mathcal{B}'_f, Y_w)$ from the extraction locations of Y_w. Comparison of $\mathcal{W}'_p$ with $\mathcal{W}_p$ gives the verification results when $\mathcal{W}_p$ is embedded. Otherwise, if message authentication code or cryptographic signature is employed, the same $\mathcal{C}_m(\mathcal{W}'_r, \mathcal{W}_p)$ is used to retrieve $\mathcal{W}'_s$. $\mathcal{W}'_s$ is subsequently decrypted via the decryption algorithm $D_k(\mathcal{W}'_s, \mathcal{K}_p)$ to obtain $\mathcal{H}'_o$ of Y, where $\mathcal{K}_p$ is the public key of the authorized user or owner. The salient features of Y_w or the intermediate images Y_{w1} is computed by $\mathcal{X}_f(Y_w)$, which are subsequently fed to the hash function to generate the hash value $\mathcal{H}_w = Hash(Y_{w1})$. Comparison of $\mathcal{H}'_o$ with $\mathcal{H}_w$ gives the authentication results. Computing message authentication code or cryptographic signature based on salient features can achieve certain robustness for the authenticator watermark. In designing a data hiding scheme for authentication, certain robustness against random noise is desirable. For this purpose, Error Correction Codes (*ECC*) can be employed to encode the watermark, the encoded messages are embedded in the image.

3 Multi-level Signature for Authentication

3.1 2-D Interlaced Morphological Binary Wavelet Transform

The 1-D morphological binary wavelet decomposition scheme in [44] is extended to a 2-D signal and an interlaced transform has been proposed in our previous paper [43]. Let i and j be the indices of the signal at level $l+1$, where $i = 0, 1, 2, ...M-1$ and $j = 0, 1, 2, ...N-1$ for a 2-D coarse signal of size $M \times N$. Designation of the samples in a 2 × 2 block is shown in Fig. 4, where $s(2i, 2j)$ denotes the signal ("0" or "1") located at row $2i$ and column $2j$ at level l. Let the operators for the coarse signal, horizontal, vertical and diagonal detail signals be ψ^{ee}, ϖ_h^{ee}, ϖ_v^{ee} and ϖ_d^{ee}, respectively, where the superscript "ee" denotes "even-even". The obtained transform is named as the *even-even* transform since it is operated on a 2 × 2 block starting from the even-even coordinates. The coarse signal, vertical, horizontal and diagonal detail signals at level $l+1$ are obtained by applying the analysis operators to yield

$$s^{ee}(i,j) = \psi^{ee+}(s)(i,j) = s(2i+1, 2j+1) \tag{1}$$

$$v^{ee}(i,j) = \varpi_v^{ee+}(s)(i,j) = s(2i+1, 2j) \oplus s(2i+1, 2j+1) \tag{2}$$

$$h^{ee}(i,j) = \varpi_h^{ee+}(s)(i,j) = s(2i, 2j+1) \oplus s(2i+1, 2j+1) \tag{3}$$

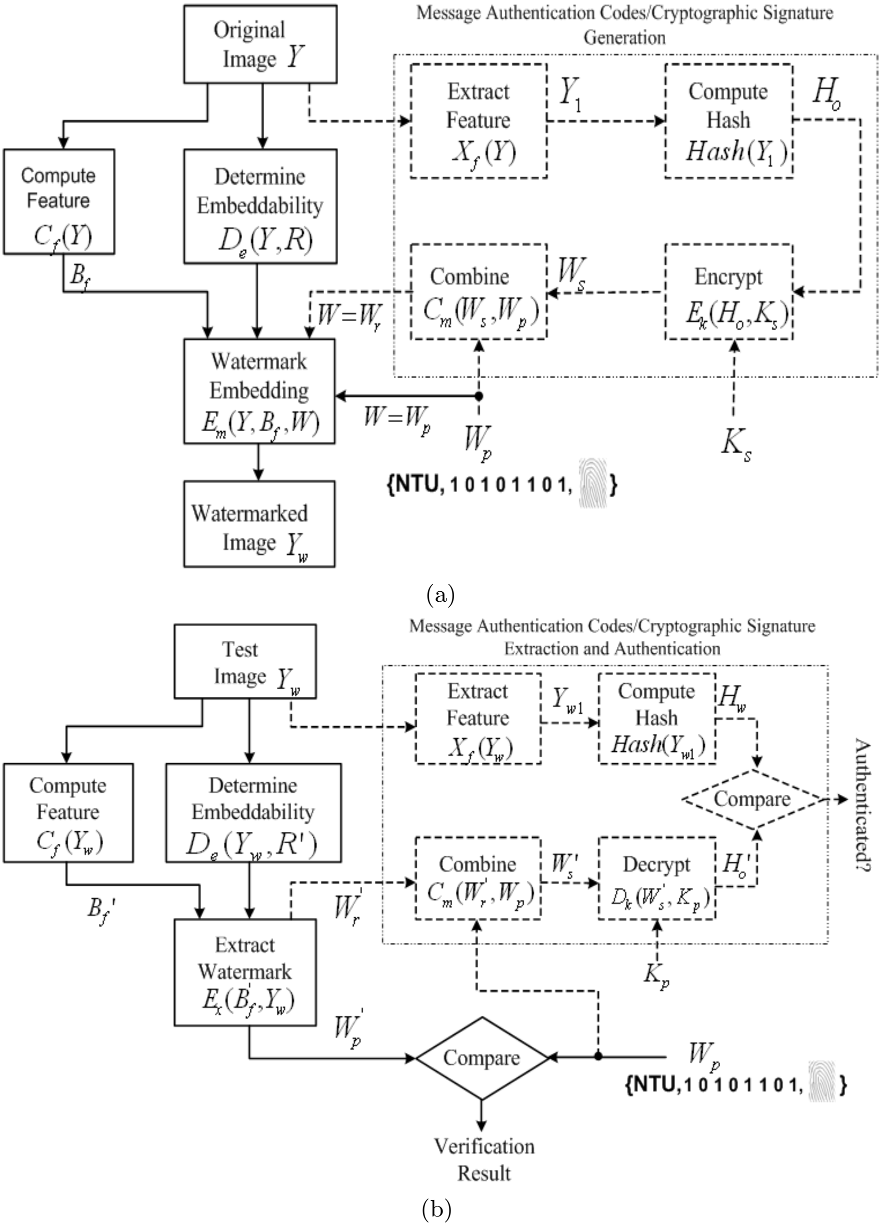
$$d^{ee}(i,j) = \varpi_d^{ee+}(s)(i,j) = s(2i, 2j) \oplus s(2i+1, 2j) \oplus s(2i, 2j+1) \oplus s(2i+1, 2j+1) \tag{4}$$

Finally, the signal at level l can be reconstructed by

$$\begin{aligned} s(2i, 2j) &= \psi^{ee-}(s)(2i, 2j) \oplus \varpi_d^{ee-}(s)(2i, 2j) \\ &= s^{ee}(i,j) \oplus v^{ee}(i,j) \oplus h^{ee}(i,j) \oplus d^{ee}(i,j) \end{aligned} \tag{5}$$

$$s(2i, 2j+1) = \psi^{ee-}(s)(2i, 2j+1) \oplus \varpi_h^{ee-}(s)(2i, 2j+1) = s^{ee}(i,j) \oplus h^{ee}(i,j) \tag{6}$$

$$s(2i+1, 2j) = \psi^{ee-}(s)(2i+1, 2j) \oplus \varpi_v^{ee-}(s)(2i+1, 2j) = s^{ee}(i,j) \oplus v^{ee}(i,j) \tag{7}$$



(a)

(b)

Fig. 3. A General Framework for Watermark Embedding (a) and Watermark Extraction and Authentication (b)

$$s(2i+1,2j+1)=\psi^{ee-}(s)(2i+1,2j+1)\oplus\varpi^{ee-}(s)(2i+1,2j+1)=s^{ee}(i,j) \tag{8}$$

Other transforms that are interlaced with the even-even transform can be similarly defined [43].

$s(2i,2j)$	$s(2i,2j+1)$
$s(2i+1,2j)$	$s(2i+1,2j+1)$

Fig. 4. Designation of samples in a 2 × 2 block

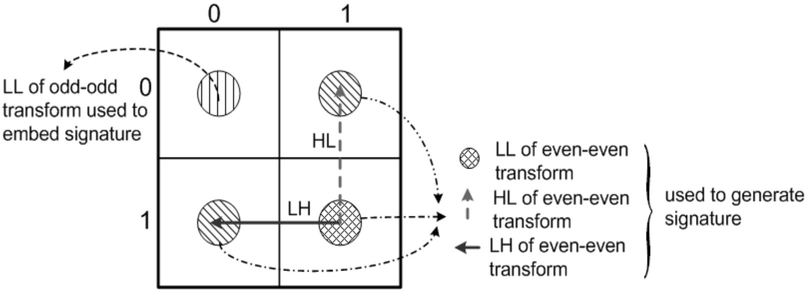

0
1
0
1
LL of odd-odd transform used to embed signature
HL
LH
LL of even-even transform
HL of even-even transform
LH of even-even transform
used to generate signature

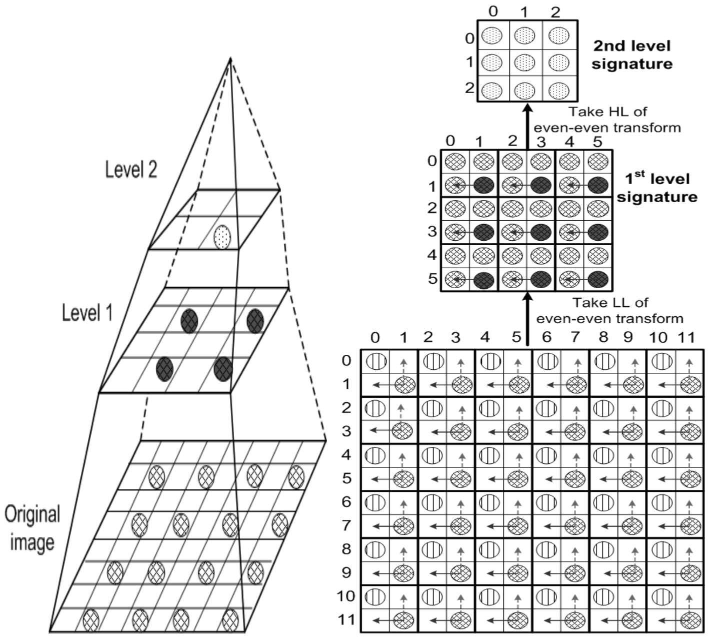
Fig. 5. A 2 × 2 block used to generate and embed the multi-level signature

3.2 Multi-level Signature

In this paper, we employ one pair of processing cases of the interlaced morphological binary wavelet transform (*IMBWT*) to generate a multi-level signature, e.g., even-even and odd-odd processing cases. As discussed in [43], the *flippability* condition of the even-even processing case is not affected by flipping the candidates in the odd-odd processing case, this feature has been employed for multi-level signature generation and subsequently for signature embedding. In particular, the coarse signals (*LL*) of the odd-odd transform are employed to embed the information, whereas the coefficients (e.g., *LL*, *HL* and *LH*) obtained from computing the even-even transform are used to generate the signature, as illustrated in Fig. 5. The non-intersection property of the pair processing case renders the correct extraction of the signature data. The hierarchical structure of the multi-level signature and an example of the signature generation are illustrated in Fig. 6.

Let $\mathcal{C}$ be the capacity of the image and let $W \times H$ be the image size. By decomposing the original image into level l, we expect to generate a signature that at least can meet the capacity requirement. Hence, the number of levels that the image should be decomposed should satisfy

$$\lfloor \frac{W \times H}{4^l} \rfloor \geq C \tag{9}$$



(a) (b)

Fig. 6. The hierarchical structure of the multi-level signature (a) and an example of the multi-level signature generation (b)

Solving this equation, we obtain the maximum number of decompositions l, which is given by

$$l \leq \frac{1}{2} log_2 \lfloor \frac{W \times H}{C} \rfloor \tag{10}$$

Typically, when the decomposition $l > \frac{1}{2} log_2 \lfloor \frac{W \times H}{C} \rfloor$, we could concatenate multiple higher level signatures to generate the final signature. To generate a multi-level signature, we can employ the coarse signal LL (s), horizontal detail LH (h) and vertical detail HL (v) as the recursive coefficients to be used for the next level decomposition. Here again, we utilize the "non-intersection" property of the coefficients. In each level, totally there are four options to choose the recursive coefficients for the next level decomposition, e.g., LL, LH, HL and HH. Whereas only three coefficients can be chosen in the first level, e.g., LL, LH and HL. This is so since the use of HH coefficients will affect the LL coefficients of the odd-odd transform, which subsequently will affect the correct extraction of the embedded signature. Assume the signature is finally decomposed into l levels, we have $\binom{3}{1} \times \binom{4}{1}^{l-1} = 3 \times 4^{l-1}$ options to choose the recursive coefficients, which reversely means that the probability of an attack to figure out the decomposition structure to compromise the signature is $p < \frac{1}{3 \times 4^{l-1}}$. Take as an example, for an image of size 512 × 512 and with a capacity of 1024 bits, we can decompose it into 8 levels to obtain the signature. In this case, the probability for an attack to figure out the structure of the signature is only $p = 2.0 \times 10^{-5}$. Hence, the proposed scheme has sufficient security. The initial signature S_c is formulated from the higher level down to the lower level by

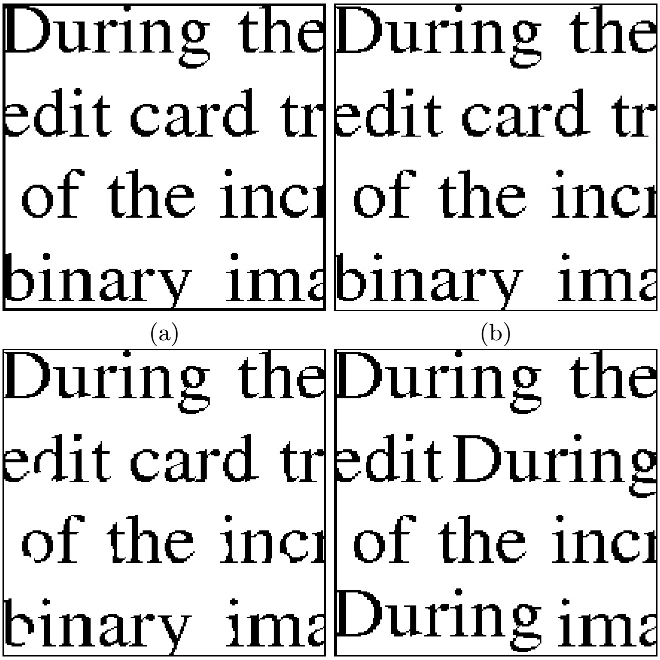
$$S_c = S_g(l) \parallel S_g(l-1) \parallel ... \tag{11}$$

To provide randomness in the signature, the concatenated signature S_c is finally shuffled and randomly chosen to obtain the final signature S_g depends on the capacity C by

$$S_g = G_s(K_s, K_p, S_c, C) \tag{12}$$

where K_s is the shuffling key used to shuffle S_c, K_p is the random key used to pick a total of C bits from the shuffled S_c and $G_s()$ is the signature generation function.

To evaluate the performance of the proposed multi-level signature scheme, we employ the hamming distance between the extracted or re-formulated signature S_{g2} and the original signature S_{g1}, which is normalized with respect the signature length L_s and given by



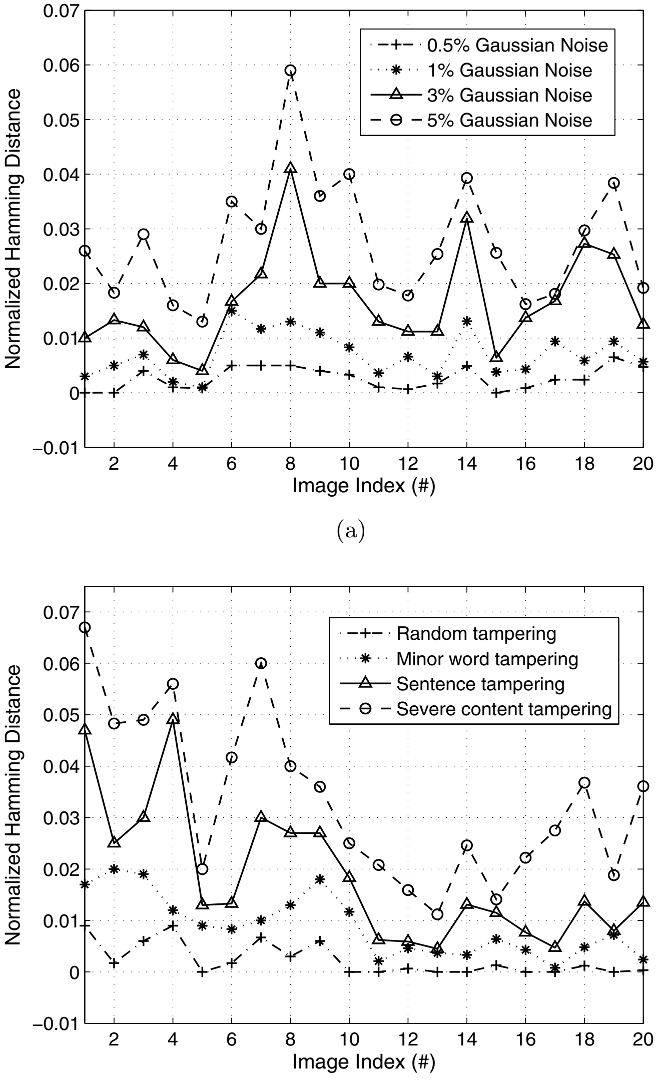
(a) (b)

(c) (d)

Fig. 7. Data hiding and tampering detection results. (a) original image of size 217× 217, (b) the watermarked image by embedding 400 bits signature, (c) the watermarked image that has been tampered by minor erasing in the boundaries of several words, and (d) the watermarked image that has been seriously tampered by cutting and pasting.

$$d(S_{g1}, S_{g2}) = \frac{1}{L_s}(\sum_{i=1}^{L_s}(S_{g1}(i) \oplus S_{g2}(i))) \tag{13}$$

For two similar sequences, we expect that $d(S_{g1}, S_{g2})$ is sufficiently small, whereas for two dissimilar sequences, we expect that $d(S_{g1}, S_{g2})$ should be larger than a selected threshold.



(a)

(b)

Fig. 8. Performance of the proposed multi-level signature under (a) additive noise and (b) different types of content tampering

3.3 Experimental Results

Experiments are conducted to test the "non-intersection" property of the data embedding mechanism and signature generation mechanism. In addition, the effectiveness of the proposed multi-level signature is also tested. The data hiding and tampering results are shown in Fig. 7. In the experiment, we decompose the image into four levels and generate a signature of 400 bits, which is subsequently embedded in the image. Without any tampering, the digital signature can be perfectly extracted and verified. When the image is tampered as shown in Fig. 7(c) and (d), the normalized hamming distances between the re-formulated signature from the tampered image and the original signature are 0 and 0.0125, respectively, indicating that the proposed signature scheme is able to differentiate minor and severe content tampering.

The performance of the proposed multi-level signature scheme against additive noise and content tampering are shown in Fig. 8(a) and (b), respectively. The tampered images are generated using the Adobe Photoshop imaging tool and divided into four categories: minor tampering without changing the contents of the image, cutting and pasting some single words, tampering the contents of some sentences and tampering the contents of the whole image. The results reveal that by properly choosing a threshold, the proposed multi-level signature is able to tolerate minor random noise while resisting malicious tamperings.

4 Conclusions

This paper reviews data hiding and watermarking techniques for binary images and propose a general framework for binary images authentication. Finally, a multi-level signature generation scheme is proposed for binary images authentication. The performance of the proposed multi-level signatures against additive noise and content tamperings reveal its capability of differentiating malicious tampering from minor tamperings. Future work involves quantifying the proposed signature scheme under more malicious attacks.

References

1. Chen, M., Wong, E.K., Memon, N., Adams, S.: Recent Development in Document Image Watermarking and Data Hiding. In: Proc. SPIE Conf., vol. 4518, pp. 166–176
2. Brassil, J.K., Low, S., Maxemchuk, N.F., O'Gorman, L.: Electronic Marking and Identification Techniques to Discourage Document Copying. IEEE Journal on Selected Areas in Communications 13(8), 1495–1504 (1995)
3. Brassil, J.K., Low, S., Maxemchuk, N.F.: Copyright Protection for the Electronic Distribution of Text Documents. Proceedings of the IEEE 87(7), 1181–1196 (1999)
4. Chotikakamthorn, N.: Electronic Document Data Hiding Technique using Inter-Character Space. In: The 1998 IEEE Asia-Pacific Conf. on Circuits and Systems, November 1998, pp. 419–422 (1998)

5. Chotikakamthorn, N.: Document Image Data Hiding Technique Using Character Spacing Width Sequence Coding. In: Proc. of 1999 Int. Conf. on Image Processing, October 1999, vol. 2, pp. 250–254 (1999)
6. Huang, D., Yan, H.: Interword Distance Changes Represented by Sine Waves for Watermarking Text Images. IEEE Trans. on Circuits and Systems for Video Technology 11(12), 1237–1245 (2001)
7. Kim, Y.-W., Moon, K.-A., Oh, I.-S.: A Text Watermarking Algorithm based on Word Classification and Inter-Word Space Statistics. In: Proc. of Seventh Int. Conf. on Document Analysis and Recognition, August 2003, vol. 4, pp. 775–779 (2003)
8. Yang, H., Kot, A.C.: Text Document Authentication by Integrating Inter Character and Word Spaces Watermarking. In: Proc. of the 2004 IEEE Int. Conf. on Multimedia and Expo (ICME 2004), June 27-30, 2004, vol. 2, pp. 955–958 (2004)
9. Zou, D., Shi, Y.Q.: Formatted Text Document Data Hiding Robust to Printing, Copying and Scanning. In: Proc. of IEEE Int. Symp. on Circuits and Systems (ISCAS 2005), May 23-26, 2005, vol. 5, pp. 4971–4974 (2005)
10. Amamo, T., Misaki, D.: A Feature Calibration Method for Watermarking of Document Images. In: Proc. 5th Int. Conf. on Document Analysis and Recognition, Bangalore, India, pp. 91–94 (1999)
11. Pan, H.-K., Chen, Y.-Y., Tseng, Y.-C.: A Secure Data Hiding Scheme for Two-Color Images. In: Proc. of the Fifth Symp. on Computers and Communications, July 3-6, 2000, pp. 750–755 (2000)
12. Tseng, Y.C., Pan, H.-K.: Data Hiding in 2-Color Images. IEEE Trans. on Computers 51(7), 873–878 (2002)
13. Wu, M., Tang, E., Liu, B.: Data Hiding in Digital Binary Image. In: IEEE Int. Conf. on Multimedia and Expo., New York, July 30 - August 2, vol. 1, pp. 393–396 (2000)
14. Wu, M., Liu, B.: Data Hiding in Binary Images for Authentication and Annotation. IEEE Trans. on Multimedia 6(4), 528–538 (2004)
15. Yang, H., Kot, A.C.: Data hiding for Bi-level Documents Using Smoothing Techniques. In: Proc. of the 2004 IEEE Int. Symp. on Cirsuits and Systems (ISCAS 2004), May 23-26, 2004, vol. 5, pp. 692–695 (2004)
16. Kim, H.Y., de Queiroz, R.L.: A Public-Key Authentication Watermarking for Binary Images. In: Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2004), October 2004, vol. 5, pp. 3459–3462 (2004)
17. Pamboukian, S.V.D., Kim, H.Y., de Queiroz, R.L.: Watermarking JBIG2 Text Region for Image Authentication. In: Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2005), September 11-14, 2005, vol. 2, pp. 1078–1081 (2005)
18. Kim, H.Y.: A New Public-Key Authentication Watermarking for Binary Document Images Resistant to Parity Attacks. In: Prof. IEEE Int. Conf. on Image Processing (ICIP 2005), September 11-14, 2005, vol. 2, pp. 1074–1077 (2005)
19. Mei, Q., Wong, E.K., Memon, N.: Data Hiding in Binary Text Document. In: Proceedings of SPIE, vol. 4314, pp. 369–375 (2001)
20. Yang, H., Kot, A.C., Liu, J.: Semi-fragile Watermarking for Text Document Images Authentication. In: Proc. of the IEEE Int. Symp. on Circuits and Systems (ISCAS 2005), May 2005, vol. 4, pp. 4002–4005 (2005)
21. Liu, J., Yang, H., Kot, A.C.: Relationships And Unification of Binary Images Data-Hiding Methods. In: Proc. of the IEEE Int. Conf. on Image Processing, vol. 1, pp. 981–984 (2005)

22. Zhao, J., Koch, E.: Embedding Robust Labels into Images for Copyright Protection. In: Proc. of the Int. Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Vienna, August 1995, pp. 1–10 (1995)
23. Hu, S.: Document Image Watermarking Algorithm Based on Neighborhood Pixel Ratio. In: Proc. of the Int. Conf. of Acoustics, Speech, and Signal Processing 2005 (ICASSP 2005), March 18-23, 2005, vol. 2, pp. 841–844 (2005)
24. Yang, H., Kot, A.C.: Data Hiding For Text Document Image Authentication by Connectivity-Preserving. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2005), March 18-23, 2005, vol. 2, pp. 505–508 (2005)
25. Yang, H., Kot, A.C.: Pattern-Based Data Hiding for Binary Image Authentication By Connectivity-Preserving. IEEE Transactions On Multimedia 9(3), 475–486 (2007)
26. Pan, G., Wu, Z., Pan, Y.: A Data Hiding Method for Few-Color Images. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2002), May 13-17, 2002, vol. 4, pp. 3469–3472 (2002)
27. Lu, H., Kot, A.C., Shi, Y.Q.: Distance-Reciprocal Distortion Measure for Binary Document Images. IEEE Signal Processing Letters 11(2), 228–231 (2004)
28. Cheng, J., Kot, A.C.: Objective Distortion Measure for Binary Images. In: IEEE Region 10 Conference TENCON 2004, November 21-24, 2004, vol. 1, pp. 355–358 (2004)
29. Ho, A.T.S., Puhan, N.B., Marziliano, P., Makur, A., Guan, Y.L.: Perception Based Binary Image Watermarking. In: Proc. of the 2004 Int. Symp. on Circuits and Systems (ISCAS 2004), May 23-26, 2004, vol. 2, pp. 37–40 (2004)
30. Lu, H., Kot, A.C., Cheng, J.: Secure Data Hiding in Binary Images for Authentication. In: Proc. of the 2003 Int. Symp. on Circuits and Systems (ISCAS 2003), May 25-28, 2003, vol. 3, pp. 806–809 (2003)
31. Hwang, K.-F., Chang, C.-C.: A Run-length Mechanism for Hiding Data into Binary Images. In: Proc. of Pacific Rim Workshop on Digital Steganography, Kitakyushu, Japan, July 2002, pp. 71–74 (2002)
32. Agaian, S.S., Cherukuri, R.C.: Run length Based Steganography for Binary Images. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 481–484. Springer, Heidelberg (2005)
33. Korzhik, V., M.-Luna, G., Marakov, D., Marakova, I.: Watermarking of Binary Messages in Conditions of an Additive Binary Noise Attack. IEEE Signal Processing Letters 10(9), 277–279 (2003)
34. Tzeng, C.-H., Tsai, W.-H.: A New Approach to Authentication of Binary Images for Multimedia Communication with Distortion Reduction and Security Enhancement. IEEE Communications Letters 7(9), 443–445 (2003)
35. Liu, T.-H., Chang, L.-W.: An Adaptive Data Hiding Technique for Binary Images. In: Proc. of the 17th Int. Conf. on Pattern Recognition (ICPR 2004), August 23-26, vol. 4, pp. 831–834 (2004)
36. Kim, H.Y., de Queiroz, R.L.: Alteration-Locating Authentication Watermarking for Binary Images. In: Proc. Int. Workshop on Digital Watermarking, pp. 125–136 (2004)
37. Yang, H., Kot, A.C.: Binary Image Authentication With Tampering Localization By Embedding Cryptographic Signature and Block Identifier. IEEE Signal Processing Letters 13(12), 741–744 (2006)
38. Fridrich, J., Goljan, M., Lisonek, P., Soukal, D.: Writing on Wet Paper. IEEE Trans. on Signal Processing 53(10), 3923–3935 (2005)

39. Wu, M., Fridrich, J., Goljan, M., Gou, H.: Handling Uneven Embedding Capacity in Binary Images: A Revisit. In: SPIE Conference on Security, Watermarking and Stegonography, San Jose, CA, Proc. of SPIE, January 2005, vol. 5681, pp. 194–205 (2005)
40. Markur, A.: Self-Embedding and Restoration Algorithms for Document Watermark. In: Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2005), March 2005, vol. 2, pp. 1133–1136 (2005)
41. Liu, Y., Mant, J., Wong, E., Low, S.H.: Marking and Detection of Text Documents Using Transform-domain Techniques. In: Proc. of SPIE, Electronic Imaging (EI 1999) Conference on Security and Watermarking of Multimedia Contents, San Jose, CA, vol. 3657, pp. 317–328 (1999)
42. Lu, H., Shi, X., Shi, Y.Q., Kot, A.C., Chen, L.: Watermark Embedding in DC Components of DCT for Binary Images. In: 2002 IEEE Workshop on Multimedia Signal Processing, December 9-11, 2002, pp. 300–303 (2002)
43. Yang, H., Kot, A.C., Rahardja, S., Jiang, X.: High Capacity Data Hiding for Binary Images in Morphological Wavelet Transform Domain. In: Proc. of the IEEE Int. Conf. on Multimedia & Expo (ICME 2007), July 2-5, 2007, pp. 1239–1242 (2007)
44. Heijmans, H.J.A.M., Goutsias, J.: Nonlinear Multiresolution Signal Decomposition Schemes-Part II: Morphological Wavelets. IEEE Trans. on Image Processing 9(11), 1897–1913 (2000)

High-Capacity Invisible Background Encoding for Digital Authentication of Hardcopy Documents

Walter Geisselhardt and Taswar Iqbal

Distributet Systems
University Duisburg-Essen, Germany
Tel.: +49 (0) 203-379-4261
gd@uni-duisburg.de, taswar.iqbal@alumni.uni-duisburg-essen.de

Abstract. A digital authentication technique for hardcopy documents (e.g. contracts, official letters, ID cards etc.) is given. In order to secure the printed contents against forgery attacks the foreground contents are encoded in the superposed constant background greyscale image (CBGI) whereas in contents authentication process the contents are decoded from the scanned superposed background image and compared with scanned or printed contents.

The distinguishing features of the novel technique are: l) high-quality superposed constant background image with multiple grey levels, 2) higher data encoding capacity enabling one-to-one contents integrity authentication rather than some selected features, and 3) underlying textual-contents independence, a challenge encountered from the languages with complex writing structures, furthermore, the above characteristics of the superposed data encoding technique allow its application: in the area of secret communication as a tool for military, original quality rather than higher quality fax transmission and digital document management.

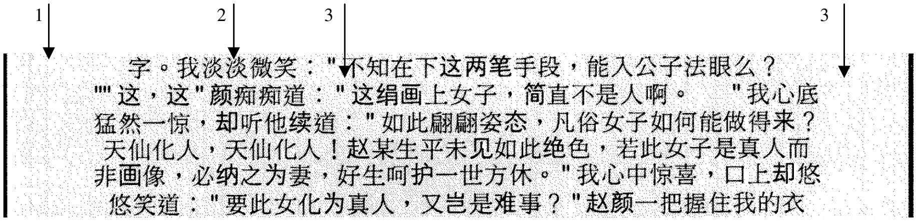
The superposed background image forming the encoded portion does not affect the aesthetic appearance of the document. The newly developed coding scheme in conjunction with a novel data-reading technique offers data encoding capacity 15 to 20 times higher than that one offered by market products.

Keywords: Smart IDs, counterfeiting, data tampering, entertainment tickets protection, printing scanning, watermarking, halftone images, 2-D Bar codes, HD-DataStripe, biometrics, copy detection, data encoding in background images.

1 Introduction

The literature dealing with digital authentication of hardcopy text documents can be divided into two categories. In first category digital watermarking techniques are used to embed authenticity verification related data in the textual contents by slightly modifying some selected features of text such as words, paragraphs, lines etc. In the authenticity verification process selected modified

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 203–221, 2008.



字。我淡淡微笑："不知在下这两笔手段，能入公子法眼么？
""这，这"颜痴痴道："这绢画上女子，简直不是人啊。 "我心底
猛然一惊，却听他续道："如此翩翩姿态，凡俗女子如何能做得来？
天仙化人，天仙化人！赵某生平未见如此绝色，若此女子是真人而
非画像，必纳之为妻，好生呵护一世方休。"我心中惊喜，口上却悠
悠笑道："要此女化为真人，又岂是难事？"赵颜一把握住我的衣

Fig. 1. Part of a document with an encoded portion as greyscale background (1: document; 2: text ; 3: encoded portion)

features are checked in the scanned image. The limitation of the watermarking approach is its low capacity and it is attributed to contents dependence of the watermarking process. Such watermarking is described e.g. in [1], [3], [4], [5], [6]. Furthermore, not all the contents can be used for embedding process due to the fact that the neighbouring contents are left unchanged in order to be used as a reference in the watermark recovery process. Otherwise original contents would be required (non-blind mode) for watermark recovery, which is not desirable as the original contents might not always be available.

In order to overcome the lower capacity problem that is attributed to contents dependence there has been used another approach in which superposed background image is used to carry authenticity verification related data. As an example, Fig. 1 shows part of a document comprising text and an encoded portion forming the superposed constant greyscale background for the human eye.

This approach is similar to two-dimensional barcodes except that here aesthetic appearance is focused upon to make the superposed background image visually pleasing. The well-known technique belonging to this category is called DataGlyphs and described in [7]. DataGlyphs that offer much higher data encoding capacity per unit area than text watermarking techniques are more pleasing than the barcode counterparts. However, its aesthetic appearance is not satisfactory and this is due to larger (visually perceptible) data encoding symbol size as well as the synchronization marks [10]. Whereas the smaller symbol size poses challenge in information recovery process due to the noise encountered from print and scan process. The DataGlyphs technique [8] is used for forgery and counterfeit detection in passports, bank checks etc., claiming that the resulting document is difficult to counterfeit.

Another technology known as SecureSeal^{TM} from EnSeal Systems is described in [10]. This technology focuses on aesthetic appearance of the data encoding Symbols. The improvement is claimed by reducing symbol size (but the coding is still visible) at a cost of lower capacity than DataGlyphs technology. Superposed background image given in [11] differs from the others discussed above that it results in very smooth image. In this technique imperceptibility constraint is achieved by using two different data encoding patterns, each one consisting of a number of very small size dots that are arranged in a specific order so that the resulting symbol remains invisible and can be identified from print and scan process. This technique also results in lower capacity than DataGlyphs. One common weakness of above techniques is that none of these allows full document contents to be encoded into the background image. Consequently they do not allow one-to-one contents integrity authentication that is necessary to check each and every modification as well as some other limitations to be discussed in the following while reviewing another technique given for digital authentication of hardcopy text documents.

In [13] while focusing on digital authentication, digital signatures of semantic information using OCR-technology are computed and encoded into the background in machine-readable form (e.g. watermarks), which are prone to high-quality copying attack. This technique has the following weaknesses: OCR dependence, non-conventional expensive ways (IR/UV inks, security patterns) are used to make the digital watermarks robustness against copying attacks. Also the given message digest computation algorithm has limitations due to 99% rather than 100% performance of OCR technology in ideal cases. It is mentionable that this technique can be applied only with drawbacks, in particular, to documents in which OCR-technology is very difficult due to complex writing structure of the language's non-alphanumeric characters. One example of such languages is Arabic, a widely used language that is very attractive from commercial point of view.

US 2006/0147082 AI [15] discloses marking of a document with invisible marks. Each mark is preceded by a marker to indicate to the scanner the beginning of a replication of a unique identifying code pattern. The code pattern is formed by dots forming a series of binary coded decimal numbers. So the number of dots depends on the coded number. The distributed marks do not allow an optimized capacity. Further, the system is sensible regarding scanning errors and dirt or other disturbances and influences.

Object of work reported here was to provide a hard copy document with a preferably invisible encoded portion and a method for generating such document, wherein the encoded portion allows an optimised high capacity of data and/or can be read with security or only few errors. More details are given in [12, 14].

The paper is organized as follows: Section 2 examines existing work which is the background of our research. This is followed by a description of our novel data hiding technique for background images in section 3. In Section 4 the capacity offered by the novel technique is defined and a formula is given. Section 5 explains the process of data recovery from background images, and Section 6 reports experimental results. Finally, conclusions are given in Section 7.

2 Background of Work

In [11] a constant background greyscale image (CBGI) is superposed to the foreground text and this background image is used as a channel for hidden communication to encode information related to the foreground contents. To encode data in the background image two different symbol patterns are used to encode "0" and "1" bits. As a hidden message some selected features of

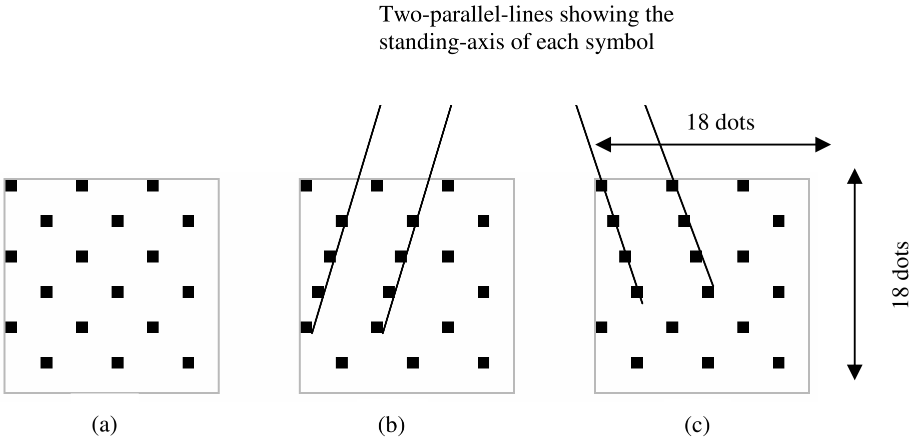


Fig. 2. Data encoding symbol patterns (18 x 18 dots) used by Suzaki et al. [11], (a) Null Symbol Pattern, (b) "1" bit encoding Symbol Pattern, (c) "0" bit encoding Symbol Pattern

the foreground text are used for contents integrity verification. To recover the hidden data after printing-scanning (PS) process the 2-D Gabor filter is applied to recognize each of the symbol patterns used to encode the information bits.

The CBGI is obtained by repeatedly applying a specially designed fixed-size binary pattern, known as Null Symbol (NS) pattern. Two different symbol patterns used to encode data in background image are obtained from the NS pattern with the constraint that a good symbol pattern should have the following characteristics: offers higher capacity, difference in symbol patterns is visually imperceptible and can be recognized efficiently during data recovery process. The information encoding symbol patterns along with the NS pattern, given in [11] are shown in Fig. 2.

To differentiate between the symbol patterns representing "0" and "1" bit, two parallel lines (indicating the direction of the standing-axis) each consisting of 4 black dots within the symbol pattern, are aligned in a specific direction and this difference in alignment/standing-axis is used to identify a particular symbol pattern after PS process. The capacity offered by an A-4 size background image (6684 x 4900 dots at 600 dpi) with symbol size (18 x 18 dots) without taking into account noise encountered during practical applications is 95,347 bits. To compensate for the errors encountered in decoded symbol patterns due to the foreground text overlapping, BCH ECC with 2047 bits code length is used, which reduces the resultant capacity to 51.5 kbits, 25.0 kbits, and 11.5 kbits for 5%, 10% and 15% errors, respectively. It is mentionable that 324 dots are used to encode one single information bit.

It is clear from above discussion that to gain any further increase in the capacity offered by [11] either symbol pattern size has to be decreased or the number of different symbol patterns is to be increased, while keeping the symbol pattern size fixed. However, either of these approaches poses challenge at symbol identification stage when the encoded data is to be recovered after the PS process. Consequently, according to the existing technique while designing symbol pattern to get further increase in capacity, the size and shape of the symbol pattern should be selected in such a way that it can be identified correctly during data recovery process.

There is another technique called DataGlyphs? [7], [9] developed by Xerox PARK, which allows encoding data in the background image. In DataGlyphs two different symbols: backward slash "\" and forward slash "/", each measuring 1/100th of an inch are used to encode "1" and "0" bits. This technique allows encoding most of the foreground text in full-page size background image and is used by Bern et al. [2] for contents integrity authentication.

3 Novel Data Hiding Technique for Background Images

The main objective of the novel data hiding technique for background images is to get further gain in the capacity while using a similar approach for data encoding as the one discussed above. In order to achieve this goal a single small-size information encoding symbol pattern with multi-bits data encoding capability is employed. The usage of such symbol pattern is encouraged by the capability to identify the individual dots reliably in the noisy environment. Furthermore, this technique takes into account the underlying characteristics of the printing device to get further gain in capacity. While considering errors caused by the foreground character overlapping, instead of relying on ECC, overlapping is avoided by replacing such areas with null symbols to eliminate any visual artifacts. To minimize ECC overhead to combat for unavoidable errors, the encoded data is scrambled all over the background image. The nature of the information encoding symbol allows multiple grey levels to be used with the same data encoding capacity.

3.1 Novel Data Encoding Symbols

In this research the Null Symbol Pattern given in [11] is modified, so that it results in the same background image quality. Next the 18 x 18 NS Pattern is partitioned into nine sub-symbol patterns and each New Symbol Pattern (NSP) with size six by six contains only two black dots, which are known as *primary* and *reference* dots. The primary dot is used to encode the information, whereas the reference dot is intended to assist in data recovery process, and with respect to this dot the location of primary dot is found in information decoding process. In this technique multiple bits of information are encoded in one NSP and this is achieved by shifting the position of the primary dot within the symbol pattern at four different locations equivalent to two bits. In Fig. 3(a) the Null Symbol Pattern is shown. In Fig. 3(b) nine NSPs are shown, each encoding two bits of information which totals 18 bits.

It is also mentionable that here only one dot (primary dot) is used to encode multiple information bits and no strong constraint like Standing-Axis in [11] is

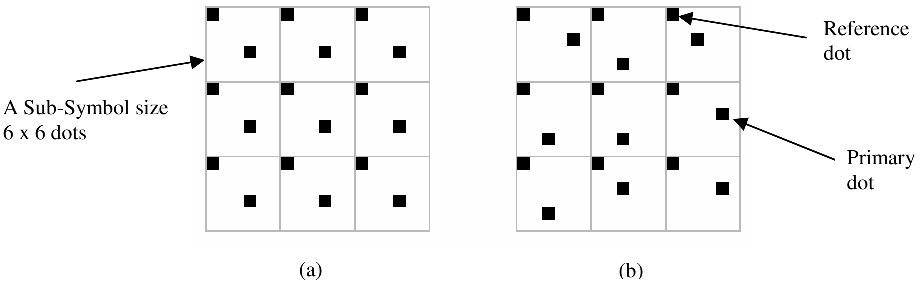


Fig. 3. a) Nine NSP Sub-symbols with Null Symbol Pattern [11], b) Nine Sub-symbols with encoded bits

imposed on data encoding symbol. The alignment constraint used in the existing technique would make it more sensitive against the skewing distortion in data recovery process as compared with the new approach.

3.2 Detailed Description of Preferred Embodiment

Fig. 1 shows an example of a document (1) comprising text (2) and an encoded portion (3). The latter forms a substantially constant background for the human eye. The document is preferably a laser printer paper printout. The text is superposed onto the encoded background, i.e. the encoded portion. In particular, the encoded portion is interleaved between the text. Various differing arrangements are possible. Fig. 4 illustrates the preferred construction of encoded portion 3 (Fig. 1) in more detail than Fig. 3.

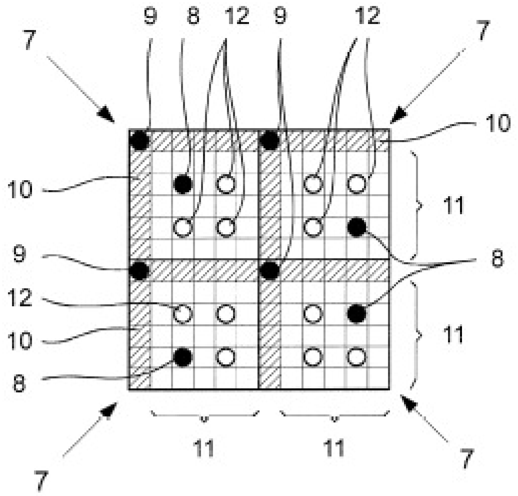


Fig. 4. Schematic representation of NSP

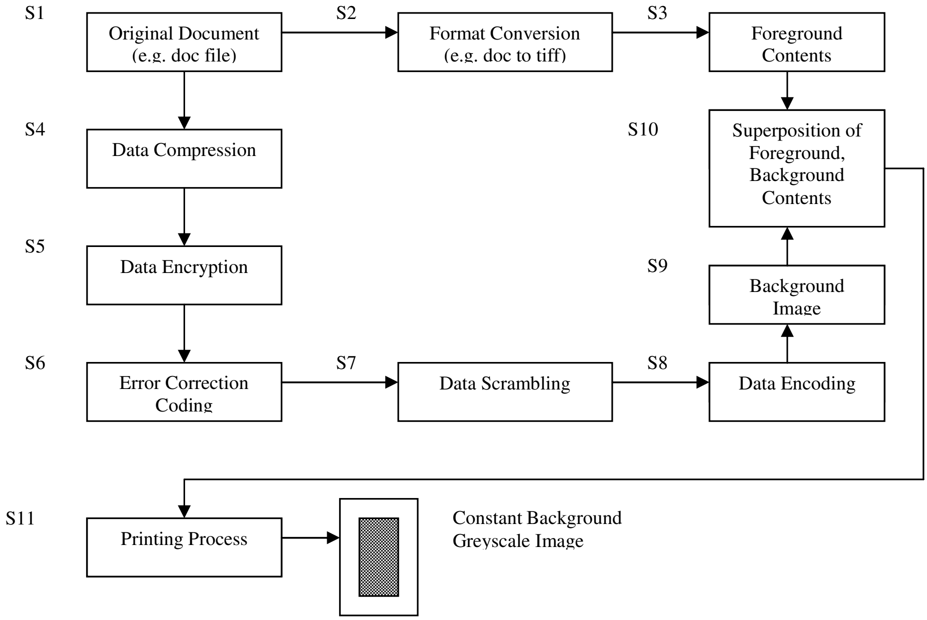


Fig. 5. Generating a CBGI-document with encoded portion

The encoded portion consists of symbols 7 forming a uniform region of greyscale or halftone. The symbols 7 and, thus, the encoded portion, are preferably unreadable or invisible for human eyes. However, the symbols 7 are machine-readable and encode information as data bits. Fig. 4 shows four such symbols. Each symbol consists of preferably only spatially spaced dots 8, 9 where dot 8 represents one data bit and dot 9 one synchronization bit. Each symbol comprises a synchronization region 10 and a data region 11. In the present embodiment each data dot 8 of a symbol 7 has n different possible positions 12 and preferably forms $\lfloor$log2(n)$\rfloor$ data bits ($\lfloor\ \rfloor$means 'floor operation'). The maximum size of dots 8, 9 forming symbol 7 is preferably at most 1/300 inch, more preferably 1/600 inch or less, depending on printer resolution.

Preferably, the encoded portion 3 encodes the complete text 2. It might, as well, comprise a fingerprint (e.g. for watermarking and/or biometric aspects) and/or a digital signature. Fig. 5 shows a schematic flowchart representing a preferred method for generating a document. The original electronic document, e.g. a file readable by a word processing program or the like, is provided in step S1. The document is converted into a graphic (printable) image in step S2 and provided as the foreground contents in step S3. Simultaneously the document is compressed and encrypted in steps S4, S5. An error correction and data scrambling step S6, S7 may follow. The final data encoding process, step S8, provides background image being superposed with foreground image in step S10.

The superposition is done by two different ways. Furthermore, it consists of two stages: 1) selection of suitable data encoding region, and 2) elimination of artifacts caused by data encoding symbols on the foreground contents in overlapping regions, whereas the latter stage is common and handled in same way in both cases.

According to the first approach, data is encoded uniformly over the entire background image and the errors caused by the overlapping of foreground contents are compensated by data scrambling and error correction coding (ECC). This approach results in lower capacity due to the increased overhead for ECC. With the second approach the graphic image file of foreground content is processed to look for free regions (shown in Fig.6) where data could be encoded in the background image without overlapping. Such a region in text document is defined by the rectangle of minimum area surrounding a text line, and the four coordinates of all such regions encountered are encoded separately into the background image with higher overhead for ECC. It is mentionable that the interleaved regions shall be encoded with Null Symbols, resulting in whole background image with data encoding symbols. Finally, only these non-overlapping regions (decoded in the beginning from background image) are used to recover data from scanned image.

By denoting a pixel value at position (i,j) for the foreground text image, superposed background image and the resulting image after superposition by X(i,j), Y(i,j), and Z(i,j), respectively, the process to count for artifacts caused by data encoding on foreground contents works as follows:

Z(i,j) = X(i,j) if X(i,j) = black,
Z(i,j) = Y(i,j) if X(i,j) = white.

According to that rule, the given pixel (i,j) of resulting mage Z(i,j) takes the pixel value of foreground text image X(i,j) when X(i,j) has black pixel. Otherwise, always the pixel value of the resulting image is the value of the superposed background image Y(i,j) regardless of information encoding dots. Above can be implemented efficiently using logical AND operation on two binary matrices $\boldsymbol{X}$, and $\boldsymbol{Y}$ of same size.

In digital authentication process the document to be authenticated is scanned at sufficiently higher over-sampling rate, preferably at least twice the printing resolution, and then data-reading technique is applied to decode the contents in the background image. On the recovered data all the operations shown in Fig. 5 are performed in reverse order and the resulting contents are output, e.g. as a doc file. At this stage human interaction based authentication can be ensured. Technique for automatic contents integrity authentication is described in [14].

4 Capacity Offered by the Novel Technique

In general the capacity, C, offered by either of the techniques under consideration, without taking into account the noises encountered during practical applications, can be computed using the following relation:

$$C = [(x \cdot y)/\sigma^2] \cdot \omega \tag{1}$$

where $\mathbf{x}, \mathbf{y}$ are number of dots in the horizontal and vertical direction of CBGI, respectively. σ is symbol size in dots and ω denotes the number of bits encoded per symbol.

The existing technique with $\mathbf{x} = 4629, \mathbf{y} = 6685, \sigma{=}18$ and $\omega{=}1$, results in $\mathbf{C} = 95.5$ kbits. Whereas the new technique with $\mathbf{x} = 4749, \mathbf{y} = 6775, \sigma{=}6$ and $\omega{=}2$, results in $\mathbf{C} = 1721$ kbits, which is 18 times more than the capacity offered by the existing technique. It is to be pointed out here and afterwards 1 kbits represents 1000 and not 1024 bits.

As the CBGI under consideration usually will be superposed by the text image in the final applications, so this scenario is investigated for the novel technique as well. It is to be pointed out that in existing technique [11] it is suggested to use ECC with higher overhead to overcome the errors in the data encoding symbols that are overlapped by the superposed text.

In our approach only those areas which are not overlapped by the superposed text are used for data encoding purpose. This choice is governed by the fact that the net capacity for novel technique while considering only those regions not overlapped by the characters is much higher as compared with the scenario that the data is encoded at each location and then errors caused by the character overlapping are eliminated using ECC with higher overhead. In the following a general (i.e. enables to encode data in certain areas that can be overlapped by superposed text) capacity computation technique is described. Fig. 6 illustrates the various regions. With the present approach only region A an B are considered.

Let $\boldsymbol{X}$ be a matrix of size $I$by J,where each element of $\boldsymbol{X}$ represents an information encoding symbol. After text image superposition an element of $\boldsymbol{X}$ is defined as:

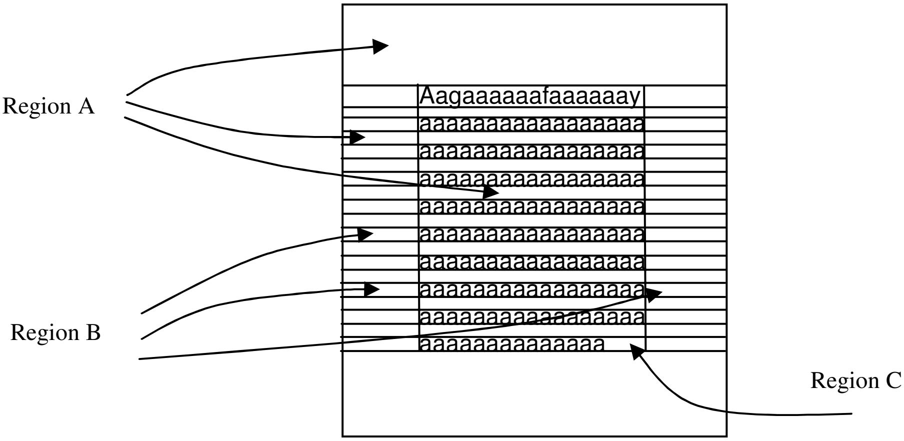


Fig. 6. Schematic illustration of regions used in capacity computation

$$x_{i,j} = \begin{cases} 1\ if\ \sum\limits_{\varepsilon_1=1}^{\sigma} \sum\limits_{\varepsilon_2=1}^{\sigma} x'_{\varepsilon_1,\varepsilon_2} \neq 0 \\ 0\ \sum\limits_{\varepsilon_1=1}^{\sigma} \sum\limits_{\varepsilon_2=1}^{\sigma} x'_{\varepsilon_1,\varepsilon_2} = 0 \end{cases}$$

Where $x'_{\varepsilon_1,\varepsilon_2}$ is a sub-element (single dot) within the information encoding symbol and σ is size of the information encoding symbol. It is to be noted that $x_{i,j} = 1$, means that there is at least one black dot in the region under consideration due to the superposed text image, otherwise it takes value 0 if all elements are white.

Then, the capacityY_c can be found as follows:

$$Y_c = \sum_{i=1}^{I} J \cdot \alpha_i + \sum_{i=1}^{I} \sum_{j=1}^{J} \beta_i \cdot \gamma_j \tag{2}$$

$$\text{where } \alpha_i = \begin{cases} 1\ if\ \sum\limits_{j=1}^{J} x_{i,j} \leq \kappa \\ 0\ \sum\limits_{j=1}^{J} x_{i,j} > \kappa \end{cases},\ \beta_i = \begin{cases} 1\ if\ \sum\limits_{j=1}^{J} x_{i,j} > \kappa \\ 0\ \sum\limits_{j=1}^{J} x_{i,j} \leq \kappa \end{cases}$$

$$\text{and } \gamma_j = \begin{cases} 1\ if\ \sum\limits_{i=1}^{I} x_{i,j} = 0 \\ 0\ \sum\limits_{i=1}^{I} x_{i,j} \neq 0 \end{cases}$$

The 1^{st} term of eq.2 takes into account region-A defined by all rows which are not overlapped or overlapped minimally. For instance, rows corresponding to the top and bottom parts the first line of text in Fig. 6 are not overlapped highly and can be considered as part of region-A by controlling the value of κ. The 2^{nd} term of eq.2 takes into account the region-B defined by all rows found overlapping by the first term of eq.2. It considers columns which are completely white and column elements corresponding to non-overlapping rows are counted in capacity computation. Obviously, in Fig. 6 the superposed CBGI and the foreground text image are of same size. Foreground text image is binary graphic image with black and white pixels being represented by decimal “1” and “0” values, respectively.

In eq.2 first summation increments by the factor J whenever the constraint given by α_i is satisfied (i.e. $\alpha_i = 1$), meaning there is sufficient free space for information encoding. Whereas the second summation increments by one whenever $\beta_i \cdot \gamma_j = 1$.

By varying the value of parameter κ, the capacity can be controlled and this is possible due to the fact that for *each text line* of the superposed image the top as well as bottom four rows of information encoding symbols have only a small number of symbols which are overlapped by the text characters. The errors caused by the overlapping characters in these regions can be tackled using either of the two methods: 1) these rows can be encoded separately using maximum run-length encoding technique, which would identify the sequence of connected locations where information can be encoded, and 2) information being encoded in the area under consideration is encoded with ECC with slightly higher error correction capability. It is noteworthy that using top and bottom four rows (i.e. $\kappa = \lambda$, λ is a fixed constant, defining how many overlaps are allowed in a given row) of each line of the superposed text image, an additional 200-300 kbits of information can be encoded, which is a significant gain in capacity. In this research parameter κ takes the value zero.

Finally, those regions, which are not used for information encoding, are encoded separately, and these regions are identified first from the printed and scanned (PS) image and are ignored while recovering original message. Due to the fact that this information is responsible of synchronization recovery and if synchronization is lost, then the remaining operations will be of no usage. This information is encoded using ECC with higher error correction capability.

5 Data Recovery from Background Images

Despite its higher capacity as well as simplicity, the new technique is of any value only if it enables successful data recovery, whereas the size of the each and every isolated dot is 25% of the size used in existing techniques, e.g. HD-DataStripe. Also, the new technique has only one dot (primary dot) for recognition, unlike [11] where a certain pattern is identified to decode single bit of information. The strength of noise being encountered during printing and scanning process can be envisioned from the fact that an 18 by 18 size symbol pattern (means 324 dots) is used in [11] just to make the encoded symbol robust against the PS process. Now, the task of the data-reading algorithm is to identify each of the printed dots, especially the primary dots, used for information encoding. It finds the position (middle point) of the primary dots with respect to the neighboring reference dots and eliminates the noise introduced during the PS process. In order to develop a successful data recovery algorithm the underlying characteristics of the printing device being used are to be studied and taken into account in advance.

5.1 Data-Reading Algorithm

It was found by experiments with various Laser printers at 600 dpi printing resolution that errors were mainly caused by 1) unprinted primary dots, 2) inaccuracy in position and size of primary dots, and 3) noise in the neighborhood of primary dots. An inaccurately printed Symbol Pattern may result in an erroneously decoded primary dot, the center position of the symbol being ‘the critical area’.

Under the assumption that any good quality Laser printer should be able to print reliably at half of its resolution, data encoding and data reading were adjusted to printer characteristics.

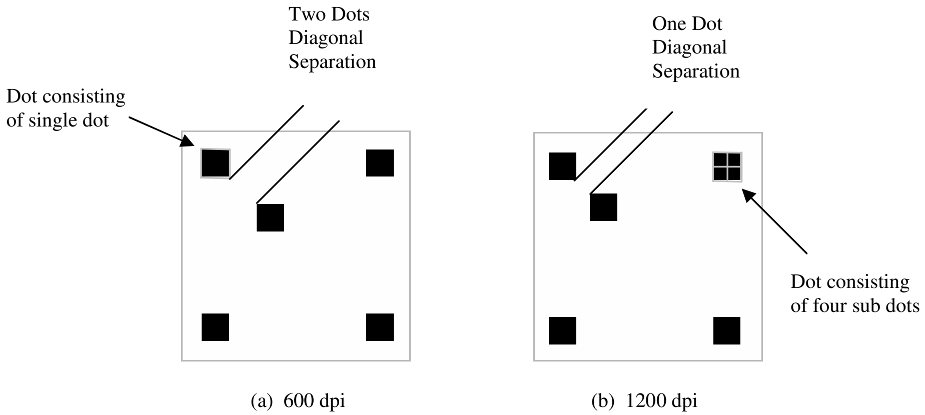


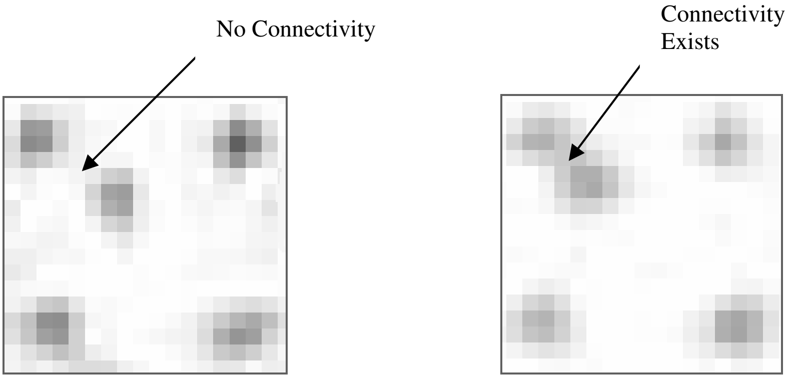
Fig. 7. Illustration of the size and diagonal-distance between the reference and primary dots for two Symbol Patterns: (a) Symbol Pattern is used at 600 dpi, (b) at 1200 dpi diagonal-distance is decreased one half while dots size is kept same

Higher Resolution: For further improvement in terms of higher capacity and lower error rate LaserJet printer offering resolution 1200 dpi are considered.

Primary Dot Position Modulation: The same background image quality as well as symbol pattern and dot size are used, but the image is printed using the device offering resolution 1200 dpi. Now, each dot is represented by four dots arranged in a 2 by 2 square. As shown in Fig. 7(a), at 600 dpi the minimum diagonal-distance between the primary dot and its nearest neighboring reference dot is one dot. Thus, at 1200 dpi this minimum diagonal distance is reduced to 50 %, as it is shown in Fig. 7(b).

This change in diagonal distance provides more robustness against the errors due to positional inaccuracy as well as unprinted dots. This is due to the fact that now the possibility that a primary dot is printed at critical region is further decreased, as the distance between the critical point and primary dot is increased by 50%. Furthermore, there is much lower possibility now that an isolated black dot is not printed. This is due to the fact that when two dots separated diagonally by only one dot are printed at 1200 dpi, then either primary dot is printed or at least some slightest hint (noise) is made (due to 50 % less distance between the primary and reference dots), which results in very slight connectivity as illustrated in Fig. 8. This slight connectivity can be utilized in information decoding process (e.g. while developing a filter for noise elimination).

This *slight connectivity* is attributed to the physical dot gain effects (i.e. error or inaccuracy due to the inability of printing device to precisely transfer the toner while making the primary and reference dots, separated by very small diagonal distance). The connectivity might also be resulting from heating and pressing process when toner is being fixed permanently to the paper surface.



(a) Unmodulated SP

(b) Modulated SP

Fig. 8. An illustration of connectivity after printing and scanning process for the symbol patterns (SP) shown in Fig. 6

5.2 Dot Amplitude Modulation

As mentioned above that at 1200 dpi resolution a dot printed at 600 dpi is represented by four dots, which has another advantage as well that it allows to vary primary dot size in 25% steps during data encoding process. This is called *Dot Amplitude Modulation* (DAM). The DAM may be used to improve the performance of relatively poor quality printing devices (e.g. older devices offering 1200 dpi resolution, but not able to print isolated dots) by varying the size of the primary dot so that it results in a minimum error.

In the data reading procedure the printed document is scanned at sufficiently higher over-sampling (at least twice) rate and then the data reading technique is applied. The data reading technique applies different filters that deal with synchronization recovery from noisy environment, noise elimination from data encoding region, and identification of information encoding dots.

The objective of synchronization recovery filter is to identify and accurately locate the dot 9 used for the synchronization recovery. A synchronization recovery dot suffers from dot gain effects caused by the up to four neighbouring data encoding dots 8 separated by 1/600 of an inch. The accuracy of the located position is 1/600 of an inch. If synchronization error is encountered (for instance due to overlapped symbols), the average value of next immediate neighbouring synchronization dots 9 is taken.

The filter dealing with information decoding process identifies and locates the position of a data encoding dot in the region, defined by the synchronization dots of three neighbouring data encoding symbols (and a fourth one for the symbol under consideration), while taking into account the noise from the print and scan process. It has to distinguish between different positions that are used by the data encoding symbol for encoding multiple data bits per symbol. It also takes measures (e.g. using slope of greyscale region) to decide when a given position lies within the critical region resulting from dot gain effects of the PS process. The accuracy required for successful decoding a given symbol is again 1/600 of an inch.

Once the information from all the data encoding symbols have been decoded, then the resulting data is converted to the bit stream corresponding to the doc file (encoded into the background image), while performing the operations (data compression, data encryption, error correction coding and data scrambling) in reverse order.

The data-reading algorithm proceeds as follows:

Printed CBGI is scanned as greyscale image at sufficiently higher resolution.

The position of each of the reference dots is estimated using three different filters referred to as Filter-1, Filter-2 and Filter-3. It is mentionable that in the present scenario the *reference dots* are noisier and require more sophisticated techniques to correctly identify their central points.

The first filter identifies only the isolated reference dots.

The second filter identifies the reference dots, which have been influenced from the neighboring information carrying dots (number of such dots vary from one up to four) and results in the noisy reference dots.

The third filter checks as well as corrects if there is an error in the estimated position of any of the reference dots, estimated using the previous filters.

Using Filter-4, identify and eliminate noise in the possible region where information might have been encoded.

Recover the encoded information by identifying the location of the primary dot and measuring its distance from each of the neighboring reference dots. These steps (applying Filter 1 to 4) are applied repeatedly until all the data has been recovered. For a detailed description of filter functions see [12].

6 Experimental Results: CBGI with and without Superposed Text

To check performance of the data recovery process of the novel data encoding technique an experiment was run on a HP LaserJet 4100 printer offering 1200 dpi resolution. A background image of 4200 by 3000 dots (at 600 dpi resolution) offering 700 kbits capacity (equivalent to 350,000 NSPs) was used. That image was transformed into 1200 dpi by representing each dot at 600 dpi by four dots at 1200 dpi. At 1200 dpi the minimum distance between the primary and the reference dots is changed to one dot (half of the diagonal distance at 600 dpi) with the aim that it results in minor connectivity between the reference and the primary dots which is helpful for noise characterization and elimination. Next, the 700 kbits data is encoded into the background image and multiple copies of the image are printed. The results of the data recovery after PS process are given in Table 1.

By comparing the new results with the existing work it can be seen that the new technique not only provides higher data encoding capacity but it also results in lower error rate as compared with [11] in which 90 errors are encountered for

Table 1. Number of errors encountered in data recovery process using novel data encoding technique in grayscale background images

Experiment No.	No. of errors	Experiment No.	No. of errors
1	15	6	21
2	20	7	8
3	18	8	8
4	17	9	14
5	17	**Average**	**15.33**

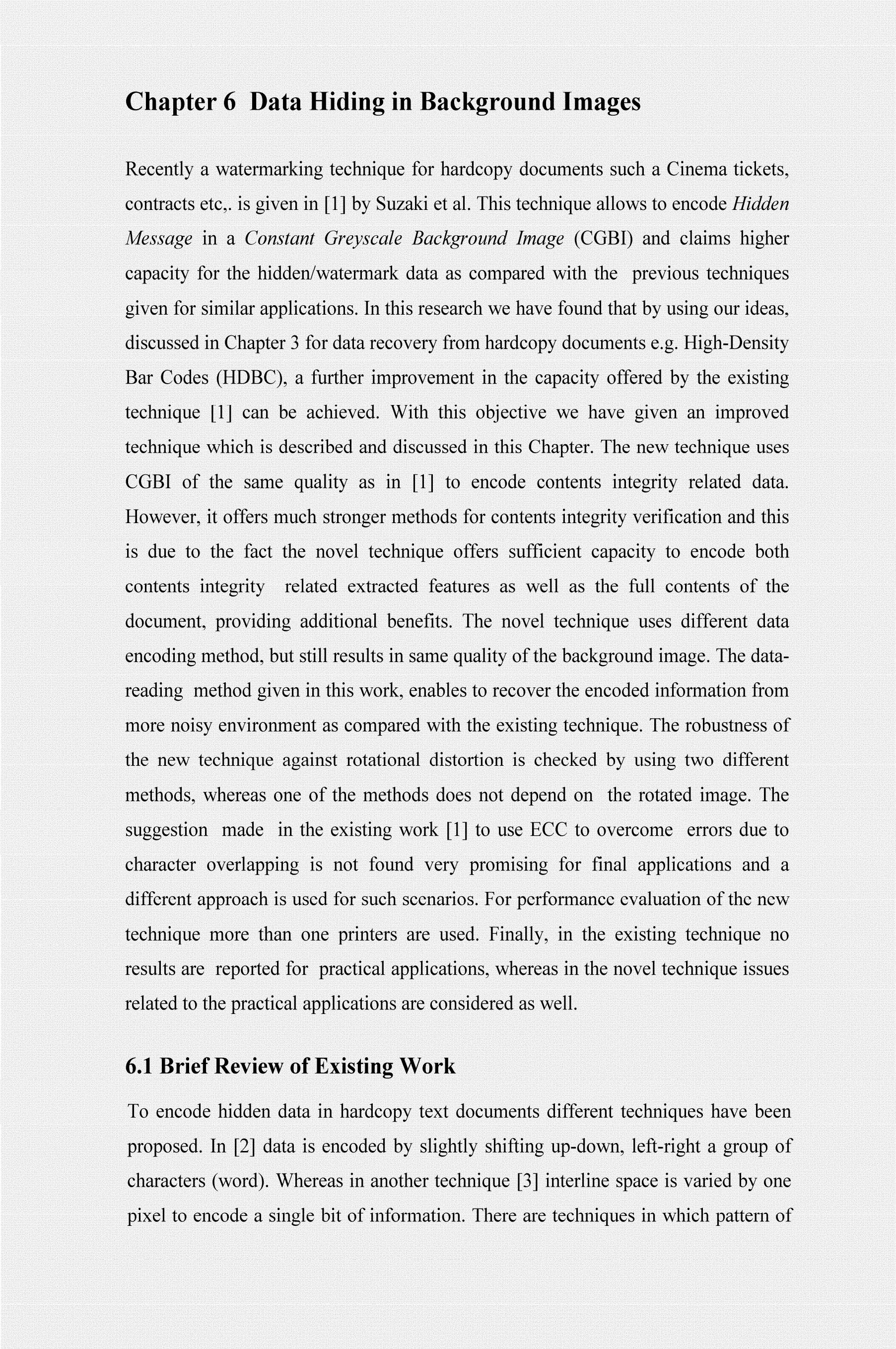
98,568 data encoding symbols. While considering a full A-4 page with printable area measuring 4629 x 6585 at 600 dpi, it results in 860,000 symbols or 1720 kbits of information as compared with the 95.5 kbits from [11]. It is to be pointed out that while considering *error rate*, in the existing work the error rate is given for symbol pattern, when only one type of pattern is encoded and printed in the whole image. In this research, the real information is encoded into the background image and the number of errors are reported in Table 1.

With two other sets of experiments the maximum capacity for error free encoding was checked. In the first set of experiments an A-4 size CBGI without superposed text image was considered. The data being encoded were generated using the MATLAB routines. Three different types of printers were considered for experimental purpose. To achieve the zero bit error rate, data is encoded using BCH ECC technique and then scrambled all over the CBGI of size 6675 by 4649 dots at 600 dpi. The capacity for user data with zero bit error rate (BER) along with used BCH parameters, are given in Table 2.

Table 2. Results for error-free recovery of encoded data from background image without superposed text

Exp. No.	Device	ECC	User payload kbits	Capacity-1 pages	Capacity-2 pages
1	HP 4100	BCH(511,493,2)	1626.9	53	77
2	HP 8150	BCH(511,455,6)	1508.1	49	72
3	HP 8000	BCH(511,455,6)	1508.1	49	72

Capacity-1: File-size for the number of text pages in word document (i.e. a .doc file).
Capacity-2: File-size for the number of pages with each page having 2,623 characters (8 bits per char), excluding space character.



Chapter 6 Data Hiding in Background Images

Recently a watermarking technique for hardcopy documents such a Cinema tickets, contracts etc,. is given in [1] by Suzaki et al. This technique allows to encode *Hidden Message* in a *Constant Greyscale Background Image* (CGBI) and claims higher capacity for the hidden/watermark data as compared with the previous techniques given for similar applications. In this research we have found that by using our ideas, discussed in Chapter 3 for data recovery from hardcopy documents e.g. High-Density Bar Codes (HDBC), a further improvement in the capacity offered by the existing technique [1] can be achieved. With this objective we have given an improved technique which is described and discussed in this Chapter. The new technique uses CGBI of the same quality as in [1] to encode contents integrity related data. However, it offers much stronger methods for contents integrity verification and this is due to the fact the novel technique offers sufficient capacity to encode both contents integrity related extracted features as well as the full contents of the document, providing additional benefits. The novel technique uses different data encoding method, but still results in same quality of the background image. The data-reading method given in this work, enables to recover the encoded information from more noisy environment as compared with the existing technique. The robustness of the new technique against rotational distortion is checked by using two different methods, whereas one of the methods does not depend on the rotated image. The suggestion made in the existing work [1] to use ECC to overcome errors due to character overlapping is not found very promising for final applications and a different approach is used for such scenarios. For performance evaluation of the new technique more than one printers are used. Finally, in the existing technique no results are reported for practical applications, whereas in the novel technique issues related to the practical applications are considered as well.

6.1 Brief Review of Existing Work

To encode hidden data in hardcopy text documents different techniques have been proposed. In [2] data is encoded by slightly shifting up-down, left-right a group of characters (word). Whereas in another technique [3] interline space is varied by one pixel to encode a single bit of information. There are techniques in which pattern of

Fig. 9. Example of an A-4 size CBGI in the presence of superposed text

Table 3. Error-free data recovery results for the encoded data in the background image with superposed text

Exp. No.	Device	ECC	User payload kbits	Capacity-1	Capacity-2
1	HP 4100	BCH(511,493,2)	1068.33	33	50
2	HP 8150	BCH(511,455,6)	1009.36	31	48
3	HP 8000	BCH(511,455,6)	> 909.36	27	43

It can be seen that the capacity varies between 1508.1-1626.9 kbits, which is sufficient to encode a word document file having 49-53 pages of text. Even the higher number of pages 72-77 with text contents can be encoded when only text information rather than ".doc" file is encoded. This capacity is more than 16 times higher compared to existing work. Trivial to state that capacity can be further increased using state of the art compression techniques for text data. The error correction capability of BCH in each experiment is determined by the underlying printing device, whereas the codeword length is selected based on the processing time to encode message. Concerning data scrambling, it was found that it improves the capacity significantly. Data-reading time, excluding BCH decoding and data de-scrambling time, amounted to 15 minutes, which is quite high. Replacing MATLAB routines by dedicated software would reduce these times considerably.

For the second set of experiments data is encoded in an A-4 size CBGI in the presence of superposed text, shown in Fig. 9 given in appendix. Three different printers were considered and the results are given in Table 3. Capacity varies between 909.36 and 1068.33 kbits, which is sufficient to encode a word file having 27-33 pages of text. The higher number of pages 43-50 with text contents can be encoded when only text information rather than .doc file is considered. The variation in capacity is again associated with the printing device under consideration. As mentioned earlier, a further gain (i.e. a few hundred kbits) in capacity can be achieved by considering all those rows in which there are relatively minor number of errors caused by character overlapping, unlike the present scenario in which rows with non-overlapping symbols were considered only. This gain in capacity is much higher than that offered by the existing work. Unlike prior art, it eliminates the need for any feature extraction to verify the contents allowing one-to-one content authentication. The higher capacity gives a direction to investigate novel applications for the hardcopy documents, some applications are described in [14]. In this scenario data-reading time, excluding BCH decoding and data de-scrambling time, amounted to 10 minutes. The lower data-reading time in this case is due to the fact that overlapping regions are not considered in data-reading process, as these regions have been taken into account during the data encoded process (discussed before). Same remarks hold for Capacity-1 and Capacity-2 in Table-3 as in Table-2.

7 Conclusion

A new approach to high-capacity invisible background encoding for digital authentication of hardcopy documents (e.g. contracts, official letters etc.) has been described. It allows full-content of the foreground text to be encoded into the superposed background image in machine-readable format. Before encoding the content the following operations can be applied, (1) data compression, (2) ECC against unavoidable errors, (3) data scrambling, and (4) data encryption. It is ideal for digital authentication of languages with complex writing structures. For those languages it is very difficult to develop OCR-technology. In absence of data compression file size of one-page of, e.g. Arabic text due to the underlying language structure, is much higher than languages with Roman characters. The capacity offered by the superposed background image is sufficient to encode at least few copies of full foreground text.

The individual data encoding symbols are completely imperceptible and do not affect the aesthetic appearance of the document. Visual inspection of the superposed background image does not give any indication about the existence of encoded data in background image. The higher data encoding capacity and visual quality are attributed to smaller data encoding symbol size, data encoding symbol pattern and synchronization recovery mechanism and the data-reading technique.

The data-reading technique takes the scanned image, which is over-sampled at least twice the printing resolution, as input and recovers the encoded contents from the scanned document image. It handles intentional /unintentional skewing distortion and noise encountered from the print-and-scan process [12]. Almost all existing scanning devices in market satisfy the over-sampling constraint.

The underlying characteristics: invisibility, higher data encoding capacity, and blind-data decoding capability of proposed technique from superposed background image make it very attractive for military communications and other government departments due to the fact that digital communication is not applicable in all scenarios (e.g. the conventional cryptographic techniques make the digital communication suspicious).

Acknowledgements. The authors like to thank Dr. Stefan Häckel who, while preparing patent submission based on Taswar Iqbal's Ph.D. thesis, helped to clarify many facts and formulations by constant penetrating enquiry.

References

1. Alattar, A.M., Alattar, O.M.: Watermarking electronic text documents containing justified paragraphs and irregular line spacing. In: SPIE, Security and Watermarking of Multimedia Contents VI, San Jose, CA, USA, vol. 5306 (2004)
2. Bern, M., Breidenbach, J., Goldberg, D.: Trustworthy paper documents. In: Information Hiding Workshop, Pittsburgh, PA, April 2001, pp. 25–27 (2001)

3. Brassil, J.T., Low, S., Maxemchuk, N.F., O'Gorman, L.: Electronic marking and identification techniques to discourage document copying. IEEE Journal on Selected Areas in Communication 13(8), 1495–1504 (1995)
4. Brassil, J.T., Low, S., Maxemchuk, N.F.: Copyright protection for the electronic distribution of text documents. Proc. of the IEEE 87(7), 1181–1196 (1999)
5. Low, S.H., Maxemchuk, N.F.: Performance comparison of two text marking methods. IEEE Transaction on Selected Areas in Communication (1998)
6. Low, S.H., Maxemchuk, N.F.: Capacity of text marking channel. IEEE Signal Processing Letters 7(12), 345–347 (2000)
7. Motwani, R., Breidenbach, J.A., Black, J.: Collocated Dataglyphs for Large Message Storage and Retrieval. In: SPIE, Security and Watermarking of Multimedia Contents VI, San Jose, CA USA, vol. 5306 (2004)
8. Brewington, G.T.: System and method for providing hardcopy secure documents and for validation of such documents, Xerox Corp., Pub. No. US. 2004/0117627 AI, Pub. date, June 17 (2004)
9. Hecht, D.: Embedded DataGlyph Technology for Hardcopy Digital Documents. In: Proc. of SPIE Color Hard Copy and Graphic Arts III, February 1994, vol. 2171 (1994)
10. Hilton, D., Tan, W., Wells, P.: Document printed with graphical symbols which encode Information Enseal Systems Limited (GB). US Patent 6871789 (2005)
11. Suzaki, M., Mitsui, Y., Suto, M.: New alteration detecting technique for printed documents using dot pattern watermarking. In: SPIE 2003, Security and Watermarking of Multimedia Contents V, San Jose, CA, USA (2003)
12. Iqbal, T.: High Capacity Analog Channels for Smart Documents., Ph.D. thesis, University Duisburg-Essen, Germany, Fac. of Engineering (2006)
13. Zhao, J.: Digital authentication with digital and analog documents, MediaSec technologies GmbH (DE), US Patent No. 6,751,336 (June 2004)
14. Iqbal, T., Geisselhardt, W.: Doument with Encoded Portion, Int. Patent PCT/EP2007/006126, filed on March 2, 2007, Essen, Germany (2007)
15. Jordan, Kutter, M.: Method for Robust Asymmetric Modulation Spatial Marking with Spatial Sub-Sampling., Alpvision SA, Pub. No. US 2006/0147082 AI, Pub. date, July 6 (2006)

Reversible Quantization-Index Modulation Using Neighboring Correlation

Chin-Chen Chang[1,2] and Wen-Chuan Wu[2]

[1] Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan, R.O.C.
ccc@cs.ccu.edu.tw

[2] Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi 621, Taiwan, R.O.C.
wenn@cs.ccu.edu.tw

Abstract. In this paper, a novel reversible data hiding scheme, which can recover the original information without any distortions after the hidden secret data have been extracted, based on quantization-index modulation (QIM) technique is presented. This proposed scheme utilizes the local spatial correlation of neighboring quantized VQ indices to conceal secret data under image coded stream and achieve data reversibility simultaneously. As for some not-well-correlated indices, this embedding system burdens the quantized codes with additional indicators for the hereafter recovery. Experimental results proved that the proposed scheme is well furnished with the lossless recovery facility than other QIM-based data hiding schemes.

Keywords: Image hiding, reversibility, spatial correlation, VQ compression.

1 Introduction

Over the last decade, a great many hiding methods [1,3,4,9] have been widely investigated to conceal important data in still images. Intrinsically, the embedding process itself is an injury to cover images and also causes the negative effects to cover images. The more the quantity of secret data is hidden, the more quality degradation would be yielded. In [5], it pointed out the significance of data reversibility, especially for the medical and military imaging applications. The so-called reversibility is considered a technique to perfectly reverse stego-images back to original cover ones after the hidden data have been extracted out. Subsequently, more and more researchers tended to develop a reversible (or called lossless) data embedding approach, such as [6,7,8]. However, these methods mentioned as above which achieve perfect restoration of cover signals were designed for the raw format of images. In fact, digital images are generally stored or transmitted in the compression format. Here comes a serious concern that: The hidden information inside it will be lost when the stego-image is encoded afterwards by any one lossy compression algorithm. Therefore, reversible

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 222–232, 2008.

data hiding methods processed on spatial pixel domain are neither practical nor sensible [10].

Vector quantization (VQ) [11] is a kind of lossy compression technique frequently used in gray-scale images. Throughout the block coding, VQ system successfully translates an image into the small-scale form of indices which represent the relationship between image blocks and code vectors. Hence, its effectiveness and success depends upon the used codebook $C = \{c_0, c_1, ..., c_{N-1}\}$, where c_i is the i-th code vector of k dimensions and N is the size of the codebook. Because of the limitation of code vectors in C used to meet all possible blocks (i.e. $N << 256k$), it is obviously that VQ compression might cause image distortion. In order to enhance the robustness of the hidden data, quite a few data hiding methods, such as [10,12,13,14,17], were put forward based on quantization-index modulation technique (QIM).

In brief, the policy of QIM is to embed secret data by directly modulating the coded stream of an image. In 2002, Jo and Kim addressed the paired index modulation method [13] (PQIM) which separates the codebook C into three groups C_0, C_1, and C_{-1}. In which, all code vectors in groups C_0 and C_1 are mutually paired by two members from C_0 and C_1, respectively, and each pair of code vectors is provided for alternative modification according to the secret bit to be embedded. In the cause of high payload, later, an adaptive clustered QIM method [14] integrated with Cartesian product technique was introduced by Du and Hsu to do the index substitution. These quantization-index modulation methods are pretty easy and realistic, but they are all irreversible. Only the literature in [10] is presently capable of holding the distortion-free property. Its reversible embedding operation is based on the modulation of side-match VQ quantized indices. In this paper, we attempt to develop a reversible data hiding method based on VQ index modulation. The rest of this paper is organized as follows. Section 2 briefly reviews some previous works. Section 3 describes the details of our new lossless data embedding method for VQ images. And further, Section 4 evaluates the experimental results for proving our proposed scheme. Finally, conclusions are summarized in Section 5.

2 Related Works

In order to further improve the compressed bit rate of VQ, Kim proposed the side-match vector quantization method (SMVQ) [15], a variation of VQ system, in 1992. SMVQ compression is a memory coding technique, in other words, it requires keeping the pixel intensities of neighboring encoded blocks for the current block's coding. The detailed concept of SMVQ is presented as follows. For the current block to be encoded, two neighboring blocks (i.e. the upper and left adjacencies) are first given to forecast some pixel intensities inside it. As shown in Figure 1, pixels l_3, l_7, l_{11}, and l_{15} of block L as well as pixels u_{12}, u_{13}, u_{14}, and u_{15} of block U are used to temporarily assign partial pixels x_0, x_1, x_2, x_3, x_4, x_8, and x_{12} of the un-encoded block X, where the 4×4 blocks U and L

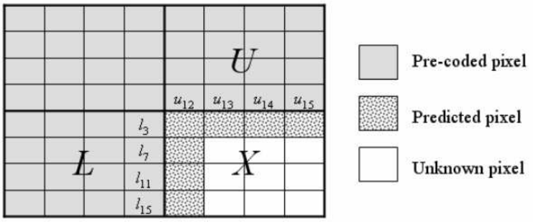


Fig. 1. An example of SMVQ prediction

colored gray have been already encoded and reconstructed by the selected code vectors.

Next, some code vectors that are very similar to the block X with predicted and unknown pixels are selected out from the master codebook C to create a unique state codebook SC. Here, we define the metric of similar matching for a code vector c_i as the side-match distortion $SMD(c_i)$, where $SMD(c_i) = UD(c_i)+LD(c_i)$ The upper distortion $UD(c_i)$ between the code vector c_i and the upper block U is displayed in Equation (1); Likewise, the left distortion $LD(c_i)$ between the code vector c_i and the left block L is displayed in Equation (2). These n code vectors with the smallest side-match distortions are collected in order to build a small state codebook. From this codebook SC, finally, we search for the best-matched code vector to the original block X by the real distortion distance, and then use the index of the found one to encode it.

$$UD(c_i) = \sum_{j=12}^{15} (u_j - c_{ij}). \tag{1}$$

$$LD(c_i) = \sum_{\substack{j=0 \\ s=4\times j+3}}^{3} (l_s - c_{is}). \tag{2}$$

For the sake of subsequent prediction, the blocks in the first row from the top and the first column from the left of the image must be encoded ahead by original VQ. The other blocks in the image are encoded by SMVQ instead. Nevertheless, this way might suffer from the derailment problem in the encoded quality when there are many low correlations among image blocks. Later, many literatures were presented to improve the encoded quality of the original SMVQ. For example, the simplest two-pass SMVQ method and the four-sided SMVQ method. In two-pass SMVQ, a pre-defined threshold is given to determine if a block is complicated enough to be encoded by VQ instead of SMVQ [17]. Thus, an extra bit is needed for the future decoding as the left example of Figure 2. Additionally, in four-sided SMVQ, the predicted blocks are fixedly interlaced within the coded stream. As shown in the right example of Figure 2, the block X in the four-sided SMVQ is predicted by four neighboring and encoded blocks U, L, R, and D. By this way, the accuracy of prediction will be greatly improved.

In 2005, Chang et al. applied this four-sided SMVQ algorithm to develop a reversible quantization-index modulation method [10], FSSM-QIM for short. Because the blocks encoded by VQ partake in the prediction of SMVQ blocks, FSSM-QIM method does not modify their codes to embed any secret data regarding the precision perspective. It only hides secret data into the compressed indices of the predicted blocks. Index modulation can be directly enforced, but the modification must satisfy data reversibility. Here, FSSM-QIM exploits side-match distortion as a clue to explicitly guide the decoder to extract the embedded data and recover the original compressed codes.

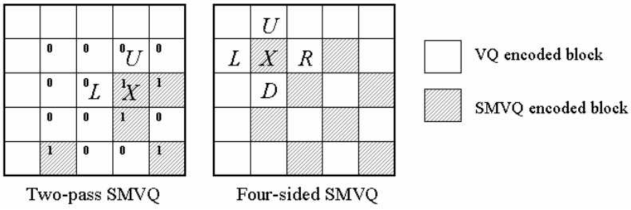


Fig. 2. Example encoded by two-pass SMVQ and four-side SMVQ

3 The Proposed Scheme

Once the better approximations of image blocks fail to be found in state codebook, FSSM-QIM method actually yields bad encoded quality. Besides that, it merely embeds the small amount of secret data in an image. In order to reform the encoded quality and increase the embedding capacity, we shall propose a reversible modulation method based on VQ quantized indices. The proposed method consists of three procedures, the pre-processing, data embedding, and data extraction and recovery procedures, and the details of these procedures are described in Subsections 3.1, 3.2, and 3.3, respectively.

3.1 Pre-processing Procedure

Two essential works should be pre-processed before the data embedding. One is the encryption of the secret data. Concerning about the security factor, we encrypt the secret data S ahead by certain one cryptographic system, such as DES and RSA, to prevent attackers' intercept. Further, the encrypted secret data, named $S^e = (b_1, b_2, \ldots, b_t)$, is turn into a smaller bit stream of length t, where $b_i \in \{0, 1\}$ and $1 \leq i \leq t$. The other work is the organization of VQ codebook. Like the previous VQ compression method, VQ codebook was designed by using some training images through the well-known LBG iterative algorithm [11]. The only difference between former and present is the number of code vectors in the codebook. Here, we only demand $N-2$ genuine code vectors instead of N. The major purpose is to reserve certain positions (i.e. index 0 and index $N-1$) in the codebook for the sign indication in the recovery process.

After acquiring the VQ codebook of $N - 2$ code vectors, it must be further done to sort the codebook by applying principal components analysis technique (PCA) [16]. The power of PCA is to project high-dimensional data onto a lower-dimensional space while still maximally holds the variation among the original input data. Hence, we utilize this property to sort the codebook in one-dimensional space such that the code vectors with similar contents will be closer together and otherwise will be farther away from one another. This way will facilitate our later embedding process to achieve reversibility. Here, we nominate the codebook of $(N - 2)$-sorted code vectors as $C' = \{c'_1, c'_2, \ldots, c'_{N-2}\}$, where two positions situated at the first and last of C' are preserved. This codebook C' is then used in what follows below.

3.2 Data Embedding Procedure

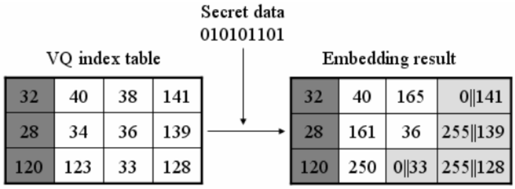
The goal of the data embedding procedure is to utilize the codebook C' produced above to carry secret data into the VQ coded stream by a reversible way. Actually, the reversible embedding can be regarded as a transformation of one-to-one mapping. This mapping will constitute a unique relation between elements from two sets respectively. In mathematics, an element a of set A can be only homologized to an element b in set B by the one-to-one mapping function F and its invertible function F^{-1} also can correctly map b of set B to element a of set A, i.e. $F(a) = b$ and $F^{-1}(b) = a$, where $a \in A$ and $b \in B$. By such a reversible function F, we can not only hide secret data certainly but also recover the original cover data. How to, however, design such a one-to-one mapping function for data embedding?

Here, we exploit the smooth property of images to define a reversible function and make it feasible in the embedding. Each natural image mostly possesses the local spatial correlation, that is to say, the neighboring positions in image pixels have the similar amplitude contents. In order to still keep this property in the coded stream, we thereby sort VQ codebook in order such that VQ indices of the adjacent image blocks are strongly related to each other after encoded by VQ compression. This natural phenomenon would highly facilitate the future data recovery. In the proposed embedding procedure, we define a reversible function F as the following equation by the idea that elements a and b are as large as possible: $F(a) = \{\ b | b = ((a - 1 + |C'|/2) \bmod |C'|) + 1\}$, where $|z|$ means the number of genuine elements in z, and a as well as b being the elements of sets A and B, respectively, belong to $\{c'_1,\ c'_2,\ \ldots,\ c'_{N-2}\}$.

Go further detailed, assume that two adjacent image blocks of 4×4 pixels are B_1 as well as B_2 and their VQ indices through C' are idx_1 and idx_2, respectively. When embedding secret bit b ($b = \{0, 1\}$) into the value idx_2, we must first compute its transformed value idx'_2 by the above mapping function F, i.e. $idx'_2 = ((idx_2 - 1 + |C'|/2) \bmod |C'|) + 1$. If $D(idx_2, idx_1)$ is smaller than $D(idx'_2, idx_1)$, then the VQ index idx_2 is replaceable; Otherwise, idx_2 is non-replaceable, where $D(\alpha, \beta) = |\alpha - \beta|$ is the absolute distance between positive integers α and β. For the foregoing two cases, their embedding rules to secret bit b are stated as follows:

(1) If idx_2 is replaceable and b equals to 0, value of idx_2 will not be modified.
(2) If idx_2 is replaceable but b equals to 1, replace idx_2 with index value idx'_2.
(3) If idx_2 is non-replaceable, replace idx_2 by (BIN(b)||idx_2), where notation "||" represents data concatenation and BIN(b) is the value formed by the consecutive $\lceil log(N-2) \rceil$-bit b.

Intensively studying, we extend this instance of two VQ indices to a row of sequential m indices $\{idx_1, idx_2, \ldots, idx_m\}$ in an image. Because the modulation of latter index is based on the original value of former one, it means clearly that the first one (i.e. idx_1) in these m indices can not be altered. In other words, it must be kept primitive without carrying any data. Other $m-1$ indices are, instead, used to embed the secrets. For the most part, the number of indices which belong to replaceable case will be large when the image is almost even and smooth. The non-replaceable indices will be also occurred yet in small quantity. In order to distinguish between these two cases and carry data simultaneously in the non-replaceable indices, we exploit the preserve indices of codebook C' to indicate as the Rule (3) above.



VQ index table

32	40	38	141
28	34	36	139
120	123	33	128

Embedding result

32	40	165	0||141
28	161	36	255||139
120	250	0||33	255||128

Fig. 3. An example of the proposed method of embedding secret data '010101101'

Figure 3 depicts a simple example to clearly explain the proposed embedding method. In this example, the amount of genuine code vectors used in the codebook C' is 254 in all. A VQ index table shown in the left side of Figure 3 is used to load with secret data '010101101' and then turned into the result in the right side of Figure 3, where the value deeply colored represents the one without carrying any data, the value lightly colored represents the one being in the non-replaceable case, and the others are naturally in replaceable case. Initially, the first index in each row of VQ table is not modified. For the second index '40' of the first row, next, we want to embed the first secret bit '0'. Since it is in the replaceable case, Rule (1) is applied to maintain the value '40'. Similarly, the third index '38' used to embed the second secret bit '1' is also replaceable, hence we replace it with the value '165' by the application of Rule (2). After that, in the case of value '141' being non-replaceable, we append the code of index '0' to it in order to embed the third secret bit '0' according to Rule (3). Repeating the operations similar as above to the indices of the remaining rows, finally, the embedding result is thereby produced as that in Figure 3. Here, it is remarkable that the size of the resulted coded stream will become bigger, where each replaceable index occupies $\lceil log(N-2) \rceil$ bits for the embedding while each non-replaceable index costs $2 \times \lceil log(N-2) \rceil$ bits.

3.3 Data Extraction and Recovery Procedure

The goal of the data extraction and recovery procedure is to take the hidden secret data out from the index stream and then restore the original VQ compressed image. Before starting it, the decoder must derive the two materials identical to that of the encoder namely the sorted codebook C' and the cover object's width size. Here, we suggest sending and receiving the two materials above by a confidential look. Furthermore, secret data extraction is performed with the recovery process. These two working are actually the reverse of the proposed embedding procedure. Overall speaking, the spatial correlation among adjacent indices and the two signs (0 as well as $N-1$) can be explicitly applied to imply the carried data. To restore the origins of replaced indices, further, just transform all replaceable indices with secret bit '1' back through the reverse function F^{-1} and then the original indices of VQ compressed image can be recovered without any distortion. In which, F^{-1} deduced from the function F is as that: $F^{-1}(b) = \{a|a = ((b-1+|C'|/2) \bmod |C'|) + 1\}$. The detailed steps for the extraction and recovery are stated below.

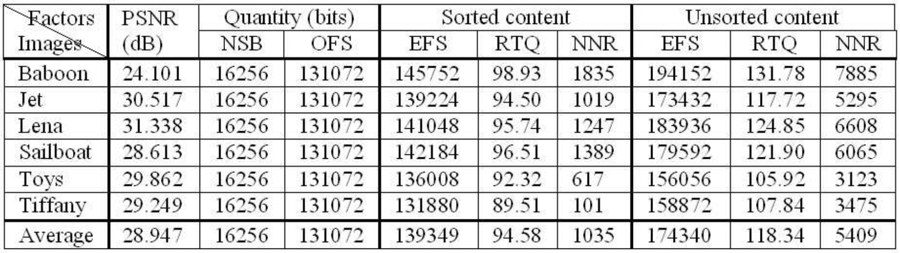
Step 1: Skip the first index, which not participate in the embedding process, in each row.
Step 2: Select the next index and let it as well as its former index be $\overline{idx_2}$ and $\overline{idx_1}$, respectively.
Step 3: If $\overline{idx_2}$ is unequal to any preserved code, then it belongs to the replaceable case; otherwise, it belongs to the non-replaceable case.
Step 4: If replaceable and $D(\overline{idx_2}, \overline{idx_1})$ is smaller than $D(F^{-1}(\overline{idx_2}), \overline{idx_1})$, then it implies that the secret bit is '0' and the original VQ index is $\overline{idx_2}$.
Step 5: If replaceable but $D(\overline{idx_2}, \overline{idx_1})$ is larger than $D(F^{-1}(\overline{idx_2}), \overline{idx_1})$, then it implies that the secret bit is '1' and the original VQ index is $F^{-1}(\overline{idx_2})$.
Step 6: If non-replaceable and $\overline{idx_2}$ is equal to 0 (or $N-1$), then it implies that the secret bit is '0' (or '1') and the original VQ value is the following index of $\overline{idx_2}$.
Step 7: Repeat Steps 2 to 6 above until all the indices are processed.
Step 8: Decrypt the extracted data to derive the original secret data S and reconstruct the VQ compressed image by the codebook C'.

4 Experimental Results

In this section, a variety of experiments were conducted to show the performance of the reversible VQ index modulation method we presented. Experimental fidelity is supported by embedding a random bit stream into standard images (such as Baboon, Jet, Lena, Sailboat, Toys, and Tiffany), each of which has a resolution of 512 × 512 pixels of 256 gray levels. These test images as the cover objects were divided into 16384 blocks of 4 × 4 pixels for VQ encoding. Here, four VQ codebooks with sizes of 126, 254, 510 and 1022 used were all acquired by the LBG training algorithm [11], and each code vector in the codebooks was the one with 16 dimensions (i.e. $k = 16$).

Before starting the test simulations, these codebooks just produced need to be sorted in ascendant order by PCA technique [16] to ensure our method is effective. The effect of additional sorting process will be also presented in the following experimental results. In the first simulation, we attempted analyzing the power of the proposed method among various factors, such as cover objects, codebook sizes, and codebook content, and these results were displayed in Table 1. In which, NSB represents the number of embedded secret bits, OFS represents the original file size, EFS means the file size after embedding, NNR denotes the number of non-replaceable indices, and RTQ denotes the ratio of real transmitted quantity (i.e. RTQ = EFS / (OFS + NSB)).

Table 1. Results of the proposed method using the codebook of 254 code vectors

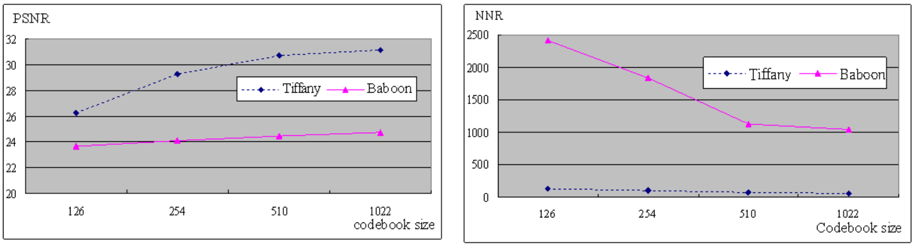


Factors / Images	PSNR (dB)	Quantity (bits)		Sorted content			Unsorted content		
		NSB	OFS	EFS	RTQ	NNR	EFS	RTQ	NNR
Baboon	24.101	16256	131072	145752	98.93	1835	194152	131.78	7885
Jet	30.517	16256	131072	139224	94.50	1019	173432	117.72	5295
Lena	31.338	16256	131072	141048	95.74	1247	183936	124.85	6608
Sailboat	28.613	16256	131072	142184	96.51	1389	179592	121.90	6065
Toys	29.862	16256	131072	136008	92.32	617	156056	105.92	3123
Tiffany	29.249	16256	131072	131880	89.51	101	158872	107.84	3475
Average	28.947	16256	131072	139349	94.58	1035	174340	118.34	5409

According to the outcome presented in this table, it is obvious that the proposed method using the sorted codebook brought fewer non-replaceable indices than that using the unsorted codebook. In such a scenario, hence, the size of VQ index table after embedding (i.e. EFS) is growing small-scale by using the sorted codebook, and the modified table still is capable of depicting the secret data (NSB) and the original table (OFS). That is to say, a bit of network bandwidth is utilized to pass the same information on while the sorted codebook is used. The reason the sorting process achieved is that the neighboring blocks of images have very alike contents as mentioned earlier and the closer VQ indices in those can raise the possibility of making replaceable blocks. Therefore, this way produces a small amount of extra data and the resulted table size is relatively small.

In addition, different images adopted in the proposed method also caused distinct results in terms of EFS and NNR. Observing the result of arranging NNR in ascendant order in Table1, we found that smooth images (Tiffany and Toys for example) have fewer non-replaceable indices than the textured ones (Baboon for example). It is also caused as a result of the introduction of the local spatial correlation. Hence, the table size in image Tiffany after embedding is the least of all. The proposed method embedded the fixed quantity of secret data regardless of what kind of image it is, but a smooth image is especially helpful to reduce the amount of the transmitted data.

Considering the codebook, we found that its size affects the value of NNR. The larger the codebook size is, the smaller the value of NNR would be yielded,



(a)
Average encoded quality

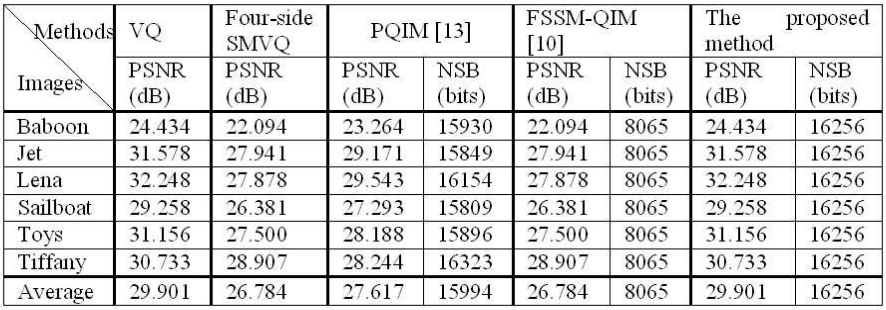
(b) Average number of non-replaceable indices

Fig. 4. Comparisons of PSNR and NNR to different codebook sizes among images 'Baboon' and 'Tiffany'

and vice versa. Because a codebook with large size has many code vectors to choose so that the distance between any two adjacent indices within the smooth area also becomes small, the number of non-replaceable indices is relatively few. In particular, this phenomenon is more noticeable in textured image as shown in Figure 4, where Figure 4 is the comparisons of PSNR and NNR to different codebook sizes among images 'Baboon' and 'Tiffany'. As what can be seen, image 'Baboon' is strongly affected by the codebook size compared to 'Tiffany'. However, the large-scale codebook would demand excessive encoding cost even though it contributes to the quality of cover image and the transmitted quantity. In conclusion, we suggest only small-scale codebooks, such as 126 and 254, and smooth cover images are used for conducting our proposed method.

In addition to the discussions about the proposed method itself, we also conducted the second simulation which was compared with other QIM methods, such as PQIM [13] and FSSM-QIM [10]. The experimental results were shown in Table 2 while the codebook size used is 510. Evidently, Table 2 revealed that PQIM method lacks for data recovery ability by contrast to other two

Table 2. Comparative results of the three QIM methods when the codebook size is 510



Methods / Images	VQ	Four-side SMVQ	PQIM [13]		FSSM-QIM [10]		The proposed method	
	PSNR (dB)	PSNR (dB)	PSNR (dB)	NSB (bits)	PSNR (dB)	NSB (bits)	PSNR (dB)	NSB (bits)
Baboon	24.434	22.094	23.264	15930	22.094	8065	24.434	16256
Jet	31.578	27.941	29.171	15849	27.941	8065	31.578	16256
Lena	32.248	27.878	29.543	16154	27.878	8065	32.248	16256
Sailboat	29.258	26.381	27.293	15809	26.381	8065	29.258	16256
Toys	31.156	27.500	28.188	15896	27.500	8065	31.156	16256
Tiffany	30.733	28.907	28.244	16323	28.907	8065	30.733	16256
Average	29.901	26.784	27.617	15994	26.784	8065	29.901	16256

methods. The qualities of cover images were proportionally distorted due to the embedding. For FSSM-QIM, the number of embedded secret data is relatively smaller than that of the proposed method although it works reversibility. In brief, the proposed method not only performs well, but also makes the modulation reversible.

5 Conclusions

In this paper, a new reversible data modulation method is proposed for VQ compressed images. The new method utilizes the property of natural images to artfully carry the secret data by modifying the original VQ indices. In order to further recover the origin without any distortions after extracting, the size of the modified index table was set slightly larger than the previous one. Simulation results proved that the use of the local spatial correlation among indices makes data recovery practicable and it effectively decreases the actually transmitted quantity, especially for the sorted codebook. Moreover, the new method is superior to other index modulations in terms of data reversibility and the capacity. In the future, we intend to investigate the extension of our proposed method to carry more secret bits in each VQ index.

References

1. Chang, C.C., Wu, W.C.: A Novel Data Hiding Scheme for Keeping High Stego-Image Quality. In: Proceedings of the IEEE 12th International Conference on Multi-Media Modelling, Beijing, China, pp. 225–232 (2006)
2. Wang, R.Z., Chen, Y.S.: High-Payload Image Steganography Using Two-Way Block Matching. IEEE Signal Processing Letters 13(3), 161–164 (2006)
3. Chang, C.C., Hsiao, J.Y., Chan, C.S.: Finding Optimal LSB Substitution in Image Hiding By Dynamic Programming Strategy. Pattern Recognition 36(7), 1583–1595 (2003)
4. Chan, C.K., Cheng, L.M.: Hiding Data in Images by Simple LSB Substitution. Pattern Recognition 37(3), 469–474 (2004)
5. Tian, J.: Reversible Data Embedding Using a Difference Expansion. IEEE Transactions on Circuits and Systems for Video Technology 13(8), 890–896 (2003)
6. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless Generalized-LSB Data Embedding. IEEE Transactions on Image Processing 14(2), 253–266 (2005)
7. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible Data Hiding. IEEE Transactions on Circuits and Systems for Video Technology 16(3), 354–362 (2006)
8. De Vleeschouwer, C., Delaigle, J.F., Macq, B.: Circular Interpretation of Bijective Transformations in Lossless Watermarking for Media Asset Management. IEEE Transactions on Multimedia 5(1), 97–105 (2003)
9. Liu, S.H., Chen, T.H., Yao, H.X., Gao, W.: A Variable Depth LSB Data Hiding Technique in Images. In: Proceedings of the IEEE 3rd International Conference on Machine Learning and Cybernetics, Shanghai, China, vol. 7, pp. 3990–3994 (2004)
10. Chang, C.C., Tai, W.L., Lin, C.C.: A Reversible Data Hiding Scheme With Modified Side Match Vector Quantization. In: Proceedings of the IEEE 19th International Conference on Advanced Information Networking and Applications, Taipei, Taiwan, vol. 1, pp. 947–952 (2005)

11. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic Publishers, Boston (1992)
12. Chang, C.C., Wu, W.C.: Hiding Secret Data Adaptively in VQ Index Tables. IEE Proceedings on Vision, Image and Signal Processing 153(5), 589–597 (2006)
13. Jo, M., Kim, H.D.: A Digital Image Watermarking Scheme Based on Vector Quantisation. IEICE Transactions on Information and Systems E85-D(6), 1054–1056 (2002)
14. Lu, Z.M., Sun, S.H.: Digital Image Watermarking Technique Based on Vector Quantization. Electronics Letters 36(4), 303–305 (2000)
15. Kim, T.: Side Match and Overlap Match Vector Quantizers for Images. IEEE Transactions on Image Processing 1(2), 170–185 (1992)
16. Chang, C.C., Lin, D.C., Chen, T.S.: An Improved VQ Codebook Search Algorithm Using Principal Component Analysis. Journal of Visual Communication and Image Representation 8(1), 27–37 (1997)
17. Chang, C.C., Wu, W.C.: A Steganographic Method for Hiding Secret Data Using Side Match Vector Quantization. IEICE Transactions on Information and Systems E88-D(9), 2159–2167 (2005)

High Capacity Reversible Data Hiding for 3D Meshes in the PVQ Domain

Zhe-Ming Lu[1,2] and Zhen Li[2]

[1] Media Processing and Communication Research Center, School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, P.R. China
zhemingl@yahoo.com

[2] Department of Electronic and Information Engineering, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, P.R. China
lizhen007_0@hotmail.com

Abstract. In the digitized world nowadays, digital content (audio, images, video, 3D meshes, etc.) can be easily copied, manipulated and distributed. Copyright protection and integrity verification of digital content have become urgent problems for multimedia owners and distributors. In addition, to alleviate bandwidth requirements, vector quantization (VQ) is an effective and popular vertex data compression technique for triangle meshes. In this paper, we present a new data hiding method for 3D triangle meshes in the predictive VQ domain. The proposed method is reversible and enables the cover mesh data to be completely restored when the payload is removed from the VQ bitstream. The mechanism is embedding the payload by modifying the prediction rules during the VQ process. Besides, the hash of the cover mesh can be hidden for the self authentication purpose. Experimental results demonstrate the high capacity of the proposed data hiding scheme.

1 Introduction

Data hiding has become an accepted technology for enforcing multimedia protection schemes. While major efforts concentrate on still images, audio and video clips, recently the research interests in 3D mesh data authentication to ensure data authenticity and integrity have been increasing.

One possible drawback of most previously proposed data hiding schemes is the fact that this distortion cannot be removed even when the host signal is deemed authentic. Although the distortion is often small, when multiple data hiding operations are performed on the host signal, the distortion may be accumulated and unacceptable in certain military or medical applications with a high strategic importance. So it is important to analyze the conditions under which it is possible to remove the changes introduced by data hiding if the host signal is verified as authentic so that anyone who possesses the authentication key can revert to the exact copy of the original host signal.

Reversible data hiding [1-11], which is also called invertible, lossless,distortion-free or erasable data hiding, has only recently been focused on. It embeds the

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 233–243, 2008.

payload (data to be hidden) into a digital content in a reversible manner. A reversible data hiding algorithm guarantees that when the payload is removed from the stego content (the content hidden with payload), the cover content (the original content before embedding) can be exactly restored.

The first publication on invertible authentication that we are aware of is the patent of Honsinger et al. [1], owned by the Eastman Kodak Company. In 2003, Jana Dittmann and Oliver Benedens [10] first explicitly presented a reversible authentication scheme for 3D meshes. In their elegant scheme, a concept for distortion-free invertibility and a concept for adjustable minimum distortion invertibility are introduced. In 2005, Wu and Cheung [11] proposed a reversible data hiding method to authenticate 3D meshes by modulating the distances from the mesh faces to the mesh center to embed a fragile watermark, achieving high capacity with very little distortion between the cover mesh and the restored one. It keeps the modulation information in the stego mesh so that the reversibility of the embedding process is achieved. In 2006, Li et al. proposed a reversible data hiding approach for 3D mesh in the predictive VQ domain [12]. The watermarking scheme is very robust to zero-mean Gaussian noise in a noise channel. However, the main drawback of the algorithm is that the capacity for data hiding is not high.

It is noticeable that when combining the graphics technology with the Internet, the transmission delay for 3D meshes becomes a major performance bottleneck, especially for meshes consisting of tens of thousands of triangles. Under the limited network bandwidth, as well as the storage problem within host systems, reducing the amount of data is, go without saying, an effective solution. Consequently, many 3D mesh compression techniques based on vector quantization have surged in recent years and thus more and more 3D meshes have been represented in the form of VQ bitstream. So it is urgent to authenticate the VQ bitstream of a 3D mesh that is equivalent to its counterpart in the original format, as proposed in this paper.

The rest parts of the paper are organized as follows. Section 2 describes the proposed reversible data hiding scheme, followed by the simulation results in Section 3. Finally, Section 4 concludes the paper.

2 Proposed Data Hiding Approach

The proposed reversible data hiding diagram is illustrated in Fig. 1. If we compress the original mesh M_0 with VQ technique, the mesh will be converted to the cover mesh M in its bitstream form. The payload is hidden in M during the PVQ encoding process by modifying its residual vectors with different prediction mechanisms, and then we obtain the stego mesh *M'* in its bitstream form. Before *M'* is sent to the decoder, it might or might not have been tampered by some intentional or unintentional attack. If the decoder finds that no tampering happened in *M'*, i.e. *M'* is authentic, then the decoder can remove the hidden payload from *M'* to restore the cover mesh, which results in a new mesh *M"*.

According to the concept of reversible data hiding, the restored mesh *M"* should be exactly the same as the cover mesh *M*, vertex by vertex and bit by bit.

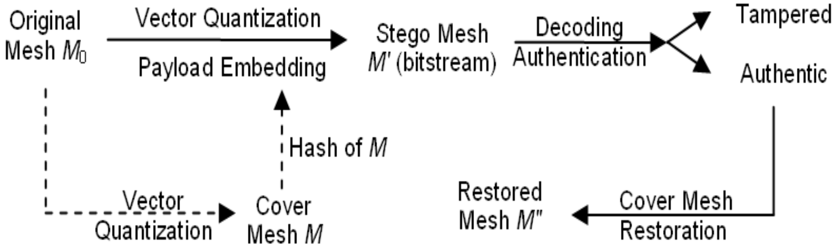


Fig. 1. Reversible data hiding framework

Although the VQ compression technique introduces a small amount of distortion to the mesh, as long as the distortion is small enough, the distortion can be acceptable. Besides, VQ technique enables the distortion to be as tiny as possible by simply choosing a higher quality level of codebook. In fact, when the codebook is generated by the cover mesh itself and the codebook size is the same as the number of VQ quantized vertices of the mesh, the distortion will be 0, i.e., the proposed reversible authentication scheme can be applied to un-quantized meshes. In addition, some existing reversible data hiding techniques, e.g. [9] and [11], also introduce a small amount of distortion between the restored data and the cover data. In [9], a very small distortion is introduced in the restored image because the DCT coefficients used for embedding have a quantization factor. In this sense, in our proposed scheme, M_0, as well as *M*, can both be reversibly authenticated as long as the distortion is small enough.

2.1 Predictive VQ Scheme

Conceptually, VQ is a generalization of nonuniform scalar quantization to operate on vectors rather than scalars [13]. Vector quantization can be defined as a mapping procedure from the k-dimensional Euclidean space to a finite subset, i.e. $Q : R^k \rightarrow C$, where the subset $C = \{\mathbf{c}_i | i = 1, 2, ..., N\}$ is called a codebook, where $\mathbf{c}_i$ is a codevector and N is the codebook size. The mapping procedure is as follows. Select an appropriate codevector $\mathbf{c}_p = (c_{p0}, c_{p1}, ..., c_{p(k-1)})$ as the decoded vector for the input vector $\mathbf{x} = (x_0, x_1, ..., x_{(k-1)})$, which guarantees that the codevector $\mathbf{c}_p$ is the closest vector to $\mathbf{x}$ among all the codevectors in C. The distance metric is usually the squared Euclidean measure.

This process is lossy since different input vectors may map to the same approximating codevector. The quantized input vector can then be specified by p, i.e. the index of the codevector $\mathbf{c}_p$, which is an integer that can be encoded as a fixed-length binary index and thus all indices are transmitted over the network in place of the original geometry data. With VQ, the decompression process is highly efficient, requiring only table lookup operations.

The vertex $\mathbf{v}_n$ in a 3D triangle mesh can be predicted by its neighboring quantized vertices

$$\tilde{\mathbf{v}}_{n(1)} = \hat{\mathbf{v}}_{n-1} + \hat{\mathbf{v}}_{n-2} - \hat{\mathbf{v}}_{n-3} \tag{1}$$

The prediction sketch is depicted in Fig. 2, where $\hat{\bullet}$ denotes the quantized vertex and $\tilde{\bullet}$ denotes the predicted vertex. The detailed prediction design is illustrated in [14].

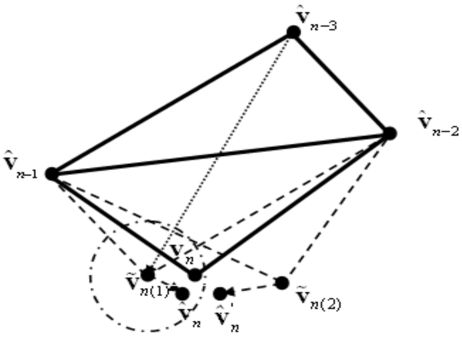


Fig. 2. The sketch of mesh vertex prediction (the radius of the circle is αD)

A common prediction mechanism is the parallelogram prediction as follows:

$$\tilde{\mathbf{v}}_{n(1)} = \hat{\mathbf{v}}_{n-1} + \hat{\mathbf{v}}_{n-2} - \hat{\mathbf{v}}_{n-3}$$

which corresponds to the $\tilde{\mathbf{v}}_{n(1)}$ in Fig. 2.

If we consider $\mathbf{v}_i$ in M, a vertex whose 3D coordinates are given by the vector $\mathbf{v}_i$, from the 3D mesh M, we define its r-ring neighbourhood as the vertices connected to it with a r-edge distance:

$$N_r(\mathbf{v}_i) = \{\forall \mathbf{v}_j, j = 1, \cdots, L \,||\mathbf{v}_j\mathbf{v}_i| = r, r = 0, 1, \cdots\} \tag{2}$$

where $|\mathbf{v}_j\mathbf{v}_i|$ is the cardinality which counts the minimum set of points on the line segment joining the two vertices, but excluding the end-points, while L denotes the total number of vertices of the neighbourhoods N_r $(\mathbf{v}_i)$. The vertex $\mathbf{v}_i$ is considered as being its own 0-ring neighbourhood.

Then we can represent $\mathbf{v}_n$'s neighbourhood in the i-th ring as follows:

$$\mathbf{v}_{n-i}^{(m)} = \tilde{\mathbf{v}}_{n-i}^{(m)} + \mathbf{e}_{n-i}^{(m)}, i = 0, 1, ..., k \tag{3}$$

where $\mathbf{v}_{n-i}^{(m)}$ is a i-ring neighbourhood to $\mathbf{v}_n$, $m = 0, 1, ..., L_i$ and L_i is the maximum number of vertices in the i-ring neighbourhood to $\mathbf{v}_n$, while $\mathbf{e}_{n-i}^{(m)}$ is the corresponding residual vector for $\mathbf{v}_{n-i}^{(m)}$.

With the presumption that vertices with 1-edge distance have roughly the same prediction residual vector, we have another prediction mechanism for $\mathbf{v}_n$ as follows:

$$\tilde{\mathbf{v}}_{n(2)} = \tilde{\mathbf{v}}_{n(1)} + \varepsilon_{n-1} \tag{4}$$

where $\tilde{\mathbf{v}}_{n(2)}$ is the newly prediction of $\mathbf{v}_n$ and is shown in Fig. 2, while ε_{n-1} is the average of the residual vector $\mathbf{e}_{n-i}^{(m)}$ for each predicted vertex.

During the encoding process without data hiding, we employ the mechanism (1). The residual $\mathbf{e}_n = \mathbf{v}_n$ - $\tilde{\mathbf{v}}_{n(1)}$ is quantized, resulting in $\hat{\mathbf{e}}_n$ and its corresponding codevector index i_n. Consequently, vertex $\mathbf{v}_n$ is approximated by the quantized vertex $\hat{\mathbf{v}}_n$ as follows:

$$\hat{\mathbf{v}}_n = \tilde{\mathbf{v}}_{n(1)} + \hat{\mathbf{e}}_n \tag{5}$$

The three vertices in the initial triangle are not VQ quantized. With the quantization procedure going on, the quantized vertices increase, until all the vertices except that the initial three ones are quantized.

In our scheme, 20 meshes were randomly selected from the famous Princeton 3D mesh library [15] and 42507 training vectors were generated from these meshes for training the approximate universal codebook off-line. The residual vectors are then used to generate the codebook based on the minimax partial distortion competitive learning (MMPDCL) method [16] for optimal codebook design. In this way, we expect the codebook to be suitable for nearly all triangle meshes for VQ compression and can be pre-stored in each terminal in the network. Thus the compressed bitstream can be transmitted alone with convenience.

2.2 Data Hiding Process

The payload is hidden by modifying the prediction mechanism. In order to ensure reversibility, we should select specific vertices as candidates.

Let

$$D = \left\| \tilde{\mathbf{v}}_{n(2)} - \hat{\mathbf{v}}_n \right\|_2 \tag{6}$$

Then we select an appropriate parameter $\alpha (0 < \alpha < 1)$, which is used for payload capacity control. $\hat{\mathbf{v}}_n$ can be hidden with a bit of payload when it satisfies the following condition:

$$\left\| \tilde{\mathbf{v}}_{n(1)} - \hat{\mathbf{v}}_n \right\|_2 < \alpha \cdot D \tag{7}$$

Under the above condition, if the payload bit is *"0"* we maintain the codevector index unchanged. Otherwise, if the payload bit is *"1"*, we should make a further judgment as follows. Firstly, $\tilde{\mathbf{v}}_{n(2)}$ is adopted as the new prediction of $\mathbf{v}_n$. Thus we quantize the residual vector $\mathbf{e}'_n$ as follows

$$\hat{\mathbf{e}}'_n = Q[\mathbf{e}'_n] = Q[\hat{\mathbf{v}}_n - \tilde{\mathbf{v}}_{n(2)}] \tag{8}$$

where $Q[\bullet]$ is the VQ operation.

The quantized residual vector $\hat{\mathbf{e}}'_n$ and its corresponding codevector index i'_n are acquired by matching the codebook. Thus, the new quantized vector is:

$$\hat{\mathbf{v}}'_n = \tilde{\mathbf{v}}_{n(2)} + \hat{\mathbf{e}}'_n \tag{9}$$

Then we compute a temporary vector $\hat{\mathbf{v}}''_n$ as follows:

$$\hat{\mathbf{v}}''_n = Q[\hat{\mathbf{v}}'_n - \tilde{\mathbf{v}}_{n(1)}] + \tilde{\mathbf{v}}_{n(1)} \tag{10}$$

If the following condition is satisfied:

$$\hat{\mathbf{v}}''_n = \hat{\mathbf{v}}_n \tag{11}$$

in other words, the reconstructed vector after the change of prediction mechanisms can be exactly restored to the original reconstructed vector before embedding, $\hat{\mathbf{v}}_n$ can be hidden with the payload bit *"1"*. In this situation, we replace the codevector index of $\hat{\mathbf{e}}_n$ with i'_n, while $\hat{\mathbf{v}}_n$ remains unchanged.

The payload bit *"1"* cannot be hidden even when (7) and (11) are satisfied, in the unlikely case as follows: the nearest vertex to the vector $\hat{\mathbf{v}}_n - \hat{\mathbf{e}}'_n$ out of $\tilde{\mathbf{v}}_{n(1)}$ and $\tilde{\mathbf{v}}_{n(2)}$ is not $\tilde{\mathbf{v}}_{n(2)}$. This instance can be avoided by increasing the size of the codebook to achieve a better quantization precision. When the payload bit *"1"* cannot be hidden, proceed to the next vertex until the bit satisfies the hiding conditions.

One flag bit of the side information is required to indicate if a vertex is hidden with a payload bit. In this work, the bit *"1"* indicates that the vertex is hidden with a payload bit while *"0"* indicates not. The vertex order in the payload embedding process is the same as the VQ quantization process.

2.3 Payload Extraction

When the flag bit is *"1"*, we find the codevector specified by the received index by table lookup operations in the codebook. Then we compute a temporary vector $\mathbf{x}_n$ by subtracting the codevector, $\hat{\mathbf{e}}_n$ or $\hat{\mathbf{e}}'_n$, from $\hat{\mathbf{v}}_n$. It can be easily deduced from the payload hiding process that, if the nearest vector to $\mathbf{x}_n$ out of $\tilde{\mathbf{v}}_{n(1)}$ and $\tilde{\mathbf{v}}_{n(2)}$ is $\tilde{\mathbf{v}}_{n(1)}$, the hidden payload bit is *"0"*; otherwise, the hidden payload bit is *"1"*.

Whenever (11) is not satisfied during the decoding process, we terminate the procedure because the mesh bitstream must have been tampered with and is certainly unauthorized. Thus if a mesh bitstream is tampered with, the decoding process cannot be completed in most cases.

2.4 Cover Mesh Restoration

When the hidden payload bit is judged to be *"1"*, $\hat{\mathbf{v}}'_n$ is computed by adding $\tilde{\mathbf{v}}_{n(2)}$ and $\hat{\mathbf{e}}'_n$. Then we can easily acquire $\hat{\mathbf{v}}_n$ according to (10) and (11). When the hidden payload bit is judged to be *"0"*, no operation is needed.

After all vertices have been restored to their original values, the restored mesh *M"* in its un-compressed form is acquired. For content authentication, we compare the authentication hash hidden in the bitstream with the hash of *M"*. If they match exactly, then the mesh content is authentic, and the restored mesh is exactly the same as the cover mesh *M*. Most likely a tampered mesh will not go through to this step because some decoding error could happen as mentioned in the payload extraction process. Note that we use a slightly different order of operations from Fig. 1. We reconstruct a restored mesh first, and then authenticate the content of the stego mesh.

2.5 Performance Analysis

The capacity bottleneck is to satisfy equation (11), which is the same as that in [12]. In [12], two other uncommon prediction rules are used other than the parallelogram prediction. When the payload bit *"1"* is embedded, one of the two uncommon prediction rules is used, resulting in a large residual vector, so the vector quantization error is large. As a result, the equation (11) is not likely to satisfy in [12]. In this work, both $\hat{\mathbf{e}}_n$ and $\hat{\mathbf{e}}'_n$ are small, so small vector quantization error ought to be expected, and thus equation (11) is more likely to satisfy. As a result, high capacity of payload hiding can be achieved.

Attacks on mesh topology such as mesh simplification, re-sampling or insection are not available because the geometric coordinates and topology of the mesh are unknown before the VQ bitstream is decoded.

Residual vectors are kept small after the payload hiding process, so the statistic characteristic of the bitstream does not change much, so one cannot judge whether a codevector index corresponds to a payload bit by simply observing it. Instead, the payload can only be extracted by the payload extraction algorithm. The flag bits in the bitstream can be shuffled with a secure key. In this sense, the payload is imperceptible. Another benefit is that, if the receiver decodes the bitstream without knowing some payload has been embedded, the decoding quality is also acceptable.

Any small change to the authenticated mesh will be detected with a high probability because the chances of obtaining a match between the calculated mesh hash and the extracted hash are equal to finding a collision for the hash.

In addition, In order to reduce the encoding time of VQ, we adopt the mean-distance-ordered partial codebook search (MPS) [17] as an efficient fast codevector search algorithm which uses the mean of the input vector to dramatically reduce the computational burden of the full search algorithm without sacrificing performance.

3 Experimental Results

To evaluate the effectiveness of the proposed methods, we first adopt 8 meshes as the experimental objects. First we quantize the original mesh M_0 to acquire the cover mesh *M* with a universal codebook consisting of 8192 codevectors. The

PSNR values between M_0 and M are 50.99 dB and 56.40 dB, for Stanford Bunny and Dragon meshes, respectively. The PSNR values can be further improved by many other sophisticated VQ encoding techniques that are not what we aim at in this work.

M_0, M, M” and the decoded meshes without removing the payload for Bunny and Dragon are shown in Fig. 3. Comparing these meshes visually, we can know that there are no significant differences among the Bunny meshes and the Dragon meshes. Other original meshes used here are depicted in Fig. 4.

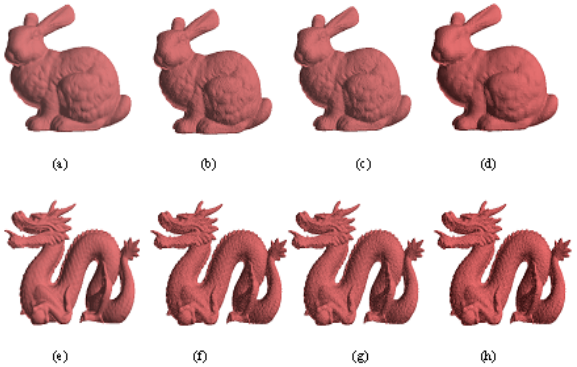


Fig. 3. Comparisons of rendered meshes (implemented with OpenGL). (a) Original Bunny mesh. (b) Cover Bunny mesh. (c) Restored Bunny mesh. (d) Restored Bunny mesh without removing payload (e) Original Dragon mesh. (f) Cover Dragon mesh. (g) Restored Dragon mesh. (h) Restored Dragon mesh without removing payload.

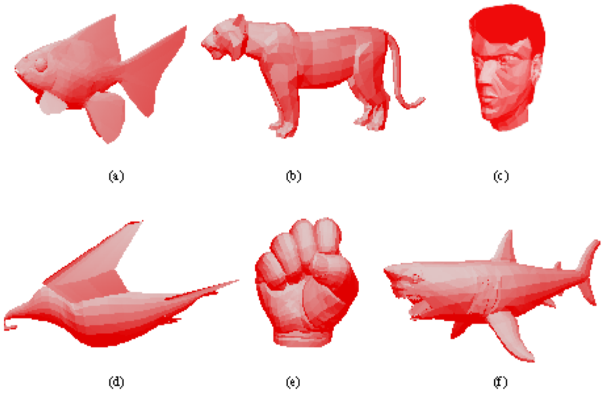


Fig. 4. Other original meshes (implemented with OpenGL). (a) Goldfish (b) Tiger (c) Head (d) Dove (e) Fist (f) Shark.

Table 1. PSNR values of the vector quantized meshes (PSNR1) and PSNR values of the decoded meshes without removing the payload (PSNR2)

Mesh	Bunny	Dragon	Goldfish	Tiger	Head	Dove	Fist	Shark
PSNR1	50.99	56.40	41.15	44.19	42.31	39.33	38.82	47.90
PSNR2	42.73	47.50	35.35	35.75	35.21	37.11	33.64	36.76
#V	8171	100250	1004	956	1543	649	1198	1583
#F	16301	202520	1930	1908	2688	1156	2392	3164

Table 2. Capacity values for various meshes with different α

α Mesh	0.5	0.6	0.7	0.8	0.9	1.0
Bunny	0.06	0.12	0.21	0.30	0.38	0.47
Dragon	0.04	0.06	0.09	0.12	0.15	0.21
Goldfish	0.11	0.17	0.27	0.36	0.42	0.50
Tiger	0.10	0.15	0.24	0.33	0.40	0.48
Head	0.12	0.18	0.25	0.32	0.40	0.51
Dove	0.12	0.15	0.19	0.27	0.33	0.42
Fist	0.12	0.19	0.25	0.32	0.42	0.50
Shark	0.15	0.22	0.30	0.39	0.45	0.51

Table 1 lists PSNR values of the vector quantized (restored) meshes and the PSNR values of the decoded meshes using the parallelogram prediction mechanism without removing the payload, as well as their vertices and faces numbers. As shown in Table 2, with α increasing, the embedding capacities for various meshes increase while the correlation values between the extracted payloads and the original ones remain to be 1.0. Each capacity in all tables is represented by the ratio of hidden payload bits to the number of mesh vertices. Evident in Table 2, the capacity for each mesh is as high as about 0.5, except for Dragon model. This is because Dragon model has very high definition and the prediction error vectors are of small amplitude compared to the codevectors in the universal codebook. Payload in this case can be increased by using a larger codebook that contains enough small codevectors. The payload of the proposed data hiding method is about 2 to 3 times of the capacity reported by [12].

PSNR values of meshes are computed between M" and M_0 as follows:

$$PSNR = 10 \cdot \log \frac{B}{\sum_{i=1}^{B} \left\| \mathbf{v'}_i - \mathbf{v}_i \right\|_2^2} \tag{12}$$

where B is the number of vertices of M_0, $\mathbf{v'}_i$ and $\mathbf{v}_i$ are the i-th vertex of M" andM_0, respectively, and all the vertices in M_0 are previously normalized in a zero mean sphere with a radius of 1.0. A higher PSNR value corresponds to better quality.

4 Conclusions

In this paper, we proposed a new invertible authentication scheme for 3D meshes based on a data hiding technique. The payload hidden has cryptographic strength and is global in the sense that they can detect every modification made to the mesh with probability equivalent to finding a collision for a cryptographically secure hash function. This technique embeds the hash or some invariant features of the whole mesh as a payload. The proposed method can be localized to blocks rather than applied to the whole mesh.

In addition, it is argued in the paper that all typical meshes can be authenticated and this technique can be further generalized to other data types, Dd e.g. 2D vector maps, arbitrary polygonal 3D meshes and 3D animations. Our future work will focus on the further improvement of the capacity and security of the proposed scheme.

Acknowledgement

This work was supported by the Program for one-hundred persons with ability in Sun Yat-Sen University, the Program for New Century Excellent Talents in University of China under grant NCET-04-0329 and the Foundation for the Author of National Excellent Doctoral Dissertation of China (No. 2003027).

References

1. Honsinger, C.W., Jones, P., Rabbani, M., Stoffel, J.C.: Lossless recovery of an original mesh containing embedded data. US Patent application, Docket No: 77102/E–D (1999)
2. Jun, T.: High capacity reversible data embedding and content authentication. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 3, pp. 517–520 (2003)
3. Xuan, G.R., Shi, Y.Q., Ni, Z.C., Chen, J., Yang, C., Zhen, Y., Zhu, J.: High capacity lossless data hiding based on integer wavelet transform. In: Proceedings - IEEE International Symposium on Circuits and Systems, vol. 2 (2004)
4. Shi, Y.Q., Ni, Z., Zou, D., Liang, C., Xuan, G.: Lossless data hiding: Fundamentals, algorithms and applications. In: Proceedings - IEEE International Symposium on Circuits and Systems, vol. 2 (2004)
5. Zhicheng, N., Yun-Qing, S., Nirwan, A., Wei, S.: Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology 16(3), 354–361 (2006)
6. Celik Mehmet, U., Gaurav, S., Murat, T.A., Eli, S.: Reversible data hiding. In: IEEE International Conference on Image Processing, vol. 2, pp. II/157–II/160 (2002)
7. Guorong, X., Chengyun, Y., Yizhan, Z., Yunqing, S., Zhicheng, N.: Reversible data hiding based on wavelet spread spectrum. In: IEEE 6th Workshop on Multimedia Signal Processing, pp. 211–214 (2004)
8. Zhicheng, N., Yunqing., S., Nirwan, A., Wei, S., Qibin, S., Xiao, L.: Robust lossless image data hiding. In: IEEE International Conference on Multimedia and Expo, vol. 3, pp. 2199–2202 (2004)

9. Fridrich, J., Goljan, M., Du, R.: Invertible Authentication Watermark for JPEG Images. In: Proc. IEEE International Conference on Information Technology: Coding and Computing (2001)
10. Dittmann, J., Benedens, O.: Invertible Authentication for 3d-Meshes. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 5020, pp. 653–664 (2003)
11. Wu, H.-T., Ming-Cheung, Y.: A Reversible Data Hiding Approach to Mesh Authentication. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (2005)
12. Sun, Z., Lu, Z.-M., Li, Z.: Reversible Data Hiding for 3D Meshes in the PVQ-Compressed Domain. In: IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 593–596 (2006)
13. Gray, R., Neuhoff, D.: Quantization. IEEE Transactions on Information Theory 44, 2325–2384 (1998)
14. Chou, P.H., Meng, T.H.: Vertex Data Compression through Vector Quantization. IEEE Transactions on Visualization and Computer Graphics 8(4) (October-December 2002)
15. Princeton University. 3D model search engine, `http://shape.cs.princeton.edu`
16. Zhu, C., Po, L.M.: Minimax Partial Distortion Competitive Learning for Optimal Codebook Design. IEEE Transactions on Image Processing 7(10), 1400–1409 (1998)
17. Ra, S.W., Kim, J.K.: Fast Mean-Distance-Ordered Partial Codebook Search Algorithm for Image Vector Quantization. IEEE Transactions on Circuits and Systems-II 40(9), 576–579 (1993)

Reversible Data Hiding Using Prediction Error Values Embedding

Yongjian Hu[1,2], Heung-Kyu Lee[1], Jianwei Li[2], and Kaiying Chen[2]

[1] Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
Daejeon 305-701, Republic of Korea
hklee@mmc.kaist.ac.kr
[2] College of Automation Science and Engineering
South China University of Technology, Guangzhou 510641, China
eeyjhu@scut.edu.cn

Abstract. In this paper, we propose a new embedding method for reversible data hiding. Unlike traditional methods using difference-expansion or feature modifications, we propose an even- and odd-number-based embedding method. This method is implemented in a prediction error image. The experimental results show that our embedding method can achieve satisfactory embedding capacity and image quality.

Keywords: Data hiding, digital watermarking, reversible watermarking, lossless watermark, prediction error.

1 Introduction

Reversible data hiding techniques in literature can be grossly classified into two types: one uses data compression (e.g. [1]-[5]), and the other does not (e.g. [6]-[8]). However, even relying on data compression, the typical methods in [1] and [2] take different strategies. Fridrich *et al.* [1] select and compress visually unimportant image details to acquire spare space. The data stream to be embedded contains three parts: the embedding location map, the compressed bit-stream of the original image details in the embedding area, and the payload. The preservation of both the location map and the original image details is a character of [1]. The embedding capacity usually depends on both selected image details and compression efficiency. On the other hand, Tian [2] proposes another way to use compression. He defines expandable pixels and compresses the location map that records the locations of selected expandable pixels. Since the size of compressed location map is usually much smaller than that of original location map, the preservation of the compressed location map only consumes a portion of the embeddable locations provided by the selected expandable pixels. Therefore, the rest expandable pixels can provide locations for saving the payload. The methods in [1] and [2] are two reversible data hiding prototypes using compression. They have already had many variants and extensions (e.g., [4] for [1] and [3] for [2]). In [5], Kamstra *et al.* even propose two extensions, one for [1], and the

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 244–253, 2008.

other for [2]. The principle of data hiding using compression is also adopted for reversible visible watermarking in [10].

Typical methods without using compression are [6] and [7], which need neither the location map nor the storage of original image details. Their main task is to seek special image features and change them to embed information bits.

In this paper, we propose a different embedding method. Taking into account the structure of even and odd numbers, we propose to embed an information bit in the least significant bit (LSB) of an even or odd number. We implement our embedding method in a prediction error image. The experiments show that the proposed method can achieve reasonable embedding capacity as well as satisfactory image quality.

The paper is organized as follows. Section 2 introduces an even-number-based embedding scheme. Section 3 introduces an odd-number-based embedding scheme. Section 4 implements our embedding method in a prediction-error-based reversible data hiding algorithm. Section 5 gives experimental results and discussion. We draw the conclusion in Section 6.

2 Embedding Using Even Numbers

Difference-expansion embedding in [2] is a popular embedding technique. It has many advantages and has been extended into many variants (e.g. [3][5][9]). However, one apparent disadvantage is that the binary representation of the difference has to be left-shifted one bit to obtain the vacant LSB position for embedding; otherwise, the decoder can not identify the location of the hidden bit. Since the LSB of an even number is always vacant, we intuitively think that using this spare position for embedding will benefit. On the one hand, the embedding distortion is small; on the other hand, the embedding location is fixed. However, there are several questions to be answered before using this even number embedding scheme. The first one is how we record the locations of the selected differences. The second one is how we ensure that there are enough differences to satisfy the requirement of capacity. We answer these two questions while we describe our even-number-based embedding scheme.

Generally, even and odd difference values are uniformly spread across a difference image. The amount of even differences and that of odd differences are often close. If we directly select even difference values for embedding, we can seldom obtain spare space by compression as long as we need to save the location map of these difference values. The reason is that the compression of the location map is inefficient. Therefore, we have to find a pixel selection scheme and rearrange embeddable differences to make the compression efficient. We use a prediction error image as the embedding domain to explain our pixel selection and embedding scheme. The prediction error, p_e, is computed by $p_e = x - \hat{x}$, where x and $\hat{x}$ denote a pixel and its predicted value, respectively.

There are three types of prediction error values: 0, even and odd integers. Typically, the distribution of prediction errors is close to a Laplacian distribution. In order to select suitable pixels, we choose an even threshold $T(> 0)$.

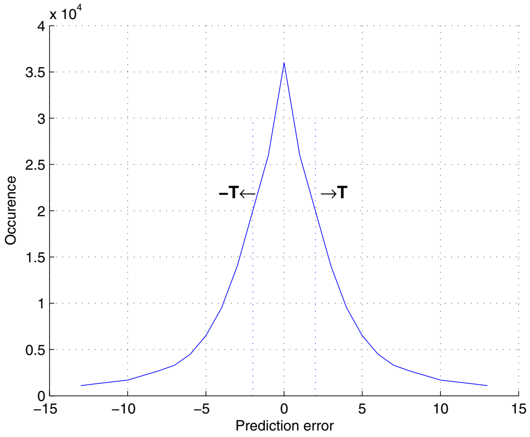


Fig. 1. Selection of embeddable difference values

Initially, we set T to 2. Then, we shift outwards all of even prediction errors by T along the axis *prediction error*, as shown in Figure 1. In other words, we add T to the absolute of all even prediction errors except 0. The set of selected p_e includes 0, even prediction errors, and odd prediction errors whose value is less than T. We embed a binary bit i into each p_e. The embedded location is recorded in a location map. We describe our even-number-based embedding scheme in a pseudo C code below.

```
if ($p_e == 0$),
  {$p_e = i;$}
else if ($(|p_e|\%2 == 0)$ && $(p_e \neq 0)$),
  {
   $|\tilde{p_e}| = |p_e| + T;$
   $|p_e'| = |\tilde{p_e}| + i = |p_e| + T + i;$
  }
else if ($(|p_e|\%2 \neq 0)$ && $(|p_e| < T)$),
  {
  $|\tilde{p_e}| = |p_e| + 1;$
  $|p_e'| = |\tilde{p_e}| + i = |p_e| + 1 + i;$
  }
```

where $\tilde{p_e}$ and p_e' denote an even prediction error and an embedded prediction error, respectively. $|\cdot|$ is the absolute operator. % refers to *mod* operation. Apparently, we select all of even prediction error values in the prediction error image. In addition, we select some odd prediction error values. The selection of odd $p_e(< T)$ is for increasing the embedding capacity. Implicitly, this selection makes compression more efficient. Note that we turn those odd prediction error values into even numbers by adding 1 instead of subtracting 1. The reason is to avoid confusion with the embedding of $p_e = 0$. This will be clear in the decoding process. If the current T can not get enough capacity for the payload, we enlarge T by 2. The enlarged T can get more embeddable locations. The enlargement process continues until there are enough embeddable locations for the payload.

At the decoder, the data extraction is straight and simple. With the threshold T and the location map, we can get back the hidden bits from the altered prediction errors. The decoding rule is described as follows.

```
if the length of $|p'_e|$ is 1,
  {
   $i = p'_e$;
   $p_e = 0$;
  }
else if $(|p'_e| > T + 2)$,
  {
   $i = |p'_e| \% 2$;
   $|p_e| = |p'_e| - T - i$;
  }
else if $(|p'_e| < T + 2)$,
  {
   $i = |p'_e| \% 2$;
   $|p_e| = |p'_e| - 1 - i$;
  }
```

One problem we have to discuss in data hiding is overflow or underflow of pixel values. The above embedding operation only takes place in a prediction error that will not result in an overflow or underflow of the embedded pixel value. Here we give the constraints on an embeddable pixel. For an embedded prediction error, p'_e, the embedded pixel value, x', is computed as

$$x' = \hat{x} + sign(p_e)|p'_e| = x - p_e + sign(p_e)|p'_e| \tag{1}$$

where $sign(\cdot)$ is a function that denotes the sign of a variable. To prevent overflow and underflow, x' should be restricted in the range of [0,255]. So we have

```
if $(p_e == 0)$,
  {
  $x' = x + i$;
  the constraint on embeddable pixels : $0 \leq x \leq 254$
  }
if $(|p_e| \% 2 == 0)$ and $(p_e \neq 0)$,
  {
  $x' = x - p_e + sign(p_e)(|p_e| + T + i)$;
  if $(p_e > 0)$,
  {the constraint on embeddable pixels : $x + T \leq 254$ }
  if $(p_e < 0)$,
```

Table 1. Embeddable and non-embeddable prediction errors of an image

	embeddable	non-embeddable
$p_e = 0$	$0 \in Z_m$	$0 \in Z_n$
Even p_e	2,4,6,...$\in E_m$	2,4,6,...$\in E_n$
Odd p_e	1,3,...$< T$ and $\in O_m$	1,3,...$< T$ and $\in O_n$, or $p_e > T$

```
    {the constraint on embeddable pixels : $1 \leq x - T$ }
    }
  if ($|p_e|\%2 \neq 0$) and ($|p_e| < T$),
    {
    $x' = x - p_e + sign(p_e)(|p_e| + 1 + i)$;
    if ($p_e > 0$),
    {the constraint on embeddable pixels : $x \leq 253$}
    if ($p_e < 0$),
    {the constraint on embeddable pixels : $2 \leq x$ }
    }
```

Based on the above constraints, we can determine whether p_e is embeddable or non-embeddable before embedding. For simplicity, we define the following notations. Let Z_m and Z_n denote the sets of embeddable and non-embeddable $p_e(= 0)$, respectively. Let E_m and E_n denote the sets of embeddable and non-embeddable even p_e, respectively. Let O_m and O_n denote the sets of embeddable and non-embeddable odd p_e, respectively. We list all of the embeddable and non-embeddable prediction errors in Table 1. We record embeddable locations with "1" in a 2-D (2-dimensional) binary location map, which we call as the first location map (FLM). It is worth mentioning that even-number-based embedding is only applied to embeddable p_e in FLM.

So far we have described our even-number-based embedding scheme. To further increase embedding capacity, we introduce our odd number embedding scheme below.

3 Embedding Using Odd Numbers

In section 2, we have selected some odd prediction errors for embedding. We turn the selected odd values into even ones prior to embedding. In this section, however, we introduce a way to directly embed a bit into an odd number. Before doing that, we introduce the second 2-D binary location map (SLM). SLM is used to indicate the locations where we perform odd-number-based embedding.

From Table 1, we observe that, when $p_e < T$ and $p_e \in O_n$, or when $p_e > T$, odd p_e is unusable for even-number-based embedding. On the other hand, the experiments show that there are only few non-embeddable even p_e and non-embeddable $p_e(= 0)$ in the prediction error image. These observations motivate us to use the above unusable odd prediction errors for odd-number-based

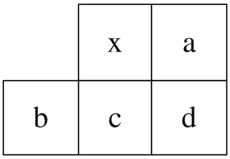


Fig. 2. Predictor

embedding. We first record by using “1” the locations of non-embeddable even p_e and non-embeddable $p_e(=0)$ in SLM. That is, we record the elements of E_n and Z_n in SLM. By jointly using FLM and SLM, we can easily distinguish the above unusable odd prediction errors from those that are embeddable in FLM. Then, we perform odd-number-based embedding on this portion of odd prediction errors. It can be seen that, due to the introduction of this selection scheme, we can practically use all of odd prediction errors in the image. The cost of using these unusable odd p_e is small in that the compression of SLM is very efficient.

The odd-number-based embedding is very simple. We directly replace the LSB of an odd number with an information bit i. One advantage of odd-number-based embedding is that we do not worry about the overflow and underflow. Another advantage is that the recovery of the odd number is straight. After extracting the bit i, we recover the original odd number by simply setting 1 on its LSB position.

So far we have introduced our even- and odd-number-based embedding schemes. To sum up, we describe our complete embedding method using the joint information of FLM and SLM.

- If the location corresponds to “1” in FLM and “0” in SLM, then perform even-number-based embedding.
- If the location corresponds to “0” in FLM and “0” in SLM, then perform odd-number-based embedding.
- If the location corresponds to “1” in FLM and “1” in SLM, impossible.
- If the location corresponds to “0” in FLM and “1” in SLM, no action.

4 Implementation in Prediction Error Image

A prediction error image is the difference between the image and its predicted value. We implement our embedding method in the prediction error image. The predictor we use is borrowed from predictive coding and is shown in Figure 2. The predicted value, $\hat{x}$, is computed as

$$\hat{x} = (2a + b + 2c + d)/6 \tag{2}$$

The embedding process begins from the upper left corner of the image in a raster scan manner. If p_e is embeddable, even- or odd-number-based embedding is performed; otherwise, go to the next pixel and test again. The embedded data consists of two parts: the payload bits and the bit-stream of FLM and SLM. The latter part is attached to the first one. Before embedding, FLM and SLM are compressed by the popular JBIG-kit in [11].

It is easy to understand the way of raster scan embedding; however, there is a problem when we carry out blind data extraction. Without the location map, we do not know whether the current p_e is embeddable or not. So we need the overhead information; otherwise, we have to directly preserve the location map in the embedded image. A possible solution to this problem is that we embed the location map in a fixed region of the embedded image, for example, the first segment of the embedded image; meanwhile, we save the original information of this region in the place originally allotted for the storage of the location map. Thus, we can easily get back the location map at the decoder and resume the original image after removing the hidden data bits. Suppose the length of the bit-stream of the compressed FLM and SLM is L_c. The above idea can be realized in the following steps.

- Step 1: We first embed the payload bits. If the current p_e is embeddable for either even- or odd-number-based embedding, perform the embedding operation. After that, save the LSB of the embedded pixel in the auxillary data D. Then, put a bit of the compressed FLM and SLM bit-stream in the LSB position of the embedded pixel.
- Step 2: If the current p_e is not embeddable for either even- or odd-number-based embedding, directly save the LSB of the image pixel in the auxillary data D. Then, put a bit of the compressed FLM and SLM bit-stream in the LSB position of the pixel.
- Repeat the above embedding and preservation process until the end of the compressed FLM and SLM bit-stream. After that, perform the normal embedding. That is, if the current p_e is embeddable for even- or odd-number-based embedding, perform the embedding operation; otherwise, go to the next pixel and test again.
- After all of the payload bits are embedded, we begin to save the auxillary data D in the last L_c embeddable prediction error values. Actually, these values are originally allotted for saving the compressed FLM and SLM bit-stream.

Before we perform data extraction from the lower right corner of the embedded image, we extract the LSBs of the first L_c pixels in the beginning region of the embedded image. This data package is the compressed FLM and SLM bit-stream. We decompress this bit-stream and get back the FLM and SLM. Then we use them to guide the data extraction process. We put the first extracted L_c bits back into the LSBs of the pixels in the beginning region of the embedded image. After that, perform the normal data extraction process. This original idea for saving the location map in this way was proposed in [9].

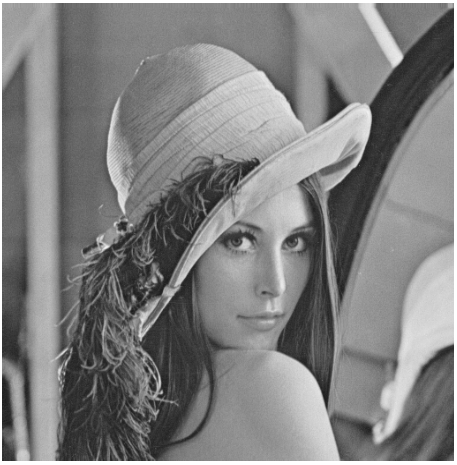


Fig. 3. Reversibly embedded image *Lena* with a 112991 bits (0.43 bpp) payload

5 Experiment and Discussion

To estimate the performance of the proposed algorithm, we test it on different standard images. We give some results in this section.

We first test the embedding capacity of the proposed algorithm under a proper image quality. Figure 3 shows the embedded image *Lena* with a PSNR (peak signal-to-noise ratio) value of about 40dB. It can be seen that both the image quality and the payload bit rate are satisfactory.

We then compare our algorithm with Thodi's prediction-error expansion method in [9]. In Figure 4, we list the best image quality Thodi's algorithm achieves. To make a fair comparison, we modulate our algorithm to yield the embedded image with similar PSNR values. We calculate the payload bit rate

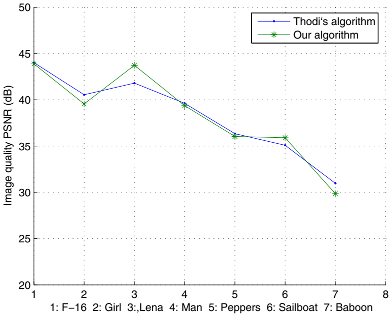


Fig. 4. The maximum PSNR of Thodi's algorithm

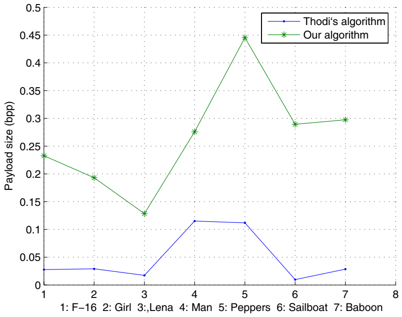


Fig. 5. Comparison of payload bit rate under the maximum PSNR of Thodi's algorithm

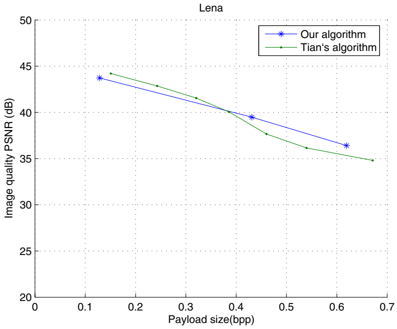


Fig. 6. Performance comparison between our algorithm and Tian's algorithm [2]

under that image quality. Figure 5 demonstrates that our algorithm produces much better payload bit rate than Thodi's algorithm.

We also compare our algorithm with Tian's difference-expansion technique. Both our algorithm and Tian's algorithm possess multi-embedding ability. However, for simplicity, we only compare the results under single embedding. Figure 6 shows that our algorithm is a little better than Tian's method when the bpp is close to 0.5.

During the experiments, we find that we can improve our embedding method in some technical details. For example, we find that the case of $p'_e = -1$ will never happen. The reason is that an embeddable $p_e(= 0)$ may become $p'_e = 0$ or $p'_e = 1$ after embedding, but it will never become $p'_e = -1$. Thus, we add 1 to all $p'_e(< 0)$ when calculating the embedded pixel value. This modification is for decreasing the embedding distortion. In the process of data extraction, as long as $p'_e < 0$, we subtract 1 from p'_e before performing normal data extraction. This improvement can increase the image quality by about 1 dB.

6 Conclusion

The use of even or odd numbers for embedding has obvious advantages. The embedding in the LSB position of an even or odd number makes data extraction and original image recovery very easy. Motivated by difference-expansion embedding and histogram shifting embedding, we have presented an even- and odd-number-based embedding method. Although our research is in the primary stage, the results demonstrate that this embedding method is promising. Theoretically, if a difference image consists of more even values or more odd values, the proposed embedding method would be more efficient. One of our future efforts is to find a way to produce more even values or more odd values in the prediction error image. Practically, Tian's difference-expansion embedding can be thought of as two operations. The first operation is a transformation that turns a difference value into an even number by expansion. The second operation performs even-number-based embedding.

Acknowledgement. This work was supported in part by BK 21 project of KAIST, KOSEF grant NRL program R0A-2007-000-20023-0 of MOST, NSF of China 60772115, 60572140, and NSF of Guangdong 04020004.

References

1. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding - New paradigm in digital watermarking. EURASIP Journal on Applied Signal Processing 2002(2), 185–196 (2002)
2. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. on Circuits and Systems for Video Technology 13(8), 890–896 (2003)
3. Alattar, A.M.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans. on Image Processing. 13(8), 1147–1156 (2004)
4. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless generalized-LSB data embedding. IEEE Trans. on Image Processing. 12(2), 157–160 (2005)
5. Kamstra, L., Heijmans, H.J.A.M.: Reversible data embedding into images using wavelet techniques and sorting. IEEE Trans. on Image Processing. 14(12), 2082–2090 (2005)
6. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Trans. on Circuits and Systems for Video Technology. 16(3), 354–362 (2006)
7. Xuan, G., Yao, Q., Yang, C., Gao, J., Chai, P., Shi, Y.Q., Ni, Z.: Lossless data hiding using histogram shifting method based on integer wavelets. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 323–332. Springer, Heidelberg (2006)
8. De Vleeschouwer, C., Delaigle, J.-F., Macq, B.: Circular interpretation of bijective transformations in lossless watermarking for media asset management. IEEE Trans. on Multimedia 5(1), 97–105 (2003)
9. Thodi, D.M., Rodriguez, J.J.: Prediction-error based reversible watermarking. In: Proc. International Conference on Image Processing., vol. 3, pp. 1549–1552 (2004)
10. Hu, Y., Jeon, B.: Reversible visible watermarking and lossless recovery of original images. IEEE Trans. on Circuits and Systems for Video Technology. 16(11), 1423–1429 (2006)
11. http://www.cl.cam.ac.uk/~mgk25/jbigkit/

An Improved Reversible Difference Expansion Watermarking Algorithm

Sachnev Vasiliy[1], Hyoung-Joong Kim[1], Shijun Xiang[3], and Jeho Nam[2]

[1] CIST, Graduate School of Information Security and Management,
Korea University, Seoul 136-701, Korea
[2] ETRI Daejeon 305-700, Korea
khj-@korea.ac.kr
[3] College of Information Science and Technology, Jinan University,
Guangzhou 510632, P.R. China

Abstract. In this paper, we propose an improved reversible watermarking algorithm by using the simplified location map. The proposed embedding method is based on the Alattar integer transform [3]. Here, we extend the case of using four pixels. Proposed simplified location map we propose just covers those necessary quads, the improved watermarking algorithm has a larger embedding capacity in comparison with the work of Alattar[3]. Simulation testing shows that our embedding strategy has achieved a better performance for all tested images.

1 Introduction

Reversible watermarking algorithms [1–22] hide data (or payload) into host signals (i.e. image pixels or audio samples) under reversible fashion. The embedded message and host image can be recovered exactly in the reversible scheme. The efficiency of embedding data into host signal has two significant requirements: large embedding capacity and smaller image distortion, which are contradictive each other. Embedding more data to the host signal will cause more degradation on the host signal. Thus, idea reversible watermarking algorithms should have a larger embedding capacity under the constraint of the same distortion.

Difference expansion method proposed by Tian [15] has been an innovative scheme. Tian divided image to pairs and computed the difference values between neighboring two pixels for the purpose of hiding data. In order to avoid overflow/ underflow problem and minimize the embedding distortion, he discarded those pairs which can cause significant distortion at the output image or to overstep the limits of image pixel values (0-255). For the purpose of controlling the embedding capacity and the distortion, all pairs are divided into two sets: the expandable set and the not expandable set. The expandable set includes those pairs with difference values less than a predefined threshold. The rest of pairs is not expandable, which will be able to cause overflow or underflow. During data hiding, the difference values from the expandable set are used for expanding and embedding data. However, pairs from the not expandable set are used for embedding data only. Expandable and not expandable pairs are mixed, which will

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 254–263, 2008.

cause that positions of the expandable pairs are missed. Thus, the position information of all expandable and not expandable pairs should be saved in a special location map, which is saved as part of the payload. In location map that covers all pairs has a huge size (i.e half of image resolution). Thus, this method cannot be used for hiding data without compression of the location map. Location map can be compressed well using those efficient lossless compression algorithms such as JBIG2. Compressed location map releases rooms for embedding more useful payload. This is the basic reason that performances of Tian's difference expansion method are related to the efficiency of JBIG2 compression algorithm. If the location map cannot be compressed well, this method may be fail to data hiding.

Alattar improved the different expansion method using another integer transform, which increases the units for data embedding from pair to triplet [1] and quad [2]. Alattar [1] used three pixels to hide two bits in each triplet (in an ideal case). In location map of this method contains information about expendability of each triplet. Thus, the location map size is equal to one third of the host image size. After compression, the location map size is smaller in general, so the capacity is larger than Tian different expansion algorithm. Alattar also proposed another reversible watermarking algorithm for quads [2]. This algorithm can hide 3 bits of information in each quad. The location map size is equal to one fourth of image size. This is the basic reason why Alattar method based on quads has achieved the highest embedding capacity under the same PSNR among existing difference expansion algorithms. This algorithm introduced by Alattar's [2] for color image can hide 3.3 bits/colored pixel with PSNR of 33.5 dB.

Efficiency of the methods [1], [2], [15], depends on the efficiency of location map compression. If the location map can not be compressed well, the payload will be low. Thus, reversible embedding techniques which don't depend on lossless compression algorithm will be more efficient. Even for Alattar's triplet and quads, the location map is still huge in size, which will decrease payload to some extent. In other word, any methods with smaller location map can be better than [1], [2], [15].

Our proposed reversible embedding method has a smaller size of location map and does not depend on the lossless compression algorithm. The proposed simplified location map only contains the information of those intersected cells (for pairs, triplets or quads) after hiding data. The size of the simplified location map is smaller than that in [1], [2], [15]. Thus, our proposed embedding technique based on using simplified location map can hide more data under the same embedding distortion (measured by PSNR). As a conclusion, our proposed method in this work will be able to achieve better performances compare to Tian's difference expansion method and the improved difference expandable methods for triplet and quads by Alattar.

The rest of this paper is organized as follows. In Section 2, the proposed embedding technique based on simplified location map is described. Section 3 will present the exploited features of the proposed method. Experimental results are reported in section 4. Section 5 is conclusion.

2 Simplified Location Map

The proposed reversible data hiding scheme is based on particular features of the embedding process for difference expansion method under threshold T. Assume that the input signal for embedding data is set as $D \in [–5 \cdot T; 5 \cdot T]$, which is differences between neighbor pixels h. According to the difference expansion method, the set D should be divided into two parts: the expandable set E and the not expandable set nE. In expandable set E satisfies the condition $|h| \leq T$, while the not expandable set is under the condition of the expression $nE = D – E$. Thus, before data embedding, the expandable set is denoted as $E \in [-T; T]$, the not expandable set is denoted by $nE \in [–5 \cdot T;–T) \cup (T; 5 \cdot T]$. The set E is expanded according to the rule $H = 2 \cdot h + bit$, where h is the original difference, bit is a bit of watermarked message, and H is the expanded difference value. Expandable set E after expanding belongs to the range $[–2\cdot T; 2\cdot T + 1]$, whereas the set nE stays unchanged. Thus, part of H from the expandable set E is intersected with non expandable set nE in intersected set $I \in [–2\cdot T; -T) \cup(T; 2 \cdot T + 1]$. Sets S_1 and S_4 have no intersection and can be excluded from the location map. The proposed simplified location map covers the differences h from the intersected set I only, but the location map of the original difference expansion methods as Tian's pair [15], Alattar's triplet [1] and quad [2] have to cover all embedded cells as pairs, triplets or quads, respectively. Even though the location maps of the previous difference expansion methods [1] [2] are compressed well, the compressed ones are usually larger in size of the intersected set I. Fig. 1 shows difference between the expanded image and the original image, the intersected set I $(S_2 \cup S_3)$ and the not intersection set nI $(S_1 \cup S_4)$. The set S_1 is a union of the differences $h \in [-T/2; T/2]$. The set S_2 is a union of the differences $h \in [-T;–T/2) \cup (T/2; T]$. The set $S3$ is given as $h \in [–2 \cdot T;–T) \cup (T; 2 \cdot \mathrm{T} +1]$. Note that $S_4 = D \cap S_1 \cap S_2 \cap S_3$. The expandable set is given as $E = S_1 \cup S_2$. The not expandable set is $nE = S_3 \cup S_4$. The sets S_1 and S_2 are respectively expanded to $S_1^E \in [-T; T]$ and $S_2^E \in [–2 \cdot T;–T) \cup (T; 2 \cdot T + 1]$. The not expandable set stays unchanged.

Thus, the sets S_2^E and S_3 intersect in $[–2 \cdot T;–T) \cup (T; 2 \cdot T + 1]$. Decoding process is impossible to implement if there is no information about positions of all expanded differences $H \in [–2\cdot T;–T)\cup(T; 2\cdot T + 1]$. The intersection problem can be easy to solve by using the location map. All expanded difference values from set S_2^E are marked by "1" at the location map. All original difference values from set S_3 are marked by "0". Thus, the location map covers all differences from the sets S_2^E and S_3. Let |D| be the length in D set. Size of the location map is $|L| = |S_2^E|+|S_3|$ or $|L| = |S_2|+|S_3|$, because of $|S_2^E| = |S_2|$. Compression ratio is low for the proposed simplified location map, thus compression algorithm is not used in our proposed methods. Of course, the location map will be compressed for further compression.

Capacity of the proposed method is computed as follows:

$$|P| = |S_1| + |S_2| - |L| = |S_1| + |S_2| - (|S_2| + |S_3|) = |S_1| - |S_3| \qquad (1)$$

Capacity can be easy to compute using the histogram of the input image.

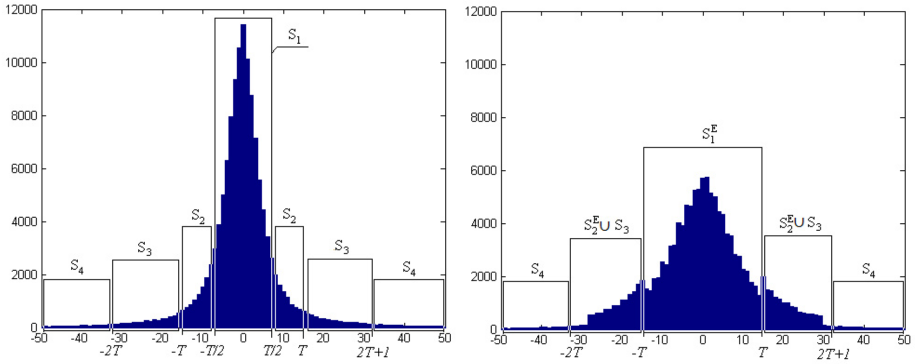


Fig. 1. Prediction pattern

Reversible watermarking technique based on simplified location map has one additional requirement. The location map should be recovered first in the decoder before extracting data from the intersected set I. Set S_1 is more suitable for hiding the simplified location map. Set S_1 has no intersection after data embedding, so the embedded data can be successfully recovered without any additional information.

The simplified location map size depends on the histogram shape of the input signal (for example, the differences h) and the used thresholds T. Proposed embedding technique fails if the capacity of the set S_1 is not enough for hiding the location map $|S_1| < |L|$, when the threshold is not selected reasonably.

The proposed data technique based on simplified location map can be applied for many existing reversible watermarking algorithms. At the next section we will describe improved different expansion method by embedding data to quads of pixels. The proposed reversible watermarking method is the combination of Alattar method based on quads and the simplified location map introduced in this section.

3 Improved Difference Expansion Method

Main point of the algorithm proposed by Alattar is to use quads of pixels for hiding data instead of pairs and triplets (see Figure 2). For each quad, Alattar used a new integer transformation, which allows to hide data in reversible fashion.

Integer transform for the quad of pixels u_0, u_1, u_2, u_3.

Forward transform:

$$v_0 = \left\lfloor \frac{a_0 \cdot u_0 + a_1 \cdot u_1 + a_2 \cdot u_2 + a_3 \cdot u_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor \qquad (2)$$

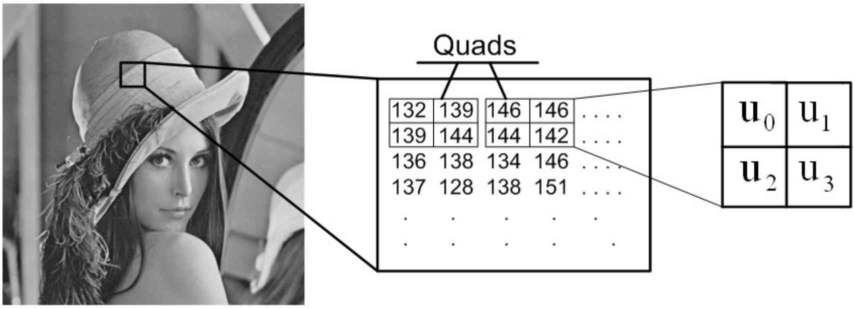


Fig. 2. Prediction pattern

$$v_1 = u_1 - u_0 \tag{3}$$

$$v_2 = u_2 - u_1 \tag{4}$$

$$v_3 = u_3 - u_2 \tag{5}$$

Embedding data:

$$V_1 = 2 \cdot v_1 + b_1 \quad V_2 = 2 \cdot v_2 + b_2 \quad V_3 = 2 \cdot v_3 + b_3 \tag{6}$$

where b_1, b_2 and b_3 are 3 bits of watermark massage.
Inverse transform:

$$U_0 = v_0 - \left\lfloor \frac{(a_0 + a_1 + a_2) \cdot V_1 + (a_2 + a_3) \cdot V_2 + a_3 \cdot V_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor \tag{7}$$

$$U_1 = V_1 - U_0 \tag{8}$$

$$U_2 = V_2 - U_1 \tag{9}$$

$$U_3 = V_3 - U_2 \tag{10}$$

Extension of embedding cell from 2 pixels (Tian's method) to 3 pixels (Alattar's triplet method) and to 4 pixels (Alattar quad method) will significantly decrease the location map size. Among the existing methods, Alattar's quad has better performance. In order to decrease the location map size of quad based embedding method, we propose the simplified location map.

According to the proposed data hiding technique based on the simplified location map, the set of all quads Q should be divided into four subsets S_1, S_2, S_3, S_4, under the threshold T. Each quad has 3 differences v_1, v_2, v_3 for embedding data. Where v_{max} is the difference with the maximum modulo of magnitude among v_1, v_2, v_3.

Thus, the set S_1 is a union of quads with $v_{max} \in [-T/2;\ T/2]$.
The set S_2 is a union of quads with $v_{max} \in [-T;-T/2) \cup (T/2;\ T]$.
The set S_3 is union of quads with $v_{max} \in [-2 \cdot T;-T) \cup (T;\ 2 \cdot T + 1]$.
$S_4 = D \cap S_1 \cap S_2 \cap S_3$.

The expandable set is $E = S_1 \cup S_2$. The not expandable set is $nE = S_3 \cup S_4$. The sets S_1 and S_2 are expanded to S_1^E and S_2^E after embedding. The simplified location map covers all quads from the sets S_2 and S_3. All quads from the set S_2 are marked by "1" and all quads from set S_3 by "0". The location map size is calculated as $|L| = |S_2| + |S_3|$. Payload size is $|P| = 3 \cdot (|S_1| + |S_2|)$.

For avoiding overflow and underflow, the extended quad pixels U_0, U_1, U_2 and U_3 should satisfy the following conditions:

$$0 \leq U_0 \leq 255, 0 \leq U_2 \leq 255, 0 \leq U_3 \leq 255, 0 \leq U_4 \leq 255$$

We propose to keep v_0 unchanged after bits being embedded and then to sort them for solving overflow and underflow problems. Those quads under the threshold T, which possibly cause overflow and underflow, should be found first. In these questioned quads v_0 is usually close to 0 or 255. Quads that v_0 are around 128 have no overflow and underflow problems. For the purpose of avoiding overflow and underflow, we propose to sort all quads according to the magnitude $\mu = |128 - v_0|$. Thus, all questioned quads are moved to end of the sorted row of quads. All questioned quads and part of unquestioned quads are unified in set R by using two parameters v_0^t (top) and v_0^b (bottom). Here, v_0^t is the smallest v_0 value among all questioned quads with $v_0 > 128$. Similarly, v_0^b is the biggest v_0 of among all questioned quads with $v_0 < 128$. The set R includes those quads satisfying the condition $v_0 \geq v_0^t$ and $v_0 > 128$ and those quads satisfying $v_0 \leq v_0^b$ and $v_0 < 128$. All quads from the set R are marked in O/U location map. These questioned quads in the set R are marked by "1" and unified to the "questioned" set R_q, whereas the unquestioned quads are marked by "0" and unified to the "unquestioned" set R_u. Thus, for extracting embedded data and recovering original image two location maps are needed (the simplified location map and the O/U location map).

The embedding process of the proposed data hiding technique for embedding payload P_n to the quads Q is described as follows:

1) Compute v_0, v_1, v_2, v_3 and μ for all quads.
2) Sort quads according to μ.
3) Find the necessary threshold T for the sets S_1, S_2, S_3, S_4, the simplified location map L and the O/U location map $L_{O/U}$
   (a) Assign $T = 1$
   (b) Find S_1, S_2, S_3, S_4 using the histogram of Q under the threshold T.
   (c) Find the parameter values v_0^t , v_0^b for the set R and the O/U location map $L_{O/U}$.
   (d) Exclude the set R from the sets S_1 and S_2.

(e) Compute the location map L with the size of $|L| = |S_2| + |S_3|$ and the possible embedding payload by the expression $|P| = 3 \cdot (|S_1| + |S_2| + |R_u|) - |L| - |L_{O/U}|$

(f) If $|P| < |P_n|$ or $|S_1| < |L| + |L_{O/U}|$, increase the threshold $T = T + 1$ and repeat step 1.a - 1.e

4) Embed the simplified location map L and the O/U location map $L_{O/U}$ onto the set S_1. If $|L| + |L_{O/U}| < |P|$, add a part of payload P_I to those unused subset of S_1. Here, we have $|P_I\ | = |\ S_1| - |L| - |L_{O/U}\ |$.

5) Embed the other part of payload P_{II} to set S_2 and R_u. $|\ P_{II}\ | = |P| - |P_I\ |$.

Thus, the payload $P_n = P_I + P_{II}$ is successfully added to image.

Decoding processing of the proposed data hiding technique is presented as follows:

Decoding data:

The embedded bits b_i are decoded as follows:

$$b_i\ =\ V_i\ mod\ 2 \tag{11}$$

where Vi are the extended difference values. i=1,2,3.

The original difference values vi are computed as follows:

$$v_i = \left\lfloor \frac{V_i}{2} \right\rfloor \tag{12}$$

Assume that the threshold T and the parameter values v_0^t, v_0^b are known for the decoder. Decoder can distinguish three sets: $S_1^E, I = S_2^E \cup\ S_3$ and S_4 (see Fig 1).

Decoding process:

1) Compute v_0, V_1, V_2, V_3 and μ for all quads.
2) Sort the quads according to μ.
3) Find the set R by using the parameter values v_0^t, v_0^b and the threshold T.
4) Find the sets: S_1^E and I.
5) Exclude set R from the sets S_1^E and I by using the parameters values v_0^t, v_0^b.
6) Extract the payload P_I using the simplified location map L and O/U location map $L_{O/U}$ from set S_1^E. Recover the original quad pixels from set S_1^E
7) Use the simplified location map L and O/U location map $L_{O/U}$ to extract payload P_{II} and recover original quad pixels from set I and R_u.

Thus, after decoding the values h, the hidden message P_n is recovered exactly.

4 Experimental Results

Efficiency of the proposed reversible watermarking algorithm was estimated by using different test images in comparing with Alattar's different expansion method based on quads. The proposed method better improves the performances for all estimated capacities. Efficiency of the proposed method is very significant

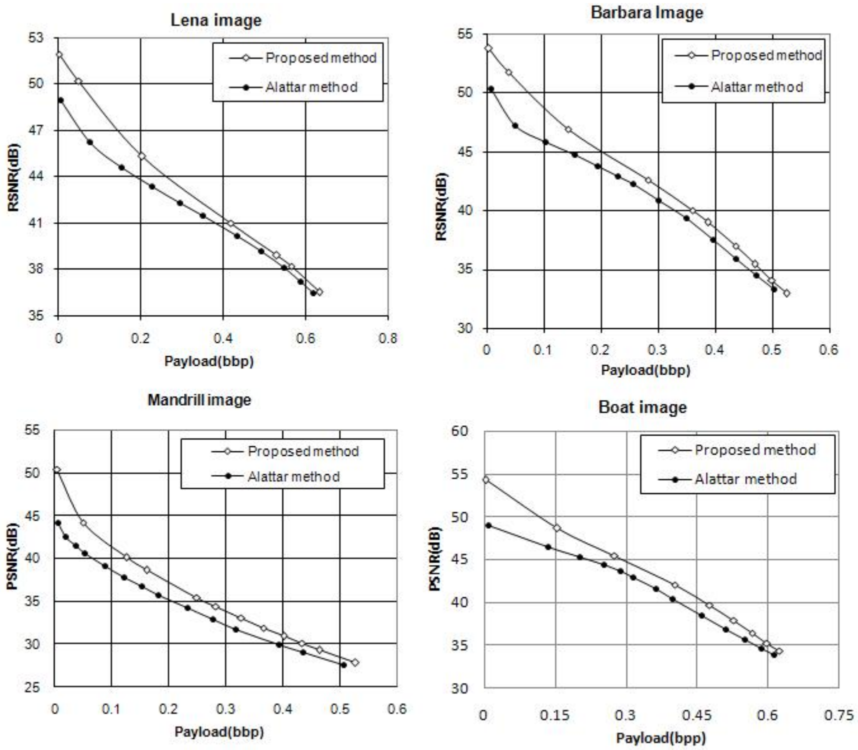


Fig. 3. Prediction pattern

for small capacities, where the location map compression for the Alattar's quad based method is inefficient.

Figure 3 shows the experimental results using four different test images (*Barbara, Mandrill, Lena and Boat*). The proposed reversible watermarking method is able to hide 0.2 bpp of payload into Lena image with embedding distortion of PSNR of 45.30 dB. Alattar's quads based method has around PSNR of 43.89 dB under the condition of embedding the same capacity of data. Thus, the proposed method enhances 1.41 dB PSNR, meaning the embedding distortion is smaller. When capacity is 0.6 bpp, the PSNR value of the proposed method is 37.02 dB, where Alattar's method has the PSNR of 36.62 dB under the same embedding capacity. Our method increases the PSNR of 0.4 dB. As for others images, the similar conclusions can be drawn. The proposed method also has a better performance for small capacity embedding, because the compression ratio of the location map is usually lower for the Alattar's method.

For a larger capacity embedding, since the compression ratio is quite higher for the Alattar's method, whereas the proposed simplified location map is just slightly lower. This is why for large capacities, performance of our method is similar (or little higher) to the Alattar's method.

5 Conclusion

The proposed reversible data hiding technique based on a simplified location map. As a bonus, the proposed method will be able to further improve the embedding capacity in comparison with the previous existing methods such as Tian's different expansion method [15], the later improved the sorting methods of Kamstra and Heijmans [10], the improved difference expansion method of Alattar based on triplets [1] quads [2]. The performance enhancement of the proposed data hiding technique is due to using the simplified location map, which is usually smaller than the location map size of those existed methods. The smaller location map, the larger embedding capacity according to the reversible watermarking point of view, so the proposed method has a better PSNR under the same payload or bigger embedding capacity with the same PSNR value. In addition, in the proposed algorithm, lossless compression algorithm is unnecessary for location map compression. Skipping the step of location map compression in reversible watermarking scheme is beneficial to solving serious capacity control problem. Instead, an appropriate threshold for embedding necessary capacity can be easy to find by using the histogram of the input signal. Experimental works show that the proposed method has achieved better results for all four well-known test images.

Acknowledgments

This work was in part supported by the Information Technology Research Center (ITRC) by the Ministry of Information and Communication, Korea.

References

1. Alattar, A.M.: Reversible watermark using dierence expansion oftriplets. In: Proceedings of the International Conference on Image Processing, vol. 1, pp. 501–504 (2003)
2. Alattar, A.M.: Reversible watermark using dierence expansion of quads. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 377–380 (2004)
3. Alattar, A.M.: Reversible watermark using the dierence expansion of a generalized integer transform. IEEE Transactions on Image Processing 13(8), 1147–1156 (2004)
4. Bao, F., Deng, R.H., Ooi, B.C., Yang, Y.: Tailored reversible watermarking schemes for authentication of electronic clinical atlas. IEEE Transactions on Information Technology in Biomedicine 9(4), 554–563 (2005)
5. Barton, J.M.: Method and apparatus for embedding authentication information within digital data, U.S. Patent 5,646,997 (1997)
6. Celik, M., Sharma, G., Tekalp, A.M., Saber, E.: Reversible data hiding. In: Proceedings of the International Conference on Image Processing, Rochester, NY, pp. 157–160 (2002)
7. Fridrich, J., Goljan, M., Du, R.: Invertible authentication. In: Proc. SPIE, Security and Watermarking of Multimedia Contents, SanJose, CA, pp. 197–208 (2001)

8. Goljan, M., Fridrich, J., Du, R.: Distortion-free data embedding. In: Proceedings of the Information Hiding Workshop, Pittsburgh, PA, pp. 27–41 (2001)
9. Honsinger, C.W., Jones, P., Rabbani, M.: Lossless recovery of an original image containing embedded data, US Patent 6,278,791B1 (2001)
10. Kamstra, L., Heijmans, H.J.A.M.: Reversible data embedding into images using wavelet techniques and sorting. IEEE Transactions on Image Processing 14(12), 2082–2090 (2005)
11. Macq, B., Deweyand, F.: Trusted headers for medical images. In: DFG VIII-D II Watermarking Workshop, Erlangen, Germany (1999)
12. Ni, Z., Shi, Y.Q., Ansari, N., Su, W., Sun, Q., Lin, X.: Robust lossless image data hiding. In: IEEE International Conference on Multimedia and Expo, Taipei, Taiwn, pp. 2199–2202 (2004)
13. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE transactions on circuits and systems for video technology 16(3), 354–362 (2006)
14. Thodi, D.M., Rodriguez, J.J.: Reversible watermarking by prediction-error expansion. In: IEEE Southwest Symposium on Image Analysis and Interpretation, Lake-Tahoe, CA, pp. 21–25 (2004)
15. Tian, J.: Reversible data embedding using a difference expansion. IEEE Transaction on Circuits and Systems for Video Technology 13(8), 890–896 (2003)
16. de Vleeschouwer, C., Delaigle, J.F., Macq, B.: Circular interpretation of bijective transformations in lossless watermarking for media asset management. IEEE Transactions on Multimedia 5(1), 97–105 (2003)
17. Xuan, G., Zhu, J., Chen, J., Shi, Y.Q., Ni, Z., Su, W.: Distortionless data hiding based on integer wavelet transform. IEE Electronics Letters 38(25), 1646–1648 (2002)
18. Xuan, G., Shi, Y.Q., Ni, Z.C., Chen, J., Yang, C., Zhen, Y., Zheng, J.: High capacity lossless data hiding based on integer wavelet transform. In: Proceedings of IEEE International Conference on Circuits and Systems, Vancouver, Canada (2004)
19. Xuan, G., Shi, Y.Q.: Integer wavelet transform based lossless data hiding using spread spectrum. In: IEEE International Workshop on Multimedia Signal Processing, Siena, Italy (2004)
20. Yang, B., Schmucker, M., Funk, W., Busch, C., Sun, S.: Integer DCT-based reversible watermarking for images using companding technique. In: Proceedings of SPIE, vol. 5306, pp. 405–415 (2004)
21. Yang, B., Schmucker, M., Busch, C., Niu, X., Sun, S.: Approaching optimal value expansion for reversible watermarking. In: Proceedings of the 7th workshop on Multimedia and Security, pp. 95–102 (2005)
22. Zou, D., Shi, Y.Q., Ni, Z.: A semi-fragile lossless data hiding scheme based on integer wavelet transform. In: IEEE International Workshop on Multimedia Signal Processing, Siena, Italy (2004)

Optimum Histogram Pair Based Image Lossless Data Embedding

Guorong Xuan[1,⋆], Yun Q. Shi[2], Peiqi Chai[1], Xia Cui[1], Zhicheng Ni[1], and Xuefeng Tong[1]

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

Abstract. This paper presents an optimum histogram pair based image lossless data embedding scheme using integer wavelet transform and adaptive histogram modification. This new scheme is characterized by (1) the selection of best threshold T, which leads to the highest PSNR of the marked image for a given payload, (2) the adaptive histogram modification, which aims at avoiding underflow and/or overflow, is carried out only when it is necessary, and treats the left side and the right side of histogram individually, seeking a minimum amount of histogram modification, and (3) the selection of most suitable embedding region, which attempts to further improve the PSNR of the marked image in particular when the payload is low. Consequently, it can achieve the highest visual quality of marked image for a given payload as compared with the prior arts of image lossless data hiding. The experimental results have been presented to confirm the claimed superior performance.

Keywords: Optimum histogram pair, lossless data embedding, integer wavelets, selection of best threshold, adaptive histogram modification, selection of suitable embedding region.

1 Introduction

Lossless data embedding requires that not only hidden data can be extracted correctly but also the marked image can be inverted back to the original cover image exactly after the hidden data has been extracted out. Recently, Ni et al. [1] proposed the lossless data embedding algorithm based on the spatial domain histogram shifting. Tian [2] proposed the difference expansion method that can achieve a high payload, but it can only be used with integer Haar wavelet. Kamstra and Heijmans [3] improved the PSNR of marked image achieved by Tian in the case of small payload. Xuan et al. [4] proposed the thresholding lossless embedding using the integer wavelet transform (IWT) and histogram modification.

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

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 264–278, 2008.

In [5], some improvements over [4] have been made about the selection of threshold. However, the three optimality measures taken in this paper have not been developed in [5]. In [6], Yang et al. applied the histogram shifting embedding to the integer discrete cosine transform. The lossless data hiding scheme proposed in this paper is based on optimum histogram pairs. It is characterized by selection of optimum threshold T, most suitable embedding region R, and minimum possible amount of histogram modification G, in order to achieve highest PSNR of the marked image for a given data embedding capacity.

The rest of this paper is organized as follows. The principle of reversible data embedding based on histogram pairs is illustrated in Section 2. Integer wavelets and histogram modification are briefly discussed in Section 3. The proposed reversible data hiding algorithm is presented in Section 4 and 5. Section 6 provides experimental results. The performance of three newest algorithms published in 2007 [7,8,9] is cited for comparison. Discussion and conclusion are presented in Section 7.

2 Principle of Histogram Pair

2.1 Lossless Data Embedding Using Histogram Pair

Histogram $h(x)$ is the number of occurrence as the variable X assumes value x. Since digital image and IWT are considered in this paper, we assume X can only assume integer values. In order to illustrate the concept of histogram pair, we first consider a very simple case. That is, only two consecutive integers a and b assumed by X are considered, i.e. $x \in a, b$. Furthermore, let $h(a) = m$ and $h(b) = 0$. We call these two points as a histogram pair, and sometimes denote it by, $h = [m, 0]$, or simply $[m, 0]$. Furthermore, we assume $m = 4$. That is, X actually assumes integer value a four times, i.e., $X = [a, a, a, a]$. Next, let's see how we can losslessly embed bits into $X = [a, a, a, a]$. Suppose the to-be-embedded binary sequence is $D = [1, 0, 0, 1]$. In data embedding, we scan the 1-D sequence $X = [a, a, a, a]$ in certain ordering, say, from left to right. When we meet the first a, since we want to embed bit 1, we change a to b. For the next two to-be-embedded bits, since they are bit 0, we do not change a. For the last to-be-embedded bit 1, we change a to b. Therefore, after the four-bit embedding, we have $X = [b, a, a, b]$, and the histogram is now $h = [2, 2]$. Embedding capacity is $C = 4$. The hidden data extraction, or histogram pair recovery, is the reverse process of data embedding: after extracting the data $D = [1, 0, 0, 1]$, the histogram pair becomes $[4, 0]$ and we can recover $X = [a, a, a, a]$ losslessly.

We define histogram pair here. If for two consecutive non-negative integer values a and b that X can assume, we have $h(a) = m$ and $h(b) = n$, where m and n are the numbers of occurrence for $x = a$ and $x = b$, respectively. When $n = 0$, we call $h = [m, n]$ as a histogram pair. From the above example, we observe that when $n = 0$, we can use this histogram pair to embed data losslessly. We call $n = 0$ in the above defined histogram pair as an "expansion" element, which is ready for lossless data embedding. During the embedding, we scan all of x values, a, in certain order. If bit 1 is to be embedded, we change the a under scanning to b, otherwise, we keep the a unchanged. If a is a negative integer, then $h = [m, n]$ is a histogram-pair as $m = 0$ and $n \neq 0$.

2.2 Information Theory

Histogram expansion and data embedding will cause the histogram change from up-and-down to relatively more flat. In this way, the entropy increases. On the other hand, data extraction will lead to the opposite, and the entropy decreases to its original value. This process can be shown below.

The entropy can be expressed as $H(x) = -\int h(x)\log(h(x))dx$. It can be proved according to information theory that the probability distribution $h(x)$ will be uniform within a limited range $u \sim v$ when the maximum entropy $H(x)$ is achieved:

$$h(x) = \arg_{h(x)} \max[H(x) - \lambda(1 - \int_u^v h(x)dx)] = \frac{1}{v-u} \tag{1}$$

Proof. If $d[h(x)]/dx = 0$, we have $-(1+\log h(x))+\lambda = 0$. Consider the condition of probability distribution $\int_u^v h(x)dx = 1$, the solution $h(x) = \frac{1}{v-u}$ is obtained.

In summary, after data embedding, the entropy increases and histogram becomes more flat. If the histogram is absolutely flat, the total entropy $H(x)$ is maximum, no more data can be embedded.

3 Integer Wavelets and Histogram Adjustment

3.1 Integer Wavelet Transform (IWT)

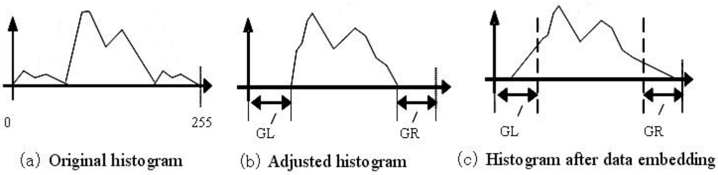
In this proposed method, data is hidden into IWT coefficients of high-frequency subbands. The motivation of doing so is as follows. (1) The high frequency subband coefficients represent the high frequency components of the image. Human visual system is less sensitive to high frequency. Hence, data embedding into high frequency subbands can lead to better imperceptibility of marked image. (2) The histogram distribution of high-frequency subbands is Laplacian-like with a huge peak around zero. This makes high data embedding capacity feasible. (3) Owing to the de-correlation property among the wavelet subbands in the same decomposition level, data embedding using IWT results in higher PSNR than embedding into other transform coefficients such as DCT.

Due to the losslessness constraint, we choose to use IWT. Specifically, the integer Haar wavelet transform and integer (5,3) wavelet transform are used in our experimental works. The results are shown in Section 5, from which we observe that both perform well, however, the integer (5,3) transform performs better, while the integer Haar transform is more simple in implementation.

3.2 Histogram Modification

For a given image, after data embedding into some IWT coefficients, it is possible to cause *underflow* and/or *overflow*, which means that the grayscale values may exceed the range $[0, 255]$ for an 8-bit gray image, thus violating the losslessness constraint. In order to prevent this from happening, we adjust histogram, i.e., we shrink the histogram from one or both sides towards the central portion. In narrowing down a histogram, we need to record the histogram modification (referred to as bookkeeping information) as a part of the embedded data. Instead of adjusting histogram as a preprocessing as done in [4] and [5] (meaning that histogram adjustment is always done at the beginning no matter if necessary or not), we do it only when this is necessary (meaning that the adjustment is done adaptively in data embedding when it is necessary), furthermore the left side and right side of histogram is treated individually. More discussion in this regard will be given below.

If the overflow occurs (at the right side, i.e., the grayscale value is larger than 255), the right end of the histogram will be shrunk towards the center by an amount GR. If the underflow occurs (at the left side, i.e., the grayscale value is smaller than 0), the left end of the histogram will be shrunk towards the center by an amount GL. Together the histogram is shrunk by an amount of G, and $G = GL + GR$. The histogram is narrowed down from "0 to 255" to "GL to $(255 - GR)$". This histogram shrinkage uses the histogram pair principle described above as well, specifically it is the reverse process of data embedding, i.e., the data extraction using histogram pair. This new dynamic way contributes to the superior performance over that of [4] and [5] and will be shown in Sections 5 and 6. There, it is observed that it may not need to do histogram modification for some images with some payloads. When the embedding capacity increases, we may need histogram modification. In addition, the amount of histogram modification for the right side and the left side of histogram may be different. All of these have been implemented with an efficient computer program in this proposed scheme. Fig. 3 shown below is an illustration of histogram modification.



(a) Original histogram (b) Adjusted histogram (c) Histogram after data embedding

Fig. 1. Histogram modification

4 Proposed Reversible Data Hiding Algorithm

4.1 Thresholding Method

To avoid the possible underflow and/or overflow, often only the wavelet coefficients with small absolute value are used for data embedding [4]. This is the so-called thresholding method, which first sets the threshold T depending on the payload and embeds the data into those IWT coefficients with $|x| \leq T$. It does not embed data into the wavelet coefficients with $|x| > T$. Furthermore, for all high frequency wavelet coefficients with $|x| > T$ we simply add T or $-T$ to x depending x is positive or negative so as to make their magnitude larger than $2T$. In this way, the data embedding into coefficients with $|x| \leq T$ will not be confused with the large coefficients in which there is no data embedding. Therefore, the so-called thresholding method [4] is in fact the *minimum* thresholding method.

4.2 Optimum Thresholding Method Based on Histogram Pairs

As shown below, however, the minimum threshold T does not necessarily lead to the highest PSNR of mark image for a given payload. (This was also reported in [5].) The reason is as follows. If a smaller threshold T is selected, the number of coefficients with $|x| > T$ will be larger. The magnitudes of these coefficients need to be increased by a certain amount in order to create histogram pairs to embed data. This may lead to a lower PSNR owing to moving a larger end part of histogram. On the other hand, if a larger T is selected, more coefficients having larger magnitude are to be changed for data embedding, possibly resulting in a lower PSNR of the marked image. Instead of arbitrarily picking up some threshold T (as the starting point for data embedding) and some stopping point S for stopping data embedding as done in [5], it is found that for a given data embedding capacity there does exist an optimum value for T. In this proposed optimum histogram pair lossless embedding, the best threshold T for a given data embedding capacity is searched with computer program automatically and selected to achieve the highest PSNR for the marked image. This will be discussed in Section 5.

The proposed method divides the whole histogram into three parts: (1) the 1^{st} part where data is to be embedded; (2) the central part - no data embedded and the absolute value of coefficients is smaller than that in the 1^{st} part; (3) the end part - no data embedded and the absolute value of coefficients is larger than that in the 1^{st} part. The whole embedding and extraction procedure can be expressed by the formulae in Table 1, where T is the selected threshold, i.e., start position for data embedding, S is stop position, x is feature (wavelet coefficient) values before embedding, x' is feature value after embedding, $u(S)$ is unit step function (when $S \geq 0, u(S) = 1$, when $S < 0, u(S) = 0$), $\lfloor x \rfloor$ rounds x to the largest integer not larger than x. A simple example is presented in Section 4.5 to illustrate these formulae.

4.3 Data Embedding Algorithm

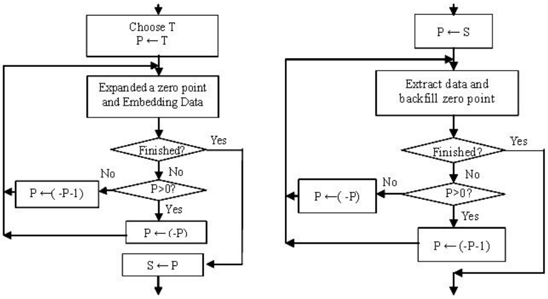
The high frequency subbands (HH, HL, LH) coefficients of IWT are used for data embedding in this proposed method. Assume the number of bits to be embedded is L. The data embedding steps are listed below.

(1) For a given data embedding capacity, apply our algorithm to the given image to search for an optimum threshold T as shown in Fig. 5 of Section 5. And set the $P \leftarrow T$, where T is a starting value for data embedding.

Table 1. Formulas of optimum thresholding method of lossless data hiding

	Embedding		Recovering	
parts of histogram	after embedding	condition	after recovering	condition
Data hiding region (right side) (positive or zero)	$x' = 2x + b - \|S\|$	$\|S\| \leq x \leq T$	$x = \lfloor \frac{(x'+\|S\|)}{2}, b = x' + \|S\| - 2x$	$\|S\| \leq x' \leq 2T - 1 - \|S\|$
Data embedded region (left side) (negative)	$x' = 2x - b + \|S\| + u(S)$	$-T \leq x \leq -\|S\| - u(S)$	$\lfloor \frac{(x'-\|S\|-u(S)+1)}{2} \rfloor, b = x' + \|S\| - 2x$	$-2T - 1 + \|S\| + u(S) \leq x' \leq -\|S\| - u(S)$
No data embedded region (central part) (small absolute value)	$x' = x$	$\|S\| < x < \|S\|$	$x = x'$	$-\|S\| - u(S) < x' < \|S\|$
No data embedded region (right edge part) (positive)	$x' = x + T + 1 - \|S\|$	$x > T$	$x = x' - T - 1 + \|S\|$	$x' > 2T + 1 - \|S\|$
No data embedded region (left edge part) (negative)	$x' = x - T - 1 - \|S\| + u(S)$	$x < -T$	$x = x' + T + 1 - \|S\| - u(S)$	$x' < -2T - 1 + \|S\| + u(S)$

(2) In the histogram of high frequency wavelet coefficients, move the portion of histogram with the coefficient values greater than P to the right-hand side by one unit to make the histogram at $P+1$ equal to zero (call $P+1$ as a zero-point). Then embed data in this point.
(3) If some of the to-be-embedded bits have not been embedded yet, let $P \leftarrow (-P)$, and move the histogram (less than P) to the left-hand side by 1 unit to leave a zero-point at the value $(-P-1)$. And embed data in this point.
(4) If all the data have been embedded, then stop embedding and record the P value as the stop value, S. Otherwise, $P \leftarrow (-P-1)$, go back to (2) to continue to embed the remaining to-be-embedded data, where S is a stop value. If the sum of histogram for $x \in [-T, T]$ is equal L, the S will be zero.



(a) embedding

(b) extracting

Fig. 2. Flowchart of proposed lossless data embedding and extracting

4.4 Data Extraction Algorithm

The data extraction is the reverse of data embedding. Without loss of generality, assume the stop position of data embedding is S, $S > 0$. Steps are as follows.

(1) Set $P \leftarrow S$.
(2) Decode with the stopping value P. Extract all the data until $P + 1$ becomes a zero-point. Move all the histogram greater than $P + 1$ towards the left-hand by one unit to cover the zero-point.
(3) If the extracted data is less than L, set $P \leftarrow (-P - 1)$. Continue to extract data until it becomes a zero-point in the position $(P - 1)$. Then move histogram (less than $P - 1$) towards the right-hand side by one unit to cover the zero-point.
(4) If all the hidden bits have been extracted, stop. Otherwise, set $P \leftarrow -P$, go back to (2) to continue to extract the data.

4.5 A Simple Yet Complete Example

In this simple but complete example, the to-be-embedded bit sequence $D = [110001]$ has six bits and will be embedded into an image by using the proposed histogram pair scheme with threshold $T = 3$, and stop value $S = 2$. The image 5×5 shown in Fig. 3 (a) has 12 distinct feature (grayscale) values, i.e., $x \in [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6]$. The grayscale values of this image have the histogram $h_0 = [0, 1, 2, 3, 4, 6, 3, 3, 1, 2, 0, 0]$ (as shown in 1st row of Fig. 4). As said before, for $x \geq 0$, the histogram pair is of form h=[$\underline{\text{m,0}}$], for $x < 0$, the histogram pair is $h = [\underline{0, n}]$. The 2nd row of Fig. 4 is expanded image histogram: h_1 (expanded), it has three histogram pairs. The 1st histogram pair is in the far-right-hand side $h = [\underline{1, 0}]$; the 2nd histogram pair is in the left-hand side $h = [\underline{0, 2}]$; the 3rd histogram pair is in the right-hand side near the center $h = [\underline{3, 0}]$. The 3rd row of Fig. 4 is the image histogram after data embedding: h_2 (bits embedded).

0	4	0	-4	1
0	$\boxed{2}$	-2	3	-1
4	-3	0	$\boxed{2}$	-3
-1	-2	0	-1	0
-2	1	$\boxed{2}$	-1	1

(a)

0	6	0	-5	1
0	$\boxed{2}$	-2	4	-1
6	-3	0	$\boxed{2}$	-3
-1	-2	0	-1	0
-2	1	$\boxed{2}$	-1	1

(b)

0	6	0	-5	1
0	$\boxed{2}$	-2	5	-1
6	-4	0	$\boxed{2}$	-3
-1	-2	0	-1	0
-2	1	$\boxed{3}$	-1	1

(c)

Fig. 3. 5×5 wavelet subband (a) original one, (b) after 3 expanding, (c) after 6-bit embedding (what marked is how the last 3 bits are embedded)

Fig. 4 uses solid(orange) line squares to mark the third histogram pair. The first histogram pair $[\underline{1, 0}]$ is used to embed the 1st bit 1, the second histogram pair [$\underline{\text{0,2}}$] is used to embed the next two bits 1,0, and the third histogram pair $[\underline{3, 0}]$ is used to embed three bits: 0,0,1.

During expanding, we are first making $h(4) = 0$, then making $h(-4) = 0$, finally making $h(3) = 0$. Note that $h(3) = 0$ at this time makes $h(4) = 0$

"shifting" to $h(5) = 0$. During each zero-point creation, the histogram shifting towards one of two (left and right) ends is carried out, the resultant histogram becomes $h_1 = [1, \underline{0, 2}, 3, 4, 6, 3, \underline{3, 0}, \underline{1, 0.2}]$ (refer to Fig. 3 (b) and 2^{nd} row of Fig. 4). There histogram pairs are thus produced: in the right-most $h = [\underline{1, 0}]$, in the left $h = [\underline{0, 2}]$ and in the right (near center) $h = [\underline{3, 0}]$. After data embedding with bit sequence $D = [110001]$ and the selected scanning order, the histogram becomes $h_2 = [1, \underline{1, 1}, 2, 4, 6, 3, \underline{2, 1}, \underline{0, 1}, 2]$ (refer to Fig. 3 (c) and 3^{rd} row of Fig. 4). The three histogram pairs changed: in the right most from $h = [1, 0]$ to $h = [0, 1]$, in the left from $h = [0, 2]$ to $h = [1, 1]$, and in the right (near center) from $h = [3, 0]$ to $h = [2, 1]$.

After embedding, the grayscale values changed too. For example, embedding the last three bits (001) causes the right histogram pair (near center) to change from $h = [3, 0]$ to $h = [2, 1]$, and three grayscale values marked with small rectangles to change from $X = [\boxed{2}, \boxed{2}, \boxed{2}]$ to $X = [\boxed{2}, \boxed{2}, \boxed{3}]$ (refer to Fig. 3 (c) and 3^{rd} row of Fig. 4).

Through this example, it becomes clear that the threshold can also be viewed as the starting point to implement histogram pair lossless data hiding. The formulas associated with this example are shown in Table 3, which provides a specific illustration of general formulas in Table 1.

Table 2. Histogram pair data embedding example ($T = 3, S = 2$) (6 bit sequence D=[1 10 001]) (What marked is how the last 3 bits are embedded.)

X	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
h_0(original)		1	2	3	4	6	3	3	1	2		
h_1(extended)	1	0	2	3	4	6	3	3	1	2		
h_2(embedded)	1	1	1	2	4	6	3	2	1	0	1	2
embedded (ordering)	no embedding	[1 0] embedded (second)		no embedding				[0 0 1] embedded (third)		[1] embedded (first)		no embedding

4.6 Data Embedding Capacity

Data embedding capacity L can be calculated as follows. Without loss of generality, assume $T > 0$. When the stopping value S is negative: $L = \sum_{-T}^{S} h(X) + \sum_{-S}^{T} h(X)$; when S is positive: $L = \sum_{-T}^{-S-1} h(X) + \sum_{S}^{T} h(X)$; and when S is zero: $L = \sum_{-T}^{-1} h(X) + \sum_{0}^{T} h(X) = \sum_{-T}^{T} h(X)$. As $T < 0$, the capacity can be determined accordingly.

5 Selection of Optimum Parameters

For a given required data embedding capacity, the proposed method selects the optimum parameter to achieve the highest possible PSNR. The optimum parameters include: the best threshold T, the adaptive histogram modification

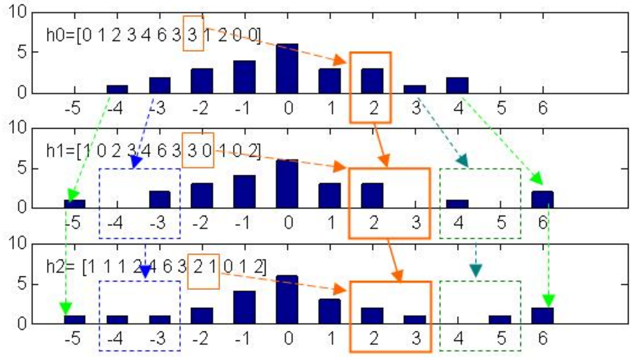


Fig. 4. Histogram pair data embedding example ($T = 3, S = 2$ to-be-embedded bit sequence D=[1 10 001]). What marked in solid (orange) line squares shows how the last 3 bits are embedded.

Table 3. Formulas of lossless data hiding in the example of Section 4.5

	Embedding		Recovering	
	after embedding	condition	after recovering	condition
central	$x' = x, x' = [2, -1, 0, 1]$	$if - 2 < x < 2, x = [-2, -1, 0, 1]$	$x = x', x = [-2, -1, 0, 1]$	$if - 2 - u(S) < x' < \|S\|,$ x'=[-2,-1,0,1]
right end	$x' = x + 2, x' = [6]$	$if x > 3, x = [4]$	$x = x' - 2, x = [4]$	$x' > 5, x' = [6]$
left end	$x' = x - 1, x' = [-5]$	$if x < -3, x = [-4]$	$x = x' + 1, x = [-4]$	$x' > -4, x' = [-5]$
right to be embedded	$x' = 2x + b - 2, b = 0 : x' = [2, 4], b = 1 : x' = [3, 5]$	$if 2 \leq x \leq -3, b = 0 : x = [2, 3], b = 1 : x = [2, 3]$	$x = \lfloor \frac{x'+2}{2} \rfloor, b = x' + 2 - 2x, b = 0 : x = [2, 3], b = 1 : x = [2, 3]$	$if 2 \leq x' \leq 5, b = 0 : x' = [2, 4], b = 1 : x = [3, 5]$
left to be embedded	$x' = 2x - b + 3, b = 0 : x' = [-3], b = 1 : x' = [-4]$	$if -3 \leq x \leq -3, b = 0 : x = [-3], b = 1 : x = [-3]$	$x = \lfloor \frac{x'-2}{2} \rfloor, b = x' - 3 - 2x, b = 0 : x = [-3], b = 1 : x = [-3]$	$if - 4 \leq x' \leq -3, b = 0 : x' = [-3], b = 1 : x = [-4]$

value G (in spatial domain), and the suitable data embedding region R. That is, optimal parameters can be selected as follows.

$$[T, G, R] = \arg_{T,G,R} \max (PSNR) \tag{2}$$

(1) Best threshold T: Fig. 5 shows the PSNR of marked image vs the threshold T with embedding capacity 0.02 bpp. It is observed there does exist an optimal threshold T for each of the three commonly used images: Lena, Baboon and Barbra.

(2)Adaptive modification value G: the histogram modification is carried out in this method adaptively on real time, instead of as a necessary preprocessing in [4,5]. That is, after data embedding into each wavelet coefficient, underflow and/or overflow is checked. If underflow and/or overflow occurs, and it occurs from the left side (< 0), the left end of the histogram will be shrunk towards the

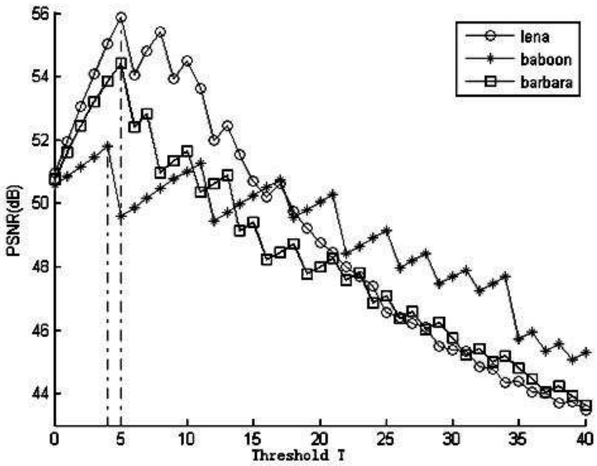


Fig. 5. Selection of the best threshold T with embedding capacity 0.02 bpp

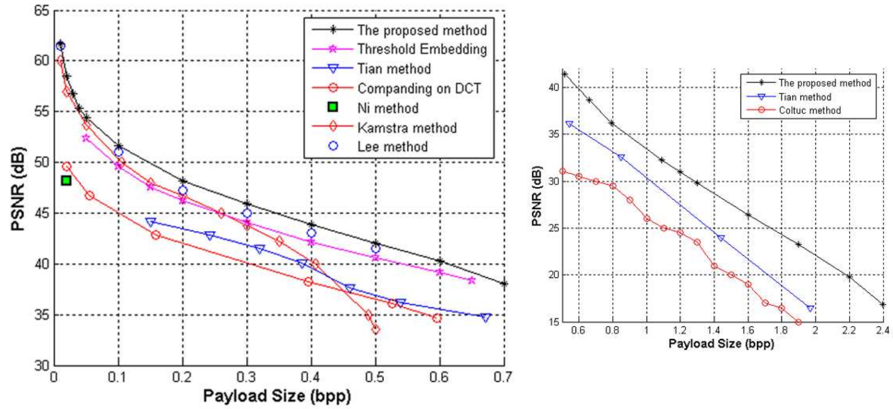
center by an amount GL. The right hand, hence, GR is similarly handled. Our experiments have shown that only when the embedding data rate is higher than certain amount it needs histogram modification ($G > 0$). Otherwise, there is no need for histogram modification. This adaptive histogram modification leads to the higher PSNR of marked image for a given payload.

(3) Suitable data embedding region R: in order to improve the PSNR when the payload is small (e.g., < 0.1 bpp), we choose to only embed data into the HH subband , i.e., $R = HH$. When the payload is large, all three high frequency subbands are used, i.e., $R = HH, HL, LH$. This measure further enhances the PSNR for a given payload. Our experimental works have shown that the PSNR achieved by this new method is obviously better than that by the method [5] when the payload is rather small.

6 Experiments

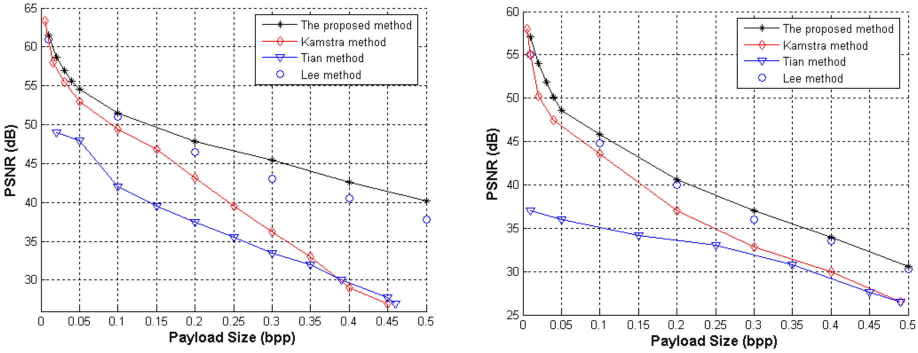
6.1 Experimental Results and Performance Comparison

For Lena, Barbra and Baboon three commonly used images (all in 512 × 512), we compare the results of our proposed histogram pair technique with that of other techniques, i.e., techniques of [1,2,3,4,5,6,7,8,9]. The results of performance comparison are shown in Fig. 6, 7, and 8. What shown in Fig. 8 (a) is the original Lena image and the marked Lena images generated by using the proposed method with the same group of three different payloads as used in [2], in Fig. 8 (b), the corresponding test results of [9] are shown. From these figures, it is observed that the PSNR achieved by our proposed method for the same embedded payload is the highest among the nine methods for these three commonly used test images.



(a) (b)

Fig. 6. (a) Performance comparison on Lena (b) Comparison of multiple-time data embedding into Lena image among [2],[8] and the proposed method



(a) (b)

Fig. 7. (a) Comparison on Barbara (b) Comparison on Baboon

In Table 4, the three parameters, GR, GL, and G, are listed when applying our proposed adaptive histogram modification method to the above-mentioned three commonly used images with various embedding rates. It is interesting to observe that only when the embedding data rate for Lena is greater than 1.0 bpp, for Barbara greater than 0.57 bpp, and for Baboon greater than 0.008 bpp, it needs histogram modification ($G > 0$). Otherwise, there is no need for histogram modification when losslessly embedding data into these three commonly used images.

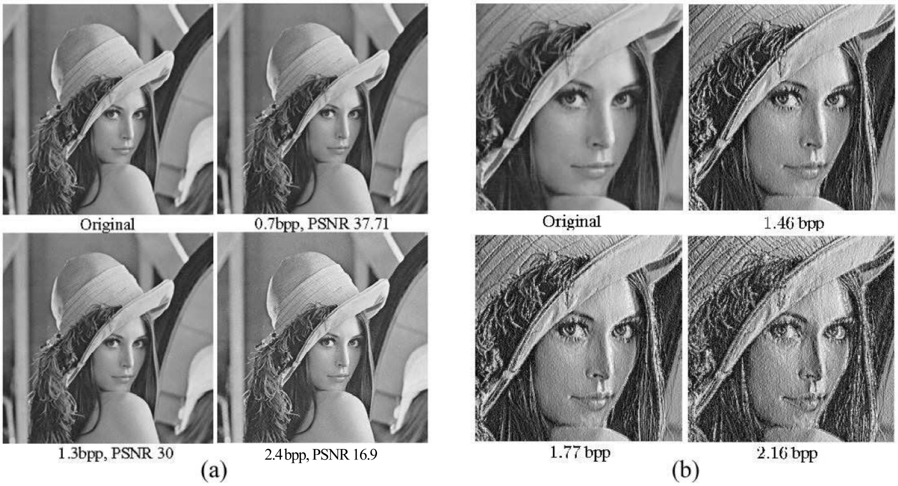


Fig. 8. (a) Original and marked Lena image with three different payloads by the proposed method (b) Performance on Lena image reported in [9]

Table 4. *GL* and *GR* values when using our lossless data embedding algorithm

Fig. 6 (a),6 (b),10 Lena					Fig. 7 (a) Barbara					Fig. 7 (b) Baboon				
bits	bpp	GL	GR	G	bits	bpp	GL	GR	G	bits	bpp	GL	GR	G
3132	0.0119	0	0	0	2819	0.0108	0	0	0	1037	0.0040	0	0	0
6744	0.0257	0	0	0	6041	0.0230	0	0	0	2089	0.0080	0	0	0
11986	0.0457	0	0	0	13064	0.0498	0	0	0	5082	0.0194	2	0	2
21206	0.0809	0	0	0	20095	0.0767	0	0	0	14936	0.0570	6	0	6
36421	0.1389	0	0	0	31729	0.1210	0	0	0	31148	0.1188	8	0	8
58672	0.2238	0	0	0	42639	0.1627	0	0	0	55990	0.2136	10	0	10
82720	0.3156	0	0	0	66702	0.2544	0	0	0	80122	0.3056	13	0	13
109009	0.4158	0	0	0	98430	0.3755	0	0	0	99015	0.3777	15	0	15
135062	0.5152	0	0	0	119672	0.4565	0	0	0	125005	0.4769	18	0	18
172983	0.6599	0	0	0	133784	0.5103	0	0	0	141066	0.5381	22	0	22
207273	0.7907	0	0	0	150320	0.5734	0	0	0					
285027	1.0873	0	0	0	174944	0.6674	0	3	3					
317999	1.2131	2	4	6	180910	0.6901	5	15	18					
336236	1.2826	6	22	28										
430482	1.6422	22	24	46										
510079	1.9458	42	48	90										

6.2 Comparison by Using Integer (5,3) and Haar Wavelets

Table 5 and Table 6 list the PSNR of marked image versus payload (bpp), and parameters of G, T, and S of the proposed lossless data embedding when using integer (5,3) and Haar wavelet, respectively. As expected, integer (5,3) wavelet provides higher PSNR for a given payload than integer Haar wavelet, while

Table 5. PSNR of lossless data embedding using (5,3) IWT

Payload (bpp)	Lena				Barbara				Baboon			
	T	S	PSNR	G	T	S	PSNR	G	T	S	PSNR	G
0.05	4	-4	54.1	0	4	3	53.8	0	4	-3	49.7	0
0.1	0	0	51.2	0	2	-2	50.6	0	4	-3	45.5	0
0.2	2	1	47.8	0	3	-2	47.5	0	3	0	40.1	2
0.3	2	-1	45.4	0	2	0	49.4	0	5	0	36.4	13
0.4	2	0	43.4	0	3	0	42.2	0	8	0	34.1	18
0.5	3	0	41.5	0	4	0	39.7	0	13	0	30.8	22

Table 6. PSNR of lossless data embedding using Haar IWT

Payload (bpp)	Lena				Barbara				Baboon			
	T	S	PSNR	G	T	S	PSNR	G	T	S	PSNR	G
0.05	2	2	52.7	0	2	-2	52.1	0	7	-6	48.1	1
0.1	2	-2	50.1	0	2	1	46.9	0	5	3	43.6	3
0.2	3	-2	46.9	0	2	0	46.1	0	4	0	38.4	8
0.3	2	0	43.7	0	3	0	42.1	0	8	-1	35	15
0.4	4	-1	41.8	0	4	-1	40.2	0	11	0	31.9	21
0.5	6	-1	39.5	0	7	-1	37.1	0	17	0	29.1	28

integer Haar is simpler in implementation. Specifically, as payload is as low as 0.05 bpp, for these three commonly used test images, the PSNR of marked versus original images achieved by using integer (5,3) filter is at least 1.4 dB higher than that by using Haar IWT; as payload is 0.5 bpp, the PSNR by integer (5,3) is at least 1.7 dB higher than that by using Haar IWT.

7 Discussion and Conclusion

(1) An optimum histogram pair based image lossless data embedding scheme using integer wavelet transform is presented in this paper. It uses the new concept of histogram pair to losslessly embed data into image. Furthermore, it is characterized by the selection of best threshold, adaptive histogram modification parameters and suitable embedding region. The experimental results have demonstrated its superior performance in terms of the visual quality of marked image measured by PSNR versus data embedding capacity over, to our best knowledge, all of the prior arts including [1,2,3,4,5,6,7,8,9].

(2) Different from the method in [5], our proposed method uses histogram pair to losslessly embed data, which provides more flexibility in implementation. The procedure of histogram pair is also used in adaptive histogram modification in an inverse way. That is, the same histogram pair technique is used in data embedding as well as in histogram modification to avoid overflow and underflow.

(3) Furthermore, the new method systematically and computationally selects the best threshold, the adaptive histogram modification parameters GR and GL, and suitable data embedding region. As a result, our experimental works have shown that the PSNR of marked image achieved by this new method is distinctly better than that achieved by the method [5] when the payload is low because the new method only selects IWT coefficients in HH subband for data embedding as the payload is low.

(4) The computational complexity, including optimal histogram pair based data embedding and possible histogram modification, is shown affordable for possible real applications. Specifically, for data embedding ranging from 0.01 bpp to 1.0 bpp into Lena, Barbara and Baboon, the execution time varies from 0.25 sec. to 2.68 sec.

(5) If the data embedding rate is not high (e.g., as shown in Table 4, not higher than 1.0873 bpp for Lena image, 0.5734 bpp for Barbara image, and 0.0080 bpp for Baboon image), the amount of histogram modification G = 0, meaning that the histogram shrinkage is not needed. This means that when the data embedding rate is not larger than a specific amount for a given image, our proposed lossless data hiding method does not need to calculate point-by-point to avoid the possible underflow and/or overflow as required in the methods proposed in [2] or [3]. Under this circumstance, the proposed lossless data hiding becomes very simple in implementation.

(6) The proposed method uses integer (5,3) and Haar wavelet transforms in our experiments. It can also be applied to any other integer wavelet transforms. On the contrary, the difference expansion lossless data hiding method described in [2] cannot be used for arbitrary general integer wavelet transform except integer Haar wavelet transform. Our experimental results in Tables 5 and 6 show that the performance achieved by using integer (5,3) wavelet is better than that by using integer Haar wavelet transform.

Acknowledgement

Authors sincerely thank Ms. Haifeng Xiao and Mr. Patchara Sutthiwan for their time and efforts put into the preparation of this paper at the very early and very final stages, respectively.

References

1. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology 16(3), 354–362 (2006)
2. Tian, J.: Reversible data embedding using a difference expansion. IEEE Transactions on Circuits and Systems for Video Technology, 890–896 (August 2003)
3. Kamstra, L., Heijmans, H.J.A.M.: Reversible data embedding into images using wavelet techniques and sorting. IEEE transactions on image processing 14(12), 2082–2090 (2005)

4. Xuan, G., Shi, Y.Q., Yang, C., Zheng, Y., Zou, D., Chai, P.: Lossless data hiding using integer wavelet transform and threshold embedding technique. In: IEEE International Conference on Multimedia and Expo (ICME 2005), Amsterdam, Netherlands, July 6-8 (2005)
5. Xuan, G., Shi, Y.Q., Yao, Q., Ni, Z., Yang, C., Gao, J., Chai, P.: Lossless data hiding using histogram shifting method based on integer wavelets. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, Springer, Heidelberg (2006)
6. Yang, B., Schmucker, M., Funk, W., Busch, C., Sun, S.: Integer DCT-based reversible watermarking for images using companding technique. In: Proceedings of SPIE, Security and Watermarking of Multimedia Content, Electronic Imaging, San Jose, CA, USA (January 2004)
7. Lee, S., Yoo, C.D., Kalker, T.: Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform. IEEE Transactions on Information Forensics and Security 2(3) Part 1 (September 2007)
8. Coltuc, D., Chassery, J.M.: Very fast watermarking by reversible contrast mapping. IEEE Signal Processing Letters 14(4), 255–258 (2007)
9. Coltuc, D.: Improved capacity reversible watermarking. In: International Conference on Image Processing (ICIP), San Antonio, Texas, USA, September 16-19, 2007, pp. III-249–252 (2007)

Multiple Domain Watermarking for Print-Scan and JPEG Resilient Data Hiding

Anu Pramila, Anja Keskinarkaus, and Tapio Seppänen

MediaTeam Oulu,
Department of Electrical and Information Engineering,
University of Oulu,
P.O. Box 4500,
FIN-90014 University of Oulu, Finland
anu.pramila@ee.oulu.fi

Abstract. In this paper, we propose a print-scan resilient watermarking method which takes advantage of multiple watermarking. The method presented here consists of three separate watermarks out of which two are used for inverting the geometrical transformations and the third is the multibit message watermark. A circular template watermark is embedded in magnitudes of the Fourier transform to invert rotation and scale after print-scan process and another template watermark is embedded in spatial domain to invert translations. The message watermark is embedded in wavelet domain and watermark robustness in both approximation coefficient and detail coefficients is tested. Blind, cross-correlation based methods are utilized to extract the watermarks. The obtained success ratios were at least 91% with JPEG and JPEG200 quality factors of 80-100 and scanner resolution of 300dpi. The BER obtained with the previous settings was less than 1.5%.

Keywords: Digital image watermarking, multiple watermarking, inversion, JPEG, JPEG2000.

1 Introduction

The grown possibilities (Internet, P2P) of misuse of digital content has raised an interest in so called watermarking techniques for protecting digital content, in addition to conventional data protection methods such as scrambling or encryption. The basic idea of the watermarking techniques is fairly simple; to embed information in the signal itself in an imperceptible way. Since the interest in protecting IPR (Intellectual Property Rights) is high, the most studied applications of watermarking are proving ownership, authenticating and fingerprinting. However, as the embedded information can be generically anything that can be represented by bits, interest in other kinds of applications that take advantage of the watermarking techniques is rising. Among them especially interesting are the ones where the embedded information is beneficial to the user, relaxing the robustness requirements to the unintentional attacks caused by the noisy channel.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 279–293, 2008.

Here we consider image watermarking and embedding and extraction of multi-bit information, in particular, where the information is expected to be reliably extracted from printed and scanned images, i.e., after print-scan process. The print-scan process expresses various severe attacks on the watermarked content, the most serious ones being the synchronization attacks caused by geometrical transformations.

Some methods to overcome the synchronization problem have been proposed. Kutter [1] proposed a method for embedding periodic watermarks, the localization of which can be detected by applying autocorrelation function. Deguillaume et al. [2] utilize a fairly similar approach where the geometrical transforms are detected by interpreting repeated patterns and the watermark is extracted after inversion of geometrical transformations. The previous methods are often referred as self-synchronizing watermarks in the watermarking literature, because the periodic watermark efficiently carries information of the orientation and scale of the watermark and additionally the actual information payload.

Another ways for solving problems related to geometrical transforms are methods where an additional template watermark has been embedded in the image. Examples of these kinds of approaches include the methods proposed by Pereira and Pun [3], Lee and Kim [4] and Chiu and Tsai [5]. Pereira and Pun embed a frequency domain template by modifying midfrequency coefficients in a controlled way to produce local peaks, which are detectable and can be used for inverting geometrical transforms. Lee and Kim utilize also frequency domain template; however in their approach the template is a circularly arranged random sequence. In [5] a local peak in frequency domain is utilized to synchronize the reading of a message arranged in a circular structure. Apart from self-synchronizing watermarks and additional synchronization templates, some approaches using invariant domains have been proposed: O'Ruanaidh and Pun [6] utilize RST invariant domain named FourierMellin transform and Bas et al. [7] use feature points to bind the watermark into signal content, therefore achieving the robustness to geometrical attacks.

In this paper, we investigated the possibility of taking advantage of properties of different domains in designing a print-scan and additionally JPEG resilient water-marking scheme. The disadvantages and advantages of different domains are generally well known. Spatial domain has the advantage of maintaining location information, Fourier domain magnitudes are efficiently translation invariant, and DWT has gain much interest as a multiresolution representation to conquer the format conversions, especially JPEG.

In our approach, the rotation and scale transformations are detected and inverted with a frequency domain template, somewhat similar to the one in [4]. Translation is detected with a spatial domain template embedded, utilizing a JND-model as in [8] for adaptation to HVS. Efficiently the template proposed discards the need for full translation search or manual removal of white background before message extraction. The actual information bits are embedded in the wavelet domain with spread spectrum techniques. The purpose of the chosen DWT-method is twofolded, firstly, to examine the possibility to utilize

DWT as a carrier of message payload when resiliency to print-scan process and JPEG/JEPG2000 compression is expected. Secondly, the success ratio/BER of the DWT-domain watermark gives an objective measure of the accuracy of the detection/inversion of geometrical transforms. As, due to the multi domain approach, the interference caused by different watermarks in each other is minor, and imperceptibility criteria can be fulfilled, as shown by the experimental tests.

2 Print-Scan Process

The print-scan process produces various attacks to the watermarked image such as geometrical distortions, pixel distortions, DA/AD transform and noise addition due to the scanner/printer properties and human interaction. Fig. 1 shows the user interface of a scanner where the user defines the scanning area with a dash line quadrilateral. Along the watermarked image, a large portion of the scanner background is also being cropped. The size of the scanned image depends on the user settings and therefore the watermark cannot be assumed to be located in the centre of the scanned image, nor is the watermark perfectly straight and scaled in the scanned image obtained. Therefore the watermark should endure through geometrical transforms and be readable after the userdependent scanning operations.

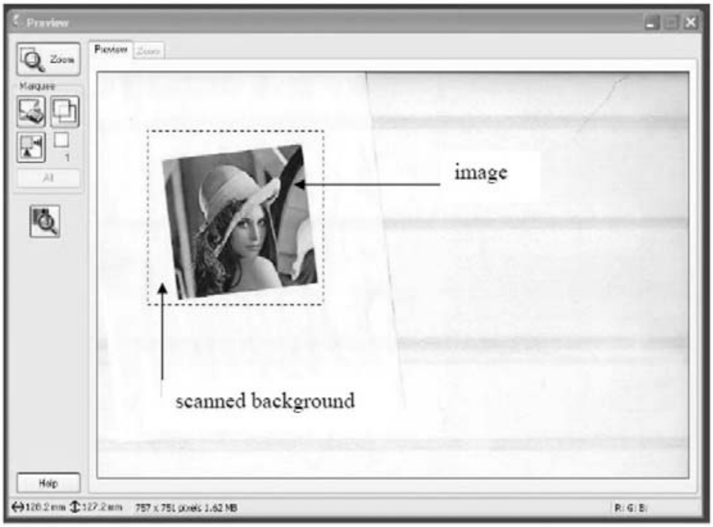


Fig. 1. A scanning area selected by the user

In the print-scan process, the devices used in the experiments must be chosen carefully. The print-scan process is printer/scanner-dependent and time-variant - even for the same printer/scanner [9] [10]. Perry et al. [11] concluded that the printing quality varies between different manufacturers and even between identical models from the same manufacturer.

3 Print-Scan Resilient Watermarking Method

Solanki et al. [10] studied the print-scan process by examining watermarking in Fou-rier transform magnitudes and noted that the low and mid frequency coefficients are preserved much better than the high frequency ones. Inspired by these results, we take advantage of the properties of the Fourier domain magnitudes and embed there a template to recover from rotation and scaling attacks, as was done in [3] by Pereira and Pun and in [4] by Lee and Kim. However, the Fourier domain magnitudes are invariant to shifts in spatial domain and thus another template is required to recover the watermark from translation attack. For this task, a template watermark was designed and embedded in the spatial domain. The message watermark was embedded in the wavelet domain because of its robustness and superior HVS properties.

The watermarks are embedded in the luminance values of the image, and, as can be seen from the block diagram in Fig. 2, the message watermark is embedded last. This order of embedding the watermarks was chosen because every watermark em-bedded can be considered as an attack against previously embedded watermarks. If the message watermark had been embedded first, the template watermarks could have worsened the BER of the message watermark when extracted.

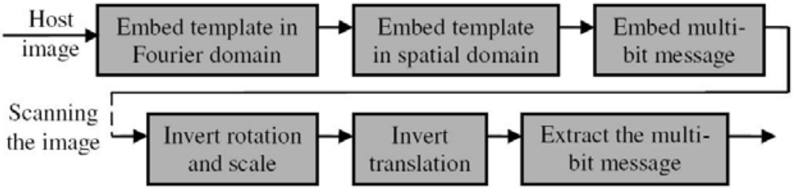

Host image
Embed template in Fourier domain
Embed template in spatial domain
Embed multi-bit message
Scanning the image
Invert rotation and scale
Invert translation
Extract the multi-bit message

Fig. 2. Block diagram of the proposed print-scan robust method

3.1 Inverting Rotation and Scale

To invert rotation and scale, a template watermark is embedded in the magnitudes of the Fourier domain which are invariant to translations in spatial domain. This invari-ance makes it easier to determine the rotation angle and scaling factor of the image because the translation needs not to be considered.

Embedding the Fourier Domain Template. After Fourier transforming the image, the low frequencies are moved to the centre. The template is a pseudorandom sequence consisting of 0's and 1's arranged around the origin symmetrically, as depicted in Fig. 3. The template is embedded in the middle frequencies of the magnitudes of the Fourier domain in a form of a sparse circle in which the 1's of the pseudorandom sequence form peaks and 0's appear as gaps. The strength at which the peaks are embedded varies with local mean and standard deviation, because the embedding strength should clearly be larger close to the low frequencies, where, in general, are the highest values of the Fourier transform. Every point on the circle is embedded in the Fourier domain at an angle pi/20 from each other, where the value pi/20 is chosen for convenience.

By taking into account the results obtained by Solanki et al. [10] and He and Sun [9], we settled down to the middle frequency band of the Fourier domain. Low fre-quencies were discarded because the low frequencies contain most of the energy in the image and thus all the changes made to the low frequencies are highly visible in the image. On the other hand, robustness against JPEG compression is required and consequently the high frequencies were also discarded.

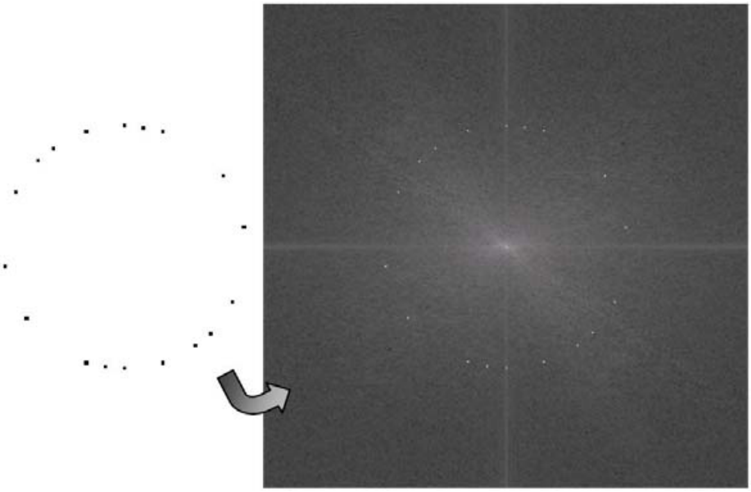


Fig. 3. Block diagram of the proposed print-scan robust method

Extracting the Fourier Domain Template. The extraction of the template watermark is conducted with cross-correlation. First, the image is padded with zeros to its original geometry, a square. This is done in order to prevent the template from being stretched to an ellipse. After zero padding, Fourier transform is applied to the image and the obtained Fourier magnitudes are filtered with a Wiener filter. The result of the filtering is then subtracted from the magnitudes of the scanned image to reduce the effect of the image itself. The Wiener filter minimizes the mean square error between an estimate and the original image and is thus a powerful tool in noise reduction, or noise extraction, as is the case in most of the watermarking techniques.

To reduce noise that does not contain watermark information even more, the Wiener filtered values are thresholded. If a point exceeds certain predefined limit, the point is divided with the local mean to achieve better comparability between points in different locations in the magnitudes of the Fourier transform. If the point does not exceed the limit, the point is replaced with a zero.

To find the circle around the origin, an annulus between two predefined frequen-cies f1 and f2 is chosen, as in the paper by Pereira and Pun [3]. The first frequency f1 is chosen such that the 'noise' of the image in the low frequencies does not disturb the detection process of the circle and the second frequency f2 is selected such that the calculations stay well within image boundaries.

The circle is detected by calculating a cross-covariance value between the embedded pseudorandom sequence and a one dimensional sequence corresponding to a ra-dius between f1 and f2. The cross-covariance value is related to cross-correlation value and can be defined as a cross-correlation of mean removed sequences

$$c^*_{xy}(m) = \begin{cases} \sum_{n=0}^{N-|m|-1} \left(x(n+m) - \frac{1}{N} \sum_{i=0}^{N-1} x_i \right) \left(y^*_i - \frac{1}{N} \sum_{i=0}^{N-1} y^*_i \right) \ , \ m \geq 0 \ , \\ c^*_{yx}(-m) \ , \ m < 0 \end{cases} \tag{1}$$

where x is a sequence of the image at some radius with length N and y is the pseudo-random sequence interpolated to the length N. The maximum of each of the resulting cross-covariances is saved to a vector. After all the integer radii between frequencies f1 and f2 are examined, the maximum is selected from the vector shown in Fig. 4.

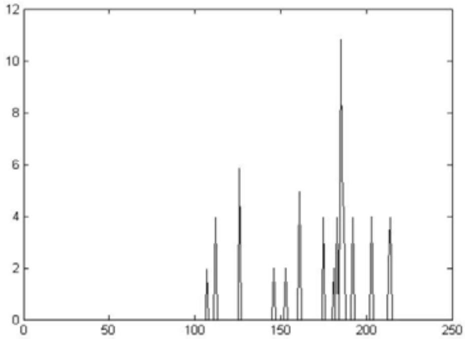


Fig. 4. The vector containing maximums of the cross-correlations

The maximum of the vector is an estimation of the radius of the circle. The exact radius is found by finding the locations of the template peaks by examining the space at wide 2 pixels around the estimation of the radius. The search of the peaks is performed by using adaptive thresholding and the final threshold value is found by iterating the calculations until the correct amount of the peaks is found. The pseudorandom sequence used in embedding is known and thus the number of peaks that should be found is also known. The point is selected to be a peak if the value in that point exceeds the threshold value and if the peak is a maximum on that area. The locations of the peaks are specified further by interpolating a small area around the peak and selecting the maximum value of the interpolation as the peak.

Some of the peaks located this way may be discarded because we know that the peaks should be located at angle pi/20 from each other. The points at pi/20 from each other are selected as template peaks and the other are discarded. The obtained piece of sequence is then cross-correlated with the embedded pseudo-random se-quence and the maximum of the cross-correlation signal shows the amount of rotation in a multiple of pi/20.

From here on the scaling factor and rotation angle are straightforward to determine. The scaling factor is calculated by taking a trimmed mean of the radii of the peaks and dividing this value by the original radius of the embedded pseudorandom sequence. The rotation angle is obtained by subtracting the original angles of the peaks from the angles of the detected peaks and taking a median from the resulting values. Fig. 5 shows the watermarked image after print-scan process and after the corrections have been made.

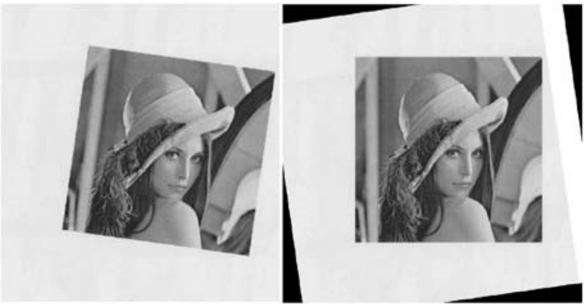


Fig. 5. a) Watermarked image after print-scan process. b) Watermarked image after correction of rotation and scale.

3.2 Inverting Translation

The magnitude domain of the Fourier transform is invariant to shifts in spatial do-main and thus, if we want to apply watermarking techniques that are robust against translation attack, we need to find the amount of translation with a different way. Here, the problem is solved by introducing a second template watermark which is embedded in spatial domain and which is capable of finding the amount of translation after rotation and scale have been found. Therefore full search is not needed and the translation is found efficiently. The method for finding the translation was developed in [12], where a template watermark is embedded in the spatial domain and extracted after print-scan process with cross-correlation tech-niques. The shape of the template watermark is illustrated in Fig. 6, in which it can be seen that the template consists of two parts, horizontal and vertical segments. Both of the segments are built with a pseudorandom sequence of size 127 which is repeated over the image to form the template pattern. The embedding strength of the watermark was chosen based on the JND (Just Noticeable Difference) model, proposed by Chou and Li [8], in which a separate JND threshold for each pixel in the image was calculated.

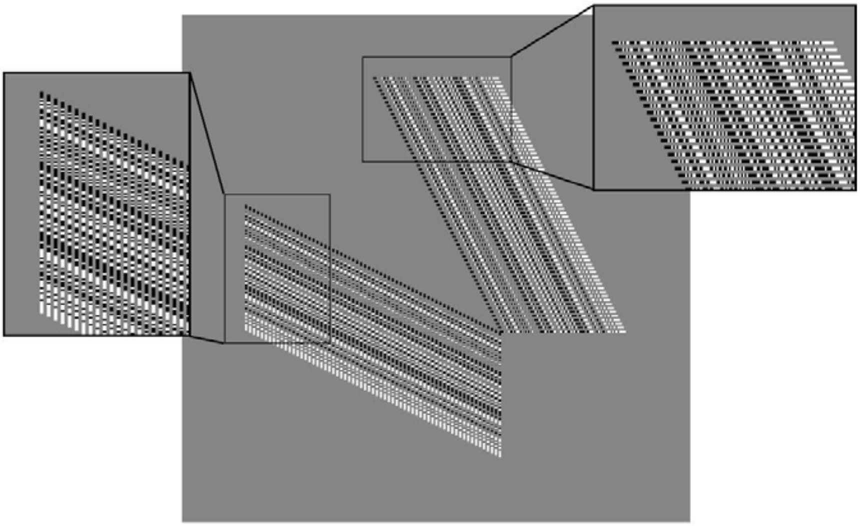


Fig. 6. The template embedded in the image in order to recover the message watermark from a translation attack

The extraction problem is illustrated in Fig. 7, where four unknowns determine the amount of translation in four directions. The extraction of the watermark information is performed first for horizontal direction and then for the vertical direction. Before the extraction process the image is interpolated with quarter pixel interpolation to achieve better precision and Wiener filtering to remove noise, i.e. the effect of the image.

The location of the template is known in the original image and the information is utilized in determining the translation. The translation template is found by calculating cross-correlations with every other line in the image and the pseudorandom sequence used in embedding. There is no need to calculate cross-correlations with every line, because of the design of the watermark and this saves time and processing power, but does not affect robustness significantly.

Due to the shape of the template watermark, the cross-correlation peaks obtained are not located in the same location, and in order to place the peaks to the same posi-tion relative to each other, each of the cross-correlation results should be shifted. After the cross-correlation sequences have been shifted into the same position, the sequences are then summed up and consequently the peaks are strengthened.

The location of the maximum of the cross-correlation sequence contains the information about the translation. The cross-correlations are calculated in two directions which results in two cross-correlation sequences. However, the locations of the two maximums do not tell the amount of translation straightforwardly because of all the shifts and additions. The translation is found by first subtracting the known location of the template in the original image from the two locations of the peaks and then combing the two values in order to find

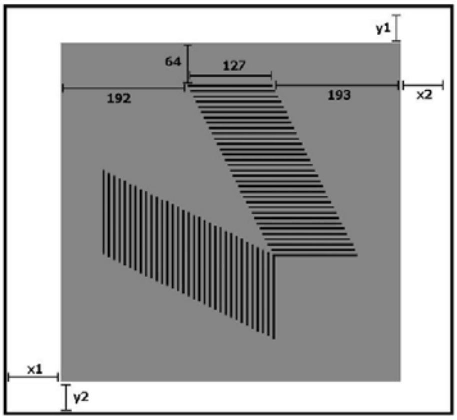


Fig. 7. The translation template after spatial shift

the exact values for the translation. There are, however, four unknown parameters that determine the location of the image, one for each side, as shown in Fig. 7, and therefore the location of the maximum peak is calculated from both directions of the cross-correlation sequences. This results in four values which can be combined with the following equations to find out the exact values of the translation:

$$\begin{aligned} val1 &= x_2 + \tfrac{1}{2}y_1 \\ val2 &= \tfrac{1}{2}x_1 + y_2 \\ val3 &= x_1 + \tfrac{1}{2}y_2 \\ val4 &= \tfrac{1}{2}x_2 + y_1 \end{aligned} \tag{2}$$

where the val1, val2, val3 and val4 are the four values calculated from cross-correlation peaks. After the translation parameters are calculated, the image can be extracted from the background and the actual value-adding watermark can be read.

An algorithm of the extraction method is as follows:

1. Apply quarter-pixel interpolation to the image;
2. Process horizontal part of the template;
2.1. Calculate cross-correlation with every other row of the image;
2.2. Shift every cross-correlation result with one more that the result of the previ-ous row so that the peaks are in the same line;
2.3. Add all the results together;
2.4. Find the maximum peak and calculate the distance from both ends of the sequence;
2.5. Remove the location information of the template in the original image;
3. Process vertical part of the template;
4. Solve the amount of translation from the received results.

3.3 Embedding and Extracting the Multibit Message

In our method, we have applied the method by Keskinarkaus et al. [13] for embedding and extracting the message watermark. In their method, the watermark is em-bedded in the approximation coefficients of the Haar wavelet transform to gain better robustness. Detail coefficients of the wavelet transform, on the other hand, might offer better imperceptibility properties [14]. In our method, the robustness of both of the coefficient domains have been experimented. The watermark is embedded with

$$\begin{cases} Y_{l,f}^{**}(n) = Y_{l,f}^{*}(n) + \beta \cdot m(k), \ messagebit = 1 \\ Y_{l,f}^{**}(n) = Y_{l,f}^{*}(n) - \beta \cdot m(k), \ messagebit = 0 \end{cases} \tag{3}$$

where Y* l,f is an image which has already been watermarked with the templates in Fourier and spatial domain. Y*l,f(n) is the sub-band of Y* in the lth resolution level and fth frequency orientation. Y**l,f(n) is a new watermarked sub-band, where ** means that multiple watermarking has been applied. b is a scaling coefficient to control the embedding strength and m(k) is the m-sequence the length of which controls the chip rate for spreading.

The message watermark is extracted with cross-correlation after the geometrical distortions have been corrected. The cross-correlation is calculated with the m-sequence used for embedding the message and a small segment of wavelet coefficients of the same size as the m-sequence. The results of the cross-correlations are analyzed: if the correlation value is above certain value the message bit is chosen to be 1 and otherwise 0.

4 Experiments and Results

The method was tested with a 512x512 Lena image by embedding an error coded message of size 135 bits to the image. The message size was chosen large enough for the application but small enough to enhance robustness. The error-correction coding applied to the message was (15,7) BCH coding which is capable of correcting 2 bits. The Lena image was watermarked and tested both embedding the message in the detail coefficients and embedding the message in the approximation coefficients of the wavelet domain. This results in two message watermark embedding methods which are examined in the following section.

The watermarked images were compressed with JPEG and JPEG2000 algorithms with Matlab quality factors of 100, 80, 60 and 40. The quality factors correspond approximately to compression ratios 3.3, 16.2, 26.0 and 36.1, respectively. The qualities of the images were examined by calculating PSNR (Peak Signal to Noise Ratio) and PSPNR (Peak Signal to Perceptible Noise Ratio) values [8] for each image. The PSPNR value takes into account only the part of the distortion that exceeds the JND threshold calculated earlier. Thus the PSPNR value gives a better description of the quality of the image. The obtained values were gathered to Table 1, and as seen, the qualities of the images stayed fine through the embedding process.

Table 1. PSNR and PSPNR values for the images where the message watermark is embedded in different domains

JPEG	PSNR	PSPNR
approximation coefficients	40.5	59.3
detail coefficients	38.4	53.3
JPEG2000		
approximation coefficients	38.0	51.0
detail coefficients	37.6	50.6

The watermarked and compressed images were printed with Hewlet Packard ColorLaserJet 5500 DTN printer. One JPEG compressed image was also printed out with Hewlet Packard ColorLaserJet 4500 DN printer, and it was noted that the result was significantly darker and fuzzier than the corresponding image printed with ColorLaserJet 5500 DTN printer, as shown in Fig. 8.

The scanner used in the experiments was Epson GT-15000, and every image was scanned 100 times with 300dpi and then 100 times with 150dpi and saved to uncompressed tiff-format. The printed and watermarked images were scanned by rotating the image manually on the scanner plate between degrees -45 and 45 and the scanning area was chosen around the image by hand, as illustrated in Fig. 1. The embedded watermarks were extracted from each of the images and the results obtained were collected to following tables. The reliability of the method is shown with success ratios that is, percentage of times when the message was extracted correctly and BER (Bit Error Rate) information.

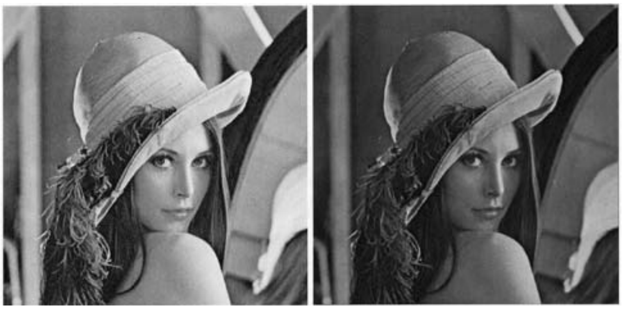


Fig. 8. The translation template after spatial shift

The properties of the wavelet transform complicate the comparison of the two embedding domains as the values in detail and approximation coefficients represent different things. Thus to make the methods comparable, the embedding strengths were chosen for each based on the similar PSNR values. The results obtained after the print-scan process are quite good, as seen from the Tables 2 and 3, and the method is robust against heavy JPEG compression and JPEG2000 compression. It seems that the method where the watermark was embedded in the approximation coefficients of the wavelet transform is more robust against JPEG2000 compression than the method where the watermark was embedded in the detail coefficients but lose in comparison with robustness against JPEG compression. The method in which the watermark was embedded in the detail coefficients shows very strong robustness against JPEG compression, as seen from Tables 2 and 3.

Table 2. Success ratio with different JPEG quality factors and scanning settings

	approx. coefficients		detail coefficients	
Quality factor	300dpi	150dpi	300dpi	150dpi
100	98%	98%	100%	99%
80	97%	94%	100%	99%
60	97%	91%	98%	98%
40	49%	46%	90%	85%

Table 3. Success ratio with different JPEG2000 quality factors and scanning settings

	approx. coefficients		detail coefficients	
Quality factor	300dpi	150dpi	300dpi	150dpi
100	96%	100%	98%	99%
80	100%	99%	91%	91%
60	84%	79%	23%	12%
40	0%	4%	0%	0%

The BER values of the tests are shown in Tables 4 and 5 where the value in the brackets indicates the BER after error correction. The BER values show that a stronger error correction coding might improve the results significantly. All the BER values are beneath 50%, which indicates that the watermark is not destroyed entirely under heavy JPEG and JPEG2000 attacks.

Solanki [10] and He and Sun [9] showed in their papers that the printer should also be considered when designing a print-scan robust watermarking system. The results obtained with different printers are collected to Tables 6 and 7. The pictures were printed with HP LaserJet 5500 DTN and HP LaserJet 4500 DN printers. Both printed images were compressed with JPEG quality factor of 100 and scanned with the same scanner.

The tables show the importance of finding the template peaks correctly from the magnitudes of the Fourier domain. When the interpolation is not used while determining the exact locations of the template peaks, the amount of scale and rotation are not found accurately enough. This clearly affects the performance of the algorithm. From the tables, it can be seen that the interpolation does not strongly affect watermark extraction when a printer with a good quality is used but destroys the watermark when a different printer, possibly with lower quality, is used.

Table 4. BER with different JPEG quality factors and scanning settings

	approx. coefficients		detail coefficients	
Quality factor	300dpi	150dpi	300dpi	150dpi
100	1.9% (0.56%)	1.4% (0.1%)	0.8% (0%)	1.2% (0.5%)
80	2.6% (1.0%)	2.9% (1.0%)	0.7% (0%)	1.0% (0.5%)
60	3.0% (0.6%)	5.0% (2.8%)	1.8% (0.6%)	1.8% (1.0%)
40	8.3% (4.3%)	8.7% (4.0%)	6.2% (3.6%)	6.8% (4.2%)

Table 5. BER with different JPEG2000 quality factors and scanning settings

	approx. coefficients		detail coefficients	
Quality factor	300dpi	150dpi	300dpi	150dpi
100	2.5% (1.1%)	1.3% (0%)	1.3% (0.6%)	0.9% (0.3%)
80	1.3% (0%)	1.7% (0.5%)	3.6% (1.5%)	2.4% (0.8%)
60	8.7% (4.9%)	9.2% (5.7%)	20.2%(12.5%)	21.8%(15.3%)
40	26.8%(24.1%)	28.6%(26.0%)	41.4%(38.6%)	42.2%(39.6%)

Table 6. Success ratio with different printers and JPEG quality factor of 100

	without interpolation		with interpolation	
Printer	300dpi	150dpi	300dpi	150dpi
5500 DTN	96%	96%	100%	99%
4500 DN	86%	80%	96%	99%

Table 7. BER with different printers and JPEG quality factor of 100

	without interpolation		with interpolation	
Printer	300dpi	150dpi	300dpi	150dpi
5500 DTN	3.0% (5.3%)	3.2% (2.0%)	0.8% (0%)	1.2% (0.5%)
4500 DN	8.6% (5.3%)	14.5% (11.0%)	4.1% (3.9%)	2.7% (0.1%)

Comparison of the method with other print-scan robust watermarking techniques is difficult, because print-scan robustness has not been extensively researched and the watermarking algorithms depend heavily on the application and properties required. Many of the print-scan robust methods focus on detecting watermarks [7] [15] and multibit message watermarking methods are rare. In nearly every paper, the robustness has been tested with a different way or the testing process has not been explained properly, thus preventing others to repeat the experiments. In some papers, there is no information about the strength of the distortions inflicted [5] [9] and it is difficult to compare the methods. For comparison it is important to know if the image has been scanned in random alignment or by placing the image on the scanner plate straight. In this method, the alignment of the image does not matter and the robustness against large angles of rotation and JPEG compression is high. All the distortions are inflicted to the image simultaneously thus simulating the real world situation. He and Sun [9] exchanged robustness with capacity and the results were thus significantly worse than in the proposed method. They reported BER values of 15% while in the proposed method the BER values were mostly below 5% with worse scanning resolutions than He and Sun applied in their method. PSNR values were approximately the same in both of the methods.

5 Conclusion

The experiments show that the method proposed is robust against rotation, scaling and translation attacks, as well as JPEG and JPEG2000 compressions, and is thus robust against a print-scan attack. In this system, multiple watermarking was used successfully where each of the watermarks was embedded in different domain and each had a purpose of its own. The quality of the watermarked image stayed fine through the embedding process in spite of multiple watermarks embedded, and the template watermarks were able to correct geometrical distortions. The method also worked well under JPEG and print-scan attacks. It is shown that even the ordinary methods such as spread spectrum methods can survive through print-scan process and no special equipment or method is required.

It was found out that the message watermark was more robust against JPEG com-pression when it was embedded in detail coefficients than the approximation coefficients of the wavelet domain. On the other hand, when JPEG2000 compression was used, the watermark was more robust when it was embedded in approximation coefficients that detail coefficient of the wavelet domain. It was also discovered that, although the selection of the printer is important in de-signing a print-scan robust watermarking system, some of the effects of the printer can be minimized by improving the watermark extraction process. In the future, a better message watermarking method should be developed, and ro-bustness against JPEG2000 compression algorithms, in particular, should be re-searched more extensively. It can, however, be concluded that the multiple watermarking method, proposed here, can correct rotation, scale and translation accurately for the message watermark extraction.

References

1. Kutter, M.: Watermarking resisting to translation, rotation and scaling. In: Proc. of SPIE Multimedia Systems and Applications, vol. 3528, pp. 423–431 (1998)
2. Deguillaume, F., Voloshynovskiy, S., Pun, T.: Method for the Estimation and Recovering from General Affine Transforms. In: Proc. of SPIE, Electronic Imaging, 2002, Security and Watermarking of Multimedia Contents IV, vol. 4675, pp. 313–322 (2002)
3. Pereira, S., Pun, T.: Robust Template Matching for Affine Resistant Image Watermarks. IEEE Trans. on Image processing 9(6), 1123–1129 (2000)

4. Lee, J.-S., Kim, W.-Y.: A Robust Image Watermarking Scheme to Geometrical Attacks for Embedment of Multibit Information. In: Proc. of Advances in Multimedia Information Processing, 5th Pacific Rim Conference on Multimedia, vol. 355, pp. III-348–355 (2004)
5. Chiu, Y.-C., Tsai, W.-H.: Copyright Protection against Print-and-Scan Operations by Watermarking for Colour Images Using Coding and Synchronization of Peak Locations in Frequency Domain. WCVGIP 2006, Taiwan, Journal of information science and engineering 22, 483–496 (2006)
6. O'Ruanaidh, J.J.K., Pun, T.: Rotation, scale and translation invariant digital image watermarking. IEEE Proc. of ICIP 1997 1, 536–539 (1997)
7. Bas, P., Chassery, J.-M., Marq, B.: Geometrically Invariant Watermarking Using Feature Points. IEEE Trans. on Image Processing 11, 1014–1028 (2002)
8. Chou, C.-H., Li, Y.-C.: A Perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile. IEEE Trans. on Circuits and Systems for Video Technology 5(6), 467–476 (1995)
9. He, D., Sun, Q.: A Practical Print-scan Resilient Watermarking Scheme. IEEE Proc. of ICIP 2005 1, 257–260 (2005)
10. Solanki, K., Madhow, U., Manjunath, B.S., Chandrasekaran, S.: Estimating and Undoing Rotation for Print-scan Resilient Data Hiding. In: IEEE International Conference on Image Processing, vol. 1, pp. 39–42 (2004)
11. Perry, B., MacIntosh, B., Cushman, D.: Digimarc MediaBridge -The birth of a consumer product, from concept to commercial application. In: Proc. of SPIE Security and Watermarking of Multimedia Contents IV, vol. 4675, pp. 118–123 (2002)
12. Pramila, A.: Watermark synchronization in camera phones and scanning devices. Master's thesis, Department of electrical information engineering, University of Oulu, Oulu, p.83
13. Keskinarkaus, A., Pramila, A., Seppänen, T., Sauvola, J.: A Wavelet Domain Print-scan and JPEG Resilient Data Hiding Method. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 82–95. Springer, Heidelberg (2006)
14. Barni, M., Bartolini, F., Capellini, V., Lippi, A., Piva, A.: A DWT-based technique for spatio-frequency masking of digital signatures. In: Proc. of the SPIE/IS&T International Conference on Security and Watermarking of Multimedia Contents, vol. 3657, pp. 31–39 (1999)
15. Lin, C.-Y., Chang, S.-F.: Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process. In: ISMIP 1999 (1999)

GA-Based Robustness Evaluation Method for Digital Image Watermarking

G. Boato, V. Conotter, and F.G.B. De Natale

Department of Information and Communication Technology, University of Trento
Via Sommarive 14, I-38050, Trento, Italy
Tel.: +39 0461 883193; Fax: +39 0461 882093

Abstract. The present paper proposes a new flexible and effective evaluation tool based on genetic algorithms to test the robustness of digital image watermarking techniques. Given a set of possible attacks, the method finds the best possible un-watermarked image in terms of Weighted Peak Signal to Noise Ratio (WPSNR). In fact, it implements a stochastic search of the optimal parameters to be assigned to each processing operation in order to find the combined attack that removes the watermark while producing the smallest possible degradation of the image in terms of human perception. As a result, the proposed method makes it possible to assess the overall performance of a watermarking scheme, and to have an immediate feedback on its robustness to different attacks. A set of tests is presented, referring to the application of the tool to two known watermarking approaches.

1 Introduction

Digital watermarking techniques have raised a great deal of interest in the scientific community after the pioneering contribution by Cox et al. [1] (see for instance the books [2], [3], [4], and the references therein). The practice of imperceptible alteration of a document to embed a message into it plays a key role in the challenging field of copyright and copy protection and motivates the search for more efficient solutions. As widely known in cryptography, by identifying algorithms failings benchmarking tools speed up development and improvement of watermarking techniques and research in this field is still on going [4].

The mark embedded into the cover work is usually required to be robust against manipulations of the host image (except for particular techniques like fragile watermarking), including a great variety of digital and analog processing operations: lossy compression, linear and non-linear filtering, scaling, noise addition, etc. As underlined in [4], a theoretical analysis and comparison of the performance of different watermarking algorithms is possible only on a limited number of simple cases (see for instance Chapter 7 in [4]) and requires complicate feature and attack modeling. Therefore an extensive experimental analysis is necessary.

As far as existing watermark benchmarking systems are concerned, they provide efficient and complex evaluation on a wide range of single attacks. In StirMark [5] the to-be-tested algorithm is repeatedly used to mark a set of reference features, then the host image is attacked by means of many processing operators. If the system is not able

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 294–307, 2008.

to recover the embedded mark an error message is given. Increasing attack strength is applied to the host image in order to see whether the watermark survives them or not and evaluation is done using the standard Peak Signal to Noise Ratio (PSNR) metric. Finally, StirMark assigns them an average score for each class of attacks. Improving StirMark, the so-called second generation of watermark benchmark [6], [7] provides the introduction of new types of attacks, the use of a perceptual metric to measure the introduced degradation, the possibility to distinguish between watermark detection and decoding, and finally application driven evaluation [8], [9].

The scope of the present work is to provide a flexible and effective benchmarking tool, exploiting Genetic Algorithms (GA) to test the robustness of watermarking techniques under a combination of given attacks and to find the best possible un-watermarked image. The degradation evaluation is done taken into account both the classical PSNR metric and the perceptual metric Weighted Peak Signal to Noise Ratio (WPSNR). Optimization is performed on WPSNR.

GA are characterized by versatility and ability to optimize in complex multimodal search spaces [10]. They have been successfully applied for a wide range of problems characterized by a large number of unknown parameters and highly non-linear behavior [10]. The major advantages of the GA with respect to the other optimization algorithms, such as gradient conjugate-based methods, are mainly related to their independence from the initialization and their ability to prevent local minima. Moreover it is well known from the scientific literature that it is possible to enhance the convergence ratio making a good choice of the algorithm parameters (for details see [10] and the more recent [11]).

In the field of watermarking, GA are mainly used in the embedding procedure. In [12] Maity et al. attempts to use GA for finding out values of parameters, both with linear and non linear transformation functions, for achieving the optimal data imperceptibility for the watermark. A watermarking method based on DCT and GA is proposed in [13]: the watermark is embedded with visually recognizable patterns into the image by selectively modifying the middle-frequency part of the image, and the GA is applied to search for locations to embed the watermark in the DCT coefficients block such that the quality of the watermarked image is optimized. In [14] Shieh et al. present a watermarking optimization similar to [13]: they make use of GA to find the optimum frequency bands for watermark embedding into a DCT-based watermarking system, which can improve both security and robustness, and also image quality of the watermarked image. A genetic watermarking approach based on neural network is presented in [15]: a neural network to classify DCT blocks of images in training sets and for each cluster GA is then performed to find out the optimal coefficients for watermark embedding, while a new approach for optimization in wavelet-based image watermarking using GA is described in [16]: the watermark insertion and watermark extraction are based on the code division multiple access techniques and the GA is used to search for optimal strength of the watermark in order to improve both quality of watermarked image and robustness of the watermark. In [17], the authors propose a spread spectrum image watermarking algorithm using the discrete multiwavelet transform and performance improvement is obtained by GA optimization: GA allow the search for threshold value and embedding strength to improve the visual quality of watermarked images and the watermark robustness for images with different characteristics. Recently, a new concept of developing a robust steganographic system has been introduced in [18], by artificially counterfeiting statistic features instead of exploiting the traditional strategy to avoid changes of statistic features. GA-based methodology is applied by adjusting gray values of a cover-image while creating the desired statistic features to generate the stego-images that can break the inspection of steganalytic systems.

In the present work we use GA in the detection phase in order to remove the watermark. The optimization process searches the optimal parameters associated to the attacks which allow to remove the embedded watermark preserving the maximum perceptual quality of the image. Such benchmark allows to underline merits and drawbacks of the analyzed algorithm in terms of robustness. Once the set of attacks is defined, considering in the first instance the application scenario, it is possible to evaluate the performance of the tested technique under any combination of processing operations, finding the combined attack that removes the watermark while ensuring maximum WPSNR. The major difference with the existing benchmarking approaches consists of a stochastic search of optimal parameters to be assigned to each attack in order to recover the un-watermarked image perceively closest to the watermarked one.

The structure of the paper is the following: in Section 2 we give a detailed description of the proposed evaluation method and in Section 3 we set up the evaluation of two known watermarking techniques reporting the corresponding experimental results in Section 4. Finally in Section 5 we draw some concluding remarks.

2 GA-Based Robustness Evaluation Method

In order to design a watermarking benchmark, first of all we need to define the watermark properties to be measured. Our aim is to evaluate the robustness of the methods and, giving a pattern of possible attacks, to find the best combination which removes the mark given rise to the smaller degradation perceived by the Human Visual System (HVS). For this reason, the degradation of the image is measured through the classical Peak Signal to Noise Ratio (PSNR) metric but also through the Weighted Peak Signal to Noise Ratio (WPSNR), which overcomes the drawback of classical metrics taking into account how modifications are perceived by HVS. Indeed, to evaluate the artifacts several metrics can be used. The most used one is the PSNR metric though several tests show that such metric is not always valid to measure the quality perceived by HVS [19]. Two differently distorted images having the same PSNR with respect to the original image, may often give significantly different visual impact.

A modified version of PSNR, the so-called Weighted PSNR, is introduced in [20]: it takes into account that HVS is less sensitive to changes in highly textured areas and hence introduces an additional parameter called the Noise Visibility Function (NVF) which is an arbitrary texture masking function

$$\mathrm{WPSNR(dB)} = 10 \log_{10} \frac{I_{peak}^2}{\mathrm{MSE} \times \mathrm{NVF}^2} \tag{1}$$

where I_{peak} is the peak value of the input image. The NVF can be modeled as a Gaussian to estimate the local amount of texture in the image. The value of NVF ranges from approximately zero, for extremely textured areas, and up to one, for clear smooth areas of an image

$$\mathrm{NVF} = norm\left\{\frac{1}{1+\delta^2_{block}}\right\} \in [0;1] \tag{2}$$

where $norm$ is a normalization function and δ^2_{block} is the luminance variance of the 8×8 block. The NVF is inversely proportional to the local image energy defined by the local variance and identifies textured and edge areas where the modification are less visible. Therefore, for images with no high texture areas WPSNR is almost equivalent to PSNR.

Given the function WPSNR that we want to maximize, if we want to consider combination of attacks, and avoid a brute force computation in order to find the best solution, a suitable optimization technique is needed. Genetic Algorithms (GA) can be used to achieve optimal solution in this multidimensional nonlinear problem of conflicting nature. GA are robust, stochastic search methods modeled on the principles of natural selection and evolution [10]. GA differ from conventional optimization techniques in that: i) they operate on a group (population) of trial solutions (individuals) in parallel: a positive number (fitness) is assigned to each individual representing a measure of goodness; ii) they normally operate on a coding of the function parameters (chromosome) rather than on the parameter themselves; iii) they use stochastic operators (selection, crossover, and mutation) to explore the solution domain. The metric is regarded as the fitness of the GA. A set of individuals is encoded with chromosome-like bit strings. The cardinality of the set of individuals is called population size [10]. At each iteration, called generation, the genetic operators of crossover and mutation are applied to selected chromosomes with probability P_{cross} and P_{mut}, respectively, in order to generate new solutions belonging to the search space. The optimization process terminates when a desired termination criterion is satisfied, for example, the maximum number of generations is reached, or the fitness value is below a threshold.

The efficiency of a GA is greatly dependent on its tuning parameters, such as the population size, the cross-over and mutation probabilities. These must be set empirically, depending on the configuration of the problem, in order to fine tune the performance of the GA. The computational complexity of GA is heavily affected by the population size; in fact, the larger the population size the more quickly the solution space can be explored, that means that the chance to quickly discover the optimum solution increases. But the population size is directly proportional to the time taken by the algorithm (calculate the fitness value for each chromosome). Viceversa, the smaller the populations the more quickly it can converge to optimal solution once it is found. Also the mutation and cross-over rate affects the efficiency of the algorithm. The lower the crossover probability the lower will be the local search near the best individuals (i.e. it will tend to less often interpolate between good individuals to try to find a solution). A high mutation probability can cause random excursions into possibly unexplored regions and, as a consequence, a degradation to random search. A too low mutation probability can cause the GA to lock in on a local minima and never find the global one (at least in a reasonable amount of time).

In the proposed robustness evaluation technique GA are applied to digital watermarking schemes (see Fig. 1): an image, in which a watermark as been previously embedded

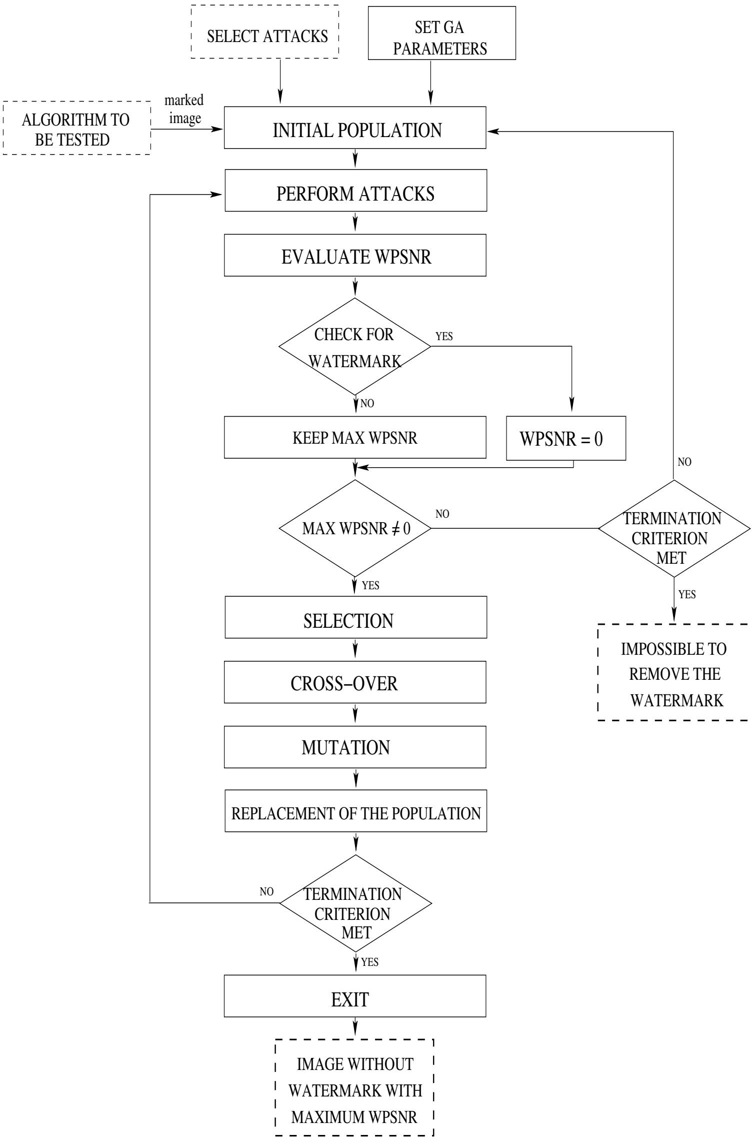


Fig. 1. Block diagram of implemented algorithm

according to the algorithm that is going to be tested, is attacked with different combinations of selected attacks, in order to remove the watermark. The aim is to find the best combinations of attacks to apply to remove the watermark granting a good perceptual quality of the resulting image. The optimization process is performed by GA and WPSNR is the fitness value we want to maximize.

Step 1. Randomly generate combinations of attacks to be applied to the watermarked image and convert them into chromosomes for initial generation. The population size has to be set at the very beginning; typically it is set to 10 times the number of variables the algorithms has to deal with (length of the chromosome). Such length depends on the number of different degradation we want to perform in the robustness evaluation. In Section 3 we report experimental analysis performed on 2, 3 or 4 attacks and therefore population size is set to 30. In this work, where the population size is not particularly large, we can reach a good efficiency of the GA; in fact, we have small population over a large search space with a consequent fast convergence to optimal solution. How the attack parameters are coded into the chromosome also depends on the selected operations and on the selected range of admitted values (see Section 3 for detail on chosen attacks and parameters ranges).

Step 2. Evaluate the WPSNR of each chromosome in current population which removes the watermark, i.e. which generates an un-watermarked image, and then create a new population by repeating following steps until it is complete: i) select as parents the chromosomes with the higher WPSNR; ii) with probability P_{cross} cross over the parents to form new children (new patterns of attacks); iii) with probability P_{mut} mutate the position in the chromosome. In this work widely used parameters for genetic operators has been selected: a roulette-wheel selection, a single-point crossover with $P_{cross} = 0.7$ and a uniform mutation with different P_{mut} depending on the number of performed attacks. Among all individuals of the population that are able to remove the watermark, the one that provides an image with the maximum WPSNR is going to survive to the next generation. Note that if an individual does not represent an attack able to remove the watermark, its related fitness value is set to zero.

If in *Step 2* none of the individuals of the population solve the problem, another population is re-initialized and the process is repeated until a termination criteria is met (number of generation exceeded). In this case the result of the test is that the analyzed watermarking scheme is robust to the selected attacks and hence it is not possible to remove the watermark.

Step 3. New iteration with the already generated population. This new population provides the parameters of new attacks, their corresponding fitness values are evaluated and at every generation the best individual with the highest fitness value is kept.

Step 4. The process ends when a given number of generations is exceeded (termination criteria). At that point the best combination of attacks to apply to remove the watermark from the image has been discovered. In this way the lacks of the tested algorithm are stressed out.

3 Evaluation Setup

In this section we set up the robustness analysis of two different watermarking algorithms. The first one is the perceptual-based symmetric scheme presented by Barni et. al. in [21], while the second one is an asymmetric watermarking scheme proposed in [22] and afterwards improved by the authors in [23]. First, a brief description of the two methods is provided, in order to specify procedures which are crucial for the presented tool operation, and then the selected attacks involved in the robustness evaluation are described.

Algorithm A. The watermarking method [21] works on wavelet domain and exploits perceptual masking in order to embed the mark improving invisibility and robustness. The mark w is inserted into the DWT coefficients of the three largest detail subbands and its strength is adjusted pixelwise. Detection is accomplished by the decision function

$$\delta_A(\phi_e) = \begin{cases} 1 & \textit{if } \rho(\phi_e, w) \geq T_\rho \\ 0 & \textit{otherwise} \end{cases} \tag{3}$$

where T_ρ is the threshold and ρ is the correlation between the extracted DWT coefficients ϕ_e and the watermark.

Algorithm B. In order to evaluate the performance of the proposed tool, a further algorithm is selected in order to be tested. The chosen [23] represents a different class of watermarking techniques both because it is asymmetric and it does not take into account the characteristics of the HVS. Concerning the algorithm implementation, we fix an integer $n = 200$, as feature space $\mathcal{X}$ the space corresponding to the 32×32 entries in the top left corner of the DCT and decompose it into two orthogonal subspaces $\mathcal{W}$ of dimension $2n$ and $\mathcal{V}$. Next, we split $\mathcal{W}$ into two orthogonal subspaces $\mathcal{G}$ and $\mathcal{H}$ of dimension n by randomly selecting half entries of the upper-left 20×20 DCT submatrix and $\mathcal{H}$ by taking the remaining ones. Finally, we pick an arbitrary watermarking sequence $w \in \mathbb{R}^n$ and we apply the procedure described in [23] with just a modification. The scheme makes use of the secret key (G, H, A, B) in the embedding procedure while the detector needs only the public key $(D, s + w)$. Matrix A is required in the definition of D and, although the existence of matrix A is ensured by the trivial choice A equal to the identity matrix, we use here the definition proposed in [24] (with N as in [23], $K = 10^3$ and $k = 199$) to obtain higher detection performances. The watermark detection is accomplished by the decision function [23]

$$\delta_B(\phi_e) = \begin{cases} 1 & \textit{if } |\mathrm{sim}(s + w, D\phi_e)| \geq \varepsilon \\ 0 & \textit{otherwise} \end{cases} \tag{4}$$

where ϕ_e is an extracted feature, $0 \leq \varepsilon << 1$ is a suitable threshold and

$$\mathrm{sim}(s + w, D\phi_e) = \begin{cases} \frac{(s+w)^T D\phi_e}{\|s+w\|\|D\phi_e\|} & \textit{if } D\phi_e \neq 0 \\ 0 & \textit{if } D\phi_e = 0 \end{cases} \tag{5}$$

Once the to-be-tested algorithm is designed, the selection of processing operations is needed to apply our GA-based robustness evaluation method, as specified in Fig. 1. Indeed, *Step 1* in Section 2 converts the attack parameters into chromosomes for initial generation. In Fig. 2 a chromosome is shown, where for the $i - th$ attack a certain number of parameters a_{i,n_i} need to be specified. The lenght of the chromosome in bits will be $log_2 n$, where n is the number of possible values each parameter can take.

In this work, and in particular in the following Section 4, four attacks will be taken into account, which of them implies just a single parameter ($n_i = 1 \forall i$):

1. White Gaussian Noise Addition (AWGN), parameterized with the Noise Power ranging from 0 to 36 dB as an integer number;
2. JPEG compression, parameterized with the Quality Factor (QF) ranging from 100% down to 30% as an integer number;

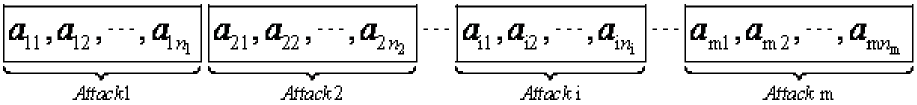

$a_{11}, a_{12}, \cdots, a_{1n_1}$
$a_{21}, a_{22}, \cdots, a_{2n_2}$
$\cdots$
$a_{i1}, a_{i2}, \cdots, a_{in_i}$
$\cdots$
$a_{m1}, a_{m2}, \cdots, a_{mn_m}$
Attack1
Attack 2
Attack i
Attack m

Fig. 2. Chromosome definition

3. 3×3 Gaussian filter, parameterized with its standard deviation (DEV) ranging from 0.1 to 1.9;
4. Resize, parameterized with the resize factor (RES) ranging from 1 down to 0.1.

The choice of taking into consideration only four attack is lead by the fact to have the possibility to run GA over a small population with a pretty fast time of convergence to optimal solution. By increasing the number of attacks and, corresponding, the number of parameters to look for, the search space enlarges. Therefore the population size needs to be increased to better explore it. This means of course that a larger time is taken by the algorithm to compute all the fitness values for each individual. By the way, it is known that GAs are useful and efficient for those problem where the search space is large. So, as the attack parameters increase the GAs are still suitable to be applied to watermarking detection.

4 Experimental Analysis

In order to evaluate the robustness of the two selected algorithm, we run simulations with the tool described in Section 2 on both Lena and Boat images, under various combination of the four attacks described in Section 3.

As far as *Algorithm A* is concerned, tested images are reported in Fig. 3, where the watermark intensity is set as proposed in [21] to meet imperceptibility and but also robustness constraints. In both cases the WPSNR is very high (WPSNR=49.98 dB for Lena-A and WPSNR=48.45 dB for Boat-A). In Table 1 results are reported for the Lena-A image tested under combination of three processing operations: White Gaussian Noise Addition (AWGN), JPEG compression and 3×3 Gaussian filter. The corresponding parameters to be optimized are the noise power (ranging from 0 to 36 dB), the JPEG Quality Factor (QF) (ranging from 100% down to 30%), and the filter standard deviation (DEV) ranging from 0.1 to 1.9, as already described in Section 3. The values of the threshold have been chosen relating to the false alarm probability, as explained in [21]. So we have three thresholds, namely 0.12, 0.15, 0.18, corresponding to P_{fa} equal to $10^{-4}, 10^{-6}, 10^{-8}$ respectively. It is evident that the combination of attacks presented in Table 1 does not impact to the method robustness a lot, since for the chosen threshold no effective patterns of attacks have been found.

Different results are reported in Table 2 for the Lena-A image attacked with AWGN, JPEG compression and Resize, where the resize factor (RES) ranges from 1 down to 0.1. For all values of T_ρ it is possible to remove the watermark; we calculate the WPSNR and the PSNR of the resulting un-watermarked images and the corrisponding ρ values used for watermark detection function (see Equation (3)). This allows to underline a drawback of [21]: such an algorithm suffers in particular under scaling operation,

(a) (b)

Fig. 3. Watermarked images (a) Lena-A (WPSNR=49.98 dB, PSNR=27.47 dB) and (b) Boat-A (WPSNR=48.45 dB, PSNR=25.99 dB)

Table 1. Lena-A attacked with AWGN, JPEG compression and Gaussian filter

T_ρ	Noise power [dB]	QF	DEV	WPSNR [dB]	PSNR [dB]	rho
0.12	impossible to remove the watermark					
0.15	impossible to remove the watermark					
0.18	impossible to remove the watermark					

since this is the only attack to play a key role in the removing process. Similar conclusion can be drawn looking at Tables 3 and 4, which report the results of combination of all four image processing operations on Lena-A and Boat-A, respectively.

These attacks are able to remove the watermark keeping high the WPSNR of the un-watermarked image; scaling attack remains the dominant one in the process for removing the watermark. Furthermore the trend of the fitness (WPSNR) can be seen in Fig. 5 for the performed evaluation corresponding to $T_\rho = 0.12$ (similar trends for all other simulations are not reported here).

As far as *Algorithm B* is concerned, tested images are reported in Fig. 6, where the watermark intensity is scaled in order to meet imperceptibility constraints. Indeed, in both cases (Lena-B and Boat-B) WPSNR=49.95 dB. The Lena-B image is tested under combination of just two processing operations: AWGN and JPEG compression. Results are presented in Table 5 for different values of the threshold ε. The threshold $\varepsilon = 0.06$

Table 2. Lena-A attacked with AWGN, JPEG compression and Resize

T_ρ	Noise power [dB]	QF	RES	WPSNR [dB]	PSNR [dB]	rho
0.12	0	48	0.24	36.96	25.20	0.1132
0.15	0	97	0.25	37.62	25.36	0.146
0.18	0	97	0.26	37.98	25.43	0.1755

Table 3. Lena-A attacked with AWGN, JPEG compression, Gaussian filter and Resize

T_ρ	Noise power [dB]	QF	DEV	RES	WPSNR [dB]	PSNR [dB]	rho
0.12	0	98	0.2	0.23	36.64	25.12	0.1153
0.15	1	46	0.2	0.26	37.91	25.41	0.1492
0.18	0	95	0.2	0.26	37.97	25.43	0.1768

Table 4. Boat-A attacked with AWGN, JPEG compression, Gaussian filter and Resize

T_ρ	Noise power [dB]	QF	DEV	RES	WPSNR [dB]	PSNR [dB]	rho
0.12	0	99	0.2	0.22	32.96	22.38	0.1103
0.15	0	55	0.2	0.24	33.64	22.58	0.146
0.18	0	98	0.2	0.25	34.07	22.69	0.1755

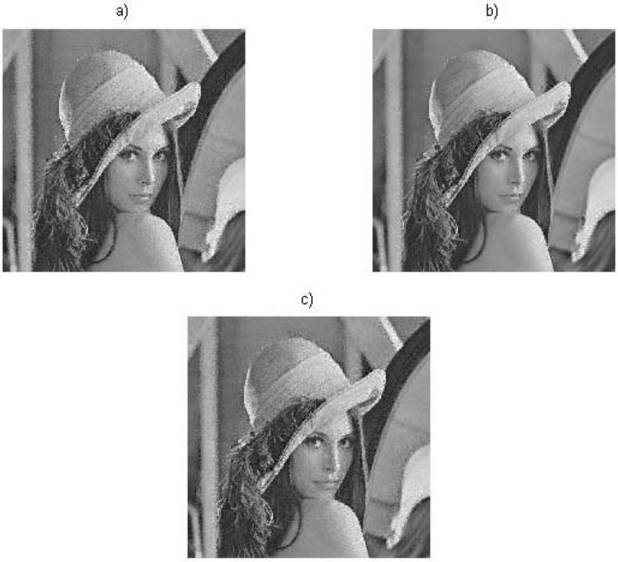


Fig. 4. Un-watermarked Lena-A corresponding to Table 3 a) WPSNR=36.64 dB, b) WPSNR=37.91 dB, c) WPSNR=37.97 dB

has been suggested in [23] and the other values have been increased in order to keep a quality similarity with algorithm A. We calculate the WPSNR and the PSNR of the resulting un-watermarked image (measured with respect to the watermarked one) and its sim value used for watermark detection function (see Equation (4)).

In all cases the benchmarking tool succeeds to remove the watermark and obviously the higher we set the threshold the lower is the perceived quality of the attacked image.

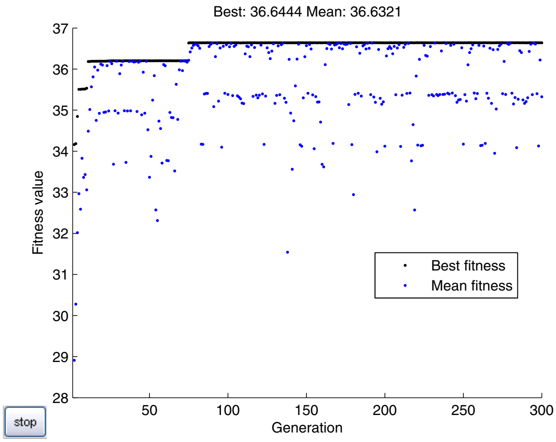
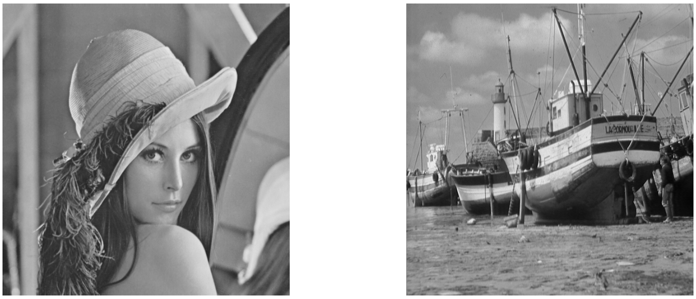


Fig. 5. Fitness trend in case of combination of four attacks for $T_\rho = 0.12$



(a) (b)

Fig. 6. Watermarked images (a) Lena-B (WPSNR=49.95 dB, PSNR=44.07 dB) and (b) Boat-B (WPSNR=49.95 dB, PSNR=44.95 dB)

Table 5. Lena-B attacked with WGN Addition and JPEG compression

ε	Noise Power [dB]	QF	WPSNR [dB]	PSNR [dB]	sim
0.06	34	73	27.51	17.22	0.0537
0.1	31	87	30.27	20.10	0.0873
0.15	27	64	34.14	23.96	0.146

Similar results are reported in Table 6 for the Lena-B image attacked with both AWGN and Gaussian filter. Tables 7 and 8 report the results of combination of four image processing operations on Lena-B and Boat-B, respectively. In this case AWGN, JPEG compression, Gaussian filter and scaling are performed. Notice that the presented values

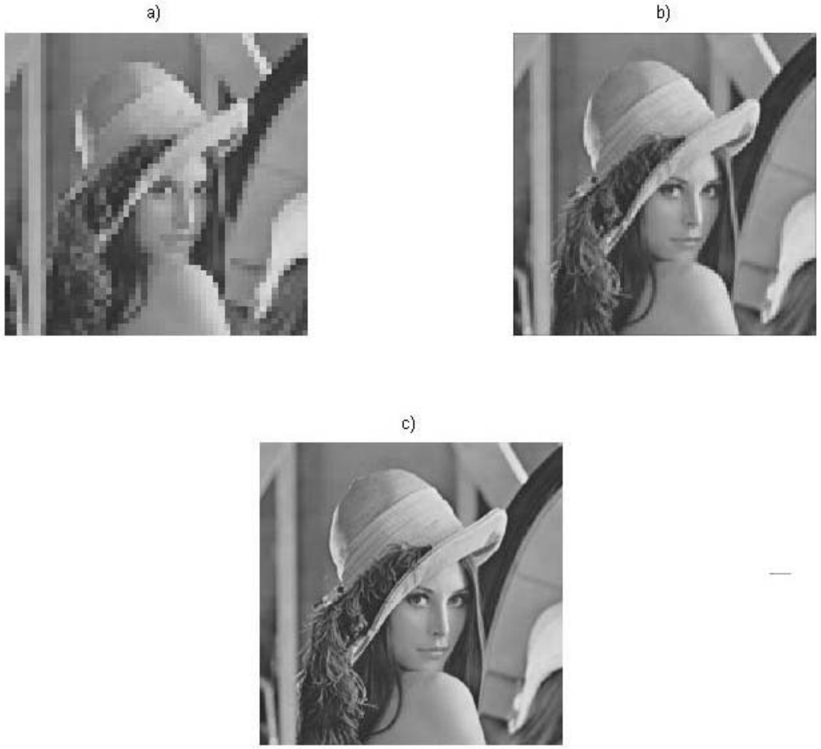


Fig. 7. Un-watermarked Lena-B corresponding to Table 7 a) WPSNR=30.76 dB, b) WP-SNR=38.01 dB, c) WPSNR=46 dB

underline a drawback of [23]: such an algorithm suffers in particular under Gaussian filtering. For $\varepsilon = 0.15$ this filter is able to remove the watermark alone, while for $\varepsilon < 0.15$ other degradations come into play.

Table 6. Lena-B attacked with WGN Addition and Gaussian Filtering

ε	Noise power [dB]	DEV	WPSNR [dB]	PSNR [dB]	sim
0.06	33	1.8	28.7	22.11	0.0576
0.1	29	1.2	32.45	25.52	0.088
0.15	0	1.1	45.99	33.72	0.149

The combination of these attacks removes the watermark keeping high the WPSNR of the un-watermarked image. As reported also in Fig. 7 the perceived quality of the Lena image is good after application of selected processing operators parameterized with optimized values for at least $\varepsilon = 0.15$.

Table 7. Lena-B attacked with AWGN, JPEG compression, Gaussian filter and Resize

ε	Noise power [dB]	QF	DEV	RES	WPSNR [dB]	PSNR [dB]	sim
0.06	1	72	0.2	0.1	30.76	24.63	0.036
0.1	3	34	1.2	0.3	38.01	29.46	0.098
0.15	0	99	1.1	1	46	33.73	0.148

Table 8. Boat-B attacked with AWGN, JPEG compression, Gaussian filter and Resize

ε	Noise power [dB]	QF	DEV	RES	WPSNR [dB]	PSNR [dB]	sim
0.06	2	78	1.2	0.2	31.89	24.15	0.057
0.1	0	40	1.6	1	42.57	29.18	0.098
0.15	0	57	0.7	1	45.71	31.48	0.144

5 Conclusion

This work presents a new tool for robustness evaluation of watermarking techniques. Given a watermarked image, GA are used to find the combination of attacks which remove the mark given rise to the smaller degradation perceived by the HVS in terms of WPSNR. We present the analysis of both a symmetric and an asymmetric watermarking method, namely [21] and [23], and we draw their merits and drawbacks in terms of robustness under selected patterns of attacks. In particular, a GA-based search is run for the estimation of optimal parameters to be assigned to each attack in order to recover the un-watermarked image perceively closest to the watermarked one. Given a fixed false-positive probability, the application of the proposed tool to different methods will allow to have an immediate feedback on their robustness. Hence, future work will extend this tool for algorithms comparison.

References

1. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure Spread Spectrum Watermarking for Multimedia. IEEE Transactions on Image Processing 6, 1673–1687 (1997)
2. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Academic Press, London (2002)
3. Eggers, J., Girod, B.: Informed Watermarking. Kluwer Academic Publishers, Norwell (2002)
4. Barni, M., Bartolini, F.: Watermarking Systems Engineering. Enabling Digital Assets Security and Other Applications. Signal Processing and Communications Series (2004)
5. `http://www.petitcolas.net/fabien/watermarking/stirmark/`
6. Voloshynovskiy, S., Pereira, S., Iquise, V., Pun, T.: Attack Modelling: Towards a Second Generation Watermarking Benchmark. Signal Processing 81, 1177–1214 (2001)
7. Pereira, S., Voloshynovskiy, S., Madueno, M., Marchand-Maillet, S., Pun, T.: Second Generation Benchmarking and Application Oriented Evaluation. In: Proceedings of the International Workshop on Information Hiding, pp. 340–353 (2001)
8. `http://watermarking.unige.ch/Checkmark/`
9. `http://www.certimark.org/`

10. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1999)
11. Tsoy, Y.R.: The Influence of Population Size and Search Time Limit on Genetic Algorithm. In: Proceedings of the 7th Korea-Russia International Symposium on Science and Technology, vol. 3, pp. 181–187 (2003)
12. Maity, S.P., Kundu, M.K., Nandi, P.K.: Genetic Algorithm for Optimal Imperceptibility in Image Communication through Noisy Channel. In: Proceedings of the International Conference on Neural Information Processing, pp. 700–705 (2004)
13. Huang, C.H., Wu, J.L.: A watermark optimization technique based on genetic algorithms. In: Proceeding of SPIE - Visual Communications Image Processing, pp. 516–523 (2000)
14. Shieh, C.S., Huang, H.C., Wang, F.H., Pan, J.S.: Genetic watermarking based on transform-domain techniques. PElsevier Science Pattern Recognition 37, 555–565 (2004)
15. Chang, Y.-L., Sun, K.-T., Chen, Y.-H.: ART2-Based Genetic Watermarking. In: Proceedings of the International Conference on Advanced Information Networking and Applications, pp. 729–734 (2005)
16. Kumsawat, P., Attakitmongcol, K., Srikaew, A.: A New Approach for Optimization in Wavelet-based Image Watermarking by using Genetic Algorithm. In: Proceedings of the International Conference on Artificial Intelligence and Applications, pp. 328–332 (2005)
17. Kumsawat, P., Attakitmongcol, K., Srikaew, A.: A New Approach for Optimization in Image Watermarking by using Genetic Algorithm. IEEE Transactions on Image Processing 53, 4707–4719 (2005)
18. Wu, Y.-T., Shih, F.Y.: Genetic algorithm based methodology for breaking the steganalytic systems. IEEE Transactions on Systems, Man, and Cybernetics 36, 24–31 (2006)
19. Wang, Z., Bovik, A.C.: Modern Image Quality Assessment. Morgan & Claypool, San Francisco (2006)
20. Voloshynovskiy, S., Herrigel, A., Baumgaertner, N., Pun, T.: A Stochastic Approach to Content Adaptive Digital Image Watermarking. In: Proceedings of the International Workshop on Information Hiding, pp. 211–236 (1999)
21. Barni, M., Bartolini, F., Piva, A.: Improved Wavelet-Based Watermarking Through Pixel-Wise Masking. IEEE Transactions on Image Processing 10, 783–791 (2001)
22. Tzeng, J., Hwang, W.-L., Chern, I.-L.: An asymmetric subspace watermarking method for copyright protection. IEEE Transactions on Signal Processing 53, 784–792 (2005)
23. Boato, G., De Natale, F.G.B., Fontanari, C.: An Improved Asymmetric Watermarking Scheme Suitable for Copy Protection. IEEE Transactions on Signal Processing 54, 2833–2834 (2006)
24. Boato, G., De Natale, F.G.B., Fontanari, C.: Digital Image Tracing by Sequential Multiple Watermarking. IEEE Transactions on Multimedia 9, 677–686 (2007)

Dither Modulation in the Logarithmic Domain

Pedro Comesaña and Fernando Pérez-González*

Signal Theory and Communications Department
University of Vigo, Vigo 36310, Spain
{pcomesan,fperez}@gts.tsc.uvigo.es

Abstract. Scaling attacks are well-known to be some of the most harmful strategies against quantization-based watermarking methods, as they desynchronize the decoder, completely ruining the performance of the watermarking system with almost non perceptually altering the watermarked signal. In this paper we propose a new family of quantization-based methods, based on both Dither Modulation and Spread Transform Dither Modulation, oriented to deal with those attacks, and which presents another outstanding property: they produce perceptually shaped watermarks.

1 Introduction

After that Chen and Wornell [1] showed that the capacity of an Additive White Gaussian Noise could be achieved in a scenario where the state channel is known by the encoder but not known by the decoder using quantization-based techniques, this kind of techniques has been paid increasing interest by the data hiding researcher community. Nevertheless, when non-additive channels are employed the performance of quantization-based techniques could be worse that the classic spread-spectrum based methods. This is the case, for example, of the scaling attacks, that have also the good property of producing a reduced perceptual distortion, explaining why the interest on quantization-based methods robust to scaling is awakening. Although some proposals are already available in the literature [2,3], some of them based on a non-linear transformation (e.g., A-law compansion) previous to the embedding [4], this is still an open topic that we will study in this paper from an innovative approach: the watermark will be embedded in the logarithmic domain using a quantization based system; the cases where a projection is performed previously to the quantization, and where the logarithmic transform of the host signal is not projected will be compared.

* This work was partially funded by *Xunta de Galicia* under projects PGIDT04 TIC322013PR, PGIDT04 PXIC32202PM, and Competitive research units program Ref. 150/2006; MEC project DIPSTICK, reference TEC2004-02551/TCM and European Comission through the IST Programme under Contract IST-2002-507932 ECRYPT. ECRYPT disclaimer: The information in this paper is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 308–322, 2008.

The followed notation, as well as the description of the proposed methods are provided in Sect. 2. Those methods are analyzed from power and probability of error perspectives in Sect. 3 and 4, respectively. The projection based versions of these schemes are presented in Sect. 5, whereas in Sect. 6 we deal with their perceptual properties, and some interesting links with multiplicative watermarking are established. Finally, conclusions and future lines are given in Sect. 7.

2 Method Description

2.1 Notation and Framework

In this section we introduce our proposed methods to solve the problems due to the valumetric attack. In order to do so, we previously need to introduce some notation. We will denote scalar random variables with capital letters (e.g., X) and their outcomes with lowercase letters (e.g. x). The same notation criterion applies to random vectors and their outcomes, denoted in this case by bold letters (e.g. $\mathbf{X}$, $\mathbf{x}$). The ith component of a vector $\mathbf{X}$ is denoted as X_i. In this way, the data hiding problem can be summarized as follows: the embedder wants to transmit a symbol b, which we assume to be binary ($b \in \{0,1\}$), to the decoder by adding the watermark $\mathbf{w}$ to the original host vector $\mathbf{x}$, both of them of length L. Merely for analytical purposes, we will model these signals as realizations of random vectors $\mathbf{W}$, and $\mathbf{X}$, respectively, being the components of the last one i.i.d.. Let $Q_\Delta(\cdot)$ be the base uniform scalar quantizer, with quantization step Δ, and $\mathbf{D}$ denote the dithering vector, $\mathbf{D} \sim U[-\Delta/2, \Delta/2)^L$. The power of the original host signal will be denoted by $D_h \triangleq \frac{1}{L}\sum_{i=1}^{L} \sigma_{X_i}^2$, where $\sigma_{X_i}^2 \triangleq \mathrm{Var}\{X_i\}$, whereas the power of the watermark is given by $D_w \triangleq \frac{1}{L}\sum_{i=1}^{L} \mathrm{E}\{W_i^2\}$. The resulting watermarked signal can be written as $\mathbf{y} = \mathbf{x} + \mathbf{w}$. On the other hand, the decoder receives the signal $\mathbf{z} = \mathbf{y} + \mathbf{n}$, where $\mathbf{n}$ is a noise vector, which can be seen as realization of random i.i.d. vector $\mathbf{N}$, with $D_n \triangleq \frac{1}{L}\sum_{i=1}^{L} \mathrm{E}\{N_i^2\}$. Finally, the decoder estimates the embedded symbol with a suitable decoding function.

In order to compare the power of the host signal and the watermark, we use the Document to Watermark Ratio (DWR), defined as $\mathrm{DWR} = D_h/D_w$; similarly, the Document to Noise Ratio (DNR) is defined as $\mathrm{DNR} = D_h/D_n$.

2.2 Proposed Methods

The proposed techniques are based on the quantization of the original host signal *in the logarithmic domain.* Firstly, we will address the logarithmic version of Dither Modulation (DM) [1], whose embedding function is given by

$$\log(|y_i|) = Q_\Delta\left(\log(|x_i|) - \frac{b_i\Delta}{2} - d_i\right) + \frac{b_i\Delta}{2} + d_i.$$

A further step toward a scaling resistant scheme would be a differential watermarking method in the logarithmic domain, where the embedding procedure can be described as

$$\log(|y_i|) = Q_{\Delta}\left(\log(|x_i|) - \log(|y_{i-1}|) - \frac{b_i \Delta}{2} - d_i\right) + \log(|y_{i-1}|) + \frac{b_i \Delta}{2} + d_i,$$

being $\log(|y_0|)$ an arbitrary number shared by embedder and decoder. In both cases

$$y_i = \mathrm{sign}(x_i) \cdot e^{\log(|y_i|)}. \quad (1)$$

3 Power Analysis

Given that the components of the involved vectors are assumed to be i.i.d., the power of the watermark, both for the differential and non-differential methods, is given by

$$\mathrm{Var}\{w\} \triangleq \sigma_W^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2}\left(\sum_{m=-\infty}^{\infty}\int_{e^{m\Delta-\Delta/2+\tau}}^{e^{m\Delta+\Delta/2+\tau}}(|x| - e^{m\Delta+\tau})^2 f_{|X|}(|x|)dx\right)d\tau.$$

If the host signal follows a zero-mean Gaussian distribution, then we can write

$$\sigma_W^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} 2\left(\sum_{m=-\infty}^{\infty}\int_{e^{m\Delta-\Delta/2+\tau}}^{e^{m\Delta+\Delta/2+\tau}}(x - e^{m\Delta+\tau})^2 \frac{e^{-\frac{x^2}{2\sigma_X^2}}}{\sqrt{2\pi\sigma_X^2}}dx\right)d\tau.$$

Defining $x_1 \triangleq \log(x) - \tau$ and $x_2 \triangleq \tau - \log(\sigma_X)$, we can write

$$\sigma_W^2 = \frac{1}{\Delta}\int_{-\Delta/2-\log(\sigma_X)}^{\Delta/2-\log(\sigma_X)} 2\cdot\left[\sum_{m=-\infty}^{\infty}\int_{m\Delta-\Delta/2}^{m\Delta+\Delta/2}\right.$$
$$\left.\sigma_X^2 e^{2x_2}(e^{x_1} - e^{m\Delta})^2 \frac{e^{-\frac{e^{2(x_1+x_2)}}{2}}}{\sqrt{2\pi}} e^{x_1+x_2}dx_1\right]dx_2.$$

Denoting by $g(x_2)$ the function inside the brackets in the last formula, it is clear that σ_W^2 would be proportional to σ_X^2[1] if $g(x_2)$ were a periodic function with period Δ, for any given value of Δ. In fact,

$$g(x_2 + \Delta) = \sum_{m=-\infty}^{\infty}\int_{m\Delta-\Delta/2}^{m\Delta+\Delta/2}$$
$$\sigma_X^2(e^{x_1+x_2+\Delta} - e^{m\Delta+x_2+\Delta})^2 \frac{e^{-\frac{e^{2(x_1+x_2+\Delta)}}{2}}}{\sqrt{2\pi}} e^{x_1+x_2+\Delta}dx_1,$$

which making $x_3 = x_1 + \Delta$, yields

$$\sum_{m=-\infty}^{\infty}\int_{(m+1)\Delta-\Delta/2}^{(m+1)\Delta+\Delta/2}\sigma_X^2(e^{x_3+x_2} - e^{(m+1)\Delta+x_2})^2 \frac{e^{-\frac{e^{2(x_3+x_2)}}{2}}}{\sqrt{2\pi}} e^{x_3+x_2}dx_3,$$

showing the periodicity of $g(x)$.

[1] This would imply that the *Document to Watermark Ratio* (DWR) would be independent of σ_X^2, and therefore just a function of Δ.

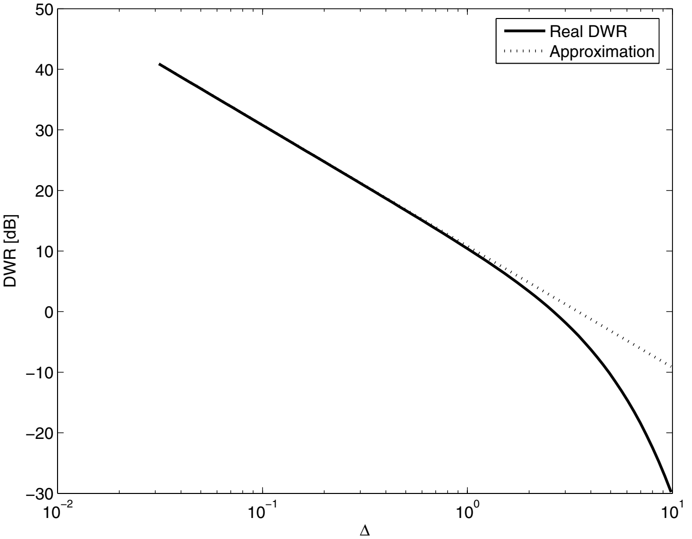


Fig. 1. Comparison of the exact DWR and the obtained approximation as a function of Δ

3.1 Computation of an Approximation to the Embedding Distortion for Small Values of the Quantization Step

Taking into account that the dither is independent of the host, and uniformly distributed in $[-\Delta/2, \Delta/2)^L$, $\log(|y_i|) - \log(|x_i|)$ will be also uniformly distributed in $[-\Delta/2, \Delta/2)^L$, regardless of the value of $\mathbf{x}$. This implies that we can write $\log(|\mathbf{y}|) = \log(|\mathbf{x}|) + \mathbf{v}$, where $\mathbf{v}$ is uniform in $[-\Delta/2, \Delta/2)^L$, so $|y_j| = |x_j|e^{v_j}$, with $1 \leq j \leq L$. Therefore, the power of the watermark, both for the differential and non-differential methods, is given by

$$\sigma_W^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2}\int_{-\infty}^{\infty}[x(1-e^v)]^2 f_X(x)dxdv. \tag{2}$$

For small values of Δ, i.e. $\Delta << 1$, which is reasonable due to imperceptibility constraints, we can approximate $1 - e^v \approx -v$, so $\frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2}(1-e^v)^2 dv \approx \frac{\Delta^2}{12}$, yielding $\sigma_W^2 \approx \sigma_X^2 \frac{\Delta^2}{12}$, for any distribution of the original host signal. The actual values of the DWR can be compared with the previous approximation in Fig. 1, showing the good behavior of the proposed approximation whenever $\Delta << 1$.

4 Probability of Error

4.1 Non-differential Scheme

Considering the periodic nature of the decision region in the logarithmic domain, it is straightforward to show that the probability of decoding error when the minimum distance decoder is used is given by

$$P_e = \Pr\left\{ |\log(|Z_i|) - D_i - Q_{\Delta}(\log(|Z_i|) - D_i)| \geq \frac{\Delta}{4} \right\}$$
$$= \Pr\left\{ |[\log(|Z_i|) - D_i] \mod \Delta| \geq \Delta/4 \right\}, \tag{3}$$

where we have assumed, without loss of generality in the obtained results, that $b = 0$. Noticing that $\log(|Y_i|) = D_i + m\Delta$, with $m \in \mathbb{Z}$, such probability of error can be rewritten as

$$P_e = \Pr\left\{ |[\log(|Z_i|) - \log(|Y_i|)] \mod \Delta| \geq \Delta/4 \right\}$$
$$= \Pr\left\{ \left| \log\left(\left| 1 + \frac{N_i}{Y_i} \right| \right) \mod \Delta \right| \geq \Delta/4 \right\}.$$

Given that both $\mathbf{N}$ and $\mathbf{Y}$ are i.i.d., we will disregard the subindex, and write $\log(|N/Y|) = \log(|N|) - \log(|Y|)$. If both the host signal and the noise are Gaussian we have that $f_{\log(|X|)}(x) = \frac{2}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{e^{2x}}{2\sigma_X^2}} e^x$, and similarly for $f_{\log(|N|)}(n)$, so taking into account that $\log(|Y|) = \log(|X|) + V$, where V follows a uniform distribution on $[-\Delta/2, \Delta/2)$, the pdf of $\log(|N/Y|) = \log(|N|) - \log(|X|) - V$ can be written as

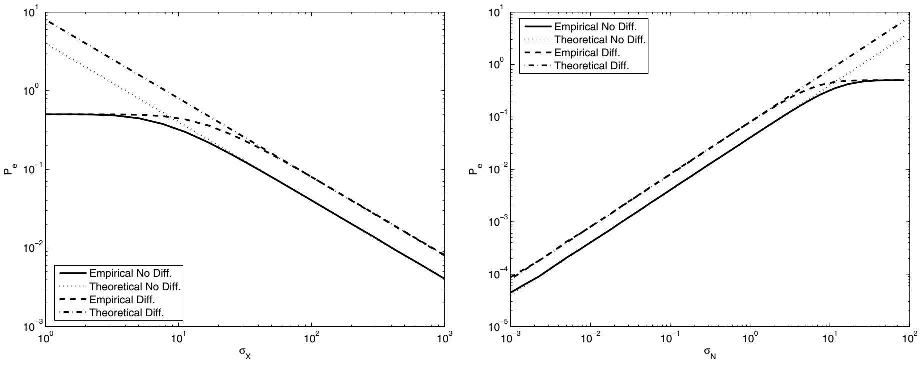
$$f_{\log(|N/Y|)}(x) = \frac{1}{\Delta} \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi\sigma_N^2}} e^{-\frac{e^{2(x-\tau_2)}}{2\sigma_N^2}} e^{x-\tau_2}$$
$$\int_{-\Delta/2}^{\Delta/2} \frac{2}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{e^{2(-\tau_2-\tau_1)}}{2\sigma_X^2}} e^{-\tau_2-\tau_1} d\tau_1 d\tau_2$$
$$= \frac{2\left[\operatorname{arccot}\left(\frac{e^{-\Delta/2+x}\sigma_X}{\sigma_N} \right) - \operatorname{arccot}\left(\frac{e^{\Delta/2+x}\sigma_X}{\sigma_N} \right) \right]}{\pi\Delta}.$$

For large values of σ_X/σ_N, the ratio $|N/Y|$ will take small values with high probability, so in practical scenarios we can approximate $|\log(|1 + N/Y|)| \approx |N/Y|$, where we have used the fact that $\log(|1+x|) \approx x$, for $|x| << 1$. Therefore, $f_{|\log(|1+N/Y|)|}(x) \approx \frac{2\left[\operatorname{arccot}\left(\frac{e^{-\Delta/2}x\sigma_X}{\sigma_N}\right) - \operatorname{arccot}\left(\frac{e^{\Delta/2}x\sigma_X}{\sigma_N}\right)\right]}{\pi\Delta x}$. Assuming that $\Delta << 1$ and $\sigma_X/\sigma_N >> 1$, and considering that $\operatorname{arccot}(x) \approx 1/x$ when $|x| >> 1$, the last expression can be approximated for those values relevant for the computation of the probability of error by $f_{|\log(|1+N/Y|)|}(x) \approx \frac{2\sigma_N}{\sigma_X \pi x^2}$ so we can write $P_e \approx \sum_{m=1}^{\infty} \frac{2\sigma_N}{(-3\Delta/4+m\Delta)\sigma_X\pi} - \frac{2\sigma_N}{(-\Delta/4+m\Delta)\sigma_X\pi}$.

4.2 Differential Scheme

Following a reasoning similar to that described for the non-differential case, it is straightforward to see that the probability of error is now

$$P_e = \Pr\left\{ \left| \left[\log\left(\left| 1 + \frac{N_i}{Y_i} \right| \right) - \log\left(\left| 1 + \frac{N_{i-1}}{Y_{i-1}} \right| \right) \right] \mod \Delta \right| \geq \Delta/4 \right\}.$$



(a) (b)

Fig. 2. (a) Empirical and theoretical decoding error probabilities as a function of σ_X, for both the differential and non-differential logarithmic versions of DM. $\sigma_N = 1$, $\Delta = 0.5$, and both **X** and **N** are Gaussian distributed. (b) Empirical and theoretical decoding error probabilities as a function of σ_N, for both the differential and non-differential logarithmic versions of DM. $\sigma_X = 100$, $\Delta = 0.5$, and both **X** and **N** are Gaussian distributed.

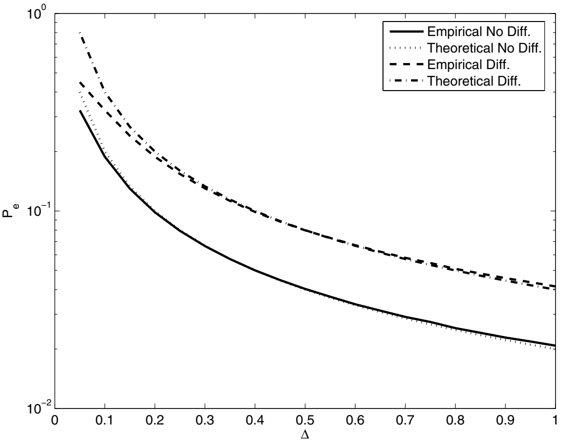


Fig. 3. Empirical and theoretical decoding error probabilities as a function of Δ, for both the differential and non-differential logarithmic versions of DM. $\sigma_X = 100$, $\sigma_N = 1$, and both **X** and **N** are Gaussian distributed.

In this case we will use the fact that the distribution of Y, and therefore the distribution of $\frac{N}{Y}$, is asymptotically independent of Δ for small values of Δ, so we can approximate the distribution of $\log(|N/Y|)$ as

$$f_{\log(|N/Y|)}(x) \approx f_{\log(|N/X|)}(x) = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi\sigma_N^2}} e^{-\frac{e^{2\tau}}{2\sigma_N^2}} e^{\tau} \frac{2}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{e^{2(\tau-x)}}{2\sigma_X^2}} e^{\tau-x} d\tau$$
$$= \frac{2\sigma_X\sigma_N e^x}{\pi\left(\sigma_X^2 e^{2x} + \sigma_N^2\right)},$$

and given that $|N/Y| \approx |N/X| << 1$, we can write $\log(|1+N/Y|) \approx N/Y \approx N/X$, so $f_{|\log(|1+N/Y|)|}(x) \approx \frac{2\sigma_X\sigma_N}{\pi\left(\sigma_X^2 x^2+\sigma_N^2\right)}$, $x \geq 0$. Be aware that for large values of σ_X/σ_N the last formula can be approximated by $\frac{2\sigma_N}{\pi\sigma_X x^2}$, which coincides with the approximation to the pdf of $|\log(|1+N/Y|)|$ obtained in Section 4.1. Considering that N/Y will take positive and negative values with the same probability it follows that

$$f_{\log(|1+N/Y|)}(x) \approx \frac{\sigma_X\sigma_N}{\pi\left(\sigma_X^2 x^2+\sigma_N^2\right)}, \text{ for all } x \in \mathbb{R}. \quad (4)$$

From the last equation, it can be shown that the pdf of $x_{\text{diff}} \triangleq \log\left(\left|1+\frac{N_i}{Y_i}\right|\right) - \log\left(\left|1+\frac{N_{i-1}}{Y_{i-1}}\right|\right)$ can be approximated by $f_{x_{\text{diff}}}(x) \approx \frac{2\sigma_X^3\sigma_N^2 x}{\pi(4\sigma_X^2\sigma_N^3 x+\sigma_X^4\sigma_N x^3)}$, which assuming that $\sigma_X >> \sigma_N$, it can be approximated as $f_{x_{\text{diff}}}(x) \approx \frac{2\sigma_N}{\pi\sigma_X x^2}$, so the probability of decoding error can be written as

$$\begin{aligned}
&\Pr\{|x_{diff} \mod \Delta| \geq \Delta/4\}\\
&\approx \sum_{m=-\infty}^{\infty} \frac{2\sigma_N}{(-3\Delta/4+m\Delta)\sigma_X\pi} - \frac{2\sigma_N}{(-\Delta/4+m\Delta)\sigma_X\pi}\\
&= 2\left(\sum_{m=1}^{\infty} \frac{2\sigma_N}{(-3\Delta/4+m\Delta)\sigma_X\pi} - \frac{2\sigma_N}{(-\Delta/4+m\Delta)\sigma_X\pi}\right).
\end{aligned}$$

This is nothing but twice the probability of decoding error obtained for the non-differential scheme, implying that for a given value of Δ, and therefore a fixed value of DWR, the WNR needed for achieving a certain probability of decoding error is increased by 6 dB (compared with the non-differential one) when the differential scheme is used. On the other hand, the differential scheme makes the resulting scheme completely invulnerable to valumetric attacks using a constant scaling factor, and even robust to attacks where that factor changes slowly. In Figs. 2(a), 2(b) and 3, we can see the good fit of the empirical results with the obtained approximations, especially for the specified asymptotic values.

5 Logarithmic STDM

A further step in the side-informed logarithmic data hiding techniques introduced in this paper is the adaptation of classical projection and quantization based techniques, e.g. *Spread Transform Dither Modulation* (STDM) [1]. These techniques have been extensively studied in several works in the literature [5,6], analyzing their embedding distortion, robustness to additive attacks, to quantization and to valumetric attacks. This last attack was shown to be really harmful to STDM techniques [6], so it seems reasonable to think of of a logarithmic version of STDM which could simultaneously deal with this problem and produce perceptually shaped watermarks.

The embedding process for logarithmic STDM in a M-dimensional projected domain using uniform scalar quantizers can be described for the non-differential case as

$$\begin{aligned}\mathbf{x}_p &\triangleq \mathbf{S}^T \log(|\mathbf{x}|),\\ y_{p_i} &= Q_\Delta\left(x_{p_i} - \frac{b_i\Delta}{2} - d_i\right) + \frac{b_i\Delta}{2} + d_i,\ \ 1 \le i \le M,\\ \log(|\mathbf{y}|) &= \log(|\mathbf{x}|) + \mathbf{S}(\mathbf{S}^T\mathbf{S})^{-1}(\mathbf{y}_p - \mathbf{x}_p),\end{aligned}$$

where $\mathbf{S}$ is a $L \times M$ projection matrix, $\mathbf{D}$ is now uniformly distributed in $[-\Delta/2, \Delta/2)^M$ and (1) is still applied for computing the samples of the watermarked signal in the original domain. Correspondingly, for the differential case the projected watermarked signal in the logarithmic domain is computed as $y_{p_i} = Q_\Delta\left(x_{p_i} - y_{p_{i-1}} - \frac{b_i\Delta}{2} - d_i\right) + y_{p_{i-1}} + \frac{b_i\Delta}{2} + d_i$, where y_{p_0} is again assigned an arbitrary number shared by embedder and decoder.

For the sake of simplicity, through this section we will assume that $\mathbf{S}$ is a scaled orthonormal matrix, so $\mathbf{S}^T\mathbf{S} = K_1\mathbf{I}_{M\times M}$, $K_1 > 0$, where $\mathbf{I}_{M\times M}$ denotes the M-dimensional identity matrix. Additionally, we will require $\mathbf{S}$ to verify $\sum_{i=1}^M s_{j,i}^2 = K_2$, $K_2 > 0$, for all $1 \le j \le L$, i.e., all its rows will have the same Euclidean norm. These two assumptions imply that $M \cdot K_1 = L \cdot K_2$.

5.1 Power Analysis

Similarly to the logarithmic DM scheme, the subsequent power analysis is valid for both the non-differential and differential logarithmic STDM schemes.

Defining $\mathbf{V} \triangleq \mathbf{S}(\mathbf{S}^T\mathbf{S})^{-1}(\mathbf{Y}_p - \mathbf{X}_p) = \frac{1}{K_1}\mathbf{S}(\mathbf{Y}_p - \mathbf{X}_p)$, we can see that $(\mathbf{Y}_p - \mathbf{X}_p)$ is independent of $\mathbf{X}$ due to the dither vector $\mathbf{D}$ being uniformly distributed on $[-\Delta/2, \Delta/2)^M$; therefore, the average power per dimension of $\mathbf{V}$ can be computed as

$$\frac{1}{L}\mathrm{E}\{||\mathbf{V}||^2\} = \frac{1}{K_1 \cdot L}\mathrm{E}\{||(\mathbf{Y}_p - \mathbf{X}_p)||^2\} = \frac{M}{K_1 \cdot L}\frac{\Delta^2}{12}. \quad (5)$$

Based on the independence of $\mathbf{X}$ and $\mathbf{V}$, on the fact that all the rows of $\mathbf{S}$ have the same Euclidean norm, and on the value of the power per dimension of $\mathbf{V}$, we can recover (2) to write in this case

$$\sigma_W^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[x(1-e^v)]^2 f_X(x) f_V(v) dx dv.$$

For $\Delta << 1$ it is reasonable to approximate $1 - e^v \approx -v$, so whenever that condition is verified (as it will be the case in most practical applications) we can write

$$\sigma_W^2 \approx \left[\int_{-\infty}^{\infty} v^2 f_V(v) dv\right]\left[\int_{-\infty}^{\infty} x^2 f_X(x) dx\right] = \frac{M}{K_1 \cdot L}\frac{\Delta^2\sigma_X^2}{12},$$

for any distribution of the original host signal.

In order to compare this result with previous ones in the literature, we can refer to [1] and [5], where the *classical* STDM scheme is studied. In [1] the projecting vector is assumed to be normalized in power (i.e., $K_1 = 1$), so the embedding distortion is given by $\sigma_W^2 = \frac{\Delta^2 M}{12 \cdot L}$. On the other hand, in [5] $M = 1$ and $K_1 = L$, yielding a watermark power $\sigma_w^2 = \frac{1}{L^2}\Delta^2 \cdot I(\sigma_X, \Delta)$, where $I(\sigma_X, \Delta)$ accounts for the non-uniformity of the host within the quantization cells, and takes values in the interval $[1/16, 1/12]$. In the present case, we are assuming that uniform dither is used, so $I(\sigma_X, \Delta) = \frac{1}{12}$ for all pairs (σ_X, Δ). These results agree with (5); nevertheless, for logarithmic STDM, as it also happens for the logarithmic DM, the embedding power is given by (5) multiplied by the power of the original host signal, so the DWR is just a function of Δ for any distribution of the original host signal.

5.2 Probability of Error

Non-differential Scheme. In this case, the probability of error of the minimum distance decoder is similar to (3), taking the value

$$P_e = \Pr\left\{|\left(Z_{p_i} - D_i\right) \mod \Delta| \geq \Delta/4\right\},$$

with $\mathbf{Z}_p = \mathbf{S}^T \log(|\mathbf{Z}|)$. Reasoning in the same way as in Sect. 4.1, one can see that $Y_{p_i} = D_i + m\Delta$, so the probability of error can be rewritten as

$$\begin{aligned} P_e &= \Pr\left\{|(Z_{p_i} - Y_{p_i}) \mod \Delta| \geq \Delta/4\right\} \\ &= \Pr\left\{\left|\left[\sum_{j=1}^{L} s_{j,i} \log\left(\left|1 + \frac{N_j}{Y_j}\right|\right)\right] \mod \Delta\right| \geq \Delta/4\right\}. \end{aligned}$$

In order to obtain analytical forms for the last formula, hereafter we will consider the case where $s_{j,i} \in \{-1, 0, +1\}$, for all $1 \leq i \leq M$ and $1 \leq j \leq L$. Under that assumption, the constraint stating that $\mathbf{S}^T\mathbf{S} = K_1 I_{M\times M}$ on the beginning of this section can be interpreted as all the columns of $\mathbf{S}$ having $K_1 = \frac{LK_2}{M}$ non-zero elements. Therefore, we can define

$$T_i \triangleq \sum_{k=1}^{K_1} s_{j_i(k),i} \log\left(\left|1 + \frac{N_{j_i(k)}}{Y_{j_i(k]}}\right|\right), \quad 1 \leq i \leq M,$$

with $j_i(k)$ the index of the k-th element of the i-th column which is non-zero. As it was discussed in Sect. 4.2 for small values of Δ and large values of σ_X, $\log(|1 + N_j/Y_j|)$ goes asymptotically to N_j/X_j, so we have $T_i \approx \sum_{k=1}^{K_1} s_{j_i(k),i} \frac{N_{j_i(k)}}{X_{j_i(k)}}$, $1 \leq i \leq M$. Therefore, for the Gaussian case the pdf of T_i is asymptotically the convolution of K_1 i.i.d. random variables, each of them with pdf

$$f_{\log(|1+N/Y|)}(x) \approx \frac{\sigma_X \sigma_N}{\pi\left(\sigma_X^2 x^2 + \sigma_N^2\right)}, \text{ for all } x \in \mathbb{R}, \tag{6}$$

so we can write

$$f_{T_i}(x) \approx \frac{K_1 \sigma_X \sigma_N}{\pi\left(\sigma_X^2 x^2 + K_1^2 \sigma_N^2\right)}. \tag{7}$$

Given that the pdf (6) is just an approximation of the true pdf, the exactness of (7) will decrease when K_1 is increased, i.e. when the number of these approximated pdfs which are convoluted is increased. On the other hand, the larger σ_X, the smaller Δ, and the smaller σ_N, the more accurate (7) will be.

Finally, taking into account (7), the probability of decoding error is given by

$$\begin{aligned} P_e &\approx \sum_{m=-\infty}^{\infty} \int_{-3\Delta/4+m\Delta}^{-\Delta/4+m\Delta} \frac{K_1 \sigma_X \sigma_N}{\pi(\sigma_X^2 x^2 + K_1^2 \sigma_N^2)} \\ &= \sum_{m=-\infty}^{\infty} \frac{1}{\pi}\left[\arctan\left(\frac{\sigma_X \cdot (-\Delta/4+m\Delta)}{K_1\sigma_N}\right) - \arctan\left(\frac{\sigma_X \cdot (-3\Delta/4+m\Delta)}{K_1\sigma_N}\right)\right] \end{aligned} \tag{8}$$

Be aware that in this case we have not disregarded the variance of the attacking noise σ_N^2, as it was done in the computation of $f_{x_{\text{diff}}}(x)$ in Sect. 4.2, since in the current case this variance is multiplied by the number of non-zero elements in each column of $\mathbf{S}$, i.e. K_1.

In order to perform a fair comparison between the performance of the non-differential version of STDM and the non-differential version of DM, we will choose a value of Δ for STDM yielding the same embedding power than that obtained for DM, so $\Delta_{\text{STDM}} = \sqrt{\frac{L \cdot K_1}{M}} \Delta_{\text{DM}}$. On the other hand,

$$\sum_{m=-\infty}^{\infty} \arctan\left(x(-1/4+m)\right) - \arctan\left(x(-3/4+m)\right),$$

is a decreasing function of x, so introducing the value of Δ_{STDM} in (8), one can easily see that the probability of decoding error is minimized by minimizing K_1. But K_1 is equal to $\frac{LK_2}{M}$, so its minimum value for a given value of M is $\frac{L}{M}$; this corresponds to the case where only one element per row of $\mathbf{S}$ is non-zero, coinciding the computed probability of error with that obtained for the logarithmic DM, independently of M. Therefore, given that small values of M imply a reduction in the achievable rate, and the probability of decoding error of the system is not modified by this parameter, one would be interested in having a large value of M; in fact, it turns out that in the current framework, and upon the aforementioned approximations, the optimal strategy of STDM is that with $M = L$ and $K_1 = K_2 = 1$, which is nothing but DM without the projecting operation. Therefore, although DM could be seen as a particular case of STDM, we will focus in the remainder of the paper on DM, as it is the optimal choice according to the former performance analysis. Finally, we would like to emphasize that we have just taken into account the probability of error in order to compare DM and STDM, disregarding other criteria that could be also valuable when designing a watermarking method, as it might be the security of the resulting scheme.

Differential scheme. For the differential scheme the probability of decoding error is given by

$$P_e = \Pr\left\{ \left| \left(Z_{p_i} - Z_{p_{i-1}} - D_i \right) \mod \Delta \right| \geq \Delta/4 \right\},$$

so from the fact that $D_i = Y_{p_i} - Y_{p_{i-1}} + m\Delta$, with $m \in \mathbb{Z}$, one can write

$$\begin{aligned} P_e &= \Pr\left\{ \left| \left(Z_{p_i} - Y_{p_i} - Z_{p_{i-1}} + Y_{p_{i-1}} \right) \mod \Delta \right| \geq \Delta/4 \right\} \qquad (9) \\ &= \Pr\left\{ \left| \left[\sum_{j=1}^{L} s_{j,i} \log\left(\left| 1 + \frac{N_j}{Y_j} \right| \right) - \sum_{j=1}^{L} s_{j,i-1} \log\left(\left| 1 + \frac{N_j}{Y_j} \right| \right) \right] \mod \Delta \right| \geq \Delta/4 \right\}. \end{aligned}$$

For the sake of simplicity we will constrain our analysis to the case where the assumptions on $\mathbf{S}$ introduced for the non-differential case are verified, i.e. $s_{j,i} \in \{-1, 0, +1\}$, for all $1 \leq i \leq M$ and $1 \leq j \leq L$. Therefore, (9) is equivalent to $P_e = \Pr\{|T_i \mod \Delta| \geq \Delta/4\}$, where

$$T_i \triangleq \sum_{k=1}^{K_1} s_{j_i(k),i} \log\left(\left| 1 + \frac{N_{j_i(k)}}{Y_{j_i(k)}} \right| \right) - \sum_{k=1}^{K_1} s_{j_{i-1}(k),i-1} \log\left(\left| 1 + \frac{N_{j_{i-1}(k)}}{Y_{j_{i-1}(k)}} \right| \right),$$

with $1 \leq i \leq M$.

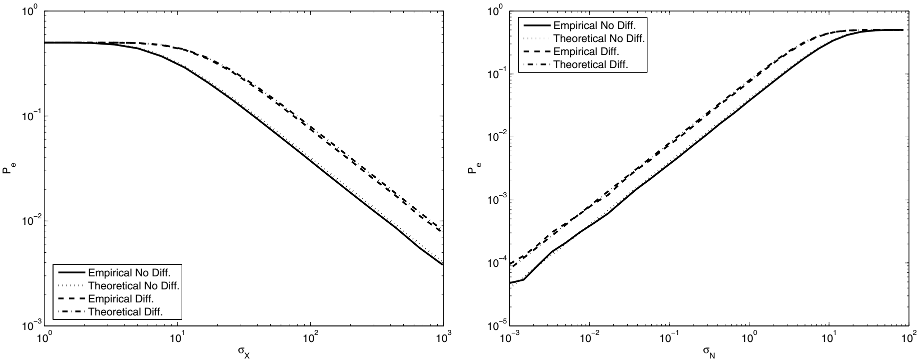
Furthermore, whenever Δ takes small values and σ_X takes large ones, T_i can be approximated as $T_i \approx \sum_{k=1}^{K_1} s_{j_i(k),i} \frac{N_{j_i(k)}}{X_{j_i(k)}} - s_{j_{i-1}(k),i-1} \frac{N_{j_{i-1}(k)}}{X_{j_{i-1}(k)}}$. If $\mathbf{S}$ is pseudorandomly computed depending on a secret key (as it will happen in most of practical applications in order to improve the security of the resulting scheme), verifying the constraints previously introduced, and if L is large, and K_1 is small, then the probability of finding a pair (k_1, k_2) such that $j_i(k_1) = j_{i-1}(k_2)$, with $1 \leq k_1, k_2 \leq K_1$, will be small, so T_i can be approximated as the sum of $2 \cdot K_1$ i.i.d. random variables, each of them following the pdf given by (6). Therefore, the pdf of T_i can be approximated as $f_{T_i}(x) \approx \frac{2K_1\sigma_X\sigma_N}{\pi\left(\sigma_X^2 x^2 + 4K_1^2\sigma_N^2\right)}$, being still valid the considerations about its accuracy discussed for the non-differential case. From the last equation the probability of decoding error can be approximated as

$$\begin{aligned} P_e &\approx \sum_{m=-\infty}^{\infty} \int_{-3\Delta/4+m\Delta}^{-\Delta/4+m\Delta} \frac{2K_1\sigma_X\sigma_N}{\pi\left(\sigma_X^2 x^2 + 4K_1^2\sigma_N^2\right)} \\ &= \sum_{m=-\infty}^{\infty} \frac{1}{\pi} \left[\arctan\left(\frac{\sigma_X \cdot (-\Delta/4 + m\Delta)}{2K_1\sigma_N} \right) - \arctan\left(\frac{\sigma_X \cdot (-3\Delta/4 + m\Delta)}{2K_1\sigma_N} \right) \right]. \end{aligned}$$

Finally, Figs. 4 and 5 show the empirical probability of decoding error, as well as their theoretical approximations, for both the non-differential and differential cases, showing the accuracy of the proposed approximations, and the validity of our analysis.

6 Perceptual Masking

Another interesting characteristic of the proposed methods is the perceptual shape of the obtained watermark; the quantization step in the original domain



(a) (b)

Fig. 4. (a) Empirical and theoretical decoding error probabilities as a function of σ_X, for both the differential and non-differential logarithmic versions of STDM. $\sigma_N = 1$, $\Delta = 5$, and both $\mathbf{X}$ and $\mathbf{N}$ are Gaussian distributed. (b) Empirical and theoretical decoding error probabilities as a function of σ_N, for both the differential and non-differential logarithmic versions of STDM. $\sigma_X = 100$, $\Delta = 5$, and both $\mathbf{X}$ and $\mathbf{N}$ are Gaussian distributed.

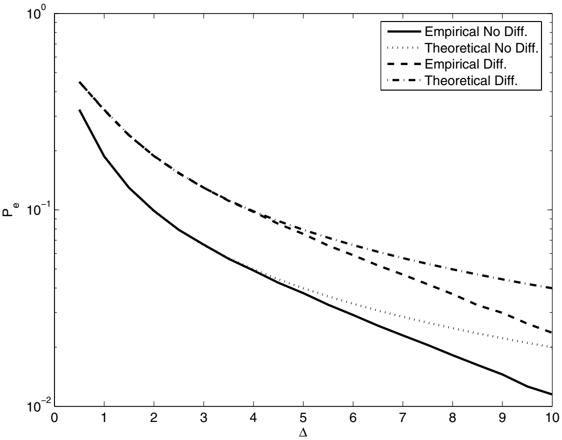


Fig. 5. Empirical and theoretical decoding error probabilities as a function of Δ, for both the differential and non-differential logarithmic versions of STDM. $\sigma_X = 100$, $\sigma_N = 1$, and both $\mathbf{X}$ and $\mathbf{N}$ are Gaussian distributed.

is increased with the magnitude of the host, introducing a larger watermark amplitude when the host signal takes large values. This effect makes sense from a perceptual point of view, since the human visual system performs the so-called *contrast masking*, the reduction of the visibility of one image component in presence of another. This phenomenon, which is reflected on the perceptual distortion measure introduced by Watson in [7], constitutes the motivation for multiplicative spread spectrum data hiding techniques, where it is *desirable that larger host features bear a larger watermark* [8]; recent works on

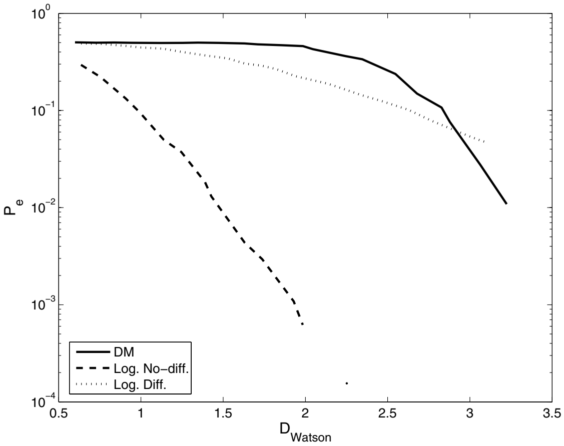


Fig. 6. Probability of error vs. Watson's perceptual embedding distortion for DM and the proposed differential and non-differential schemes, when the watermarked signal is attacked with i.i.d. Gaussian noise. Watermark introduced in the DCT domain. DNR = 35 dB. Repetition rate = 1/100. Image *Baboon* 256 × 256.

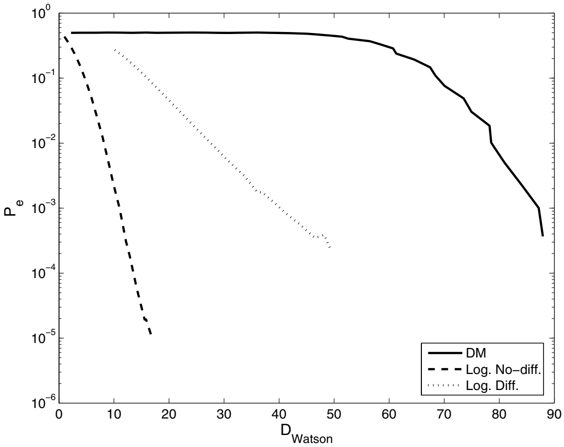


Fig. 7. Probability of error vs. Watson's perceptual embedding distortion for DM and the proposed differential and non-differential schemes, when the watermarked signal is attacked with i.i.d. Gaussian noise. Watermark introduced in the DCT domain. DNR = 35 dB. Repetition rate = 1/100. Image *Man* 1024 × 1024.

video watermarking have also chosen multiplicative methods based on perceptual considerations [9]. Furthermore, these techniques, where the embedding process is given by $y_i = x_i(1 + \eta s_i)$, with $\mathbf{s}$ the spreading sequence and η a distortion controlling parameter, can be interpreted in logarithmic terms, as for $|\eta s_i| << 1$ we can approximate $1 + \eta s_i \approx e^{\eta s_i}$, and $y_i \approx x_i e^{\eta s_i}$. Therefore, we can say that multiplicative spread spectrum is to additive spread spectrum watermarking, what the logarithmic techniques presented here are to Dither Modulation.

In that sense, the introduced methods can be considered as the side-informed version of the previous multiplicative spread spectrum techniques.

Returning to the perceptual justification of logarithmic (or multiplicative) techniques, in this section we will use Watson's perceptual measure to illustrate with some experimental results the performance advantages, for a given embedding perceptual distortion, of the proposed techniques when they are compared with the *classical* scalar DM data hiding technique. In order to perform this comparison, we embedded the watermark in the AC coefficients of the 8 × 8 block DCT of real images, using a repetition rate of 1/100,[2] where the attack is i.i.d. Gaussian noise with variance yielding a DNR = 35 dB. In Fig. 6 and 7 we can see the achieved probability of error as a function of the perceptual distortion measure introduced by Watson [7] due to the embedding. As expected, the non-differential strategy clearly outperforms the differential one, although the ratio between the probability of error for both cases somewhat differs from the theoretical one, due to the fact that DCT coefficients do not really follow a Gaussian distribution, as it was assumed throughout the previous sections. Nevertheless, one can also verify the good performance of the proposed logarithmic schemes compared with the *classical* DM; this improvement is based on the fact that for a given perceptual embedding distortion, DM will need a fixed (and small) quantization step, whereas the logarithmic schemes introduced in this paper can be seen as using increasing quantization steps for large values of the host signal, yielding a perceptually shaped watermark, and therefore allowing a larger Mean Squared Error (MSE) distortion for a fixed perceptual distortion.

7 Conclusions and Further Lines

In this paper we have analyzed the performance of a new family of data hiding schemes based on the quantization of the host signal in the logarithmic domain. Both non-differential and differential strategies have been considered. The intuitive idea that the last ones are more sensitive to additive noise attacks has been quantified; nevertheless, one should also consider that the differential schemes are invulnerable to valumetric attacks.

Furthermore, we have analyzed those techniques that perform a projection before the quantization, as well as those techniques that do not consider that projection, obtaining the interesting result that, under some reasonable assumptions on the projecting matrix, the performance of the latter is better than that of the former.

The usefulness of the proposed techniques is also proved by some empirical results that show the perceptual advantages of the logarithmic schemes. This goodness is based on the fact that the logarithmic schemes proposed in this paper are perceptually shaping the watermark, i.e. embedding a larger amplitude watermark in those coefficients where the original host signal is larger, so they take advantage of contrast masking.

[2] For the analysis of the probability of error for DM based on uniform scalar quantizers with repetition coding, and additive noise, the reader is referred to [10].

Finally, as future research lines it would be interesting to study generalized versions of the proposed schemes, including their distortion compensated, or their lattice based versions. Another open question is the study of improved decoding strategies: it is straightforward to see that the decoding strategy followed in this paper, i.e. minimum distance decoding, is not the optimal one, since even if the attacking noise were Gaussian, it would not longer have that distribution after applying the logarithmic transformation.

References

1. Chen, B., Wornell, G.W.: Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory 47(4), 1423–1443 (2001)
2. Abrardo, A., Barni, M.: Informed watermarking by means of orthogonal and quasi-orthogonal dirty paper coding. IEEE Transactions on Signal Processing 53(2), 824–833 (2005)
3. Pérez-González, F., Mosquera, C., Barni, M., Abrardo, A.: Rational dither modulation: a high-rate data-hiding method robust to gain attacks. IEEE Transaction on Signal Processing 53(10), 3960–3975 (2005)
4. Lee, K., Kim, D.S., Moon, K.A.: Amplitude-modification resilient watermarking based on a-law companding. In: IEEE International Conference on Image Processing, September 2003, vol. 3, pp. 459–462 (2003)
5. Pérez-González, F., Balado, F., Hernández, J.R.: Performance analysis of existing and new methods for data hiding with known-host information in additive channels. IEEE Transactions on Signal Processing 51(4), 960–980 (2003); Special Issue Signal Processing for Data Hiding in Digital Media & Secure Content Delivery
6. Bartolini, F., Barni, M., Piva, A.: Performance analysis of ST-DM watermarking in presence of nonadditive attacks. IEEE Transactions on Signal Processing 52(10), 2965–2974 (2004)
7. Watson, A.B.: DCT quantization matrices visually optimized for individual images. In: Proceedings of SPIE, vol. 1913(14), pp. 202–216 (1993); Human Vision, Visual Processing and Digital Display IV
8. Barni, M., Bartolini, F.: Watermarking Systems Engineering. Marcel Dekker, New York (2004)
9. Lemma, A., van der Veen, M., Celik, M.: A new modulation based watermarking technique for video. In: Proceedings of SPIE, vol. 6072 (2006); Security, Steganography, and Watermarking of Multimedia Contents VIII
10. Comesaña, P., Pérez-González, F., Balado, F.: On distortion-compesated dither modulation data-hiding with repetition coding. IEEE Transactions on Signal Processing 54(2), 585–600 (2006)

A Practical Real-Time Video Watermarking Scheme Robust against Downscaling Attack

Kyung-Su Kim[1], Dong-Hyuck Im[1], Young-Ho Suh[2], and Heung-Kyu Lee[1]

[1] Department of EECS, Korea Advanced Institute of Science and Technology,
Guseong-dong, Yuseong-gu, Daejeon, Republic of Korea
{kskim,iammoni,hklee}@mmc.kaist.ac.kr
[2] Digital Content Research Division, Electronics and Telecommunications Research Institute, Gajeong-dong, Yuseong-gu, Daejeon, Republic of Korea
syh@etri.re.kr

Abstract. In this paper, we propose a practical real-time video watermarking scheme for downscaling attack. We embed watermark into an arbitrary size of a host video and extract it from the watermarked video under a preset false error rate. Both watermark embedding and extraction are done in the uncompressed domain. Also, blind detector is used (i.e., no original video needs during the detection). As our watermarking system uses a video coder and decoder according to a format of an input video, it can be applicable to DCT-based coding videos (e.g., MPEG-2) as well as MPEG-1 and MPEG-4. Moreover, for a practical use both embedder and detector are developed as directshow filters that make it easy to connect after another within directshow applications. The payload of this system is totally 192 bits. PSNR after embedding is around 45dB for the host video that ranges from QVGA to High Definition (HD) size. Finally, several experimental results prove that the proposed scheme has good robustness.

1 Introduction

The development of digital multimedia technique and commercial market places where digital media are broadcast through the high-speed Internet service have already made any one has access to digital media such as audio, images and video. Since digitalized media are perfectly copied and easily modified, contents providers (CP) and sellers have difficulty protecting their copyrights. Accordingly, it is very important to detect copyright violations and control access to the digital media. Digital watermarking is used as one of popular techniques for authentication and copyright protection of the digital media [1].

Figure 1 shows a general digital right management (DRM) system in which digital watermarking technique is adopted. At first, User A chooses a favorable content to buy and requests it to DRM center. After being confirmed the payment from DRM center through payment gateway, CP verify the requested content and then packages it into the protected content by watermarking. After that, the user can access the protected content through the Internet service.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 323–334, 2008.

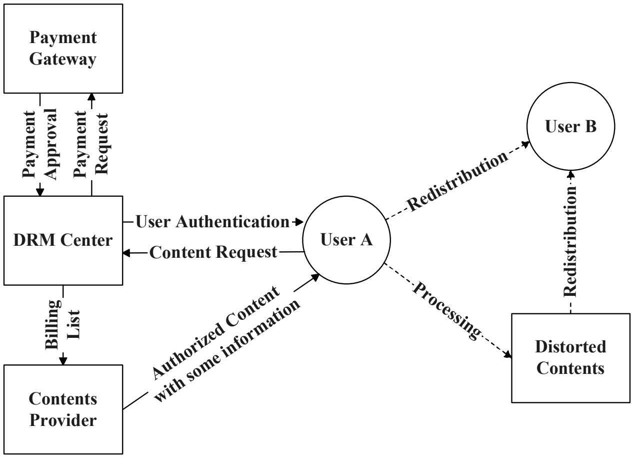


Fig. 1. A general DRM system: a solid line stands for legal flow and a dotted line stands for illegal or unauthorized flow

There are two ways to communicate between CP and the user: transmitting streaming format and using downloading service with a secure web browser. In here, watermark embedding process should be done in real-time, so watermarking algorithm should have low cost and low computational complexity. From the aspect of visual quality, the authenticated user has to want to watch it without any degradation. Moreover, the user can record it into own storages and modify for playing portable devices. In order to meet real-time, visual quality, and robustness requirements, watermark is embedded in real-time and human visual system (HVS) should be considered during watermark embedding process.

One of possible attacks in the above system is geometric transformations such as rotation, scale and translation (RST) attack. There are some watermarking methods resistant to these attacks such as embedding watermark into invariant domain [2,3], using template (template-based synchronization) [4,5], and using self-synchronizing watermark (autocorrelation) [6,7]. First of all, embedding watermark into a RST invariant domain spends considerable computing time converting into the invariant domain and extracting feature points. Next, template insertion is additional distortion to digital data because the template is intended for only synchronization. Moreover, even if the synchronization is obtained, there can be failure to correctly extract watermark. Finally, self-synchronizing watermark is specifically designed to have both synchronization and watermark information. It has periodic peaks by using autocorrelation function, so detector measures attack parameters and recovers as close as possible to the original data. However, since the detector correlates the received watermarked data with itself, computational cost is too high when a high resolution video (e.g., 1920 × 1080 or 1280 × 720) is received as an input. Thus, it is difficult to satisfy real-time embedding or extraction of watermark and simultaneously make watermark robust against the attacks.

Some distortions affecting viewing angles such as translation and rotation processing are beyond of our scope because it does not commonly happen, so we focused on downscaling attack. As it is easy for people to purchase portable electronic devices such as a personal digital assistant (PDA) and a portable multimedia player (PMP), people want to watch the downloaded video clip in their own devices by resizing the video clip and converting into other file formats using a specific tool. The downloaded video clip is originally adjust to playing on a personal computer with sufficient performance to show high quality and high resolution. However, a PDA is typically about 5 inches in height and 3 inches in width. Due to this limited size most PDAs have a small amount of memory, include slow processors, and feature small display screens. In order to play it on the PDAs it is not only required downscaling transformation but also encoded at low-bit rate. Despite these transformations embedded watermark must still have robustness. This is why downscaling attack is preferred to other geometric attacks.

In [8], they have suggested a robust video watermarking technique for downscaling attack in the compressed domain. According to them, spatial downsizing is related to the truncation of full DCT with the same geometrical size. Based on this criterion their watermarking technique is robust against downscaling attack about all DCT-based coding videos. However, if matched video encoder and decoder are available, a robust video watermarking can be accomplished in decoded video frames.

For the convenience we use an image watermarking technique and then extend it to a single decoded video frame. This paper is organized as follows. Section 2 begins by explaining a real-time video watermarking algorithm in the uncompressed domain. Experimental results demonstrate the performance in section 3. Section 4 concludes.

2 The Proposed Watermarking Scheme

Now, we introduce our watermarking scheme. For embedding watermark the main idea is to fix a size of basic pattern and scale the basic pattern with respect to a size of a host video. Then, the scaled pattern is embedded into the host video in additive way. If the watermarked video is received as an input at detector, we recover a basic pattern which has the same size as the original at the embedder by using the size of the received video. Thus, the recovered pattern is correlated with the original basic pattern to extract watermark. We explain watermark insertion and extraction in detail as follows.

2.1 Watermark Insertion

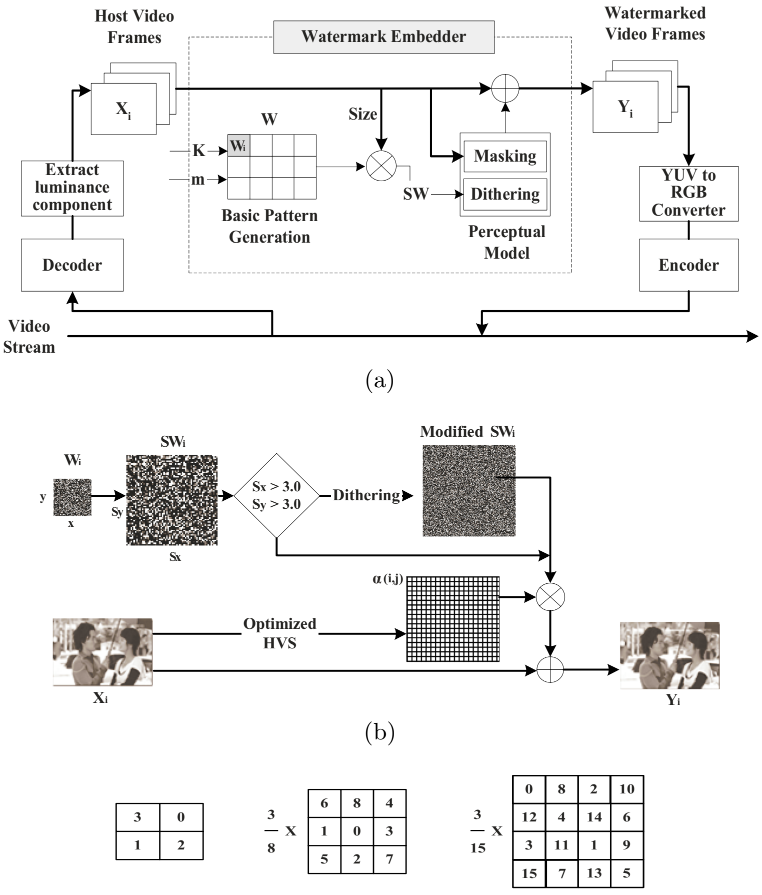
Let X_i be a set of host video frames, W_i be a watermark sequence, and Y_i be a set of watermarked video frames ($i = 1, \ldots, n$). The first step in the insertion process is basic pattern generation. This step consists of two parts: (1) a codebook generation and (2) deciding the size of basic pattern. The codebook comprises M floating random patterns of length N, where the random patterns are identically independent distribution ($i.i.d$) Gaussian pseudorandom patterns with zero mean and unit variance. The patterns are generated by private key K. So messages to hide are encoded each random pattern W_i. The size of the basic pattern is decided by how many messages are embedded into a single decoded frame and how many the encoded pattern is repeated within it. We define the length of messages as m bytes and the number of repetition as R times. Thus, we obtain the fixed size of the basic pattern $m \times N \times R$ and make it of a 2-D dimension pattern $x \times y$. To resist resizing attack, the size of the 2-D pattern should be smaller than the size of attacked videos. All variables are also known to detector.

The next step begins with acquiring the size of the host video. Then, W_i is enlarged according to width and height of the host video. That is, W_i becomes the low-frequency spatial watermark SW_i with scaled width S_x and scaled height S_y. If both S_x and S_y are bigger than 3.0, as we know, the low-frequency watermark causes blocking artifacts, so an imperceptible embedding technique should be needed. After SW_i has been generated, perceptual model is applied to both the scaled watermark pattern for imperceptible embedding and the original frame for calculating a local strength. The perceptual model that we used consists of two components: (1) variable dithering using pixel-by-pixel method and (2) the optimized noise visibility function (NVF) based on the HVS. To achieve former, SW_i is modified to decrease the blocking effect as stated [9,10]. Dithering matrix is originally used for printing on a 1-bit printer, but we utilize it to reduce the visibility (see Fig. 2(c)). Pre-defined matrices have different sizes that a value of a smooth region of 2-D scaled watermark is taken from the dithered value using large size, whereas a value of a detail region is taken using small size. For latter the spatially perceptual masking is adopted to the original frame and then we calculate local scaling factor α by pixel-by-pixel estimation. Finally, the watermarked frames Y_i are obtained by adding the dithered version of the watermarks to host video frames X_i.

For boosting detector performance the same watermark is inserted into a fixed number of t consecutive frames. Since it should need high computational costs to estimate α and add the watermark to the host frames, we utilize MMX technology to accomplish real-time insertion process [11]. Figure 2 illustrates our proposed watermark embedding scheme as mentioned above.

2.2 Watermark Extraction

In the extraction process, we use blind watermark detector performed by normalized cross correlation. The first step of the watermark extraction is base pattern generation with same secret key K at the embedder. It only includes codebook generation because the same variables as the embedder are already known to the detector. Next, the watermarked video frame $\overline{Y_i}$ is sent to a denoising filter so that we obtain the estimated watermark $\overline{W_i}$ by subtracting the estimated original video frame from $\overline{Y_i}$. Adaptive wiener filter [12] is used as the denosing filter and a 3×3 window is applied to computing mean and variance of an individual pixel.



(a)

(b)

(c)

Fig. 2. Block diagram of proposed watermark embedding scheme: (a) overview of watermark embedding into video streams, (b) the details of the watermark embedder, and (c) used 2×2, 3×3, and 4×4 matrices as dithering

$$\overline{W}(i,j) = \frac{V_{\overline{W}}(i,j)}{V_{\overline{W}}(i,j) + V_{\overline{Y}}(i,j)}[\overline{Y}(i,j) - M_{\overline{Y}}(i,j)] \tag{1}$$

Equation (1) represents how to compute the estimated watermark. $V(i,j)$ and $M(i,j)$ are the local variance and the local mean for the (i,j) location respectively. Since detector has no knowledge of probability distribution and properties of watermark pattern, we can replace $V_{\overline{W}}(i,j)$ with the mean value of $V_{\overline{Y}}(i,j)$. Although the performance of the adaptive wiener filter is lower than a few denosing filters such as phase only filter (POF) and binary phase only filter (BPOF)[13], it is sufficient to our system. To be robust against various attacks and enhance the watermark energy, we accumulate each extracted noise-like signal from the

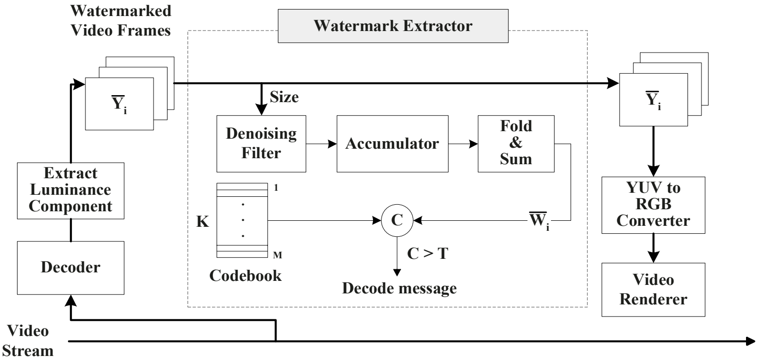


Fig. 3. Block diagram of proposed watermark extraction scheme

correspondence frame during t consecutive frames and the estimated watermark is constructed simultaneously by folding and summing every t frames. Finally, watermark can be extracted by cross correlating it with the codebook. If estimated correlation value C exceeds threshold T, there is a watermark and then decode the hidden message. Otherwise the detector is failed to extract. This means that the detector has no guarantee its reliability under false positive error rate f_p. T is slightly different every t frames because our detection approach is based on f_p. For example, T under the fixed $f_p = 10^{-6}$ is given by

$$f_p = \frac{1}{2} \cdot erfc(\frac{T - mean}{\sqrt{2 \cdot var}}) = 10^{-6} \tag{2}$$
$$T \geq mean + 3.36\sqrt{2 \cdot var}$$

where T can be computed by means of the mean and the variance of the estimated watermark. The watermark extraction process is depicted in Fig. 3.

2.3 Implementation of Directshow Filter for Practical Use

Directshow is composed of two objects, filters and filter graphs. The filter graph is a collection of filters and consists of three kinds of filters such as source filters, transform filters, and renderer filters. Among them the transform filter is commonly used to handle tasks such as data compression and decompression, video decoding, stream splitting, and so on. So, it is possible for desirable results or effects to add specific operations to the filter graph. Also, each filter of the filter graph is derived from a modular design; it hides its internal operation and we don't need to understand complicated interfaces between filters. Using these features both embedder and detector are developed as transform filters. This means that some applications built with directshow can make use of it to achieve watermark insertion and detection process. In our case, watermark embedder and detection filters are connected after a video decoder within our MPEG video player.

Fig. 4. Snapshot examples of test videos

3 Experimental Results

Our experimental system is composed of an Intel Pentium IV CPU with a 3.6Ghz core, 2 GB DDR2, and ATI X1600 with 256 MB graphics memory. The host videos we have tested are 3 categories of video clips with length 17 seconds at 30 frame per second (fps): (1) TV drama, (2) documentary, and (3) music show as depicted in Fig. 4. All measured correlation values in this section average three of them. The payload of watermark is totally 192 bits (24 bytes) and same watermark is repeated 2 times in whole frames. Prior to the watermark insertion and extraction we have set each variable, $N = 1024$, $m = 2$ bytes, $R = 12$ times, $t = 30$ frames, and $f_p = 10^{-6}$. Thus, the size of 2-D basic pattern becomes dimensions 256×96.

Table 1 reveals that we accomplish real-time processing with our MPEG video player for decoding a video frame, watermark insertion or extraction, and displaying video compared to the previous scheme. Note that real-time processing means that for a 30fps video a single frame is decoded, processed, and displayed on a screen in around 0.03 sec. Also, an average PSNR value after embedding is around 45dB for all kinds of test videos. Results show that our watermarking

Table 1. Execution time and PSNR value comparison between the proposed watermarking scheme and Kang et al. [11]. This table shows a performance of our MPEG video player with insertion and extraction when an input video is 1920 × 1080 HD MPEG-2.

[sec.]

Operations	Kang et al.[11]	Proposed
Frame decoding	0.01317	0.01317
Watermark insertion / extraction	0.011 / 0.004	0.011 / 0.004
Displaying video	0.00481	0.00481
Total insertion time / extraction time	0.029 / 0.022	0.029 / 0.022
Average PSNR	43dB	45dB

Table 2. Normalized correlation value comparisons between the proposed watermarking scheme and Kang et al. [11]. After various attacks, watermark is survived with the following values.

	Kang et al. [11]	Proposed
Original(1920×1080)	0.71	0.67
Resize to VGA	0.46	0.41
Cropping	0.60	0.56
Frame rate change (30fps → 24fps)	0.60	0.56
Color conversion (RGB to Gray)	0.58	0.55
MPEG-1 conversion	0.53	0.49
MPEG-4 conversion	0.47	0.44

scheme can produce high-fidelity watermarked videos. Besides the analysis of execution time and visual quality, correlation values after various attacks are shown in Table 2 when the 1920 × 1080 HD MPEG-2 watermarked video is received as an input. The correlation differences between two schemes are approximately 0.04 due to adding dithered values instead of watermark values itself for visual quality. However, our watermark is also survived with the acceptable correlation values as maintaining visual quality.

In order to prove our robustness to downscaling attack we have constrained 6 host video sizes for simplicity, e.g., $1920 \times 1080, 1280 \times 720, 720 \times 480, 640 \times 480, 352 \times 288$, and 320×240. We remind that the detector has no information about the original size of an input video. From Fig. 5 to Fig. 10 shows the downscaling performance of the system; x-axis stands for the resized video from the watermarked video and y-axis stands for normalized cross correlation. The detection threshold is normally estimated 0.13 subject to $f_p = 10^{-6}$. Although some values are below the threshold on the spot of a second, the whole messages

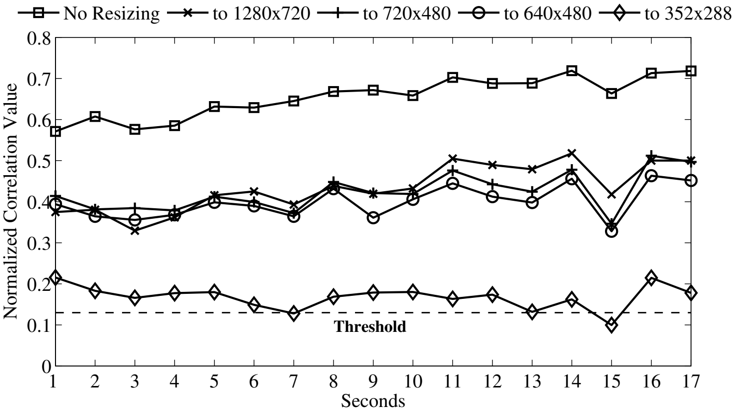


Fig. 5. Performance against downscaling attack (Input Video: 1920 × 1080)

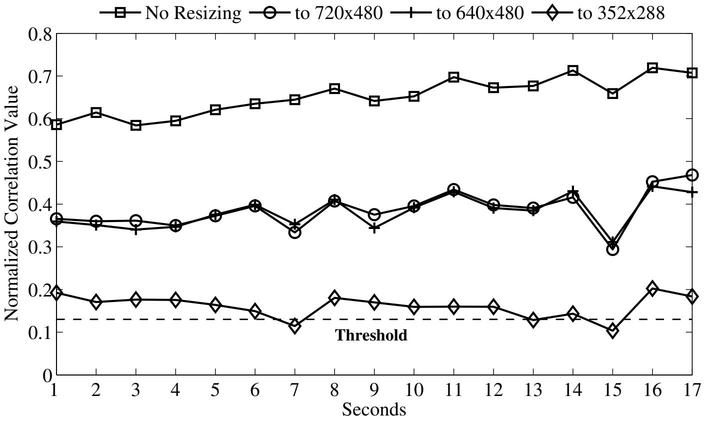


Fig. 6. Performance against downscaling attack (Input Video: 1280 × 720)

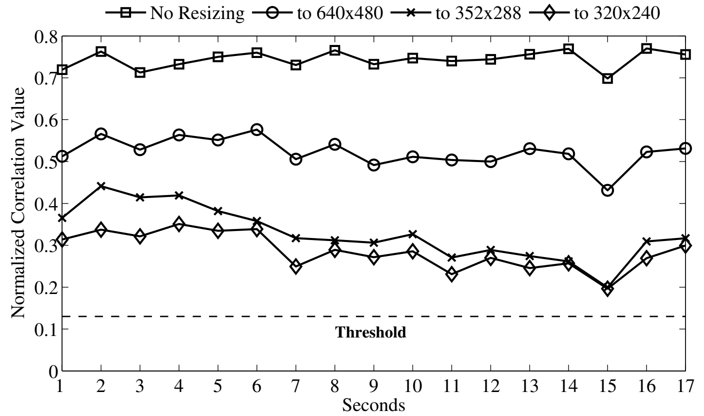


Fig. 7. Performance against downscaling attack (Input Video: 720 × 480)

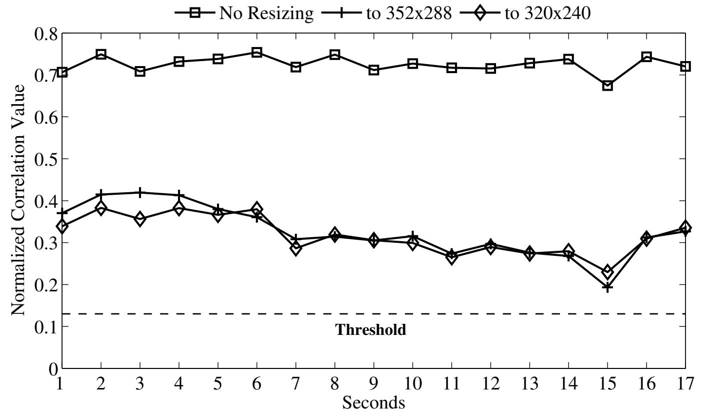


Fig. 8. Performance against downscaling attack (Input Video: 640 × 480)

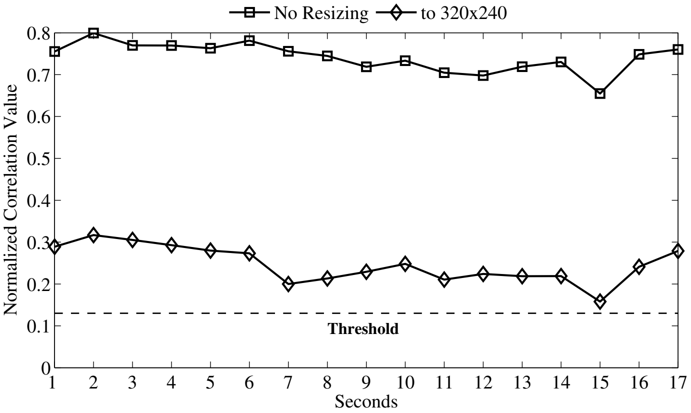


Fig. 9. Performance against downscaling attack (Input Video: 352 × 288)

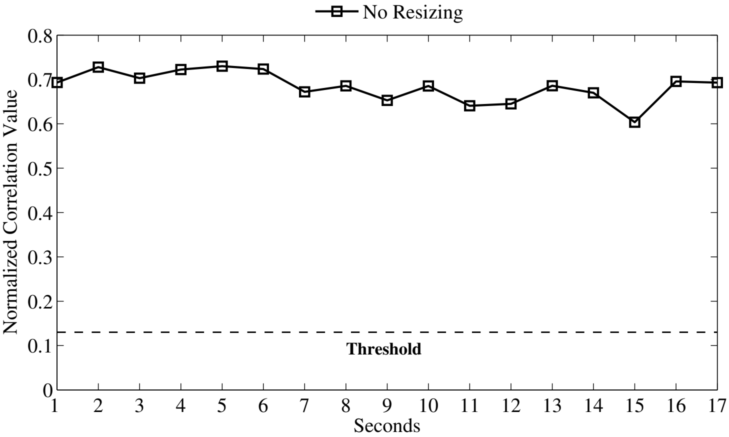


Fig. 10. Performance against downscaling attack (Input Video: 320 × 240)

are correctly extracted since the repetition time of same watermark is two. If the length of video is more longer than the tested video, it can carry out better results.

Besides downscaling attack, for each video size the scheme is also robust against various attacks such as cropping, format conversion, frame rate change, color-to-gray conversion attack as shown in Table 3. All results indicate that not only the proposed scheme has good robustness against downscaling attack and other malicious attacks but also it allows us to be confident that the specified error rate will not be exceeded.

Table 3. Summarization of robustness to other attacks (threshold = 0.13)

Input video	MPEG-1	MPEG-4	Frame rate change	Color conversion
1280 × 720	0.42	0.35	0.51	0.51
720 × 480	0.51	0.42	0.61	0.62
640 × 480	0.44	0.41	0.59	0.59
352 × 288	0.39	0.38	0.55	0.54
320 × 240	0.35	0.35	0.51	0.49

4 Conclusions

In this paper, we present a practical real-time video watermarking scheme for downscaling attack. We mainly focus on the robustness to downscaling attack because more portable devices spread out, more downscaled and distorted video data are in public. In addition to this attack, the proposed scheme is also robust against not only spatial attacks such as cropping, format conversion, and color-to-gray conversion but also temporal attacks such as frame rate change. Both watermark insertion and the extraction are done in uncompressed domain, so the proposed scheme can be straightforwardly applied to all video coding if there are corresponding video coder and decoder. Moreover, using the beneficial features of directshow our scheme is suitable for directshow built-in applications.

Acknowledgement

This work was in part supported by the KOSEF grant NRL program funded by the Korea government(MOST) (No. R0A-2007-000-20023-0), and the IT R&D program of MIC/IITA(2007-S017-01, Development of user-centric contents protection and distribution technology).

References

1. Sencar, T., Ramkumar, M., Akansu, N.: Data Hiding Fundamentals and Application. Elsevier, Oxford (2004)
2. Kim, H.Y., Baek, Y.J., Lee, H.K.: Rotation, scale, and translation invariant watermark using higher order spectra. The Journal of SPIE 42, 340–349 (2003)
3. Lee, H.Y., Kim, H.S., Lee, H.K.: Robust image watermarking using invariant features. The Journal of SPIE 45, 1–11 (2006)
4. Delannay, D., Macq, B.: A method for hiding synchronization marks in scale and rotation resilient watermarking schemes. In: Security and Watermarking of Multimedia Contents IV. Proceedings of SPIE, vol. 4675, pp. 548–554 (2002)
5. Pereira, S., Ruanaidh, J., Deguillaume, F., Csurka, G., Pun, T.: Template-based recovery of fourier-based watermarks using log-polar and log-log maps. In: International Conference Multimedia Computing and Systems, pp. 870–874. IEEE CS Press, Los Alamitos (1999)

6. Lee, C.H., Lee, H.K.: Improved autocorrelation function based watermarking with side information. The Journal of SPIE 14, 1–13 (2005)
7. Kutter, M.: Watermarking resisting to translation, rotation, and scaling. In: Security and Watermarking of Multimedia Contents. Proceedings of SPIE, vol. 3657, pp. 423–431 (1999)
8. Wang, Y., Pearmain, A.: Blind mpeg-2 video watermarking robust against geometric attacks: a set of approaches in dct domain. IEEE Transaction on Image Processing 15, 1536–1543 (2006)
9. Park, J.W., Lee, C.H., Lee, H.K.: A visual quality enhancing method for low frequency. In: Proceeding of International Workshop on Image Analysis for Multimedia Interactive Services, IEE&EURASIP, pp. 49–52 (2006)
10. Hel-Or, Z., Zhang, M., Wandell, A.: Adaptive clust dot dithering. The Journal of SPIE 8, 133–144 (1999)
11. Kang, I.K., Im, D.H., Lee, H.K.: Implementation of real-time watermarking scheme for high quality video. In: Proceeding of the ACM Multimedia and Security Workshop, pp. 124–129 (2006)
12. Karybali, I., Berberidis, K.: Efficient spatial image watermarking via new perceptual masking and blind detection scheme. IEEE Transaction on Information Forensics and Security 1, 256–274 (2006)
13. Liu, Y., Zhao, J.: A new filtering method for rst invariant image watermarking. In: The 2nd IEEE Internatioal Workshop, pp. 101–106 (2003)

Secure Video Multicast Based on Desynchronized Fingerprint and Partial Encryption

Zhongxuan Liu[1], Shiguo Lian[1], Josselin Gautier[1,2], Ronggang Wang[1], Zhen Ren[1], and Haila Wang[1]

[1] France Telecom R & D Beijing, Beijing, 100080
zhongxuan.liu@orange-ftgroup.com
[2] University of Nantes Polytech'Nantes School, France

Abstract. Recently, a collusion resilient video fingerprint method has been proposed utilizing desynchronization. But a question has not been considered for this method: secure transmission for multicast / broadcast. Because every copies using this method have different desynchronization, all the traditional fingerprint broadcasting techniques do not work. In this paper, we propose a method for securely broadcasting desynchronized video copies. The video is firstly lengthened and partially encrypted in the sender side and broadcasted to every user, and then on the receiver side, the video is desynchronized and embedded by fingerprint, and then decrypted. We give a method to implement this for MPEG-2 stream. Experiments indicate the effectiveness of the fingerprinted copies for collusion robustness and the efficiency of transportation. Security of the system is also analyzed with probable attacks and corresponding solution is proposed.

1 Introduction

The fast development of digital techniques not only improves the convenience of transmitting digital media data, but also increases the menace of illegal redistribution of multimedia objects especially films and TV. Video fingerprint is the technique to indicate user of the media by embedding the user's information imperceptibly. Then the copies received by different users will be visually the same while in fact different copies. When the media is illegally redistributed, the information in the media is used to identify the illegal users.

There're two main problems for video fingerprint systems [1]. First, there's a serious attack called collusion attack for fingerprint which combines several copies of the same content but with different fingerprints to make fingerprint hard to extract. The main collusion attacks [2][3] include linear collusion (averaging and cut-and-paste), nonlinear collusion (minimum / maximum / median / minmax / modified negative / randomized negative attacks) and Linear Combination Collusion Attack (LCCA). Traditional collusion resilient fingerprint

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 335–349, 2008.

methods are mainly two classes (we call these methods OEDF–Only Embedding Different Fingerprints): Orthogonal fingerprinting [4][5] makes fingerprint orthogonal to each other utilizing orthogonal random signals of uniform or Gaussian distribution and ensures the colluded copy still have detectable fingerprint. Shortcomings of orthogonal fingerprint are the high cost on fingerprint detection and the limitation in customer and colluder population [2] (more customers and colluders will enhance the false match rate); Another class of methods are coded fingerprints which carefully design the fingerprint in codeword that can detect the colluders partially or completely. The Boneh-Shaw scheme [6][7], Tardos scheme [8], IPP (identifiable parent property) code [9] and the Wu's scheme [2] belong to this method. The Boneh-Shaw and Tardos schemes are not so fitful to image / video fingerprint as Wu's scheme (for example, it only considers bit switching while averaging is more probable for image / video fingerprint collusion) and has long code ($O(\log^4 N \log^2(1/\epsilon))$, $O(c^2 \log \frac{N}{\varepsilon})$ respectively). Another problem of these schemes is that they can only ensure the caught user is colluder in the sense of probability. IPP can ensure the caught user is colluder, but it does not resist average and nonlinear collusion attack because the attack model of IPP is that the colluders are not allowed to output any other symbols than the ones they see on the detected positions. The Wu's method also has three major problems: hard to form code supporting large number of users and colluders; low embedding efficiency because each bit needs a orthogonal signal; suffering LCCA attack [3]. Although some methods are proposed to solve some of these problems [10] [11] [12], none of these methods can solve all the problems.

Another technique named as Both Desynchronizing and Embedding Fingerprints (BDEF) is proposed by [13] and first introduced for video by [14]. By desynchronizing the carrier, the BDEF degrades the quality of the colluded copy seriously. Then the difficult problem of devising and embedding collusion robust fingerprint for identifying colluders is avoided. The scheme in [13] employs pseudo-random warping for images. In that case, since each user gets a differently randomly warped copy, it is claimed that it is hard to perform a collusion attack using warped copies without causing perceptual artifacts. Mao proposed the desynchronizing fingerprint method for raw video [14]. 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. This method has the shortcoming of high computing complexity. A scheme of embedding desynchronized fingerprint for compressed image is proposed in [15] where a metric for degradation of desynchronization and colluding desynchronized copies is given. In [16], a technique of embedding desynchronized fingerprint when compressing video is proposed. This method has the priority for both time and memory efficiency compared with Mao's method because it utilizes the motion estimation of compression for frame interpolation. An initial security analysis is also given in [16] including proposing the re-synchronization and re-desynchronization attacks with probable solution.

Second, distributing fingerprinted video copies to multiple users (broadcast / multicast) is difficult. Because of the difference between different fingerprinted copies, it is not practical to transmit copies individually considering the bandwidth consumption especially for streaming applications. Then broadcasting fingerprinted copies is becoming hot research topic. In [1], a joint fingerprint design and distribution scheme to multicast shared fingerprinted coefficients to the users in the subgroup. In the scheme, higher bandwidth efficiency and collusion resistance ability are achieved. In [17], a system capable of watermarking MPEG streaming videos using standard streaming protocols such as RTSP / RTCP / RTP is proposed. A proxy server is included between the video server and clients. In [18], joint source fingerprinting (JSF) which jointly achieves fingerprint design and media design is proposed. Given the semantic representation which is "coarse" representation of media and feature representation which is supplementary to semantic representation, fingerprints can be created from the feature class. In [19], two uniquely fingerprinted copies were generated, encrypted and multicasted, where each frame in the two copies was encrypted with a unique key. Each user was given a unique set of keys for decryption and reconstructed a unique sequence. In [20], a hierarchy of trusted intermediaries was introduced in the network. All intermediaries embedded their unique IDs as fingerprints into the content as they forwarded the packets through the network, and a user was identified by all the IDs of the intermediaries that were embedded in his received copy. In [21], for each fingerprinted copy, a small portion of the MPEG stream was encrypted and unicasted to the corresponding user, and the rest was multicasted to all users to achieve the bandwidth efficiency. In [23], the content owner encrypted the extracted features from the host signal with a secret Ks known to the content owner only, multicasted the encrypted content to all users, and transmitted to each user I a unique decryption key $K_i \neq K_s$. The fingerprint information is essentially the asymmetric key pair (K_i, K_s).

Although there have been quite a few methods for boradcasting fingerprinted copies as above and some initial research for BDEF which has priorities of collusion robustness and embedding efficiency, a question has not been considered – how to Broadcast Desynchronized Fingerprint Copies (BDFC). The central questions for BDFC are two aspects: how to implement the task of one copy sent from sender side with different copies obtained in user side, and how to securely embed fingerprint in user side. For this problem, our solution is as follows: at sender side, the video is lengthened and partially encrypted; at receiver side, the video is desynchronized and embedded by fingerprint, and then decrypted. This method is called the Broadcasting Desynchronized Fingerprint copies based on Partial encryption (BDFP) which accomplishes the following objectives (see Section 2): For security, encrypted content should be sent to receivers; For efficiency of broadcasting, same content (except little amount of different information for users such as decryption key or others) should be sent to users; For preventing videos from plain-text leak, decryption process should not be done before fingerprint embedding process.

The left part of the paper is composed of following sections. The new system of broadcasting desynchronzied videos is presented in Section 2 followed by performance analysis in Section 3. Conclusion and future work is in Section 4.

2 Broadcasting Desynchronized and Fingerprinted Copies Based on Partial Encryption (BDFP)

2.1 Illustration of Time Desynrhonization

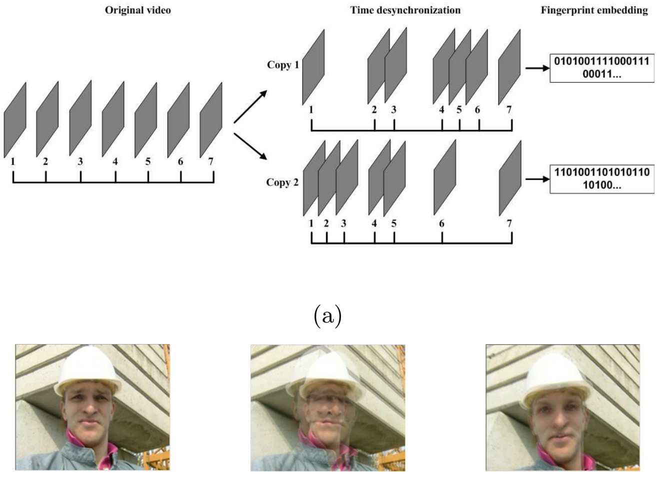
Time desynchronization (TD) is the method to modify different copies of a video by changing the time samples of the frames (see Fig. 1). In Fig. 1(a), the time samples of the original video are changed by different forms to make different copies (such as Copy1 and Copy2) and then different fingerprints are embedded into the copies. If some of the copies are used to collude, the resulted copy will have degradation (see Fig. 1(c) and 1(d)) compared with frames of the original copy (see Fig. 1(b)). There have been some methods for TD including the method forming new frames by estimating motion vectors utilizing optical flow algorithm for raw video [14] and the method forming new copies by utilizing the motion vectors in compression process [15]. The former method has the priority of less artifacts for formed frames but the later method has better tradeoff between less degradation and higher efficiency.

2.2 Elementary Architecture of Our Scheme

Our proposed technique of BDFP is illustrated in Fig. 2. The processing is in the following:

At sender side,

1. The original video is lengthened by inserting some frames;
2. Then the lengthened video is encrypted during compressing process for broadcasting.



(a)

(b) (c) (d)

Fig. 1. (a): Illustration for time desynchronization; (b)(c)(d) are respectively one frame of original video and two frames of colluded video copies after time desynchronization

At receiver side,

1. The encrypted video is desynchronized by skipping different groups of frames for different copies;
2. Then fingerprint is embedded followed by decryption;
3. After being compressed, the copies are displayed with fingerprint for tracing pirates.

The above process completes two functions: broadcasting desynchronized copies and securely embedding fingerprint at user side. In the following two subsections we explain why our BDFP scheme have the two functions.

2.3 Broadcasting Desynchronized Copies

We implement the BDFP for MPEG-2 standard. Until now, MPEG-2 is still the most widely used video coding standard applied in DVD, DVB et al. MPEG-2 video include multiple GOP (Group Of Pictures) which includes multiple frames. There're three classes of frames in MPEG-2 video: I frame (use intra-frame encoding and can be independently decoded), P frame (decoded referencing former frames), B frame (decoded referencing forward and backward I or P frames). Here I and P frames are called referenced frames because they can be used for compensating P and B frames, B frames are called un-referenced frames because they can not be used for compensating.

Considering the different types of frames in MPEG-2 video, the function of broadcasting desynchronized copies is implemented as follows (see Fig. 2):

1. At sender side, the original video copy (OC) is lengthened to form lengthened copy (LC) by regularly or randomly inserting frames (assuming M frames are inserted);

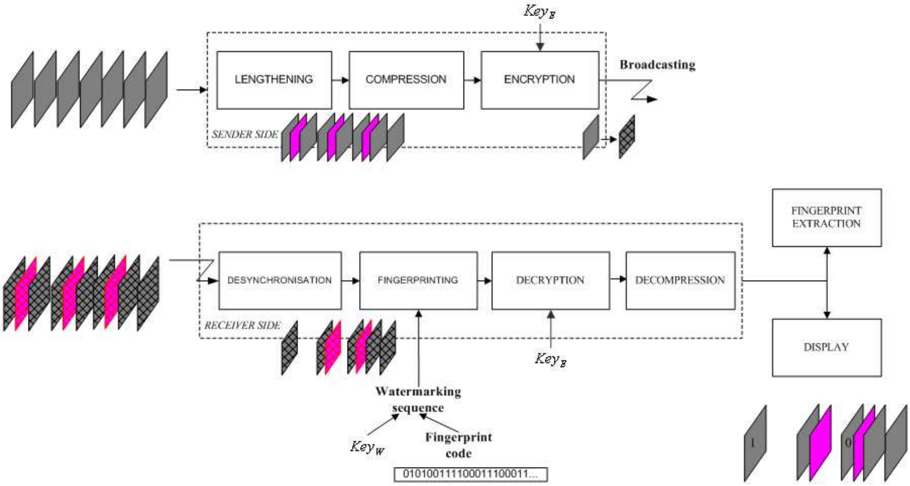


Fig. 2. Our scheme of BDFP

2. At user side, M B frames are skipped because B frames are not referenced frames. Different skipped B frames controlled by different keys make different copies (see the Copy 1 and Copy 2 in Fig. 2).

Remarks: The reason why only B frames are used for repeating and skipping is two aspects: firstly, inserting B frames only needs repeating motion vectors and residue of B frames because the repeated B frame use the same reference frames with its former B frame; secondly, skipping B frames will not influence decoding of other frames because B frames are not referenced.

2.4 Embedding Fingerprint in the Partial Encrypted Video (EFPEV)

Because video encryption and video fingerprint realize different functionalities, they can be combined together to protect both the confidentiality and the identification. In our system, fingerprint is embedded into encrypted video for security (plain text without fingerprint will not exist, and the watermark embedder embeds watermarks into the encrypted media data directly without knowing the decryption key, which avoids the leakage of media content). For this question, some elementary methods have been proposed [24][25].

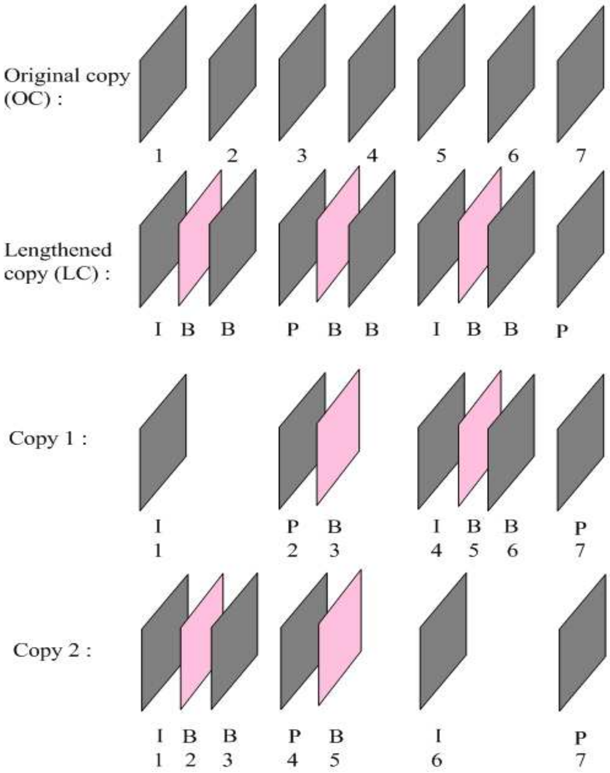


Fig. 3. Principal of broadcasting desynchronized copies

In this paper, media data X is partially encrypted in the sender side and fingerprinted in the receiver side. Set X be composed of independent parameters Y and Z. Among them, Y is encrypted, and Z is fingerprinted. The process is defined as

$$\begin{cases} Y' = \quad E(Y, K_e) \\ Z' = W(Z, B, K_F) \end{cases} \tag{1}$$

Here, $Y', K_e, E(), Z', B, K_F, W()$ are the encrypted copy of Y, encryption key, encryption algorithm, fingerprinted copy of Z, fingerprint, fingerprint key and fingerprint algorithm, respectively. The produced media data X' is composed of Y' and Z'. Fingerprint can be embedded into Z' without decrypting Y'. Based on this architecture, we propose the scheme combined with MPEG2 codec, which encrypts and marks suitable MPEG2 parameters independently. The scheme is composed of several components: the compression component, encryption component and watermarking component. The encryption process and fingerprinting process are controlled by different keys.

The compression component includes intra-prediction, inter-prediction, Variable Length Coding (VLC), etc. During the process, at sender side, the video is encrypted by encrypting the DC coefficients difference for the signs (by adding a random integer sequence and computing the last bit) and values (by module adding a random integer sequence) for intensity and signs for color and also the motion vector signs. The fingerprint is embedded in the AC coefficients of I frames (this is for convenience of extraction considering some B frames being skipped, see Section 3.3).

At the sender side, the video is compressed by MPEG2 encoder including following processes

1. Some B frames are duplicated by rewriting the B frames information;
2. Signs and value of DC of I frames and signs of motion vectors of P/B frames are encrypted controlled by key_E.

Note: Because duplicated B frames use the same reference frames (in MPEG2, there's no referenced B frames which is not the case for H.264), then rewiting the B frame information into stream is enough.

At the receiver side, the video is decompressed by MPEG2 decoder including following processes

1. Some B frames are skipped by directly finding the next frame start code;
2. For I frames, AC coefficients are embedded by fingerprint [26] controlled by K_w.
3. Signs and values of DC and AC of I frames and motion vector signs of P/B frames are decrypted controlled by key_E.

Note: The number of total skipped frames should be near/similar to the inserted frames to ensure the video/audio synchronization. When an illegally distributed copy is found, every frame of the copy is compressed by the form for I frame and if fingerprint can be extracted, the extracted bits are added to the fingerprint code.

When a copy is illegally redistributed, the fingerprint can be extracted from the AC coefficients of the copy's I frames [26]. If several copies are colluded, the desynchronization between the copies will prevent the attack from obtaining copy with acceptable quality (see Section 3.1).

3 Performance Analysis

3.1 Security of BDFP against Collusion Attack

The security of our BDFP scheme against collusion is for two aspects: Firstly, the security against simple collusion attack (such as linear collusion, nonlinear collusion and Linear Combination Collusion Attack (LCCA)); Secondly, the security against resynchronization attack (several copies are processed to be synchronizing with a same copy and then these copies are directly colluded).

Simple Collusion Attack (SCA). When the traitors obtained several copies of raw video embedded by DVF, the easiest method for the traitors to collude is simply averaging the copies. The following is the analysis to the security of BDFP for this attack.

The ability of Desynchronized Video Fingerprint (DVF) to resist collusion depends on the different desynchronization forms of different copies. The space of the different forms is very large. For example, for a four seconds video (30 frames/s, 2 B frames between neighbor I/P frames, 1/10 B frames ($4 \times 30 \times 2/3 \times 1/10 = 8$) are duplicated), the number of different forms is $\mathrm{C}^{8}_{121+8-41} \approx 2^{36}$.

When copies with different desynchronization forms are combined to collude, the security of BDFP against SCA is the degradation caused by SCA with the degradation by desynchronization being lower than a certain constraint. With the degradation by desynchronization constrained in a certain range, degradation by SCA should be increased.

The colluded effects of two BDFP copies are displayed in Fig. 4. Horizontal axis is the frame number of the copy, vertical axis is the copy's corresponding frames in the original copy. "Copy 1" and "Copy 2" are the corresponding frame numbers of two BDFP copies; "TimeDif" is the difference of the two copies' corresponding frames; "CopyLen" is the time samples of lengthened copy. Three frames of the colluded result of the two BDFP copies are shown in Fig. 5. The degradation of the frames are uniform to the time difference indicated in Fig. 4: frame 60 and 51 has the largest and the least value, and frame 90 has the middle value.

To analyze security of BDFP against SCA theoretically, we consider the mathematical model in the following.

Definition 1. *$i_k (k \in [1, K])$ are the copies attending collusion. The metric for collusion degradation of time desynchronized copies is:*

$$DegTC_K(a^{i_1}, a^{i_2}, ..., a^{i_K}) = \frac{1}{K} \sum_{n=1}^{N} \sum_{k=1}^{K} |a_n^{i_k} - \frac{1}{K} \sum_{j=1}^{K} a_n^{i_j}|.$$

where a^{i_k} are the time samples of the kth copy.

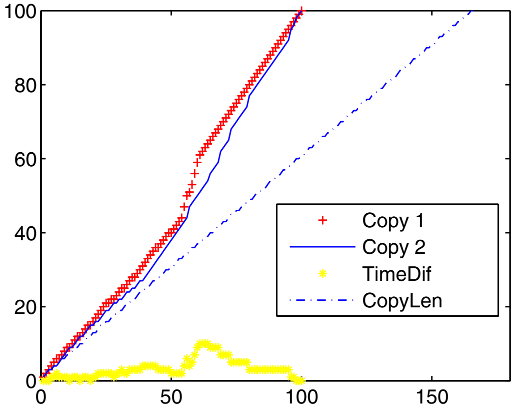


Fig. 4. Illustration of the collusion of two BDFP copies. "Copy 1" and "Copy 2" are the two copies' corresponding frame numbers in original copy; "TimeDif" (vertical axis) is the difference of the two copies' corresponding frame numbers; "CopyLen" (horizontal axis) is the corresponding frames of lengthened copy.

Fig. 5. Left image: Frame 51; Middle image: Frame 60; Right image: Frame 90

When $K = 2$, this is equivalent to

$$DegTC_2(a^{i_1}, a^{i_2}) = \sum_{n=1}^{N} |a_n^{i_1} - a_n^{i_2}|.$$

Theorem 1 (Expectation of degradation when only one duplicated frame)
Assume the original video has N frames and the frame M is duplicated. Two copies are made by randomly skipping one of the $N+1$ frames. Then the expectation of the degradation by colluding the two copies is $\frac{3M(M-1)+3(N-M)(N-M+1)+N^3-N}{3N^2}$.

Proof

$$E\{DegTC_2\} = \frac{1}{N^2} \sum_{m=1}^{N+1} \sum_{n=1}^{N+1} D_{mn}$$

where

$$D_{mn} = \begin{cases} abs(m-n) & 1 \le m \le M \text{and} 1 \le n \le M \\ abs(m-n)-1 & 1 \le m \le M \text{and} M+1 \le n \le N+1 \\ abs(m-n)-1 & M+1 \le m \le N+1 \text{and} 1 \le n \le M \\ abs(m-n) & (M+1 \le m \le N+1 \text{and} M+1 \le n \le N+1) \end{cases}$$

is the time difference of copy 1 (skipping frame m from seq) and copy 2 (skipping frame n from seq), seq is the sequence of inserting frame M in the original video. Then

$$\begin{aligned} E\{DegTC_2\} &= \frac{1}{N^2}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1} D_{mn} \\ &= \frac{1}{N^2}(2(S_{M-1}+S_{N-M})+2T_{N-1}) \\ &= \frac{3M(M-1)+3(N-M)(N-M+1)+N^3-N}{3N^2} \\ &= \frac{2(M-\frac{N+1}{2})^2+\frac{N^2}{2}+\frac{N^3-N}{3}}{N^2} \end{aligned} \qquad \square$$

Then the average degradation of every frame is

$$\frac{3M(M-1)+3(N-M)(N-M+1)+N^3-N}{3N^2 \cdot N} \approx \frac{1}{3}.$$

$argmin_M E\{DegTC_2\} = \frac{N+1}{2}$, $min_M E\{DegTC_2\} = \frac{2N^2+3N-2}{6N}$;
$argmax_M E\{DegTC_2\} = \{1, N\}$, $min_M E\{DegTC_2\} = \frac{2N^3+6N^2-8N+3}{6N^2}$.

Theorem also shows degradation by duplicating the former or later frame will be larger than that by duplicating middle frame.

Remarks: For convenience of analysis, here we assume all frames can be skipped (in practice, I or P frames can not be skipped because they will influence other referenced frames). In the future, we'll consider the case that number of skipped frame larger than 1.

Resynchronization Attack (RA). For RA, there're two types of attacks according to attackers:

1. The decompressed video copies are used for RA ($RA^{\mathbf{I}}$);
2. The decrypted by still compressed video copies are used for RA ($RA^{\mathbf{II}}$).

a. Security against $RA^{\mathbf{I}}$

There're several $RA^{\mathbf{I}}$ attacks. Assume the copies attending collusion are C_i ($i = 1, ..., N$) and the watermark in I frame will not give information to attackers.

The searching space for finding MSF is too large considering the desynchronization. For example, 10 seconds video for a one hour video (30 frames/sec) is

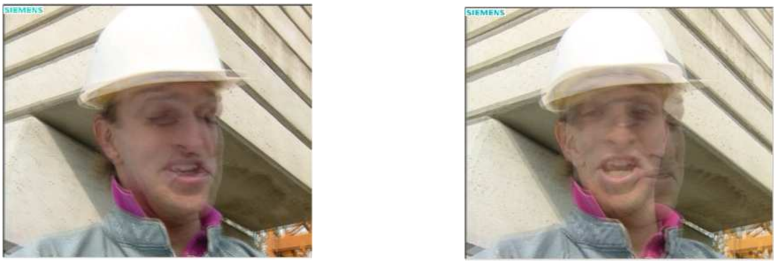


Fig. 6. Left image: colluded result after RA; Right image: colluded result without RA

searched for the N copies attending the collusion, $RA^{\mathbf{I}}$ needs $30 \times 10 \times 10800 \times (N-1) \approx (N-1)2^{22}$ frame comparisons for forming the $N-1$ resynchronized copies and colludes them with the reference copy. Not only this, because some frames have no corresponding frames and the I frames with different fingerprint bit, there will be degradation after resynchronized frame collusion (see Fig. 6).

b. Security against $RA^{\mathbf{II}}$

$RA^{\mathbf{II}}$ is a more serious menace to BDFP. Because positions of I/P frames are known, resynchronization is easier. For example, also for a one hour video (30 frames/sec), assume original video has 3 frames between neighbor I/P frames. After frame inserting, at most 6 frames are between neighbor I/P frames. Then times of matching frames decrease to be less than $8 * 10800 * 3/4 * (N-1) \approx (N-1)2^{16}$. What is more serious is that with help of fixed I/P frames, the matching will be more accurate, and watermark in I frames will not enhance the difficulty of finding matching frames. For this question, the technique like the Certified Output Protection Protocol (COPP) can be used to prevent attackers from knowing positions of I/P frames.

3.2 Perception Security and Watermark Degradation

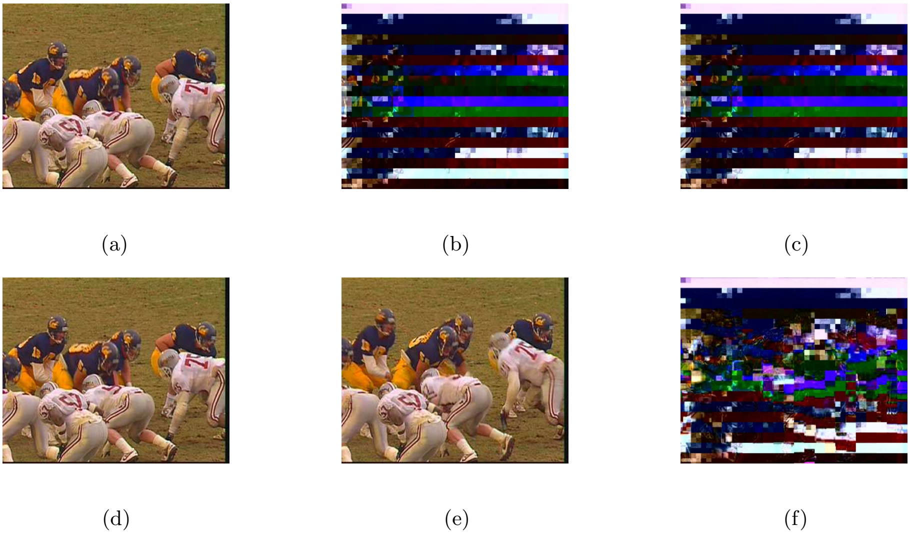
For video encryption, it is important to keep the encrypted video unintelligible, which is regarded as perception security [24]. It depends on the encryption scheme's properties. The proposed scheme encrypts both I frames and motion vectors, which keeps perception security. Another requirement of our method is limiting the degradation introduced by watermark. The perceptual security and watermark degradation is shown in Figure 5. 6(a) is an original I frame. 6(b) and 5(c) is the encrypted I frame without and with watermark embedded. The 5(d) is the decrypted I frame with watermark left. The 5(e) and 5(f) are unencrypted and encrypted B frame respectively. From the figure 5, it is shown that our method can ensure the perceptual security after encryption and also the degradation by watermark is hard to perceive.

3.3 Time and Compression Efficiency

Compared with sending one stream at sender side, the only extra computing is duplicating B frame and encryption process. The time costs of both the

Table 1. Comparison of time cost of original encoder and our method (with encryption and B frame duplication (50 frames))

	Original time (ms)	Time of our algorithm (ms)	Ratio (%)
$football_q$	9563	9824	2.73
$foreman_q$	5081	5392	6.12
$paris_q$	4481	4751	6.03
$football_c$	40092	40502	1.02
$foreman_c$	26418	26803	1.46
$paris_c$	18996	19467	2.48


(a) (b) (c)

(d) (e) (f)

Fig. 7. 'football' (cif): (a) Original I frame; (b) Encrypted I frame; (c) Encrypted I frame with watermark; (d) Decrypted I frame with watermark; (e) Original B frame; (f) Encrypted B frame

encryption and duplicating B frames are very low (see Table 1), in fact, only less than 7 percents of more time cost especially for higher definition video (less than 3 percents for CIF). The ratio of extra time cost is lower for fast moving video (such as 'football') because of the more time cost on motion estimation.

Because the encryption method we adopt in this paper will not influence the compression efficiency, we only show the result of data quantity after different ratio of duplicated B frames (see Fig. 7). In practical applications, less than 10 percents of duplicated B frames is enough for desynchronization, which means average 2.99 percents of more data (see Table 2).

Table 2. Comparison of change of data with and without 1/10 B frames duplication (50 frames)

	Original data (kbits)	Data processed by our algorithm (kbits)	Ratio (%)
$football_q$	814	841	3.32
$foreman_q$	813	841	3.44
$paris_q$	813	839	3.20
$football_c$	814	839	3.07
$foreman_c$	814	836	2.70
$paris_c$	814	832	2.21

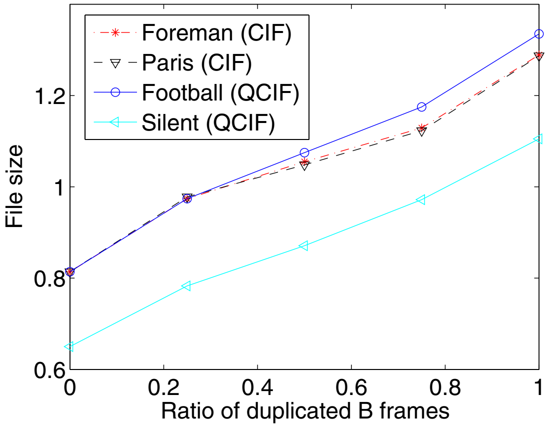


Fig. 8. Size of file (megabits) for different portions of duplicated B frames

4 Conclusion and Future Work

In this paper, we proposed a technique for broadcasting desynchronized and fingerprinted copies. At server side, some B frames are duplicated and all the frames are partially encrypted. Then fingerprint is embedded into the partially encrypted copies at receiver side. After decryption every user will get the content with desynchronized and fingerprinted copies with collusion resilience. The performance of the technique is concretely discussed and validated by experiments including the security against non-collusion attack and collusion attack, the perceptual security and time and compression efficiency. In the future, we'll research for applying BDFP technique to other codec such as H.264.

References

1. Zhao, H.V., Liu, K.J.R.: Fingerprint Multicast in Secure Video Streaming. IEEE Trans. on Image Processing 15(1), 12–29 (2006)
2. Wu, M., Trappe, W., Wang, Z.J., Liu, R.: Collusion-resistant fingerprinting for multimedia. IEEE Signal Processing Magazine 21(2), 15–27 (2004)

3. Wu, Y.D.: Linear Combination Collusion Attack and its Application on an Anti-Collusion Fingerprinting
4. Wang, Z.J., Wu, M., Zhao, H.V., Trappe, W., Liu, K.J.R.: Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation. IEEE Trans. Image Processing 14(6), 804–821 (2005)
5. Swaminathan, A., He, S., Wu, M.: Exploring QIM based Anti-Collusion Fingerprinting for Multimedia
6. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Trans. Inform. Theory 44(5), 1897–1905 (1998)
7. Schaathun, H.G.: The Boneh-Shaw fingerprinting scheme is better than we thought. IEEE Trans. Information Forensics and Security 1(2), 248–255 (2006)
8. Tardos, G.: Optimal probabilistic fingerprint codes. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, STOC, pp. 116–125 (2003)
9. Hollmann, H.D.L., Lint, J.H.V., Linnartz, J.P., Tolhuizen, L.M.G.M.: On codes with the identifiable parent property. Journal of Combinatorial Theory 82, 121–133 (1998)
10. Seol, J.M., Kim, S.W.: A scalable AND-ACC fingerprinting scheme for practical contents distribution. In: SPIE Visual Communications and Image Processing (2005)
11. Kang, K., Lee, C.H., Lee, H.Y., Kim, J.T., Lee, H.K.: Averaging attack resilient video fingerprinting. In: IEEE ISCAS 2005, pp. 5529–5532 (2005)
12. Jang, D., Yoo, C.D.: A novel embedding method for an anti-collusion fingerprinting by embedding both a code and an orthogonal fingerprint. In: IEEE ICASSP 2006, pp. 485–488 (2006)
13. Celik, M.U., Sharma, G., Tekalp, A.M.: Collusion-Resilient Fingerprinting by Random Pre-Warping. IEEE Signal Processing Letters 11(10), 831–835 (2004)
14. Mao, Y.N., Mihcak, K.: Collusion-Resistant Intentional De-Synchronization for Digital Video Fingerprinting. In: IEEE Int. Conf. Image Processing 2005, vol. 1, pp. 237–240 (2005)
15. Liu, Z.X., Lian, S.G., Ren, Z.: Image Desynchronization for Secure Collusion-Resilient Fingerprint in Compression Domain. In: Zhuang, Y.-t., Yang, S.-Q., Rui, Y., He, Q. (eds.) PCM 2006. LNCS, vol. 4261, pp. 56–63. Springer, Heidelberg (2006)
16. Liu, Z.X., Lian, S.G., Wang, R.G., Ren, Z.: Desynchronization in compression process for collusion resilient video fingerprint. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 308–322. Springer, Heidelberg (2006)
17. Hosseini, H.M.M., Huang, F.M.: A Proxy-based System for Fingerprinting of Streaming MPEG Video. In: IEEE Pacific Rim Conference on Communications, Computers and signal Processing, vol. 1, pp. 73–76 (2003)
18. Luh, W., Kundur, D.: New Paradigms for Effective Multicasting and Fingerprinting of Entertainment Media. IEEE Communications Magazine, 77–84 (June 2005)
19. Chu, H., Qiao, L., Nahrstedt, K.: A secure multicast protocol with copyright protection. In: Proc. ACM SIGCOMM Computer Communication Rev., vol. 32, pp. 42–60 (2002)
20. Judge, P., Ammar, M.: Whim: Watermarking multicast video with a hierarchy of intermediaries. In: NOSSDAC, June 2000, Chapel Hill, NC (2000)
21. Wu, T., Wu, S.: Selective encryption and watermarking of mpeg video. In: The Int. Conf. Imaging Science, Systems, and Technology (June 1997)

22. Simitopoulos, D., Zissis, N., Georgiadis, P., Emmanouilidis, V., Strintzis, M.G.: Encryption and Watermarking for the Secure Distribution of Copyrighted MPEG Video on DVD. ACM Multimedia Systems Journal, Special Issue on Multimedia Security 9(3), 217–227 (2003)
23. Kundur, D., Karthik, K.: Video Fingerprinting and Encryption Principles for Digital Rights Management. Proceedings of the IEEE 92(6), 918–932 (2004)
24. Lian, S.G., Liu, Z.X., Ren, Z., Wang, H.L.: Communitative Encryption and Watermarking in Video Compression. IEEE TCSVT (accepted)
25. Lian, S.G., Liu, Z.X., Ren, Z., Wang, H.L.: Commutative watermarking and encryption for media data. Optical Engineering Letters 45(8), 080510–080511 (2006)
26. Barni, M., Bartolini, F., Cappellini, V., Piva, A.: A DCT-domain system for robust image watermarking. Signal Processing 66(3), 357–372 (1999)

Design an Aperiodic Stochastic Resonance Signal Processor for Digital Watermarking

Shuifa Sun[1,2,3,⋆], Bangjun Lei[2], Sheng Zheng[1,2], Sam Kwong[3], and Xuejun Zhou[1]

[1] College of Electrical Engineering and Information Technology,
China Three Gorges University, Yichang 443002, China
{watersun,zsh,zhxuejun}@ctgu.edu.cn

[2] Institute of Intelligent Vision and Image Information,
China Three Gorges University, Yichang 443002, China
{Bangjun.Lei}@ieee.org

[3] Department of Computer Science, City University of Hong Kong, Kowloon,
Hong Kong SAR, P.R. China
CSSAMK@cityu.edu.hk

Abstract. In this paper, we propose an aperiodic stochastic resonance (ASR) signal processor for communication systems based on a bi-stable mechanism. This processor can detect base-band binary pulse amplitude modulation (PAM) signals. In this processor related parameters can be adjusted. The adjustment mechanism is explained from the perspective of the conventional noise-induced nonlinear signal processing. To demonstrate this processors usability, based on it we implemented a digital image watermarking algorithm in the discrete cosine transform (DCT) domain. In this algorithm, the watermark and the DCT alternating current (ac) coefficients of the image are viewed as the input signal and the channel noise, respectively. This phenomenon that the detection bit error ratio (BER) of the system suffering from certain attacks is lower than that of the system not suffering from any attack is systematically analyzed.

Keywords: digital watermarking, stochastic resonance, signal processing.

1 Introduction

Stochastic resonance (SR) is an effective approach for signal processing [1-8]. From this perspective, SR effect is commonly understood as first an increase and then a decrease in the signal-to-noise ratio (SNR) at the output with varying noise level at the input. Other quantitative measures, such as bit error ratio (BER), can also be employed. On the other hand, digital watermarking plays a very important role in preventing multimedia works from being pirated. Its idea is to embed some critical information by replacing parts of original media data with certain so-called watermark. The watermark can then be detected purpose by the receiver. Imperceptivity and robustness are two basic requirements

[⋆] Corresponding author.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 350–364, 2008.

to digital watermarking. Imperceptivity means that the difference between the embedded media and the original media should be imperceptible. For instance, the difference should not be easily perceived by humans eyes for an image watermarking. Robustness means that the watermarking system should be able to survive some attacks. From the signal processing perspective, digital watermarking involves detecting weak signals in the presence of strong background noises. In this paper, we attempt to connect the SR with practical signal processing systems, digital watermarking system and design an ASR signal processor [2-5] for digital watermarking.

Two approaches to digital watermarking were proposed for grayscale images in spatial domain [9]. A frequency-domain watermarking was introduced by Cox and Kilian [10]. The watermark is inserted into the image in the frequency-domain to produce the watermarked image. To verify the presence of the watermark, the similarity between the recovered watermark and the original watermark is measured. A blind watermarking detection algorithm based on the correlation detection was further proposed by Barni [11]. Barni computed the global DCT transformation for a given image, and then selected some middle and low frequency DCT coefficients to embed the marker. Quantization-index modulation (QIM) methods, introduced by Chen and Wornell [12], possess attractive practical and theoretical properties for watermarking. In [13], we proposed a digital watermarking based on the parameter-induced stochastic resonance (PSR) in the global DCT domain, and in [14], Wu and Qiu proposed a novel watermarking scheme based on stochastic resonance in the 8*8 DCT domain. However, how to adjust the parameters of the SR system according the given communication condition to get the optimal detection performance is not yet addressed. In this study, a signal processor based on aperiodic stochastic resonance (ASR) [2-5] is investigated. A digital image watermarking algorithm in the discrete cosine transform (DCT) domain is implemented based on this ASR signal processor. The paper is organized as follows. A signal processor based on the nonlinear bi-stable dynamic system is investigated in Section 2. A digital image watermarking algorithm in DCT domain is proposed in Section 3. Experimental results are presented in Section 4 and conclusions are drawn in Section 5.

2 Bi-stable ASR Signal Processor

2.1 A Bi-stable ASR Signal Processor

From the signal processing perspective, the mathematical model of a nonlinear bi-stable dynamic system can be written as

$$dx/dt = -dV(x)/dx + Input(t) \tag{1}$$

where $V(x)$ is the quartic potential function and can be written as $V(x) = -ax^2/2 + \mu x^4/4$. The parameters a and are positive and given in terms of the potential parameters. The quartic potential function $V(x)$ represents a bi-stable nonlinear potential with two wells and a barrier. The input can be written as

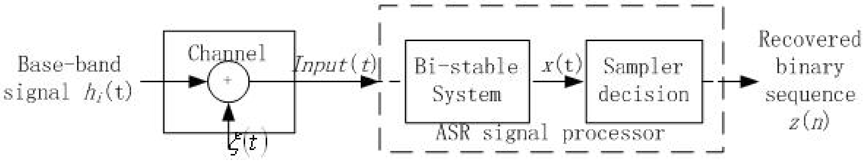


Fig. 1. ASR signal processor in a base-band binary communication system

$Input(t) = h(t) + \xi(t)$, where $h(t)$ is the input signal and $\xi(t)$ is the noise. If the signal $h(t)$ is an aperiodic signal and the SR effect occurs, it is called an ASR system [4]. This bi-stable system was used by Hu et al. [2], Godivier & Chapeau-Blondeau [3] and Duan & Xu [4] to detect the base-band pulse amplitude modulated (PAM) aperiodic binary signal $h(t)$ in the presence of the channel noise $\xi(t)$, , as shown in Fig. 1. The signal waveforms can be represented as $h_1(t) = -A$ and $h_2(t) = A$ for $(n-1)T_s \leq t < nT_s$, n=1,2, If the amplitude of the aperiodic signal A is not larger than the critical value $A_{CR} = 4a^3/27\mu$ [6], then the input signals is called sub-threshold signal and supra-threshold signal otherwise. Here, a parameter Q_{SR} called SR-Degree is defined as follows:

$$Q_{SR} = A/A_{CR}. \tag{2}$$

When the SR-Degree $Q_{SR} < 1$, the aforementioned nonlinear bi-stable dynamic system corresponds to the sub-threshold system and supra-threshold system otherwise. This study is limited in the sub-threshold system, i.e. $A < A_{CR}$. The time interval T_s is termed as bit duration and the code rate $r = 1/T_s$. The BER of the system is $P_e = P(1)P(0|1) + P(0)P(1|0)$. Readers should refer to [2-5] for more details about this signal processor.

2.2 Design of the ASR Signal Processor

To design the ASR signal processor, we should first select the SR-Degree Q_{SR} of the nonlinear bi-stable dynamic system according to the requirement of the system's robustness to signal. According to the results of [7], for a nonlinear bi-stable dynamic system, the bigger the SR-Degree Q_{SR} is, the worse the robustness of the system to signal. The robustness of the system to the signal is described as follows: to transfer from a sub-threshold system to a supra-threshold system, if the change of the signal amplitude of a system A is larger than that of a system B, system A is said to be more robust to signal than system B. This will put the system at risk of becoming a supra-threshold system and the noise will not play a constructive role in the signal transmitting. When the SR-Degree Q_{SR} and the signal amplitude of the system A are fixed, according to the definition of SR-Degree, the critical value A_{CR} is also fixed. In this case, only one parameter of the nonlinear bi-stable dynamic system is adjustable, a or μ. In current study we use a. Figure 2 shows the relationship between the BER and the parameter a when Q_{SR} is smaller than 1. In the simulation, Q_{SR} is 0.86. a changes from 1.8 to 9.8 with a step of 0.8. The bit duration T_s =1 second. The sampling interval Δt

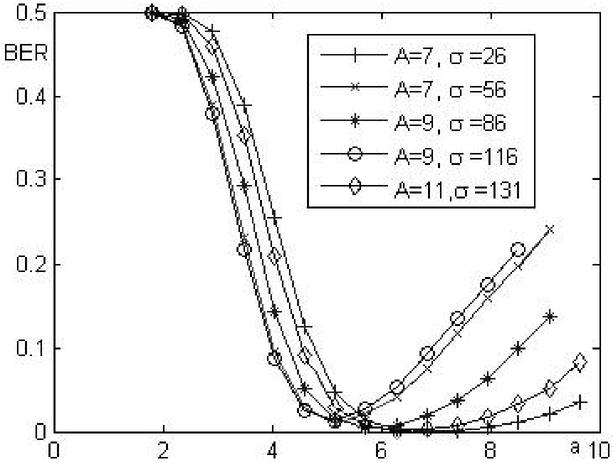


Fig. 2. Parameter-induced stochastic resonance effect in a nonlinear bi-stable dynamic system

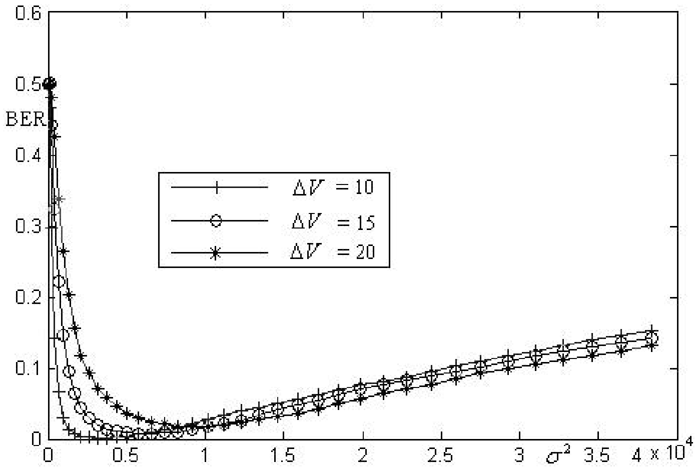


Fig. 3. Relationship between BER and σ^2, with $\varDelta V$ changing from 10 to 20 with a step of 5. The SR-Degree Q_{SR} is 0.86. The amplitude of the signal A =15. The mean of the noise is zero and the RMS amplitude of the noise σ changes from 1 to 96 with a step of 5. Other parameters of the simulation were the same as those in Figure2.

is 0.002 second. This means that there are $N = T_S/\varDelta t$=500 samples in one bit duration. The noise $\xi(t)$ is white Gaussian noise. The noise density D is given by $\delta^2\varDelta t/2$ [3]. In total 10000 bits were tested in this simulation. When building the binary signal from the recovered waveform of the bi-stable system, the measure was the sample of $x(t)$ at the end time of each bit duration [3] and the threshold was zero. We can see that the BER decreases first and then increases with a. For some practical communication systems, such as the watermarking system given in Sec. 3, the robustness to the noise is another issue to consider when we design the system. The communication system's robustness to noise is defined as follows: When the change of the BER of a system A is smaller than that of another system B with the same change of the density of the noise, A is more robust to the noise than B. Figure 3 shows the relationship between the BER and σ^2 with different $\Delta V$when Q_{SR} is fixed. We can see that the larger ΔV is, the better the system's robustness to noise, but the larger the BER minimum. We can now apply this ASR signal processor to a practical communication system and select a according to the specific requirements. That is, we can select a according to what is more important to the system, the minimum of the BER or the system's robustness to noise.

3 Watermarking Based on ASR Signal Processor

In the remainder of this paper, a digital image watermarking algorithm based on the aforementioned ASR signal processor is implemented. The watermark sequence and the DCT alternating current (ac) coefficients of the image are viewed as the weak signal and the noise of the ASR signal processor, respectively, making up the input of the ASR signal processor.

3.1 Watermark Embedding

Referring to Eq. (1), we can view the DCT ac coefficients as the sampled noise $\xi^E(l)$ and the watermark as the signal $h_i^E(j) \in \{1, -1\}$, where $0 \leq l < L$ and L is the total number of DCT ac coefficients used, $i \in \{1, 2\}$, h_1^E =1, h_2^E =-1, $0 \leq j < J$, and J is the length of the binary watermark sequence. The steps for the watermark embedding are as follows:

STEP 1. Take the global DCT transform of a given image to embed watermark.

STEP 2. Select L band pass coefficients, rank order them, and scramble them with a key k. The index for the first coefficient is M and $1 \leq M$. This means that we do not use the DCT direct current (DC) coefficient. Any change in this coefficient will modify the mean level of the image and thus makes the watermarking perceptible [10]. In fact, the same logic also holds for the first few DCT ac coefficients. Such as for the simulations given in Sec.4, M is 1000 for an image of size 515$*$ 512 [11]. The parameter M and L can also work as keys of the watermarking system if necessary. But it is more appropriate to transmit it as side information of the system.

STEP 3. Embed the J-long watermark message additively using a chip rate of $Q = L/J$ and strength f.

STEP 4. Take the inverse DCT to obtain the watermarked image.

3.2 Watermark Detection

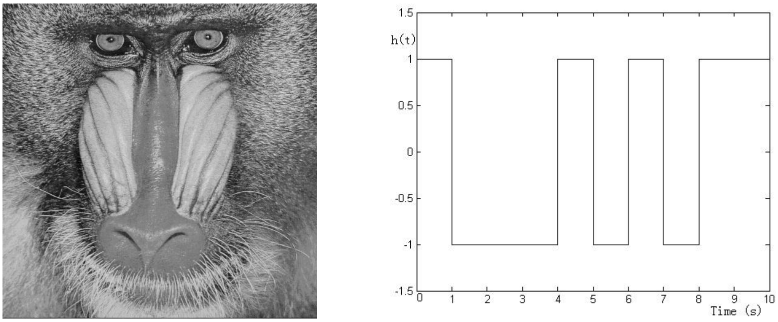
The first three steps for the watermark detection are the same as those in the above embedding procedure. The whole procedure for watermark detection is as follows: STEP 1. Take the global DCT transform of a given image to detect watermark. STEP 2. Select L band pass coefficients, rank order them, and scramble them with a key k. STEP 3. Substituting the select and scrambled sequence $\xi^U(l)$ into Eq.(1) as the sampled input $Input(t)$ of the bistable system, we get the following Eq. (3) [8]

$$x(l+1) = \Delta t(ax(l) - \mu x^3(l) + \xi^U(l)) + x(l). \tag{3}$$

Without loss of generality, we may assume that $x(0)$=0 in the numerical simulation. We can then get the detected sequence $h_i^D(jq) = x(l+1)$ and recover the watermark h_i^D by using the sample at the end time of each bit duration [3] or use the statistical method proposed in [5] to improve the detection performance of the system. Comparing h_i^D with h_i^E, we obtain the total bit error number and BER of the watermarking system. For both watermarking algorithms in [10] and [11], the DCT ac coefficients of the image are actually viewed as the additive white Gaussian noise because the correlation detector, which is the optimal detector for the communication system in the presence of this kind of noise, is used. In the present watermarking system, not only the selected DCT ac coefficients sequence but also the noise imported by the attack is viewed as the additive white Gaussian noise. If the image A^U is the watermarked image A^M, which means that the watermarking system has not been attacked, the aforementioned ASR signal processor works well. If indeed the watermarking communication system suffers from some kinds of attack, $\xi^U(l)$ is still a white Gaussian noise because it is a combination of two additive white Gaussian noises. Therefore, the aforementioned ASR signal processor will work well in both cases.

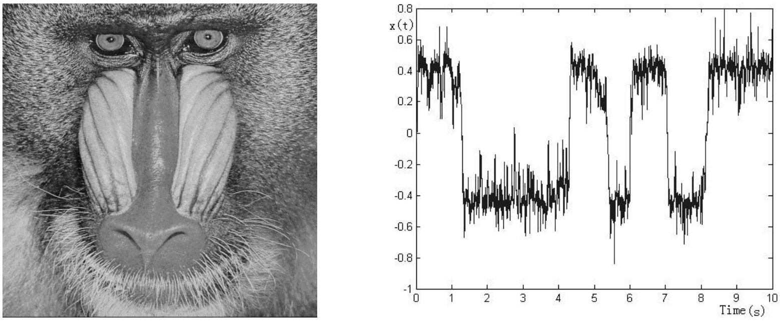
4 Numerical Simulations and Results

In this section we introduce our numerical simulations on the above implementation. The original image is the standard gray-scale image "mandrill" with a size of 512∗512, as shown in Fig. 4(a). The watermark is a PN sequence h_i^E generated



(a) (b)

Fig. 4. (a) Original image mandrill. (b) Original watermark code.



(a) (b)

Fig. 5. No attack case. (a) Watermarked image. (b) Detected sequence when the bi-stable system parameter a =22. Other simulation parameters are the same as those in Fig. 6.

by a linear feedback shift register with 8 registers. The length of the watermark code J is 255. The first 10 codes are shown in Fig. 4(b). The parameters M and L for the simulations were 1000 and $255 * 500$, respectively. This means that the parameter $Q = 500$. The scaling parameter f was 5. The watermarked image is shown in Fig. 5(a) and the peak signal to noise ratio (PSNR) was 37.28 dB. Comparing it with the original image in Fig. 4(a), we hardly can see any

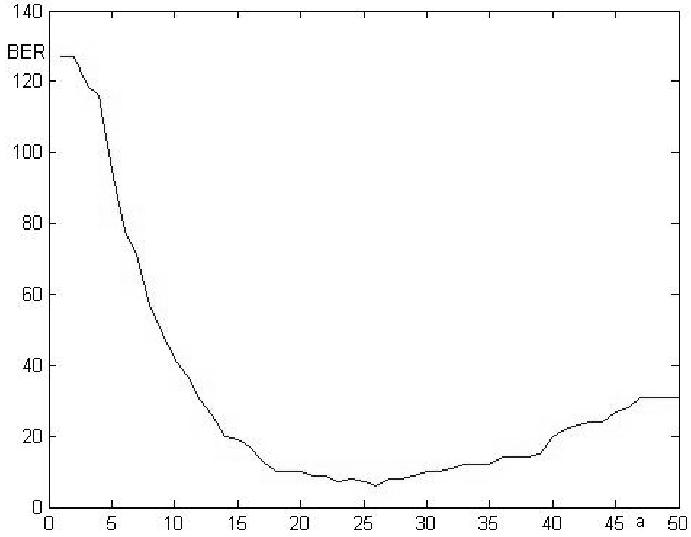
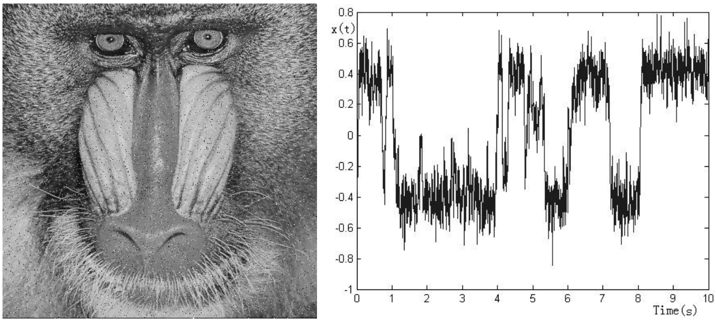


Fig. 6. Relationship between the BER and the parameter a of the bi-stable system. The SR-Degree Q_{SR} is 0.86. a changes from 1 to 50 with a step of 1. The sample interval Δt in Eq. (3) is 0.002. The recovered waveform for the first 10 codes is shown in Fig. 5 (b) where the bi-stable system parameter a =22.

difference. From Fig.6 we can clear see that the parameter-induced stochastic resonance phenomenon happened in the watermarking system. In the following attack tests, we will use the optimal value a_{SR} (=22) to detect the watermark.

4.1 Adding Salt and Pepper Noise Attack Test

Some pulse noises are added into the image when the image is transmitted or processed. These kinds of noises are often called salt and pepper noises and make the image looking quite dirtier and older. Fig. 7(a) is the image when the watermarked image was corrupted by the salt and pepper noise with density of 2.5%, the peak signal to noise ratio (PSNR) was 21.63 dB. Comparing it with the watermarked image in Fig. 5(a), the difference is obvious. As shown in Fig. 7(b), however, the trace of the watermark is clear and there are only 12 error codes among all 255 transmitted codes.



(a) (b)

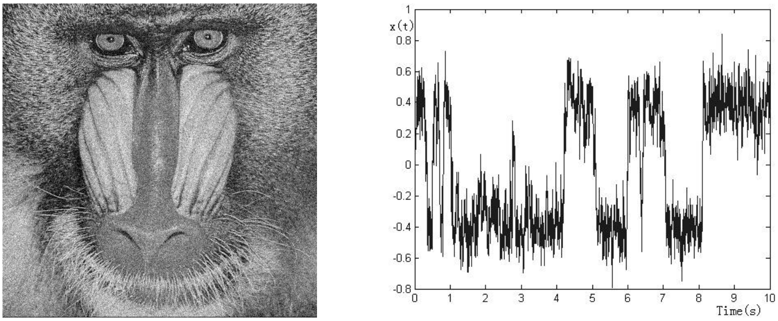
Fig. 7. Adding salt and pepper noise attack case. (a) Attacked image after adding spiced salt noise. (b) Detected waveform for the first 10 codes. Other simulation parameters are same with those of Fig. 6.

4.2 Adding Gaussian Noise Attack Test

Another typical noise in the image transmission or processing is Gaussian noise, which makes an image blurry. Fig. 8 (a) is the image when the watermarked image is corrupted by a Gaussian noise whose mean and variance are 0 and 0.01, respectively. Comparing it with the watermarked image in Fig.5 (a), we can see the difference between them clearly and PSNR=20.03 dB. But the trace of the watermark is still obvious as shown in Fig.8 (b) and there were only 10 error codes among all 255 transmitted codes.

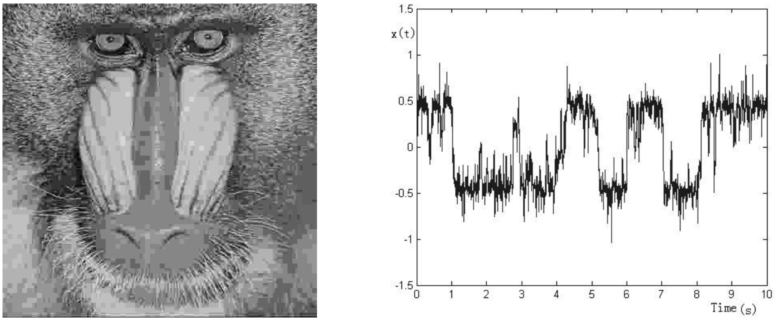
4.3 JPEG Compression Attack Test

JPEG compression is a widely used image processing method. Fig. 9(a) is the image when the watermarked image was compressed using JPEG with a quality factor 5. Comparing it with the watermarked image in Fig. 5 (a), the difference



(a) (b)

Fig. 8. Adding Gaussian noise attack case. (a) Image after adding Gaussian noise. (b) Detected waveform for the first 10 codes. Other simulation parameters are same with those of Fig. 6.



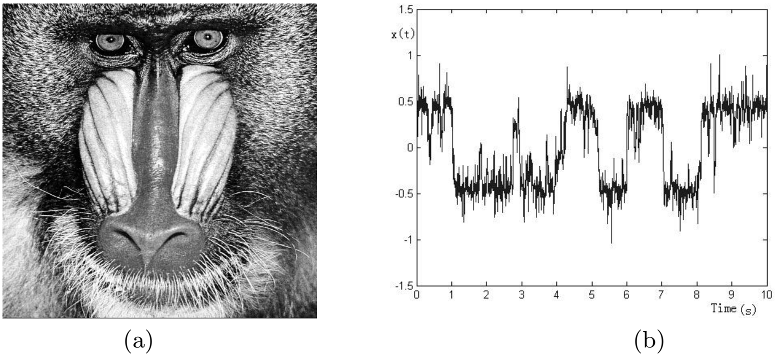
(a) (b)

Fig. 9. JPEG compression attack case. (a) Image after JPEG compression. (b) Detected waveform for the first 10 codes. Other simulation parameters are the same as those in Fig. 5.

caused by block effect is obvious and PSNR= 21.43 dB. But from the detected result shown in Fig. 9(b), there was still a trace of the watermark and the BER is 29.0%. Our explanation is given as follows: the JPEG compression forced many high frequency DCT ac coefficients to zeros and the watermark embedding in these coefficients were then removed. It is why that many digital watermarking algorithms do not use high frequency DCT ac coefficients as the carrier of watermark. Embedding watermark only into the low frequency DCT ac coefficients will overcome this defect of the watermarking system.

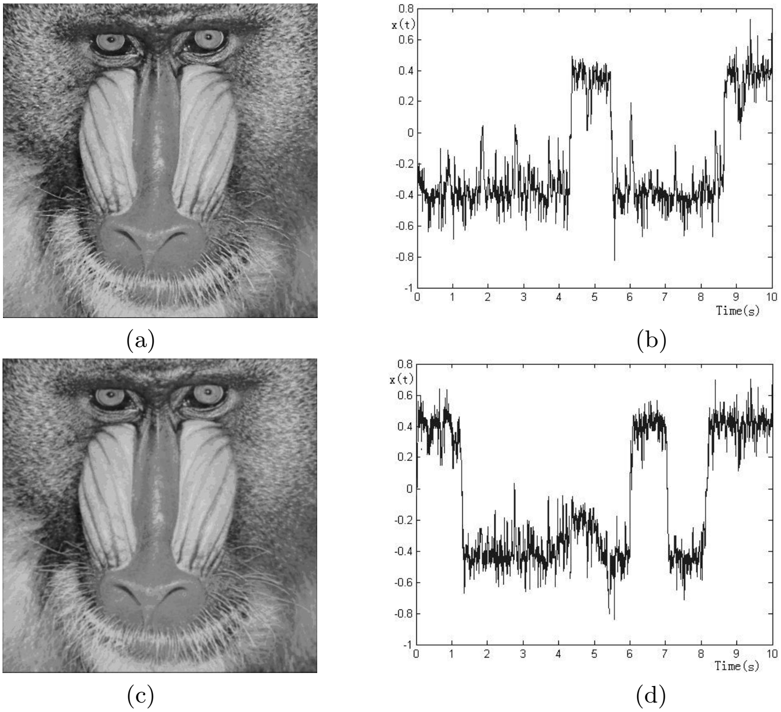
4.4 Histogram Equalization Attack Test

In the image processing, histogram equalization is usually used to increase the contrast of an image. Fig. 10(a) is the resulting image when the histogram of the



(a) (b)

Fig. 10. Histogram equalization attack case. (a) Image after histogram equalization. (b) Detected waveform for the first 10 codes. Other simulation parameters are the same as those in Fig. 5.



(a) (b)

(c) (d)

Fig. 11. Median filtering and Gaussian low pass filtering attack case. (a) Image after median filter attack, the size of the filter is 3∗3, and PSNR =33.79 dB. (b) Recovered waveform and the BER is 22.3%. (c) Image after Gaussian low pass filter attack, the size of the filter is 5∗5 and the standard deviation is 1, and PSNR=23.55 dB. (d) Recovered waveform and the BER is 24.7 %. Other simulation parameters are same with those of Fig. 6.

watermarked image was equalized. We can easily tell the difference between them by comparing it with the watermarked image in Fig. 5(a) (PSNR=17.60dB). But from the detected result shown in Fig. 10(b), there were trances of the watermark code and there were only 3 error codes among all 255 transmitted codes.

4.5 Other Low Pass Filter Attack Tests

To further test the performance of the developed watermarking system, other low pass filter attacks, such as median filter and Gaussian low pass filter, were carried out and the results are shown in Fig. 11. As shown in Fig. 11, the performance of the watermarking is not as good as the system suffers from the attacks by adding noise, such as the attacks of adding Gaussian noise or pulse noise. Our explanation to this is given as follow: both median filter and Gaussian low pass filter are low pass filters, which will remove the watermark embedded in the high frequency DCT ac coefficients. To make the watermarking system have the ability to resist against these kinds of attacks, the watermark should be embedded into the low frequency DCT ac coefficients, as pointed out for the JPEG compression attack.

4.6 SR Effect in the Presence of Attack

Figure 12 shows the relationship between the parameter a and the BER for all of the preceding attacks cases, as well as the no attack case. We can clearly see that the parameter-induced stochastic resonance effect in some of the watermarking communication systems being attacked. So, not only the DCT ac coefficients of

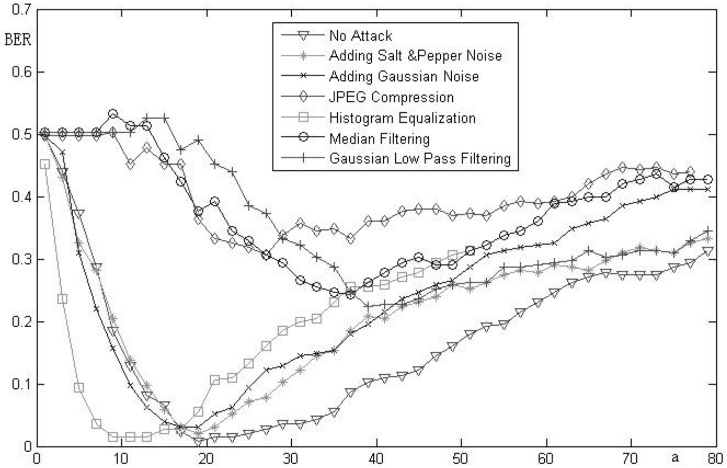


Fig. 12. Relationship between BER and the parameter a of the bi-stable system in the watermarking communication systems for the image mandrill. Other simulation parameters are same with those of Fig. 6.

Fig. 13. Test images lena, peppers, boat, bridge, goldhill and Barbara are shown by (a), (b), (c), (d), (e) and (f), respectively

the image themselves but also the combination of them and the noise imported by the attacks will cooperate with the bi-stable system to improve the watermark detection performance.

4.7 Tests on Other Images

The tests on other images, as shown in Fig. 13, were also carried out and the results are shown in Fig. 14. It is shown that the more the components of the image in high frequency is, such as the image mandrill, the more obvious the ASR effect is. For both watermarking algorithms proposed by Cox and Kilian [10], and Barni *et al.* [11], the DCT ac coefficients of the image are also viewed as the noise of the watermarking communication system. Whereas, for watermarking algorithms based on the ASR signal processor, the detection BER when the system suffers from some kinds of attacks is lower than that when the system does not suffer from any attacks. This is unbelievable for conventional watermarking systems but is reasonable for the ASR signal processor based on the nonlinear system, which the conventional linear signal processor does not have [3,7].

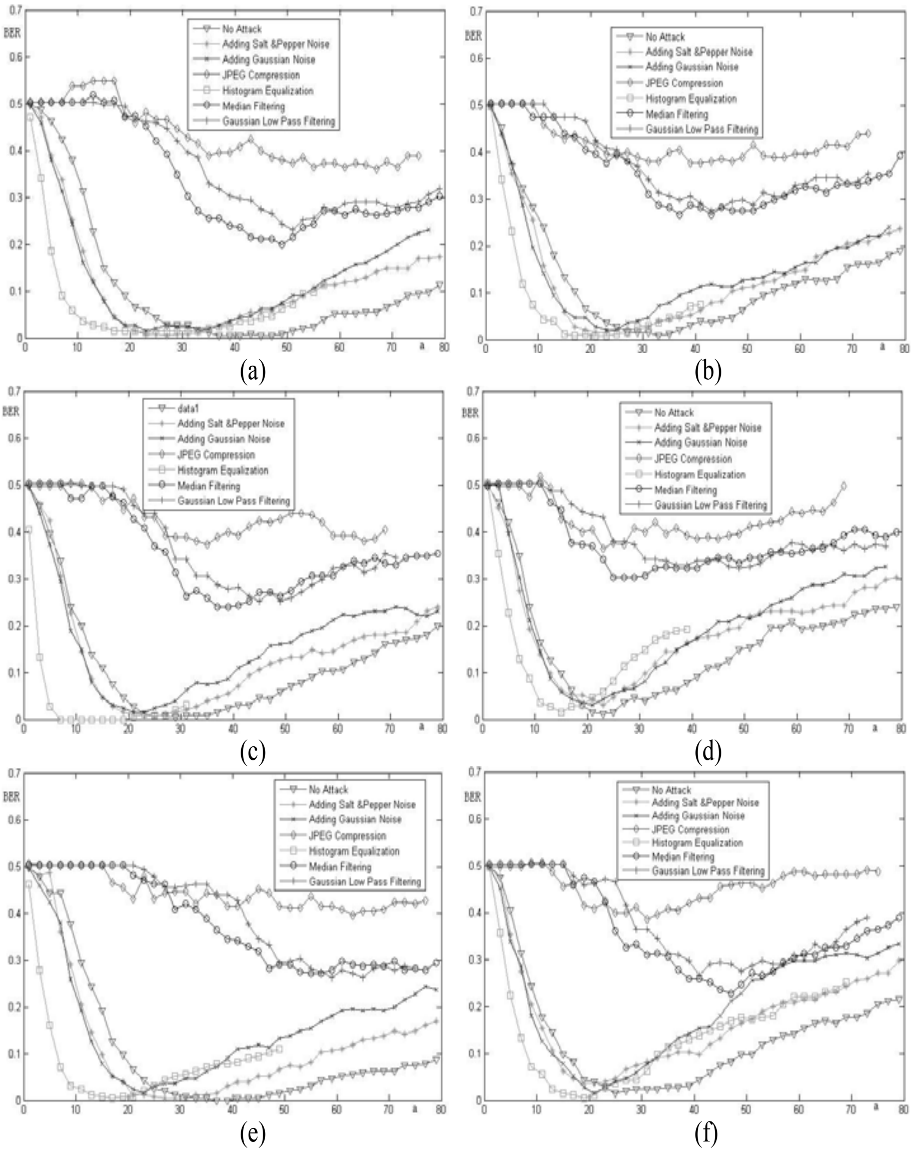


Fig. 14. Attack test results for the images lena, peppers, boat, bridge, goldhill and Barbara are shown by (a), (b), (c), (d), (e) and (f), respectively. The parameters for these experiments are same with those of Fig. 10.

5 Conclusion

In this paper, we investigated a signal processor based on ASR. The approach to design such a signal processor is proposed. A digital image watermarking algorithm based on this ASR signal processor was then implemented. The experimental results showed that, under certain circumstances, extra amount of noises can in fact improve rather than deteriorate the performance of some communication systems. When we implemented the watermarking system based on the ASR signal processor, both the selected DCT ac coefficients and the noise imported by the attacks are viewed as the additive white Gaussian noise. However, as shown by the results in [15], the characteristics of cover data (the selected DCT ac coefficients here) and the distortion vectors (the noise imported by the attack) are different for given images and attacks. So, further extension to this paper could be to study the ASR signal processor in the presence of other kinds of channel noises [16] and apply it to watermarking.

Acknowledgments. Shuifa Sun acknowledges the City University of Hongkong Strategic Grant 7001955.

References

1. Benzi, R., Sutera, S., Vulpiani, A.: The Mechanism of Stochastic Resonance. J. Phys. A 14, 453–457 (1981)
2. Hu, G., Gong, D., Wen, X., et al.: Stochastic resonance in a nonlinear system driven by an aperiod force. Phys. Rev. A 46, 3250–3254 (1992)
3. Godivier, X., Chapeau-Blondeau, F.: Stochastic resonance in the information capacity of a nonlinear dynamic system. Int. J. Bifurcation and Chaos 8(3), 581–590 (1998)
4. Duan, F., Xu, B.: Parameter-induced stochastic resonance and baseband binary PAM signals transmission over an AWGN channel. Int. J. Bifurcation and Chaos 13(2), 411 (2003)
5. Sun, S., Kwong, S.: Stochastic resonance signal processor: principle, capacity analysis and method. Int. J. Bifurcation and Chaos 17, 631–639 (2007)
6. Moss, F., Pierson, D., O'Gorman, D.: Stochastic resonance: Tutorial and update. Int. J. Bifurcation and Chaos 4(6), 1383–1398 (1994)
7. Xu, B., Duan, F., Chapeau-Blondeau, F.: Comparison of aperiodic stochastic resonance in a bistable system realized by adding noise and by tuning system parameters. Physical Review E 69, 061110, 1–8 (2004)
8. Mitaim, S., Kosko, B.: Adaptive stochastic resonance. Proc. IEEE. 86, 2152–2183 (1998)
9. Schyndel, R.G., Tirkel, A.Z., Osborne, C.F.: A Digital Watermark. In: Proc. IEEE Int. Conf. on Image Processing (IEE Computer Soc., Austin USA), pp. 86–90 (1994)
10. Cox, I., Kilian, J.: Secure spread spectrum watermarking for multimedia. IEEE Trans. on Image Processing 6(12), 1673–1687 (1997)
11. Barni, M., Bartolini, F., Capellini, V., Piva, A.: A DCT-domain system for robust image watermarking. Signal Processing 66, 357–372 (1998)
12. Chen, B., Wornell, G.: Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. IEEE Trans. on Information Theory 47, 1423–1443 (1998)

13. Sun, S., Qiu, P.: Algorithm of digit watermarking based on parameter-induced stochastic resonance. Journal on Communications 26(12), 48–55 (2005)
14. Watermarking Scheme Based on Stochastic Resonance. In: Proc. IEEE Int. Conf. on Signal Processing, Guiling China, pp. 1265–1269 (2006)
15. Cox, I., Miller, M.L., McKellips, A.L.: Watermarking as communications with side information. Proc. IEEE 87(7), 1127–1141 (1999)
16. Xu, B., Li, J., Duan, F.: Effects of colored noise on multi-frequency signal processing via stochastic resonance with tuning system parameters. Chaos. Solitons and Fractals 16, 93–106 (2003)

Watermarking for Authentication of LZ-77 Compressed Documents*

Yanfang Du, Jing Zhang, and Yuting Su

School of Electronic Information Engineering, Tianjin University,
Tianjin 300072, China
zhangjing@tju.edu.cn, duyanfang@eyou.com

Abstract. Fragile Watermarking by exploiting the multiplicity of encoding of a file using LZ77- based compressed method (we called"LZS-77"), has been proposed recently as a new method of document authentication. It allows one to hide enough information within the compressed document to warrant its authenticity. However, the watermark embedding will degrade the compression performance. In this paper, some variations have been made to the LZS-77 algorithm to reduce the compression degradation. The experimental results show that our algorithm (we called"LZSC-77"), compared to the LZS-77 algorithm, can improve compression rates on many files of the Cargary corpus and Canterbury corpus.

Keywords: LZ77 algorithm, document authentication, fragile watermark.

1 Introduction

So far, most of the research in watermarking has been focused on image, audio and video. Techniques that hide messages in documents are rare, and most of them treat the documents as images. However, nowadays it is common practice to distribute documents in compressed form over the network. Fragile watermark, which is a new type of watermark, is designed to ensure that the multimedia can not be changed without destroying the watermark, thus it can be used for document authentication.

LZ77 algorithm and its variations are well known lossless compression techniques. The gzip[1], an efficient variation of LZ77 algorithm, is widely used for document compression with its good compression rate and relatively high speed. Recently, a fragile watermark by exploiting the multiplicity of encoding of a file using LZ77-based compressed method (we called"LZS-77"), has been proposed for document authentication [2]. By modifying the gzip encoding, LZS-77 can embed secret information into compressed documents, but the watermark embedding will degrade the compression rate of gzip. In this paper, some changes will be done to the LZS-77 algorithm to reduce the compression degradation.

* The National High Technology Research and Development Program of China This work is supported by 863 Project (No. 2006AA01Z407).

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 365–372, 2008.

The rest of this paper is organized as follows. Section 2 reviews LZS-77 algorithm and the related researches. Section 3 describes the changes done to the LZS-77 algorithm. Section 4 presents the experimental results. Section 5 explains the security of document authentication, and section 6 gives the conclusion.

2 Review of LZS-77 Algorithm

In [3], Lempel and Ziv proposed the LZ77 algorithm, which is a technique that encodes the text by finding repeated parts and replacing them by references to preceding occurrences. Since then, a lot of its variations have been proposed, one of which is used in gzip and named as LZSS Variant [4]. In LZSS, repeated parts of the text are encoded as length-distance pairs that refer to previously seen text.

In LZ77-based compression algorithms, given the length of the longest match, there may be more than one match. The compressor is free to select any of these matches since the decompressor is able to recover the same text anyway. The mere act of selecting one of these matches can be used to convey information to the decompressor [2]. Assume that there are q matches($q > 1$), by selecting one particular match out of the q choices, it is able to embed $log_2(q)$bits of the secret message. In order to assign a unique binary code to each match, a complete binary tree is built in LZS-77. For example, for the case of $q = 5$, i.e. there are five matches in the window, their binary codes can be given as Fig.1 shows. We get the code 000, 001, 01, 10, 11. Thus, by selecting one of the matches, the corresponding secret bits can be embedded.

On the basis of LZS-77, Dube and Beaudoin [5] use parts of the compressed document as embedded data to improve compression rate. Their experimental results show that this technique can provide substantial improvements in some cases and substantial deteriorations in other cases. Moreover, documents

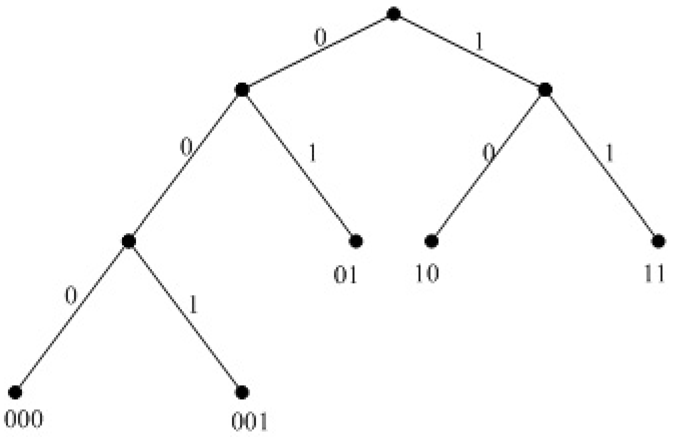


Fig. 1. The tree for q=5 in LZS-77

compressed by this technique can not be decoded by an ordinary LZ77 decompressor. In [6], Dube and Beaudoin extend the technique mentioned above for further improving compression rate.

In [7], Yoshioka et.al propose a method to control the trade-off between the maximum embeddable data size and compression rate, which is an extension and generalization of the LZS-77 algorithm. This algorithm has an advantage over LZS-77 in the maximum embeddable data size, but the compression performance is still unsatisfactory.

3 Improvement on the LZS-77 Algorithm

In the case of gzip, a statistical encoder - Huffman encoder is implemented following LZSS. When there are several feasible matches to select from, LZSS always chooses the closest one. There are two reasons to do so [5]. The first reason is that, by systematically selecting the closest match, the statistical distribution of the transmitted distances tends to be uneven, with higher frequencies for the short distances. This allows the Huffman coding to take advantage of the unevenness and send shorter code words on average. The second reason is that some original texts have a tendency to be locally similar. That is, the characteristics of the original text are not constant throughout its whole length. Consequently, copies of a particular sequence of bytes often appear relatively close to the sequence itself. Such texts also contribute to the unevenness of the distribution of the distances.

However, the LZS-77 algorithm does not always select the closest longest match, which tends to flatten the distribution of the distances and so goes against the following Huffman coding. So, in order to decrease the drop of compression rate, we should keep the unevenness of the transmitted distances' statistical distribution. In other words, we should do our best to select the nearer positions to embed the watermark.

Here, some changes will be done to the LZS-77 algorithm in order to improve its compression rate. Still suppose that there are five matches in the window, and the match positions are p_0, p_1, p_2, p_3 and p_4, whose distance to the current position grows in ascending order(that is, p_0 the nearest, p_1 the second nearest,...,and p_4 the farthest), Fig.2 shows the complete binary tree for assigning their binary codes. From the tree, we can get the code 000, 001, 01, 10, 11. Thus, if selecting one of the matches located in the left child, the first bit of the secret bits to be embedded must be '0', and if selecting one of matches located in the right child, the first bit of the secret bits to be embedded must be '1'. So if we add an index bit '0' or '1' to the beginning of the next secret bits to be embedded, then we can limit the match selection only in the left child or only in the right child. In Fig.2, it is clear that the positions located in the left child of the complete binary tree are nearer to the current position than the positions located in the right child. Thus by limiting the selection to the positions located in the left child, we can reduce the compression degradation.

In order to do so, an index bit of '0' must be added to the beginning of the next secret bits to be hidden. For example, in Fig.3, if the next bits to be hidden

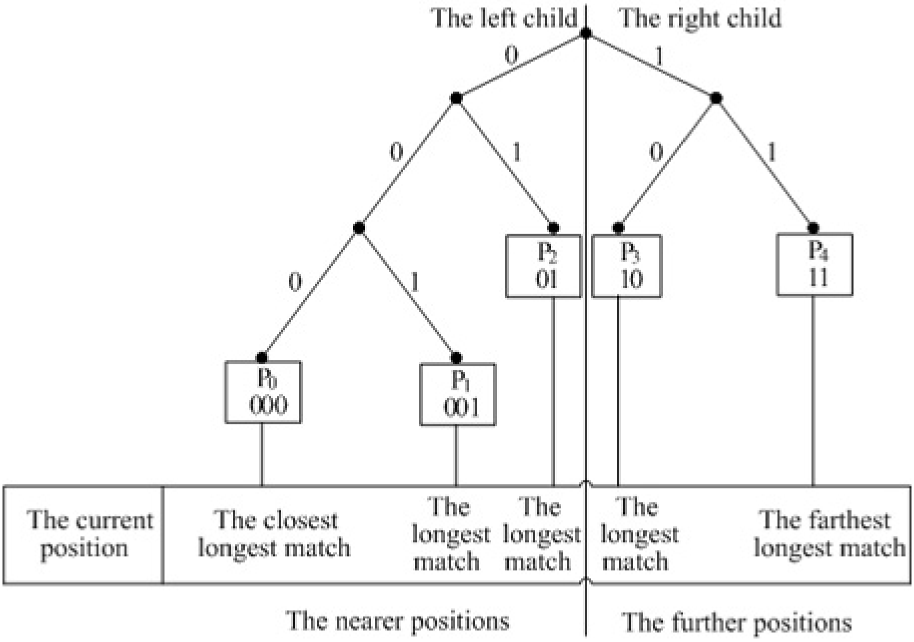


Fig. 2. The tree for q=5 in LZSC1-77

are "11011", according to the LZS-77 algorithm, p_4 should be selected, which is located in the right child of the complete tree, and meanwhile the bits "11"are embedded. But by adding an index bit '0' , the next bits become as "011011", in this case, p_2 should be selected, which is located in the left child of the complete tree, and meanwhile the bits "01" are embedded. Thus, with this index of '0', the encoder can only choose the relatively small offsets so that the following

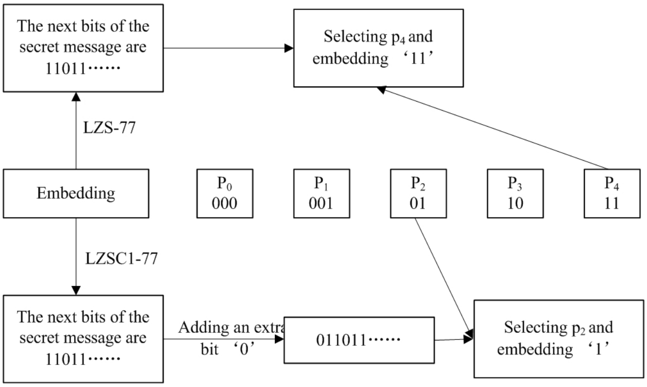

The next bits of the secret message are 11011······
Selecting p4 and embedding '11'
LZS-77
Embedding
P0 000
P1 001
P2 01
P3 10
P4 11
LZSC1-77
The next bits of the secret message are 11011······
Adding an extra bit '0'
011011······
Selecting p2 and embedding '1'

Fig. 3. The embedding difference between LZS-77 and LZSC1-77

Huffman encoding is still not disturbed too much. So this method can be used to decrease the drop of compression rate. We call it the LZSC1-77 algorithm.

When extracting the secret message using LZSC1-77, the index bit ‘0’ should be discarded. For example, if the distance received by the decompressor equals to the distance presented by p_2, according to the tree, we can obtain that the binary code is “01”. But the first bit “0” should be discarded as the index bit, so the valid message bit is “1” .

If the tree is built in such a way that the nearer positions are located in the right child, we should add an index bit ‘1’ to the beginning of the next secret bits to be embedded. Similar to the LZSC1-77 algorithm, we call this method the LZSC2-77 algorithm.

In the above two methods, an index bit is added to the beginning of the next bits to limit the selection to the nearer matches. In order to shorten the range of positions again, two bits can be added, e.g. “00” on the condition of Fig.2. This method is called the LZSC3-77 algorithm. But this method requires that the multiplicity of$q$ is no less than 4.

Our algorithms can be combined with the technique mentioned in [7]. First a threshold $offset_{\mathrm{th}}$ is given (for example, 1K), and then our method is used (for example, the LZSC2-77 algorithm). We call it the LZSC4-77 algorithm.

4 Experimental Results

We have verified our algorithms on many files of the Cargary corpus and Canterbury corpus [8], which are both widely available dataset for evaluation of data compression algorithms. Table 1 shows the size of the files compressed using each algorithm in maximum compression mode. The sizes are all measured in bytes.

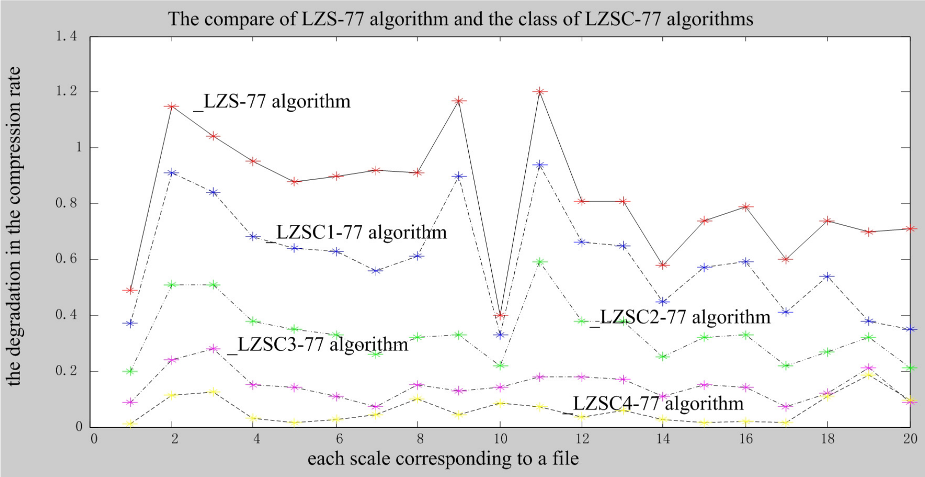


Fig. 4. A comparison of the LZS-77 algorithm and various LZSC-77 algorithms

Table 1. The experimental results of the size of the files compressed

$File$	$File - size$	$gzip$	$LZS - 77$	$LZSC1-77$	$LZSC2-77$	$LZSC3-77$	$LZSC4-77$
bib	111,261	34,900	35,449	35,312	35,123	35,005	34,913
obj1	21,504	10,320	10,568	10,516	10,430	10,371	10,344
obj2	246,814	81,087	83,643	83,156	82,341	81,766	81,390
paper1	53,161	18,543	19,047	18,906	18,745	18,623	18,559
paper2	82,199	29,667	30,386	30,193	29,954	29,779	29,679
paper3	46,526	18,074	18,491	18,367	18,226	18,126	18,086
paper4	13,286	5,534	5,656	5,608	5,569	5,543	5,540
pzper5	11,954	4,995	5,104	5,068	5,033	5,013	5,007
pzper6	38,105	13,213	13,662	13,555	13,340	13,261	13,230
pic	513,216	52,381	54,433	54,059	53,490	53,082	52,815
progc	39,611	13,261	13,738	13,633	13,493	13,331	13,290
progl	71,646	16,164	16,743	16,634	16,435	16,293	16,190
progp	49,379	11,186	11,585	11,506	11,373	11,270	11,215
trans	93,695	18,862	19,402	19,280	19,095	18,969	18,887
alice29.txt	152,089	54,191	55,313	55,062	54,681	54,416	54,212
asyoulik.txt	125,179	48,829	49,812	49,571	49,240	49,007	48,854
cp.html	24,603	7,981	8,129	8,083	8,035	7,999	7,991
fields.c	11,150	3,136	3,218	3,196	3,166	3,149	3,148
grammar.lsp	3,721	1,246	1,272	1,260	1,258	1,254	1,253
xargs.1	4,227	1,756	1,786	1,771	1,765	1,760	1,760

Table 2. The experimental results of bytes embedded

$File$	$LZS - 77$	$LZSC1 - 77$	$LZSC2 - 77$	$LZSC3 - 77$	$LZSC4 - 77$
bib	1,470	922	662	291	71
obj1	319	221	176	100	90
obj2	3,722	2,516	1,926	1,011	744
paper1	833	513	371	158	71
paper2	1,392	861	622	272	85
paper3	774	467	325	127	51
paper4	202	116	80	29	30
pzper5	178	106	71	29	33
pzper6	569	356	50	105	70
pic	2,584	1,924	1,576	1,041	786
progc	906	606	454	220	121
progp	562	371	275	136	82
trans	849	547	407	196	95
alice29.txt	2,641	1,668	1,198	543	137
asyoulik.txt	2,262	1,396	995	424	120
cp.html	264	159	112	46	28
fields.c	127	78	52	19	30
grammar.lsp	43	28	20	10	16
xargs.1	51	28	18	5	13

Fig.4 shows a comparison of the LZS-77 algorithm and the class of the LZSC-77 algorithms, on the aspect of degradation in compression. It's easy to see from the figure that, the LZSC-77 algorithms can really improve the compression rate compared to the LZS-77 algorithm. But this is the cost of sacrificing the bytes embedded, which could be seen in Table 2 (The sizes are all measured in bytes). It shows a comparison of the LZSC-77 algorithms on the aspect of bytes embedded. In practice, if the secret message to be embedded is not too much, the LZSC-77 algorithms can be used.

5 The Security of Authentication

In the LZS-77 algorithm, the security of authentication is ensured by shuffling all the multiplicity q matches with a certain class of pseudo-random generators. But this method can not be applied to the LZSC-77 algorithm, because we have to know which match is nearer. So we adopt a different method for the security of authentication. We use a pseudo-random generator to obtain a pseudo-random sequence consisting of "0" and "1". According to the pseudo-random sequence, when it's the bit "0", we use the LZSC1-77 algorithm to embed the watermark; and when it's the bit "1", we use the LZSC2-77 algorithm. Thus, Even if the attacker knows the internal implementation of the algorithm, he can't retrieve the hiding secret message from the compressed data without destroying it, unless the secret bit-string key is known to him.

6 Conclusions

In this paper, an algorithm, named as LZSC-77, is proposed to decrease the drop of the compression rate of gzip due to the embedding of watermark using the LZS-77 algorithm. The basic idea is to add an index bit to the beginning of the next bits of secret message so as to select the nearer matches. The experimental results are given to verify the feasibility of our algorithm.

References

1. Gailly, J.L., Adler, M.: The GZIP Compressor, `http://www.gzip.org`
2. Atallah, M.J., Lonardi, S.: Authentication of LZ-77 Compressed Data. In: Proceedings of the 18th ACM Symposium on Applied Computing, Melbourne, Florida, pp. 282–287 (2003)
3. Ziv, J., Lempel, A.: A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory 23(3), 337–343 (1977)
4. Blelloch, G.E.: Introduction to Data Compression. Computer Science Department, Carnegie Mellon University (2001)
5. Dube, D., Beaudoin, V.: Recycling bits in LZ77-based compression. In: Proceedings of the Conference of Sciences Electroniques, Technologies of Information and Telecommunications, Sousse, Tunisia (2005)

6. Dube, D., Beaudoin, V.: Improving LZ77 Data Compression using Bit Recycling. In: Proceedings of the International Symposium on Information Theory and its applications, Seoul, Korea (2006)
7. Yoshioka, K., Sonoda, K., Takizawa, O., Matsumoto, T.: Information Hiding on Lossless Data Compression. In: Proceedings of the International Conference on Intelligent Information Hiding and Multimedia, Pasadena, California, pp. 15–18 (2006)
8. Witten, I.H., Bell, T.C.: The Calgary/Canterbury Text Compression Corpus (1990), ftp://ftp.cpsc.ucalgary.ca/pub/projects/text.compression.corpus/

Robust Audio Zero-Watermark Based on LWT and Chaotic Modulation

Rangding Wang and Wenji Hu

CKC Software Laboratory, Ningbo University, Zhejiang 315211, China
wangrangding@nbu.edu.cn

Abstract. A robust audio zero-watermark scheme which constructed the watermark in lifting wavelet domain based on chaotic modulation is presented in this paper. Since the constructed zero-watermark is not embedded into the original audio indeed, the contradiction between imperceptibility and robustness is perfectly solved. The algorithm based on coefficient maximization guarantees that the zero-watermark which constructed by low frequency coefficients represents the significant characteristics of the audio signal, experiment results show that it is robust against general audio processing such as noise addition, low-pass filtering, requantization, resampling and mp3 compression. And after the watermark is created, an initial vector is obtained by chaotic modulation, so the storage space is reduced after generating the watermark.

Keywords: Lifting wavelet transform (LWT), zero-watermark, chaotic modulation.

1 Introduction

Rapid development of broadband communication networks facilitated fast transfer and perfect copying of digital media, including image, audio and video. These properties raised the issue of protection of intellectual property rights, monitoring of broadcast signals, etc. Digital watermark [1,2] is an efficacious technique to protect the copyright and ownership of digital information. But in the traditional methods of audio watermark, the information of original audio will be distorted more or less [3]. In order to find a new balance of imperceptibility and robustness different from the traditional watermark, a new watermark approach, zero-watermark technique, is proposed. The zero-watermark Algorithm changes the traditional methods that watermark is embedded into audios, and makes the watermarked audio distortion-free. Zero-watermark [4] which no additive information is embedded in the original audio can successfully solve the contradiction between imperceptibility and robustness. However, in recent research the original zero-watermark should be all saved for detection [5], thus more storage space is needed.

In this paper, a new audio zero-watermark scheme based on lifting wavelet transform and chaotic modulation [6] is proposed. The zero-watermark is created by

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 373–381, 2008.

selecting some maximum absolute value of low frequency wavelet coefficients of original audio and computing the character of the selected coefficients. The construction of the watermark is random by chaotic sequence. After generating the watermark, chaotic inverse search is adopted to get the initial value of the watermark sequence. The watermark extracting process is invert process and the initial value, instead of the original audio is required for recovering the embedded watermark. The ownership of the audio can be determined by computing normalized correlation coefficient between the original watermark and extracted watermark. The experimental results prove that the method of watermark attains a high quality of imperceptibility and robustness. The attack experiments, such as additive Gaussian noise, low-pass filter, requantization, re-sample and mp3 compression, give strong evidences for the robustness of the approach.

The paper is organized as follows: Section 2 explains lifting wavelet decomposition scheme. In Section 3, the zero-watermark algorithm based on lifting wavelet transform and chaotic modulation is introduced. And in Section 4 some results are given and analyzed, followed by the concluding of the remarks in Section 5.

2 Wavelet Lifting

The traditional wavelet transform is performed by convolution, the computation is complex and auxiliary memory is needed. Thus, wavelet lifting is introduced in this paper. The lifting scheme is a new idea of constructing wavelets and has several unique advantages in comparison with conventional convolution-based wavelet transform. It allows for an in-place implementation of wavelet transform and reduces computation time and memory requirement greatly.

Generally, lifting scheme is composed of three steps: Split/Merge, Prediction and Update. The detail reasoning and proving about lifting scheme is given by reference [7].

1) Split step: The split step is also called the lazy wavelet transform. The operation just splits the input signal $x(n)$ into odd and even samples: $x_e(n)$ and $x_o(n)$.

$$x_e(n) = x(2n),\ x_o(n)= x(2n+1) \tag{1}$$

2) Prediction step: Keep even samples changeless, and use $x_e(n)$ predicts $x_o(n)$. The difference between the prediction value of $P[x_e(n)]$ and the real value of $x_o(n)$ is defined as detail signal $d(n)$.

$$d(n) = x_o(n) - P[x_e(n)] \tag{2}$$

Where $P[\bullet]$ is the predict operator. The detail signal $d(n)$ denotes the high-frequency component of original signal $x(n)$.

3) Update step: Introduce the update operator $U[\bullet]$, and use detail signal $d(n)$ to update even samples $x_e(n)$. Then the approximate signal $c(n)$ denotes the low-frequency component of original signal.

$$c(n) = x_e(n) + U[d(n)] \tag{3}$$

Then the next transform step can be performed, but only using the low-frequency component just as wavelet transform. The reconstruction of lifting wavelet transform is an inverse process of decomposition. The lifting scheme of decomposition and reconstruction is illustrated in Fig.1.

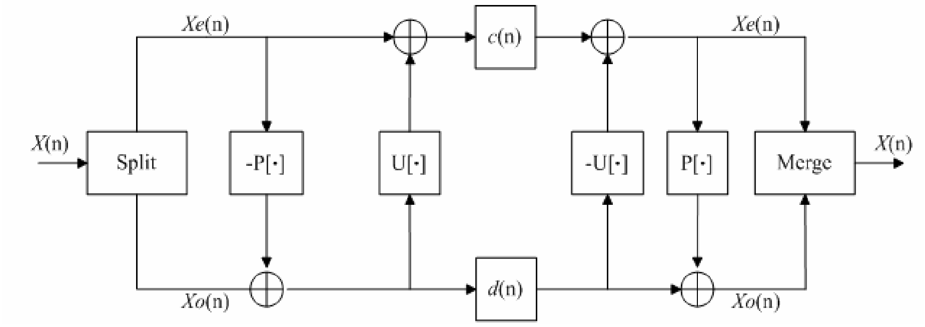


Fig. 1. The decomposition and reconstruction of the lifting scheme

3 Zero-Watermark Algorithm

3.1 Zero-Watermark Construction Process

Decompose the original audio and identify low-frequency coefficients of the largest absolute value of locations *N*, and rank these coefficients to one-dimensional sequence *X* from 1 to *N* according to the location of the coefficients. Then a renewable pseudo-random sequence *D* which ranges from 1 to *N* is generated by a secret key *K* for the length of *N*.

$$\begin{aligned} X &= \{x(i), 1 \le i \le N\} \\ D &= \{d(i), 1 \le i \le N\} \end{aligned} \tag{4}$$

which results in $d(i) \in [1, N]$. Then make the random sequence *D* as suffix of the new sequence X_d.

$$X_d = \{x_d(i), 1 \le i \le N\} = \{x(d(i)), 1 \le i \le N\} \tag{5}$$

Finally we define zero-watermark $W = \{w(i), 1 \le i \le N\}$ as follows:

$$\begin{cases} w(i) = 1, & if \quad x_d(i) \ge 0 \\ w(i) = 0, & if \quad x_d(i) < 0 \end{cases} \tag{6}$$

Thus, the zero-watermark has been constructed. Fig.2 is the flowchart of zero-watermark construction process.

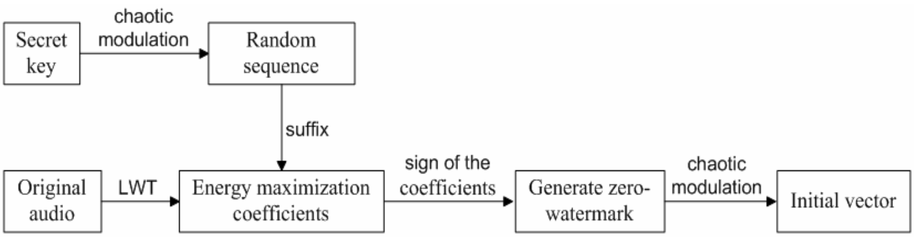


Fig. 2. Zero-watermark construction flowchart

3.2 Search Initial Value Using Chaotic Modulation

Suppose x_0 stand for initial value, the range of x_0 is in the interval (0, 1). According to Equation (7), we get chaotic sequence x_i. Input x_i into the mapping Equation (8) below, we can get the chaotic sequence as the watermark $\overline{W}$.

$$x_{i+1} = 3.99x_i(1 - x_i) \tag{7}$$

$$\overline{W(i)} = \begin{cases} 1, & if \quad 0.5 \le x_i < 1 \\ 0, & if \quad 0 < x_i < 0.5 \end{cases} \quad 1 \le i \le N \tag{8}$$

If W equals to $\overline{W}$, then x_0 is the initial value we expected. Otherwise, the initial value should be searched again.

In the search process for initial value, x_0 is initially set to a small value, for instance $x_0 = 0.0001$, if Logistic mapping can't result $W = \overline{W}$, then x_0 adds a small increment. For example, $x_0 = x_0\ 0.0001$. When x_0 reaches 1, but $W = \overline{W}$ is still inaccessible. In that case, either x_0 or the increment should be cut down further. In other hand, we can cut the watermark sequence into several sections. For each section there is an initial value. Search in different section can greatly enhance the probability of reaching $W = \overline{W}$ and get faster of searching process. In our experiments, the watermark is cut into fifty sections, the initial value searching process is very fast. Thus, x_0 is a vector instead of a single value when the watermark is divided into several pieces, and we mark it as H. The initial values (may be a vector) that are determined in the search from the original zero-watermark. The initial vector H and the secret key K are both needed in watermark detection process.

3.3 Zero-Watermark Detection Process

Given a redistributed audio, which may not be the same as the original audio due to any attack, the zero-watermark can be detected by correlation detection using a three-step process. Original audio is not needed in our method. The detailed detection steps are described as follows:

Step 1: Generate the watermark *W* of the original audio, according to initial vector *H*, through Equation (8) above;

Step 2: Decompose the detecting audio and identify low-frequency coefficients of the largest absolute value of locations *N*, and rank these coefficients to one-dimensional sequence X' from 1 to *N* according to the location of the coefficients. With the secret key *K*, obtain a random sequence D' and generate X_d' according to Equation (5). Then the watermark W' is generated through Equation (6);

Step 3: Compute the normalized correlation coefficient ρ between the original zero-watermark *W* and the extracted watermark W' with Equation (9):

$$\rho = \frac{\sum_{i=1}^{N} w(i) \cdot w'(i)}{\sqrt{\sum_{i=1}^{N} w^2(i)} \cdot \sqrt{\sum_{i=1}^{N} w'^2(i)}} \tag{9}$$

Set a threshold *T*, if $\rho > T$, then it is considered that the detecting audio contains the original zero-watermark, and the ownership of this audio belongs to author. Otherwise, the detecting audio is considered to contain no original watermark, and the ownership is illegal. The flowchart of zero-watermark detection process is given in Fig.3.

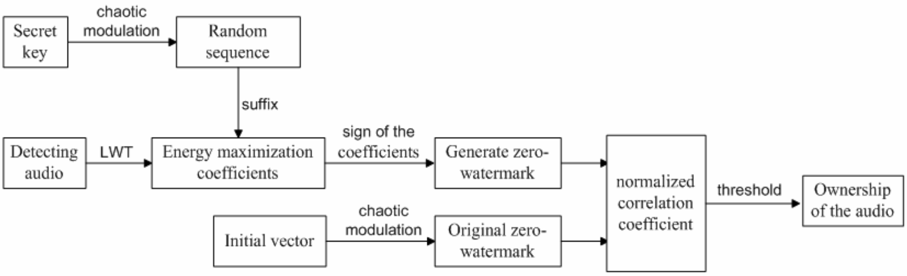


Fig. 3. Zero-watermark detection flowchart

3.4 The Choice of Threshold

The zero-watermark *W* is composed of binary sequence either 0 or 1. Each element reflects the difference between two adjacent coefficients of the corresponding locations. For positive difference, element value is 1, for the negative difference it is 0. For a test audio that is unrelated to the original one, each element in its zero-watermark is equal to the one in that of the original audio in a probability of 0.5, so the total similarity degree is approximately 0.5. Fig.4 shows the similarity degree between the different audio. We can see the normalized correlation coefficient is between 0.42 and 0.62. Thus, in our experiment, we set the threshold *T*=0.85.

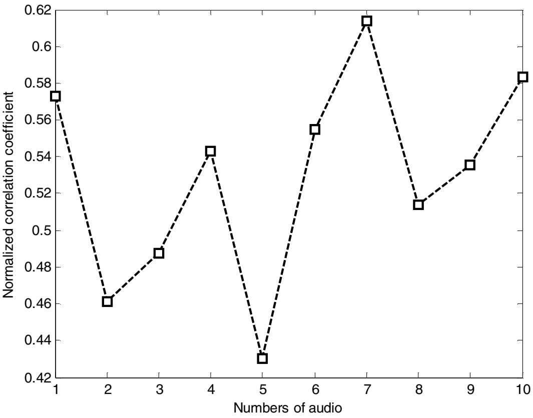


Fig. 4. Similarity degree between the different audio

4 Simulation Results and Analysis

The audio files used in the experiments include Blues, Classical, Country, Pop and Folk. Each audio piece has duration of 10 seconds and was sampled at 44.1 kHz with 16 bits/sample. The lifting wavelet basis is Haar, and N is set to 2000. In order to search the initial value quickly after generating the watermark, the watermark is cut into fifty sections. No additive information is embedded in the original audio, so the SNR of the watermarked audio is not necessary to be computed. We can compare the normalized correlation coefficients to judge whether the digital audio signal has been tampered or not. In order to test the robustness of zero-watermark, a set of attack experiments such as noise addition, low-pass filtering, requantization, resampling and mp3 compression were performed and the normalized correlation coefficients ρ is given in Table 1. The parameters of simulation attack in our experiments are set as follows:

Noise addition: White noise with a constant level of 20 dB is added to the watermarked audio signals.

Low-pass filtering: Low-pass filtering used a 6-tap Butter-worth filter with cutoff frequency 11025 Hz.

Requantization: The 16-bit watermarked audio signals have been requantized down to 8 bits/sample and back to 16 bits/sample.

Resampling: Watermarked audio signals with original sampling rate 44100 Hz have been subsampled down to 22050 Hz and upsampled back to 44100 Hz.

Mp3 compression: Compress the audio signals at 64kbps with mp3 coding and decompress.

Time-scale modification (TSM): Some change after time scaling up to ±5% is shown in Fig.5(b,c) (Just take the music type of Classical for example, fig5(a) is the original classical waveform). We can see that the shape of the waveform does not change a lot compared to the original audio signal but the length of the audio has been changed.

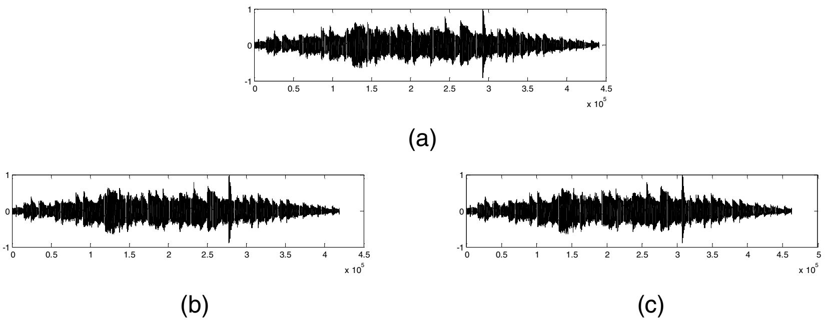
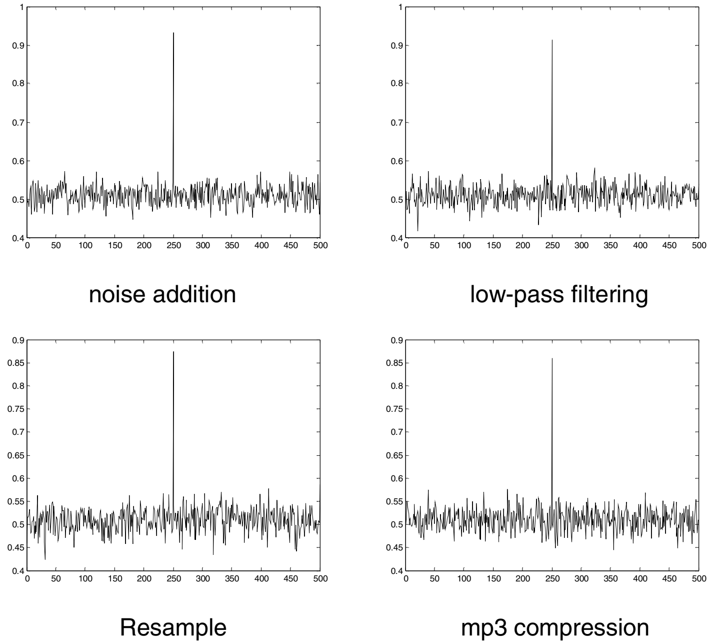


Fig. 5. Waveform of the original classical music and the ±5% time-scaled music

Take the music type of Blues for example, Fig 6 shows the corresponding correlation coefficients between the extracted watermark and 500 watermarks which are produced randomly and the 250th watermark is our original watermark.



noise addition

low-pass filtering

Resample

mp3 compression

Fig. 6. Normalized correlation coefficients after attack

From Table 1, the normalized correlation coefficients after attacks such as noise addition, low-pass filtering, requantization, resampling and mp3 compression are all above the threshold we set. Since the shape of the waveform after is similar with the

Table 1. Normalized correlation coefficient after attacking

Audio type Attack	Blues	Classical	Country	Folk	Pop
No processing	1.0000	1.0000	1.0000	1.0000	1.0000
Noise addition	0.9317	0. 9458	0.9386	0.9408	0.9817
Low-pass filtering	0.9134	0. 9249	0.9172	0.9234	0.9501
Requantization	0.9055	0.9003	0.8991	0.9155	0.9136
Resampling	0.8737	0. 8731	0.8637	0.8883	0.9064
Mp3 compression	0.8603	0. 8567	0.8559	0.8816	0.9163
TSM(-5%)	0.8270	0.9188	0.9376	0.9362	0.9600
TSM(+5%)	0.7832	0.9020	0.9344	0.9232	0.9536

original audio signal, and the audio types like Classical, Country, Folk and Pop can resist the attack of TSM±5% because the normalized correlation coefficients are all larger than 0.9. However, normalized correlation coefficients of Blues under the attack of TSM±5% are both below 0.85 and the music piece is not able to elude this challenging attack. To sum up, the zero-watermark Algorithm is robust to many attacks, such as noise addition, low-pass filtering, requantization, resampling and mp3 compression.

5 Conclusion

In this paper, we proposed a robust audio zero-watermark scheme based on LWT and chaotic modulation. The algorithm based on energy maximization guarantees that the zero-watermark which constructed by low frequency coefficients represents the significant characteristics of the audio signal, thus it is robust against general audio processing such as noise addition, low-pass filtering, requantization, resampling and mp3 compression. And after the watermark is constructed, a initial vector is obtained by chaotic modulation, so the storage space is reduced after generating the watermark. Since the constructed zero-watermark indeed is not embedded into the original audio, the contradiction between imperceptibility and robustness which are required in many applications is avoided. But the algorithm is not robust against TSM ±5% by the music type of Blues, and we will continue to research on it.

Acknowledgment

This work is supported by the National Natural Science Foundation(No. 60672070) and Ningbo Science &Technology Project (No. 2006B100067) of China.

References

[1] van Schyndel, R.G., Tirkel, A.Z., Osborne, C.F.: A Digital Watermark[A]. In: First IEEE International Image Processing Conference(ICIP 1994), vol. 2, pp. 86–90 (1994)

[2] Cano Rodriguez, G., Nakano Miyatake, M., Perez Meana, H.M.: Analysis of audio watermarking schemes[A]. In: 2nd International Conference on Electrical and Electronics Engineering (ICEEE) and XI Conference on Electrical Engineering (CIE 2005), pp. 17–20 (2005)

[3] Gurijala, A., Deller, J.R., Joachim, D.: Robustness optimization of parametric speech watermaking[A]. In: ISCAS 2006, pp. 273–276 (2006)

[4] Quan, W., Tanfeng, S., Shuxun, W.: Concept and Application of Zero-Watermark[J]. Acta Electronica Sinica 31(2), 214–216 (2003)

[5] Na, W., Xia, L.: RST Invariant Zero-watermarking Scheme Based on Matching Pursuit[J]. Chinese Journal of Electronics 15(2), 269–272 (2006)

[6] Hanqiang, C., Hua, X., Xutao, L., et al.: A zero-watermarking algorithm based on DWT and chaotic modulation[J]. In: Proceedings of SPIE, vol. 6147(16), pp. 1–9 (2006)

[7] Sweldens, W.: The lifting scheme: A construction of second generation wavelets[J]. SIAM J. Math. Anal. 29(2), 511–546 (1998)

A Modified Kernels-Alternated Error Diffusion Watermarking Algorithm for Halftone Images

Linna Tang[1], Jiangqun Ni[1,2,⋆], Chuntao Wang[1], and Rongyue Zhang[1]

[1] Department of Electronics and Communication Engineering, Sun Yat-Sen University, Guangzhou, 510275, P.R. China
[2] Guangdong Key Laboratory of Information Security Technology, Guangzhou, 510275, P.R. China
Tel.: +86-20-84036167,
issjqni@mail.sysu.edu.cn

Abstract. Digital Halftoning is the rendition of continues-tone images on two-level displays. A modified kernels-alternated error diffusion (KAE DF) watermarking algorithm for halftone images is presented in this paper, which can achieve relatively large embedding rate with good visual quality and high robustness. With the introduction of threshold modulation in error diffusion, the proposed algorithm greatly eliminates the edge sharpening and noise shaping distortion of the watermarked halftone image due to the conventional error diffusion algorithm. Consequently, more spectral distribution features in DFT domain characterized with the two alternated kernels, i.e., Jarvis and Stucki, are preserved in the resulting watermarked halftone image, which greatly improves the performance of watermark decoding. Instead of the original grey level image, the one generated with the inverse halftone algorithm is utilized to determine the local threshold for blind watermark detection. Extensive simulations are carried out, which demonstrates that the modified KAEDF watermarking algorithm achieves significant improvements in performance of visual quality and watermark decoding rate.

1 Introduction

With the popularity of Internet, the copyright protection, authentication and tamper proofing of digital media are becoming increasingly important. Thus digital watermarking has become the domain of extensive researches, among which watermarking for halftone images has drawn more attentions these years. Digital halftoning produces two-tone binary images which approximate the original continuous-tone images when viewing from a distance by the low-pass filtering in the human visual system. Halftoning is widely used in two-tone devices such as newspapers, magazines and printers. There exist several halftoning algorithms for grey level images, such as error diffusion[1]-[3], ordered dithering [4], dot diffusion [5], direct binary search [6], and etc., among which, error diffusion is the most popular one due to its good image quality and reasonable computation

⋆ Corresponding author.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 382–394, 2008.

complexity. Based on the error diffusion halftoning, a modified KAEDF watermarking algorithm is developed in this paper.

Recently numerous watermarking algorithms for halftone image have been proposed [7]-[9]. In [9], S. C. Pei and et al. find that the two kernels proposed by Jarvis and stucki work compatibly, and present a block-based error-diffused watermarking algorithm KAEDF (kernels-alternated error diffusion) for halftone images by alternating the two different kernels in the process of halftoning. For watermark extraction, the CSED (cumulative squared Euclidean distance) measure for the two different kernels in frequency domain is developed to determine whether the observed block has been processed by Jarvis or Stucki. In general, the original grey level image is required to determine the threshold T_{JS} for robust watermark decoding.

However, the halftone images watermarked with KAEDF still suffer from the edge sharpening and noise shaping distortion as the conventional error diffused algorithms do. Both the aforementioned distortions can be accurately predicted by the quantization model developed by T. D. Kite [10], where the quantizer is modeled as a linear gain plus additive noise. The incorporation of threshold modulation based on the quantizer model results in the modified error diffusion algorithm [10], which can greatly improve the visual quality of the watermarked halftone image when it is applied in the KAEDF watermarking algorithm. The elimination of noise shaping with the modified error diffusion implies that more spectral distribution features in DFT domain characterized with the two alternated kernels, i.e., Jarvis and Stucki, are preserved in the resulting watermarked halftone image. Consequently, the watermark decoding performance with the CSED approach is greatly improved. And it also enables the possibility to develop a blind watermarking algorithm for halftone images with high robustness.

Instead of the original grey level image, the one generated with the inverse halftoning technique is utilized to determine the local threshold of CSED for blind watermark detection. By combing the modified error diffusion and the inverse halftoning strategy, a new blind KAEDF watermarking algorithm for halftone images is developed in this paper. Extensive simulation results demonstrate that the modified KAEDF watermarking algorithm has significant improvements in performance of visual quality and detection robustness compared to the conventional one. The remainder of this paper is organized as follows. The modified error diffusion algorithm and its performance are presented in section 2. The modified KAEDF watermarking algorithm for halftone images is given in section 3. Simulation results and analysis are given in section 4. And section 5 draws the collusions.

2 The Modified Error Diffusion and Its Performance

The conventional error diffusion model is illustrated in Fig.1, where $x(i,j)$, $x^{'}(i,j)$ and $y(i,j)$ represent the pixel of the input gray image, the input of the quantizer, and the output halftone image, respectively; $H(Z)$ is the filter kernel controlling the proportion of error diffusing to neighbor region so as to keep the average intensity in local area; $e(i,j)$ represents the quantizer error. Although the conventional error diffusion has relatively good visual quality and low computation complexity, it still suffers from the distortion of limit cycles, edge sharpening and noise shaping.

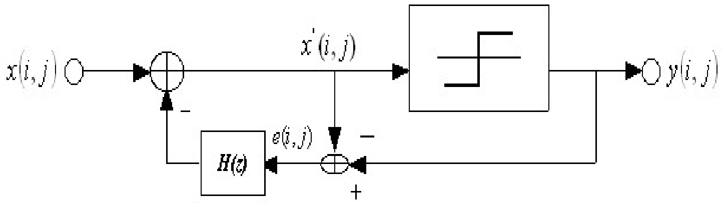
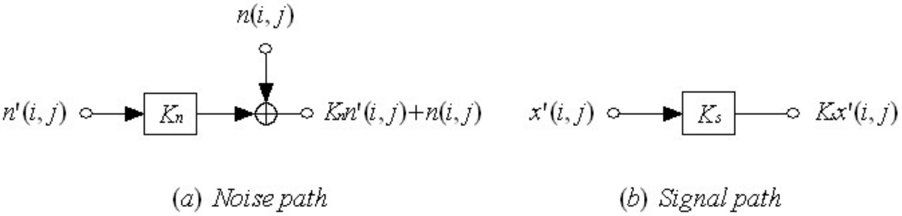


Fig. 1. Conventional error diffusion model

To analyze the distortion introduced by the conventional error diffusion, T. D. Kite and A. C. Bovik [12] presented a model of linear gain plus additive noise for the quantizer in Fig.1. Consequently, the output of the quantizer can be separated into the signal and noise components, $y_s(i,j)$ and $y_n(i,j)$, respectively. And the quantizer model is illustrated in Fig.2, where k_s and k_n are the linear gains for signal and noise components, respectively. In general, k_s is set to 1 and k_s is a scalar optimized with the input image [12]. Therefore, the output of the quantizer $Y(Z)$ can be represented as follows:

$$Y(Z) = \mathit{STF} \cdot X(Z) + \mathit{NTF} \cdot N(Z), \tag{1}$$

where *STF* and *NTF* are the signal and noise transform function, respectively, and their definitions can be found in [12].



(a) Noise path

(b) Signal path

Fig. 2. The linear gain model of the quantizer

With the linear gain model, one can easily explain the sharpening and noise shaping caused by the conventional error diffusion algorithm. According to the simulation results given in [12], the linear gain in the high frequency of *STF* is far greater than 1, which quantitatively coincides the edge sharpening inherent to error diffusion; while the sharpening distortion itself further results in the noise shaping, i.e., the residual image after quantization preserves the visible contour of the original grey level image. Thus the feasibility of the quantization model is well justified.

To decrease the effects of the edge sharpening and noise shaping with the error diffusion, a modified error diffusion algorithm is proposed in [12] by adding a multiplicative parameter L into the input of the quantizer to modulate the threshold of halftoning process, which is shown in Fig.3. In general, the L is determined such that the composite gain for the signal path is 1, thus eliminating the edge shaping and noise shaping distortion.

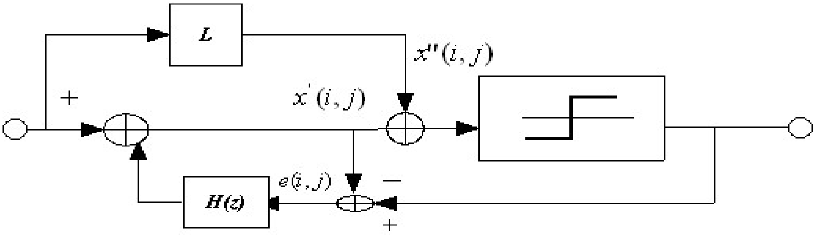


Fig. 3. The modified error diffusion

The performance of error diffusion algorithm can be well evaluated with following two measures, namely, C_{EI} and *WSNR*. The C_{EI} is defined as the correlation between the error or residue image of quantization and original continuous-tone image [11], i.e.,

$$C_{EI} = \frac{|Cov[E, I]|}{\sigma_E \sigma_I}, \tag{2}$$

The larger the correlation C_{EI} is, the worse the visual quality of the resulting halftone image would be. While the measure of *WSNR* is the weighted signal-to-noise ratio (*SNR*), which weights the SNR according the contrast sensitivity function (CSF) of the human visual system and is defined as follows [13]:

$$WSNR(dB) = 10 \log_{10} \left(\frac{\sum_{uv} |X(u,v)C(u,v)|^2}{\sum_{u,v} |(X(u,v) - Y(u,v))\, C(u,v)|^2} \right), \tag{3}$$

where $X(u,v)$ and $Y(u,v)$ represent the discrete Fourier transforms of the input image, and the output image, respectively; $C(u,v)$ is the value of CSF; M and N are the size of the input image. Our simulation results also demonstrate that, compared with the conventional one, the halftone image generated with the modified error diffusion algorithm has much lower C_{EI} and larger *WSNR* values, and thus has a much better visual quality.

3 The Modified KAEDF Watermarking Algorithm

The KAEDF is a widely used watermarking algorithm for halftone image due to its simple implementation and high robustness. However, the resulting watermarked halftone image also suffers from the edge sharpening and noise shaping distortion with the conventional error diffusion. And the algorithm requires the original continues-tone image for watermark detection. By introduction of the threshold modulation error diffusion to KAEDF, a modified KAEDF (MKAEDF) watermarking algorithm is developed in this paper. With the MKAEDF, the visual quality of the watermarked halftone image is greatly improved. On the other hand, as more spectral distribution features in DFT domain characterized with the two alternated kernels, i.e., Jarvis and Stucki, are preserved in the resulting watermarked halftone image due to elimination of noise shaping distortion, the image with inverse halftoning is possible to be used to determine the local CSED threshold for blind watermark detection.

3.1 Watermark Embedding of MKAEDF

The process of watermark embedding with MKAEDF for halftone image is given in Fig.4, where the error diffusion with threshold modulation is adopted. Considering the good compatibility of Jarvis and Stucki kernels, they are alternately used according to the to-be-embedded message in the process of image halftoning. In the proposed algorithm, the Jarvis and Stucki kernels are corresponding to the bit 0 and 1, respectively.

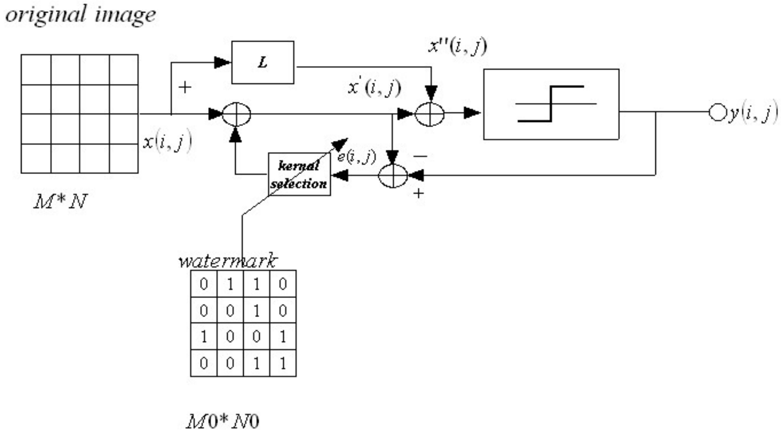


Fig. 4. MKAEDF based watermark embedding process

Assume that the size of the original grey image and the to-be-embedded watermark binary image are $M \times N$ and $M_0 \times N_0$, respectively, and the embedding process is described as follows:

1. The original grey level image is divided into $M_0 \times N_0$ blocks, each of which has $\frac{M}{M_0} \times \frac{N}{N_0}$ pixels. Assuming that (m,n) $(1 \le m \le M_0, 1 \le n \le N_0)$ denote the $(m,n)^{th}$ block, then the $(m,n)^{th}$ bit of the watermark image is correspondingly allocated to that block, and thus each block has one bit to be inserted;
2. For the $(m,n)^{th}$ block, either Jarvis or Stucki kernel is used to implement the modified error diffusion process according to the to-be-embedded bit. The

embedding process can be formulated as:

$$\begin{aligned} x(i,j) &= x^{'}(i,j) + h(i,j) \cdot e(i,j) \\ h(i,j) &= \begin{cases} Jarvis\ kernel & bit = 0 \\ Stucki\ kernel & bit = 1 \end{cases} \\ e(i,j) &= y(i,j) - x^{'}(i,j) \\ x^{"}(i,j) &= x^{'}(i,j) + L \cdot x(i,j) \\ y(i,j) &= Q(x^{"}(i,j)) \end{aligned} \tag{4}$$

In this way, the watermark bits are equivalently inserted into the resulting halftone image. With the MKAEDF, the good visual quality of the watermarked halftone image can be expected.

3.2 Blind Watermark Detection with CSED

Instead of the original continues-tone image I_0, the one obtained with inverse halftoning is used to determine the local CSED threshold T_{JS} for watermark detection. Therefore, the blind detection is achieved. Fig.5 gives the process of blind watermark detection.

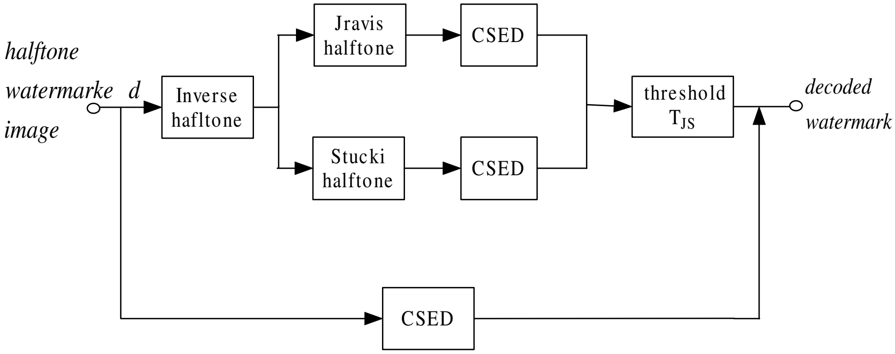


Fig. 5. The process of blind watermark detection

Inverse Halftone Process. The fast inverse halftoning algorithm in [11] is employed to recover the continues-tone image from its halftone one, which is shown in Fig.6. The inverse halftoning process includes 4 main steps: (1) computes the gradients at two scales in both the horizontal (x) and vertical (y) directions; (2) correlates the gradient estimates to give maximum output when a large gradient appears in both scales, such as at a sharp edge, which is referred as *control functions*; (3) constructs an FIR filter according the control functions, which increases the amount of smoothing in each direction as the estimated image gradient decreases; and (4) applies the FIR filter. The detailed implementation of inverse halftoning can be found in [11]. With the aforementioned steps, the employed fast algorithm [11] can recover continues-tone image with satisfactory visual quality from error diffusion halftone one at low computational cost.

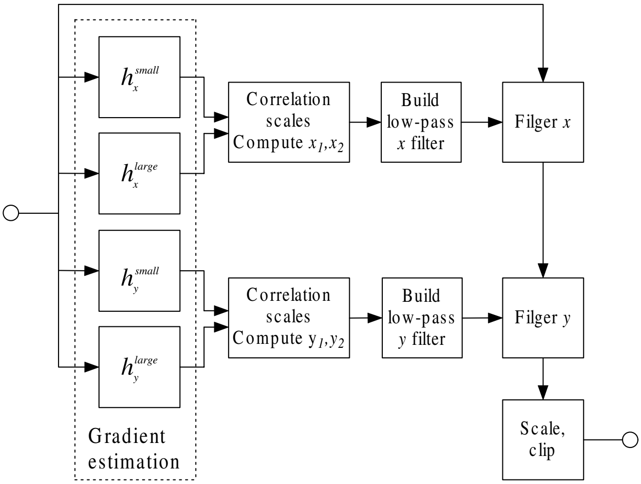

h_x^{small}
h_x^{large}
h_y^{small}
h_y^{large}
Gradient estimation
Correlation scales Compute x_1,x_2
Correlation scales Compute y_1,y_2
Build low-pass x filter
Build low-pass y filter
Filger x
Filger y
Scale, clip

Fig. 6. Fast inverse halftoning for error diffusion

Blind Watermark Detection. Assuming that the received halftone image is I_{HW} with size $M \times N$, then the watermark detection process is as follows:

1. Recover the continues-tone image I_r with the inverse halftoning algorithm described in the previous subsection;
2. Implement the halftoning transform for I_r with the Jarvis and Stucki kernels, respectively; the corresponding halftone images are I_{HI} and I_{HS}, respectively;
3. Divide the I_{HI} and I_{HS} into $M_0 \times N_0$ blocks, respectively. For the $(m,n)^{th}$ block, execute the 2-D FFT transform, among which coefficients with high energy are selected to compute the CSED values $CSED^J_{mn}$ and $CSED^S_{mn}$, i.e.,

$$CSED = \sqrt{\sum_{i=1}^{\omega}\left[\left(x_i - \frac{M_1}{2}\right)^2 + \left(y_i - \frac{N_1}{2}\right)^2\right]}, \tag{5}$$

   where (x_i, y_i) is the coordinate of the i^{th} selected pixel in the $(m,n)^{th}$ block, and $M_1 = \frac{M}{M_0}$ and $N_1 = \frac{N}{N_0}$ is the width and height of each block, respectively. Note that the value of ω is determined by experiments;
4. Calculate the threshold T'_{JS} so as to determine whether the block has been processed by Jarvis or Stucki kernel. The threshold T'_{JS} for the $(m,n)^{th}$ block is computed as follows:

$$T'_{JS} = \frac{(CSED^J_{JS} + CSED^S_{JS})}{2} \tag{6}$$

5. Divide the I_{HW} into $M_0 \times N_0$ blocks, and then calculate the CSED value $CSED_{mn}$ for the $(m,n)^{th}$ block;

6. The watermark is detected according to the following rule:

$$\begin{cases} bit = 0 & if\ CSED_{mn} > T'_{JS} \\ bit = 1 & otherwise \end{cases} \tag{7}$$

After all the bits from $M_0 \times N_0$ blocks are detected, the recovered binary watermark image is obtained.

4 Experimental Results and Analysis

In our simulation, we test 8 512 × 512 × 8 gray level images with different texture characteristics, including Barb, Boat, F16, Earth, Lake, Lena, Mandrill, and Peppers. A 16 × 16 binary image (see Fig. 8) is generated as watermark.

4.1 Visual Quality of the Watermarked Halftone Image

The binary watermark image is embedded into the 8 test images with the conventional KAEDF and the proposed MKAEDF algorithm, respectively. Fig. 7 shows the two resulting watermarked Lena images and the original one. It can be observed that the watermarked image generated with MKAEDF has better visual quality than that with KAEDF, especially in the areas with edges, such as hairs, hat, eyes and etc. The other test images also have the similar results.

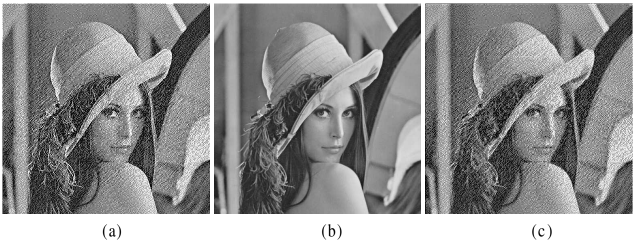


Fig. 7. The watermarked image for Lena: (a) Watermarked halftone image with the KAEDF. (b) Original continuous-tone image. (c) Watermarked halftone image with the MKAEDF.

Furthermore, the two measures, namely, *WSNR* and C_{EI} are employed to evaluate the degradation of the watermarked image against the original one. Table 1 gives the performance in visual quality for all the test images, which demonstrates that the watermarked halftone images with MKAEDF have significant improvement in visual quality over those with KAEDF.

4.2 Detection Performance without Attacks

The watermarks are directly detected from the watermarked images given in Fig. 7 with the KAEDF and MKAEDF algorithms, respectively. The extracted

Table 1. Fidelity comparison between KAEDF and MKAEDF

Image	Modified KAEDF		KAEDF	
	C_{EI}	*WSNR*	C_{EI}	*WSNR*
Barb	0.0464	27.1658	0.3978	25.4457
Boat	0.0125	28.2036	0.4027	25.6205
F16	0.0188	30.0627	0.3864	27.3849
Earth	0.0017	27.3384	0.3282	24.9975
Lake	0.0327	28.7260	0.5031	24.4529
Lena	0.0054	27.7743	0.3606	26.7116
Mandrill	0.0513	28.0420	0.3920	25.0123
Peppers	0.0134	28.5711	0.3868	26.7608
Average	0.0228	28.2355	0.3947	25.7983

images are shown in Fig.8 (b) and (d). It is observed that the proposed blind detection of MKAEDF has similar performance as the non-blind one of KAEDF, which demonstrates the robustness of the proposed algorithm.

We then have the MKAEDF with blind detection and KAEDF with non-blind detection. For impartial comparison, the detection procedure of KAEDF is also revised to adopt the same blind strategy as the MKAEDF, which is then named as *b-KAEDF* for convenience. The corresponding result is given in Fig.8 (c), which shows that MKAEDF has significant improvements in detection performance over that of b-KAEDF.

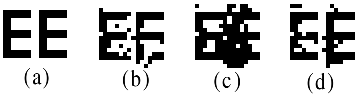


Fig. 8. Detected watermark: (a) Original watermark; (b) Detected watermark with KAEDF; (c) Detected watermark with b-KAEDF; (d) Detected watermark with MKAEDF

Moreover, Table 2 gives the detection performance comparisons with KAEDF, b-KAEDF and MKAEDF for all the test images, where the blind MKAEDF has consistently superior performance over that of b-KAEDF and similar performance with that of non-blind KAEDF. The results in Table 2 further justifies the ones observed in Fig.8.

4.3 Performance Comparison against Attacks

In this section, we further compare the detection performance of the KAEDF, b-KAEDF, and the proposed MKAEDF under attacks, which include the tampering, random noise, cropping, and etc. Here, the watermark is the 8×8 binary image shown in Fig.9(b), which is embedded into the 8 test images with the KAEDF and the proposed MKAEDF algorithms, respectively.

Table 2. Detection performance comparison for KAEDF, b-KAEDF, and MKAEDF

Image	KAEDF(%)	b-KAEDF(%)	MKAEDF(%)
Barb	91.80	75.39	91.80
Boat	94.92	83.20	90.63
F16	87.89	82.03	83.98
Earth	96.88	92.19	94.92
Lake	85.94	77.34	85.16
Lena	97.66	92.58	95.31
Mandrill	97.27	70.31	98.05
Peppers	96.09	82.42	85.16
Average	93.56	81.93	90.63

Tampering Attack. The 5% bits of the watermarked halftone images are temperedly attacked, i.e., substituted with the other binary bits. Fig.9 (a) and (b) give the attacked version of the watermarked Barb images embedded with KAEDF and MKAEDF algorithms, respectively. Then the watermarks are detected with KAEDF, b-KAEDF, and MKAEDF, which are shown in Fig.9 (d)-(f), respectively. It can be observed that, among the three watermark detection algorithms, the watermark detected with MKAEDF is the one most similar to the original, which is further justified by their corresponding decoded rates, i.e., 87.50%, 73.44%, and 90.63%.

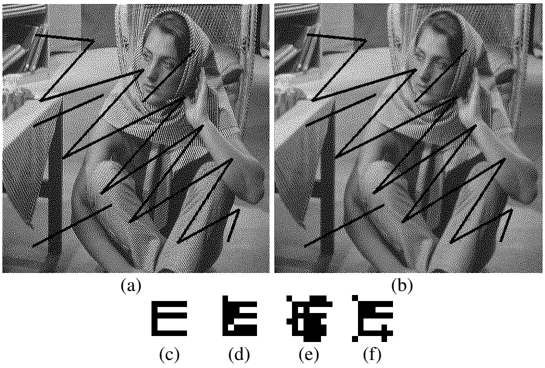


Fig. 9. Performance against tampering attack: (a) attacked watermarked image for KAEDF; (b) attacked watermarked image for MKAEDF; (c) original Watermark; (d) watermark detected with KAEDF; (e) watermark detected with b-KAEDF; (f) watermark detected with MKAEDF

For all 8 tested images, the average decoded rates for KAEDF, b-KAEDF, and MKAEDF are 83.52%, 76.17%, 83.60%, respectively, which shows that the proposed MKAEDF gains significant improvement in detection performance over b-KAEDF while has similar performance with KAEDF.

Random Noise. The halftone watermarked images are attacked with random noise whose energy is 5% of the total energy of the corresponding watermarked image. The attacked Barb images watermarked with KAEDF and MKAEDF are shown in Fig.10 (a) and (b), respectively. And the watermarks detected with KAEDF, b-KAEDF, and MKAEDF are given in Fig.10 (d)-(f). Their decoded rates are 100%, 73.44%, and 95.31%, respectively, which shows that the non-blind KAEDF algorithm and the proposed blind one can counter the 5% noise attack while the b-KAEDF one fails.

For all 8 test images, the average decoded rate for KAEDF, b-KAEDF, and MKAEDF are 97.66%, 81.84%, and 93.76%, respectively, which implies that the proposed algorithm has comparable performance with the non-blind KAEDF whereas it obtains considerable gains in detection performance over b-KAEDF.

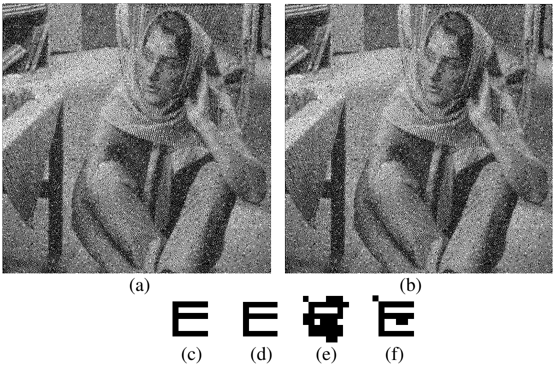


Fig. 10. Performance against random noise: (a) attacked watermarked image for KAEDF; (b) attacked watermarked image for MKAEDF; (c) original Watermark; (d) watermark detected with KAEDF; (e) watermark detected with b-KAEDF; (f) watermark detected with MKAEDF

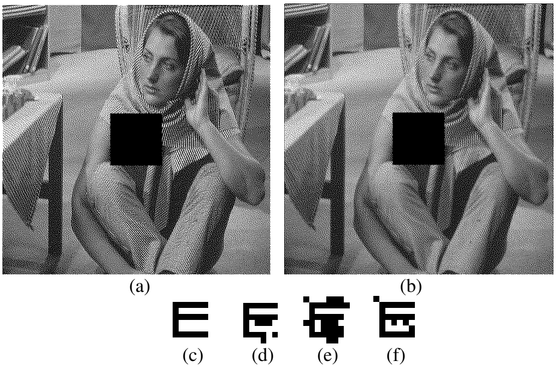


Fig. 11. Performance against cropping: (a) attacked watermarked image for KAEDF; (b) attacked watermarked image for MKAEDF; (c) original Watermark; (d) watermark detected with KAEDF; (e) watermark detected with b-KAEDF; (f) watermark detected with MKAEDF

Cropping. A 100*100 block is cropped from the center of the halftone watermarked images, and the cropped Barb images embedded with KAEDF and MKAEDF are shown in Fig.11 (a) and (b), respectively. Then the watermarks are detected with KAEDF, b-KAEDF and MKAEDF, which are shown in Fig.11 (d), (e), and (f). Their decoded rates are 92.19%, 76.56%, and 93.75%, respectively, which shows that the watermark detected with MKAEDF has the best performance.

For all 8 test images, the average decoded rates for KAEDF, b-KAEDF, and MKAEDF are 94.93%, 82.23%, and 91.21%, respectively, which implies that MKAEDF obtains considerable improvements in detection performance over the b-KAEDF while it has comparable performance with KAEDF.

5 Conclusion

In this paper, a modified KAEDF watermarking algorithm for halftone images is proposed, which can achieve relatively large embedding rate with good visual quality and high robustness. The modified KAEDF watermarking algorithm is developed by introducing the threshold modulation for error diffusion, which greatly reduces the edge sharpening and noise shaping distortion of the watermarked halftone image and enhance the robustness. Consequently, as more spectral distribution features in DFT domain characterized with the two alternated kernels, i.e., Jarvis and Stucki, are preserved in the resulting watermarked halftone image, the image generated with inverse halftoning is utilized to determine the local threshold for blind watermark detection. Simulation results demonstrate that the proposed modified KAEDF halftone watermarking algorithm achieves significant improvements in performance of visual quality and watermark decoded rate.

Acknowledgments

The authors appreciate the supports received from NSFC (60773200) and NSF of Guangdong (7003722).

References

1. Floyd, R., Steinberg, L.: An adaptive algorithm for spatial grayscale. Proc. Society for Information Display 17(2), 75–77 (1976)
2. Jarvis, J.F., Judice, C.N., Ninke, W.H.: A survey of techniques for the display of continuous-tone pictures on bilevel displays. Computer Graphics and Image Process 5(1), 13–40 (1976)
3. Stucki, P.: MECCA - a multiple error correcting computation algorithm for bi-level image hard copy reproduction. Research report RZ1060, IBM Research Laboratory, Zurich, Switzerland (1981)
4. Ulichney, R.A.: Dithering with blue noise. Proc. IEEE 76(1), 56–79 (1987)
5. Knuth, D.E.: Digital halftones by dot diffusion. ACM Trans. Graph. 6(4), 245–273 (1987)

6. Seldowitz, M.A., Allebach, J.P., Sweeney, D.E.: Synthesis of digital holograms by direct binary search. Appl. Opt. 26(14), 2788–2798 (1987)
7. Fu, M.S., Au, O.C.: Data hiding by smart pair toggling for halftone images. In: Proc. IEEE Int. Conf. Acoustics. Speech and Signal Processing, vol. 4(6), pp. 2318–2321 (2000)
8. Fu, M.S., Au, O.C.: Data hiding watermarking for halftone images. IEEE Trans. Image Processing 11(4), 477–484 (2002)
9. Pei, S.C., Guo, J.M.: Hybrid pixel-based data hiding and block-based watermarking for error-diffused halftone images. IEEE Trans. Circuits and System 13(8), 867–884 (2003)
10. Knox, K., Eschbach, R.: Threshold modulation in error diffusion. J. Elect. Imaging 2(7), 185–192 (1993)
11. Kite, T.D., Damera-Venkata, N., Evans, B.L., Bovik, A.C.: A fast, high-quality inverse halftoning algorithm for error diffused halftones. IEEE Trans. Image Processing 9(9), 1583–1592 (2000)
12. Kite, T.D., Evans, B.L., Bovik, A.C.: Modeling and quality assessment of halftoning by error diffusion. IEEE Trans. on Image Proc. 9(5), 636–650 (2000)
13. Mista, T., Varkur, K.: Evaluation of contrast sensitivity functions for the formulation of quality measures incorporated in halftoning algorithms. In: Proc. IEEE Inf. Conf. Acoustics, Speech, Signal Processing, vol. 5(4), pp. 301–304 (1993)
14. Knox, K.: Error image in error diffusion. Proc. SPIE Image Proc, Alg. Tech. 1657(2), 268–279 (1992)
15. Damer-venkata, N., Kite, T.D., Geisler, W.S., et al.: Image quality assessment based on a degradation model. IEEE Trans. on Image Proc. 9(4), 636–650 (2000)

On the Complexity of Obtaining Optimal Watermarking Schemes

Julien Lafaye

Cnam - Cédric
292 rue Saint Martin - CC 432
75141 PARIS Cedex 3, France
julien.lafaye@cnam.fr

Abstract. In this paper, we try to answer the question: is the task of obtaining optimal watermarking schemes computationally feasible ? We introduce a formalism to express watermarking algorithms over finite domains of contents and exploit it to study two aspects of watermarking: the *relevance* of the detection process and its *robustness*. We formulate the search for an optimal scheme as an optimisation problem. We prove that (1) finding the algorithm which has the highest relevance for a fixed robustness against a known attack is NEXP-complete relatively to the domain encoding size and that (2) finding the algorithm which has the highest robustness for a fixed relevance is NP-complete relatively to the cardinality of the application domain.

1 Introduction

Watermarking Watermarking is an information hiding technique which enables the embedding of marks within digital media. First use of watermarking date back to the 13th century [9] but the interest in these methods has been revived by the spread of digital data through computer networks. Despite watermarking can be used in various applications, we focus here on Intellectual Property Protection (IPP). In this case, a copyright mark is embedded within the data: the capability to recover the mark can be part of a proof of ownership. A watermarking scheme consists of a pair of algorithms. The *embedding* algorithm takes as input a content to be watermarked, some secret owned by the data owner and outputs a watermarked content. The *detection* algorithm takes as input a suspect content, a secret and outputs Yes (the document contains a watermark) or No (the document does not contain a watermark). To be relevant, the detection process must output Yes only when the secret used for detection is the one previously used for embedding. We focus here on this kind of watermarking, called 0-bit watermarking, but many variations exist (see e.g. [9]). Following Kerckhoffs' recommendations [10], watermarking schemes are usually public. The security of the method relies on the secret whose disclosure must be avoided.

Context. First watermarking techniques were designed for still images but were rapidly adapted to handle other kind of media: videos, sound samples, printed

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 395–409, 2008.

text, natural language, source code, databases. Despite the fact that different watermarking techniques are used for this wide range of applications, they share common properties. First, the presence of the mark should not lower the quality of the original content. Second, the target application, here IPP, requires that embedded watermarks are *robust* (it is difficult for a malicious attacker to remove the watermark). Third, the detection process must be *relevant* (watermarks are not detected in random objects).

Trading-off Robustness for Relevance. A common objection [7] often formulated against watermarking is that existing schemes are robust ... until a new attack is discovered. Obviously, such attacks appear on a regular basis. First watermarking schemes for images could not cope with compression of an image, mosaicing or rotation attacks. But researchers and engineers constantly design new and more robust schemes making watermarking more and more relevant to practical situations. One could question whether the loop new attack/new scheme will eventually stop ? Apart from designing innovative watermarking schemes, a known method to improve the robustness of an existing scheme is to make its detection process more tolerant. For instance, for some algorithms, watermark detection is performed by calculating the correlation between the data and the secret and comparing the value obtained with a predefined threshold. Lowering the threshold allows for better robustness since watermarks are then detected in more altered documents. But doing this also entails the detection of watermarks in a larger set of documents. Then, the risk is to raise false detections, i.e. detection of a mark in a document where there is none. There is an intrinsic and unavoidable trade-off between the *robustness* of the method and the *relevance* of the detection process. This trade-off has been extensively discussed in various contexts (see e.g. [3,12,13]). Then, the comparison of two schemes must take into account at least *robustness* and *relevance* levels. In this paper, we consider as *optimal* a watermarking scheme if there is no other scheme that beats it on these two levels. There are usually several optimal schemes since gaining on one level often implies loosing on the other one.

Contribution. We investigate the difficulty of obtaining optimal schemes from a computational complexity viewpoint. As far as we know, no previous work adopted this approach even if optimal schemes, in a broad sense, motivated some research works [3,12,14]. To achieve this, we consider a finite domain $\mathcal{X}$ where original and watermarked data can reside. We adopt an abstract approach that considers a watermarking scheme $\mathcal{W}$ as pair $(\mathcal{M}, \mathcal{D})$ of mappings: $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{X}$ and $\mathcal{D} : \mathcal{X} \rightarrow \{0, 1\}$ such that $\mathcal{D}(\mathcal{M}(x)) = 1$ for all x and such that x and $\mathcal{M}(x)$ are similar. $\mathcal{M}$ is the embedder and $\mathcal{D}$ is the detector. The *relevance* of a scheme is defined as $1 - p_f$ where p_f is the probability that a false positive occurs during detection. The *robustness* is defined as $1 - p_{sa}(\mathcal{A})$ where $p_{sa}(\mathcal{A})$ is the probability that a known attack $\mathcal{A}$ removes the watermark. We formulate the search for optimal schemes as a bi-objective optimisation problem: *Find the mappings* $\mathcal{M}, \mathcal{D}$ *which minimise* p_f *and* $p_{sa}(\mathcal{A})$. The optimal watermarking schemes are to be found among the 'best' solutions of this optimisation problem. In multi-objective optimisation theory, such solutions are called efficient and constitute the so-called Pareto frontier of the optimisation problem. The exploration of the Pareto frontier is generally a very difficult task [17] so we focus on two sub-problems: FP and PSA. The problem FP consists of deciding whether there exists a watermarking scheme with a probability of false positive under a threshold k while having a given robustness. The dual problem PSA is similarly defined for a fixed probability of false positive. We prove that FP is NEXP-complete when complexity is measured relatively to the domain encoding size (i.e. the logarithm of its cardinality) and that PSA is NP-complete when complexity is measured relatively to the cardinality of $\mathcal{X}$. The distribution and similarity relationships of contents are central in our proofs, characterising the fact thatcomplexities of FP and PSA problems highly rely on them.

Organisation. The paper is organised as follows. After introducing the model in Section 2, we present some aspects of computational complexity on succinct instances in section 3. In Section 4, we prove NEXP-completeness of the FP problem whereas we prove NP-completeness of the PSA problem in Section 5. Section 6 presents the related works, Section 7 concludes.

2 Model

2.1 Data Model

Let $\mathcal{X}$ be the domain of the contents to be watermarked. Elements from $\mathcal{X}$ are denoted by small letters like $x, y, \ldots$. We denote by $p_{\mathcal{X}}(x)$ a probability mass function on $\mathcal{X}$. For now, we do not impose any restriction on the probability. Not being bound to some specific distribution, i.e. gaussian or uniform, enables to capture more realistic situations. When the domain $\mathcal{X}$ is finite, we call *encoding size* of $\mathcal{X}$ the number n of bits required to enumerate all the elements of $\mathcal{X}$.

If $\mathcal{X}$ is the set of images that have 512×512 pixels, each of them having a colour taken from a $256 = 2^8$ gray levels palette, the dimension of $\mathcal{X}$ is $9 \times 9 \times 8 = 648$.

We also suppose that there exists some similarity predicate $\sigma : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ that is two elements x and y are considered as similar when $\sigma(x, y) = 1$. We suppose that σ is symmetric. If there is a natural distance d on $\mathcal{X}$, a similarity predicate can be obtained by thresholding d, i.e. $\sigma(x, y) = 1 \Leftrightarrow d(x, y) \leq \tau$.

2.2 Watermarking Model

Now, we define a watermarking scheme as a pair of mappings which comply with specific constraints.

Definition 1. *If $\mathcal{X}$ is a domain with encoding size n and σ a similarity predicate on $\mathcal{X}$, a σ-preserving watermarking scheme $\mathcal{W}$ on $\mathcal{X}$ is a pair of functions $(\mathcal{M}, \mathcal{D})$ such that:*

- $\mathcal{M} : \mathcal{X} \to \mathcal{X}$,
- $\mathcal{D} : \mathcal{X} \to \{0, 1\}$,
- $\mathcal{D} \circ \mathcal{M}(x) = 1$ *for all* $x \in \mathcal{X}$,
- $\sigma(\mathcal{M}(x), x) = 1$ *for all* $x \in \mathcal{X}$,
- *both* $\mathcal{M}$ *and* $\mathcal{D}$ *are efficiently computed that is there exists some* $q \in \mathbb{N}$ *such that the computation of* $\mathcal{M}$ *and* $\mathcal{D}$ *on* $i \in \mathcal{X}$ *requires at most* $(\log_2 |\mathcal{X}|)^q = n^q$ *steps.*

$\mathcal{M}$ *is called the embedding function and* $\mathcal{D}$ *the detection one.*

2.3 Properties

Once a watermarking scheme is defined, one should measure its *relevance* and *robustness.* In this paragraph, we give their definitions.

Relevance. To be relevant, a watermarking scheme much achieve a low false positive rate: the probability that a watermark is detected in a random document is small. We note p_f this probability. The lower p_f is, the higher the relevance of the scheme.

Definition 2 (False Positive Probability). *We define the probability* p_f *that a false positive occurs as:*

$$p_f = p_{\mathcal{X}}(\mathcal{D}(x) = 1)$$

Robustness. There is no watermarking scheme that can cope with any kind of attack. At least, the inversion of the watermarking process is likely to output a non-watermarked element. So, robustness is a relative notion that imposes to take into account a specific attack or some subset of attacks. As in [12,14], we restrict the analysis to some known attack $\mathcal{A}$. So, we consider an attack $\mathcal{A} : \mathcal{X} \to \mathcal{X}$ that is characterised by its transition probability $p_{\mathcal{A}}(x|y)$, i.e. the probability that $\mathcal{A}$ outputs x given y as input.

Definition 3 (Successful Attack Probability). *We define the probability of successful attack* $p_{sa}(\mathcal{A})$ *under the attack* $\mathcal{A}$ *as follows:*

$$p_{sa}(\mathcal{A}) = 1 - E\left(\mathcal{D}\left(\mathcal{A}\left(\mathcal{M}\left(x\right)\right)\right)\right).$$

So, $p_{sa}(\mathcal{A})$ is one minus the probability that the detector outputs 1 after $\mathcal{A}$, averaged on all elements of $\mathcal{X}$.

3 Computational Complexity on Succinct Instances

3.1 Classical Graph Problems

We recall two classical decision problems of complexity theory that are known to be NP-complete [6].

Vertex Cover. An instance of the Vertex Cover problem (VC) is a couple (G, k) where $G = (V, E)$ is a graph and $k \in \mathbb{N}$. The problem is to decide whether G admits a vertex cover of size at most k, i.e. a subset $V' \subset V$ with $|V'| \leq k$ such that for each edge $(u, v) \in E$, at least one of u and v belongs to V'.

Dominating Set. An instance of the Dominating Set problem (DS) is a couple (G, k) where $G = (V, E)$ is a graph and $k \in \mathbb{N}$. The problem is to decide whether G admits a dominating set of size at most k, i.e. a subset $V' \subset V$ with $|V'| \leq k$ such that for each $u \in V \setminus V'$ there is $v \in V'$ such that (u, v) is an edge of G.

3.2 Complexity of Graph Problems on Succinct Presentations

As we will see in the next section, the problem FP, presented in the introduction, on a set $\mathcal{X}$ is linked with the complexity of an NP-complete problem on a graph which has a number of nodes exponential in $n = \log_2 |\mathcal{X}|$. If edges are described using an adjacency matrix, i.e. a matrix (m_{ij}) such that $m_{ij} = 1$ if and only if there is an edge between i and j, the size of of the description of G is exponential in n. Some graphs show a regular structure which enables them to have a succinct description whose size is logarithmic in the number of their nodes. By succinct description, we mean polynomial size boolean circuits [19]. Complexity being defined on the size of the input, reducing the number of bits to describe the input increases complexity. We will see that in the case of the FP problem, this actually exponentiates complexity.

Definition 4 (Succinct instances). *Let $G = (V, E)$ a graph with 2^n nodes. A* succinct presentation *of G is a boolean circuit C_G with $2.n$ inputs, one output and a polynomial number of gates*[1] *which computes the adjacency matrix of G. If i and j are two nodes of G, there is an edge between i and j if and only if C_G outputs 1 given the (binary representations) of i and j as inputs.*

If G has 2^n nodes, n bits are required to code the nodes numbers, hence the number of inputs of C_G. Such circuits are particularly useful to describe very large scale integrated circuitry. The class of decision problems decidable by this type of devices is called NC^1 [15]. The class NC^1 can be thought of as the class of problems that can be efficiently be solved on parallel computers.

As pointed out by Papadimitriou and Yannakakis [16], considering the complexity of a problem Π on succinct instances exponentiates the complexity provided a *projection* from SAT to the ordinary problem Π exists. We say that a mapping π from a language $L \subset \{0, 1\}^*$ to another M is a projection if the following properties hold. For any string x of length n,$\pi(x)$ has length n^k, where k is constant.There is a PTIME algorithm A such that, given a (binary presentation of) an integer $i \leq n^k$, A computes one of the three things: either (a) the value (0 or 1) or the ith bit of $\pi(x)$, or (b) an integer $j \leq n$, such that the ith bit of $\pi(x)$ is the jth bit of x, or (c) an integer $j \leq n$, such that the ith bit of $\pi(x)$ is one minus the jth bit of x. Lemma 1 [16] shows that when such projections

[1] The input arity of each gate is 1 or 2.

exist, the complexity on succinct instances is exponentiated compared to the ordinary descriptions.

Lemma 1. *[16] Let Π be a property of graphs, such that there is a projection from SAT to the ordinary problem for Π. Then the problem Π for succinct graphs is NEXP-hard.*

4 The FP Problem

4.1 Definition

The decision problem FP if the problem of deciding whether there exists a preserving scheme with a false positive occurrence probability below some predefined threshold. It is first considered in a non-adversarial setting, i.e. $\mathcal{A} = Id$, and with a uniform probability on $\mathcal{X}$. It is formalised as follows:

FP	**Instance** an integer n, a domain $\mathcal{X}$ such that $\lvert\mathcal{X}\rvert = 2^n$, a similarity predicate σ encoded as a circuit C_σ on $\mathcal{X}$, optimisation bound $k \in \mathbb{N}$. **Question** Is there a σ-preserving scheme on $\mathcal{X}$ such that $p_f \leq k/2^n$?

Theorem 1. *FP is NEXP-complete according to the domain encoding size.*

4.2 Outline of the Proof

The outline of the proof is the following:

1. Show that VC on succinct instances is NEXP-hard (and therefore NEXP-complete) using the projection in the sense of Skyum and Valiant [18] and a note from Papadimitriou [16]. The projection is inspired by the classical 3SAT to VC reduction [6].
2. Show that DS on succinct instances is NEXP-complete by polynomial time reduction from VC. The reduction is inspired by the classical reduction from VC to DS [6].
3. Show that FP is NEXP-complete by polynomial time reduction from DS.

4.3 Proof of Theorem 1

The first two steps are technical and therefore presented in Appendix A. The essence of the proof resides in the third step. Its idea is to reduce a DS instance to an FP instance in which elements are similar if there are linked in the DS instance and to define the set of watermarked items as the dominating set, and conversely.

FP is in NEXP. Suppose that a non-deterministic algorithm guesses a watermarking scheme $\mathcal{M} = (\mathcal{M}, \mathcal{D})$ on $\mathcal{X}$ such that $\mathcal{M}$ and $\mathcal{D}$ require at most n^q steps to be evaluated. Then, checking that $\mathcal{W}$ is σ-preserving requires enumerating the 2^n elements i of $\mathcal{X}$, computing for each of them $\mathcal{M}(i)$ and checking whether $C_\sigma(\mathcal{M}(i), i) = 1$ takes at most $2^n(n^q + n^t)$ operations i.e. exponential time. In the same manner, computing p_f can be done in exponential time. FP is in NEXP.

FP is NEXP-hard. Given a instance (G, k) of DS on succinct instances, we construct an instance (C_σ, k) of FP by defining $C_\sigma = C_G$. This trivial reduction is clearly polynomial.

If FP has a solution, DS has a solution. Let $\mathcal{W} = (\mathcal{M}, \mathcal{D})$ a σ-preserving watermarking scheme solution to FP. We define V' as the set of elements i such that $\mathcal{D}(i) = 1$. For each node j in the graph, $C_\sigma(j, \mathcal{M}(j))$ so there is a node between j and $\mathcal{M}(j)$ in G and $\mathcal{M}(j)$ is in V'. V' is a dominating set of size at most k.

If DS has a solution, FP has a solution. Suppose now that G has a dominating set V' of size at most k. We define the algorithm `isInDominatingSet` which takes as input (the binary presentation of) an element i and checks whether it belongs to V' using nested `if..else` conditions. For the example below, we arbitrarily assumed that the element 0 is in V' but not 1:

```
isInDominatingSet(i)
  If first bit of i is 0 then
    If second bit of i is 0 then
      ....
        if nth bit of i is 0 then
          return 1;
        else
          return 0;
      ....
    else
  else
    ...
```

The execution of `isInDominatingSet` requires n comparisons i.e. is linear in the size of the input. From this algorithm, we can easily construct the detection function $\mathcal{D}$ and the embedding function by replacing the $0-1$ output by the index of the smallest j such that $C_\sigma(i, j)$. The scheme $(\mathcal{M}, \mathcal{D})$ is then clearly a σ-preserving watermarking scheme with false positive probability smaller than $k/2^n$ (k elements i such that $\mathcal{D}(i) = 1$ among 2^n). □

4.4 General Setting

Considered in a non-adversarial setting, i.e. an adversarial setting in which $\mathcal{A} = Id$, and with uniform probability on $\mathcal{X}$, FP is already NEXP-complete. So is the general problem.

5 The PSA Problem

In this section, we study the problem PSA of deciding, for a fixed p_f^0, whether there exists some scheme such that $p_f = p_f^0$ and whose probability of successful attack is below some given rational value. We show in Theorem 2 that PSA is NP-complete in the number of elements of $\mathcal{X}$ which is exponential in the domain encoding size. We note $N = |\mathcal{X}|$. In particular, objects of size polynomial in N are allowed as inputs of decision problems. The proof is by reduction from a slight modification of the famous KNAPSACK problem [6]. We call Constant KnapSack (CKS) this variant.

5.1 Definition

If k is a rational number between 0 and 1, PSA is a decision problem that can be stated as follows:

PSA

Instance a domain $\mathcal{X}$, an attack $\mathcal{A} : \mathcal{X} \to \mathcal{X}$, $x \in [0,1]$, a similarity predicate σ, an optimisation bound $k \in \mathbb{Q}$, a fixed value p_f^0.

Question Is there a σ-preserving scheme on $\mathcal{X}$ such that $p_{sa}(\mathcal{A}) \leq 1 - k$ and $p_f = p_f^0$?

Theorem 2. *PSA is NP-complete according to the domain cardinality.*

5.2 Constant Knapsack

Given a finite set of objects I, each of them having a cost $c(i)$ and a weight $w(i)$, the KNAPSACK problem consists of finding the set of items that has the highest cost while fitting in a bag (its weight is less than k). In our variant CKS, we search the set of items that has the highest cost while exactly filling the bag. As a minor variation on the KNAPSACK [6] problem, CKS is NP-complete.

CKS

Instance A set of items I, a weight function w, a cost function c, a weight W, a rational number k.

Question Is there a subset of items I' whose weight is exactly W and whose cost is $c(I') \geq k$.

The idea of the proof of Theorem 2 is to construct a watermarking problem PSA such that p_f and $1 - p_{sa}$ are respectively the weight and the cost of a set of items in a CKS problem. In a CKS problem, each time an item i is put into the bag, the corresponding weight and cost are added. In the PSA problem, setting $\mathcal{D}(i)$ to 1 for an element does not contribute directly to p_{sa} and makes the analogy with CKS difficult. It modifies the probability of successful attack on each element j such that $p_{\mathcal{A}}(i|j) \neq 0$. So evaluating the impact on p_{sa} requires taking into account all the other elements of $\mathcal{X}$. To overcome this difficulty, we define a specific attack which enables a PSA problem to simulate a CKS problem. This is achieved by setting $p_{\mathcal{A}}(j|j)$ to $c(j)$ and giving equal values to $p_{\mathcal{A}}(i|j)$ for $i \neq j$. Details are given below.

5.3 Proof of Theorem 2

PSA is in NP. If $\mathcal{W} = (\mathcal{M}, \mathcal{D})$ is a watermarking scheme, computing $p_{sa}(\mathcal{A})$ takes time $O(N^3)$, computing p_f is $O(N)$ and checking whether it is a σ-preserving scheme is $O(N)$. Hence, checking whether $\mathcal{W}$ is a solution to PSA is polynomial. The decision problem PSA is in NP.

Reduction. If (I, w, c, W, k) is an instance of KS, we construct an instance $(\mathcal{X}, \mathcal{A}, \sigma, k, p_f^0)$ of PSA. We define $\mathcal{X} = I$, $p_{\mathcal{X}}(i) = w(i)$, $p_f^0 = W$. We define σ to be the constant 'true' predicate: $\sigma(i,j) = 1, \forall i, j$. We define: $\beta_j = c(j)$, $\alpha_j = 1 - c(j)/(N-1)$ and the attack $\mathcal{A}$ as follows:

$$p_{\mathcal{A}}(i|j) = \begin{cases} \alpha_j \text{ when } i = j, \\ \beta_j \text{ otherwise.} \end{cases}$$

To be a valid definition of an attack, we must have $\sum\limits_i a_{ij} = \sum\limits_i p_{\mathcal{A}}(i|j) = 1$. This relation holds here. A calculation not detailed here shows that $p_{sa}(A) = 1 - \sum\limits_j \mathcal{D}(j)\beta_j$. Hence, defining the object j as watermarked through $\mathcal{D}(j) = 1$ directly decreases p_{sa} of the quantity β_j. Then the cost C of items in a bag is such that $C \geq k \Longleftrightarrow p_{sa}(\mathcal{A}) \leq 1 - k$.

If CKS has a solution, PSA has a solution. Let I' a set of items solution to CKS. Then, $\sum\limits_{i \in I'} w(i) = W$ and $\sum\limits_{i \in I'} c(i) \geq k$. We define $\mathcal{D}$ and $\mathcal{M}$ as follows:

$$\mathcal{D}(i) = \begin{cases} 1 \text{ if } i \in I', \\ 0 \text{ otherwise.} \end{cases} \qquad \mathcal{M}(j) = \begin{cases} 1 \text{ if } j \in I' \\ min\{i | i \in I'\} \text{ otherwise.} \end{cases}$$

Since σ is the constant true predicate, $(\mathcal{M}, \mathcal{D})$ is a σ-preserving watermarking scheme. Furthermore, $p_f = \sum p_{\mathcal{X}}(i).\mathcal{D}(i) = \sum_{I'} w(i) = W = x$ and $p_{sa} = 1 - \sum c(i) \leq 1 - k$. $\mathcal{W} = (\mathcal{M}, \mathcal{D})$ is a solution of PSA.

[If PSA has a solution, CKS has a solution.] Let $\mathcal{W} = (\mathcal{M}, \mathcal{D})$ a watermarking scheme solution of PSA. We define $I' = \{i | \mathcal{D}(i) = 1\}$. Then $\sum\limits_{i \in I'} w(i) = \sum \mathcal{D}(i) p_{\mathcal{X}}(i) = p_f^0 = W$. Furthermore, $\sum\limits_{i \in I'} c(i) = 1 - p_{sa}(\mathcal{A}) \geq k$. I' is a solution of CKS. □

5.4 Remarks

PSA has been proven to be NP-complete when complexity is defined relatively to the cardinality of $\mathcal{X}$ which is exponential in the domain encoding size. One may be tempted to affirm NEXP-completeness when complexity is measured relatively to the logarithm of its cardinality but we could not find any formal proof of it. This drives us to the point where we should discuss what must be the parameter through which complexity must be defined. For images, the size of the image, i.e. its number of pixels must clearly be this parameter. In other contexts, it might be more relevant to focus on the cardinality of the domain. It is the case e.g. for watermarking English words where a sequence of letters must be mapped to another similar sequence of letters. Unfortunately, not all sequences of letters are valid, only the ones belonging so some dictionnary. The important parameter is the number of words that can be manipulated.

6 Related Work

There is a huge literature on what are the good properties that should be awaited from a watermarking algorithm. Kalker [8] proposed to distinguish between *robustness* and *security*. According to him, *security refers to the inability to have access to the raw watermarking channel* and robustness refers to the fact that *the capacity of the watermark channel degrades as a smooth function of the degradation of the marked content.* The two concepts are related: if security is not high enough, access to the raw watermarking channel can be used to degrade the watermark. In our abstract approach, only robustness is taken into account, mainly because a security study requires a precise and concrete watermarking algorithm while we only keep an abstract model of it. Security aspects of watermarking are a very intense and promising research area [2,4].

Robustness has been very soon identified as one of the most challenging issue of watermarking. The works on designing more and more robust algorithms are numerous. A great breakthrough was the use of spread-spectrum techniques [5]. Theoretical works on watermarking robustness are made difficult because they require some model of the attacker. Moulin and Sullivan [14] used information theory to model watermarking as the communication of an hidden message through a distorted communication channel. The trade-off between the rate of information hiding (or the number of times the watermark can be inserted, directly correlated to robustness) and the allowed distortion for embedding and attacking is explored. The optimal embedding strategy consists of maximising the capacity while knowing that the attacker is trying to minimise it. Chen et al. [3] followed a similar approach to design Quantisation Index Modulation (QIM). In both works, the optimal watermarking algorithm is defined as the one maximising the capacity of the channel under embedding and attack distortion constraints. In some sense, this approach is a FP problem, i.e. optimisation of the reliable transfer rate of the embedded mark under a bounded attack. Both point out the difficulty of optimisation in a general setting but provide constructive schemes in special case: gaussian channels and small distortions for [14], gaussian noise and mean-squared error constrained attack for [3].

In [12], Merhav and Sabbag studied a problem very similar to FP. Assuming p_f and the distortion of the scheme upper bounded, they show how to optimise the detection process so that it is as robust as possible. The restrictive assumption that the detector is only based on the empirical joint distribution between watermark and suspect signals is made. They show how to derive the associated 'optimal' embedding. They point out an exponential complexity in the length of the host data, to obtain this optimum. Expressed in terms of the number of possible host data, this complexity is linear. This shows that, in certain circumstances (here, some specific form of detection and embedding) an optimal scheme can be computed in polynomial time.

In our approach, we considered that the attacker is only allowed one modification of the watermarked object. This is a practical but limited protocol. If the attacker is able to try several attacks (as in [11]) and combine the results, he may be more likely to erase the watermark. There is a requirement to take into account a richer model of attacks to define robustness. In [1], Katzenbeisser et al. propose a formal framework in which a more general definition of robustness can be given. Their idea is to consider a scheme as robust if there is no tractable (i.e. polynomial time in terms of the length of the secret used for watermarking) attack that breaks the watermark. Using basic notions similar (false positive occurrence probability and probability of successful attack) to ours, they consider the computational complexity of the attacks whereas we consider the computation complexity of obtaining an optimal scheme for a specific attack.

7 Conclusion and Future Work

In this paper we presented a formal framework through which we defined the search for the best watermarking problem as a multi-objective optimisation problem. After pointing out that obtaining efficient solutions of this optimisation problem is a difficult computational task, we considered two sub-problems. We proved that deciding whether there exists a scheme with a false positive occurrence below some predefined threshold and a fixed probability of successful attack is NEXP-complete when complexity is measured relatively to domain encoding size. We also showed that the dual problem where the probability of successful attack is fixed is NP-complete when complexity is measured to the cardinality of the domain.

We are working on using this framework so that it can handle a richer model of robustness. Extensions envisioned are to take into account classes of attacks and to measure the robustness of a scheme by, e.g. the maximum probability of a successful attack for all the attacks in this class. Another challenge would be to be able to perform the same kind of analysis for an attack that is not explicitly known but on which we have only some information, e.g. its computational complexity of execution or the fact that it is σ-preserving for some known σ.

References

1. Adelsbach, A., Katzenbeisser, S., Sadeghi, A.-R.: A computational model for watermark robustness. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437. Springer, Heidelberg (2007)
2. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: Theory and practice. IEEE Transactions on Signal Processing 53(10), 3976–3987 (2005)
3. Chen, B., Wornell, G.W.: Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory 47(4), 1423–1443 (2001)

4. Comesaña, P., Pérez-González, F., Balado, F.: Watermarking security: a survey. IEEE Transactions on Signal Processing 54(2), 585–600 (2006)
5. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann Publishers, Inc., San Francisco (2001)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness, Apr. 1979. W.H. Freeman & Company, New York (1979)
7. Herley, C.: Why watermarking is nonsense. IEEE Signal Processing Magazine 19(5), 10–11 (2002)
8. Kalker, T.: Considerations on watermarking security. In: Proceedings of the IEEE Multimedia Signal Processing Workshop, IIHMSP 2006, pp. 201–206 (2001)
9. Katzenbeisser, S., Petitcolas, F.A.: Information hiding, techniques for steganography and digital watermarking. Artech house (2000)
10. Kerckhoffs, A.: La cryptographie militaire. Journal des sciences militaires IX, 5–38 (1883)
11. Li, Q., Chang, E.: Security of public watermarking schemes for binary sequences. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 119–128. Springer, Heidelberg (2003)
12. Merhav, N., Sabbag, E.: Optimal watermark embedding and detection strategies under limited detection resources. In: IEEE Transactions on Information Theory (submitted) (December 2005), `http://www.ee.technion.ac.il/people/merhav/`
13. Moulin, P., Koetter, R.: Data-hiding codes (tutorial paper). Proceedings of the IEEE 93(12), 2083–2127 (2005)
14. Moulin, P., O'Sullivan, J.A.: Information-theoretic analysis of information hiding. IEEE Transactions on Information Theory 49(3), 563–593 (2003)
15. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1993)
16. Papadimitriou, C.H., Yannakakis, M.: A note on succinct representations of graphs. Information and Control 71, 181–185 (1986)
17. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: IEEE Symposium on Foundations of Computer Science, FOCS 2000, pp. 86–92 (2000)
18. Skyum, S., Valiant, L.G.: A complexity theory based on boolean algebra. Journal of the ACM 32(2), 484–502 (1985)
19. Wegener, I.: The Complexity of Boolean Functions, January 1991. Wiley, Chichester (1991)

A Proof of Theorem 1

A.1 NEXP-Completeness of VC on Succinct Instances

NEXP hardness. To prove NEXP-hardness of VC on succinct instances, we use lemma 1 so we need to construct a projection π from SAT to the ordinary problem DS. This projection mimics the classical reduction [6] from 3SAT to VC in the NP-completeness proof of VC. We recall the reduction.

We consider a 3SAT instance C with m variables and n clauses. We construct an instance (G, k) of VC using the rules below, illustrated on Figure 1.

– For each variable x_i, two nodes q_{x_i} and $q_{\bar{x}_i}$ are created and an edge between q_{x_i} and $q_{\bar{x}_i}$ is added.

- For each clause C_j, three nodes q_j^1, q_j^2 and q_j^3 are created and an edge added from q_j^k to q_{x_k} if x_k is the kth variable of C_j or $q_{\bar{x}_k}$ if $\bar{x}_k$ is the kth variable of C_j. The nodes q_j^1, q_j^2 and q_j^3 are linked alltogether to form a triangle. If a clause is made of only one or two variables, the missing ones are created by duplicating the last one.
- $k = n + 2m$.

Then G has $2.m + 3n$ vertices and admits a vertex cover of size at most k if and only if C is satisfiable [6] . We define $\pi(C)$ as the binary string encoding the adjacency matrix of G. To define s, we order the nodes $(q_{x_1}, q_{\bar{x}_1}, \dots, q_{x_n}, q_{\bar{x}_n}, q_1^1, q_1^2, q_1^3, \dots, q_m^1, q_m^2, q_m^3)$ and choose the i-th value of s according to the edge relationship between vertices $i_1 = i \mod |G|$ and $i_2 = (i - i_1)/|G|$ of$G$. The total length of $\pi(C)$ is $O((2m + 3n)^2)$.

$$(x \vee y) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee z)$$

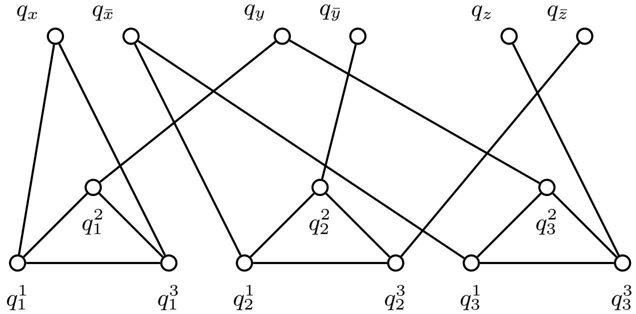


Fig. 1. Reduction from 3SAT to VC

We consider a position i within $\pi(C)$. Then the value of the ith bit of $\pi(C)$ can be obtained either by a direct computation or through a single look-up in (the binary presentation of) C as follows. First we compute $i_1' = i \mod (2m + 3n)$ and $i_2 = i/(2m + 3n)$.

- If $i_1 \leq 2.n$ and $i_2 \leq 2.n$ output 0 unless $|i_1 - i_2| = 1$. item If $i_1 \geq 2.n$ and $i_2 > 2.n$, compute the index j of the clause and the position k within the clause i_2 refers to as well as the variable x and the parity i_1 refers to. Output 1 if x is the kth variable of the jth clause with this parity, 0 unless. This requires one lookup in the original string x.
- If $i_2 \geq 2.n$ and $i_1 > 2.n$, we proceed exactly as in the previous case.
- If $i_1 \geq 2.n$ and $i_1 > 2.n$, we output 1 if $i_1 \mod 3 = i_2 \mod 3$ which is the case only when i_1 and i_2 are two nodes associated with the same clause.

Hence, we build an algorithm which outputs the value of the ith bit of $\pi(x)$ or provides a position within x that gives its value, in polynomial time. So π is a projection from SAT to VC and by [16], VC on succinct instances is NEXP-hard.

VC on succinct instances is in NEXP. We have to show that VC on succinct instances is in NEXP. Suppose that one guesses a subset of nodes V' with size smaller than k. To verify that V' hits all the edges of G, we need to enumerate all combinations of nodes that is $O((2^n)^2)$ steps and for each step, verify that at least one of its end nodes is in V'. Then VC is NEXP and is therefore NEXP-complete.

A.2 DS on Succinct Instances Is NEXP-Complete

We show that DS on succinct instances is NEXP-complete by reduction from VC. Let (k, G) be a VC problem on succinct instances that is there is a polynomial boolean circuit C_G that computes the adjacency matrix of G. We assume, without loss of generality, that G has 2^n nodes (when the number of nodes is not a power of two, we can still add extra isolated nodes which does not change the complexity of the problem). The circuit C_G has $2n$ inputs and, given the (binary representations of) integers i and j outputs 1 if and only if there is an edge between i and j in G. To achieve the reduction, we need to construct in polynomial time, and instance (k', G') of dominating set on succinct instances such that G has a vertex cover of size k if and only if G' has a dominating set of size k'. In particular, there must be a polynomial size boolean circuit $C_{G'}$ which computes the adjacency matrix of G'. The idea of the reduction is to construct a boolean circuit from the classical reduction from Vertex Cover to Dominating Set [6]. This is achieved by adding extra vertices in G' of the form vw for each couple of nodes v, w linked by an edge in G and edges from v and w to vw. This reduction is illustrated on Figure 2. Doing this, G has a vertex cover of size k if and only if G' has a dominating set of size k. If v and w are nodes of G having binary representations V and W, we code the corresponding nodes in G' as follows: v is coded as $0Vx \dots x$, w as $0Wx \dots x$ and vw as $1VW$. Then, we construct a boolean circuit $C_{G'}$ with $2.(2n + 1) = 4n + 2$ inputs. Let $pP_1P_2qQ_1Q_2$ a binary input of $C_{G'}$ that must compute $C_G(P_1, Q_1)$ when both p and q are true, $C_G(P_1, Q_1)$ (resp. $C_G(P_2, Q_1)$) when p is false and $P_1 = Q_1$ (resp. $P_1 = Q_2$), $C_G(Q_1, P_1)$ (resp. $C_G(Q_2, P_1)$) when p is true and P_1 and Q_1 (resp. $Q_1 = P_2$), 0 when p and q are true. More formally, $C_{G'}$ must compute the boolean formula:

$$\begin{aligned} &((\neg p \wedge \neg q) \wedge 0) \\ \vee\, &((p \wedge q) \wedge C_G(P_1, P_2)) \\ \vee\, &((\neg p \wedge q) \wedge A(P_1, P_2, Q_1)) \\ \vee\, &((p \wedge \neg q) \wedge A(Q_1, Q_2, P_1)) \end{aligned}$$

in which $A(P, Q, R)$ is a boolean circuit with $3.n$ inputs which outputs 1 if and only if $P = R$ or $Q = R$ and $C_G(P, Q)$. It remains to be shown that the size of the circuit computing this formula is polynomial in the size of its inputs.

It is clear that the first two terms are polynomial (the first one can even be removed). We show now that A is polynomial. We use the $Eq(.,.)$ circuit,

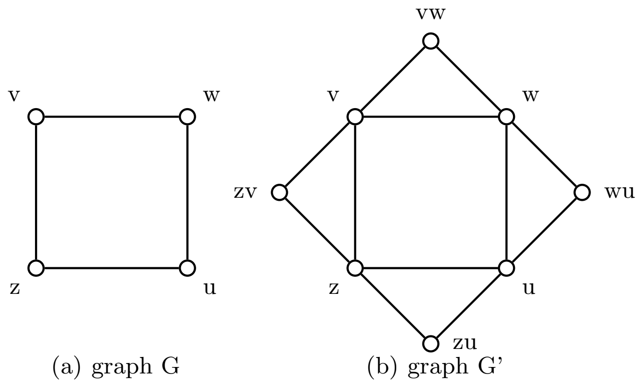


Fig. 2. Reduction [6] from an instance (G,k) of VC to an instance (G',k) of DS

a boolean circuit with $2.n$ inputs which outputs 1 if and only if the integers represented by its arguments are equal. Using only binary AND gates, the size of Eq is $O(log_2(n))$. Then, A can be implemented as:

$$(Eq(P, R) \vee Eq(Q, R)) \wedge C_G(P, Q),$$

i.e. with a polynomial size boolean circuit since C_G has polynomial size. This concludes the proof.

A Theoretical Framework for Watermarking of Compressed Signals

Séverine Baudry and Philippe Nguyen

Thomson STS, Cesson Sévigné, France
{severine.baudry,philippe.nguyen}@thomson.net

Abstract. Nowadays, digital video, audio and multimedia signals are mostly transported, stored, exchanged, broadcast in compressed form. Thus, most of the multimedia processing operations, amongst them watermarking, shall be done in the compressed domain. When the bitrate is small, the quantization steps are coarse, and the inserted watermark may be heavily distorted or visible; moreover, watermark insertion may increase the stream rate, which is most of the time not acceptable. We propose to model the watermark insertion as an optimization problem on a discrete set, with visibility and rate constraints. We give an iterative method to get the optimal watermarked signal. We show by experimental results that this approach increases the robustness of the watermark.

Keywords: Stream watermarking, compression, quantization, optimization, bit-rate model.

1 Introduction

Nowadays, digital video, audio and multimedia signals are mostly transported, stored, exchanged, broadcast in compressed form. Huge improvements in compression efficiency and complexity, as well as low cost of encoding / decoding device, make it easy to save bandwidth and storage by compressing content. Uncompressed signals are found only at the very end of the chain, just before the rendering or displaying on the user device, or at the very beginning (most of capturing devices, like camcorders or recorders, enable embedded compression just after acquisition). Even in the professional world, where the switch to compressed (and even digital) domain is more recent due to the very high quality requirements and the cost of appropriate equipments, most of the content is now stored, processed, transmitted in high bit rate compressed formats. To allow maintain of content quality and seamless insertion in the framework, watermarking has thus to be applied in the compressed domain, since decompressing and recompressing without taking into account the initial structure of the coded stream leads, most of the time, to very poor resulting quality. Moreover, it is most of the time mandatory to keep the bit-rate strictly constant after watermark insertion, to avoid re-multiplexing and to fit inside fixed capacity channels.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 410–422, 2008.

Several methods have been proposed to watermark compressed video streams; most of them exploit the specificities of the compression scheme (transform space, prediction) to design a suitable algorithm. Early methods directly transpose spread spectrum base band watermarking in the compressed domain, potentially resulting in visible artifacts or robustness decrease [3]. Some schemes are especially designed to work into the compressed domain, either for JPEG, MPEG 2 or H264 [6], but still allow to retrieve the watermark on the base-band signal. Other methods modify motion vectors of a MPEG 2 stream, making it uneasy to detect the watermark on the decompressed stream [7].

We propose here a generic theoretical framework for watermarking of compressed signals, by controlling jointly the two main constraints: resulting quality (invisibility or inaudibility of the watermark) and robustness. A third constraint may be added to keep the bitrate of the stream constant. For this purpose, we consider this as a general optimization problem: we want to maximize the robustness of the watermark under an "energy" constraint (invisibility / inaudibility), and we allow the signal to take only a subset of all possible values (quantized values). This is equivalent to maximizing a function of a set of variables (quantized values to modify), bounded by a set of constraints. Note that this method can be applied to non-compressed signals, since digital signals are essentially quantized signals; however, we will see in Section 5 that the proposed method is really interesting when the quantization steps are coarse. When it is not the case, the continuous approximation is sufficient.

The paper is organized as follows: Section 2 presents a generic model of watermarking in the compressed domain. Section 3 sets the generic equation of the optimization problem to solve to optimally embed the watermark, and describes an optimal resolution method. Section 4 is a more generic description of the problem, including the bit-rate constraints. Section 5 gives some results and shows that the proposed method is interesting in the difficult cases (coarse quantization, low visibility threshold).

2 A Model for Compressed Signals

The quantized signal can be seen as a lattice in a multidimensional space, the whole space being the set of all possible continuous signals. The "shape" and volume of a mesh of the lattice depend on the type of quantization and on the quantization steps of the coefficients (for vectorial quantization, the mesh's shape is not restrained to a hyper parallelogram). Like the original signal, the watermarked signal can only belong to the lattice.

Let $\mathbf{X} = \{X_i\}_{i=1\ldots n}$ be the original (compressed) signal (coordinates may be given in the transform domain), $\mathbf{Y} = \{Y_i\}_{i=1\ldots n}$ be the watermarked signal, both signals belonging to the quantization lattice. Let n be the number of (quantized) signal samples. In most of the compression schemes used, the coefficients obtained after prediction and transformation are scalar quantized, with quantization steps Q_i (quantization step may vary with i, i.e. with the frequency) ;

therefore, we have

$$X_i = x_i Q_i \tag{1}$$

with $x_i \in \mathbb{Z}$. In the same way, $Y_i = y_i Q_i$ with $y_i \in \mathbb{Z}$.

Note, however, that this lattice is not universal, even for a given compression scheme. It does not reflect the whole information available on the stream, since it does not take into account higher level information which is difficult to translate in pure signal processing words (e.g. in H264: macroblock size, spatial prediction modes, motion vectors ...). This means that two different streams will have different values for these high level structures, and the resulting lattices will not be comparable. In the following, we assume that the watermarking process will not change high level data, but only the residual noise represented by the lattice subset.

Let $f(\mathbf{Y})$ be a function measuring the robustness of the watermarked signal, and $g(\mathbf{X}, \mathbf{Y})$ be a function measuring the watermark perceptibility. We want to maximize $f(\mathbf{Y})$ under the constraint $g(\mathbf{X}, \mathbf{Y}) \leq 0$. The maximization takes place only on the set of authorized quantized values.

In the following, we assume that the watermarking scheme belongs to the class of spread spectrum methods, and that the watermark demodulation is performed by correlation; thus, we use the robustness function given by

$$f(\mathbf{Y}) = \sum_{i=1}^{n} Y_i \omega_i, \tag{2}$$

$\{\omega_i\}_{i=1 \ldots n}$ being the watermark signal to embed.This is not always true, because the channel may impair selectively the coefficients; in this case, the coefficients may be weighted according to the susceptibility to channel noise. Normalized correlation may also be used to render the watermark robust to valumetric changes. In this case, correlation will be weighted by the norm of the signal $\mathbf{Y}$ in Equation 2. If we suppose that $\|\mathbf{Y}\| \simeq \|\mathbf{X}\|$, i.e. that the watermark energy is negligible compared with the signal energy, using normalized correlation is equivalent to dividing the correlation by a constant $\|\mathbf{X}\|$; this constant will not impact further calculations. For other embedding schemes (for instance for quantization or side information methods), other functions will be used to reflect the robustness, like for instance the distance between the watermarked signal and the elements of the watermarking lattice.

Equation 2 is equivalent to

$$f(\mathbf{Y}) = \sum_{i=1}^{n} y_i w_i, \tag{3}$$

with y_i integer and $w_i = \omega_i Q_i$.

We define the perceptibility function as

$$g(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n} \gamma_i^2 (Y_i - X_i)^2 - T^2. \tag{4}$$

T^2 is a perceptibility threshold and γ_i the psycho-visual weight of coefficient i; γ_i may, for instance, be inversely proportional to the quantization step Q_i. Note that $g(\mathbf{X}, \mathbf{Y}) \leq 0$ for solutions with acceptable imperceptibility. Here again, more complex and accurate visual models may be used, integrating for instance contrast sensitivity.

Equation 4 is equivalent to

$$g(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n} b_i^2 (y_i - x_i)^2 - T^2, \tag{5}$$

where x_i and y_i are integers and $b_i = \gamma_i Q_i$.

In the following we will use $f(\mathbf{y})$ instead of $f(\mathbf{Y})$, and $g(\mathbf{x}, \mathbf{y})$ instead of $g(\mathbf{X}, \mathbf{Y})$.

3 Watermarking as an Optimization Problem

3.1 Definition of the Optimization Problem

We introduce the Lagrange multiplier λ (with $\lambda \geq 0$) and define the following cost function

$$R(\mathbf{y}, \lambda) = -f(\mathbf{y}) + \lambda g(\mathbf{x}, \mathbf{y}). \tag{6}$$

We also define

$$L(\lambda) = \min_{\mathbf{y}} R(\mathbf{y}, \lambda). \tag{7}$$

To find the watermarked signal which maximizes robustness with bounded visility, we have to minimize $R(\mathbf{y}, \lambda)$. Optimization theory (see for instance [2][5]) shows that maximizing $f(\mathbf{y})$ with bounded $g(\mathbf{x}, \mathbf{y})$ is equivalent to the dual problem of finding

$$\max_{\lambda} L(\lambda). \tag{8}$$

3.2 Resolution in the Continuous Domain

In this section we solve the problem in the continuous domain (i.e. we do not restrict to quantized y_i and x_i). We will thus have the expression of the optimum watermarked signal, since any quantized signal will only approach this optimum. This will be useful to evaluate the distortion at embedding due to quantization constraints, as will be shown in Section 5. We will also compare our method to the simplest solution consisting in quantizing the continuous solution.

Equation 6 is equivalent to

$$R(\mathbf{y}, \lambda) = -\sum_{i=1}^{n} w_i y_i + \lambda \left(\sum_{i=1}^{n} b_i^2 (y_i - x_i)^2 - T^2 \right). \tag{9}$$

$R(\mathbf{y}, \lambda)$ is convex in each y_i since λ is non negative; furthermore, notice that $R(\mathbf{y}, \lambda)$ is a sum of n terms $R_i = -w_i y_i + \lambda \left((y_i - x_i)^2 b_i^2 - T^2/n \right)$, each R_i

depending only on y_i: thus, the minimization may be performed independently on each y_i. As R_i is convex, it is minimal when its derivative is zero:

$$\frac{\partial R_i}{\partial y_i} = 2\lambda b_i^2 (y_i - x_i) - w_i = 0; \tag{10}$$

thus,

$$y_i = \frac{w_i}{2\lambda b_i^2} + x_i. \tag{11}$$

Hence $L(\lambda)$ is given by

$$L(\lambda) = -\sum_i \left(\frac{w_i}{2b_i}\right)^2 \frac{1}{\lambda} - \lambda T^2 - \sum_i x_i w_i. \tag{12}$$

Now, we search for the maximum of $L(\lambda)$, function of the form $-\alpha\lambda - \beta/\lambda - \gamma$ with $\alpha \geq 0$ and $\beta \geq 0$. It is concave for $\lambda > 0$, therefore it is maximal when its derivative is zero. We have

$$\frac{\partial L}{\partial \lambda} = -\alpha + \frac{\beta}{\lambda^2} = 0, \tag{13}$$

and then

$$\lambda = \sqrt{\frac{\beta}{\alpha}} \tag{14}$$

$$= \frac{\sqrt{\sum_i \left(\frac{w_i}{2b_i}\right)^2}}{T}. \tag{15}$$

Hence, the optimal solution y_i^C in the continuous case is:

$$y_i^C = x_i + \frac{w_i T}{b_i^2 \sqrt{\sum_j \left(\frac{w_j}{b_j}\right)^2}}. \tag{16}$$

3.3 Resolution in the Discrete Domain

The simplest way to obtain a quantized signal $\mathbf{y}$ is to quantize the value of each coefficient given by Equation 16. As x_i is already quantized, this is equivalent to quantizing the "watermark" signal $\frac{w_i T}{b_i^2 \sqrt{\sum_j \left(\frac{w_j}{b_j}\right)^2}}$. However, this may not be optimal. For instance, if the quantization step is high compared with the psycho-visual threshold T, this method will lead to round every watermark value to 0; whereas, by concentrating the available energy on some few coefficients, one could get some distorted but non zero watermark signal. The energy "saved" on unmodified coefficients can be reallocated to other ones. The purpose of the following equations is to mathematically define which coefficients will be modified. Two antagonist reasons will trigger the selection of a coefficient: the ability to fulfill the energy condition, and the ability to increase the watermark robustness.

Properties

Concavity. Let $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$; for all $0 \leq \zeta \leq 1$ let's define $\lambda = \zeta\lambda_1 + (1 - \zeta)\lambda_2$. From the definition of $L(\lambda)$, there exists one vector $\mathbf{y}^0$ such that :

$$L(\lambda) = f(\mathbf{y}^0) + \lambda g(\mathbf{y}^0) \tag{17}$$

and, by the definition of $L(\lambda_1)$ and $L(\lambda_2)$, we have

$$\begin{array}{l} L(\lambda_1) \leq f(\mathbf{y}^0) + \lambda_1 g(\mathbf{y}^0) \\ L(\lambda_2) \leq f(\mathbf{y}^0) + \lambda_2 g(\mathbf{y}^0). \end{array} \tag{18}$$

Multiplying the first equation by $\zeta \geq 0$ and the second one by $(1 - \zeta) \geq 0$, and summing them up,we get:

$$\zeta L(\lambda_1) + (1 - \zeta)L(\lambda_2) \leq f(\mathbf{y}^0) + \lambda g(\mathbf{y}^0) = L(\lambda). \tag{19}$$

Thus, $L(\lambda)$ is concave; note that this property is general and does not depend of the values of $g()$ and $f()$.

Piecewise linearity. As the set of acceptable solutions $\{\mathbf{y}^j\}$ is discrete, $L(\lambda)$ is made of pieces of lines of equations :

$$\begin{array}{l} z = f(\mathbf{y}^{i1}) + \lambda g(\mathbf{y}^{i1}) \\ z = f(\mathbf{y}^{i2}) + \lambda g(\mathbf{y}^{i2}) \\ \dots \end{array} \tag{20}$$

with $\mathbf{y}^{i1}$ solution of $L(\lambda)$ for λ in a range $[\lambda^1, \lambda^2]$, $\mathbf{y}^{i2}$ solution in the range $[\lambda^2, \lambda^3]$, and so on.

Solving the Dual Problem with Subgradients. We know that $L(\lambda)$ is concave; thus, any local optimum is a global one. However, since it is not differentiable everywhere (due to the discrete constraints), the classical gradient descent techniques used in the continuous domain must be adapted.

We propose to use the concavity and piecewise linearity of $L(\lambda)$ to find its maximum value (see Figure 1). As the function is piecewise linear, we have:

$$\frac{\partial L}{\partial \lambda} = \frac{\partial \left(g(\mathbf{x}, \mathbf{y})\lambda - f(\mathbf{y}) \right)}{\partial \lambda} = g(\mathbf{y}), \tag{21}$$

with $\mathbf{y}$ being the solution of $L(\lambda)$. Thus, its sign gives us information about the direction of the maximum, which may be approached by dichotomy.

The algorithm is the following:

1. Initialize two values of λ, λ^0 and λ^1, such that $\lambda^0 \leq \lambda^{opt} \leq \lambda^1$. Compute the average $\lambda' = \frac{\lambda^0 + \lambda^1}{2}$ (see Figure 1 for a visual illustration of λ^0, λ^1, λ').

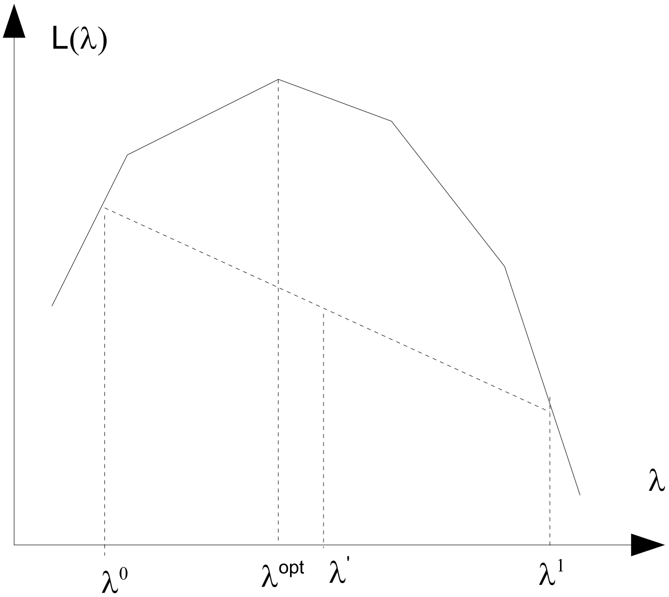


Fig. 1. $L(\lambda)$; notice that $L(\lambda)$ is piecewise linear and concave. λ_{opt} maximizes $L(\lambda)$; λ_0 and λ_1 are the initialization parameters.

2. Find the solution $\mathbf{y}'$ which minimizes $R(\mathbf{y}, \lambda')$. As in the continuous domain (see section 3.2), minimization may be performed independently on each y_i. Recall that the solution y_i^C in the continuous case is

$$y_i^C = x_i + \frac{w_i}{2\lambda' b_i^2}. \tag{22}$$

Thus the solution for the quantized domain is just the allowed value (i.e. belonging to the quantized set) closest to the continuous solution :

$$y_i' = x_i + Round\left(\frac{w_i}{2\lambda' b_i^2}\right). \tag{23}$$

3. Compute $g(\mathbf{y}')$, which is the value of the derivative of $L()$ in λ'. The sign of $g(\mathbf{y}')$ indicates whether λ^{opt} is smaller or greater than λ'.
4. Repeat the process from Step 1 to Step 3, with

$$\begin{cases} \lambda^0 = \lambda' \text{ if } g(\mathbf{y}') > 0 \\ \lambda^1 = \lambda' \text{ if } g(\mathbf{y}') < 0. \end{cases}. \tag{24}$$

When close to the solution, the algorithm oscillates between two values $\mathbf{y}^{opt0}$ and $\mathbf{y}^{opt1}$ (these values correspond to the two segments intersecting in λ^{opt}). We have

$$g\left(\mathbf{y}^{opt0}\right) \geq 0 \tag{25}$$

and

$$g\left(\mathbf{y}^{opt1}\right) \leq 0; \tag{26}$$

thus, $\mathbf{y}^{opt1}$ is a solution of the optimization problem. Note that $\mathbf{y}^{opt0}$ is a solution only if there is equality in Equation 26 (else $\mathbf{y}^{opt0}$ does not satisfy the invisibility constraints).

Note that a more efficient way to compute λ' is to take

$$\lambda' = \frac{f\left(\mathbf{y}^0\right) - f\left(\mathbf{y}^1\right)}{g\left(\mathbf{y}^0\right) - g\left(\mathbf{y}^1\right)} \tag{27}$$

This is the abscissa of the intersection of the two lines prolonging the segments bearing λ^0 and λ^1. If the solution of $L(\lambda')$ is λ^0 or λ^1, we have found the maximum of $L(\lambda)$.

A good initializing value λ^{init} for λ is the optimum in the continuous domain given by Equation 14, since the optimal quantized solution will not be too far away from the optimal continuous one. Depending on the sign of $g(\mathbf{y}^{init})$, with $\mathbf{y}^{init}$ solution of $g(\lambda^{init})$, we assign λ^{init} to either λ^0 or λ^1.

4 Joint Optimization with Bit-Rate Constraint

4.1 Bit-Rate Modeling

We want now to take into account the bit-rate constraint: this constraint specifies that watermarking should not change the size of the bitstream, or at least not increase it. To be able to integrate this constraint into the mathematical framework defined above, we thus have to define the function $h(\mathbf{y})$, which gives the bit-rate, or size, of the signal $\mathbf{y}$ after entropic encoding. Such a function is quite difficult to model, although it can be computed for any $\mathbf{y}$, by simply entropy coding $\mathbf{y}$. One of the simplest models for MPEG-2 encoding is to use ρ, the fraction of zeros amongst quantized DCT coefficients. In [4], authors suggest that the rate $r(\mathbf{y})$ may be closely approached by

$$r(\mathbf{y}) = \theta\left(1 - \rho(\mathbf{y})\right), \tag{28}$$

θ being a constant. Hereafter, we treat the problem for the MPEG-2 case; therefore, $\mathbf{y}$ is a set of quantized DCT coefficients.

4.2 Integrating Bit-Rate Constraints into the Optimization Problem

Let's define the bit-rate constraint function $h(\mathbf{y})$ as

$$h(\mathbf{y}) = \theta\left(1 - \rho(\mathbf{y})\right) - S, \tag{29}$$

S being the bit-rate of the original stream; a solution is acceptable only if $h(\mathbf{y}) \leq 0$ (i.e. watermarking decreases or keeps constant the bit-rate). As $\rho(\mathbf{y})$ is the percentage of null y_i amongst all the coefficients, we have:

$$\rho(\mathbf{y}) = \sum_{i=1}^{n} \frac{\delta(y_i)}{n}, \tag{30}$$

where

$$\delta(y_i) = \begin{cases} 1 \text{ if } y_i = 0 \\ 0 \text{ if } y_i \neq 0 \ . \end{cases} \tag{31}$$

To simplify the notation, we define $S' = \theta - S$; thus, $h(\mathbf{y}) = S' - \theta\rho(\mathbf{y})$. Now define the cost function integrating the bit-rate constraint:

$$R(\mathbf{y}, \lambda, \mu) = -f(\mathbf{y}) + \lambda g(\mathbf{x}, \mathbf{y}) + \mu h(\mathbf{y}), \tag{32}$$

with $\lambda \geq 0$ and $\mu \geq 0$. Similarly, let's define

$$L(\lambda, \mu) = \min_{\mathbf{y}} R(\mathbf{y}, \lambda, \mu). \tag{33}$$

To apply a "gradient descent like" technique as presented in Section 3.3, we need to be able to compute $L(\lambda, \mu)$ for all λ and μ. We have:

$$R(\mathbf{y}, \lambda, \mu) = -\sum_{i=1}^{n} w_i y_i + \lambda \left(\sum_{i=1}^{n} (y_i - x_i)^2 b_i^2 - T^2 \right) + \mu \left(S' - \theta \sum_{i=1}^{n} \frac{\delta(y_i)}{n} \right). \tag{34}$$

As in the previous case, $R(\mathbf{y}, \lambda, \mu)$ can be defined as a sum of n terms $R_i(y_i, \lambda, \mu)$, each term depending only on y_i:

$$R_i(y_i, \lambda, \mu) = -w_i y_i + \lambda \left((y_i - x_i)^2 b_i^2 - \frac{T^2}{n} \right) + \mu \left(\frac{S'}{n} - \theta \frac{\delta(y_i)}{n} \right). \tag{35}$$

Hence, for given λ and μ, we may minimize R on each y_i independently. The shape of R_i is similar to the previous case (up to a constant), except in $y_i = 0$, where we have

$$R_i(0, \lambda, \mu) = -\lambda \left(x_i^2 b_i^2 - T^2/n \right) + \mu \left(S' - \theta/n \right). \tag{36}$$

Let's define $\hat{y}_i = x_i + Round\left(\frac{w_i}{2\lambda b_i^2}\right)$, the solution of $L(\lambda)$ found in the previous case (equation 23). Here we have:

$$y_i = \begin{cases} \hat{y}_i \text{ if } R_i(\hat{y}_i, \lambda, \mu) < R_i(0, \lambda, \mu) \\ 0 \ \text{ else} \end{cases}. \tag{37}$$

The gradient of R will be given by

$$\nabla R = \left(\frac{\partial R}{\partial \lambda}, \frac{\partial R}{\partial \mu} \right) = \left(g(\mathbf{y}), h(\mathbf{y}) \right). \tag{38}$$

Again, this gradient is not defined everywhere since R is piecewise linear.

Algorithm. We apply an iterative process to find the optimal λ and thus the optimal $\mathbf{y}$. At each step i we have parameters λ^i and μ^i, and the corresponding solution $\mathbf{y}$.

1. Initialization: Initialize λ and μ with any value λ^0 and μ^0.
2. Step i : find $\mathbf{y}^i$ which minimizes $R(\mathbf{y}, \lambda^i, \mu^i)$ by Equation 37. Then define λ^{i+1} and μ^{i+1} as:
$$\begin{cases} \lambda^{i+1} = \lambda^i + \epsilon^i g(\mathbf{y}^i) \\ \mu^{i+1} = \mu^i + \epsilon^i h(\mathbf{y}^i), \end{cases} \tag{39}$$
with $\{\epsilon^i\}_{0 \leq i \leq \infty}$ a sequence such that $\epsilon^i \to 0$ when $i \to \infty$. It can be shown that $L(\lambda^i, \mu^i)$ converges to the optimum $(\lambda^{opt}, \mu^{opt})$ as soon as $\sum_{i=0}^{\infty} \epsilon^i = +\infty$.

5 Results

We compare here the performances obtained by the optimal solution to the simple solution, which consists in quantizing the continuous optimum:

$$y_i^{CQ} = x_i + \left\lfloor \frac{w_i T}{b_i^2 \sqrt{\sum_j \left(\frac{w_j}{b_j}\right)^2}} \right\rfloor , \tag{40}$$

with $\lfloor . \rfloor$ being the integer part. As a reference, we consider $\langle \mathbf{w}, \mathbf{y}^C \rangle$ the correlation between the watermark vector and the optimum continuous solution. Therefore, we use the correlation ratio

$$U(\mathbf{y}) = \frac{\langle \mathbf{w}, \mathbf{y} \rangle}{\langle \mathbf{w}, \mathbf{y}^C \rangle} \tag{41}$$

to measure the loss in watermark strength caused by using a quantized vector $\mathbf{y}$.

For the following tests we choose for the watermark vector $\mathbf{w}$ a random white Gaussian signal. To focus on noise due to quantization, we choose an original signal such that $\langle \mathbf{x}, \mathbf{w} \rangle = 0$, i.e. there is no interference between the watermark and the original work.

In Figure 2 we have plotted the variation of the correlation ratio with the visibility threshold T. We note that the optimal solution outperforms the quantization of the continuous solution, in particular when the visibility threshold T is low compared to the quantization steps. This is due to the fact that with the quantization of the continuous solution, each vector component is treated independently from the others, therefore the energy allocated to each component is not sufficient to pass the quantization threshold. On the contrary, with the optimal solution, the available energy is concentrated on a few coefficients. This leads to a very distorted, but still non zero, watermark, as we can see from the correlation ratio.

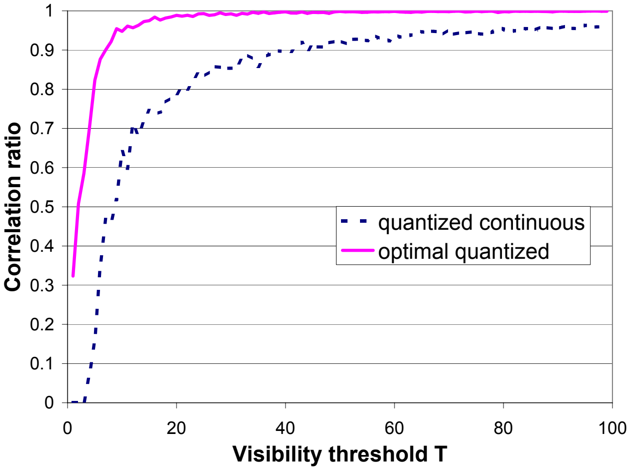


Fig. 2. Correlation ratio for the optimal solution, and for the quantization of the continuous solution, variation with the visibility threshold, for a watermark length of 100

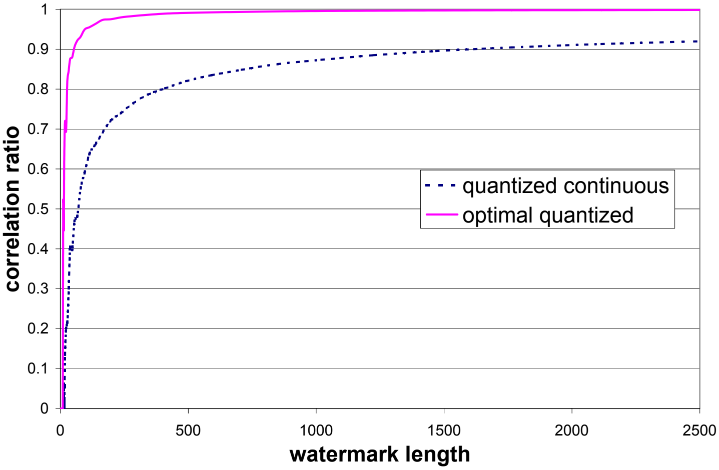


Fig. 3. Correlation ratio depending on the watermark length

Note that when the visibility threshold is low, both solutions lead to poor performances compared to the continuous case. On the contrary, the optimal solution closely approaches the performances for the continuous case for large visibility thresholds : quantization noise is then negligible.

On figure 3 we have drawn the variation of the correlation ratio with the length n of the watermark vector.We have set the visibility threshold to $T = n/10$, so that the average energy by watermark coefficient remains constant (around 1 coefficient amongst 10 can be modified). Note that the quantized schemes

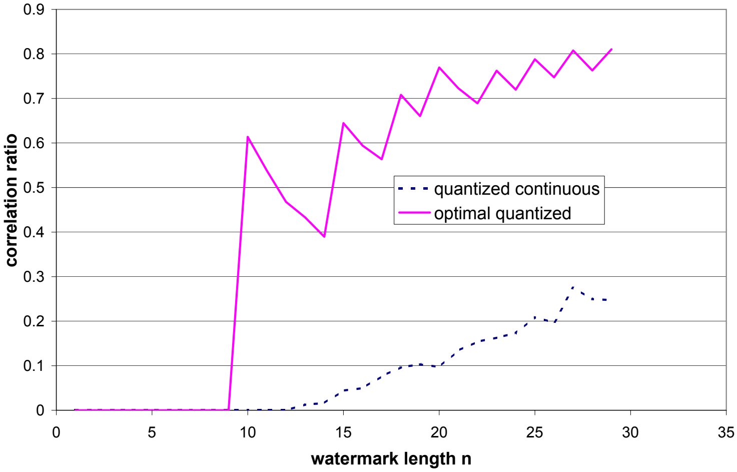


Fig. 4. Correlation ratio when the watermark length is small

perform much worse when the watermark length is small: in that case, there are not enough degrees of freedom to allocate the watermark energy on the coefficients efficiently. The gain provided by the optimal solution is again much higher in the difficult cases, i.e. when the watermark length is short. This is shown in figure 4, where we have focused on small lengths.

6 Conclusion

In this paper we proposed a generic framework for optimizing the robustness when working with compressed signals, subjected to constraints of invisibility and non-increasing bit-rate. We gave a method to find the optimum when the robustness is measured by a correlation score, the visibility by a weighted Euclidean distance, and the bit-rate by the rate of non-zero coefficients. We have shown that this method provides significant improvement compared with simple quantization of the optimal continuous solution, especially when the quantization is coarse or the watermark length is small. We also suggested that using long watermark signals (i.e. high dimensionality) increases the watermark robustness, especially when the visibility threshold is low compared to the quantization steps.

Further work will be to apply this framework on a real watermarking scheme, for instance for MPEG-2 or H264. A first step will be to check the accuracy of the rate modeling proposed. The same method could be apply to other watermarking schemes like QIM [1] or side-informed techniques. This will change the robustness function: $f(\mathbf{x}, \mathbf{y})$ will there be the distance to the watermark subset. Similarly, more efficient visual models and more accurate bit-rate model may be used.

Acknowledgments. This Work is supported by the ANR project Estivale. Authors would also like to thank Caroline Fontaine for valuable comments and suggestions.

References

1. Chen, B.: Design and analysis of digital watermarking, information embedding and data hiding systems. PhD thesis (June 2000)
2. Gondran, M., Minoux, M.: Graphes et Algorithmes. Eyrolles (1995)
3. Hartung, F., Girod, B.: Watermarking of uncompressed and compressed video. Signal Processing 66(3), 283–301 (1998)
4. He, Z., Mitra, S.: A linear source model and a unified rate control algorithm for dct video coding 12(11) (November 2002)
5. Kreher, D., Kocay, W.: Graphs, algorithms and optimization (2004)
6. Noorkami, M., Mersereau, R.: Compressed domain video watermarking for h264. In: Proc. of ICIP 2005, Genova (September 2005)
7. Vynne, T., Jordan, F.: Embedding a digital signature in a video sequence using motion vectors. In: CIP 1996 (submitted, 1996)

Least Distortion Halftone Image Data Hiding Watermarking by Optimizing an Iterative Linear Gain Control Model

Weina Jiang, A.T.S. Ho, and H. Treharne

The Department of Computing
University of Surrey
Guildford, GU2 7XH, UK
W.Jiang@Surrey.ac.uk

Abstract. In this paper, a least distortion data hiding approach is proposed for halftone image watermarking. The impacts of distortion and tonality problems in halftoning are analyzed. An iterative linear gain model is developed to optimize perceptual quality of watermarking halftone images. An optimum linear gain for data hiding error diffusion is derived and mapped into a standard linear gain model, with the tonality evaluated using the average power spectral density. Our experiments show that the proposed linear gain model can achieve an improvement of between 6.5% to 12% as compared to Fu and Au's data hiding error diffusion method using the weighted signal-to-noise ratio(WSNR).

1 Introduction

Halftoning is an important operation that transforms conventional grayscale and color images into bi-level images that are particularly useful for print-and-scan processes [11]. In fact many images and documents displayed in newsprint and facsimile are halftone images. The problems of copyrights protection and unauthorized tampering of digital content, particularly in printed and scanned form, are becoming increasingly widespread that need to be addressed. As such, digital watermarking [10] can be very useful in protecting printable halftone images for business, legal and law enforcement applications. Currently, there are two classes of data hiding methods proposed in the literature where (1) a watermark is embedded into a halftone image during an error diffusion process. In [2], Fu and Au proposed this approach. The main idea is that performing self-toggling in N pseudo random locations, the error of self-toggling plus the error of standard error diffusion halftoning diffuse to neighboring pixels. A private key is required in the verifier side to generate N pseudo random locations to be used for retrieving the watermark; (2) a watermark is embedded into two or more halftoning images, where the retrieval approaches can be overlays of halftone images [15] or using a look up table (LUT) to decode the watermark [12].

Halftone quality and robustness are the two main challenges for data hiding watermarking halftone images. The robustness of watermarking can be enhanced

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 423–439, 2008.

by incorporating error correction coding [4]. However, data hiding error diffusion is not trivial since embedding data into halftone image downgrades the perceptual quality of halftone image. It is difficult to increase robustness without causing a significant amount of perceptual distortion because error correction coding requires high capacity of watermarks to be embedded into halftone image.

In this paper, we analyze the sharpening distortion in data hiding error diffusion and propose a watermarking linear gain model to address how to preserve optimum perceptual quality via minimizing distortion in data hiding error diffusion of halftone images. In error diffusion halftoning, a grayscale image is quantized into one bit pixel via an *error diffusion kernel* [13][6]. As a consequence, it sharpens the image and adds quantization noise resulting in artifacts and idle tones. However, some artifacts and idle tones are incurred even without watermark embedding. The main aim of our proposed method is to preserve the least distortion data hiding error diffusion in halftone images.

Furthermore, we propose the use of average power spectrum $\overline{PSD}$ to measure harmonic distortion of halftone and embedded halftone images, analogous to total harmonic distortion (THD) [3]. Experiments show that the proposed iterative halftone watermarking model not only optimizes the perceptual quality of watermarking halftone images but also reduces tonality, with overall perceptual quality significantly better than Fu and Au's method tested on similar images.

The rest of the paper is organized as follows. In Section 2, related work is reviewed. In Section 3, we describe a watermarking embedding process and how the distortion can be modeled and eliminated via an iterative linear gain model. In Section 4, an iterative linear gain halftoning embedding is designed and implemented. In Section 5, we perform experiments to compare perceptual image quality of our proposed method to that of Fu and Au's approach. also the tonality problem is analyzed. We conclude and discuss future work in Section 6 .

2 Related Work

Halftoning quantizes a grayscale or color image into one bit per pixel. There are mainly three existing methods [13], i.e., error diffusion, dithering and iterative methods (Direct Binary Search). Most error diffusion halftones use *an error diffusion kernel* to minimize local weighted errors introduced by quantization. The error caused by quantizing a pixel into bi-levels is diffused into the next-processing neighbour pixels, according to the weights of the diffusion kernel. The two popular error diffusion kernels are those of Jarvis [5] and Floyd and Steinberg [1]. Most error filters have coefficients that sum to one, which guarantee that the entire system would be stable.

Most of the embedding methods use standard error diffusion frameworks. Fu and Au's embedding approach [2] divides an image into macro blocks and one bit of watermark is embedded into each block. A halftone pixel is changed to an opposite value 1 → 0 or 0 → 1, if the embedded watermark is **0** or **1** which is opposite to the image value. The watermark can be retrieved simply by extracting embedding locations in the halftone image. This approach is relatively straightforward. However, as the embedding bits increase in each block, the same value pixels may cause cluster, i.e., regionally white pixels(**1**) or black pixels(**0**) together. The cluster downgrades the overall image quality. Pei et al. [12] proposed a least-mean-square data hiding halftoning where the watermark was embedded into two or more halftone images by minimal-error bit searching and a LUT was used to retrieve the watermark. Wu [15] proposed a mathematical framework to optimize watermark embedding to multiple halftone images where the watermark image and host images were regarded as input vectors. The watermark were then extracted by performing binary logical operation (i.e., XOR) to multiple halftone images.

None of the above methods addresses the problem of minimizing the distortion to preserve the perceptual quality of an embedded halftone image. In this paper we propose a method which takes into account the effects of data hiding on the error diffusion halftoning process.

3 Analysis of Halftoning Distortion in Data Hiding Error Diffusion

In this section, we analyze the key effects of sharpening and noise during the watermarking embedding process of halftone images. Idle tones also will be discussed in Section 5.1

3.1 Sharpening Problems in Data Hiding Error Diffusion

Knox [7] analyzed the quantization error in halftoning at each pixel, which is correlated with the input image. He found that the quantization error was the key reason causing annoying artifacts and *worms* in halftoning. Although worms can be reduced, the sharpness of halftone increases as the correlation of the error image with the input image increases. Sharpening distortion affects perceptual quality of the halftone image [6]. On the other hand, toggling halftone pixels in data hiding error diffusion may increase the quantization errors. Thus, the perceptual quality of image cannot be preserved.

To better model a halftone quantizer, Kite et al. [6] introduced a linear gain plus additive noise model, and applied it to error diffusion quantizer. We summarize his model as follows.

Let $x(i,j)$ is the grayscale image input and $e(i,j)$ is the quantization error caused by the quantizer output $Q(*)$ minus the quantizer input $x'(i,j)$. H(z) is the error diffusion kernel which diffuses the quantization error into neighbor pixels. A standard error diffusion can be expressed as

$$e(i,j) = y_0(i,j) - x'(i,j) \tag{1}$$

$$x'(i,j) = x(i,j) - h(i,j) * e(i,j) \tag{2}$$

$$y_0(i,j) = Q(x'(i,j)) \tag{3}$$

A quantizer output $y_0(i,j) = Q(x'(i,j))$ can be modeled as

$$Q(x'(i,j) = K_s x'(i,j) + n(i,j) \tag{4}$$

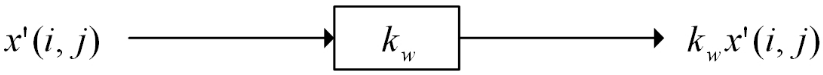


Fig. 1. Data hiding error diffusion linear gain model

where K_s is a linearization constant based on the uncorrelated white noise $n(i,j)$ assumption. The visual quality of halftone image will be preserved if $K_s x'(i,j)$ is approaching halftone output infinitely, i.e., minimizing the squared error between halftone and halftone linear gain model outputs as a criterion:

$$\min_{K_s} \sum_{i,j} (K_s x'(i,j) - y(i,j))^2 \tag{5}$$

Equations (4) and (5) will be true under the circumstance of the uncorrelated white noise assumption of residual image.

In data hiding error diffusion halftoning, we adapt Kite et al. [6] error diffusion process and combine it with data hiding watermarking embedding. We begin by specifying K_w as a linear gain in Figure 1 (K_s represents a linear gain for standard halftone [6]) and proposed a multiplicative parameter L_w compensating input image during data hiding error diffusion process as follows

$$e(i,j) = y(i,j) - x'(i,j) \tag{6}$$

$$x'(i,j) = x(i,j) - h(i,j) * e(i,j) \tag{7}$$

$$x''(i,j) = x'(i,j) + L_w x(i,j) \tag{8}$$

$$y_0(i,j) = Q(x''(i,j)) \tag{9}$$

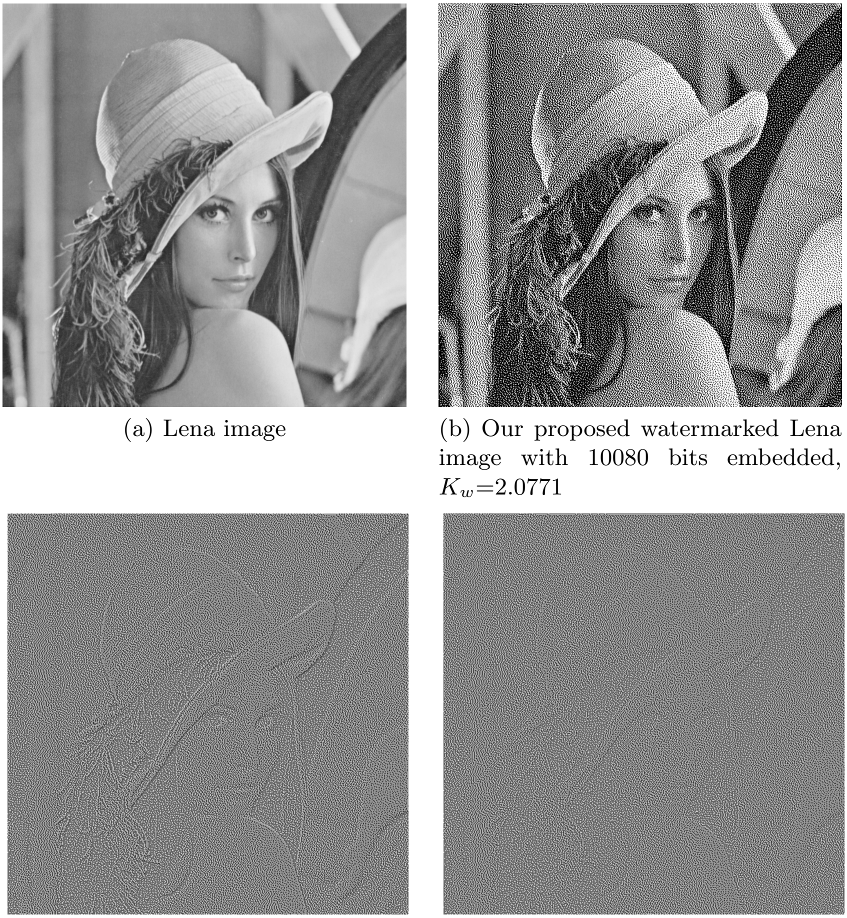
$$y(i,j) = R(y_0(i,j)) \tag{10}$$

where $R(*)$ in the Equation (10) represents the watermarking self-toggle quantizer. We substitute $K_w x'(i,j)$ into quantizer output $y(i,j)$ as a data hiding error diffusion linear gain model. By adjusting a multiplicative parameter L_w, the signal linear gain model output $K_w x'(i,j)$ would approach to watermarked halftone output infinitely, i.e., minimizing the criterion (5) with K_s replaced by K_w. However, K_w can not be estimated by the criterion (5) as long as a watermarked halftone ($y(i,j)$) is obtained. But we can map K_w to standard halftone linear gain K_s in section 3.2.

Now we use Lena image and Jarvis kernel error diffusion as examples to illustrate how L_w could be useful to reduce the correlation between the residual error image and input image(original). The correlation can be quantified as correlation coefficient [14].

$$C_{EI} = \frac{|COV[EI]|}{\sigma_E \sigma_I} \tag{11}$$

Where σ_E and σ_I are the standard deviation of residual image E and input image I, and COV[EI] is the covariance matrix between them. The residual error images (halftone - original) for Fu and Au method and our proposed method are analyzed in Figure 2.



(a) Lena image

(b) Our proposed watermarked Lena image with 10080 bits embedded, K_w=2.0771

(c) Fu and Au's method residual error image,corr=0.0238

(d) Our proposed method's residual error image,corr=0.0110,K_w=2.0771

Fig. 2. Watermarke Halftone and Residual Error Images

Both Fu and Au's method and our proposed method embedded a binary image logo. In our case, the University of Surrey logo (90*112) [9] was used. Figure 2(a) is original Lena image. Figure 2(b) is watermarked halftone lena image based on our proposed method. Figure 2(c) is the residual error image (embedded halftone - original) based on Fu and Au's method. This residual image shows the correlation between the residual image with the input image (corr=0.0238) during data hiding watermarking error diffusion. Figure 2(d) represents our residual image based on our embedding model. From the above figures, we observe in our proposed method the residual image is approximately representing the noise (corr=0.0110). This correlation is significantly reduced as compared to Fu and Au's method. Thus, according to our experiments the sharpness of a watermarked halftone image decreases as the correlation reduces.

3.2 Determine K_w for Data Hiding Error Diffusion

In this section, we derive the mapping from K_w to K_s. Here K_s can be estimated from a standard error diffusion kernel [6]. We consider the embedding of watermark into a block-based halftoning process based on Fu and Au's method. The watermark embedding locations are determined via a Gaussian random variable. As a result, the watermark sequences become white noise embedded into the halftone image. However, due to self-toggling, the watermark bit $w(i,j)$ (0 or 1) can change the pixels in the original halftone image at the selected embedding locations of standard halftone output $y_0(i,j)$(1 or 0) in the host image. We have developed two cases for the embedding procedure.

Case 1: Embedded bit $w(i,j) = y_0(i,j)$. Let a linear gain K_w represents data hiding error diffusion linear gain. The best case scenario is that all watermark bits equal to the standard halftone output $y_0(i,j)$. In this case, none of pixel $y_0(i,j)$ will be toggled. We can simplify by taking $K_w = K_s$ corresponding with $L_w = L_s$ to reduce the sharpening of original halftone.

Case 2: Embedded bit $w(i,j) \neq y_0(i,j)$. The worst case scenario is that all standard halftone outputs $y_0(i,j)$ have to be changed. In this case, the watermarked halftone output $y(i,j)$ becomes $1 - y_0(i,j)$. Our watermarked error diffusion process is described in Equation (6) to Equation (10). We can simply Equation (10) as follow:

$$y(i,j) = 1 - y_0(i,j) \tag{12}$$

By taking z-transformation to Equations (6-10), we obtain the Equation (13)(the detailed derivation of Equations see Appendix).

$$L_w = \frac{1 - K_w}{K_w} \tag{13}$$

Equation (13) establishes the mapping from K_w to L_w.

Now we derive the linear gain K_w for data hiding error diffusion mapped to K_s. Using the watermarking linear gain $K_w x'(i,j)$ to reach the watermarked halftone output $y(i,j)$, this can be realized by minimizing the squared error between watermarked halftone and linear gain model output as indicated in Equation (14).

$$\min_{K_w} \sum_{i,j} (K_w x'(i,j) - y(i,j))^2 \tag{14}$$

In data hiding error diffusion, the criterion (5) can be approximated with an infinite small real number $\delta_1 \geq 0$ for the watermarking linear gain model:

$$|K_w x'(i,j) - y(i,j)| = \delta_1 \tag{15}$$

By relaxing the absolute value of Equation (15) (we know watermarked halftone $y(i,j) \in [0,1]$) , we obtain

$$K_w x'(i,j) = y(i,j) + \delta_1 \tag{16}$$

Recall in case 1, in order to minimize (5) for standard halftone linear gain K_s, we derive (17) with a small real number $\delta_2 \geq 0$

$$K_s x'(i,j) - y_0(i,j) = \delta_2 \tag{17}$$

Recall in case 2, replace $y(i,j)$ in Equation (5) with (12), and a small real value $\delta_3 \geq 0$, and relax absolute value, we obtain

$$K_s x'(i,j) + y_0(i,j) = 1 + \delta_3 \tag{18}$$

Combine (17) and (18), we obtain

$$2K_s x'(i,j) = 1 + \delta_2 + \delta_3 \tag{19}$$

From (16) and (19), for halftone image $|y(i,j)| \leq 1$, we obtain

$$\frac{K_w x'(i,j)}{2K_s x'(i,j)} = \frac{y(i,j) + \delta_1}{1 + \delta_2 + \delta_3} \leq 1 \tag{20}$$

Therefore, we derive $K_w \leq 2K_s$. The watermarked halftone linear gain can be represented by $K_w \in [K_s, 2K_s]$. As we mentioned K_w cannot be obtained without a watermark embedded. Each watermark embedded in the halftone image has a unique K_w value that can minimize the criterion $\sum_{i,j}(K_w x'(i,j) - y(i,j))^2$. This minimization is achieved by our proposed iterative linear gain model as described in section 4.

4 Iterative Linear Gain for Watermarking Error Diffusion

In Section 3, we analyzed sharpening distortion in data hiding watermarking error diffusion. In this section, we propose our visual quality preserving algorithm for data hiding error diffusion via iterative linear gain for watermarking halftone. By adjusting a multiplicative parameter L_w to compensate the input image in data hiding halftoning, the sharpening distortion decreases as the correlation between original image and residual image decreases. Thus, we can obtain the least distortion watermarked halftone image. This results in our proposed iterative linear gain model for optimum perceptual image quality of halftone images as illustrated in Figure 3.

4.1 Iterative Data Hiding Error Diffusion Algorithm

We found that the greater linear gain K_w the lesser the sharpening and harmonic distortion. To accurately measure a perceptual quality of a halftone image, Kite et al. [6] proposed the use of weighted SNR (WSNR) for the subjective quality measure of halftone image. WSNR weights the Signal-to-Noise Ratio (SNR) according to the contrast sensitivity function (CSF) of the human visual system. For an image of size $M \times N$ pixels, WSNR is defined as

$$WSNR(dB) = 10\log_{10}\left(\frac{\sum_{u,v}|(X(u,v)C(u,v)|^2}{\sum_{u,v}|(X(u,v) - Y(u,v)C(u,v)|^2}\right) \tag{21}$$

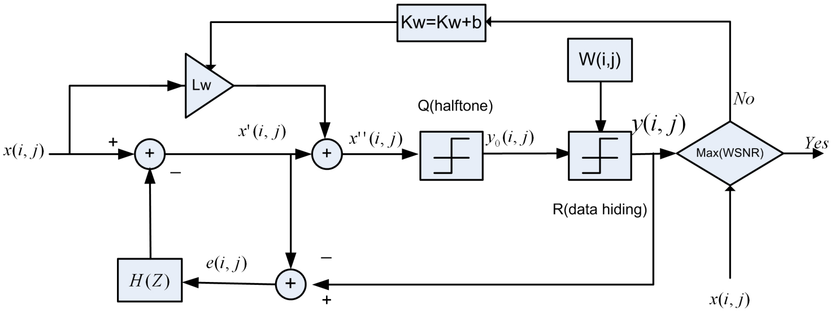


Fig. 3. Iterative linear gain halftone embedding

where $X(u, v), Y(u, v)$, and $C(u, v)$ represent the Discrete Fourier Transforms (DFT's) of the input image, output image, and CSF, respectively, and $0 \leq u \leq M-1$ and $0 \leq v \leq N-1$. With WSNR, we optimize K_w to minimize the halftone watermarking image distortion. In Section 3, as the K_w increases, the correlation between residual error image and input image is reduced. We use an iterative approach to find the best K_w for maximum WSNR of embedded halftone image. In this way, we can find the least distortion halftone watermarking image. Based on this concept, our new halftone embedding process is illustrated in Figure. 3. The embedding process first divides an image into macro blocks. Each macro

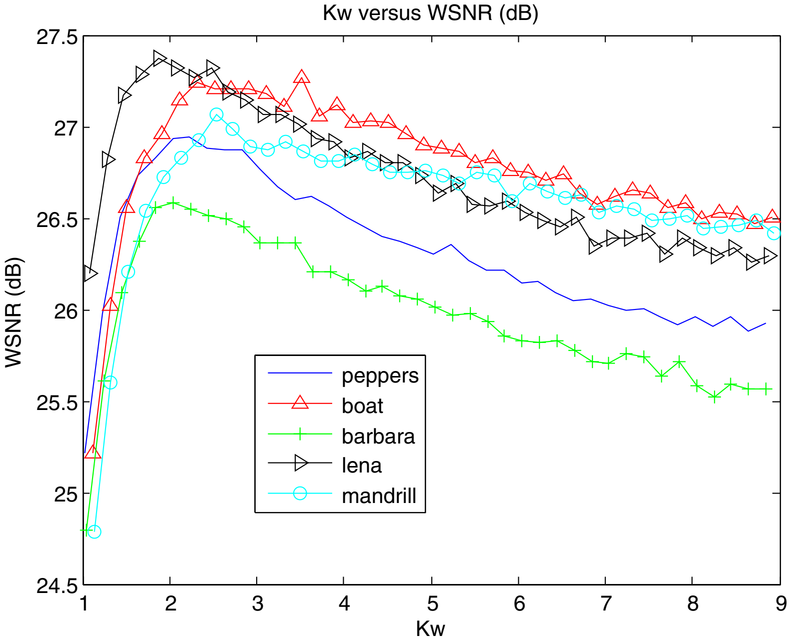


Fig. 4. The relationship between K_w and WSNR

block embeds one bit. Toggling halftone pixels is the same as Fu and Au's method [2] so that if the bit of watermark is the same as the original halftone pixel, no action is taken. If the watermark bit is different from the halftone pixel, then the bit is toggled. Iteration starts from $K_w = K_s$ to process embedding in halftone, and loops by adding one additive amount b, i.e., $b = 0.2$ to K_w until the WSNR reaches the maximum.The relationship between K_w and WSNR is presented in Figure. 4, which illustrates the principle of the iterative algorithm. Each image in Figure. 4 has a peak value of WSNR corresponding to a unique K_w which minimizes the criterion indicated in Equation (14).

5 Experiments and Results Analysis

Modified Peak Signal-to-Noise Ratio (MPSNR) [2] and Weighted Signal-to-noise Ratio(WSNR) [6] have been commonly used for evaluating halftone image quality. However, one main disadvantage of MPSNR is that it only compares the original image with the watermarked halftone image. The watermarked halftone image first undergos a Gaussian low-pass filter while the original image still contains high frequency components. This results in the inaccurate calculation of SNR because errors are incurred due to high frequency components remained in the grayscale image.

Our experiments were performed on five images with different sizes of watermark embedded using The University of Surrey logo in Jarvis kernel. Each experiment uses the same embedding locations and different watermark sizes. Figure 2(b) is lena halftone image with 10080 bits (90*112) embedded at K_w= 2.077 with WSNR=27.37 (dB). We use WSNR metric for subjective quality comparison of watermarked halftone quality as given in Table 1 and MPSNR in Table 2. From Table 1, our approach has an average improvement of 6.5% over Fu and Au's method. MPSNR measure shows that our approach is slightly higher than Fu and Au's method except for the image mandrill and barbara in high capacity embedding. This may be caused by the fact that both mandrill and barbara contained some high frequency components.

Table 1. WSNR Comparison Between Our's Method and Fu and Au's Method (dB)

mark size	32x32		64x64		90x90		90x112		Avg. Impr.%
image	our	Fu & Au	our	Fu & Au	our	Fu & Au	our	Fu & Au	our impr.
peppers	27.643	25.273	27.380	25.164	27.073	25.098	26.936	25.068	8.392
boat	27.582	24.674	27.533	24.673	27.231	24.609	27.112	24.570	11.470
barbara	27.224	24.923	27.133	24.825	26.778	24.669	26.558	24.585	8.734
lena	27.729	26.038	27.618	25.920	27.449	25.868	27.375	25.830	6.565
mandrill	27.222	24.116	27.066	24.094	26.972	24.076	26.909	24.005	12.474

Figure 5 illustrates the results of applying our proposed method and Fu and Au's method data hiding error diffusion to five test images. From this figure, we conclude our approach can preserve the high quality of watermarked halftone

Table 2. MPSNR Comparison Between Our's Method and Fu and Au's Method (dB)

mark size	32x32		64x64		90x90		90x112	
image	our	Fu and Au	our	Fu and Au	our	Fu and Au	our	Fu and Au
peppers	27.168	26.729	26.958	26.567	26.705	26.429	26.601	26.327
boat	26.048	25.710	25.965	25.651	25.733	25.455	25.598	25.404
barbara	24.095	24.078	23.998	23.997	23.844	23.856	23.772	23.785
lena	27.017	26.917	26.895	26.783	26.701	26.653	26.623	26.560
mandrill	22.620	22.724	22.581	22.670	22.505	22.620	22.463	22.572

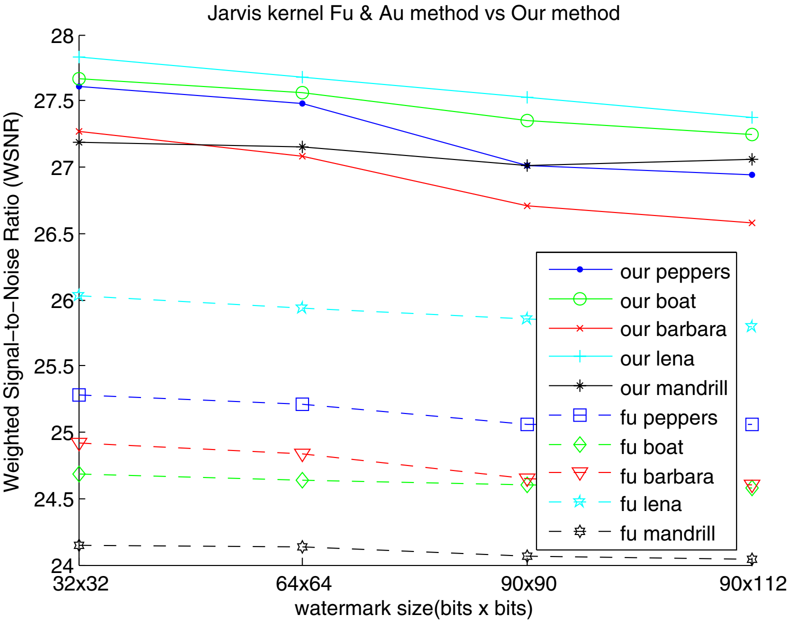


Fig. 5. WSNR of Fu method vs Our method

with different sizes of watermark embedded. Overall Fu and Au's method cannot maintain the WSNR as good as our method for different host images. This is due to our iterative linear gain model can effectively compensate the watermarking effects during halftone error diffusion process.

Figure 6 illustrates the percentage of improvement of our method over Fu and Au's method. Even for the worst case image *lena*, our method achieved an improvement of approximately 6-7% compared to Fu and Au's method. The other watermarked halftone images are shown in Figure 8.

5.1 Tonality Validation in Watermarked Halftone Image

Idle tones appears as strong cycle patterns. Idle tones affect the quality of halftone. Kite et.al [6] analogized the halftone distortion, which was caused by

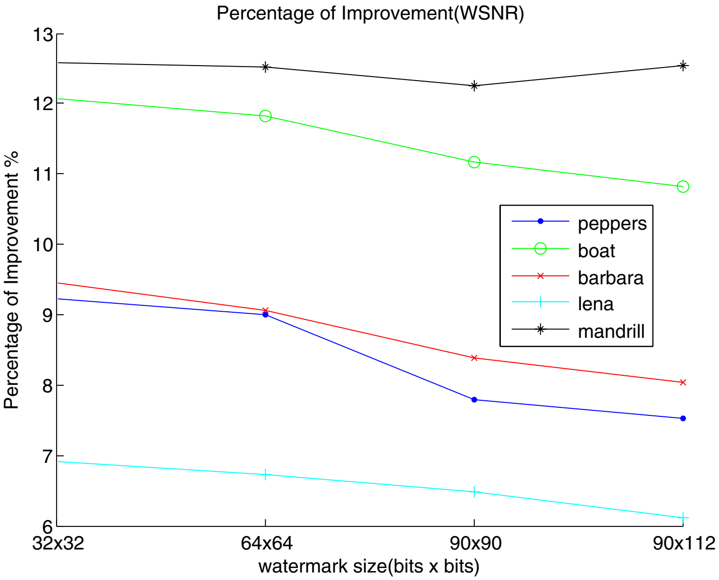


Fig. 6. Percentage of improvement of our method

idle tone, with total harmonic distortion. By computing the power spectral density of watermarking halftone, we propose our method to adapt to the *total harmonic distortion* [3] for analyzing tonality, i.e. the signal power distribution over frequencies. The power spectral density (PSD), describes how the power (or variance) of a time series is distributed with frequency. Mathematically, it is defined as the Fourier Transform of the autocorrelation sequence of the time series. Let x is the signal of halftone image, the PSD is the Fourier transform of the autocorrelation function,$autocorr(\tau)$, of the signal if the signal can be treated as a stationary random process,

$$S(x) = \int_{-\infty}^{\infty} autocorr(\tau)\, e^{-2\,\pi\, i\, x\, \tau}\, d\tau. \tag{22}$$

$$PSD = \int_{F_1}^{F_2} S(x)\, dx + \int_{-F_2}^{-F_1} S(x)\, dx. \tag{23}$$

where the power of the signal in a given frequency band can be calculated in (23) by integrating over positive and negative frequencies.

The spectral density is usually estimated using Welch's method [8], where we define a Hann window to sample the signal x. To this end, we define the average PSD under a Hann window of 512 samples (two-side 256 sample dots)as

$$\overline{PSD} = \frac{1}{257} PSD \tag{24}$$

For the Jarvis kernel, we embedded the surrey logo(90x112 bits) into the image *boat* of sizes 512x512. For $\overline{PSD}$ comparison, we analyzed the tonality of our proposed watermarking model compared with Fu and Au's method. The $\overline{PSD}$s of image *boat* are illustrated in Figure 7. As described in Section 3, our proposed model reduces the tonality of watermarked halftone images by adaptively adjusting K_w. This figure shows that Fu and Au's method(red line) generated higher $\overline{PSD}$ than the original halftone (green line). However, our method (blue line, while K_w=2.514) achieved $\overline{PSD}$ much smoother than the others. The same experiments were performed to all five images, and measured average power spectral desnity ($\overline{PSD}$) to all five images, as shown in Table 3. Where the K_w^1 in Table 3 is the initial value of K_w and the last K_w^{opt} is the optimum value of K_w. We found that image peppers's halftone has zero $\overline{PSD}$. This means its autocorrelation function is zero. However, when a watermark was embedded, it introduced harmonic distortion. The $\overline{PSD}$ for watermarked image peppers was approximately 0.0096 (dB/Hz) for both Fu and Au's method and our proposed method. Based on our model, we also found that the overall $\overline{PSD}$ of five images was reduced as K_w increased until it reached approximately $2K_w^1$. For example, image *boat*'s $\overline{PSD}$ reduces from 0.0893 (dB/Hz) (K_w^1=1.114) to 0.0096 (dB/Hz)(K_w^{opt}=2.514). We conclude that the lower the value and more

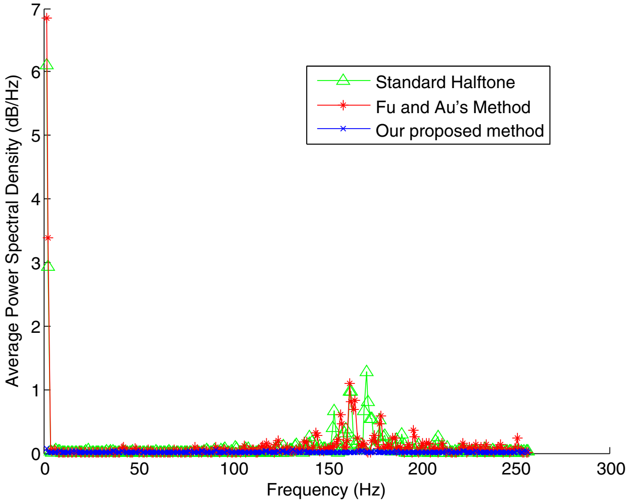
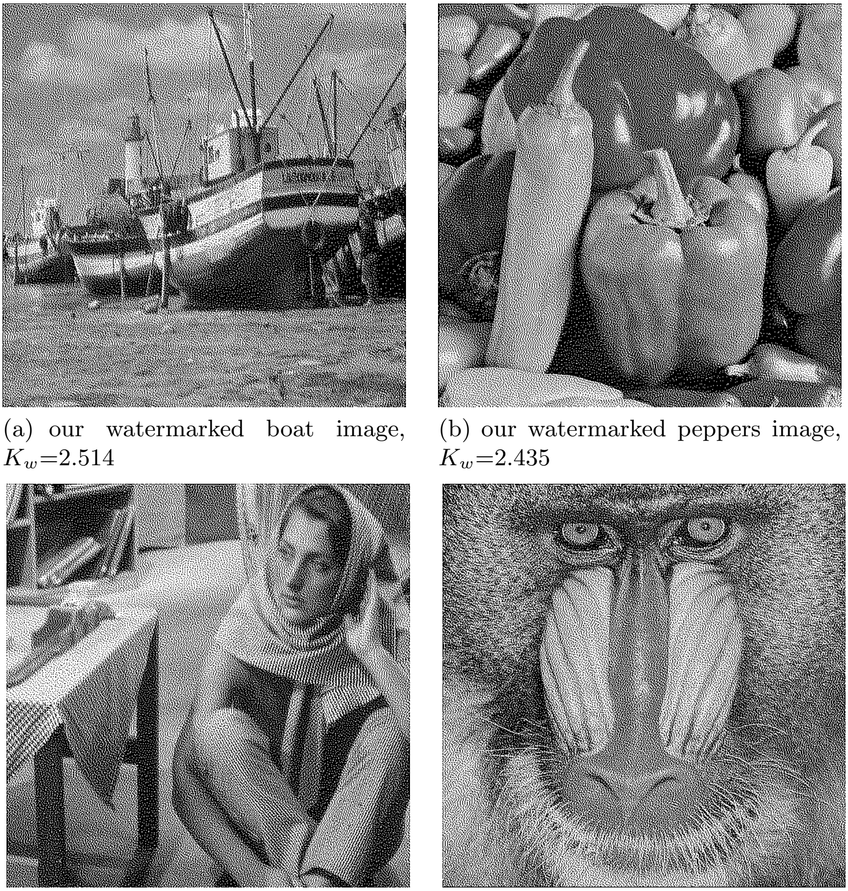


Fig. 7. Average Power Spectral Density of halftone boat images

Table 3. Average Power Spectral Density

image	halftone	Fu and Au method	our proposed method (dB/Hz)	
peppers	0	0.0096	0.0096(K_w^1=1.035)	0.0096(K_w^{opt}=2.435)
boat	0.1053	0.1122	0.0893(K_w^1=1.114)	0.0096(K_w^{opt}=2.514)
barbara	0.2225	0.2185	0.2153(K_w^1=1.045)	0.1071(K_w^{opt}=2.445)
lena	0.0535	0.0521	0.0507(K_w^1=1.077)	0.0099(K_w^{opt}=2.477)
mandrill	0.1796	0.1914	0.1632(K_w^1=1.132)	0.0322(K_w^{opt}=2.532)



(a) our watermarked boat image, K_w=2.514

(b) our watermarked peppers image, K_w=2.435

(c) our watermarked barbara image,K_w=2.445

(d) our watermarked mandrill image,K_w=2.532

Fig. 8. Watermarked Halftone Images with Surreylogo 10080 bits embedded

uniformly distributed the $\overline{PSD}$, the lower would be the harmonic distortion. By finding an optimum K_w, the least distorted watermarked halftone image is obtained. Therefore, an optimum perceptual quality is preserved via minimizing distortion in the data hiding error diffusion halftone images.

6 Conclusion

In this paper, we analyzed the perceptual distortion of data hiding watermarking in error diffusion of halftone image. Two major impacts generated by data hiding error diffusion of halftone images have been clarified as: firstly, sharpening distortion, as a consequence of the quantization error image being correlated with original grayscale image in the standard data hiding error diffusion; Secondly, the error diffused by *an error diffusion kernel* is distributed directionally in standard data hiding error diffusion process. It causes tonality, which results in strong cycle patterns. From our mathematical framework, we proposed the watermarking linear gain model for data hiding error diffusion halftoning and derived our optimized linear gain parameter K_w by mapping it to the halftoning linear gain model K_s. This model minimizes a significant amount of distortion resulting in 6.5% to 12% improvement of WSNR, which finalized an embedded halftone image to be unsharpened and quantization error becoming an uncorrelated Gaussian white noise. This model adopts an iterative approach to minimizing the impacts of distortion in the data hiding error diffusion process. In other words, it minimizes the differences between the grayscale image and the embedded halftone image. Consequently, the WSNR quality for halftone watermark embedding was found to be better than Fu and Au's method. The proposed linear gain control model was also validated using an average power spectral density.

Our future work will focus on robust watermarking approach for halftone image using error correction coding. Moreover, authentication and restoration will also be investigated.

Acknowledgments

We would like to thank Dr. Thomas D. Kite and Prof. Brian L. Evans from the University of Texas for their invaluable advice. We wish to thank Mr. Chris Culnane and Mr. Vinod Pankajakshan in our Watermarking and Multimedia Security group for their comments and feedback during the preparation of this paper.

References

1. Floyd, R., Steinberg, L.: An adaptive algorithm for spatial grayscale. Proceedings of the. Society for Information Display 17(2), 75–77 (1976)
2. Fu, M.S., Au, O.C.: Data hiding watermarking for halftone images. IEEE Transactions on Image Processding 11(4) (April 2002)
3. Horowitz, P., Hill, W.: The art of Eletronics. Combridge Univ. Press, Cambridgem UK (1980)
4. Hsieh, C.-T., Lu, Y.-L., Luo, C.-P., Kuo, F.-J.: A study of enhancing the robustness of watermark. In: Proceedings of International Symposium on Multimedia Software Engineering, 2000., pp. 325–327 (2000)
5. Jarvis, W.N.J., Judice, C.: A survey of techniques for the display of continuous tone pictures on bilevel displays. Comp. Graph. and Image Proc 5, 13–40 (1976)

6. Kite, T.D., Evans, B.L., Bovik, A.C.: Modeling and quality assessment of halftoning by error diffusion. IEEE Transactions on Image Processing 9(5) (May 2000)
7. Knox, K.: Error image in error diffusion. In: SPIE, Image Processing Algorithms and Techniques III, vol. 1657, pp. 268–279 (1992)
8. Lfeachor, E.C., Jervis, B.W.: Digital Signal Processing: A Practical Approach. In: Dagless, E.L. (ed.). Addison-Wesley, Reading (1993)
9. U. of Surrey, "Surrey logo" (2007), `http://www.surrey.ac.uk/assets/images/surreylogo.gif`
10. Petitcolas, F., Anderson, R., Kuhn, M.: Information hiding-a survey. IEEE Proceedings 87(7), 1062–1078 (1999)
11. Solanki, K., Madhow, U., Manjunath, B.S., Chandrasekaran, S., El-Khalil, I.: 'print and scan' resilient data hiding in images. IEEE Transactions on Information Forensics and Security (August 2006), `http://vision.ece.ucsb.edu/publications/solanki_TIFS06.pdf`
12. Soo-Chang Pei, J.-M.G.: High-capacity data hiding in halftone images using minimal-error bit searching and least-mean square filter. IEEE Transactions on Image Processding 15(6) (June 2006)
13. Ulichney, R.: Digital Halftoning. MIT Press, Cambridge (1987)
14. Williams, R.: Electrical Engineering Probability. West, St.Paul,MN (1991)
15. Wu, C.W., Thompson, G., Stanich, M.: Digital watermarking and steganography via overlays of halftone images. IBM Research Division,Thomas J. Watson Research Center, P.O. Box 218,Yorktown Heights, NY 10598, IBM Research Report (July 2004)

Appendix

This appendix is derivation of the Equation L_w. In [6], the standard error diffusion transfer equation with signal transform function and noise transform function, can be expressed as z-transformation

$$Y(z) = \underbrace{\frac{K_s}{1+(K_s-1)H(z)}}_{STF} X(z) + \frac{1-H(z)}{1+(K_n-1)H(z)} N(z) \tag{25}$$

Replace Equation (12) into Equations (6) and (10). We obtain:

$$e(i,j) = 1 - y_0(i,j) - x'(i,j) \tag{26}$$

$$x'(i,j) = x(i,j) - h(i,j) * e(i,j) \tag{27}$$

$$x''(i,j) = x'(i,j) + L_w x(i,j) \tag{28}$$

$$y_0(i,j) = Q(x''(i,j)) \tag{29}$$

$$y(i,j) = 1 - y_0(i,j) \tag{30}$$

Substituting $x'(i,j)$ in Equation (27) into Equation (28), and taking z transform, we have

$$X''(z) = (1 + L_w)X(z) - H(z)E(z) \tag{31}$$

From Equation (26) and Equation (27), taking the z transform, we have

$$E(z) = \frac{\frac{Z}{Z-1} - Y_0(z) - X(z)}{1 - H(z)} \tag{32}$$

From Equation (31) and Equation (32), we derive

$$X''(z) = [1 + L_w + \frac{H(z)}{1-H(z)}]X(z) + \frac{H(z)}{1-H(z)}Y_0(z) - \frac{Z}{Z-1}\frac{H(z)}{1-H(z)} \tag{33}$$

In Figure 9, we draw the equivalent modified circuit to watermarking linear gain model. we obtain

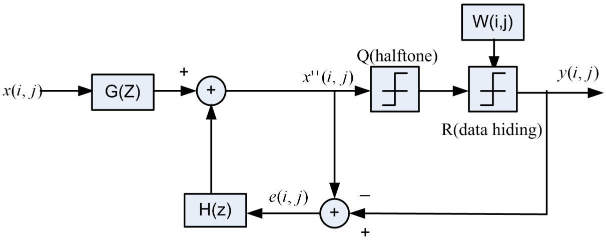


Fig. 9. Equivalent modified circuit

$$e(i,j) = 1 - y_0(i,j) - x''(i,j) \tag{34}$$
$$x''(i,j) = g(i,j) * x(i,j) - h(i,j) * e(i,j) \tag{35}$$
$$y_0(i,j) = Q(x''(i,j)) \tag{36}$$
$$y(i,j) = 1 - y_0(i,j) \tag{37}$$

where $g(i,j)$ is an impulse response of G(z). From Equation (34) and Equation (35), we have

$$E(z) = \frac{\frac{Z}{Z-1} - Y_0(z) - G(z)X(z)}{1 - H(z)} \tag{38}$$

we substitute Equation (38) into the z-transform of Equation (35) ,we derive

$$X''(z) = [G(z) + \frac{G(z)H(z)}{1-H(z)}]X(z) + \frac{H(z)}{1-H(z)}Y_0(z) - \frac{Z}{Z-1}\frac{H(z)}{1-H(z)} \tag{39}$$

Equation (33) is equal to Equation (39),when

$$1 + L + \frac{H(z)}{1-H(z)} = G(z) + \frac{G(z)H(z)}{1-H(z)} \tag{40}$$

Then, we obtain

$$G(z) = 1 + [1 - H(z)]L_w \tag{41}$$

If we compare Equation (41) with the Signal Transform Function expressed in Equation (25),in which the watermarked halftone image with linear gain K_w has the same signal transfer function, G(z) can be expressed as the reciprocal of the STF. Thus,

$$L_w = \frac{1 - K_w}{K_w} \tag{42}$$

□

Multiple Scrambling and Adaptive Synchronization for Audio Watermarking

Yiqing Lin and Waleed H. Abdulla

Department of Electrical and Computer Engineering
The University of Auckland, New Zealand
y.lin@ece.auckland.ac.nz, w.abdulla@auckland.ac.nz

Abstract. Imperceptibility, robustness and security are the vital considerations in the design of any audio watermarking scheme for copyrights protection. In this paper, a secure and robust audio watermarking scheme involving multiple scrambling and adaptive synchronization is proposed. To prevent the unauthorized detection, the new scheme integrates multiple scrambling operations into the embedding process. That is, encrypting the watermark with a coded-image and randomly selecting certain subbands for the embedding process. Moreover, the detection utilizes adaptive synchronization to enhance the robustness under some destructive de-synchronization attacks, like random samples cropping/inserting, pitch-invariant time stretching, and tempo-preserved pitch shifting. Theoretical analysis and simulation results have revealed that the proposed scheme is self-secured indeed and also immune to a wide range of severe attacks.

Keywords: audio watermarking, information hiding, copyrights protection, multimedia watermarking.

1 Introduction

Digital watermarking is the process of embedding imperceptible information within digital multimedia products for the purpose of content authentication, data monitoring and tracking, and copyrights protection [1,2]. Amongst the applications, the most prominent function of watermarking techniques is helping in identifying the producer of multimedia files and resolving ownership disputes [2].

Audio watermarking is the application on audio signals, a more challenging field than image and video watermarking [3]. As an effective tool to enforce copyrights, any eligible audio watermarking scheme must meet the following requirements [2,4]. Firstly, imperceptibility is a prerequisite to practicality. To preserve the perceptual quality of the watermarked data, psychoacoustic model based on audio masking phenomenon will be relied on to deceive human perception of the audio signals [5,6]. Consequently it appears as if there is nothing added to the host media. Secondly, robustness is a prerequisite to reliability, which refers to the capability of resisting a variety of unintentional and intentional attacks. For example, noise addition, resampling, lowpass filtering, echo addition, reverberation and

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 440–453, 2008.

MPEG compression are some of common signal processing manipulations. Also, random samples cropping/inserting, jittering, time stretching and pitch shifting belong to de-synchronization attacks [7,8], which would cause the displacement and threaten the survival of the watermark heavily [9]. Thirdly, security is a prerequisite to existence. Since the watermarking algorithms are likely to be open to the public, we should guarantee that the watermarks cannot be ascertained even by reversing the embedding process or performing statistical detection [4].

Our previously implemented audio watermarking scheme in [10] has excellent capabilities for copyrights protection. In this paper, we improve its performance in terms of security and robustness using multiple scrambling and adaptive synchronization. Since every scrambling operation has its own secret key; a pseudorandom sequence, the detection can only be conducted properly if all the secret keys are known. Although the unauthorized crackers might know the way of watermarking, it is still nearly impossible to approach the embedded watermark. In addition, to cope with various signal manipulations (especially de-synchronization attacks) on the watermarked data, adaptive synchronization is developed to search for the best synchronization position during the detection, so that we are able to revive the watermark from the attacked audio files suffering loss of synchronization.

The paper is structured as follows. In Section 2, the embedding algorithm including multiple scrambling is depicted. Section 3 is focused on the detection method where adaptive searching is emerged. Then, we carry on a complete performance assessment in Section 4. Finally, Section 5 presents the conclusion and future work.

2 Watermark Embedding

The way of embedding a watermark is to modulate the amplitude of the host audio signal in the time-frequency plane [10,11]. In this section, the basic algorithm integrated with multiple scrambling is described.

2.1 Embedding Procedure

Firstly, a pre-selection is applied on the host audio signal to determine the embedding segments. Only the regions whose power exceeds a certain threshold will be chosen for watermarking, so that silent portions and trifle intervals are skipped, as shown in Figure 1. This is because embedding watermarks in silent segments would introduce unavoidable perceived noise [12].

Secondly, short-time Fourier transform (STFT) of adjacent audio frames with 50% overlap is taken to obtain the time-frequency representation of the considered segments. These segments are further divided into N_{block} pattern blocks, as shown in Figure 2. Each pattern block is used to embed one sub-watermark $B_{sub(m)}$, which is a sub-part of the original watermark W_o and contains N_{bit} watermark bits $B_i \in \{1, -1\}$, i.e.

$$W_o = \{B_{sub(m)}\} \qquad m = 1, \cdots, N_{block} \tag{1}$$

where $B_{sub(m)} = \{B_i^{(m)}\}$, $i = 1, \ldots, N_{bit}$.

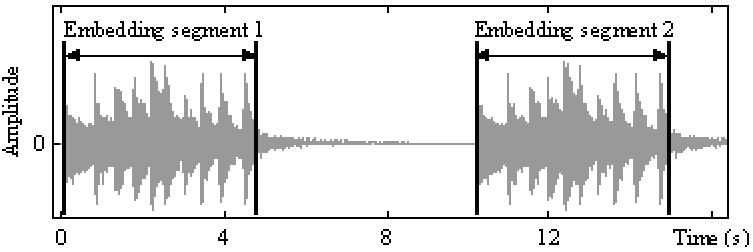


Fig. 1. Pre-selection for embedding segments

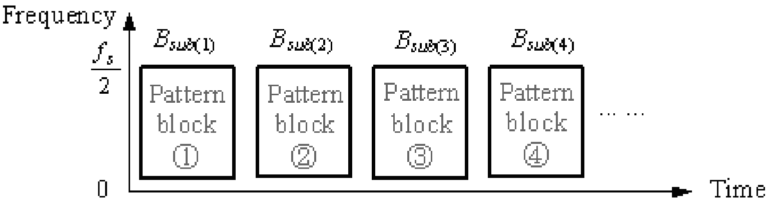


Fig. 2. Pattern blocks in one embedding segment

The obtained two-dimensional pattern blocks shown in Figure 2 are segmented into different levels of granularities, that is, unit, subband, slot, tile and bin (see Figure 3).

Along the time axis, every pattern block is divided into N_{unit} units, each of which fixedly comprises four frames. Thus, one pattern block has $4N_{unit}$ frames. Along the frequency axis, the pattern block is divided into $N_{subband}$ non-uniform perceptually motivated subbands based on Gammatone filterbank (GTF) [13]. The intersection of a subband and a unit is called slot, which is

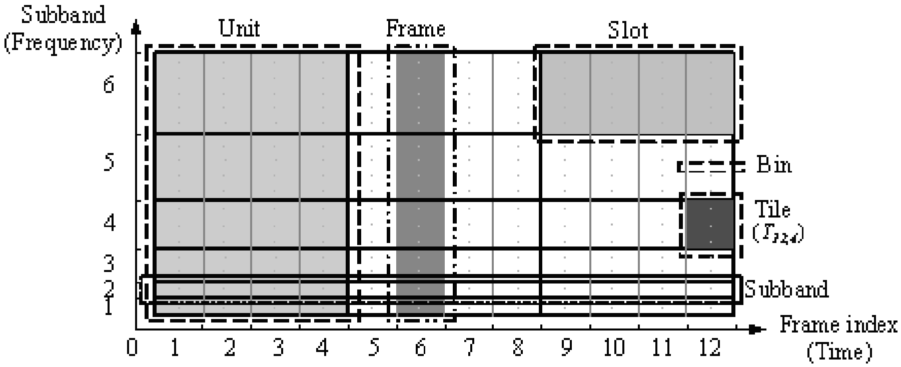


Fig. 3. Configuration of the pattern block

assigned with one watermark bit $B_i \in \{1, -1\}$ or synchronization bit B_S=1, and one pseudorandom number (PN) $P_j \in \{1, -1\}$. That is, N_B slots are randomly chosen from the total number of slots within each pattern block for embedding each B_i, and then the rest are for embedding B_S, whose total number is N_S. The value of N_S is calculated as

$$N_S = N_{unit} * N_{subband} - N_{bit} * N_B \tag{2}$$

In fact, the distribution of the bits in the pattern blocks is determined by one secret key, which belongs to confidential information shared only between the embedder and the recognized detectors. Based on that, different bits are dispensed in every pattern block, as exemplified in Figure 4. Without loss of generality, it is assumed here that each sub-watermark consists of two watermark bits, i.e. $B_{sub} = \{B_1, B_2\}$, which are allocated with five slots separately, i.e. N_B=5 . Thereby, N_S=3∗6-2∗5=8 slots are for embedding B_S. Then the sign of the slot is determined by the multiplication of its bit and the corresponding *PN*. Finally, the finer element of the slot is tile, the primitive module for amplitude modulation. In practice, the tiles will be forced to contain several FFT bins, since single frequency coefficient is sensitive towards slight modification.

2.2 Embedding Algorithm

Coded-Image Watermark

In our scheme, we adopt coded-image like with bit '1' and '0' (mapped to '-1') as visual watermark [14], instead of meaningless pseudorandom or chaotic sequence. Not only because coded-image can be identified visually (after decryption), which is a kind of ownership stamp indeed, but also post processing on the extracted watermark could be done to enhance the binary image [1] and consequently the detection accuracy will increase. Image denoising and pattern recognition are examples of post processing techniques for automatic character

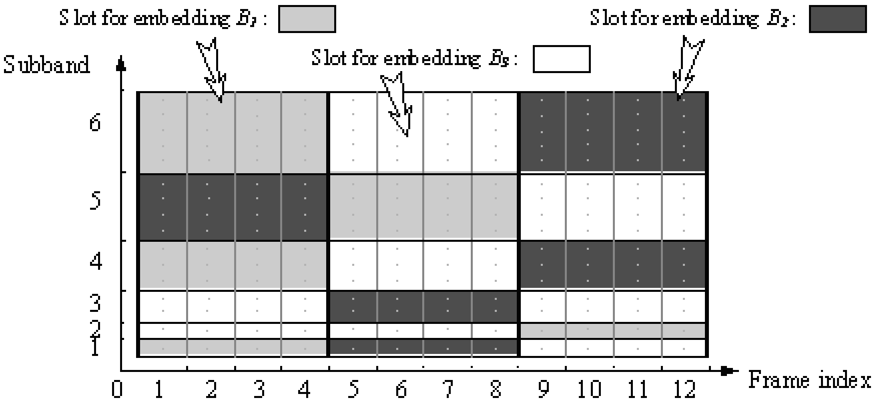

Slot for embedding B_1 :
Slot for embedding B_2 :
Slot for embedding B_S :
Subband
6
5
4
3
2
1
0
1
2
3
4
5
6
7
8
9
10
11
12
Frame index

Fig. 4. Distribution of the watermark bits and synchronization bit

recognition. Thus, on top of the bit error rate (*BER*), coded-image provides a semantic meaning for reliable verification.

Embedding Method

Generally, the watermark bits are embedded via amplitude modulation of the tiles. For each frame, the imperceptible watermark signal is constructed in frequency domain using the magnitude, phase and sign of the signal spectrum. Then it is added to the host frame after inversely transformed to time domain.

The magnitude spectrum of the watermark signal is estimated from the minimum masking threshold derived from the psychoacoustic model-I [5,15]. Thus, the noise introduced by the watermarking is kept inaudible to human ears. The phase of the watermark signal is the same as that of the host signal to avoid phase distortion. As for the signs, they are determined by spreading the sign of a slot over its four tiles. That is, if the sign of a slot is positive, the sign of the first two tiles in that slot is positive and that of the last two tiles is negative, and vice versa if the sign is negative. By this way, all the FFT bins obtain their own signs.

So the real and imaginary parts of frequency spectrum of the watermark signal are constructed as follows.

$$\begin{array}{l}\text{Real part} = \text{sign}*\text{magnitude}*\cos(\text{phase}) \\ \text{Imaginary part} = \text{sign}*\text{magnitude}*\sin(\text{phase})\end{array} \tag{3}$$

After smoothly concatenating all the watermarked frames, the overall watermarked audio is formed.

Multiple Scrambling

To increase the level of security, multiple scrambling can be used in the embedding.

The first scrambling operation is to encrypt the coded-image watermark into incomprehensible ciphers, where one secret key is used. Furthermore, instead of using all the subbands, we randomly select $\tilde{N}_{subband}$ out of $N_{subband}$ subands and randomize their orders of encoding, where two secret keys are employed. So the number of possible ways is calculated by the following permutation

$$N_{scrambling} = P^{\tilde{N}_{subband}}(N_{subband}) = \frac{N_{subband}!}{(N_{subband} - \tilde{N}_{subband})!} \tag{4}$$

Along with the random settings on the amount and positions of slots assigned to each watermark bit, anyone without all the secret keys rarely has the possibility to find out the watermark. Since the secret keys are shared only between the embedder and authorized detectors, the aim of copyrights protection is really achieved.

3 Watermark Detection

In the detection, the watermark bits are determined by inspecting whether it is an increase or decrease in the magnitudes of the corresponding tiles of the received signal [6,10]. By correlating to the PNs, we can extract the watermark without resorting to the host signal. To exactly locate the tiles assigned to each bit, a precise synchronization must be performed to find out the beginning of every pattern block. It is especially important towards some de-synchronization attacks, such as pitch-invariant time stretching (PITS) and tempo-preserved pitch shifting (TPPS) [8,9]. Both can be implemented by audio editing tools without large perceptual impairment. To combat such challenging attacks, we utilize adaptive synchronization to search for the best synchronization position. Random stretching attack used by [6,16] which is implemented by omitting or inserting a random number of samples (usually called 'random samples cropping/inserting') and pitch shifting attack by linear interpolation are much less complicated than PITS and TPPS.

3.1 Basic Detection Algorithm

After applying pre-selection and STFT on the received signal, a series of pattern blocks is built for each segment. The detection algorithm will work on every pattern block and extract its embedded watermark bits.

Firstly, the amplitude spectrum of each FFT frame in the pattern blocks is normalized by its average and mapped into logarithmic scale.

$$\hat{F}_t = 20\log_{10}(\bar{F}_t) \tag{5}$$

where F_t refers to the t-th FFT frame (with N bins) and

$$\bar{F}_t = \frac{\mid F_T \mid}{(\frac{1}{N} \cdot \sum_{n=1}^{N} \mid F_{t,n} \mid)} \tag{6}$$

Secondly, to increase the effect of the watermark signal and reduce the effect of the host signal, the amplitude difference $\tilde{F}_t$ between each frame and the one just after the next is calculated, for the considered frames are actually non-overlapped with each other.

$$\tilde{F}_t = \hat{F}_t - \hat{F}_{t+2} \tag{7}$$

Then, we can get the magnitude of the tile located at the b-th subband of the t-th frame, $Q_{t,b}$.

$$Q_{t,b} = \frac{\sum_{n=V_b^l}^{V_b^h} \tilde{F}_{t,n}}{V_b^h - V_b^l + 1} \tag{8}$$

where V_b^l and V_b^h refer to the lower and upper bounds of the b-th subband, respectively.

Thirdly, pattern block synchronization is executed to find out the beginning frame of each pattern block, as described below.

Since every frame in its block is possible to be the starting frame, it is necessary to calculate synchronization strength $S_d(d{=}1,\ldots,4N_{unit})$ frame by frame.

On the assumption that the d-th frame is the starting frame, S_d for the d-th frame is

$$S_d = \frac{\sum_{k=1}^{N_S}[P_S(k) \cdot (Q_{t(d,k),b(k)} - \bar{Q}_S)]}{\sqrt{\sum_{k=1}^{N_S}[P_S(k) \cdot (Q_{t(d,k),b(k)} - \bar{Q}_S)]^2}} \tag{9}$$

where $\{Q_{t(d,k),b(k)}\}$ represents N_S tiles that are allotted to B_S and $\{P_S(k)\}$ corresponds to their PNs. For $\bar{Q}_S$, it is the average of $\{Q_{t(d,k),b(k)}\}$, i.e.

$$\bar{Q}_S = \frac{1}{N_S}\sum_{k=1}^{N_S} Q_{t(d,k),b(k)} \tag{10}$$

The subscripts $t(d,k)$ and $b(k)$ are computed by

$$t(d,k) = d + [R_S(k,1) - 1] * 4 \tag{11}$$

$$b(k) = R_S(k,2) \tag{12}$$

Here, factor 4 comes from the number of frames per unit. In addition, R_S is called index matrix for B_S, which solely depends on the secret key, as mentioned above. To indicate where those tiles are, the first and second column of R_S is defined as the distribution of the tiles' columns and rows, respectively. Given Figure 4, for example, its R_S is

$$R_S = \begin{array}{cc} \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 2 & 2 \\ 2 & 4 \\ 2 & 6 \\ 3 & 1 \\ 3 & 3 \\ 3 & 5 \end{bmatrix} \\ \textbf{Unit index} \quad \textbf{Subband index} \end{array}$$

Then, the beginning frame of this block (d_{sync}) is the frame that provides the maximum S_d, i.e.

$$d_{sync} = \arg \max_{1 \le d \le 4N_{unit}} (S_d) \tag{13}$$

Also, the maximum S_d of that d_{sync} is denoted as $S_{d_{sync}}$.

Fourthly, the bit strength (G_j) of each watermark bit B_j is calculated by

$$G_j = \frac{\sum_{k=1}^{N_B}[P_{B_j}(k) \cdot (Q_{t(d_{sync},k),b(k)} - \bar{Q}_{B_j})]}{\sqrt{\sum_{k=1}^{N_B}[P_{B_j}(k) \cdot (Q_{t(d_{sync},k),b(k)} - \bar{Q}_{B_j})]^2}} \tag{14}$$

where

$$t(d_{sync},k) = d_{sync} + [R_{B_j}(k,1) - 1] * 4 \tag{15}$$

$$b(k) = R_{B_j}(k,2) \tag{16}$$

Similarly, $\{Q_{t(d_{sync},k),b(k)}\}$ and $\{P_{B_j}(k)\}$ refer to N_B tiles allotted to B_j and their corresponding PNs respectively, and $\bar{Q}_{B_j}$ is the average again. As for R_{B_j}, it is the index matrix for B_j. Based on Figure 4, R_{B_1} and R_{B_2} are

$$R_{B_1} = \begin{bmatrix} 1 & 1 \\ 1 & 4 \\ 1 & 6 \\ 2 & 5 \\ 3 & 2 \end{bmatrix} \quad \text{and} \quad R_{B_2} = \begin{bmatrix} 1 & 5 \\ 2 & 1 \\ 2 & 3 \\ 3 & 4 \\ 3 & 6 \end{bmatrix}$$

After that, the value of B_j is decided according to its bit strength G_j.

$$\begin{array}{l} \text{If } G_j \geq 0, \text{ then } B_j = 1. \\ \text{If } G_j < 0, \text{ then } B_j = 0. \end{array} \tag{17}$$

At last, the watermark bits extracted from all the pattern blocks are rearranged to form an image. After decrypting with the corresponding secret key, the final watermark ($E_{watermark}$) is obtained.

3.2 Adaptive Synchronization

The basic detection algorithm cannot efficiently handle the de-synchronization attacks. Sometimes, although the d_{sync} from [13] can provide the best match with B_S, it is a false position for detecting the watermark bits.

Take cropping as an example of de-synchronization attack. Let us assume that the first three frames in the 2^{nd} pattern block are cropped, as shown in Figure 5. Since the distribution of B_S and the value of PNs are the same for each pattern block, d_{sync} would be the present 10^{th} frame that is originally the first frame of next pattern block. In such case, it is impossible to get the correct tiles for bit detection.

Consequently, we should consider the amount of misalignments (d_m) in $t(d,k)$ in (11) and replace (11) of

$$t(d,k) = d + [R_S(k,1) - 1] * 4 - d_m \tag{18}$$

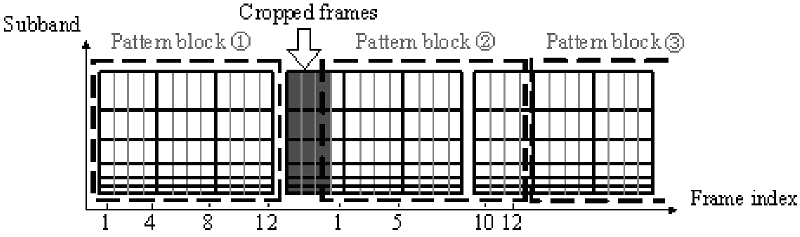


Fig. 5. Diagram of random samples cropping

where $d_m = (4N_{unit}) - d_{sync} + 2$. Then, re-execute pattern block synchronization with an updated $t(d,k)$ and re-calculate those $t(d,k)$ for the 2^{nd} pattern block, so as to get the actual d'_{sync}. Later, $t(d_{sync},k)$ in (15) for calculating G_j should be simultaneously modified into

$$t(d'_{sync},k) = d'_{sync} + [R_{B_j}(k,1) - 1] * 4 - d_m \tag{19}$$

It is worth mentioning that if $t(d,k)$ or $t(d_{sync},k)$ exceeds $Q_{t,b}$ matrix's dimension (happens when cropping the very start of the watermarked signal), the value of those tiles can be simply replaced by zero.

The procedure described above is called adaptive synchronization. The key of adaptive synchronization is to choose a threshold (T_d) for determining whether a d_{sync} is false or not. When d_{sync} is larger than T_d, we assume that d_{sync} is untrue. So d_{sync} should be recomputed by adaptive synchronization and eventually reaches a value lower than T_d, which will be taken as the correct d'_{sync}.

Note that under different attacks, T_d is varied between

$$T_d = 0.7 * (4N_u) \sim 0.9 * (4N_u) \tag{20}$$

Thus, we need to do adaptive synchronization for each possible T_d, and then calculate the average synchronization strength (A_S) that is defined as

$$A_S = \frac{1}{\sqrt{N_b}} \sum_{m=1}^{N_b} S^{(m)}_{d'_{sync}} \tag{21}$$

where $S^{(m)}_{d'_{sync}}$ is the m-*th* pattern block's $S_{d'_{sync}}$. Then, the T_d that provides the maximum A_S is regarded as the desired one. Experientially, an optimal value of T_d is $\lfloor 0.8 * (4N_{unit}) \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function.

4 Performance Evaluation

In the performance evaluation, three pieces of EBU SQAM disc tracks [17], tracks 40, 47 and 48, are taken as the host signals. Originally, they are stereo channels in WAVE format (44.1 kHz, 16 bit), but we only use one channel. As for the watermark, a coded-image with 9×35 pixels is adopted. All the results of the performance evaluation are summarized in Table 1, 2 and 3.

4.1 Security Analysis Due to Channel Scrambling

In our experiments, each pattern block is divided into 32 non-linear subbands, where 28 subbands are randomly selected for embedding. So the number of possible ways is

$$N_{scrambling} = P^{28}(32) = 32!/(32-28)! = 32!/4! \approx 1.1 \times 10^{34}$$

Such a huge number makes the unauthorized detection nearly impossible, which means that the property of the security has increased greatly. This is just one code complexity that is multiplied by the complexity introduced by the *PN*s.

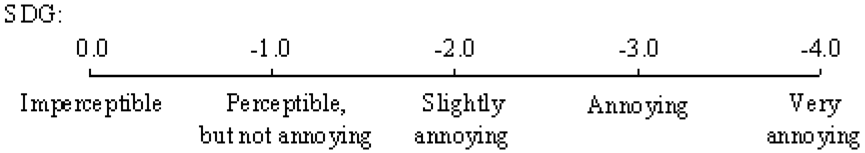


Fig. 6. Descriptions of subjective difference grade (SDG)

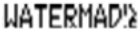
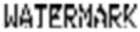
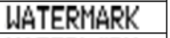
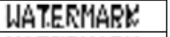
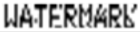
4.2 Perceptual Evaluation

Informal subjective listening test are performed in an isolated chamber, where ten trained listeners are participated. All the stimuli are presented through a high-fidelity headphone.

In the test, the listeners are asked to grade the quality of each watermarked signal related to its host signal based on the five-scale subjective difference grade (SDG) [18], as described in Figure 6.

Moreover, software 'PQevalAudio' for perceptual evaluation of audio quality (PEAQ) [18] is utilized to provide an objective difference grade (ODG), an

Table 1. Results of performance evaluation on Track 40 (*Harpsichord.wav*)

(i) Perceptual evaluation		*SSNR*	*SDG*	*ODG*
The watermarked signal		24.3dB	-0.25	-0.36
(ii) Robustness test		*BER* (%)		$E_{watermark}$
		The proposed scheme	The scheme in [6]	
No Attack		0	0	
Gaussian noise addition (40dB)		0	0	
Resampling ($f_{downsampling}$ = 22.05kHz)		0	14.56	
Lowpass filtering (f_{cutoff} = 8kHz)		8.54	22.47	
Echo addition (A_m = 0.3, t_{delay} = 0.1s)		0	0	
Reverberation (t_{reverb} = 1s)		0	0	
MP3 compression (mono)	96 kps	0	4.75	
	64 kps	0.32	18.99	
	48 kps	5.38	21.52	
Samples cropping (8 × 25ms)		0.63	30.06	
Jittering (0.1ms/0.4s)		0	0	
Zeros inserting (8 × 25ms)		0	7.91	
PITS: pitch-invariant time stretching	+4%	0	18.99	
	-4%	1.27	12.66	
TPPS: tempo-preserved pitch shifting	+4%	6.01	26.27	
	-4%	5.70	11.39	

objective measurement of SDG. In addition, the segmental signal-to-noise ratio (*SSNR*) [19] is computed as a reference.

Except that a few feel slight difference between track 47 and its watermarked signal, most listeners are hard to distinguish the host and the watermarked audio. Also, all the average SDG and ODG scores are within (-1.0, 0), which confirm that our watermarked signals are perceptually undistinguished from the host audio signals.

4.3 Robustness Test

Typical signal manipulations on audio watermarking schemes, for instance, noise addition, resampling, lowpass filtering, echo addition, reverberation, MP3 compression, random samples cropping, jittering, zeros inserting, pitch-invariant time stretching and tempo-preserved pitch shifting, are adopted to attack the watermarked signal. A premise of the robustness test is that the extent of deterioration by attacks should keep within an acceptable extent, since detection is needless to proceed on watermarked signal that is already 'severely' destroyed. Therefore, the amplitude of noise added and the extent of stretching or shifting are controlled within certain amount. Then bit error rate (*BER*) for every detection is calculated, not *BER* just based on correct detection in [6] and [16].

Table 2. Results of performance evaluation on Track 47 (*Bass.wav*)

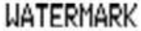
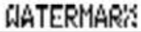
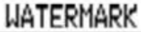
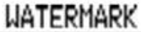
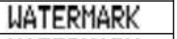
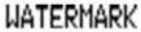
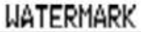
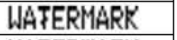
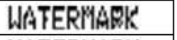
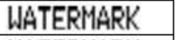
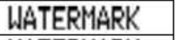
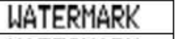
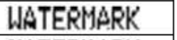
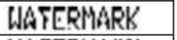
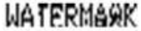
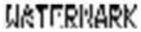
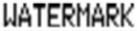
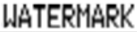
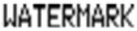
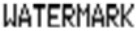
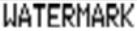
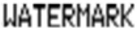
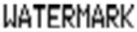
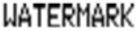
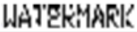
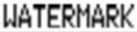
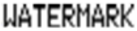
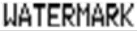
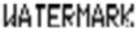
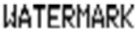
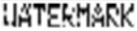
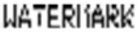
(i) Perceptual evaluation		*SSNR*	*SDG*	*ODG*
The watermarked signal		34.4dB	-0.58	-0.72
(ii) Robustness test		*BER (%)*		$E_{watermark}$
		The proposed scheme	The scheme in [6]	
No Attack		0	0	
Gaussian noise addition (40dB)		4.75	4.75	
Resampling ($f_{downsampling}$ = 22.05kHz)		0.63	0.63	
Lowpass filtering (f_{cutoff} = 8kHz)		0.32	29.75	
Echo addition ($A_m = 0.3$, $t_{delay} = 0.1$s)		0	0	
Reverberation ($t_{reverb} = 1$s)		0	0	
MP3 compression (mono)	96 kps	0.63	26.90	
	64 kps	1.27	26.27	
	48 kps	4.11	25.95	
Samples cropping (8 × 25ms)		0	37.34	
Jittering (0.1ms/0.4s)		0	0	
Zeros inserting (8 × 25ms)		0	23.73	
PITS: pitch-invariant time stretching	+4%	2.22	17.72	
	-4%	5.06	12.03	
TPPS: tempo-preserved pitch shifting	+4%	7.59	25.32	
	-4%	8.54	17.09	

Table 3. Results of performance evaluation on Track 48 (*Quartet.wav*)

(i) Perceptual evaluation		*SSNR*	*SDG*	*ODG*
The watermarked signal		29.6 dB	-0.41	-0.58
(ii) Robustness test		*BER* (%)		$E_{watermark}$
		The proposed scheme	The scheme in [6]	
No Attack		0	0	
Gaussian noise addition (40dB)		0	0	
Resampling ($f_{downsampling}$ = 22.05kHz)		0	0	
Lowpass filtering (f_{cutoff} = 8kHz)		0	26.90	
Echo addition (A_m = 0.3, t_{delay} = 0.1s)		0	9.18	
Reverberation (t_{reverb} = 1s)		0	17.41	
MP3 compression (mono)	96 kps	0	29.75	
	64 kps	0.32	28.16	
	48 kps	6.33	31.33	
Samples cropping (8 × 25ms)		0	36.39	
Jittering (0.1ms/0.4s)		0	5.70	
Zeros inserting (8 × 25ms)		0.32	0.32	
PITS: pitch-invariant time stretching	+4%	2.85	11.08	
	-4%	0.63	11.71	
TPPS: tempo-preserved pitch shifting	+4%	6.65	22.15	
	-4%	5.06	17.72	

Nearly under all the situations, the proposed scheme outperforms the one in [6] in terms of *BER*. On top of that, the coded-image embedded in three tracks can always be extracted and clearly identified, even in cases of severe malicious synchronization tamperings, such as random samples cropping, zeros inserting, PITS and TPPS. It is obvious that we can get extra assistance in confirmation when using the coded-image instead of merely meaningless bits, especially the cases with higher *BER*. Although some pixels in the coded-image are missing or mistaken, we are still able to recognize the copyrights information.

5 Conclusions and Future Work

In this paper, we propose a secure and robust audio watermarking scheme based on coded-image watermark, multiple scrambling and adaptive synchronization. The usage of the coded image will not only provide a visual identification, but also improve the accuracy of watermark detection further by employing image processing techniques and pattern matching analysis. Furthermore, the new scheme is strictly self-protected by using multiple scrambling. Any attacker lacking all the secret keys is nearly impossible to ascertain or destroy the embedded watermark without noticeably degrading the signal. In addition, the detection with adaptive synchronization can retrieve the best synchronization position of the pattern blocks and extract the watermarks even under serious de-synchronization attacks. Experimental results of perceptual quality and robustness evaluation have verified that the proposed scheme is more secure and robust than those reported in [6,10] without sacrificing perceptual quality of the audio signals.

However, when the amount of distortions in PITS and TPPS attacks become larger (beyond ±4%), the *BER*s will deteriorate to some extent. So in future work, an efficient way is needed to combat excessive PITS and TPPS. In addition, theoretical analysis of the falsely positive and negative error probability of the watermark detection will further be investigated.

Acknowledgement. The authors would like to thank the reviewers for their valuable suggestions and comments on the paper.

References

1. Erküçük, S., Krishnan, S., Zeytinoğlu, M.: A Robust Audio Watermark Representation Based on Linear Chirps. IEEE Transactions on Multimedia 8(5), 925–936 (2006)
2. Petitcolas, F.A.P.: Watermarking Schemes Evaluation. IEEE Signal Processing Magazine 17(5), 58–64 (2000)
3. Hartung, F., Kutter, M.: Multimedia Watermarking Techniques. Proceedings of the IEEE 87(7), 1079–1107 (1999)
4. Dittmann, J., Steinebach, M., Lang, A., Zmudizinski, S.: Advanced Audio Watermarking Benchmarking. In: Proceedings of SPIE International Symposium Electronic Imaging, vol. 5306, pp. 224–235 (2004)
5. Swanson, M.D., Zhu, B., Tewfik, A.H., Boney, L.: Robust Audio Watermarking Using Perceptual Masking. Signal Processing 66(3), 337–355 (1998)
6. Tachibana, R., Shimizu, S., Kobayashi, S.: An Audio Watermarking Method Using A Two-dimensional Pseudo-random Array. Signal Processing 82(10), 1455–1469 (2002)
7. Xiang, S.J., Huang, J.W., Yang, R., Wang, C.T., Liu, H.M.: Robust Audio Watermarking Based on Low-Order Zernike Moments. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 226–240. Springer, Heidelberg (2006)
8. Li, W., Xue, X.: Audio Watermarking Based on Music Content Analysis Robust Against Time Scale Modification. In: Kalker, T., Cox, I., Ro, Y.M. (eds.) IWDW 2003. LNCS, vol. 2939, pp. 289–300. Springer, Heidelberg (2004)
9. Xiang, S.J., Huang, J.W., Yang, R.: Time-Scale Invariant Audio Watermarking Based on the Statistical Features in Time Domain. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437. Springer, Heidelberg (2007)
10. Lin, Y.Q., Abdulla, W.H.: Robust Audio Watermarking Technique Based on Gammatone Filterbank and Coded-Image. In: International Symposium on Signal Processing and Its Application (ISSPA 2007) (2007)
11. Lin, Y.Q., Abdulla, W.H.: Robust Audio Watermarking for Copyrights Protection. Technical Report (No. 650), Department of Electrical & Computer Engineering, The University of Auckland (2006), `http://www.ece.auckland.ac.nz/~wabd002/Publications`

12. Kirovski, D., Malvar, H.S.: Spread-spectrum Watermarking of Audio Signals. IEEE Transactions on Signal Processing 51(4), 1020–1033 (2003)
13. Abdulla, W.H.: Auditory Based Feature Vectors for Speech Recognition Systems. In: Mastorakis, N.E., Kluev, V.V. (eds.) Advances in Communications and Software Technologies, pp. 231–236. WSEAS Press (2002)
14. Gurijala, A., Deller Jr., J.R.: Robust Algorithm for Watermark Recovery from Cropped Speech. In: IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP 2001), vol. 3, pp. 1357–1360 (2001)
15. ISO/IEC 11172-3: Information Technology - Coding of Moving Picture and Associated Audio for Digital Storage Media Up To About 1.5Mbit/s. British Standard. BSI London (1993)
16. Tachibana, R.: Improving Audio Watermarking Robustness Using Stretched Patterns Against Geometric Distortion. In: Chen, Y.-C., Chang, L.-W., Hsu, C.-T. (eds.) PCM 2002. LNCS, vol. 2532, pp. 647–654. Springer, Heidelberg (2002)
17. EBU: SQAM - Sound Quality Assessment Material, `http://sound.media.mit.edu/mpeg4/audio/sqam/`
18. Kabal, P.: An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality. TSP Lab Technical Report, Department of Electrical & Computer Engineering, McGill University (2003), `http://www.TSP.ECE.McGill.CA/MMSP/Documents`
19. Spanias, A.S.: Speech Coding: A Tutorial Review. Proceedings of the IEEE 82(10), 1541–1582 (1994)

A Comparison of DCT and DWT Block Based Watermarking on Medical Image Quality

Jason Dowling[1], Birgit M. Planitz[1], Anthony J. Maeder[1], Jiang Du[2], Binh Pham[2,4], Colin Boyd[3,4], Shaokang Chen[3], Andrew P. Bradley[3,4], and Stuart Crozier[3,4]

[1] e-Health Research Centre / CSIRO ICT Centre,
20/300 Adelaide St, Brisbane, QLD 4001, Australia
{jason.dowling,birgit.planitz,anthony.maeder}@csiro.au

[2] Faculty of Information Technology, Queensland University of Technology,
GPO Box 2434, Brisbane QLD 4001, Australia
{j.du,b.pham,c.boyd}@qut.edu.au

[3] School of Information Technology & Electrical Engineering,
University of Queensland, Brisbane QLD 4072, Australia
{shoakang,a.bradley,stuart}@itee.uq.edu.au

[4] Affiliated with National ICT Australia

Abstract. Hiding watermark information in medical image data files is one method of enhancing security and protecting patient privacy. However the research area of medical image watermarking has not been particularly active, partly due to concerns that any distortion could affect the diagnostic value of the medical image. These concerns can be addressed by ensuring that any image changes are kept below visual perception thresholds. In this paper the effects of image watermarking and common image manipulations are measured using the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Measure (SSIM) and Steerable Visual Difference Predictor (SVDP) numerical metrics. Two methods of block based watermarking are compared: the Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). To ensure a fair comparison a 128-pixel block size is used which allows an identical amount of information to be embedded for each method (3072 bits multiplied by embedding strength). The results suggest that although the two methods are similar, the DCT method is preferable if localization of changes is required. If localization is not required the DWT method is supported.

Keywords: digital image watermarking, information hiding, perceptual factors, human observers, medical image modalities.

1 Introduction

The objective of digital image watermarking is the insertion of a hidden message (or *payload*) within the body of an image. This message can be extracted by a receiver to prove ownership, identify if an image has been altered, and highlight the location of any alterations [3]. As medical images are increasingly captured,

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 454–466, 2008.

transmitted and stored in a digital format it is possible that an image could be altered for malevolent purposes (for example, insurance fraud). Software already exists to insert lesions imperceptibly into digital medical images [5]. One approach to solving this problem would be to use a digital image watermarking system where an imaging specialist could open a tampered image and receive a warning message that part of the image has been altered.

In our previous work watermarking methods which embed information using the Discrete Wavelet Transform (DWT) and the Discrete Cosine Transform (DCT) have been compared [4]. However, as the DCT used a block based approach and the DWT was applied to the entire image a different amount of information was embedded during the watermarking process. Therefore it remains difficult to evaluate the suitability of either DCT or DWT based on these results. The objective of the experiments presented in this paper are to examine the effects of DWT and DCT on image quality *where the amount of information embedded by each method is identical.*

This paper addresses the following hypotheses:

1. Is there a difference between a DWT and DCT watermarking approach when an identical amount of information is embedded by each method? As an identical amount of information (3072 bits multiplied by the embedding strength for each image block) was embedded into each image, it was expected that the results from each watermarking method would be similar.
2. Is the block based DWT more robust against JPEG2000 manipulation? The JPEG 2000 compression algorithm is based on wavelets, so it was expected that the DWT watermarking method would be more robust to this type of compression.
3. Similarly, does block based DCT perform better against JPEG manipulation? As the JPEG compression algorithm is based on quantization of DCT coefficients, it was expected that a watermarking method based on the DCT would be more robust than JPEG compression.

1.1 Method

Images. A total of 60 medical images were used, sourced from the University of Queensland (UQ), and the CSIRO ICT Centre BioMedIA Lab (BML). To investigate the effects of image modality Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) images were used. The size of these images was either 256x256 or 512x512 pixels. In most images the bit range of was [0,12], although some images with a range of [0,9], [0,10], and [0,12] were also included.

Visual quality metrics. Three numerical quality metrics, Peak Signal to Noise Ratio (PSNR) [6], Steerable Visual Difference Predictor (SVDP) , and the Structural Similarity Measure (SSIM) [9] were used to assess the amount of visual degradation caused by watermarking. These three metrics were chosen as they range from placing a low (PSNR) to high (SVDP) emphasis on the human visual system. Further descriptions of these metrics are available in [4].

Table 1. Block-based watermarking parameter selections for this paper

Parameter	Description	Values
Watermark method	Frequency domain embedding methods for each block	DCT or DWT
k	DCT/DWT embedding strength	1,2,10,20
b	Block size (bxb pixels)	128 x 128
p	Payload message length	32 bits

Payload. In this paper a 32-bit payload was redundantly embedded into each 128 pixel block within each image. This payload was generated from the DICOM header, which enables the detection of changes in individual image blocks or in the header itself (as a change in the header would result in a detectable consistent error across all image blocks). The SHA-256 algorithm [8] was used to generate a hash from all contents of the DICOM header and the right most 32-bits of this hash were used as the payload.

Watermarking method. In our previous work [4],[7], block-based code division multiplexing was used to embed information in the frequency domain and the resulting watermarked images were tested to assess differences in visual quality and robustness. As mentioned, the DCT and DWT embedding methods inserted different amounts of information into each image (the DWT method was applied to the entire image. Therefore in this paper a 128x128 pixel block based approach is used by both methods and an identical amount of information is embedded. The effect of each method (at various embedding strengths) on payload extraction and visual quality are then examined.

To increase the robustness of watermark insertion, the payload is multiplied by a global scaling factor (called the *embedding strength*) before insertion.

Table 1 presents the parameters used in this report. Note that DWT and DCT watermarking were tested separately, meaning that only one method was used to watermark an image at any one time.

The encoding and decoding procedures used by both frequency domain methods, DWT and DCT, are presented in Figure 1. In this paper b=128. Sample watermarked images are provided in Figures 5 - 7.

For the DWT method, a 2 level Haar wavelet transform was applied to each 128x128 pixel image block. The resulting Low/High (LH2), High/High (HH2), and (HL2) pass coefficients (shown in grey in Figure 2), consisted of 32 x 32 coefficients. The 32 bit payload then was embedded into each of these blocks by adding each 32 bit PN sequence (shown in Figure 3). Therefore for each 128 pixel block, the number of updated coefficients was 32 x 32 x 3 = 3072.

For the DCT method, the PN sequences for each payload bit were appended forming a 32x32 value vector. This sequence was repeated three times, generating a 32 x 32 x 3 = 3072 value vector. After each 128 x 128 pixel block in an image was transformed with the DCT, this vector was multiplied by the embedding strength (k) and then added to coefficients in the DCT domain. These updated coefficients are shown in Figure 4.

Encoding:

1. Divide original image into bxb pixel blocks
2. Generate payload of length 32 bits from image DICOM header
3. Generate watermark from payload using PN sequences
4. Compute DWT/DCT of each bxb image block
5. Embed watermark in each bxb image block
6. Compute IDWT/IDCT of each bxb image block

Decoding:

1. Divide watermarked image into bxb blocks
2. Generate PN sequences for '0' and '1' bits
3. Compute DWT/DCT of each bxb watermarked image block
4. Correlate pre-specified sections of DWT/DCT block with '0' and '1' PN sequences
5. Select bit ('0' or '1') where PN sequence corresponds to highest correlation as current bit value
6. Return extracted 32-bit payload message for each image block

Fig. 1. Method used to encode and decode the 32 bit payload

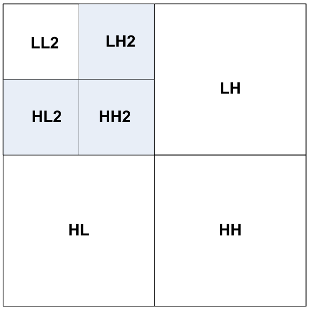
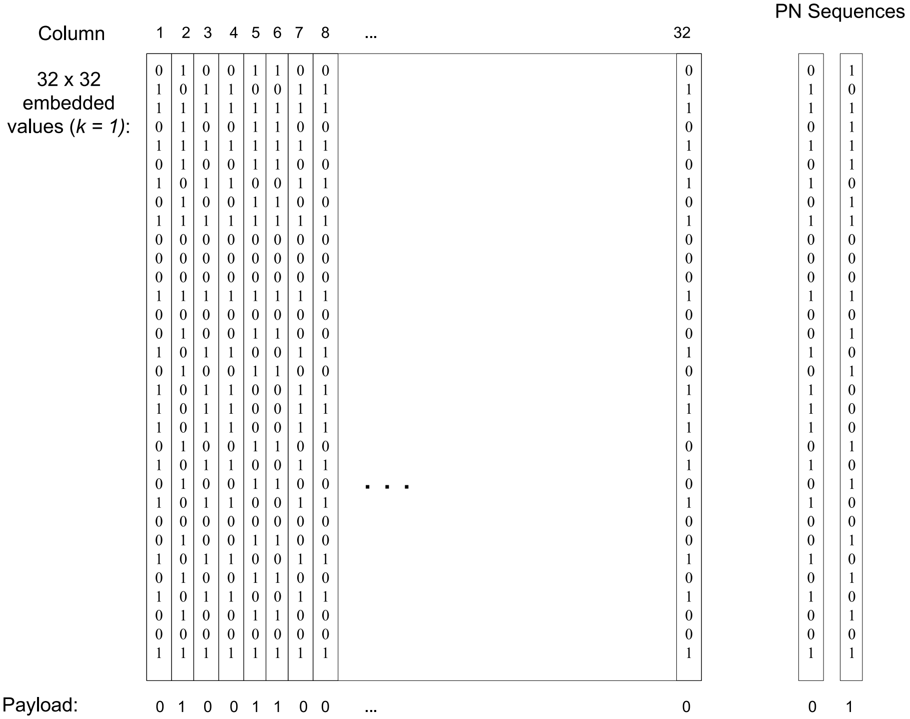


Fig. 2. DWT coefficients (shown in grey) updated in this paper. LH2, HL2 and HH2 each consist of 32x32 coefficients. The 32x32 watermark in Figure 3 is multiplied by an embedding strength (k) and then added to the coefficients in each of these blocks.

Procedure. Matlab v 7.2 scripts were used for all watermarking, manipulations and quality assessment. The software ran on a Dell Xeon 3.4 GHz PC (2.75 GB RAM) running Windows XP Professional SP 2. This research required two main steps:

1. Read each original image, apply the DCT watermark for each embedding strength (k), and save this watermarked image. Then repeat this step for the DWT method.

2. To evaluate the effect of image manipulation: Read each watermarked image, apply a required image manipulation (such as JPEG compression), and save the updated watermarked image. Repeat for all watermarked images, and all types of image manipulations.



Column	1	2	3	4	5	6	7	8	...	32	PN Sequences	
32 x 32 embedded values (*k = 1)*:	0	1	0	0	1	1	0	0		0	0	1
	1	0	1	1	0	0	1	1		1	1	0
	1	1	1	1	1	1	1	1		1	1	1
	0	1	0	0	1	1	0	0		0	0	1
	1	1	1	1	1	1	1	1		1	1	1
	0	1	0	0	1	1	0	0		0	0	1
	1	0	1	1	0	0	1	1		1	1	0
	0	1	0	0	1	1	0	0		0	0	1
	1	1	1	1	1	1	1	1		1	1	1
	0	0	0	0	0	0	0	0		0	0	0
	0	0	0	0	0	0	0	0		0	0	0
	0	0	0	0	0	0	0	0		0	0	0
	1	1	1	1	1	1	1	1		1	1	1
	0	0	0	0	0	0	0	0		0	0	0
	0	1	0	0	1	1	0	0		0	0	1
	1	0	1	1	0	0	1	1		1	1	0
	0	1	0	0	1	1	0	0		0	0	1
	1	0	1	1	0	0	1	1		1	1	0
	1	0	1	1	0	0	1	1		1	1	0
	1	0	1	1	0	0	1	1		1	1	0
	0	1	0	0	1	1	0	0		0	0	1
	1	0	1	1	0	0	1	1		1	1	0
	0	1	0	0	1	1	0	0	. . .	0	0	1
	1	0	1	1	0	0	1	1		1	1	0
	0	0	0	0	0	0	0	0		0	0	0
	0	1	0	0	1	1	0	0		0	0	1
	1	0	1	1	0	0	1	1		1	1	0
	0	1	0	0	1	1	0	0		0	0	1
	1	0	1	1	0	0	1	1		1	1	0
	0	1	0	0	1	1	0	0		0	0	1
	0	0	0	0	0	0	0	0		0	0	0
	1	1	1	1	1	1	1	1		1	1	1
Payload:	0	1	0	0	1	1	0	0	...	0	0	1

Fig. 3. DWT watermark construction. Each payload bit is allocated one of two pseudo random number (PN) sequences (examples are shown in the right-most columns). These sequences are placed in columns and form a 32x32 matrix which is then multiplied by an embedding strength (k) and applied to each 32x32 DWT coefficient block shown in Figure 2.

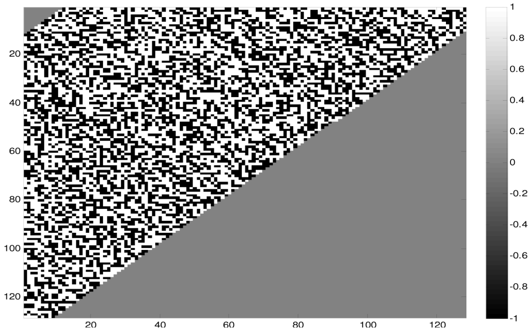


Fig. 4. DCT domain coefficients updated by the watermarking method in this paper

Four different levels of embedding strength (k = 1, 2, 10, 20) were used to evaluate embedding strength differences between the DWT and DCT methods on image quality and watermark.

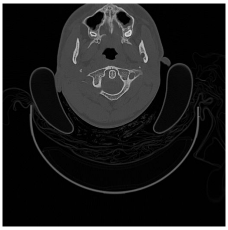


Fig. 5. Original 512 x 512 pixel CT Head image

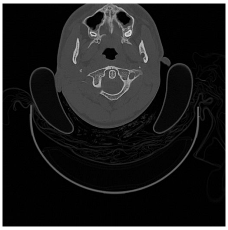


Fig. 6. Original image watermarked with DWT method (k=20)

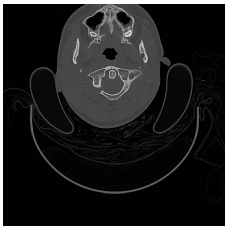


Fig. 7. Original image watermarked with DCT method(k=20)

2 Results

2.1 Watermarking

Four dependent variables were measured for each image. A summary of the mean and (standard deviation) BER and associated visual quality (PSNR, SSIM and SVDP) results are presented in Tables 2 - Table 5. The Bit Error Ratio (BER) is a commonly used measure of watermark extraction errors. BER represents the ratio of bits incorrectly extracted to the total number of bits extracted [3]. As the embedding strength of watermarking algorithm is increased, there will usually be a corresponding decrease in the BER when the watermark is extracted.

These results show that the DWT block-based method resulted in consistently lower BER results, and better visual quality results than the DCT at all embedding strength (k) levels.

The results indicate that a 128 pixel block-based method is only suitable for CT images. In order to obtain an acceptable BER (< 0.10), the visual quality degradation for both the UQ and Biomedia Lab MR images is unacceptable (PSNR < 45 dB).

Table 2. Mean BER (with *standard deviation*) and visual quality results for UQ Head MRI

	BER		**PSNR**		**SSIM**		**SVDP**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	0.16	0.34	50.95	46.92	1.00	1.00	0.07	0.59
	(0.04)	*(0.04)*	*(3.05)*	*(3.02)*	*(0.00)*	*(0.00)*	*(0.00)*	*(0.06)*
2	0.08	0.21	44.79	41.62	1.00	1.00	0.24	0.95
	(0.02)	*(0.04)*	*(3.07)*	*(2.96)*	*(0.00)*	*(0.00)*	*(0.02)*	*(0.03)*
10	0.00	0.02	32.76	28.63	0.99	0.96	1.00	1.00
	(0.00)	*(0.01)*	*(2.94)*	*(2.88)*	*(0.01)*	*(0.02)*	*(0.00)*	*(0.00)*
20	0.00	0.00	27.35	22.85	0.95	0.87	1.00	1.00
	(0.00)	*(0.00)*	*(2.89)*	*(2.88)*	*(0.02)*	*(0.06)*	*(0.00)*	*(0.00)*

Table 3. BER and visual quality results for UQ Head CT images

	BER		**PSNR**		**SSIM**		**SVDP**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	0.40	0.44	75.64	71.54	1.00	1.00	0.07	0.56
	(0.05)	*(0.04)*	*(0.80)*	*(0.81)*	*(0.00)*	*(0.00)*	*(0.01)*	*(0.10)*
2	0.34	0.40	69.56	65.96	1.00	1.00	0.26	0.95
	(0.07)	*(0.06)*	*(0.81)*	*(0.81)*	*(0.00)*	*(0.00)*	*(0.05)*	*(0.03)*
10	0.15	0.25	56.86	52.25	0.99	0.98	1.00	1.00
	(0.06)	*(0.08)*	*(0.81)*	*(0.82)*	*(0.01)*	*(0.02)*	*(0.00)*	*(0.00)*
20	0.07	0.17	50.99	46.40	0.98	0.96	1.00	1.00
	(0.05)	*(0.07)*	*(0.82)*	*(0.83)*	*(0.03)*	*(0.04)*	*(0.00)*	*(0.00)*

Table 4. BER and visual quality results for Biomedia Body CT images

	BER		**PSNR**		**SSIM**		**SVDP**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	0.31	0.37	74.96	70.87	1.00	1.00	0.07	0.60
	(0.08)	*(0.07)*	*(1.25)*	*(1.25)*	*(0.00)*	*(0.00)*	*(0.01)*	*(0.06)*
2	0.24	0.31	68.58	65.28	1.00	1.00	0.24	0.96
	(0.08)	*(0.07)*	*(1.24)*	*(1.25)*	*(0.00)*	*(0.00)*	*(0.03)*	*(0.02)*
10	0.00	0.02	32.76	28.63	0.99	0.96	1.00	1.00
	(0.03)	*(0.05)*	*(1.25)*	*(1.25)*	*(0.02)*	*(0.04)*	*(0.00)*	*(0.00)*
20	0.05	0.14	50.26	45.78	0.91	0.83	1.00	1.00
	(0.02)	*(0.03)*	*(1.25)*	*(1.26)*	*(0.06)*	*(0.09)*	*(0.00)*	*(0.00)*

Table 5. BER and visual quality results for Biomedia Head MR images

	BER		**PSNR**		**SSIM**		**SVDP**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	0.25	0.43	59.77	55.69	1.00	1.00	0.07	0.53
	(0.07)	*(0.05)*	*(7.22)*	*(7.21)*	*(0.00)*	*(0.00)*	*(0.01)*	*(0.09)*
2	0.18	0.35	53.35	50.13	1.00	1.00	0.24	0.93
	(0.08)	*(0.06)*	*(7.23)*	*(7.20)*	*(0.00)*	*(0.00)*	*(0.03)*	*(0.04)*
10	0.03	0.10	40.99	36.70	1.00	0.99	1.00	1.00
	(0.05)	*(0.08)*	*(7.14)*	*(6.98)*	*(0.00)*	*(0.01)*	*(0.00)*	*(0.00)*
20	0.01	0.04	35.44	30.94	0.98	0.96	1.00	1.00
	(0.02)	*(0.05)*	*(6.98)*	*(6.91)*	*(0.01)*	*(0.03)*	*(0.00)*	*(0.00)*

2.2 Image Manipulation

In order to evaluate the robustness of the embedded watermark against image changes, four different types of image manipulation were applied to the DCT (n=60) and DWT (n=60) watermarked images from the previous step: *edge enhancement*, *histogram stretching*, *JPEG* and *JPEG 2000* compression. Each of these manipulations were applied at three different levels (*low*, *medium* and

Table 6. Results for different levels of image degradation caused by edge enhancement and embedding strength (all image types combined)

	BER							
	None		**Low**		**Medium**		**High**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	0.28	0.40	0.29	0.40	0.37	0.45	0.41	0.47
	(0.06)	*(0.05)*	*(0.06)*	*(0.05)*	*(0.04)*	*(0.04)*	*(0.01)*	*(0.03)*
2	0.21	0.32	0.21	0.32	0.29	0.40	0.34	0.43
	(0.06)	*(0.06)*	*(0.06)*	*(0.06)*	*(0.06)*	*(0.05)*	*(0.04)*	*(0.04)*
10	0.07	0.14	0.07	0.14	0.14	0.23	0.19	0.29
	(0.04)	*(0.06)*	*(0.04)*	*(0.06)*	*(0.05)*	*(0.06)*	*(0.06)*	*(0.06)*
20	0.03	0.09	0.03	0.09	0.09	0.16	0.14	0.22
	(0.02)	*(0.04)*	*(0.02)*	*(0.04)*	*(0.04)*	*(0.06)*	*(0.05)*	*(0.06)*

Table 7. Ratio of BER change for different levels of image degradation caused by edge enhancement and embedding strength *k*

	BER							
	None		**Low**		**Medium**		**High**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	1.00	1.00	1.02	1.00	1.32	1.13	1.46	1.18
2	1.00	1.00	1.00	1.00	1.39	1.26	1.62	1.35
10	1.00	1.00	1.00	1.00	1.96	1.58	2.75	2.02
20	1.00	1.00	1.00	1.00	2.62	1.77	4.15	2.46

high) and the resulting BER was recorded. To separate the effects of the original watermarking method and the manipulation, both the BER results and their rate of change from the non-manipulated watermarked image are reported.

Edge enhancement. Three levels of Sobel edge detection (*Low*: $\alpha = 0.1$; *Medium*: $\alpha = 0.5$; *High*: $\alpha = 0.9$) were used to degrade each watermarked image. The ratio of BER before and after manipulation (Table 7) found that for all image types the DCT watermarked images were slightly more robust to edge enhancement changes. For all image types, as embedding strength *k* increased, the amount of degradation from edge enhancement also generally increased for all image types.

Histogram stretching. Three levels of histogram stretching using the window/level method were applied with different thresholds (*Low*: = background mean; *Medium*: background mean+signal variation; *High*: background mean + 2 x signal variation). The ratio of BER after / BER before manipulation (Table 9) shows that for all image types the DCT watermarked images were more robust to histogram stretching changes. As with the results for edge enhancement manipulation, as the original watermarked embedding strength *k* increased, the amount of degradation from edge enhancement also generally increased for all image types.

Table 8. Different levels of image degradation caused by histogram stretching and different embedding strength levels (all image types combined)

	BER							
	None		**Low**		**Medium**		**High**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	0.27 *(0.12)*	0.40 *(0.13)*	0.41 *(0.15)*	0.45 *(0.14)*	0.48 *(0.16)*	0.46 *(0.14)*	0.49 *(0.17)*	0.46 *(0.15)*
2	0.17 *(0.10)*	0.26 *(0.12)*	0.27 *(0.15)*	0.31 *(0.13)*	0.33 *(0.16)*	0.32 *(0.15)*	0.34 *(0.18)*	0.32 *(0.15)*
10	0.07 *(0.05)*	0.11 *(0.09)*	0.16 *(0.11)*	0.15 *(0.12)*	0.24 *(0.17)*	0.18 *(0.16)*	0.26 *(0.18)*	0.19 *(0.18)*
20	0.03 *(0.03)*	0.07 *(0.07)*	0.10 *(0.09)*	0.10 *(0.10)*	0.20 *(0.16)*	0.12 *(0.17)*	0.24 *(0.17)*	0.14 *(0.17)*

Table 9. Ratio of BER change for different levels of histogram stretching and embedding strength *k*

	BER							
	None		**Low**		**Medium**		**High**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	1.00	1.00	1.52	1.12	1.78	1.14	1.80	1.15
2	1.00	1.00	1.57	1.21	1.96	1.24	1.99	1.25
10	1.00	1.00	2.38	1.39	3.65	1.64	3.96	1.75
20	1.00	1.00	3.45	1.58	7.27	1.88	8.64	2.12

Table 10. Results for different levels of image degradation caused by JPEG compression and embedding strength levels (all image types combined)

	BER							
	None		**Low**		**Medium**		**High**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	0.27 *(0.12)*	0.40 *(0.13)*	0.27 *(0.12)*	0.40 *(0.13)*	0.38 *(0.14)*	0.42 *(0.13)*	0.43 *(0.14)*	0.44 *(0.14)*
2	0.17 *(0.10)*	0.26 *(0.12)*	0.17 *(0.10)*	0.26 *(0.12)*	0.23 *(0.12)*	0.27 *(0.12)*	0.28 *(0.13)*	0.29 *(0.13)*
10	0.07 *(0.05)*	0.11 *(0.09)*	0.07 *(0.05)*	0.11 *(0.09)*	0.07 *(0.05)*	0.11 *(0.09)*	0.07 *(0.06)*	0.11 *(0.09)*
20	0.03 *(0.03)*	0.07 *(0.07)*	0.03 *(0.03)*	0.07 *(0.07)*	0.03 *(0.03)*	0.07 *(0.17)*	0.04 *(0.03)*	0.07 *(0.07)*

JPEG compression. For this manipulation each watermarked image was compressed with the following JPEG quality factors: 100 (*low degradation*); 75 (*medium degradation*); and 50 (*high degradation*) and then saved as a DICOM image. Table 11 shows that for all image types the DCT watermarked images

Table 11. Ratio of BER change for different levels of JPEG compression and embedding strength k

	BER							
	None		**Low**		**Medium**		**High**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	1.00	1.00	1.00	1.00	1.42	1.05	1.58	1.09
2	1.00	1.00	1.00	1.00	1.32	1.06	1.62	1.13
10	1.00	1.00	1.00	0.98	1.08	1.00	1.12	0.95
20	1.00	1.00	1.00	1.00	1.09	1.04	1.27	1.04

Table 12. Results for different levels of image degradation caused by JPEG 2000 compression for different embedding strength levels (all image types combined)

	BER							
	None		**Low**		**Medium**		**High**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	0.27 *(0.12)*	0.40 *(0.13)*	0.45 *(0.15)*	0.47 *(0.14)*	0.45 *(0.15)*	0.47 *(0.15)*	0.48 *(0.16)*	0.49 *(0.15)*
2	0.17 *(0.10)*	0.26 *(0.12)*	0.30 *(0.14)*	0.33 *(0.14)*	0.31 *(0.14)*	0.33 *(0.15)*	0.36 *(0.15)*	0.36 *(0.15)*
10	0.07 *(0.05)*	0.11 *(0.09)*	0.11 *(0.08)*	0.13 *(0.11)*	0.11 *(0.08)*	0.14 *(0.11)*	0.27 *(0.16)*	0.28 *(0.15)*
20	0.03 *(0.03)*	0.07 *(0.07)*	0.05 *(0.04)*	0.07 *(0.07)*	0.05 *(0.04)*	0.07 *(0.07)*	0.21 *(0.15)*	0.20 *(0.15)*

were slightly more robust to JPEG compression changes. Compared to the other three manipulation methods, JPEG compression resulted in the lowest level of increased BER. Unlike the edge enhanced and histogram stretched watermarked images, JPEG did not increase the number of extraction errors for higher embedding strengths.

JPEG 2000 compression. The open source JPEG 2000 utility Jasper [1] was called from Matlab to compress each watermarked image. The following compression ratios were used for these experiments: 1.0 (*low*), 0.1 (*medium*); and 0.01 (*high* degradation). Each watermarked image was opened, converted to JPEG2000 and saved, then re-opened, converted to DICOM format and saved. The combined results for all image types are shown in Table 12. The low and medium levels of degradation for JPEG 2000 resulted in the same images. The reason for this is unclear; however in this case Jasper did not appear to differentiate between a compression ratio of 1.0 and 0.1. The ratio of BER after / BER before manipulation (Table 13) shows that for all image types the DCT watermarked images were slightly more robust to JPEG 2000 compression changes. Unlike JPEG, the number of extraction errors for JPEG2000 manipulated images increased as the embedding strength k increased.

Table 13. Ratio of BER change for different levels of JPEG 2000 compression and embedding strength k

	BER							
	None		**Low**		**Medium**		**High**	
k	DWT	DCT	DWT	DCT	DWT	DCT	DWT	DCT
1	1.00	1.00	1.65	1.16	1.65	1.16	1.78	1.22
2	1.00	1.00	1.78	1.28	1.79	1.28	2.10	1.40
10	1.00	1.00	1.62	1.20	1.65	1.23	4.08	2.50
20	1.00	1.00	1.73	1.08	1.82	1.08	7.45	3.12

3 Discussion

This paper has compared the effects of embedding the same size watermark using two different watermarking methods (DWT and DCT). A block based approach was applied, by dividing each image into 128x128 pixel blocks and then applying the watermark to that block. Although it performed poorly for MR images, the DWT method resulted in fewer extraction errors and less degradation to image visual quality.

There appear to be advantages to using either the DWT or DCT method for watermarking. In this experiment using the 128 pixel block-based method, acceptable results were only obtained for CT images. In order to obtain an acceptable level of extraction errors (BER < 0.10), the visual quality degradation for both Biomedia Lab and UQ MR images is unacceptable (PSNR < 45 dB).

For all images, as the manipulation level increased (from low to high), the values for BER increased. For all manipulation types apart from JPEG, increased embedding strength was associated with higher rates of extraction errors after manipulation. JPEG compression was also found to cause the least number of extraction errors.

This paper aimed to investigate the following three hypotheses:

1. *What is the effect of having the same amount of information embedded by each method?* As an identical amount of information (3072 bits multiplied by the embedding strength for each image block) was embedded into each image, it was expected that the results from each watermarking method would be similar. However results from this paper have shown that the DWT block-based at this block size method resulted in consistently lower BER results, and better visual quality results than the DCT at all embedding strength (k) levels.
2. *Is the block based DWT more robust against JPEG2000 manipulation?* As the JPEG 2000 compression algorithm is wavelet-based, it was expected that the DWT watermarking method would be more robust to this type of compression. However we found that there were more watermark extraction errors from the DWT watermarked images compared to DCT after JPEG 2000 compression.

3. *Does block based DCT perform better against JPEG manipulation?* Similarly, as the JPEG compression algorithm is based on quantization of DCT coefficients, it was expected that a watermarking method based on the DCT would be more robust JPEG compression. The DCT watermarked images were found to have less watermark extraction errors after JPEG compression than the DWT watermarked images.

These results indicate that the DWT is superior both in extraction errors and visual quality results to the DCT method *when a 128 pixel block size is used.* However, the DCT method is more effective at a block size of 64x64 or smaller [4]. In addition, the DWT method is more effective when a block-based approach is not followed (i.e. the entire image is watermarked), although an entire image approach does not allow for accurate location of image changes. Therefore, if localization of changes is required, the DCT method with a 64x64 pixel (or smaller) block size appears to be more suitable for medical image watermarking. However if localization is not required, the DWT method applied to the entire image is probably superior.

One constraint of this paper is that we have only considered the effects of image manipulation on watermark extraction errors (BER). Future work could consider the impact of image manipulation on visual quality (as measured by PNSR, SSIM and SVDP).

Acknowledgments. National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs and the Queensland Government. The authors would like to acknowledge Dr Sebastien Ourselin (ICT Centre, CSIRO) and Dr Andrew P. Bradley (University of Queensland) for providing medical images to test the watermarking systems.

References

1. Adams, M.D., Kossentini, F.: JasPer: a software-based JPEG-2000 codec implementation. In: International Conference on Image Processing, vol. 2, pp. 53–56 (2000)
2. Coatrieux, G., Main, H., Sankur, B., Rolland, Y., Collorec, R.: Relevance of Watermarking in Medical Imaging. IEEE-EMBS Information Technology Applications in Biomedicine, 250-255 (2000)
3. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, San Francisco (2002)
4. Dowling, J., Planitz, B., Maeder, A., Du, J., Pham, B., Boyd, C., Chen, S., Bradley, A., Crozier, S.: In: SPIE Conference on Medical Imaging, San Diego, vol. 6515, pp. 65151L1–12 (2007)
5. Madsen, M.T., Berbaum, K.S., Ellingson, A., Thompson, B.H., Mullan, B.F.: Lesion removal and lesion addition algorithms in lung volumetric data sets for perception studies. In: SPIE Conference on Medical Imaging, San Diego, vol. 6146, pp. 61460T-1–10 (2006)

6. Petitcolas, F.A.: Watermarking schemes evaluation. IEEE. Signal Processing 17(5), 58–64 (2000)
7. Planitz, B., Maeder, A.: In: Perceptually-limited modality-adaptive medical image watermarking SPIE Conference on Medical Imaging, San Diego, vol. 6146, pp. 61460V-1 - 10 (2006)
8. Stallings, W.: Cryptography and Network Security, 4th edn. Prentice Hall, New Jersey (2006)
9. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)

A High Payload VQ Steganographic Method for Binary Images

Chin-Chen Chang[1,2], Chih-Yang Lin[2], and Yu-Zheng Wang[2]

[1] Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan, R.O.C.
ccc@cs.ccu.edu.tw

[2] Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi 62102, Taiwan, R.O.C.
{gary,wyc92}@cs.ccu.edu.tw

Abstract. In this paper, we propose a new VQ steganographic method for embedding binary images and improving the stego-image quality. The main idea is using the new S-tree to represent the binary image and applying the genetic k-means clustering technique on the codebook to obtain strong cohesion clusters in order to reduce the replacement distortion. Experimental results show that our method outperforms the existing schemes on both image quality and embedding capacity.

Keywords: Steganography, data hiding, clustering, genetic algorithm.

1 Introduction

Image steganography is the art of conveying messages in a secret way so that only the receiver can decrypt the message. Unlike traditional cryptographic techniques that clutter the data, steganography is a method that embeds secret data in another medium in a way that the transmitted data seems meaningful and ordinary, drawing no attention from attackers.

Since steganography is to avoid the secret data hidden in the stego-image from being revealed without proper authorization, two main issues should be addressed: payload and resistance. The payload issue deals with the capacity of the cover image to hold secret data, and by the term resistance we mean the ability of the scheme to work against image manipulations such as blurring or sharpening. However, these two requirements come in a tradeoff situation. Conventionally, the former problem is referred to as *data hiding* and the latter is known as *watermarking* [1,13]. In this paper, however, we only focus on the former problem.

To do steganography, the simplest way is probably the least-significant bit (LSB) insertion method [3,19], which embeds secret data in the least-significant bits of the stego-image and usually employs a pseudo-random number to clutter the embedding order to achieve security [6,13]. This method changes the pixel values slightly, resulting in only small intensity differences.

Y.Q. Shi, H.-J. Kim, and S. Katzenbeisser (Eds.): IWDW 2007, LNCS 5041, pp. 467–481, 2008.

Although the LSB method is simple, it is, however, not appropriate to apply it to VQ-based steganography directly. Basically, VQ (Vector Quantization) is a lossy compression method [5,9,10] that uses a codebook with codewords in it to represent an image. The VQ compression process starts with partitioning an image into non-overlapping blocks, and then it maps each block to the closest codeword of the codebook. These blocks are finally represented by the indices of the codewords. This way, once the LSB method is introduced to the VQ system, the quality of the stego-image may become worse because the difference between two adjacent indices in the codebook may be large.

In recent years, many methods have been proposed to hide secret data in the VQ system. In 1999, Lin et al. proposed an LSB-like method for VQ [14], where one secret bit is embedded in one image block represented by a codeword index. They partitioned a codebook into two equally-sized sub-codebooks. The sub-codebooks are rearranged by the Pairwise Nearest Clustering Embedding (*PNCE*) method [14] such that all pairs of codewords between the sub-codebooks are as similar as possible and are located at the same positions. After that, one sub-codebook is responsible for the odd indices and the other is responsible for the even indices. For example, suppose a secret data bit is 0, the corresponding index is changed to even by referencing the alternative sub-codebook if the original index is odd. On the other hand, if the secret data bit is 1, the embedding process is performed in the opposite way. This way, Lin et al.'s method can embed the secret data in the least significant bits of the codeword indices. In order to embed k bits by performing LSB modification, Lu and Sun [16] tried to divide the codebook into even more sub-codebooks than just two. In 2002, Jo and Kim [11] solved the image quality problem caused by the *PNCE* method. However, these LSB-based methods are still confronted with the limitation of low embedding capacity.

In 2003, Du and Hsu [7] proposed a high capacity embedding method based on VQ in a non-LSB-related way. The method uses hierarchical clustering to partition the codebook into several clusters and transforms the secret data to a decimal number that is in fact an ordered list that contains a set of codewords assigned to each input block. However, the image quality produced by this method is not stably desirable because it heavily depends on the clustering result. A larger embedding capacity coming from larger clusters would significantly degrade the image quality. Furthermore, this method is not suitable for low entropy images such as binary images, since such materials contain much redundancy.

In this paper, we shall improve Du and Hsu's method to make it more effective in embedding greater quantities of data in VQ images while keeping a high image quality. To extend the functionality of Du and Hsu's method, we shall concentrate on the embedding of binary images. Instead of using the hierarchical clustering technique as Du and Hsu's method does, we shall modify the *VGA-clustering* (variable string length genetic algorithm) method [2] a little for better clustering results. In addition, we shall also construct the new S-tree structure to represent the binary image so as to improve the embedding capacity. Extensive experimental results show that our method is superior to Du and Hsu's method in terms of image quality and embedding capacity, especially when it comes to binary images.

The rest of this paper is organized as follows. First, we shall briefly review VQ and Du and Hsu's method in Section 2. Then our proposed scheme would be presented in detail in Section 3, followed by the extensive experimental results in Section 4 that demonstrate the effectiveness and efficiency of our new scheme. Finally, some concluding remarks will be stated in Section 5.

2 Related Works

2.1 Vector Quantization

Vector Quantization (VQ) [10] is one of the most popular compression techniques well-accepted due to its simple, efficient and effective encoding and decoding procedures. Figure 1 shows the VQ encoding and decoding processes.

To begin with, the original image is partitioned into non-overlapping blocks with $l \times r$ pixels each, so that each block can be represented by an lr-dimensional vector. In brief, VQ can be defined as a mapping function Q from lr-dimensional Euclidean space R^{lr} to a finite subset $CB = \{y_1, y_2, \ldots, y_n\}$; that is, Q: $R^{lr} \rightarrow CB$, where CB is generally called a codebook, and y_i is the i-th codeword in CB. The codebook used in VQ is usually generated from a number of training codewords.

In the encoding procedure, each vector $x = \{e_1, e_2, \ldots, e_{lr}\} \in R^{lr}$ of the original image is compared with the codewords in the codebook $CB = \{y_1, y_2, \ldots, y_n\}$ to find the best codeword $y_i = \{y_{i1}, y_{i2}, \ldots, y_{i(lr)}\}$ in the codebook. The best codeword for x is determined by the Euclidean distance $d(x, y_i)$ as presented in Equation (1), where e_j and $y_{i,j}$ are the j-th elements of codewords x and y_i, respectively.

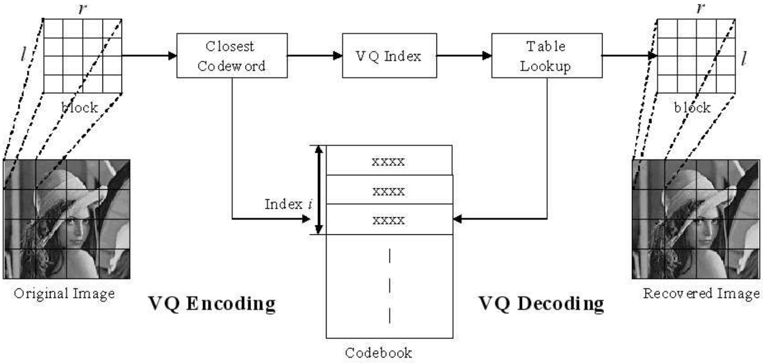


Fig. 1. VQ encoding/decoding procedure

$$d(x, y_i) = \| x - y_i \| = \left[\sum_{j=1}^{lr} (e_j - y_{ij})^2 \right]^{\frac{1}{2}} . \tag{1}$$

When the best codeword y_i of x is found, the index i of y_i is used to encode the vector x. Therefore, the original image can finally be represented by the indices of these closest codewords.

In the decoding procedure, the decoder has the same codebook as the encoder does. For each index i, by means of a simple table lookup operation, the decoder can obtain y_i and therefore can reconstruct the input codeword x. The quality of the VQ compressed image heavily depends on the codebook size and the selection of codewords.

2.2 Du and Hsu's Embedding Method

Assume that we have a cover VQ image $I = \{x_1, x_2, \ldots, x_m\}$, a codebook $CB = \{y_1, y_2, \ldots, y_n\}$, and a secret bit stream S, where the bit stream S can be transformed to an unsigned integer. Firstly, the codebook CB is grouped into k clusters $\{C_1, C_2, \ldots, C_k\}$, where $C_1 \cup C_2 \cup \ldots \cup C_k = CB$ and $C_i \cap C_j = \emptyset, \forall i \neq j$. Then, the product number P is calculated according to Equation (2):

$$P = \prod_{i=1}^{m} |C(x_i)|. \tag{2}$$

Here, $|C(x_i)|$ is the size of the cluster which the index x_i of I belongs to. If the product number P is smaller than the secret data S, then two clusters of CB will be merged by using the nearest-neighbor rule, resulting in fewer clusters, and the product number P will be recalculated. The merging operation is repeated until the product number P is greater than or equal to the secret data S. Finally, the secret data is embedded in the ordered list formed by combining the elements from the clusters that each codeword of I belongs to.

An example of how Du and Hsu's embedding method works is shown in Figure 2. Assume we have the cover image $I = \{x_1, x_2, x_3, x_4\}$ shown in Figure 2(a) to hide the secret data $(300)_{10}$ in, and the codebook sized 16 is

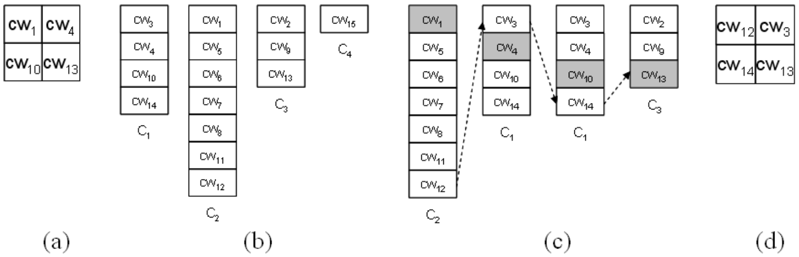


Fig. 2. An example of Du and Hsu's embedding method

grouped into four clusters as Figure 2(b) shows. This way, we come to the conclusion that the product number P is $7 \times 4 \times 4 \times 3 = 336$ according to Equation (2). Since the product number P $(336)_{10}$ is greater than the secret data S $(300)_{10}$, the secret data can be embedded in I. The clusters, C_2, C_1, C_1, C_3, which the codewords of I belong to, are selected in proper order. Then, the 300^{th}-combination, $(cw_{12}, cw_3, cw_{14}, cw_{13})$, of the selected clusters shown in Figure 2(c) is used to represent the secret data. After the embedding procedure, the stego-image, shown in Figure 2(d), turns out in the form of an ordered list of these combined elements.

3 The Proposed Hiding Scheme on VQ

The proposed method firstly clusters the codebook by using a genetic algorithm. Then the new S-tree structure is employed to represent the binary image before the embedding scheme is applied.

3.1 Preprocessing of Codebook

The first step we take here is to cluster the codewords from the codebook. The purpose of this preprocessing procedure is to help efficiently find a similar codeword in the same cluster for later use in the searching procedure. The clustering result has a strong impact on the stego-image quality and the embedding capacity. Therefore, we modify the *VGA-clustering* (variable string length genetic algorithm) method [2,18] based on a genetic k-means algorithm to obtain optimal clustering results. There are three main advantages to this method: First, k-means method discovers clusters spherically with better cohesion within each individual cluster, which is superior to having clusters in arbitrary shapes. Second, *VGA-clustering* helps the k-means method toward the optimal solution without giving the number of clusters to be grouped. Third, the clustering results by the *VGA-clustering* algorithm are irrespective of the starting configuration which is the choice of the initial cluster center.

The basic operations genetic algorithms execute are selection, crossover, and mutation, which are shown in Fig. 3. Here, let's have a brief look at the *VGA-clustering* technique.

Before a genetic operation can properly work, the chromosome and the fitness function should be defined first [8,17]. In *VGA-clustering*, each chromosome is composed of a sequence of real numbers representing the centers of the clusters. For example, a chromosome (23, 42, 31, 28, 45, 33) in the 2-dimensional space reveals the fact that there are three cluster centers (23, 42), (31, 28), and (45, 33). Assume there are k clusters $C_1, C_2, \ldots, C_k$ with the centers being $z_1, z_2, \ldots, z_k$ selected from n input points $y_1, y_2, \ldots, y_n$. The fitness function f, namely the cluster validity equation, of *VGA-clustering* is defined in Equation (3), and the chromosome with the maximum fitness value is the best clustering result when the genetic operations are finished.

$$f = \left(\frac{1}{k} \times \frac{E_1}{E_k} \times D_k\right)^p. \qquad (3)$$

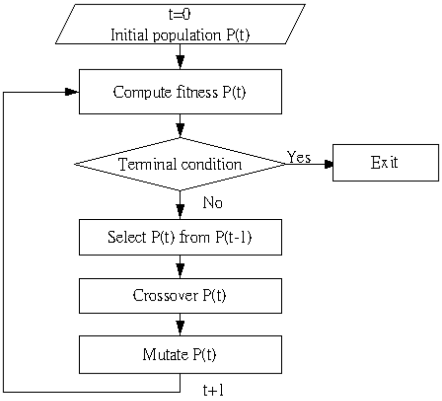


Fig. 3. Basic operations of genetic algorithms

Here, k is the number of clusters, and p is any real number greater than or equal to 1. The values E_k and D_k are defined as follows:

$$E_k = \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij}\|y_j - z_i\|, \text{ and} \tag{4}$$

$$D_k = \max_{i,j=1}^{k} \|z_i - z_j\|. \tag{5}$$

In the equations above, n stands for the total number of points in the data set, z_i is the center of the i-th cluster C_i, and u_{ij} is the membership of pattern y_j to cluster C_i. If y_j belongs to C_i, then $u_{ij} = 1$; otherwise, $u_{ij} = 0$.

In order to improve the embedding capacity, the fitness function defined in Equation (3) is changed to Equation (6), where R_k is the weight summation defined in Equation (7), in which $w(y_j)$ returns the weight of y_j and $|C_i|$ is the size of C_i. The weight of y_j is determined by its frequency in the given image. The greater the frequency is, the greater the weight will be.

$$f = \left(\frac{1}{k} \times \frac{E_1}{E_k} \times D_k \times R_k\right)^p. \tag{6}$$

$$R_k = \sum_{i=1}^{k}\sum_{j=1}^{n} w(y_j)u_{ij}|C_i|. \tag{7}$$

Suppose *VGA-clustering* starts with the initial population of p chromosomes. The length of each chromosome is non-fixed during the genetic operations. The selection operation selects p chromosomes from the mating pool by using the roulette-wheel method [17], where the higher the fitness value is, the greater the probability for the chromosome to be selected will be. The crossover operation exchanges information between two parent chromosomes at crossover rate

μ_c to generate two descent chromosomes, or offspring. Each chromosome preserves at least two clusters. In addition, the mutation operation changes a cell's value of a chromosome at mutation rate μ_m within a fixed range as Equation (8) defines below.

$$\begin{cases} v \pm 2 \times \delta \times v, & v \neq 0, \\ \pm 2 \times \delta, & v = 0. \end{cases} \tag{8}$$

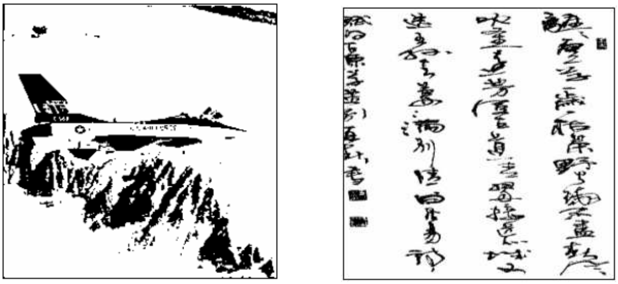
In the equation, v is one of the center values in a chromosome, and δ is in the rang [0, 1] generated with uniform distribution.

The three genetic operations are executed iteratively until the maximum fitness value hardly changes. After termination, the codewords of the codebook are clustered according to the chromosome with the maximum fitness value.

3.2 Binary Image Representation

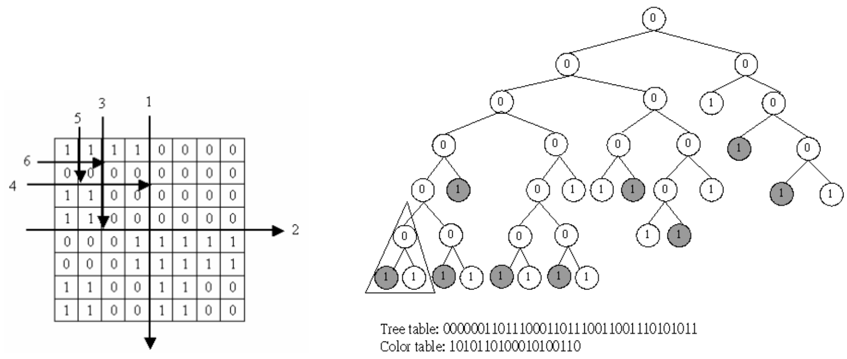
The binary image apparently has local characteristics, which means there are repeating bits in local areas as shown in Figure 4. In Du and Hsu's method, however, the local characteristics of the watermark are not taken into account, and therefore the embedding capacity is hardly improved. In order to embed a large binary image in a VQ image, we use the *new S-tree* structure [4] to represent the binary image. The new S-tree is an improved version of the S-tree [12], which has originally been used to represent calligraphic images. The process of building the new S-tree for a binary image is built upon the breadth-first search (BFS) concept as follows.

The first step of constructing a new S-tree is to build an S-tree in advance. Initially, the S-tree contains one node, and then the image is recursively divided into two equal-sized subimages until all subimages become homogeneous. A block (subimage) is called homogeneous if the pixels of the block are all 0's or 1's. At each division step, the partition is alternated between horizontal direction and vertical direction, and the left child node represents the left (or upper) subimage while the right child node represents the right (or lower) subimage. If a subimage



(a) Binary F16 (b) Chinese calligraphy

Fig. 4. Binary images



1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	0

Tree table: 0000001101110001101110011001110101011
Color table: 1010110100010100110

(a) The given image

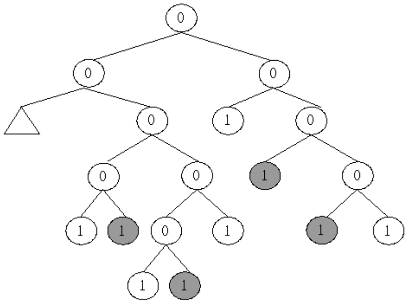
(b) The corresponding S-tree of (a)

Fig. 5. S-tree construction

is not homogeneous, the corresponding node in the S-tree is an internal node with a value "0"; in contrast, if the subimage is homogeneous, the corresponding node is a leaf node with a value "1". Figure 5(a) shows the given image partially divided for the construction of the S-tree. Figure 5(b) shows the resultant S-tree built out of Figure 5(a). The leaf nodes colored gray in Figure 5(b) indicate that the pixel values in the homogeneous blocks are 1's.

After the construction of the S-tree, the given image is encoded by two tables, *tree table* and *color table* (see Figure 5(b)): the tree table records the tree structure by using the depth first search (DFS) technique, and the color table records the colors of the leaf nodes. Now, the new S-tree can further reduce the length of the compressed code (tree table plus color table) by retrenching. The strategy of retrenching is to recursively combine the leaf nodes with their parents until the combination is harmful to the reduction of the compressed code. Since retrenching joins the internal nodes and leaf nodes together, another table, called the *raw data table*, which records the non-homogeneous raw data, is required in the new S-tree. Therefore, there are three possibilities open to each leaf node in the color table of the new S-tree: homogeneous black block, homogeneous white block, and non-homogeneous block, which are denoted by "10", "0", and "11" respectively.

Take Figure 5(b) for example. The three nodes in the triangle encoded by the S-tree consume five bits, where three bits "010" are for the tree traversal by DFS and two bits "10" are for color indication. In contrast, if the three nodes shrink into only one node, the corresponding encoded length is also five with one bit "1" for the leaf node, two bits "11" indicating the non-homogeneous block, and two bits "10" representing the raw data. Since the encoded length of the three nodes by the new S-tree is not any larger than that by the S-tree, the retrenching process goes on upward, and finally the resultant new S-tree of Figure 5(a) is the one shown in Figure 6. Comparing Figure 5(b) and 6, we learn that the encoded length of Figure 5(a) by the new S-tree is reduced from 56 bits to 50 bits.



Tree table: 0010011001110101011
Color table: 110100100010100
Raw data table: 1111000011001100

Fig. 6. The new S-tree structure

3.3 Embedding the Binary Image

The representation of a binary image B with the new S-tree structure is the concatenation of the tree table T, color table C, and raw data table R; that is, $B = T\|C\|R$. The embedding process can be carried out by applying Du and Hsu's method to embed the bit stream B, and therefore the bit stream B should be transformed to its corresponding decimal integer in advance. Then, the order of the combinations of the indices equals to the corresponding decimal integer. Our embedding results are better than Du and Hsu's mainly because of the following three reasons. First, in Du and Hsu's method, the secret image is not compressed properly in advance, so chances are that the transformed unsigned integer contains much redundancy. Second, Du and Hsu use hierarchical clustering to partition the codebook, thus easily making the clustering result undesirable. Finally, our clustering method simultaneously minimizes the variations among the codewords in the same cluster, enlarging the size of each cluster that contains the high frequency codeword. As a result, the embedding capacity of our method offers turns out better. The whole embedding process is specified as follows.

Embedding Algorithm

Input: A cover VQ image $I = \{x_1, x_2, \ldots, x_m\}$ with m codeword indices; the codebook CB; the secret binary image S.

Output: A VQ stego-image.

Step 1: Perform modified *VGA-clustering* algorithm on CB to generate k clusters $C_1, C_2, \ldots, C_k$.

Step 2: Construct the new S-tree for S and generate the corresponding compressed bit stream $B = \{b_1, b_2, \ldots, b_n\}$.

Step 3: Transform B to an unsigned integer, i.e. $B = \sum_{i=1}^{n} b_i \times 2^{n-i}$.

Step 4: Calculate the product $\prod_{i=1}^{p \leq m} |C(x_i)|$, where $C(x_i)$ returns the cluster that x_i belongs to, and p is the minimum integer such that $\prod_{i=1}^{p \leq m} |C(x_i)|$ is greater than or equal to B.

Step 5: Find the corresponding arrangement of codewords for all x_i's such that the serial number of the order of the assignment is equal to B. Output the resultant indices.

The extracting process, basically the reverse of the embedding process, is as follows.

Extracting Algorithm

Input: A cover VQ image $I = \{x_1, x_2, \ldots, x_m\}$ with m codeword indices; the codebook CB with k clusters $C_1, C_2, \ldots, C_k$.

Output: The secret binary image S.

Step 1: Get the order of the arrangement from the consecutive codewords (from left to right, upper to lower) of the image according to the given k clusters. Assign the serial number of the order to B.

Step 2: Transform the unsigned integer B to the secret bits; that is, $B \leftarrow \{b_1, b_2, \ldots, b_n\}$ and $b_i = (B div 2^{n-i}) \bmod 2$.

Step 3: Construct the new S-tree according to B, and finally the binary image can be generated.

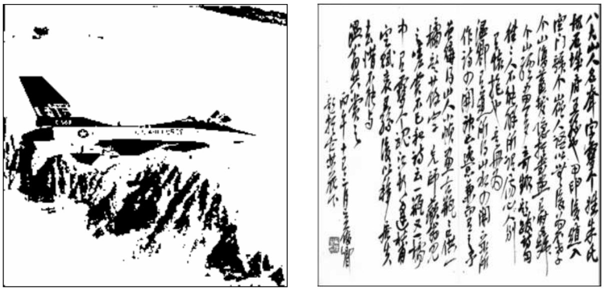
4 Experimental Results

In this section, we shall present some experiments we have conducted. The results are to evaluate the performance of our proposed method. Three standard 512 × 512 gray level images "Lena," "F16," and "Pepper," shown in Figure 7, were used as the cover images in which we hid a random bit-stream or the binary images shown in Figure 8. The codebook sized 512 used in the experiments was generated by the LBG algorithm [15].

Table 1 shows how the PSNR value compares between our method and other schemes after the embedding of the fixed size random data (kbits). The results reveal that the proposed method has performed constantly better than Du and Hsu's method when the same random data size is embedded. On the other

(a) Lena (b) F16 (c) Pepper

Fig. 7. Three cover images



(a) Binary F16 (b) Chinese calligraphy

Fig. 8. Two secret images

Table 1. Performance comparison in PSNR value among various methods for random bit-stream embedding

Embedding bits	PNNE		Du and Hsu's method		Proposed method	
	Lena	F16	Lena	F16	Lena	F16
4K	31.61	31.01	32.19	31.52	32.20	31.54
16K	28.35	27.98	31.60	31.27	31.78	31.31
36K	NA	NA	29.37	28.72	31.18	30.52
64K	NA	NA	25.31	25.07	26.76	26.64
81K	NA	NA	24.02	23.69	26.76	26.53
100K	NA	NA	NA	NA	26.09	25.70
144K	NA	NA	NA	NA	22.94	22.66
196K	NA	NA	NA	NA	22.74	22.56

hand, the PNNE (pairwise nearest-neighbor embedding) method [14] applied the nearest-neighbor rule to pair the closest codewords, resulting in a small embedding capacity. The comparisons of the embedding of the binary images in the cover images are shown in Tables 2 and 3. In Du and Hsu's method, the secret bits were just transformed into unsigned integers, which means still much redundancy remained. In contrast, our method used the new S-tree structure

Table 2. Performance comparison in PSNR value among various methods for embedding "Binary F16"

Embedding bits	Du and Hsu's method			Proposed method		
	Lena	F16	Pepper	Lena	F16	Pepper
4K	32.19	31.51	31.35	32.21	31.55	31.38
16K	31.62	31.26	30.76	32.09	31.50	31.27
36K	29.41	28.80	28.21	32.04	31.47	31.20
64K	25.34	25.09	24.80	31.24	30.87	30.58
81K	24.02	23.69	23.20	31.45	30.93	30.72
100K	NA	NA	NA	31.21	30.85	30.52
144K	NA	NA	NA	30.31	29.55	29.47
196K	NA	NA	NA	30.71	29.74	30.05

Table 3. Performance comparison in PSNR value among various methods for embedding "Chinese calligraphy"

Embedding bits	Du and Hsu's method			Proposed method		
	Lena	F16	Pepper	Lena	F16	Pepper
4K	32.19	31.52	31.35	32.20	31.54	31.37
16K	31.59	31.26	30.75	31.78	31.34	31.15
36K	29.36	28.78	28.18	31.20	31.10	30.72
64K	25.33	25.07	24.88	29.05	28.76	28.40
81K	24.02	23.78	23.27	29.06	27.75	28.44
100K	NA	NA	NA	28.50	27.45	27.85
144K	NA	NA	NA	26.71	26.49	26.14
196K	NA	NA	NA	26.88	26.68	26.39

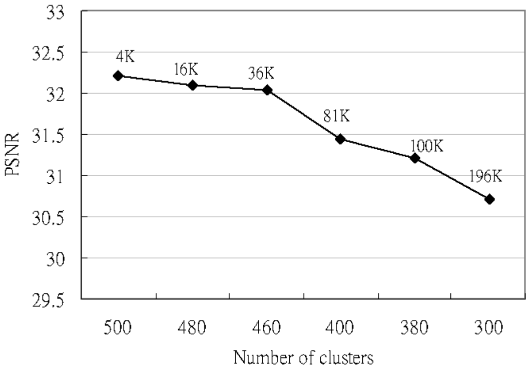
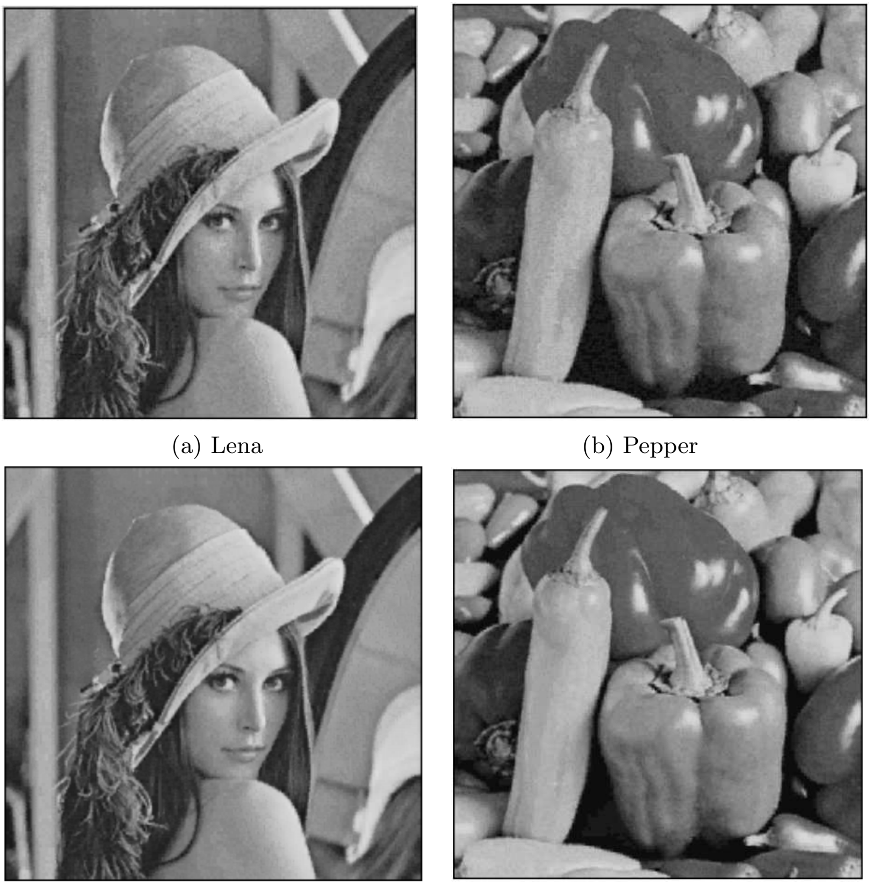


Fig. 9. Relationship between the number of clusters and the PSNR value with "Lena" as the cover image and "Binary F16" as the secret image

to represent the binary image. The results indicate that Du and Hsu's method tends to have a relatively low embedding capacity for the binary image since the secret data was not compressed in advance.

As Tables 1 to 3 reveal, the number of clusters had a strong influence on the embedding capacity and image quality that our method offered. The smaller the number of clusters is, the larger the embedding capacity will be; however, it also results in worse stego-image quality. Figure 9 shows the relationships between the numbers of clusters and the corresponding PSNR values when the size of the "Binary F16" secret image varied. The values above the curve indicate the embedding capacities when the corresponding numbers of clusters were used.

The resultant stego-images are shown in Figure 10. Figure 10(a) and 10(b) obtained by Du and Hsu's method show obvious block effect problems. The differences between the replaced blocks and the new blocks are significant, and the clustering method takes the blame. In contrast, as Figure 10(c) and 10(d) show, the results given by our method are much clearer due to the better clustering effect.



(a) Lena (b) Pepper

(a) Lena (b) Pepper

Fig. 10. Stego-image comparisons after the embedding of a 36K random bit-stream

5 Conclusions

The result of embedding secret data in VQ compressed images can be easily perceptible since changing an old codeword index value of the compressed image to a new one may cause great distortion. In this paper, we have proposed an enhanced version of Du and Hsu's method to make embedding data into VQ compressed images more feasible and practical. In our examples and experiments, the proposed method embeds binary images as secret data, but this concept can be easily extended to the embedding of general data and other low entropy images, such as halftone images or tree structure wavelet images.

Instead of introducing the technique of hierarchical clustering to Du and Hsu's method, the proposed method adopts genetic k-means clustering without inputting the value of k. Hierarchical clustering may lead to bad results when splitting or merging decisions are not well made. On the contrary, genetic k-means clustering guarantees to approach optimal clusters spherically with strong cohesion. The better clusters we get, the better stego-image quality it will achieve. In order to increase the embedding capacity, with the fitness function for genetic k-means clustering, we take into consideration not only the cluster cohesion but also the frequency of each block used in the host image. In other words, the higher the frequency of the codeword used in the host image, the higher the probability is for the codeword to merge with other codewords under acceptable variations.

As the experimental results indicate, both the stego-image quality and the embedding capacity the proposed method offers are better than those produced by Du and Hsu's method. When it comes to the embedding of binary images, the embedding capacity of our method is significantly more than what Du and Hsu's method can give. In addition, our experiments also demonstrate that Du and Hsu's method is highly sensitive to the clustering results. In other words, the clustering method adopted by Du and Hsu's method is not as powerful as the one we use. In summary, the proposed method is suitable for embedding large sized secret data while keeping good quality of the stego-images.

References

1. Anderson, R.J., Petitcolas, F.A.P.: On the limits of steganography. IEEE Journal on Selected Areas in Communications 16, 474–481 (1998)
2. Bandyopadhyay, S., Maulik, U.: Nonparametric genetic clustering: comparison of validity indices. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews 31(1), 120–125 (2001)
3. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Systems Journal 35(3&4), 313–336 (1996)
4. Chang, C.C., Jau, C.C., Chen, T.S.: Chinese calligraphy compression using new S-tree structure. In: Proceedings of International Conference on Chinese Information Processing, Beijing, China, pp. 38–47 (1998)
5. Chang, C.C., Lin, P.Y.: A compression-based data hiding scheme using vector quantization and principle component analysis. In: Proceedings of 2004 International Conference on Cyberworlds, Tokyo, Japan, pp. 369–375 (2004)
6. Chang, C.C., Tseng, H.W.: A steganographic method for digital images using side-match. Pattern Recognition Letters 25(12), 1431–1437 (2004)
7. Du, W.C., Hsu, W.J.: Adaptive data hiding based on VQ compressed images. IEE Proceedings-Vision, Image and Signal Processing 150(4), 233–238 (2003)
8. Gen, M., Cheng, R.: Genetic algorithms and engineering optimization. John Wiley & Sons, Inc., Chichester (2000)
9. Gersho, A., Gray, R.M.: Vector quantization and signal compression. Kluwer Academic Publishers, Dordrecht (1992)
10. Gray, R.M.: Vector quantization. IEEE Transactions on Acoustics, Speech, and Signal Processing 1(2), 4–29 (1984)
11. Jo, M., Kim, H.D.: A digital image watermarking scheme based on vector quantization. IEICE Transactions on Information and Systems E85-D (6), 1054–1056 (2002)

12. Jonge, W.D., Scheuermann, P., Schijf, A.: S$^+$-Trees: An efficient structure for the representation of large pictures. CVGIP: Image Understanding 59(3), 265–280 (1994)
13. Katzenbeisser, S., Petitcolas, F.A.P.: Information hiding techniques for steganography and digital watermarking. Artech House (2000)
14. Lin, Y.C., Wang, C.C.: Digital images watermarking by vector quantization. In: National Computer Symposium, vol. 3, pp. 76–87 (1999)
15. Linde, Y., Buzo, A., Gary, R.M.: An algorithm for vector quantization design. IEEE Transactions on Communications 28, 84–95 (1980)
16. Lu, Z.M., Sun, S.H.: Digital image watermarking technique based on vector quantization. Electronics Letters 36(4), 303–305 (2000)
17. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs. Springer, Heidelberg (1996)
18. Ujjwal, M., Sanghamitra, B.: Genetic algorithm-based clustering technique. Pattern Recognition 33(9), 1455–1465 (2000)
19. Wang, R.Z., Lin, C.F., Lin, J.C.: Image hiding by optimal LSB substitution and genetic algorithm. Pattern Recognition 34(3), 671–683 (2001)

Author Index